Manuale di sicurezza Debian

Javier Fernández-Sanguino Peña <jfs@debian.org>

Manuale di sicurezza Debian

di Javier Fernández-Sanguino Peña

Sommario

Questo documento descrive come viene affrontato il tema sicurezza nel progetto Debian e concretamente nel sistema operativo Debian. Si partirà dal processo di protezione ed irrobustimento dell'installazione predefinita della distribuzione Debian GNU/Linux. Verranno inoltre coperti alcuni usuali compiti per installare un ambiente di rete sicuro utilizzando Debian GNU/Linux, dando inoltre ulteriori informazioni concernenti gli strumenti di sicurezza disponibili e di come la sicurezza sia messa in pratica dal Team Debian per la Sicurezza.

Indice

1. Introduzione 1
    Autore 1
    Dove trovare il manuale (e formati disponibili) 2
    Note/Feedback organizzativi 2
    Conoscenze preliminari 2
    Argomenti da scrivere 3
    Crediti e ringraziamenti! 5
2. Prima di iniziare 7
    A cosa vi serve questo sistema? 7
    Conoscere i problemi generali di sicurezza 7
    Come gestisce la sicurezza Debian? 9
3. Prima e durante l'installazione 10
    Scegliere una password per il BIOS 10
    Partizionare il sistema 10
        Scegliere uno schema di partizionamento intelligente 10
        Selezionare il file system appropriato 11
    Non collegarsi ad Internet finché non si è pronti 12
    Assegnare una password a root 12
    Lanciare i servizi strettamente necessari 12
        Disabilitare i servizi attivi in modalità demone 13
        Disabilitare i servizi gestiti da **inetd** 14
    Installare il software strettamente necessario 15
        Rimuovere Perl 16
    Leggere la mailing list debian security 17
4. Dopo l'installazione 19
    Iscrizione alla mailing list Debian Security Announce 19
    Eseguire un aggiornamento per la sicurezza 19
        Aggiornamento di sicurezza delle librerie 20
        Aggiornamenti di sicurezza per il kernel 21
    Modificare il BIOS (ancora) 22
    Impostazione della password in LILO o GRUB 22
    Disabilitare il prompt di root su initramfs 23
    Rimuovere il prompt root nel kernel 23
    Circoscrivere l'accesso alla console 24
    Circoscrivere la possibilità di riavviare da console 24
    Restricting the use of the Magic SysRq key 25
    Montare le partizioni nel modo giusto 26
        Impostare `/tmp` come noexec 27
        Impostare /usr in sola lettura 27
    Fornire un accesso sicuro per gli utenti 27
        Autenticazione degli utenti: PAM 27
        Password security in PAM 28
        User access control in PAM 29
        User limits in PAM 29
        Control of su in PAM 30
        Temporary directories in PAM 30
        Configuration for undefined PAM applications 30
        Limitare l'uso delle risorse: il file `limits.conf` 31
        User login actions: edit `/etc/login.defs` 32
        User login actions: edit `/etc/pam.d/login` 33
        Restrizioni ftp: modificare il file `/etc/ftpusers` 34

Utilizzo di su 34
Utilizzo di sudo 34
Non permettere accessi per amministrazione remota 34
Restrizioni agli utenti per l'accesso 35
Esame delle attività degli utenti 35
Uno sguardo ai profili utente 37
Impostare delle umask per gli utenti 37
Porre limiti a ciò a cui gli utenti possono accedere 38
Generare password per gli utenti 39
Controllare le password degli utenti 40
Disconnettere gli utenti inattivi 40
Usare i tcpwrapper 40
L'importanza di log e avvisi 41
Usare e personalizzare **logcheck** 42
Configurare il file dove vengono spediti gli avvisi 43
Usare un loghost 43
Permessi dei file di log 44
Includere le patch nel kernel 44
Protezione contro i buffer overflow 46
Patch per la protezione del kernel da Kernel contro buffer overflows 46
Collaudare i programmi contro gli overflow 46
Trasferire file in sicurezza 47
Limitazioni e controllo del File System 47
Usare le quote 47
The ext2 filesystem specific attributes (**chattr**/**lsattr**) 48
Controllare l'integrità del file system 49
Impostare il controllo di setuid 50
Rendere sicuro l'accesso alla rete 50
Configurare le caratteristiche di rete del kernel 50
Configurare i Syncookies 51
Rendere sicura la rete al momento del boot 51
Configurare le caratteristiche di un firewall 54
Disabilitare la questione weak-end host 55
Proteggersi dagli attacchi di tipo ARP 56
Una fotografia del sistema 56
Ulteriori raccomandazioni 58
Non usare software che dipende dalle librerie SVGA (svgalib) 58
5. Rendere più sicuri i servizi che girano sul vostro sistema 59
Rendere sicuro ssh 59
Ssh in chroot 60
Client SSH 61
Non permettere il trasferimento di file 61
Limitare l'accesso al solo trasferimento di file 61
La sicurezza in Squid 61
Rendere sicuro FTP 63
Rendere sicuro l'accesso al sistema X Window 63
Controllare il display manager 64
Rendere sicuri gli accessi alla stampante (specifico per lpd ed lprng) 65
Rendere sicuro il servizio di posta 66
Configurare un nullmailer 66
Fornire un accesso sicuro alle mailbox 67
Ricevere posta in sicurezza 68
Rendere sicuro BIND 68
Configurazione di Bind per evitare abusi 69

Cambiare l'utente di BIND 71
Eseguire il name server in chroot 72
Proteggere Apache 74
Impedire agli utenti la divulgazione di contenuti di rete 75
Permessi sui file di log 75
Pubblicare file web 75
Rendere sicuro finger 75
Paranoie generiche riguardo chroot e suid 76
Creare ambienti chroot automaticamente 76
In generale, paranoia per le password in chiaro 77
Disabilitare NIS 77
Rendere sicuri i servizi RPC 77
Disabilitare completamente i servizi RPC 78
Limitare l'accesso ai servizi RPC 78
Aggiungere funzionalità al firewall 78
Proteggere il sistema locale con un firewall 78
Utilizzare un firewall per proteggere altri sistemi 79
Configurare il firewall 79
6. Irrobustimento automatico di un sistema Debian 87
Harden 87
Bastille Linux 88
7. Infrastrutture per la sicurezza in Debian 89
Il Team Debian per la Sicurezza 89
Avvisi di sicurezza Debian 89
Riferimenti incrociati sulle vulnerabilità 90
Compatibilità con CVE 90
Security Tracker 91
La costruzione dell'infrastruttura di sicurezza in Debian 91
Guida degli sviluppatori agli aggiornamenti sulla sicurezza 92
Firma dei pacchetti in Debian 92
The current scheme for package signature checks 93
Apt sicuro 93
Controllo di rilascio per ogni distribuzione 94
Controllo della versione su fonti esterne a Debian 104
Un modello alternativo di firma per ciascun pacchetto 105
8. Strumenti per la sicurezza in Debian 106
Strumenti per la valutazione delle vulnerabilità da remoto 106
Strumenti per effettuare scansioni di rete 106
Controlli interni 107
Revisione del codice sorgente 107
Rete privata virtuale (VPN) 108
Tunneling punto punto 108
Infrastruttura a chiave pubblica (PKI) 109
Infrastruttura SSL 109
Antivirus 109
GPG 111
9. Linee guida consigliate agli sviluppatori per la sicurezza del sistema operativo 113
Tecniche raccomandate per i controlli sulla sicurezza del software e sulla progettazione di software sicuro 113
Creazione di utenti e gruppi che verranno usati dai demoni 114
10. Prima della compromissione 117
Mantenere sicuro il proprio sistema 117
Mantenersi aggiornati sulle vulnerabilità di sicurezza 117
Aggiornare continuamente il sistema 118

Evitare di usare il ramo instabile 120
Supporto alla sicurezza per il ramo testing 121
Aggiornamento automatico in un sistema Debian GNU/Linux 122
Effettuate periodicamente dei controlli sull'integrità del sistema 123
Pianificare la ricerca di intrusi 123
Individuazione delle intrusioni sulla rete 124
Sistemi per individuare gli intrusi 124
Evitare i root-kit 125
Moduli del kernel caricabili (LKM) 125
Scoprire i root-kit 125
Genius/Paranoia Ideas - what you could do 126
Costruirsi una honeypot ("trappola al miele") 127
11. Dopo la compromissione (reazione agli incidenti) 129
Come comportarsi, in generale 129
Fare una copia di ripristino del sistema 129
Contattate il vostro CERT locale 130
Analisi "patologica" 130
Analisi di codice malevolo 131
12. Domande frequenti (FAQ) 132
La sicurezza nel sistema operativo Debian 132
Debian è più sicura di quella X? 132
Il mio sistema è vulnerabile! (Ne sei sicuro?) 142
Software specifico 145
proftpd is vulnerable to a Denial of Service attack. 145
After installing portsentry, there are a lot of ports open. 145
Domande sul Team per la sicurezza di Debian 145
A. Diario delle Revisioni 146
B. Appendix 158
Il processo di blindatura passo-passo 158
Verifica della configurazione 160
Configurazione ed installazione di un sistema autonomo IDS 163
Impostare un bridge firewall 164
Un bridge con funzionalità NAT e firewall 164
Bridge con funzionalità di firewall 165
Regole base di IPtables 166
Script di esempio per modificare l'installazione predefinita di Bind 166
Aggiornamenti di sicurezza protetti da un firewall 170
Chroot environment for SSH 171
Mettere gli utenti ssh in chroot 172
Eseguire un chroot del server ssh 175
Chroot environment for Apache 185
See also 189

Lista degli esempi

B.1. Regole base di IPtables 166

Capitolo 1. Introduzione

Una delle cose più difficili nello scrivere documenti riguardanti la sicurezza è che ogni caso è unico. Due cose a cui va prestata attenzione sono l'ambiente minaccioso e le necessità di sicurezza del singolo sito, host o rete. Per esempio, le necessità di sicurezza di un utente domestico sono completamente differenti da quelle di una rete bancaria. Mentre il rischio principale che un utente domestico deve affrontare sono i cracker tipo «script kiddie», una rete bancaria deve preoccuparsi di attacchi diretti. Inoltre, la banca deve proteggere i dati dei propri clienti con precisione matematica. In breve, ogni utente deve considerare il compromesso tra usabilità e sicurezza/paranoia.

Occorre tenere presente che questo manuale copre soltanto argomenti relativi al software. Il miglior software del mondo non vi può proteggere se qualcuno ha accesso fisico alla macchina. Si può metterla sotto la scrivania, oppure in un bunker protetto da un esercito. Tuttavia un computer desktop può essere maggiormente sicuro (da un punto di vista software) di uno protetto fisicamente se è configurato correttamente rispetto ad una macchina protetta ma dove il software è pieno di falle di sicurezza. Naturalmente, vanno considerate ambedue le situazioni.

Questo documento dà soltanto uno sguardo a quanto si può fare per incrementare la sicurezza del proprio sistema Debian GNU/Linux. Se avete letto altri documenti riguardanti la sicurezza in Linux, vedrete come argomenti comuni possono sovrapporsi a questo documento. In ogni caso, questo documento non cerca di essere l'ultima risorsa di informazioni di cui si possa avere bisogno, cerca soltanto di adattare queste stesse informazioni così che siano utilizzabili in un sistema Debian GNU/Linux. Distribuzioni diverse fanno alcune cose in modi differenti (per esempio l'avvio dei demoni); qui troverete materiale appropriato per gli strumenti e le procedure di Debian.

Autore

The current maintainer of this document is Javier Fernández-Sanguino Peña. Please forward him any comments, additions or suggestions, and they will be considered for inclusion in future releases of this manual.

This manual was started as a *HOWTO* by Alexander Reelsen. After it was published on the Internet, Javier Fernández-Sanguino Peña incorporated it into the Debian Documentation Project [http://www.debian.org/doc]. A number of people have contributed to this manual (all contributions are listed in the changelog) but the following deserve special mention since they have provided significant contributions (full sections, chapters or appendices):

- Stefano Canepa
- Era Eriksson
- Carlo Perassi
- Alexandre Ratti
- Jaime Robles
- Yotam Rubin
- Frederic Schutz
- Pedro Zorzenon Neto
- Oohara Yuuma

- Davor Ocelic

Dove trovare il manuale (e formati disponibili)

You can download or view the latest version of the Securing Debian Manual from the Debian Documentation Project [https://www.debian.org/doc/user-manuals#securing]. If you are reading a copy from another site, please check the primary copy in case it provides new information. If you are reading a translation, please review the version the translation refers to to the latest version available. If you find that the version is behind please consider using the original copy or review the to see what has changed.

If you want a full copy of the manual you can either download the text version [https://www.debian.org/doc/manuals/securing-debian-manual/securing-debian-manual.en.txt] or the PDF version [https://www.debian.org/doc/manuals/securing-debian-manual/securing-debian-manual.en.pdf] from the Debian Documentation Project's site. These versions might be more useful if you intend to copy the document over to a portable device for offline reading or you want to print it out. Be forewarned, the manual is over two hundred pages long and some of the code fragments, due to the formatting tools used, are not wrapped in the PDF version and might be printed incomplete.

Nel pacchetto http://packages.debian.org/harden-doc è contenuto il documento nei formati testo semplice, html e PDF. Notate, comunque, che il pacchetto potrebbe non essere del tutto aggiornato rispetto al documento fornito sul sito Debian (ma potrete sempre usare il pacchetto sorgente per compilarvi autonomamente una versione aggiornata).

This document is part of the documents distributed by the Debian Documentation Project [https://www.debian.org/doc/ddp]. You can review the changes introduced in the document using a web browser and obtaining information from the version control logs online [https://salsa.debian.org/ddp-team/securing-debian-manual]. You can also checkout the code using Git with the following call in the command line:

```
$ git clone https://salsa.debian.org/ddp-team/securing-debian-manual.git
```

Note/Feedback organizzativi

Ed ora la parte ufficiale. Fino ad ora io (Alexander Reelsen) ho scritto la maggioranza dei paragrafi di questo manuale, ma è mia opinione che non dovrebbe continuare così. Sono cresciuto e vivo con il software libero, è parte del mio uso quotidiano e immagino anche del vostro. Incoraggio tutti a spedirmi feedback, aggiunte od ogni altro tipo di suggerimento che possiate fornirmi.

Se ritenete di poter mantenere un certo capitolo o meglio una sezione, allora scrivete al manutentore del documento e sarete i benvenuti. Specificatamente, se trovate in una sezione dei contrassegni come "FIXME", questo significa che l'autore non ha il tempo o la conoscenza necessaria sull'argomento, inviate un'e-mail immediatamente.

L'argomento di questo manuale rende abbastanza chiara l'importanza di mantenerlo aggiornato e ognuno può fare la propria parte. Per favore, contribuite.

Conoscenze preliminari

The installation of Debian GNU/Linux is not very difficult and you should have been able to install it. If you already have some knowledge about Linux or other Unices and you are a bit familiar with basic security, it will be easier to understand this manual, as this document cannot explain every little detail of a feature (otherwise this would have been a book instead of a manual). If you are not that familiar, however, you might want to take a look at for where to find more in-depth information.

Argomenti da scrivere

Questo paragrafo descrive tutte le cose che devono essere sistemate in questo manuale. Alcuni paragrafi includono i tag *FIXME* o *TODO* per descrivere quale contenuto manca (o quale tipo di lavoro deve essere fatto). Lo scopo di questo paragrafo è descrivere tutte quelle cose che in futuro potrebbero essere incluse in questo manuale o miglioramenti che devono essere fatti (o dovrebbero essere aggiunti).

Se pensate di poter fornire aiuto nel contribuire con contenuti per sistemare alcuni degli elementi della lista (o le note incluse) contattate l'autore principale (sezione chiamata «Autore»).

- This document has yet to be updated based on the latest Debian releases. The default configuration of some packages need to be adapted as they have been modified since this document was written.
- Expand the incident response information, maybe add some ideas derived from Red Hat's Security Guide's chapter on incident response [https://web.archive.org/web/20100412191348/http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/security-guide/ch-response.html].
- Write about remote monitoring tools (to check for system availability) such as monit, daemontools and mon. See Sysamin Guide [https://web.archive.org/web/20100110040204/http://linuxdevcenter.com/pub/a/linux/2002/05/09/sysadminguide.html].
- Considerare l'opportunità di scrivere una sezione riguardante la costruzione di applicazioni di rete basate su Debian (completa di informazioni su sistema di base, equivs e FAI).
- Check if this site [https://web.archive.org/web/20040731082209/http://www.giac.org/practical/gsec/Chris_Koutras_GSEC.pdf] has relevant info not yet covered here.
- Add information on how to set up a laptop with Debian, look here [https://web.archive.org/web/20040725013857/http://www.giac.org/practical/gcux/Stephanie_Thomas_GCUX.pdf].
- Aggiungere informazioni su come installare un firewall usando Debian GNU/Linux. La sezione riguardante il firewalling è attualmente orientata verso un singolo sistema (non proteggendo gli altri...) e inoltre scrivere su come collaudare l'installazione.
- Aggiungere informazioni su come configurare un proxy firewall con Debian GNU/Linux partendo specificatamente da pacchetti che forniscono servizi di proxy (come xfwp, ftp-proxy, redir, smtpd, dnrd, jftpgw, oops, pdnsd, perdition, transproxy, tsocks). Si dovrebbe puntare al manuale per ogni altra informazione. Notate che zorp è ora disponibile come pacchetto Debian ed *è* un proxy firewall (vengono anche forniti pacchetti Debian upstream).
- Informazioni sulla configurazione dei servizi con i file-rc.
- Controllare tutte le URL di riferimento e rimuovere/correggere quelle non più disponibili.
- Aggiungere informazioni sui sostituti disponibili (in Debian) per i server comuni, utili per limitate funzionalità. Per esempio:
  - lpr locale con cups (pacchetto)?
  - lrp remoto con lpr
  - bind con dnrd/maradns
  - apache con dhttpd/thttpd/wn (tux?)
  - exim/sendmail con ssmtpd/smtpd/postfix

  - squid con tinyproxy
  - ftpd con oftpd/vsftp
  - ...
- Maggiori informazioni sulle patch per il kernel concernenti la sicurezza in Debian, incluse quelle mostrate sopra, ed informazioni specifiche su come rendere attive queste patch in un sistema Debian.
  - Linux Intrusion Detection (kernel-patch-2.4-lids)
  - Linux Trustees (nel pacchetto trustees)
  - NSA Enhanced Linux [http://wiki.debian.org/SELinux]
  - linux-patch-openswan
  - ...
- Dettagli su come disattivare servizi di rete non necessari (a parte **inetd**) vengono trattati in parte nelle procedure di irrobustimento ma potrebbero essere estesi un po'.
- Informazioni riguardanti la rotazione delle password che è strettamente collegato alle policy (convenzioni adottate in Debian).
- Policy ed educazione degli utenti al riguardo.
- Maggior dettagli per i tcpwrapper e i wrapper in generale?
- `hosts.equiv` e altri importanti buchi di sicurezza.
- Informazioni sui server di condivisione dei file come Samba ed NFS?
- suidmanager/dpkg-statoverrides.
- lpr e lprng.
- Disabilitare le "cose" IP di GNOME
- Talk about pam_chroot (see http://lists.debian.org/debian-security/2002/05/msg00011.html) and its usefulness to limit users. Introduce information related to https://web.archive.org/web/20031204060940/http://www.securityfocus.com/infocus/1575. pdmenu, for example is available in Debian (whereas flash is not).
- Talk about chrooting services, some more info on this Linux Focus article [http://www.linuxfocus.org/English/January2002/article225.shtml].
- Scrivere in merito a programmi per realizzare gabbie chroot come compartment, chrootuid, makejail e jailer che sono già in Debian.
- Maggiori informazioni su software per l'analisi dei log (per esempio logcheck e logcolorise).
- Routing "avanzato" (le politiche di traffico sono connesse alla sicurezza).
- Limitare l'accesso in **ssh** per eseguire solo alcuni comandi.
- Usare dpkg-statoverride.

- Un modo sicuro per condividere un masterizzatore tra gli utenti.
- Modi sicuri per fornire suoni sulla rete in aggiunta alle capacità di display di rete (così che i suoni dei client X siano eseguiti sull'hardware del server X).
- Rendere sicuri i web browser.
- Impostare ftp su **ssh**.
- Usare un loopback file system crittografato.
- encrypting the entire file system.
- Strumenti steganografici.
- Impostare un PKA per un'organizzazione.
- Utilizzare LDAP per gestire gli utenti. Esiste un HOWTO di ldap+kerberos per Debian presso http://www.bayour.com, scritto da Turbo Fredrikson.
- Come rimuovere le informazioni di scarsa utilità nei sistemi in produzione come `/usr/share/doc`, `/usr/share/man` (sì, sicurezza tramite riservatezza).
- Maggiori informazioni su lcap basate sul file README dei pacchetti (bene, non ancora, vedete il http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=169465) e dall'articolo da LWN: http://lwn.net/1999/1202/kernel.php3.
- Add Colin's article on how to setup a chroot environment for a full sid system (https://web.archive.org/web/20030204012846/https://people.debian.org/~walters/chroot.html).
- Aggiungere informazioni su come attivare più sensori **snort** in un dato sistema (controllare i rapporti sui bachi spediti da snort).
- Aggiungere informazioni su come configurare una honeypot (honeyd).
- Descrivere la situazione wrt rispetto a FreeSwan (orfano) ed OpenSwan. La sezione VPN necessita di una riscrittura.
- Aggiungere una sezione specifica sui database, la loro corrente installazione predefinita e come garantirne un accesso sicuro.
- Aggiungere una sezione che si occupi dei vantaggi circa l'utilizzo dei server virtuali (come Xen e gli altri).
- Spiegare come utilizzare degli analizzatori d'integrità (AIDE, integrit o samhain). Le basi sono semplici e si potrebbero anche spiegare i dettagli di alcune buone configurazioni.

Crediti e ringraziamenti!

- Alexander Reelsen ha scritto il documento originale.
- added more info to the original doc.
- Robert van der Meulen ha fornito i paragrafi su quota e molte altre ottime idee.
- Ethan Benson ha corretto il paragrafo su PAM ed ha avuto alcune buone idee.

- Dariusz Puchalak ha contribuito con informazioni in diversi capitoli.
- Gaby Schilders ha contribuito con una simpatica idea su Genius/Paranoia.
- Era Eriksson ha raffinato il linguaggio in un gran numero di sezioni ed ha contribuito all'appendice checklist.
- Philipe Gaspar ha scritto le informazioni su LKM.
- Yotam Rubin ha contribuito correggendo molti errori di battitura e anche fornendo le informazioni concernenti le versioni di bind e le password MD5.
- Francois Bayart ha fornito l'appendice che descrive come realizzare un bridge firewall.
- Joey Hess sul http://wiki.debian.org/SecureApt ha scritto la sezione che descrive come lavora Secure Apt.
- Martin F. Krafft ha scritto sul suo blog alcune informazioni circa le verifiche fingerprint che sono state anche riutilizzate per la sezione su Secure Apt.
- Francesco Poli ha svolto un'approfondita revisione del manuale ed ha fornito molte segnalazioni ed errori di battitura che hanno contribuito a migliorare ed aggiornare il documento.
- Tutte le persone che hanno fornito suggerimenti per miglioramenti che (alla fine) sono state incluse qui (vedete in sezione chiamata «Dove trovare il manuale (e formati disponibili)»).
- (Alexander) Tutte le persone che mi hanno incoraggiato a scrivere questo HOWTO (che successivamente si è trasformato in un manuale).
- L'intero progetto Debian.

Capitolo 2. Prima di iniziare

A cosa vi serve questo sistema?

Proteggere Debian non è molto diverso dal proteggere qualsiasi altro sistema; per farlo correttamente, bisogna prima decidere che cosa ci si intenda fare, dopo di che, se si vuole un sistema sicuro, si dovrà considerare la necessità di occuparsi dei compiti descritti più avanti.

Vi accorgerete che questo manuale è scritto partendo dalla fine, cioè leggerete alcune informazioni sui compiti da svolgere prima, durante e dopo l'installazione del vostro sistema Debian. Tali compiti possono essere definiti come:

- Decidere quali sono i servizi necessari e fare in modo che il sistema esegua soltanto quelli. Questo include il disattivare/disinstallare servizi superflui e aggiungere filtri come i firewall o i tcpwrappers.
- Limitare gli utenti e i permessi del sistema.
- Irrobustire i servizi forniti in modo tale che, in caso di un loro malfunzionamento, l'impatto sul sistema sia ridotto al minimo.
- Utilizzare gli strumenti atti a rilevare utilizzi non autorizzati, in modo da poter prendere le opportune contromisure.

Conoscere i problemi generali di sicurezza

Questo manuale (per lo più) non entra nei dettagli sul perché alcune richieste siano considerate a rischio per la sicurezza. Tuttavia potreste desiderare acquisire una maggiore preparazione sulla sicurezza generale su UNIX e (specifica) in Linux. Dedicate un po' di tempo alla lettura di documenti riguardanti la sicurezza, per prendere decisioni informate di fronte a scelte diverse. Debian GNU/Linux si basa sul kernel Linux, quindi vi si applicano molte delle informazioni disponibili per Linux, anche di altre distribuzioni, e sulla sicurezza in generale degli UNIX (anche se sono differenti gli strumenti usati o i programmi disponibili).

Alcuni documenti utili sono:

- The http://www.tldp.org/HOWTO/Security-HOWTO/ is one of the best references regarding general Linux security.
- The http://www.tldp.org/HOWTO/Security-Quickstart-HOWTO/ is also a very good starting point for novice users (both to Linux and security).
- The http://seifried.org/lasg/ is a complete guide that touches all the issues related to security in Linux, from kernel security to VPNs. Note that it has not been updated since 2001, but some information is still relevant. [1]
- Kurt Seifried's http://seifried.org/security/os/linux/20020324-securing-linux-step-by-step.html.
- In http://www.tldp.org/links/p_books.html#securing_linux you can find a similar document to this manual but related to Red Hat, some of the issues are not distribution-specific and also apply to Debian.
- Another Red Hat related document is https://web.archive.org/web/20050520170309/https://ltp.sourceforge.net/docs/RHEL-EAL3-Configuration-Guide.pdf.

[1] At a given time it was superseded by the "Linux Security Knowledge Base". This documentation is also provided in Debian through the lskb package. Now it's back as the *Lasg* again.

- IntersectAlliance has published some documents that can be used as reference cards on how to harden Linux servers (and their services), the documents are available at https://web.archive.org/web/20030210231943/http://www.intersectalliance.com/projects/index.html.

- For network administrators, a good reference for building a secure network is the https://web.archive.org/web/20030418093551/http://www.linuxsecurity.com/docs/LDP/Securing-Domain-HOWTO/.

- Se volete provare i programmi che state per utilizzare (o se volete elaborarne dei nuovi), dovreste leggere http://www.tldp.org/HOWTO/Secure-Programs-HOWTO/, che include illustrazioni e brani parlati dell'autore, David Wheeler, la cui stesura originale è disponibile all'indirizzo http://www.dwheeler.com/secure-programs/.

- If you are considering installing firewall capabilities, you should read the http://www.tldp.org/HOWTO/Firewall-HOWTO.html and the http://www.tldp.org/HOWTO/IPCHAINS-HOWTO.html (for kernels previous to 2.4).

- Finally, a good card to keep handy is the https://web.archive.org/web/20030308013020/http://www.linuxsecurity.com/docs/QuickRefCard.pdf.

In any case, there is more information regarding the services explained here (NFS, NIS, SMB...) in many of the HOWTOs of the http://www.tldp.org/. Some of these documents speak on the security side of a given service, so be sure to take a look there too.

The HOWTO documents from the Linux Documentation Project are available in Debian GNU/Linux through the installation of the doc-linux-text (text version) or doc-linux-html (HTML version). After installation these documents will be available at the `/usr/share/doc/HOWTO/en-txt` and `/usr/share/doc/HOWTO/en-html` directories, respectively.

Altri libri suggeriti su Linux:

- Maximum Linux Security (Il massimo della sicurezza in Linux): la guida di un hacker per proteggere i vostri server e le vostre reti Linux. Anonimo. Tascabile - 829 pagine. Sams Publishing. ISBN: 0672313413. Luglio 1999.

- Linux Security (Sicurezza in Linux), di John S. Flowers. New Riders; ISBN: 0735700354. marzo 1999.

- https://web.archive.org/web/20030202131658/https://www.linux.org/books/ISBN_0072127732.html By Brian Hatch. McGraw-Hill Higher Education. ISBN 0072127732. April, 2001

Altri libri (che possono riferirsi a esigenze generali su UNIX e sulla sicurezza, non specifiche per Linux):

- https://web.archive.org/web/20030206231652/http://www.oreilly.com/catalog/puis/ Garfinkel, Simpson, and Spafford, Gene; O'Reilly Associates; ISBN 0-56592-148-8; 1004pp; 1996.

- Firewalls and Internet Security (Firewall e sicurezza internet), Cheswick, William R. e Bellovin, Steven M.; Addison-Wesley; 1994; ISBN 0-201-63357-4; 320pp.

Alcuni siti web utili per mantenersi aggiornati sulla sicurezza:

- http://csrc.nist.gov/.

- https://cve.mitre.org/data/refs/refmap/source-BUGTRAQ.html CVE Reference Map for Source BUGTRAQ

- http://www.linuxsecurity.com/. General information regarding Linux security (tools, news...). Most useful is the https://linuxsecurity.com/howtos page.

Come gestisce la sicurezza Debian?

Poiché avete già una vista d'insieme generale della sicurezza in Debian GNU/Linux, dovreste prendere nota di come Debian affronta i diversi problemi per fornire un sistema completo e sicuro sotto tutti i punti di vista:

- Debian problems are always handled openly, even security related. Security issues are discussed openly on the debian-security mailing list. Debian Security Advisories (DSAs) are sent to public mailing lists (both internal and external) and are published on the public server. As the http://www.debian.org/social_contract states: *We will not hide problems. We will keep our entire bug report database open for public view at all times. Reports that people file online will promptly become visible to others.*

- Debian follows security issues closely. The security team checks many security related sources, the most important being http://www.securityfocus.com/cgi-bin/vulns.pl, on the lookout for packages with security issues that might be included in Debian.

- Gli aggiornamenti di sicurezza sono la prima priorità. Quando sorge un problema di sicurezza in un pacchetto Debian, l'aggiornamento viene preparato quanto più velocemente possibile e distribuito per le versioni stable, testing ed unstable, incluse tutte le architetture.

- Le informazioni riguardanti la sicurezza sono centralizzate in un'unico punto, http://security.debian.org/.

- Debian prova sempre a migliorare la sicurezza complessiva della distribuzione all'avvio di ogni nuovo progetto, come il meccanismo di verifica automatica della firma dei pacchetti.

- Debian fornisce un certo numero di utili strumenti relativi alla sicurezza per l'amministrazione e il monitoraggio del sistema. Gli sviluppatori tentano di integrare nel miglior modo possibile questi strumenti con la distribuzione, al fine di renderli uno strumento più efficace per applicare le politiche di sicurezza locali. Gli strumenti includono: verificatori d'integrità, strumenti di verifica, d'irrobustimento, strumenti per i firewall, per l'individuazione delle intrusioni, etc.

- I manutentori dei pacchetti vengono avvertiti dei problemi di sicurezza. Ciò porta a molte installazioni di servizi "sicuri in modo predefinito", cosa che può comportare, talvolta, alcuni limiti al loro normale uso. Comunque Debian tenta di bilanciare la sicurezza con la facilità d'amministrazione, ad esempio i programmi non vengono installati disattivati, come invece accade nella famiglia di distribuzioni BSD. In ogni caso, potenzialmente importanti problemi di sicurezza, come i programmi con il `setuid` attivo, sono parte della http://www.debian.org/doc/debian-policy/.

Questo documento, comunque, cerca di aiutare a realizzare installazioni ragionevolmente sicure, pubblicando informazioni specifiche su come affrontare la sicurezza in Debian e completandole con informazioni più generiche, ma sempre correlate alla sicurezza in Debian (vedete in sezione chiamata «Conoscenze preliminari»).

Capitolo 3. Prima e durante l'installazione

Scegliere una password per il BIOS

Prima d'installare qualsiasi sistema operativo sul vostro computer, è preferibile inserire una password al BIOS. Dopo l'installazione, non appena attiverete l'avvio dall'hard disk, è preferibile impostare la sequenza d'avvio da BIOS disabilitando l'avvio da floppy, da cdrom e dalle altre periferiche che non dovrebbero avviare il computer. Altrimenti ad un cracker basterebbe solo avere l'accesso fisico al computer ed un disco di boot per accedere al sistema.

Disabilitare l'avvio se non viene inserita una password è ancora meglio. Questo può essere molto efficace se avete un server, visto che non viene riavviato frequentemente. Il lato negativo di questa precauzione è che il riavvio della macchina richiede l'intervento umano e può essere problematico se la macchina non è facilmente accessibile.

Notate: molti BIOS hanno una master password (o password universale), che può essere facilmente rinvenuta, senza contare che inoltre esistono delle applicazioni che sono in grado di recuperare la password del BIOS rapidamente. Corollario: si deduce facilmente che non dipende, principalmente, da queste precauzioni la sicurezza dell'accesso al sistema.

Partizionare il sistema

Scegliere uno schema di partizionamento intelligente

Scegliere uno schema di partizionamento intelligente dipenderà da come verrà usata la macchina. È buona regola creare delle partizioni sufficientemente grandi e prestare attenzione ai seguenti fattori:

- Any directory tree which a user has write permissions to, such as e.g. `/home`, `/tmp` and `/var/tmp/`, should be on a separate partition. This reduces the risk of a user DoS by filling up your "/" mount point and rendering the system unusable (Note: this is not strictly true, since there is always some space reserved for root which a normal user cannot fill), and it also prevents hardlink attacks. [1]

- Ogni directory di dimensione variabile es. `/var` (specialmente `/var/log`) deve stare su una partizione separata. Attenzione, su un sistema Debian, dovreste creare la directory `/var` leggermente più grande di altri sistemi, perché i pacchetti scaricati (nella cache di apt) vengono conservati in `/var/cache/apt/archives`.

- Tutte le directory dove volete installare del software che non appartiene alla distribuzione devono stare in partizioni separate. Secondo File Hierarchy Standard (Gerarchia Standard dei File system) queste directory sono `/opt` oppure `/usr/local`. Se queste directory stanno su partizioni separate, non verranno cancellate se reinstallaste di nuovo Debian.

- Dal punto di vista della sicurezza è importante cercare di mettere i dati statici in partizioni proprie, montate in sola lettura. Meglio ancora se questi dati vengono messi su supporti di sola lettura. Vedete più avanti in questo capitolo per maggiori dettagli.

[1] A very good example of this kind of attacks using /tmp is detailed in http://www.hackinglinuxexposed.com/articles/20031111.html and http://www.hackinglinuxexposed.com/articles/20031214.html (notice that the incident is Debian-related). It is basicly an attack in which a local user *stashes* away a vulnerable setuid application by making a hard link to it, effectively avoiding any updates (or removal) of the binary itself made by the system administrator. Dpkg was recently fixed to prevent this (see http://bugs.debian.org/225692) but other setuid binaries (not controlled by the package manager) are at risk if partitions are not setup correctly.

Nel caso in cui gestiate un mail server è importante avere una partizione separata per la directory spool delle mail. Gli utenti remoti (sia consapevolmente che inconsapevolmente) possono riempire la directory mail spool (`/var/mail` o `/var/spool/mail`). Se la directory spool è in una partizione separata, questa eventualità non bloccherà il sistema . Altrimenti, se la directory spool è sulla stessa partizione di `/var`, il sistema avrà gravi problemi quali: non potrete creare i file di log, non potrete installare pacchetti aggiuntivi e dei programmi (se usano la directory `/var/run`) avranno problemi a partire o verranno rallentati.

Inoltre, per le partizioni per le quali non siete sicuri dello spazio che occorre, è preferibile installare Logical Volume Manager (lvm-common ed i pacchetti binari per il vostro kernel, questi possono essere lvm10, lvm6, oppure lvm5). Usando `lvm` si possono creare gruppi di volumi che possono occupare più volumi fisici multipli.

Selezionare il file system appropriato

During the system partitioning you also have to decide which file system you want to use. The default file system[2] selected in the Debian installation for Linux partitions is `ext3`, a journaling file system. It is recommended that you always use a journaling file system, such as `ext3`, `reiserfs`, `jfs` or `xfs`, to minimize the problems derived from a system crash in the following cases:

- Per i computer portatili, in tutti i file system installati. In questo modo se le batterie si esaurissero inaspettatamente o se il sistema si bloccasse per una questione relativa all'hardware (come nel caso della configurazione di X, che è piuttosto comune) la perdita di dati durante un riavvio hardware sarebbe meno probabile.

- Per i sistemi in produzione che registrano grandi quantità di dati (come i server di posta, i server ftp, i file system di rete...) si raccomanda un file system journalling sulle partizioni interessate. In questo modo, in caso di crash del sistema, il server impiegherà meno tempo a riavviarsi e controllare i file system, inoltre la perdita di dati sarà meno probabile.

Lasciando da parte le discussioni sulle prestazioni dei file system journalling (visto che talvolta possono trasformarsi in guerre di religione), è meglio, generalmente, usare il file system `ext3`. Il motivo è che questo è compatibile all'indietro con `ext2`, così se si dovessero presentare problemi con il journalling, è possibile disabilitarlo e avere comunque un file system funzionante. Inoltre, in caso si debba ripristinare il sistema con un dischetto di avvio (o CDROM) non serve un kernel modificato. Se il kernel è un 2.4 il supporto per `ext3` è già disponibile, se il kernel è un 2.2, pur perdendo le caratteristiche del journalling sarete in grado di avviare il file system. Invece, nel caso stiate usando altri tipi di file system journalling potreste non essere in grado di ripristinare, a meno che non abbiate un kernel 2.4 o 2.6 con i moduli necessari all'avvio già compilati. Se avete un dischetto di salvataggio con un kernel 2.2 potrebbe essere ancora più difficile accedere a `reiserfs` o `xfs`.

In ogni caso, l'integrità dei dati potrebbe essere meglio garantita da `ext3` visto che esegue un file-data journalling mentre gli altri eseguono soltanto il meta-data journalling, vedete in http://lwn.net/2001/0802/a/ext3-modes.php3.

Notice, however, that there are some partitions that might not benefit from using a journaling filesystem. For example, if you are using a separate partition for `/tmp/` you might be better off using a standard `ext2` filesystem as it will be cleaned up when the system boots.

[2] Since Debian GNU/Linux 4.0, codename `etch`

Non collegarsi ad Internet finché non si è pronti

Il sistema non dovrebbe essere immediatamente connesso ad internet durante l'installazione. Questo potrebbe suonare sciocco, ma l'installazione di rete è un metodo comune. Poiché il sistema, installandosi, attiva immediatamente dei servizi, se è connesso ad Internet ed i servizi non sono opportunamente configurati, sarà esposto a possibili attacchi.

Inoltre alcuni servizi potrebbero avere problemi di sicurezza non ancora corretti nei pacchetti usati per l'installazione. Questo accade di solito quando si esegue l'installazione con media vecchi (come i CD-ROM). In questo caso, il sistema potrebbe addirittura essere compromesso prima che si finisca l'installazione!

Poiché l'installazione e l'aggiornamento di Debian possono essere effettuati via Internet, si potrebbe pensare che usare questa caratteristica sia una buona idea. Se il sistema verrà direttamente connesso a Internet (e non sarà protetto da firewall o NAT), è preferibile installare senza essere connessi a Internet, usando un mirror locale dei pacchetti sia per i sorgenti dei pacchetti Debian sia per gli aggiornamenti di sicurezza. Un mirror dei pacchetti per fornire gli archivi al sistema può essere preparato usando un altro sistema connesso ad Internet con strumenti Debian specifici (se è un sistema Debian) come apt-move o apt-proxy, o altri comuni strumenti per il mirroring. Se non è possibile fare questo, potete attivare regole per il firewall che limitino l'accesso al sistema durante l'aggiornamento (vedete in sezione chiamata «Aggiornamenti di sicurezza protetti da un firewall»).

Assegnare una password a root

Setting a good root password is the most basic requirement for having a secure system. See passwd(1) for some hints on how to create good passwords. You can also use an automatic password generation program to do this for you (see sezione chiamata «Generare password per gli utenti»).

Plenty of information on choosing good passwords can be found on the Internet; two that provide a decent summary and rationale are Eric Wolfram's http://wolfram.org/writing/howto/password.html and Walter Belgers' https://web.archive.org/web/20030218000949/http://www.belgers.com/write/pwseceng.txt

Lanciare i servizi strettamente necessari

I servizi sono programmi come i server ftp e i server web. Poiché restano nello stato *di attesa (listening)* di connessioni in ingresso che richiedono il servizio, i computer all'esterno possono collegarvisi. A volte i servizi sono vulnerabili (es: possono essere compromessi da un determinato attacco) e possono rappresentare un rischio per la sicurezza.

Non dovreste installare servizi che non sono necessari sulla vostra macchina. Ogni servizio installato può introdurre nuovi, magari non evidenti (o conosciuti) buchi di sicurezza sul vostro computer.

Come forse già saprete, quando si installa un determinato servizio, l'opzione predefinita è che sia attivo. In un'installazione Debian predefinita, senza servizi installati, l'insieme dei servizi attivi è piuttosto basso e questo è ancora più vero per quanto riguarda i servizi di rete. In una classica installazione di Debian 3.1, alla fine vi troverete a disposizione come servizi di rete OpenSSH, Exim (a seconda di come l'avrete configurato) e l'RPC portmapper [3]. Se invece non farete un'installazione standard ma opterete per eseguire quella expert, potreste trovarvi alla fine senza servizi di rete attivi. Dato un sistema, l'RPC portmapper viene installato in modo predefinito perché è necessario per offrire molti servizi, come per esempio NFS.

[3] Lo schema, in Debian 3.0 e nei rilasci precedenti, non era così chiuso, dal momento che qualche servizio **inetd** veniva abilitato in modo predefinito. Anche le installazioni standard di Debian 2.2 installavano sia il server NFS che quello telnet.

Comunque, l'RPC portmapper può essere facilmente rimosso, vedete per maggiori informazioni sezione chiamata «Rendere sicuri i servizi RPC» su come rendere sicuri o disabilitare i servizi RPC.

Quando installate un nuovo servizio di rete (demone) in Debian GNU/Linux, questo può essere attivato in 2 modi: per mezzo del superdemone **inetd** (una riga verrà aggiunta a `/etc/inetd.conf`) o per mezzo di un programma indipendente che si collega all'interfaccia di rete. I programmi indipendenti vengono controllati tramite i file in `/etc/init.d`, che a loro volta vengono chiamati al momento del boot dal meccanismo SysV (o uno alternativo) usando un collegamento simbolico in `/etc/rc?.d/*` (per maggiori informazioni su come funzioni leggete `/usr/share/doc/sysvinit/README.runlevels.gz`).

Se volete mantenere alcuni servizi ma usarli raramente, utilizzate i comandi di aggiornamento come, per esempio: **update-inetd** e **update-rc.d** per rimuoverli dal processo di avvio. Per ulteriori informazioni su come disabilitare i servizi di rete vedete sezione chiamata «Disabilitare i servizi attivi in modalità demone». Se volete cambiare il comportamento predefinito, che prevede di abilitare l'avvio dei servizi una volta che i pacchetti ad essi associati vengano installati [4], dovete usare **policy-rc.d**, per favore, leggete `/usr/share/doc/sysv-rc/README.policy-rc.d.gz` per ulteriori chiarimenti.

invoke-rc.d support is mandatory in Debian, which means that for Debian 4.0 *etch* and later releases you can write a policy-rc.d file that forbids starting new daemons before you configure them. Although no such scripts are packaged yet, they are quite simple to write. See policyrcd-script-zg2.

Disabilitare i servizi attivi in modalità demone

Disabling a daemon service is quite simple. You either remove the package providing the program for that service or you remove or rename the startup links under `/etc/rc${runlevel}.d/`. If you rename them make sure they do not begin with 'S' so that they don't get started by **/etc/init.d/rc**. Do not remove all the available links or the package management system will regenerate them on package upgrades, make sure you leave at least one link (typically a 'K', i.e. kill, link). For more information read http://www.debian.org/doc/manuals/reference/ch-system.en.html#s-custombootscripts section of the Debian Reference (Chapter 2 - Debian fundamentals).

You can remove these links manually or using `update-rc.d` (see update-rc.d(8)). For example, you can disable a service from executing in the multi-user runlevels by doing:

```
# update-rc.d name stop XX 2 3 4 5 .
```

Where *XX* is a number that determines when the stop action for that service will be executed. Please note that, if you are *not* using file-rc, `update-rc.d -f service remove` will not work properly, since *all* links are removed, upon re-installation or upgrade of the package these links will be re-generated (probably not what you wanted). If you think this is not intuitive you are probably right (see http://bugs.debian.org/67095). From the manpage:

```
If any files /etc/rcrunlevel.d/[SK]??name already exist then
update-rc.d does nothing.  This is so that the system administrator
can rearrange the links, provided that they leave at least one
link remaining, without having their configuration overwritten.
```

Se usate file-rc, tutte le informazioni che riguardano l'avvio dei servizi verranno gestite da un file di configurazione comune e vengono conservate anche se i pacchetti sono disinstallati dal sistema.

È possibile usare la TUI (Text User Interface, cioè interfaccia utente in modalità testuale) fornita da sysv-rc-conf per apportare queste modifiche facilmente (**sysv-rc-conf** funziona sia con file-rc che con i nor-

[4] Questo potrebbe essere ciò che desiderate nel caso in cui stiate configurando un ambiente chroot per motivi di sviluppo, per esempio.

mali runlevel del System V). Si trovano anche interfacce grafiche simili per i sistemi desktop. Altrimenti potete anche usare la riga di comando di sysv-rc-conf:

```
# sysv-rc-conf foobar off
```

Il vantaggio nell'usare questo programma è che i collegamenti in rc.d verranno ripristinati a com'erano prima della disattivazione con la chiamata 'off' nel caso vengano riabilitati con:

```
# sysv-rc-conf foobar on
```

Other (less recommended) methods of disabling services are:

- Removing the `/etc/init.d/service_name` script and removing the startup links using:

  ```
  # update-rc.d name remove
  ```

- Move the script file (`/etc/init.d/service_name`) to another name (for example `/etc/init.d/OFF.service_name`). This will leave dangling symlinks under `/etc/rc${runlevel}.d/` and will generate error messages when booting up the system.

- Remove the execute permission from the `/etc/init.d/service_name` file. That will also generate error messages when booting.

- Edit the `/etc/init.d/service_name` script to have it stop immediately once it is executed (by adding an **exit 0** line at the beginning or commenting out the `start-stop-daemon` part in it). If you do this, you will not be able to use the script to startup the service manually later on.

Nevertheless, the files under `/etc/init.d` are configuration files and should not get overwritten due to package upgrades if you have made local changes to them.

Unlike other (UNIX) operating systems, services in Debian cannot be disabled by modifying files in `/etc/default/service_name`.

FIXME: Aggiungere informazioni su come gestire i demoni con file-rc.

Disabilitare i servizi gestiti da inetd

Considerate bene se al giorno d'oggi avete davvero bisogno del demone **inetd**. Inetd è sempre stato un modo per rimediare a deficienze del kernel, che però sono state eliminate nei moderni kernel Linux. Contro **inetd** si possono effettuare degli attacchi di tipo "Denial of Service" (che possono aumentare considerevolmente il carico della macchina) e molti preferiscono usare demoni autonomi piuttosto che lanciare servizi tramite **inetd**. Comunque, se volete far girare un qualche tipo di servizio **inetd**, dovreste almeno passare a un demone Inet più configurabile, come **xinetd**, **rlinetd** o **openbsd-inetd**.

Sarebbe bene arrestare tutti i servizi non necessari nel proprio sistema, come **echo**, **chargen**, **discard**, **daytime**, **time**, **talk**, **ntalk** e gli r-services (**rsh**, **rlogin** e **rcp**) che vengono considerati MOLTO insicuri (meglio usare **ssh** al loro posto).

Potete disabilitare dei servizi modificando direttamente `/etc/inetd.conf`, ma Debian fornisce un'alternativa migliore: `update-inetd` (che commenta i servizi in modo che possano essere facilmente riattivati). Potete rimuovere il demone **telnet** eseguendo il comando seguente, per cambiare il file di configurazione e riavviare il demone (in questo caso il servizio **telnet** viene disabilitato):

```
/usr/sbin/update-inetd --disable telnet
```

Se davvero desiderate avere servizi in ascolto, ma non volete che stiano in ascolto su tutti gli indirizzi IP del vostro host, probabilmente vorrete utilizzare una funzione non documentata di **inetd** (sostituire il nome del servizio con una sintassi del tipo servizio@ip) o usare un demone **inetd** alternativo, come **xinetd**.

Installare il software strettamente necessario

Debian comes with *a lot* of software, for example the Debian 3.0 *woody* release includes 6 or 7 (depending on architecture) CD-ROMs of software and thousands of packages, and the Debian 3.1 *sarge* release ships with around 13 CD-ROMs of software. With so much software, and even if the base system installation is quite reduced [5] you might get carried away and install more than is really needed for your system.

Poiché sapete già a cosa servirà il sistema (o no?), dovreste installare solo il software che è realmente necessario per farlo funzionare. Qualsiasi tool non necessario ma installato potrebbe essere usato da un utente che vuole compromettere il sistema o da un intruso esterno che ha ottenuto l'accesso ad una shell (o l'esecuzione di codice da remoto attraverso un servizio che lo consenta).

La presenza, ad esempio, di utility di sviluppo (un compilatore per il linguaggio C) o di linguaggi interpretati (come **perl** - ma vedete sotto -, **python**, **tcl**...) potrebbe rendere le cose più semplici a colui che attacca per compromettere il sistema, in particolare:

- Permettendogli di acquisire privilegi. È più facile, ad esempio, mandare in esecuzione exploit locali nel sistema se ci sono un debugger ed un compilatore pronti a compilarli ed a collaudarli!
- providing tools that could help the attacker to use the compromised system as a *base of attack* against other systems. [6]

Naturalmente un intruso con accesso locale ad una shell può scaricare gli strumenti che gli servono ed eseguirli, ma anche la shell stessa può essere usata per scrivere programmi complessi. Rimuovere il software non necessario non aiuterà a *prevenire* il problema, ma renderà un po' più difficile l'azione dell'intruso (ed alcuni potrebbero rinunciare se si trovano in questa situazione, cercando un bersaglio più facile). Quindi, lasciando installati detti strumenti su un sistema in produzione, che può essere usato per attaccare sistemi da remoto (vedete sezione chiamata «Strumenti per la valutazione delle vulnerabilità da remoto») è lecito aspettarsi che un intruso li utilizzi, se disponibili.

Notate che un'installazione predefinita di Debian *sarge* (come in un'installazione in cui non è stato selezionato alcun pacchetto) installerà un certo numero di pacchetti di sviluppo, di solito inutili. Ciò avviene perché alcuni pacchetti di sviluppo hanno priorità *Standard*. Se non pensate di fare sviluppo, potete rimuovere in tutta sicurezza i seguenti pacchetti dal vostro sistema, e questo aiuterà anche a liberare un po' di spazio:

```
Package                    Size
-------------------------+--------
gdb                     2,766,822
```

[5] For example, in Debian woody it is around 400-500 Mbs, try this:

```
$ size=0
$ for i in `grep -A 1 -B 1 "^Section: base" /var/lib/dpkg/available |
grep -A 2 "^Priority: required" |grep "^Installed-Size" |cut -d : -f 2
`; do size=$(($size+$i)); done
$ echo $size
47762
```

[6] Many intrusions are made just to get access to resources to do illegitimate activity (denial of service attacks, spam, rogue ftp servers, dns pollution...) rather than to obtain confidential data from the compromised system.

```
gcc-3.3                 1,570,284
dpkg-dev                  166,800
libc6-dev               2,531,564
cpp-3.3                 1,391,346
manpages-dev            1,081,408
flex                      257,678
g++                         1,384 (Note: virtual package)
linux-kernel-headers    1,377,022
bin86                      82,090
cpp                        29,446
gcc                         4,896 (Note: virtual package)
g++-3.3                 1,778,880
bison                     702,830
make                      366,138
libstdc++5-3.3-dev        774,982
```

Questo è stato corretto nelle versioni post-sarge, vedete il http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=301273 ed il http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=301138. A causa di un bug nel sistema di installazione, questo non è successo nei sistemi installati mediante l'installer della versione di Debian 3.0 *woody*.

Rimuovere Perl

You must take into account that removing **perl** might not be too easy (as a matter of fact it can be quite difficult) in a Debian system since it is used by many system utilities. Also, the perl-base is *Priority: required* (that about says it all). It's still doable, but you will not be able to run any **perl** application in the system; you will also have to fool the package management system to think that the perl-base is installed even if it's not. [7]

Quali sono le utility che usano **perl**? Potete scoprirlo da soli:

```
$ for i in /bin/* /sbin/* /usr/bin/* /usr/sbin/*; do [ -f $i ] && {
type=`file $i | grep -il perl`; [ -n "$type" ] && echo $i; }; done
```

Incluse le seguenti utility contenute in pacchetti con priorità *required* o *important*:

- `/usr/bin/chkdupexe` del pacchetto util-linux.
- `/usr/bin/replay` del pacchetto bsdutils.
- `/usr/sbin/cleanup-info` del pacchetto dpkg.
- `/usr/sbin/dpkg-divert` del pacchetto dpkg.
- `/usr/sbin/dpkg-statoverride` del pacchetto dpkg.
- `/usr/sbin/install-info` del pacchetto dpkg.
- `/usr/sbin/update-alternatives` del pacchetto dpkg.
- `/usr/sbin/update-rc.d` del pacchetto sysvinit.
- `/usr/bin/grog` del pacchetto groff-base.

[7] You can make (on another system) a dummy package with equivs.

- `/usr/sbin/adduser` del pacchetto adduser.
- `/usr/sbin/debconf-show` del pacchetto debconf.
- `/usr/sbin/deluser` del pacchetto adduser.
- `/usr/sbin/dpkg-preconfigure` del pacchetto debconf.
- `/usr/sbin/dpkg-reconfigure` del pacchetto debconf.
- `/usr/sbin/exigrep` del pacchetto exim.
- `/usr/sbin/eximconfig` del pacchetto exim.
- `/usr/sbin/eximstats` del pacchetto exim.
- `/usr/sbin/exim-upgrade-to-r3` del pacchetto exim.
- `/usr/sbin/exiqsumm` del pacchetto exim.
- `/usr/sbin/keytab-lilo` del pacchetto lilo.
- `/usr/sbin/liloconfig` del pacchetto lilo.
- `/usr/sbin/lilo_find_mbr` del pacchetto lilo.
- `/usr/sbin/syslogd-listfiles` del pacchetto sysklogd.
- `/usr/sbin/syslog-facility` del pacchetto sysklogd.
- `/usr/sbin/update-inetd` del pacchetto netbase.

Quindi, senza Perl e a meno che non riscriviate queste utility come script di shell, probabilmente non potrete gestire alcun pacchetto (quindi neanche aggiornare il sistema, il che *non è una buona cosa*).

Se siete determinati a rimuovere Perl dal sistema base di Debian e avete tempo libero, spedite rapporti sui bachi dei pacchetti sopracitati includendo (in forma di patch) codice che sostituisca le utility con script di shell.

Se desiderate verificare quali pacchetti Debian dipendano da Perl potete usare

```
$ grep-available -s Package,Priority -F Depends perl
```

o

```
$ apt-cache rdepends perl
```

Leggere la mailing list debian security

Non sarebbe sbagliato dare uno sguardo alla mailing-list debian-security-announce, dove vengono annunciate migliorie ai pacchetti rilasciate da parte del Team Debian per la Sicurezza, o in mailto:debian-security@lists.debian.org, dove potrete partecipare alle discussioni sulla sicurezza in Debian.

In order to receive important security update alerts, send an email to mailto:debian-security-announce-request@lists.debian.org with the word "subscribe" in the subject line. You can also subscribe to this moderated email list via the web page at http://www.debian.org/MailingLists/subscribe.

Questa mailing list ha un basso volume di traffico ed iscrivendovi verrete immediatamente avvisati sugli aggiornamenti di sicurezza che riguardano la distribuzione Debian. Questo permette di scaricare tempestivamente i nuovi pacchetti con le soluzioni ai bachi di sicurezza ed è veramente importante per mantenere sicuro il sistema. Leggete sezione chiamata «Eseguire un aggiornamento per la sicurezza» per trovare dettagli su questo argomento.

Capitolo 4. Dopo l'installazione

Dopo aver installato il sistema, bisogna renderlo sicuro; molti esempi descritti in questo capitolo puntano ad ottenere tale risultato. Questo dipende dal vostro setup, ma per la prevenzione dagli accessi fisici siete invitati a leggere i seguenti paragrafi: sezione chiamata «Modificare il BIOS (ancora)»,sezione chiamata «Impostazione della password in LILO o GRUB», sezione chiamata «Rimuovere il prompt root nel kernel», sezione chiamata «Circoscrivere l'accesso alla console» e sezione chiamata «Circoscrivere la possibilità di riavviare da console».

Dopo esservi connessi a qualsiasi rete, specialmente se pubblica, dovreste effettuare un aggiornamento di sicurezza (vedete sezione chiamata «Eseguire un aggiornamento per la sicurezza») e magari cogliere l'occasione per fare un backup del vostro sistema (in merito vedete sezione chiamata «Una fotografia del sistema»).

Iscrizione alla mailing list Debian Security Announce

Per ricevere informazioni sugli aggiornamenti di sicurezza disponibili dovreste iscrivervi alla mailing list debian-security-announce per ricevere i Debian Security Advisories (DSA). Leggete sezione chiamata «Il Team Debian per la Sicurezza» per maggiori informazioni su come lavora il Team Debian per la Sicurezza. Per informazioni su come iscriversi alle mailing list Debian leggete http://lists.debian.org.

I DSA sono firmati con la chiave del Team Debian per la Sicurezza che può essere ottenuta da http://security.debian.org.

Sarebbe anche opportuno prendere in considerazione la sottoscrizione della http://lists.debian.org/debian-security, sulla quale vengono discusse questioni generiche sulla sicurezza del sistema operativo Debian. Potrete così sentire il parere di altri colleghi amministratori di sistema, così come sviluppatori di Debian e di strumenti per la sicurezza, che potranno rispondere alle vostre domande ed offrire consulenza.

FIXME: aggiungere qui la chiave?

Eseguire un aggiornamento per la sicurezza

Non appena nuovi bug di sicurezza vengono scoperti nei pacchetti, i manutentori di Debian e gli autori dei programmi generalmente li correggono in pochi giorni o addirittura ore. Dopo che il bug è risolto, viene reso disponibile un nuovo pacchetto su http://security.debian.org.

Se state installando una qualsiasi versione di Debian dovete tenere a mente che potrebbero essere usciti nel frattempo degli aggiornamenti per la sicurezza per pacchetti che sono stati scoperti come affetti da vulnerabilità. Ci potrebbero essere comunque delle versioni secondarie (ce ne sono state quattro in Debian 3.0 *sarge*) che includono questi aggiornamenti dei pacchetti.

During installation security updates are configured for your system and pending updates downloaded and applied, unless you specifically opt out of this or the system was not connected to the Internet. The updates are applied even before the first boot, so the new system starts its life as up to date as possible.

To manually update the system, put the following line in your `sources.list` and you will get security updates automatically, whenever you update your system. Replace *`[CODENAME]`* with the release codename, e.g. *squeeze*.

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

Notate: se state utilizzando il ramo *testing*, utilizzate i sorgenti dei mirror di sicurezza di testing come descritto in sezione chiamata «Supporto alla sicurezza per il ramo testing».

Una volta che avete fatto questo, potrete utilizzare più strumenti per aggiornare il vostro sistema. Se state usando un sistema desktop avrete un'applicazione chiamata[1] **update-notifier** che renderà semplice il controllo della disponibilità di nuovi aggiornamenti; selezionandola potrete aggiornare un sistema dal desktop (utilizzando **update-manager**). Per maggiori informazioni vedete sezione chiamata «Controllo degli aggiornamenti dal Desktop». Negli ambienti desktop potete anche utilizzare synaptic (GNOME), kpackage o adept (KDE) per interfacce più avanzate. Se state lavorando su un semplice terminale, in console, per aggiornare potete utilizzare aptitude, apt o dselect (deprecato):

- se volete utilizzare l'interfaccia di testo di aptitude occorre solamente selezionare *u* (aggiorna) seguito da *g* (esegui aggiornamento). Altrimenti scrivete i seguenti comandi sulla riga di comando (come root):

```
# aptitude update
# aptitude upgrade
```

- se volete utilizzare apt scrivete solamente i comandi come con **aptitude**, ma sostituite le righe sopra di **aptitude** con **apt-get**.

- se volete utilizzare dselect allora prima [U]Aggiornare, poi [I]nstallare ed alla fine, [C]onfigurare i pacchetti installati/aggiornati.

If you like, you can add the deb-src lines to `/etc/apt/sources.list` as well. See apt(8) for further details.

Aggiornamento di sicurezza delle librerie

Once you have executed a security update you might need to restart some of the system services. If you do not do this, some services might still be vulnerable after a security upgrade. The reason for this is that daemons that are running before an upgrade might still be using the old libraries before the upgrade [2].

From Debian *Jessie* and up, you can install the needrestart package, which will run automatically after each APT upgrade and prompt you to restart services that are affected by the just-installed updates. In earlier releases, you can run the **checkrestart** program (available in the debian-goodies package) manually after your APT upgrade.

Some packages (like libc6) will do this check in the postinst phase for a limited set of services specially since an upgrade of essential libraries might break some applications (until restarted)[3].

Quindi portare il sistema al runlevel 1 (singolo utente) e poi di nuovo a runlevel 3 (multiutente) ed assicurare il riavvio della maggior parte dei servizi di sistema (se non di tutti). Ma questa non è un'opzione fattibile se state eseguendo l'aggiornamento di sistema da una connessione remota (come ssh) poiché questa verrà interrotta.

Excercise caution when dealing with security upgrades if you are doing them over a remote connection like ssh. A suggested procedure for a security upgrade that involves a service restart is to restart the SSH daemon and then, immediately, attempt a new ssh connection without breaking the previous one. If the connection fails, revert the upgrade and investigate the issue.

[1] In *etch* e nelle versioni successive.
[2] Even though the libraries have been removed from the filesystem the inodes will not be cleared up until no program has an open file descriptor pointing to them.
[3] This happened, for example, in the upgrade from libc6 2.2.x to 2.3.x due to NSS authentication issues, see http://lists.debian.org/debian-glibc/2003/03/msg00276.html.

Aggiornamenti di sicurezza per il kernel

Per prima cosa dovete essere sicuri che il kernel che volete aggiornare sia supportato dal vostro gestore di pacchetti. Se avete fatto l'installazione usando il sistema di installazione di Debian 3.0 o versioni precedenti, il kernel *non* è integrato nel sistema di gestione dei pacchetti e potrebbe non essere aggiornato. Potete trovare conferma di ciò lanciando:

```
$ dpkg -S `readlink -f /vmlinuz`
linux-image-2.6.18-4-686: /boot/vmlinuz-2.6.18-4-686
```

If your kernel is not being managed you will see a message saying that the package manager did not find the file associated to any package instead of the message above, which says that the file associated to the current running kernel is being provided by the linux-image-2.6.18-4-686. So first, you will need to manually install a kernel image package. The exact kernel image you need to install depends on your architecture and your prefered kernel version. Once this is done, you will be able to manage the security updates of the kernel just like those of any other package. In any case, notice that the kernel updates will *only* be done for kernel updates of the same kernel version you are using, that is, **apt** will not automatically upgrade your kernel from the 2.4 release to the 2.6 release (or from the 2.4.26 release to the 2.4.27 release[4]).

The installation system of recent Debian releases will handle the selected kernel as part of the package system. You can review which kernels you have installed by running:

```
$ COLUMNS=150 dpkg -l 'linux-image*' | awk '$1 ~ /ii/ { print $0 }'
```

Eseguite i seguenti comandi per controllare se il vostro kernel ha bisogno di essere aggiornato:

```
$ kernfile=`readlink -f /vmlinuz`
$ kernel=`dpkg -S $kernfile | awk -F : '{print $1}'`
$ apt-cache policy $kernel
linux-image-2.6.18-4-686:
  Installed: 2.6.18.dfsg.1-12
  Candidate: 2.6.18.dfsg.1-12
  Version table:
 *** 2.6.18.dfsg.1-12 0
        100 /var/lib/dpkg/status
```

Nel caso in cui stiate effettuando un aggiornamento di sicurezza che coinvolge anche il kernel è *necessario* riavviare il sistema affinché l'aggiornamento di sicurezza sia efficace. Finché il sistema non viene riavviato rimane in esecuzione la vecchia (e vulnerabile) immagine del kernel.

If you need to do a system reboot (because of a kernel upgrade) you should make sure that the kernel will boot up correctly and network connectivity will be restored, specially if the security upgrade is done over a remote connection like ssh. For the former you can configure your boot loader to reboot to the original kernel in the event of a failure (for more detailed information read Remotely rebooting Debian GNU/Linux machines [http://www.debian-administration.org/?article=70]). For the latter you have to introduce a network connectivity test script that will check if the kernel has started up the network subsystem properly and reboot the system if it did not[5]. This should prevent nasty surprises like updating the kernel and then

[4] Unless you have installed a kernel metapackage like linux-image-2.6-686 which will always pull in the latest kernel minor revision for a kernel release and a given architecture.

[5] A sample script called testnet [http://www.debian-administration.org/articles/70/testnet] is available in the Remotely rebooting Debian GNU/Linux machines [http://www.debian-administration.org/?article=70] article. A more elaborate network connectivity testing script is available in this Testing network connectivity article. [http://www.debian-administration.org/?article=128]

realizing, after a reboot, that it did not detect or configure the network hardware properly and you need to travel a long distance to bring the system up again. Of course, having the system serial console [6] in the system connected to a console or terminal server should also help debug reboot issues remotely.

Modificare il BIOS (ancora)

Ricordate il paragrafo sezione chiamata «Scegliere una password per il BIOS»? Bene, fatelo ora, se non avete bisogno che l'avvio della macchina avvenga da un supporto rimuovibile, modificate il BIOS per renderla avviabile *solamente* dall'hard disk. Prestate attenzione a non perdere la password del BIOS, altrimenti in caso di fallimento dell'avvio non potrete entrare nel BIOS e cambiare media, ad esempio per avviare da CD-ROM.

Un altro piccolo accorgimento potrebbe essere quello di cambiare la configurazione del BIOS, permettendo al sistema di partire dall'hard disk e se questo fallisse allora provare con un supporto removibile. A proposito, molte persone non usano la password per il BIOS ed è quindi facile dimenticarsene.

Impostazione della password in LILO o GRUB

Anybody can easily get a root-shell and change your passwords by entering

`<name-of-your-bootimage> init=/bin/sh`

at the boot prompt. After changing the passwords and rebooting the system, the person has unlimited root-access and can do anything he/she wants to the system. After this procedure you will not have root access to your system, as you do not know the root password.

Per essere sicuri di evitare di trovarvi in questa situazione dovete impostare una password per il vostro boot loader. Potete scegliere se impostare la password globalmente, oppure per una determinata immagine.

For LILO you need to edit the config file `/etc/lilo.conf` and add a **`password`** and **`restricted`** line as in the example below.

```
image=/boot/2.2.14-vmlinuz
   label=Linux
   read-only
   password=hackme
   restricted
```

Successivamente, assicuratevi che il file di configurazione non sia leggibile da tutti, in modo da impedire che i normali utenti possano leggere la password. Fatto ciò, riavviate lilo. Omettere la riga contenente `restricted` fa sì che lilo chieda sempre una password, indipendentemente dal fatto che gli siano stati passati dei parametri o meno. I permessi predefiniti per il file `/etc/lilo.conf` consentono l'accesso in scrittura ed in lettura all'utente root e l'accesso in lettura al gruppo utenti cui appartiene il file, root.

Se al posto di LILO usate GRUB, modificate `/boot/grub/menu.lst` ed aggiungete le seguenti due righe all'inizio (naturalmente, sostituite `hackme` con la vostra password). Questo previene che si modifichi la configurazione di avvio. La direttiva `timeout 3` istruisce **grub** ad attendere 3 secondi prima di avviare il sistema.

```
timeout 3
```

[6] Setting up a serial console is beyond the scope of this document, for more information read the Serial HOWTO [http://www.tldp.org/HOWTO/Serial-HOWTO.html] and the Remote Serial Console HOWTO [http://www.tldp.org/HOWTO/Remote-Serial-Console-HOWTO/index.html].

```
password hackme
```

Per conservare l'integrità delle password, si può generare una password cifrata. Il programma **grub-md5-crypt** genera una password compatibile con l'algoritmo di cifratura per le password di GRUB (MD5). Per specificare in **grub** che si intende usare il sistema di password nel formato MD5, bisogna usare la seguente direttiva:

```
timeout 3
password --md5 $1$bw0ez$tljnxxKLfMzmnDVaQWgjP0
```

Il parametro --md5, istruisce **grub** ad usare il processo di autenticazione MD5. L'esempio riporta la versione cifrata con MD5 della parola hackme. Usare il metodo di cifratura delle password MD5 è preferibile che lasciare il testo chiaramente leggibile. Potete trovare altre informazioni sull'uso delle password con **grub** nel pacchetto grub-doc.

Disabilitare il prompt di root su initramfs

Notate: questo si riferisce ai kernel predefiniti forniti in versioni successive a Debian 3.1

Linux 2.6 kernels provide a way to access a root shell while booting which will be presented during loading the initramfs on error. This is helpful to permit the administrator to enter a rescue shell with root permissions. This shell can be used to manually load modules when autodetection fails. This behavior is the default for **initramfs-tools** generated initramfs. The following message will appear:

```
"ALERT!  /dev/sda1 does not exist.  Dropping to a shell!
```

In order to remove this behavior you need to set the following boot argument:*panic=0*. Add this to the variable `GRUB_CMDLINE_LINUX` in `/etc/default/grub` and issue **update-grub** or to the append section of `/etc/lilo.conf`.

Rimuovere il prompt root nel kernel

Notate: questo non si applica ai kernel forniti per Debian 3.1 come anche il timeout per il ritardo del kernel è stato impostato a 0.

I kernel Linux della versione 2.4 forniscono la possibilità di accedere ad una shell da superutente durante il boot di sistema, subito dopo il caricamento del file system cramfs. Apparirà un messaggio che permetterà all'amministratore di accedere ad una shell con privilegi di superutente, questa può essere usata per caricare manualmente i moduli qualora il riconoscimento automatico fallisca. Questo è il comportamento predefinito per gli **initrd** e per `linuxrc`. Apparirà il seguente messaggio:

```
Press ENTER to obtain a shell (waits 5 seconds)
```

Per rimuovere questo comportamento dovrete modificare `/etc/mkinitrd/mkinitrd.conf` ed impostare:

```
# DELAY  The  number  of seconds the linuxrc script should wait to
# allow the user to interrupt it before the system is brought up
DELAY=0
```

Quindi dovrete rigenerare l'immagine del ramdisk. Lo potete fare con:

```
# cd /boot
# mkinitrd -o initrd.img-2.4.18-k7 /lib/modules/2.4.18-k7
```

Oppure (preferito):

```
# dpkg-reconfigure -plow kernel-image-2.4.x-yz
```

Circoscrivere l'accesso alla console

Some security policies might force administrators to log in to the system through the console with their user/password and then become superuser (with **su** or **sudo**). This policy is implemented in Debian by editing the `/etc/pam.d/login` and the `/etc/securetty` when using PAM:

`/etc/pam.d/login` In older Debian releases you would need to edit `login.defs`, and use the CONSOLE variable which defines a file or list of terminals on which root logins are allowed. enables the pam_securetty.so module. This module, when properly configured will not ask for a password when the root user tries to login on an insecure console, rejecting access as this user.

`securetty` The `/etc/securetty` is a configuration file that belongs to the login package. by adding/removing the terminals to which root access will be allowed. If you wish to allow only local console access then you need *console*, *ttyX* Or *ttyvX* in GNU/FreeBSD, and *ttyE0* in GNU/KNetBSD. and *vc/X* (if using *devfs* devices), you might want to add also *ttySX* Or *comX* in GNU/Hurd, *cuaaX* in GNU/FreeBSD, and *ttyXX* in GNU/KNetBSD. if you are using a serial console for local access (where X is an integer, you might want to have multiple instances. The default configuration for *Wheezy* The default configuration in *woody* includes 12 local tty and vc consoles, as well as the *console* device but does not allow remote logins. In *sarge* the default configuration provides 64 consoles for tty and vc consoles. includes many tty devices, serial ports, vc consoles as well as the X server and the *console* device. You can safely adjust this if you are not using that many consoles. You can confirm the virtual consoles and the tty devices you have by reviewing `/etc/inittab` Look for the *getty* calls. . For more information on terminal devices read the Text-Terminal-HOWTO [http://tldp.org/HOWTO/Text-Terminal-HOWTO-6.html]

Quando viene utilizzato PAM, altre modifiche al processo di autenticazione, comprese restrizioni a livello utente e gruppo durante orari prestabiliti, possono essere configurate in `/etc/pam.d/login`. Un' interessante caratteristica è quella di poter disabilitare l'autenticazione con password nulle. Questa caratteristica può essere abilitata rimuovendo *nullok* dalla riga:

```
auth       required   pam_unix.so nullok
```

Circoscrivere la possibilità di riavviare da console

If your system has a keyboard attached to it anyone (yes *anyone*) with physical access to the system can reboot the system through it without login in just pressing the *Ctrl+Alt+Delete* keyboard combination, also known as the *three finger salute*. This might, or might not, adhere to your security policy.

This is aggravated in environments in which the operating system is running virtualised. In these environments, the possibility extends to users that have access to the virtual console (which might be accessed over the network). Also note that, in these environments, this keyboard combination is used constantly (to open a login shell in some GUI operating systems) and an administrator might *virtually* send it and force a system reboot.

There are two ways to restrict this:

- configure it so that only *allowed* users can reboot the system,
- disable this feature completely.

If you want to restrict this, you must check the `/etc/inittab` so that the line that includes **`ctrlalt-del`** calls s**hutdown** with the **`-a`** switch.

The default in Debian includes this switch:

```
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

The **`-a`** switch, as the shutdown(8) manpage describes,makes it possible to allow *some* users to shutdown the system. For this the file `/etc/shutdown.allow` must be created and the administrator has to include there the name of users which can boot the system. When the *three finger salute* combination is pressed in a console the program will check if any of the users listed in the file are logged in. If none of them is, **shutdown** will *not* reboot the system.

If you want to disable the Ctrl+Alt+Del combination you just need to comment the line with the *ctrlaltdel* definition in the `/etc/inittab`.

Remember to run **`init q`** after making any changes to the `/etc/inittab` file for the changes to take effect.

Restricting the use of the Magic SysRq key

The *Magic SysRq key* is a key combination that allows users connected to the system console of a Linux kernel to perform some low-level commands. These low-level commands are sent by pressing simultaneously *Alt+SysRq* and a command key. The SysRq key in many keyboards is labeled as the *Print Screen* key.

Since the Etch release, the Magic SysRq key feature is enabled in the Linux kernel to allow console users certain privileges. You can confirm this by checking if the `/proc/sys/kernel/sysrq` exists and reviewing its value:

```
$ cat /proc/sys/kernel/sysrq
438
```

The default value shown above allows all of the SysRq functions except for the possibility of sending signals to processes. For example, it allow users connected to the console to remount all systems read-only, reboot the system or cause a kernel panic. In all the features are enabled, or in older kernels (earlier than 2.6.12) the value will be just 1.

You should disable this functionality ifaccess to the console is not restricted to authorised users: the console is connected to a modem line, there is easy physical access to the system or it is running in a virtualised environment and other users access the console. To do this edit the `/etc/sysctl.conf` and add the following lines:

```
# Disables the magic SysRq key
kernel.sysrq = 0
```

For more information, read security chapter in the Remote Serial Console HOWTO [http://tldp.org/HOWTO/Remote-Serial-Console-HOWTO/security-sysrq.html], Kernel SysRQ documentation [https://

www.kernel.org/doc/Documentation/admin-guide/sysrq.rst]. and the Magic_SysRq_key wikipedia entry [http://en.wikipedia.org/wiki/Magic_SysRq_key].

Montare le partizioni nel modo giusto

When mounting an `Ext` file system (`ext2`, `ext3` or `ext4`), there are several additional options you can apply to the mount call or to `/etc/fstab`. For instance, this is my fstab entry for the `/tmp` partition:

```
/dev/hda7    /tmp    ext2    defaults,nosuid,noexec,nodev    0    2
```

Potete vedere le differenze nella sezione delle opzioni. L'opzione `nosuid` ignora completamente i bit setuid e setgid, mentre `noexec` impedisce l'esecuzione di qualsiasi programma su quel punto di montaggio e `nodev` ignora i file dispositivo. Sembra grandioso ma:

- only applies to `ext2` or `ext3` file systems
- può essere facilmente aggirato

L'opzione `noexec` evita l'esecuzione diretta dei file binari, ma veniva aggirata facilmente nelle versioni precedenti del kernel:

```
alex@joker:/tmp# mount | grep tmp
/dev/hda7 on /tmp type ext2 (rw,noexec,nosuid,nodev)
alex@joker:/tmp# ./date
bash: ./date: Permission denied
alex@joker:/tmp# /lib/ld-linux.so.2 ./date
Sun Dec  3 17:49:23 CET 2000
```

Comunque i kernel più recenti gestiscono correttamente l'opzione `noexec`:

```
angrist:/tmp# mount | grep /tmp
/dev/hda3 on /tmp type ext3 (rw,noexec,nosuid,nodev)
angrist:/tmp# ./date
bash: ./tmp: Permission denied
angrist:/tmp# /lib/ld-linux.so.2 ./date
./date: error while loading shared libraries: ./date: failed to map segment
from shared object: Operation not permitted
```

However, many script kiddies have exploits which try to create and execute files in `/tmp`. If they do not have a clue, they will fall into this pit. In other words, a user cannot be tricked into executing a trojanized binary in `/tmp` e.g. when `/tmp` is accidentally added into the local PATH.

Also be forewarned, some script might depend on `/tmp` being executable. Most notably, Debconf has (had?) some issues regarding this, for more information see http://bugs.debian.org/116448.

Il seguente è un altro esempio. Una nota: `/var` può essere impostata noexec, ma certo software [7] mette i propri eseguibili in `/var`. Lo stesso si applica all'opzione nosuid.

```
/dev/sda6   /usr          ext3    defaults,ro,nodev       0       2
/dev/sda12  /usr/share    ext3    defaults,ro,nodev,nosuid        0       2
```

[7] Tra questi i programmi Smartlist ed il gestore dei pacchetti dpkg, visto che gli script di installazione (post, pre) e di rimozione (post, pre) sono in `/var/lib/dpkg/`.

```
/dev/sda7   /var          ext3    defaults,nodev,usrquota,grpquota 0      2
/dev/sda8   /tmp          ext3    defaults,nodev,nosuid,noexec,usrquota,grpquota
/dev/sda9   /var/tmp      ext3    defaults,nodev,nosuid,noexec,usrquota,grpquota
/dev/sda10  /var/log      ext3    defaults,nodev,nosuid,noexec    0       2
/dev/sda11  /var/account  ext3    defaults,nodev,nosuid,noexec    0       2
/dev/sda13  /home         ext3    rw,nosuid,nodev,exec,auto,nouser,async,usrquota,
/dev/fd0    /mnt/fd0      ext3    defaults,users,nodev,nosuid,noexec      0
/dev/fd0    /mnt/floppy   vfat    defaults,users,nodev,nosuid,noexec      0
/dev/hda    /mnt/cdrom    iso9660 ro,users,nodev,nosuid,noexec            0
```

Impostare `/tmp` come noexec

Be careful if setting `/tmp` noexec when you want to install new software, since some programs might use it for installation. apt is one such program (see http://bugs.debian.org/116448) if not configured properly `APT::ExtractTemplates::TempDir` (see apt-extracttemplates(1)). You can set this variable in `/etc/apt/apt.conf` to another directory with exec privileges other than `/tmp`.

Impostare /usr in sola lettura

Se impostate `/usr` in sola lettura non potrete più installare nuovi pacchetti sul vostro sistema Debian GNU/Linux. Dovrete prima rimontarla in lettura-scrittura, installare i pacchetti e poi rimontarla in sola lettura. apt può essere configurato per eseguire comandi prima e dopo l'installazione dei pacchetti, per cui potreste volerlo configurare correttamente.

Per farlo, modificate `/etc/apt/apt.conf` aggiungendo:

```
DPkg
{
    Pre-Invoke  { "mount /usr -o remount,rw" };
    Post-Invoke { "mount /usr -o remount,ro" };
};
```

Notate che Post-Invoke può fallire con un messaggio d'errore "/usr busy". Questo succede frequentemente quando si stanno aggiornando alcuni file mentre eseguiamo l'aggiornamento. Potete trovare questi programmi eseguendo:

```
# lsof +L1
```

Stop or restart these programs and run the Post-Invoke manually. *Beware!* This means you'll likely need to restart your X session (if you're running one) every time you do a major upgrade of your system. You might want to reconsider whether a read-only `/usr` is suitable for your system. See also this discussion on debian-devel about read-only [http://lists.debian.org/debian-devel/2001/11/threads.html#00212].

Fornire un accesso sicuro per gli utenti

Autenticazione degli utenti: PAM

PAM (Pluggable Authentication Modules) permette agli amministratori di sistema di scegliere come le applicazioni autenticano gli utenti. Notate che PAM non funziona se un'applicazione non è stata compilata con il supporto per PAM. Molte delle applicazioni fornite con Debian hanno questo supporto integrato (inoltre Debian non ha il supporto PAM per versioni precedenti alla 2.2). L'attuale configurazione predefinita per un qualsiasi servizio abilitato PAM è emulare l'autenticazione UNIX (leggete in `/usr/share/`

`doc/libpam0g/Debian-PAM-MiniPolicy.gz` per ulteriori informazioni su come i servizi PAM *dovrebbero* funzionare in Debian).

Ogni applicazione con supporto PAM ha un file di configurazione in `/etc/pam.d/` che può essere usato per modificare il suo comportamento:

- quale programma sottostante viene utilizzato per l'autenticazione.
- quale programma sottostante viene utilizzato per le sessioni.
- come si comportano i controlli delle password.

The following description is far from complete, for more information you might want to read the Linux-PAM Guides [https://packages.debian.org/sid/libpam-doc] as a reference. This documentation is available in the system if you install the libpam-doc at `/usr/share/doc/libpam-doc/html/`.

PAM offers you the possibility to go through several authentication steps at once, without the user's knowledge. You could authenticate against a Berkeley database and against the normal `passwd` file, and the user only logs in if the authentication succeeds in both. You can restrict a lot with PAM, just as you can open your system doors very wide. So be careful. A typical configuration line has a control field as its second element. Generally it should be set to `requisite`, which returns a login failure if one module fails.

Password security in PAM

Review the `/etc/pam.d/common-password`, included by `/etc/pam.d/passwd`[8] This file is included by other files in `/etc/pam.d/` to define the behaviour of password use in subsystems that grant access to services in the machine, like the console login (login), graphical login managers (such as gdm or lightdm), and remote login (such as sshd). This definition is

You have to make sure that the pam_unix.so module uses the "sha512" option to use encrypted passwords. This is the default in Debian Squeeze.

The line with the definition of the pam_unix module will look something like:

```
password   [success=1 default=ignore]      pam_unix.so nullok obscure minlen=8 s
```

This definition:

- Enforces password encryption when storing passwords, using the SHA-512 hash function (option *sha512*),
- Enables password complexity checks (option *obscure*) as defined in the pam_unix(8) manpage,
- Imposes a minimum password length (option *min*) of 8.

You have to ensure that encrypted passwords are used in PAM applications, since this helps protect against dictionary cracks. Using encryption also makes it possible to use passwords longer than 8 characters.

Since this module is also used to define how passwords are changed (it is included by **chpasswd**) you can strengthen the password security in the system by installing libpam-cracklib and introducing this definition in the `/etc/pam.d/common-password` configuration file:

```
# Be sure to install libpam-cracklib first or you will not be able to log in
password   required     pam_cracklib.so retry=3 minlen=12 difok=3
```

[8] In old Debian releases the configuration of the modules was defined directly in `/etc/pam.d/passwd`.

```
password   [success=1 default=ignore]      pam_unix.so obscure minlen=8 sha512 u
```

So, what does this incantation do? The first line loads the cracklib PAM module, which provides password strength-checking, prompts for a new password with a minimum size [9] of 12 characters, and difference of at least 3 characters from the old password, and allows 3 retries. Cracklib depends on a wordlist package (such as wenglish, wspanish, wbritish, ...), so make sure you install one that is appropriate for your language or cracklib might not be useful to you at all.

The second line (using the pam_unix.so module) is the default configuration in Debian, as described above, save for the *use_authok* option. The *use_authok* option is required if pam_unix.so is stacked after pam_cracklib.so, and is used to hand over the password from the previous module. Otherwise, the user would be prompted for the password twice.

For more information about setting up Cracklib, read the pam_cracklib(8) manpage and the article Linux Password Security with pam_cracklib [http://www.deer-run.com/~hal/sysadmin/pam_cracklib.html] by Hal Pomeranz.

By enabling the cracklib PAM module you setup a policy that forces uses to use strong passwords.

Alternatively, you can setup and configure PAM modules to use double factor authentication such as: libpam-barada, libpam-google-authenticator, libpam-oath, libpam-otpw, libpam-poldi, libpam-usb or libpam-yubico. The configuration of these modules would make it possible to access the system using external authentication mechanisms such as smartcards, external USB keys, or One-Time-Passwords generated by external applications running, for example, in the user's mobile phone.

Please note that these restrictions apply to all users but *not* to the password changes done by the root user. The root user will be able to set up any password (any length or complexity) for personal use or others regardless of the restrictions defined here.

User access control in PAM

Per essere sicuri che l'utente root possa solo autenticarsi nel sistema da terminali locali, si dovrebbe aggiungere in `/etc/pam.d/login` la seguente riga:

```
auth     requisite  pam_securetty.so
```

Then you should modify the list of terminals on which direct root login is allowed in `/etc/securetty` (as described in sezione chiamata «Circoscrivere l'accesso alla console»). Alternatively, you could enable the pam_access module and modify `/etc/security/access.conf` which allows for a more general and fine-tuned access control, but (unfortunately) lacks decent log messages (logging within PAM is not standardized and is particularly unrewarding problem to deal with). We'll return to `access.conf` a little later.

User limits in PAM

The following line should be enabled in `/etc/pam.d/login` to set up user resource limits.

```
session  required   pam_limits.so
```

Questa riga riduce le risorse di sistema che gli utenti possono usare (vedete più avanti in sezione chiamata «Limitare l'uso delle risorse: il file `limits.conf`»). Per esempio, potreste ridurre il numero di login

[9] The minlen option is not entirely straightforward and is not exactly the number of characters in the password. A tradeoff can be defined between complexity and length by adjusting the "credit" parameters of different character classes. For more information read the pam_cracklib(8) manpage.

concorrenti (di un certo gruppo di utenti, o globalmente) ammessi, il numero di processi, la dimensione della memoria etc. etc.

Control of su in PAM

If you want to protect **su**, so that only some people can use it to become root on your system, you need to add a new group "wheel" to your system (that is the cleanest way, since no file has such a group permission yet). Add root and the other users that should be able to **su** to the root user to this group. Then add the following line to `/etc/pam.d/su`:

```
auth        requisite   pam_wheel.so group=wheel debug
```

Questo garantisce che solo le persone del gruppo "wheel" possano usare **su** per diventare root. Gli altri utenti non saranno in grado di diventare root. Infatti otterranno un messaggio di errore se cercheranno di diventarlo.

If you want only certain users to authenticate at a PAM service, this is quite easy to achieve by using files where the users who are allowed to login (or not) are stored. Imagine you only want to allow users 'ref' to log in via **ssh**. So you put them into `/etc/sshusers-allowed` and write the following into `/etc/pam.d/ssh`:

```
auth        required    pam_listfile.so item=user sense=allow file=/etc/sshusers
```

Temporary directories in PAM

Since there have been a number of so called insecure tempfile vulnerabilities, thttpd is one example (see DSA-883-1 [http://www.debian.org/security/2005/dsa-883]), the libpam-tmpdir is a good package to install. All you have to do is add the following to `/etc/pam.d/common-session`:

```
session    optional     pam_tmpdir.so
```

There has also been a discussion about adding this by default in Debian configuration, but it s. See http://lists.debian.org/debian-devel/2005/11/msg00297.html for more information.

Configuration for undefined PAM applications

Finally, but not least, create `/etc/pam.d/other` and enter the following lines:

```
auth     required       pam_securetty.so
auth     required       pam_unix_auth.so
auth     required       pam_warn.so
auth     required       pam_deny.so
account  required       pam_unix_acct.so
account  required       pam_warn.so
account  required       pam_deny.so
password required       pam_unix_passwd.so
password required       pam_warn.so
password required       pam_deny.so
session  required       pam_unix_session.so
session  required       pam_warn.so
session  required       pam_deny.so
```

Queste righe forniranno una buona configurazione di base per tutte le applicazioni che supportano PAM (l'accesso viene negato in modo predefinito).

Limitare l'uso delle risorse: il file `limits.conf`

Questo file è molto importante in quanto permette di definire dei limiti nell'utilizzo delle risorse per gli utenti del sistema. Nelle vecchie versioni di Debian questo file si trovava in `/etc/limits.conf`, ma nelle nuove versioni (che utilizzano PAM) questo file è stato spostato in `/etc/security/limits.conf`.

Se non si pongono dei limiti all'utilizzo delle risorse del computer, *qualsiasi* utente che abbia accesso ad una shell sul sistema (o anche un intruso che abbia compromesso il sistema tramite un servizio o un demone) può usare tutta la CPU, la RAM, lo stack e le altre risorse a disposizione del sistema. Questo problema di *esaurimento delle risorse* può essere corretto usando PAM.

There is a way to add resource limits to some shells (for example, **bash** has **ulimit**, see bash(1)), but since not all of them provide the same limits and since the user can change shells (see chsh(1)) it is better to place the limits on the PAM modules as they will apply regardless of the shell used and will also apply to PAM modules that are not shell-oriented.

I limiti sull'utilizzo delle risorse vengono fatti rispettare dal kernel e si configurano tramite il file `limits.conf`. Inoltre, le configurazioni PAM dei vari servizi hanno la necessità di poter accedere alle corrette impostazioni PAM. Potete controllare quali servizi siano sotto controllo ed imporre dei limiti eseguendo il seguente comando:

```
$ find /etc/pam.d/ \! -name "*.dpkg*" | xargs -- grep limits |grep -v ":#"
```

Solitamente, `login`, `ssh` ed i gestori di sessioni grafiche (`gdm`, `kdm` o `xdm`) dovrebbero occuparsi di far rispettare i limiti sull'utilizzo delle risorse, ma questo compito potrebbe anche essere delegato ad altri file di configurazione di PAM, magari inseriti in `cron`, per prevenire che i demoni di sistema si approprino di tutte le risorse disponibili.

I particolari limiti imposti all'uso di risorse per il proprio sistema dipendono dalle risorse a disposizione del sistema stesso, questo è uno dei motivi per cui non vengono creati dei limiti nell'installazione predefinita di Debian.

Per esempio, nelle righe seguenti c'é un estratto di un file di configurazione che imposta, per ogni utente, un limite massimo di 100 processi in esecuzione contemporanea (per evitare che l'utilizzo indiscriminato di *fork()* blocchi il sistema), un limite massimo di occupazione di memoria per ogni processo di 10MB ed un limite massimo di 10 login contemporanei. Gli utenti nel gruppo `adm` hanno dei limiti più alti e se vogliono possono produrre dei file core (c'é solo un limite *soft* sui file core).

```
*               soft    core            0
*               hard    core            0
*               hard    rss             1000
*               hard    memlock         1000
*               hard    nproc           100
*               -       maxlogins       1
*               hard    data            102400
*               hard    fsize           2048
@adm            hard    core            100000
@adm            hard    rss             100000
@adm            soft    nproc           2000
@adm            hard    nproc           3000
```

```
@adm            hard    fsize           100000
@adm            -       maxlogins       10
```

I limiti che un utente normale (inclusi i demoni di sistema) avrebbe sono elencabili dal comando:

```
$ ulimit -a
core file size        (blocks, -c) 0
data seg size         (kbytes, -d) 102400
file size             (blocks, -f) 2048
max locked memory     (kbytes, -l) 10000
max memory size       (kbytes, -m) 10000
open files                    (-n) 1024
pipe size          (512 bytes, -p) 8
stack size            (kbytes, -s) 8192
cpu time             (seconds, -t) unlimited
max user processes            (-u) 100
virtual memory        (kbytes, -v) unlimited
```

E questi sono i limiti per un utente con poteri amministrativi:

```
$ ulimit -a
core file size        (blocks, -c) 0
data seg size         (kbytes, -d) 102400
file size             (blocks, -f) 100000
max locked memory     (kbytes, -l) 100000
max memory size       (kbytes, -m) 100000
open files                    (-n) 1024
pipe size          (512 bytes, -p) 8
stack size            (kbytes, -s) 8192
cpu time             (seconds, -t) unlimited
max user processes            (-u) 2000
virtual memory        (kbytes, -v) unlimited
```

Per ulteriori informazioni leggete:

- PAM reference guide for available modules [https://web.archive.org/web/20030601112932/http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam-6.html]
- PAM configuration article [https://web.archive.org/web/20030217012148/http://www.samag.com/documents/s=1161/sam0009a/0009a.htm].
- Seifried's Securing Linux Step by Step [http://seifried.org/security/os/linux/20020324-securing-linux-step-by-step.html] on the *Limiting users overview* section.
- http://seifried.org/lasg/users/ nella sezione *Limitare e monitorare gli utenti*.

User login actions: edit `/etc/login.defs`

The next step is to edit the basic configuration and action upon user login. Note that this file is not part of the PAM configuration, it's a configuration file honored by login and **su** programs, so it doesn't make sense tuning it for cases where neither of the two programs are at least indirectly called (the **getty** program which sits on the consoles and offers the initial login prompt *does* invoke **login**).

```
FAILLOG_ENAB        yes
```

Se si abilita questa variabile, la fallita autenticazione verrà riportata nei log. Questo è fondamentale per tenere traccia di chiunque provi degli attacchi a forza bruta.

```
LOG_UNKFAIL_ENAB    no
```

Nel caso venga impostata tale variabile a 'yes' il programma memorizzerà i nomi utenti non presenti sul sistema, nel caso in cui la procedura di login fallisca per tale ragione. È comunque consigliabile impostare tale opzione a 'no' (ovverosia lasciare il valore predefinito) in quanto è possibile che un utente, inavvertitamente, scriva la propria password al posto del suo nome utente al prompt di login. Se la variabile viene impostata a 'yes', la password dell'utente verrà registrata. Nel caso in cui la variabile sia impostata al valore 'yes', sarà importante assegnare appropriati permessi ai file di log che registrano quel tipo di informazioni (per esempio, potreste impostarli a 640 ed assegnarli ad un gruppo appropriato quale ad esempio quello degli amministratori).

```
SYSLOG_SU_ENAB      yes
```

Questo abiliterà il logging dei tentativi di esecuzione del comando **su** in `syslog`. Questione molto importante su una macchina seria ma attenzione che può creare dei problemi con la privacy.

```
SYSLOG_SG_ENAB      yes
```

The same as `SYSLOG_SU_ENAB` but applies to the **sg** program.

```
ENCRYPT_METHOD  SHA512
```

As stated above, encrypted passwords greatly reduce the problem of dictionary attacks, since you can use longer passwords. This definition has to be consistent with the value defined in `/etc/pam.d/common-password`.

User login actions: edit `/etc/pam.d/login`

You can adjust the login configuration file to implement an stricter policy. For example, you can change the default configuration and increase the delay time between login prompts. The default configuration sets a 3 seconds delay:

```
auth       optional   pam_faildelay.so  delay=3000000
```

Increasing the *delay* value to a higher value to make it harder to use the terminal to log in using brute force. If a wrong password is typed in, the possible attacker (or normal user!) has to wait longer seconds to get a new login prompt, which is quite time consuming when you test passwords. For example, if you set *delay=10000000*, users will have to wait 10 seconds if they type a wrong password.

In this file you can also set the system to present a message to users before a user logs in. The default is disabled, as shown below:

```
# auth       required   pam_issue.so issue=/etc/issue
```

If required by your security policy, this file can be used to show a standard message indicating that access to the system is restricted and user acess is logged. This kind of disclaimer might be required in some

environments and jurisdictions. To enable it, just include the relevant information in the `/etc/issue`[10] file and uncomment the line enabling the pam_issue.so module in `/etc/pam.d/login`. In this file you can also enable additional features which might be relevant to apply local security policies such as:

- setting rules for which users can access at which times, by enabling the *pam_time.so* module and configuring `/etc/security/time.conf` accordingly (disabled by default),
- setup login sessions to use user limits as defined in `/etc/security/limits.conf` (enabled by default),
- present the user with the information of previous login information (enabled by default),
- print a message (`/etc/motd` and `/run/motd.dynamic`) to users after login in (enabled by default),

Restrizioni ftp: modificare il file `/etc/ftpusers`

Il file `/etc/ftpusers` contiene l'elenco degli utenti che non sono autorizzati ad autenticarsi all'host usando il servizio ftp. Solo usando questo metodo potete autorizzare gli utenti ad accedere all'ftp (solitamente questo è sconsigliato poiché usa le password in chiaro). Se i demoni attivi supportano PAM, si può anche usare questo metodo per autorizzare o negare agli utenti l'accesso ai servizi.

FIXME (BUG): questo è un bug di Debian, la configurazione predefinita *non* include negli `ftpusers` tutti gli utenti amministratori (in base-passwd).

Un sistema utile per aggiungere al file `/etc/ftpusers` tutti gli account di sistema è eseguire

```
$ awk -F : '{if ($3<1000) print $1}' /etc/passwd > /etc/ftpusers
```

Utilizzo di su

Se veramente gli utenti hanno la necessità di diventare super user sul vostro sistema, ad esempio per installare dei pacchetti o aggiungere utenti, potete utilizzare il comando **su** per cambiare la vostra identità. Dovreste tentare di evitare ogni accesso come utente root ed utilizzare invece **su**. Al momento, la soluzione migliore è rimuovere **su** e passare a **sudo**, che consente una scelta più ampia di soluzioni rispetto a **su**. Ad ogni modo, **su** è più comune visto che viene utilizzato su numerosi altri Unix.

Utilizzo di sudo

sudo allows the user to execute defined commands under another user's identity, even as root. If the user is added to `/etc/sudoers` and authenticates correctly, the commands defined in `/etc/sudoers` get enabled. Violations, such as incorrect passwords or trying to run a program you don't have permission for, are logged and mailed to root.

Non permettere accessi per amministrazione remota

Per impedire il login da remoto di account amministrativi dovete modificare `/etc/security/access.conf`. In questo modo gli utenti dovranno usare **su** (o **sudo**) e così rimarrà sempre traccia che un utente locale vuole usare i privilegi di amministratore.

Dovete aggiungere la seguente riga a `/etc/security/access.conf`, il file di configurazione predefinito di Debian ha una riga simile commentata:

[10] The default content of this file provides information about the operating system and version run by the system, which you might not want to provide to anonymous users.

```
-:wheel:ALL EXCEPT LOCAL
```

Remember to enable the pam_access module for every service (or default configuration) in `/etc/pam.d/` if you want your changes to `/etc/security/access.conf` honored.

Restrizioni agli utenti per l'accesso

A volte potreste pensare di avere la necessità di creare utenti nel vostro sistema locale per fornire un determinato servizio (pop3 mail o ftp). Prima di fare ciò, ricordatevi che l'implementazione di PAM in Debian GNU/Linux permette di validare utenti con un'ampia varietà di servizi di directory esterni (radius, ldap, ecc.) forniti dai pacchetti libpam.

If users need to be created and the system can be accessed remotely take into account that users will be able to log in to the system. You can fix this by giving users a null (`/dev/null`) shell (it would need to be listed in `/etc/shells`). If you want to allow users to access the system but limit their movements, you can use the `/bin/rbash`, equivalent to adding the `-r` option in **bash** (*RESTRICTED SHELL* see bash(1)). Please note that even with restricted shell, a user that access an interactive program (that might allow execution of a subshell) could be able to bypass the limits of the shell.

Debian currently provides in the unstable release (and might be included in the next stable releases) the `pam_chroot` module (in the libpam-chroot). An alternative to it is to **chroot** the service that provides remote logging (**ssh**, **telnet**). [11]

Se desiderate limitare il *quando* gli utenti possono accedere al sistema dovrete configurare `/etc/security/access.conf` per i vostri bisogni.

Per informazioni su come ingabbiare con **chroot** gli utenti che accedono al sistema mediante il servizio **ssh** vedete in sezione chiamata «Chroot environment for SSH».

Esame delle attività degli utenti

Nel caso in cui siate molto sospettosi, potreste configurare l'intero sistema in modo che esami continuamente tutte le attività che gli utenti svolgono nel sistema. Questa sezione presenta alcuni consigli e introduce alcuni strumenti utili.

Controllo di input e output con script

Potete utilizzare il comando **script** per controllare sia ciò che gli utenti stanno eseguendo, sia i risultati di questi comandi. Non potete impostare **script** come una shell (anche se lo aggiungete a `/etc/shells`), ma potete far sì che il file di inizializzazione della shell esegua i seguenti comandi:

```
umask 077
exec script -q -a "/var/log/sessions/$USER"
```

Di certo, se fate questo globalmente, per tutto il sistema, impedirete che la shell legga i file di inizializzazione personali (dato che la shell viene sovrascritta da **script**). Un'alternativa è fare questo nei file di inizializzazione dell'utente (ma poi l'utente lo potrà rimuovere, vedete i commenti sotto su questo).

Dovrete anche impostare i file nella directory da controllare (nell'esempio `/var/log/sessions/`) così che gli utenti possano scrivere in essa ma non possano rimuovere il file. Questo può essere fatto, per esempio, creando i file di sessione dell'utente in anticipo ed impostandoli con l'opzione *append-only* utilizzando **chattr**.

[11] libpam-chroot has not been yet thoroughly tested, it does work for **login** but it might not be easy to set up the environment for other programs

Un'utile alternativa per gli amministratori di sistema, che include l'informazione della data, sarebbe:

```
umask 077
exec script -q -a "/var/log/sessions/$USER-`date +%Y%m%d`"
```

Uso del file storico dei comandi della shell

Se volete rivedere i comandi che l'utente ha usato sulla shell (ma non qual'è stato il risultato di quei comandi), potete impostare un `/etc/profile` a livello di sistema, che configuri l'ambiente in modo tale che tutti i comandi vengano registrati in un file storico dei comandi. La configurazione a livello di sistema deve essere fatta in modo tale che gli utenti non possano rimuovere le capacità di verifica e registrazione della propria shell. Questo è qualcosa di specifico della shell, perciò assicuratevi che tutti gli utenti stiano usando una shell che supporti questa funzionalità.

For example, for bash, the `/etc/profile` could be set as follows [12] :

```
HISTFILE=~/.bash_history
HISTSIZE=10000
HISTFILESIZE=999999
# Don't let the users enter commands that are ignored
# in the history file
HISTIGNORE=""
HISTCONTROL=""
readonly HISTFILE
readonly HISTSIZE
readonly HISTFILESIZE
readonly HISTIGNORE
readonly HISTCONTROL
export HISTFILE HISTSIZE HISTFILESIZE HISTIGNORE HISTCONTROL
```

For this to work, the user can only append information to `.bash_history` file. You need *also* to set the *append-only* option using **chattr** program for `.bash_history` for all users. [13].

Note that you could introduce the configuration above in the user's `.profile`. But then you would need to setup permissions properly in such a way that prevents the user from modifying this file. This includes: having the user's home directories *not* belong to the user (since the user would be able to remove the file otherwise) but at the same time allow the user to read the `.profile` configuration file and write on the `.bash_history`. It would be good to set the *immutable* flag (also using **chattr**) for `.profile` too if you do it this way.

Completate il controllo dell'utente con le utility per gli account

L'esempio precedente è un modo semplice per configurare il controllo dell'utente ma potrebbe non essere utile per sistemi complessi o per quelli nei quali l'utente non usa per niente o quasi le shell. Se questo è il vostro caso, potete dare un'occhiata alla utility per account acct. Questo programma registrerà tutti i comandi dati dagli utenti o dai processi nel sistema, a scapito dello spazio disco.

All'attivazione dell'accounting tutte le informazioni su processi ed utenti verranno mantenute sotto `/var/account/`, o più specificamente nel file `pacct`. Il pacchetto per l'account include alcuni strumenti, come **sa**, **ac** e **lastcomm**, che permettono di analizzare questi dati.

[12] Setting HISTSIZE to a very large number can cause issues under some shells since the history is kept in memory for every user session. You might be safer if you set this to a high-enough value and backup user's history files (if you need all of the user's history for some reason)
[13] Without the append-only flag users would be able to empty the contents of the history file running `> .bash_history`

Altri metodi di controllo dell'utente

If you are completely paranoid and want to audit every user's command, you could take **bash** source code, edit it and have it send all that the user typed into another file. Or have ttysnoop constantly monitor any new ttys [14] and dump the output into a file. Other useful program is snoopy (see also github: https://github.com/a2o/snoopy) which is a user-transparent program that hooks in as a library providing a wrapper around `execve()` calls, any command executed is logged to **syslogd** using the `authpriv` facility (usually stored at `/var/log/auth.log`).

Uno sguardo ai profili utente

Se volete *vedere* cosa fanno solitamente gli utenti quando si connettono, potete usare il database `wtmp` che include tutte le informazioni di login. Il file può essere processato con diverse utility, tra cui **sac** che può restituire un profilo per ogni utente che mostra in quale arco di tempo si è connesso.

Nel caso in cui abbiate attivato l'accounting, potete anche utilizzare gli strumenti da esso forniti per determinare quando gli utenti accedono al sistema e che cosa eseguono.

Impostare delle umask per gli utenti

In base al vostro codice di condotta riguardo agli utenti, potreste voler cambiare il modo in cui gli utenti si scambiano informazioni, ossia quali siano i permessi predefiniti dei nuovi file creati dagli utenti.

L'impostazione predefinita di Debian per l'`umask` è *022*, questo significa che i file (e le directory) possono essere lette e visitate dai membri del gruppo dell'utente e da qualsiasi altro utente del sistema. Questa definizione si trova nel classico file di configurazione `/etc/profile`, che viene utilizzato da tutte le shell.

If Debian's default value is too permissive for your system you will have to change the umask setting for all the shells. More restrictive umask settings include 027 (no access is allowed to new files for the *other* group, i.e. to other users in the system) or 077 (no access is allowed to new files to the members the user's group). Debian (by default[15]) creates one group per user so that only the user is included in its group. Consequently 027 and 077 are equivalent as the user's group contains only the user.

This change is set by defining a proper `umask` setting for all users. You can change this by introducing an **umask** call in the shell configuration files: `/etc/profile` (source by all Bourne-compatible shells), `/etc/csh.cshrc`, `/etc/csh.login`, `/etc/zshrc` and probably some others (depending on the shells you have installed on your system). You can also change the `UMASK` setting in `/etc/login.defs`, Of all of these the last one that gets loaded by the shell takes precedence. The order is: the default system configuration for the user's shell (i.e. `/etc/profile` and other system-wide configuration files) and then the user's shell (his `~/.profile`, `~/.bash_profile`, etc...). Some shells, however, can be executed with a *nologin* value which might skip sourcing some of those files. See your shell's manpage for additional information.

Per connessioni che fanno uso di **login** viene usata, prima di tutte le altre, la configurazione di UMASK definita in `/etc/login.defs`. Ad ogni modo questo valore non viene applicato ai programmi eseguiti dall'utente che non usano **login**, come ad esempio quelli eseguiti per mezzo di **su**, **cron** o **ssh**.

Non dimenticate di rivedere e magari modificare i file che iniziano con il punto sotto `/etc/skel/` dato che questi saranno i file predefiniti dei nuovi utenti quando verranno creati con il comando **adduser**. I

[14] Ttys are spawned for local logins and remote logins through ssh and telnet
[15] As defined in `/etc/adduser.conf` (USERGROUPS=yes). You can change this behaviour if you set this value to no, although it is not recommended

file di Debian predefiniti che iniziano con il punto non includono alcuna chiamata a **umask**, ma se ce ne dovessero essere, i nuovi utenti creati potrebbero avere valori differenti.

Notate comunque che gli utenti possono modificare l'impostazione della propria a propria `umask` se lo vogliono, rendendola più o meno permissiva modificando i propri file che iniziano con il punto.

Il pacchetto libpam-umask regola l'`umask` predefinita degli utenti utilizzando PAM. Aggiungete la seguente riga a `/etc/pam.d/common-session` dopo aver installato il pacchetto:

```
session    optional     pam_umask.so umask=077
```

Infine, dovreste prendere in considerazione il cambiamento dell'umask predefinita 022 di root (come definita in `/root/.bashrc`) con una umask più restrittiva. Questo impedirà all'amministratore di sistema di rimuovere inavvertitamente file sensibili quando lavora come root in directory leggibili da tutti (come `/tmp`) e averli disponibili per il proprio utente medio.

Porre limiti a ciò a cui gli utenti possono accedere

FIXME. Dei contenuti sono necessari. Descrivere le conseguenze relative al cambiare i permessi ai pacchetti quando si aggiornano (ed un admin paranoico dovrebbe mettere in **chroot** i suoi utenti, comunque), se non utilizzando **dpkg-statoverride**.

Se dovete dare accesso shell agli utenti, pensateci bene. Un utente, a meno che non sia in un ambiente pieno di restrizioni (come una gabbia `chroot`), può carpire molte informazioni sul sistema, tra cui:

- some configuration files in `/etc`. However, Debian's default permissions for some sensitive files (which might, for example, contain passwords), will prevent access to critical information. To see which files are only accessible by the root user for example

  find /etc -type f -a -perm 600 -a -uid 0

  as superuser.

- i pacchetti installati, sia guardando il database dei pacchetti in `/usr/share/doc` che tirando ad indovinare guardando gli eseguibili e le librerie installate.

- some log files at `/var/log`. Note also that some log files are only accessible to root and the `adm` group (try

  find /var/log -type f -a -perm 640

) and some are even only available to the root user (try

  find /var/log -type f -a -perm
  600 -a -uid 0

).

Cosa può visualizzare un utente nel vostro sistema? Probabilmente un sacco di cose, provate questo (fate un respiro profondo):

```
find / -type f -a -perm +006 2>/dev/null
find / -type d -a -perm +007 2>/dev/null
```

The output is the list of files that a user can *see* and the accessable directories.

Limitare l'accesso alle informazioni di altri utenti

Se permettete l'accesso tramite shell da parte degli utenti potrebbe essere desiderabile limitare quali informazioni di altri utenti possano vedere. Gli utenti con accesso alla shell tendono a creare un gran numero di file nella loro $HOME: caselle di posta, documenti personali, configurazioni di applicazioni per X/GNOME/KDE...

In Debian ogni utente viene creato con un gruppo associato e due utenti non appartengono mai allo stesso gruppo. Questo è il comportamento predefinito: quando l'utente X viene creato, viene creato anche un gruppo di nome X e l'utente viene assegnato a quel gruppo. Questo evita il concetto di gruppo di *utenti* che renderebbe più difficile per gli utenti stessi nascondere informazioni agli altri.

However, users' `$HOME` directories are created with 0755 permissions (group-readable and world-readable). The group permissions is not an issue since only the user belongs to the group, however the world permissions might (or might not) be an issue depending on your local policy.

You can change this behavior so that user creation provides different `$HOME` permissions. To change the behavior for *new* users when they get created, change *DIR_MODE* in the configuration file `/etc/adduser.conf` to 0750 (no world-readable access).

Users can still share information, but not directly in their `$HOME` directories unless they change its permissions.

Note that disabling world-readable home directories will prevent users from creating their personal web pages in the `~/public_html` directory, since the web server will not be able to read one component in the path - namely their `$HOME` directory. If you want to permit users to publish HTML pages in their `~/public_html`, then change *DIR_MODE* to 0751. This will allow the web server to access the final `public_html` directory (which itself should have a mode of 0755) and provide the content published by users. Of course, we are only talking about a default configuration here; users can generally tune modes of their own files completely to their liking, or you could keep content intended for the web in a separate location which is not a subdirectory of user's `$HOME` directory.

Generare password per gli utenti

Ci sono molti casi in cui un amministratore ha la necessità di creare molti account e di fornire password per ognuno di essi. Ovviamente l'amministratore potrebbe semplicemente scegliere come password il nome scelto dall'utente per l'account, ma ciò non sarebbe molto furbo per quanto riguarda la sicurezza. Un approccio migliore consiste nell'usare un programma per generare password. Debian offre i pacchetti makepasswd, apg e pwgen, che forniscono programmi (con nome uguale a quello del pacchetto) che possono essere usati per questo scopo. **makepasswd** genererà password totalmente casuali, con enfasi sulla sicurezza piuttosto che sulla pronunciabilità, mentre **pwgen** cercherà di creare password senza senso ma pronunciabili (ovviamente questo dipende dalla lingua madre). **Apg** ha algoritmi per entrambi (per questo programma esiste una versione client/server ma non è inclusa nel pacchetto Debian).

Passwd does not allow non-interactive assignation of passwords (since it uses direct tty access). If you want to change passwords when creating a large number of users you can create them using **adduser** with the `--disabled-login` option and then use **usermod** or **chpasswd**[16] (both from the passwd package so you already have them installed). If you want to use a file with all the information to make users as a batch process you might be better off using **newusers**.

[16] **Chpasswd** cannot handle MD5 password generation so it needs to be given the password in encrypted form before using it, with the

`-e`

option.

Controllare le password degli utenti

Le password degli utenti possono talvolta diventare il *componente più debole* nella sicurezza di un sistema. Ciò è dovuto alla scelta da parte degli utenti di password di scarsa qualità per i loro account (e più utenti hanno accesso più le possibilità che questo accada crescono). Anche se sono stati impostati controlli con il modulo PAM cracklib e limiti per quanto riguarda le password, descritti nella sezione chiamata «Autenticazione degli utenti: PAM», gli utenti saranno comunque in grado di usare password deboli. Poiché l'accesso per gli utenti potrebbe includerne uno da shell remote (si spera tramite **ssh**) è importante che la password sia robusta e quindi difficile da indovinare, soprattutto se fossero in qualche modo in grado di raccogliere importanti informazioni come i nomi degli utenti o anche i file `passwd` e `shadow` stessi.

A system administrator must, given a big number of users, check if the passwords they have are consistent with the local security policy. How to check? Try to crack them as an attacker would if having access to the hashed passwords (the `/etc/shadow` file).

An administrator can use john or crack (both are brute force password crackers) together with an appropriate wordlist to check users' passwords and take appropriate action when a weak password is detected. You can search for Debian GNU packages that contain word lists using **apt-cache search wordlist**, or visit some Internet wordlist sites.

Disconnettere gli utenti inattivi

Solitamente gli utenti inattivi sono un problema per la sicurezza. Un utente potrebbe essere inattivo perché è fuori a pranzo o perché una connessione remota interrotta non è stata ristabilita. Qualunque sia la ragione, gli utenti inattivi possono essere causa di problemi:

- perché la console dell'utente potrebbe non essere bloccata e un intruso potrebbe accedervi.
- because an attacker might be able to re-attach to a closed network connection and send commands to the remote shell (this is fairly easy if the remote shell is not encrypted as in the case of **telnet**).

Alcuni sistemi remoti sono stati compromessi attraverso un programma, **screen**, inattivo (distaccato).

La disconnessione automatica degli utenti inattivi di solito è una parte della politica di sicurezza che occorre rafforzare. Ci sono molti modi per realizzarla:

- If bash is the user shell, a system administrator can set a default `TMOUT` value (see bash(1)) which will make the shell automatically log off remote idle users. Note that it must be set with the `-o` option or users will be able to change (or unset) it.
- installare timeoutd e configurare `/etc/timeouts` secondo la propria politica di sicurezza locale. Il demone controllerà se ci sono utenti inattivi e manderà in time out le loro shell secondo la configurazione impostata.
- installare autolog e configurarlo per disconnettere gli utenti inattivi.

I demoni **timeoutd** e **autolog** sono i metodi consigliati, dal momento che gli utenti possono cambiare la loro shell predefinita o passare ad un'altra shell (non controllata) dopo aver avviato la loro shell predefinita.

Usare i tcpwrapper

TCP wrappers were developed when there were no real packet filters available and access control was needed. Nevertheless, they're still very interesting and useful. The TCP wrappers allow you to allow or

deny a service for a host or a domain and define a default allow or deny rule (all performed on the application level). If you want more information take a look at hosts_access(5) manual page.

Molti servizi installati in Debian vengono:

- lanciati mediante un servizio tcpwrapper (`tcpd`).
- compilati con il supporto a libwrapper integrato.

In un caso, per i servizi configurati mediante `/etc/inetd.conf` (inclusi **telnet**, **ftp**, **netbios**, **swat** e **finger**) il file di configurazione esegue prima **/usr/sbin/tcpd**. Nell'altro, anche se il servizio viene lanciato dal superdemone **inetd**, il supporto per le regole dei tcp wrapper può essere compilato al suo interno. In Debian, tra i servizi compilati con il supporto ai tcp wrapper ci sono **ssh, portmap, in.talk, rpc.statd, rpc.mountd, gdm, oaf** (il demone di attivazione GNOME), **nessus** e molti altri.

To see which packages use tcpwrappers [17] try:

```
$ apt-cache rdepends libwrap0
```

Occorre tenerne conto quando usate **tcpdchk** (molto utile per controllare le regole e la sintassi del file di configurazione del TCP wrapper). I servizi compilati col supporto alla libreria wrapper possono essere aggiunti ai file `hosts.deny` e `hosts.allow` ma **tcpdchk** avviserà che non è in grado di trovare questi servizi, perché li cerca in `/etc/inetd.conf` (la pagina man non è molto chiara su questo punto).

Now, here comes a small trick, and probably the smallest intrusion detection system available. In general, you should have a decent firewall policy as a first line, and tcp wrappers as the second line of defense. One little trick is to set up a `SPAWN` [18] command in `/etc/hosts.deny` that sends mail to root whenever a denied service triggers wrappers:

```
ALL: ALL: SPAWN ( \
  echo -e "\n\
  TCP Wrappers\: Connection refused\n\
  By\: $(uname -n)\n\
  Process\: %d (pid %p)\n\
  User\: %u\n\
  Host\: %c\n\
  Date\: $(date)\n\
" | /usr/bin/mail -s "Connection to %d blocked" root) &
```

Attenzione: L'esempio sopra è esposto a attacchi di tipo DoS stabilendo molte connessioni in un breve periodo di tempo. Molte email causano un elevato I/O di file spedendo solo pochi pacchetti.

L'importanza di log e avvisi

È evidente che il modo di trattare log e avvisi è una questione importante in un sistema sicuro. Supponiamo che un sistema sia perfettamente configurato e sicuro al 99%. Se viene portato un attacco al restante 1% e non ci sono misure di sicurezza pronte, innanzitutto a rilevarlo e poi ad attivare allarmi, il sistema non è per nulla sicuro.

[17] On older Debian releases you might need to do this:

```
$ apt-cache showpkg libwrap0 | egrep '^[[:space:]]' | sort -u | \
      sed 's/,libwrap0$//;s/^[[:space:]]\+//'
```

[18] be sure to use uppercase here since *spawn* will not work

Debian GNU/Linux provides some tools to perform log analysis, most notably swatch, [19] logcheck or log-analysis (all will need some customisation to remove unnecessary things from the report). It might also be useful, if the system is nearby, to have the system logs printed on a virtual console. This is useful since you can (from a distance) see if the system is behaving properly. Debian's `/etc/syslog.conf` comes with a commented default configuration; to enable it uncomment the lines and restart **syslogd** (**`/etc/init.d/syslogd restart`**):

```
daemon,mail.*;\
      news.=crit;news.=err;news.=notice;\
      *.=debug;*.=info;\
      *.=notice;*.=warn       /dev/tty8
```

To colorize the logs, you could take a look at colorize, ccze or glark. There is a lot to log analysis that cannot be fully covered here, so a good information resource would be books should as http://books.google.com/books?id=UyktqN6GnWEC. In any case, even automated tools are no match for the best analysis tool: your brain.

Usare e personalizzare logcheck

Il pacchetto **logcheck** in Debian è diviso in tre parti: **logcheck** (il programma principale), logcheck-database (un database di espressioni regolari per il programma) e logtail (visualizza le righe corrispondenti ai log che non sono stati ancora visualizzati). In Debian è predefinito (in `/etc/cron.d/logcheck`) che **logcheck** venga eseguito giornalmente ogni ora e dopo ogni riavvio del sistema.

Questo strumento, se propriamente configurato, può essere molto utile per segnalare all'amministratore eventi inusuali del sistema. **logcheck** può essere completamente personalizzato, in modo da spedire mail a proposito di eventi registrati nei log che sembrano degni di attenzione. L'installazione predefinita include profili (per gli eventi da ignorare e le violazioni delle politiche adottate) per tre diverse configurazioni (workstation, server e paranoid). Il pacchetto Debian include un file di configurazione `/etc/logcheck/logcheck.conf`, generato dal programma, che definisce a quale utente vengono spedite le verifiche. Inoltre fornisce, ai pacchetti che offrono servizi, un modo per implementare nuove politiche, nelle cartelle: `/etc/logcheck/cracking.d/_packagename_`, `/etc/logcheck/violations.d/_packagename_`, `/etc/logcheck/violations.ignore.d/_packagename_`, `/etc/logcheck/ignore.d.paranoid/_packagename_`, `/etc/logcheck/ignore.d.server/_packagename_` e `/etc/logcheck/ignore.d.workstation/_packagename_`. Tuttavia, al momento, non molti pacchetti ne fanno uso. Se avete una politica che può essere utile ad altri utenti, per favore speditela come rapporto bug per il pacchetto appropriato (come *wishlist* bug). Per ulteriori informazioni leggete `/usr/share/doc/logcheck/README.Debian`.

Il modo migliore per configurare **logcheck** dopo l'installazione è modificare il suo principale file di configurazione `/etc/logcheck/logcheck.conf`. Cambiate l'utente predefinito (root) a cui verranno spediti i rapporti. Sempre in quel file di configurazione dovrete anche modificare il livello di prolissità del rapporto. logcheck-database ha tre livelli di prolissità, ovvero: workstation, server, paranoid. "server" è il livello predefinito, "paranoid" è consigliato solo per le macchine che devono garantire un'alta sicurezza per l'espletamento di determinati servizi e "workstation" per sistemi relativamente al riparo, in posizioni non critiche. Se desiderate aggiungere nuovi file di log dovrete solamente aggiungere righe in `/etc/logcheck/logcheck.logfiles`. È già ottimizzato per l'installazione predefinita del demone syslog.

Once this is done you might want to check the mails that are sent, for the first few days/weeks/months. If you find you are sent messages you do not wish to receive, just add the regular expressions (see re-

[19] there's a very good article on it written by http://www.spitzner.net/swatch.html

gex(7) and egrep(1)) that correspond to these messages to the `/etc/logcheck/ignore.d.reportlevel/local`. Try to match the whole logline. Details on howto write rules are explained in `/usr/share/doc/logcheck-database/README.logcheck-database.gz`. It's an ongoing tuning process; once the messages that are sent are always relevant you can consider the tuning finished. Note that if **logcheck** does not find anything relevant in your system it will not mail you even if it does run (so you might get a mail only once a week, if you are lucky).

Configurare il file dove vengono spediti gli avvisi

Debian comes with a standard syslog configuration (in `/etc/syslog.conf`) that logs messages to the appropriate files depending on the system facility. You should be familiar with this; have a look at the `syslog.conf` file and the documentation if not. If you intend to maintain a secure system you should be aware of where log messages are sent so they do not go unnoticed.

Per esempio, mandare i messaggi su una console è una configurazione utile per sistemi di diversi livelli di produzione. Ma per altri sistemi è ugualmente importante aggiungere una nuova macchina che faccia da loghost (cioè che riceva i log da tutte le altre macchine del sistema).

Dovreste considerare anche la posta di root, molti programmi per il controllo della sicurezza (come snort) spediscono gli avvisi a root via posta. Questa mailbox di solito punta al primo utente creato nel sistema (controllate `/etc/aliases`). Preoccupatevi di mandare la posta di root in qualche posto dove verrà letta (localmente o remotamente).

Ci sono altri account di ruolo ed alias nel tuo sistema. In un piccolo sistema é, probabilmente, più facile far sì che tutti gli alias puntino a root e che la posta di root venga inoltrata alla casella di posta personale dell'amministratore di sistema.

FIXME: sarebbe interessante spiegare come un sistema Debian possa spedire/ricevere SNMP trap riguardanti problemi di sicurezza (jfs). Controllare: snmptrapfmt, snmp e snmpd.

Usare un loghost

A loghost is a host which collects syslog data remotely over the network. If one of your machines is cracked, the intruder is not able to cover the tracks, unless hacking the loghost as well. So, the loghost should be especially secure. Making a machine a loghost is simple. Just start the **syslogd** with

```
syslogd -r
```

and a new loghost is born. In order to do this permanently in Debian, edit `/etc/default/syslogd` and change the line

```
SYSLOGD=""
```

in

```
SYSLOGD="-r"
```

Successivamente, configurate le altre macchine perché spediscano i dati al loghost. Aggiungete una riga simile alla seguente al file `/etc/syslog.conf`:

```
  facility.level            @your_loghost
```

Guardate la documentazione per cosa usare al posto di *facility* e *level* (non dovrebbero contenere queste parole). Se volete registrare ogni cosa remotamente, scrivete:

```
*.*                             @your_loghost
```

into your `syslog.conf`. Logging remotely as well as locally is the best solution (the attacker might presume to have covered his tracks after deleting the local log files). See the syslog(3), syslogd(8) and syslog.conf(5) manpages for additional information.

Permessi dei file di log

It is not only important to decide how alerts are used, but also who has read/modify access to the log files (if not using a remote loghost). Security alerts which the attacker can change or disable are not worth much in the event of an intrusion. Also, you have to take into account that log files might reveal quite a lot of information about your system to an intruder who has access to them.

Alcuni permessi sui file di log non sono perfetti dopo una installazione (ma questo ovviamente dipende dalla vostra politica di sicurezza locale). Per prima cosa `/var/log/lastlog` e `/var/log/faillog` non necessitano di essere letti dai normali utenti. Nel file `lastlog` potete vedere chi ha effettuato un login recentemente e in `faillog` potete vedere un riassunto dei login falliti. L'autore raccomanda **chmod 660** per entrambi. Date una rapida occhiata ai vostri file di log e decidete molto attentamente quali rendere leggibili/scrivibili per utenti con UID diversi da 0 e gruppi che non siano 'adm' o 'root'. Si può facilmente controllare se quanto descritto è conforme al vostro sistema con:

```
#  find /var/log -type f -exec ls -l {} \; | cut -c 17-35 |sort -u
(see to what users do files in /var/log belong)
#  find /var/log -type f -exec ls -l {} \; | cut -c 26-34 |sort -u
(see to what groups do files in /var/log belong)
# find /var/log -perm +004
(files which are readable by any user)
#  find /var/log \! -group root \! -group adm -exec ls -ld {} \;
(files which belong to groups not root or adm)
```

Per personalizzare la creazione dei file di log, probabilmente dovreste modificare il programma che li genera. Se i file di log vengono ruotati (vgs. **logrotate**), ad ogni modo, è possibile personalizzare il comportamento di creazione e rotazione.

Includere le patch nel kernel

Debian GNU/Linux fornisce molte patch per il Kernel Linux al fine di migliorarne la sicurezza. Queste includono:

- Linux Intrusion Detection [http://www.lids.org] provided in the kernel-patch-2.4-lids package. This kernel patch makes the process of hardening your Linux system easier by allowing you to restrict, hide and protect processes, even from root. It implements mandatory access control capabilities.
- Linux Trustees [http://trustees.sourceforge.net/], provided in package trustees. This patch adds a decent advanced permissions management system to your Linux kernel. Special objects (called trustees) are bound to every file or directory, and are stored in kernel memory, which allows fast lookup of all permissions.
- NSA Enhanced Linux (in package selinux). Backports of the SElinux-enabled packages are available at https://salsa.debian.org/selinux-team. More information available at SElinux in Debian Wiki page

[http://wiki.debian.org/SELinux], at Manoj Srivastava's [http://www.golden-gryphon.com/software/security/selinux.xhtml] and Russell Cookers's [http://www.coker.com.au/selinux/] SElinux websites.

- The kernel patch http://people.redhat.com/mingo/exec-shield provided in the kernel-patch-exec-shield package. This patch provides protection against some buffer overflows (stack smashing attacks).

- The Grsecurity patch [http://www.grsecurity.net/], provided by the kernel-patch-2.4-grsecurity and kernel-patch-grsecurity2 packages [20] implements Mandatory Access Control through RBAC, provides buffer overflow protection through PaX, ACLs, network randomness (to make OS fingerprinting more difficult) and many more features [http://www.grsecurity.net/features.php].

- The kernel-patch-adamantix provides the patches developed for Adamantix [http://www.adamantix.org/], a Debian-based distribution. This kernel patch for the 2.4.x kernel releases introduces some security features such as a non-executable stack through the use of http://pageexec.virtualave.net/ and mandatory access control based on http://www.rsbac.org/. Other features include: http://www.vanheusden.com/Linux/sp/, AES encrypted loop device, MPPE support and an IPSEC v2.6 backport.

- cryptoloop-source. Questa patch consente di usare funzionalità crypto API del kernel per creare dei filesystem cifrati usando il dispositivo di loopback.

- Supporto ad IPSEC nel kernel (nel pacchetto linux-patch-openswan). Se volete usare il protocollo IPsec con Linux, occorre questa patch, con la quale potrete creare delle VPN, anche per macchine Windows, in modo molto semplice, dal momento che IPsec è uno standard affermato. Le funzionalità IPSec sono state aggiunte al kernel di sviluppo 2.5 e queste caratteristiche verranno incluse in modo predefinito nel futuro kernel 2.6. La pagina web di riferimento: http://www.openswan.org. *FIXME*: Gli ultimi kernel 2.4 forniti in Debian includono un backport del codice IPSEC dal 2.5. Commenti su questo.

Le seguenti deprecate patch per il kernel, dedicate alla sicurezza, sono disponibili solamente per le vecchie versioni dei kernel di woody:

- http://acl.bestbits.at/ (ACLs) per Linux, fornita dal pacchetto kernel-patch-acl. Questa patch per il kernel aggiunge elenchi di controllo degli accessi ed un metodo avanzato per limitare l'accesso ai file. Consente di controllare in modo granulare l'accesso a file e directory.

- La patch per il kernel linux http://www.openwall.com/linux/ della Solar Designer, fornita nel pacchetto kernel-patch-2.2.18-openwall contiene un utile insieme di restrizioni per il kernel, come link limitati, FIFO in `/tmp`, un filesystem `/proc` limitato, la gestione di descrittori speciali di file, area di stack utente non eseguibile ed altre funzionalità. Notate: questo pacchetto si applica alla versione 2.2, mentre non sono disponibili pacchetti per le patch alla versione 2.4 fornite da Solar.

- kernel-patch-int. Questa patch estende le funzioni di criptazione del kernel e può essere usata dalle versioni Debian fino a Potato. Non funziona con Woody e se usate Sarge o una versione successiva dovreste semplicemente usare un kernel più recente visto che include già questa funzionalità.

Comunque in Debian alcune patch ancora non sono state fornite. Se ritenete che alcune di queste dovrebbero essere incluse siete pregati di chiedere di loro mediante il http://www.debian.org/devel/wnpp/.

[20] Notice that this patch conflicts with patches already included in Debian's 2.4 kernel source package. You will need to use the stock vanilla kernel. You can do this with the following steps:

```
# apt-get install kernel-source-2.4.22 kernel-patch-debian-2.4.22
# tar xjf /usr/src/kernel-source-2.4.22.tar.bz2
# cd kernel-source-2.4.22
# /usr/src/kernel-patches/all/2.4.22/unpatch/debian
```

For more information see http://bugs.debian.org/194225, http://bugs.debian.org/199519, http://bugs.debian.org/206458, http://bugs.debian.org/203759, http://bugs.debian.org/204424, http://bugs.debian.org/210762, http://bugs.debian.org/211213, and the http://lists.debian.org/debian-devel/2003/09/msg01133.html

Protezione contro i buffer overflow

Buffer overflow è il nome di un comune attacco al software[21] che approfitta di un controllo insufficiente (un errore di programmazione molto comune in C) in seguito all'esecuzione di codice immesso da un input. Questo tipo di attacco, portato contro i server che rimangono in attesa di una connessione remota e contro il software che gira localmente, che solitamente assegnar alti privilegi agli utenti (grazie ai `setuid` e `setgid` attivi) può compromettere ogni sistema.

Esistono principalmente quattro metodi per proteggersi dai buffer overflow:

- Aggiungendo delle patch al kernel così da prevenire l'esecuzione dello stack.Potete usare: Exec-shield, OpenWall o PaX (incluso nelle patch Grsecurity ed Adamantix).
- Fissando il codice usando strumenti per trovare frammenti del sorgente che potrebbero introdurre vulnerabilità.
- recompile the source code to introduce proper checks that prevent overflows, using the http://www.research.ibm.com/trl/projects/security/ssp/ patch for GCC (which is used by http://www.adamantix.org)

Debian GNU/Linux, as of the 3.0 release, provides software to introduce all of these methods except for the protection on source code compilation (but this has been requested in http://bugs.debian.org/213994).

Notate che se Debian fornisse un compilatore con protezione contro stack o buffer overflow tutti i pacchetti dovrebbero essere ricompilati per poter introdurre questa funzionalità. Questo è, in effetti, quello che fa la distribuzione Adamantix (tra le altre caratteristiche). L'effetto di questa nuova funzionalità sul software è ancora da stabilire (alcuni programmi o alcune architetture di processori potrebbero non funzionare).

In any case, be aware that even these workarounds might not prevent buffer overflows since there are ways to circumvent these, as described in phrack's magazine http://packetstorm.linuxsecurity.com/mag/phrack/phrack58.tar.gz or in CORE's Advisory http://online.securityfocus.com/archive/1/269246.

Una volta implementato, se desiderate collaudare il vostro livello di protezione da buffer overflow (indipendentemente dal metodo scelto), potreste voler installare paxtest ed eseguire il test che fornisce.

Patch per la protezione del kernel da Kernel contro buffer overflows

Le patch relative al buffer overflow includono la patch Openwall che fornisce protezione contro gli overflow nel kernel 2.2. Per il kernel 2.4 o per quelli ancor più recenti, è necessario utilizzare l'implementazione Exec-shield o altrimenti PaX (fornita sia dalla patch grsecurity, kernel-patch-2.4-grsecurity, che dalla patch Adamantix, kernel-patch-adamantix). Per maggiori informazioni sull'uso di queste patch leggete sezione chiamata «Includere le patch nel kernel».

Collaudare i programmi contro gli overflow

L'uso di strumenti per trovare buffer overflows richiede, in ogni caso, un'esperienza da programmatori per sistemare (e ricompilare) il codice. Debian, fornisce, per esempio: bfbtester (un programma per collaudare la resistenza ai buffer overflow che utilizza binari con approccio a forza bruta mediante overflow da riga di comando o di ambiente). Altri pacchetti interessanti potrebbero essere anche rats, pscan, flawfinder e splint.

[21] Così comune, in effetti, che sono alla base del 20% delle vulnerabilità di sicurezza riportate tutti gli anni, come stabilito da http://icat.nist.gov/icat.cfm?function=statistics.

Trasferire file in sicurezza

Durante la normale amministrazione, solitamente si ha bisogno di importare o di esportare dei file dal sistema installato. Si può riuscire a copiare i file da un host ad un altro in maniera sicura, usando il pacchetto server ssh. Un'altra possibilità è l'impiego di ftpd-ssl, un server ftp che adotta il *Secure Socket Layer* per cifrare le trasmissioni.

Naturalmente, tutti questi metodi necessitano di client speciali e Debian ne offre alcuni: per esempio, con ssh fornisce **scp**, che funziona come **rcp** ma è completamente cifrato, cosicché i *cattivi ragazzi* non possano nemmeno capire CHE COSA si stia copiando. C'è anche un pacchetto client ftp-ssl per il server corrispondente. Per questi programmi, si trovano client anche di altri sistemi operativi (non-UNIX); per copiare in sicurezza, **putty** e **winscp** forniscono un'implementazione adatta a qualsiasi versione del sistema operativo della Microsoft.

Notate che usare **scp** consente a tutti gli utenti l'accesso all'intero file-system, a meno che non abbiate usato **chroot**, come spiegato in sezione chiamata «Ssh in chroot». Si può configurare l'accesso FTP, attraverso **chroot**, in modo anche più semplice, a seconda del demone scelto, come illustrato in sezione chiamata «Rendere sicuro FTP». Se temete che gli utenti sfoglino i file locali e volete avere una comunicazione cifrata, potete utilizzare un demone ftp con supporto SSL, o abbinare un ftp che trasmette testo in chiaro e un'impostazione VPN (cfr. sezione chiamata «Rete privata virtuale (VPN)»).

Limitazioni e controllo del File System

Usare le quote

Avere un buon criterio di assegnazione delle quote è importante, poiché evita che gli utenti riempiano gli hard disk.

Si possono avere due differenti sistemi di quote: la quota d'utente e la quota di gruppo. Come si intuisce, la quota per utente limita la quantità di spazio di cui un utente può disporre e la quota di gruppo fa la stessa cosa ma con i gruppi. Ricordatevene quando decidete la dimensione delle quote.

Nell'impostare un sistema di quote, bisogna tenere conto di alcuni punti importanti:

- Fare in modo che le quote siano abbastanza piccole, in modo che gli utenti non divorino lo spazio del disco fisso.
- Fare in modo che siano abbastanza grandi, in modo che gli utenti non abbiano a lagnarsene e che la loro quota-email impedisca loro di accettare posta per un lungo periodo.
- Usare il sistema delle quote su tutte le aree scrivibili dagli utenti, su `/home` come su `/tmp`.

Ogni partizione o cartella a cui gli utenti abbiano pieno accesso in scrittura dovrebbe essere organizzata in quote. Il calcolo e l'assegnazione di una quota su cui si possa lavorare, combina utilizzabilità e sicurezza.

Supponiamo che vogliate usare il sistema delle quote: prima di tutto, dovrete controllare che il supporto per le quote sia abilitato nel kernel, se non lo fosse andrebbe ricompilato; dopo di che, controllate che il pacchetto quota sia installato, altrimenti dovrete procedere all'installazione del pacchetto.

Enabling quota for the respective file systems is as easy as modifying the `defaults` setting to `defaults,usrquota` in your `/etc/fstab` file. If you need group quota, substitute `usrquota` to `grpquota`. You can also use them both. Then create empty quota.user and quota.group files in the roots of the file systems you want to use quotas on (e.g.

```
touch
/home/quota.user /home/quota.group
```

for a `/home` file system).

Restart **quota** by doing

```
/etc/init.d/quota stop;/etc/init.d/quota
        start
```

. Now quota should be running, and quota sizes can be set.

Editing quotas for a specific user can be done by

```
edquota -u <user>
```

. Group quotas can be modified with

```
edquota -g <group>
```

. Then set the soft and hard quota and/or inode quotas as needed.

Per maggiori informazioni sulle quote, leggete le relative pagine man ed il quota mini-HOWTO (`/usr/share/doc/HOWTO/en-html/mini/Quota.html` - mini guida su quota). Potreste anche voler dare un'occhiata a `pam_limits.so`.

The ext2 filesystem specific attributes (chattr/lsattr)

Oltre ai soliti permessi di tipo Unix, i filesystem ext2 ed ext3 offrono un insieme di attributi specifici per dare un maggiore controllo sui file del sistema. A differenza dei permessi di base, questi non vengono mostrati con il comando **ls -l** o modificati mediante **chmod**; per amministrarli, occorrono altre due utilità: **lsattr** e **chattr** (nel pacchetto e2fsprogs). Notate che ciò significa che tali attributi non verranno salvati con la copia di sicurezza del sistema; così, qualora se ne modifichi uno qualsiasi, sarebbe meglio salvare in uno script i successivi comandi **chattr**, così da poterli reimpostare in un secondo momento, all'atto di un eventuale ripristino.

Fra tutti gli attributi disponibili, i due più importanti, nell'aumentare la sicurezza vengono richiamati dalle lettere 'i' ed 'a' e possono essere impostati o rimossi dal superutente:

- L'attributo 'i' ('immutabile'): un file con questo attributo non può essere modificato, né cancellato, né rinominato e nemmeno il superutente può creare collegamenti ad esso.

- L'attributo 'a' ('append'): questo attributo ha lo stesso effetto del precedente, salvo che consente di aprire il file per aggiungervi nuovi contenuti, pur senza poter modificare quello già esistente (append mode); è molto utile per i file di log collocati nella cartella `/var/log/`, anche se dovreste considerare che, talvolta, questi vengono spostati, per via degli script di rotazione dei log.

Si possono impostare questi attributi anche alle cartelle; in tal caso nessuno può modificarne i contenuti, rinominando o rimuovendo dei file. Applicato ad una cartella, l'attributo append permette la sola creazione di file.

It is easy to see how the 'a' attribute improves security, by giving to programs that are not running as the superuser the ability to add data to a file without modifying its previous content. On the other hand, the 'i' attribute seems less interesting: after all, the superuser can already use the basic Unix permissions to restrict access to a file, and an intruder that would get access to the superuser account could always use the **chattr** program to remove the attribute. Such an intruder may first be confused when noticing not being

able to remove a file, but you should not assume blindness - after all, the intruder got into your system! Some manuals (including a previous version of this document) suggest to simply remove the **chattr** and **lsattr** programs from the system to increase security, but this kind of strategy, also known as "security by obscurity", is to be absolutely avoided, since it provides a false sense of security.

A secure way to solve this problem is to use the capabilities of the Linux kernel, as described in sezione chiamata «Difesa preventiva». The capability of interest here is called `CAP_LINUX_IMMUTABLE`: if you remove it from the capabilities bounding set (using for example the command **lcap `CAP_LINUX_IMMUTABLE`**) it won't be possible to change any 'a' or 'i' attribute on your system anymore, even by the superuser ! A complete strategy could be as follows:

- Impostare gli attributi 'a' ed 'i' sui file desiderati.
- Add the command **lcap `CAP_LINUX_IMMUTABLE`** (as well as **lcap `CAP_SYS_MODULE`**, as suggested in sezione chiamata «Difesa preventiva») to one of the startup scripts;
- Impostare l'attributo 'i' su tale script e su altri file di avvio, o anche sullo stesso binario **lcap**.
- Eseguire manualmente detto comando (o riavviare il sistema, per assicurarsi che tutto funzioni a dovere).

Now that the capability has been removed from the system, an intruder cannot change any attribute on the protected files, and thus cannot change or remove the files. If the machine is forced to reboot (which is the only way to restore the capabilities bounding set), it will easily be detected, and the capability will be removed again as soon as the system restarts anyway. The only way to change a protected file would be to boot the system in single-user mode or using another bootdisk, two operations that require physical access to the machine !

Controllare l'integrità del file system

Siete sicuri che `/bin/login` sul disco fisso è ancora il binario installato qualche mese fa? Cosa accadrebbe se fosse una versione modificata, che registra le password inserite in un file nascosto o le spedisce in chiaro per tutta Internet?

Il solo modo per avere qualche forma di protezione è controllare i file ogni ora/giorno/mese (preferibile quotidianamente) confrontando gli md5sum attuali di un file con quelli vecchi. Due file non possono avere lo stesso md5sum (l'MD5 digest è 128 bits, quindi le possibilità che due diversi file abbiano lo stesso md5sum sono approssimativamente una su 3.4e3803), perciò è il metodo più sicuro, a meno che qualcuno non abbia modificato l'algoritmo che crea l'md5sum sulla macchina; il che comunque è piuttosto complicato ed improbabile. Bisogna considerare la verifica di questi binari molto importante, poiché è un modo semplice per riconoscere le modifiche ai binari.

Strumenti comunemente utilizzati a questo scopo sono sxid, aide (Advanced Intrusion Detection Environment), tripwire, integrit e samhain. Installare **debsums** può aiutare a controllare l'integrità del file system, comparando l'md5sum di ogni file con quello usato nell'archivio dei pacchetti Debian. Ma attenzione, questi file possono facilmente essere modificati da un intruso malintenzionato ed inoltre non tutti i pacchetti offrono un elenco md5sum di tutti i file che forniscono. Per ulteriori informazioni leggete sezione chiamata «Effettuate periodicamente dei controlli sull'integrità del sistema» e sezione chiamata «Una fotografia del sistema».

You might want to use **locate** to index the whole filesystem, if so, consider the implications of that. The Debian findutils package contains **locate** which runs as user nobody, and so it only indexes files which are visible to everybody. However, if you change it's behaviour you will make all file locations visible to all users. If you want to index all the filesystem (not the bits that the user nobody can see) you can replace **locate** with the package slocate. slocate is labeled as a security enhanced version of GNU locate, but it actually provides additional file-locating functionality. When using **slocate**, the user only sees the

actually accessible files and you can exclude any files or directories on the system. The slocate package runs its update process with higher privledges than locate, and indexes every file. Users are then able to quickly search for every file which they are able to see. **slocate** doesn't let them see new files; it filters the output based on your UID.

Potreste voler usare bsign o elfsign. elfsign fornisce un'utility che aggiunge una firma elettronica a un binario ELF ed una seconda utility per verificare questa firma. L'attuale implementazione usa PKI (Public Key Infrastructure) per firmare il cheksum del binario. I benefici di questa pratica sono quelli di permettere di verificare se un binario è stato modificato e chi lo ha creato. bsign usa GPG, elfsign usa certificati PKI (X.509) (OpenSSL).

Impostare il controllo di setuid

Il pacchetto Debian checksecurity fornisce un processo **cron** che viene eseguito giornalmente attraverso `/etc/cron.daily/checksecurity`[22]. Questo processo **cron** farà partire lo script **/usr/sbin/checksecurity** che memorizzerà le informazioni riguardo questi cambiamenti.

The default behavior does not send this information to the superuser but, instead keeps daily copies of the changes in `/var/log/setuid.changes`. You should set the MAILTO variable (in `/etc/checksecurity.conf`) to 'root' to have this information mailed to the superuser. See checksecurity(8) manual page for more configuration info.

Rendere sicuro l'accesso alla rete

FIXME. Servono più contenuti (specifici per Debian)

Configurare le caratteristiche di rete del kernel

Many features of the kernel can be modified while running by echoing something into the `/proc` file system or by using **sysctl**. By entering **/sbin/sysctl** `-A` you can see what you can configure and what the options are, and it can be modified running

```
/sbin/sysctl -w variable=value
```

(see sysctl(8)). Only in rare cases do you need to edit something here, but you can increase security that way as well. For example:

```
net/ipv4/icmp_echo_ignore_broadcasts = 1
```

Questo è un *emulatore di Windows* perché funziona come Windows in broadcast ping, se questa opzione viene impostata ad 1. In pratica, le richieste ICMP_ECHO spedite all'indirizzo in broadcast vengono ignorate. Altrimenti, non succede nulla.

Se volete evitare che il sistema risponda alle richieste echo ICMP, basta attivare questa opzione di configurazione:

```
net/ipv4/icmp_echo_ignore_all = 1
```

Per loggare dei pacchetti destinati ad indirizzi inesistenti (errore dovuto ad instradamenti sbagliati) sulla vostra rete, utilizzate:

[22] Nelle distribuzioni precedenti, checksecurity era integrato in cron ed il file quindi era `/etc/cron.daily/standard`.

```
/proc/sys/net/ipv4/conf/all/log_martians = 1
```

For more information on what things can be done with `/proc/sys/net/ipv4/*` read `/usr/src/linux/Documentation/filesystems/proc.txt`. All the options are described thoroughly under `/usr/src/linux/Documentation/networking/ip-sysctl.txt`[23].

Configurare i Syncookies

Questa opzione è un'arma a doppio taglio. Da un lato, protegge il sistema contro il syn packet flooding, dall'altro viola degli standard definiti (le RFC).

```
net/ipv4/tcp_syncookies = 1
```

If you want to change this option each time the kernel is working you need to change it in `/etc/network/options` by setting `syncookies=yes`. This will take effect when ever `/etc/init.d/networking` is run (which is typically done at boot time) while the following will have a one-time effect until the reboot:

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

Questa opzione è disponibile solamente se il kernel è stato compilato con l'opzione `CONFIG_SYNCOOKIES`. Tutti i kernel Debian vengono compilati con questa opzione all'interno del kernel, potete verificarlo eseguendo:

```
$ sysctl -A |grep syncookies
net/ipv4/tcp_syncookies = 1
```

Per maggiori informazioni sui syncookies TCP leggete http://cr.yp.to/syncookies.html.

Rendere sicura la rete al momento del boot

Quando impostate le opzioni del kernel relative al networking, dovete configurarle in maniera tale che siano caricate ogni volta che il sistema viene riavviato. L'esempio seguente abilita molte delle precedenti opzioni che avete visto, insieme ad altre utili opzioni.

There are actually two ways to configure your network at boot time. You can configure `/etc/sysctl.conf` (see: sysctl.conf(5)) or introduce a script that is called when the interface is enabled. The first option will be applied to all interfaces, whileas the second option allows you to configure this on a per-interface basis.

Sotto viene mostrato un esempio di configurazione di `/etc/sysctl.conf` che rende sicure alcune opzioni di rete a livello di kernel. Guardate la parte commentata, `/etc/network/options` potrebbe ignorare alcuni valori, se in contrasto con quelli in questo file quando viene eseguito `/etc/init.d/networking` (che viene dopo `procps` nella sequenza di boot).

```
#
```

[23] In Debian the kernel-source-`version` packages copy the sources to `/usr/src/kernel-source-version.tar.bz2`, just substitute `version` to whatever kernel version sources you have installed

```
# /etc/sysctl.conf - Configuration file for setting system variables
# See sysctl.conf (5) for information. Also see the files under
# Documentation/sysctl/, Documentation/filesystems/proc.txt, and
# Documentation/networking/ip-sysctl.txt in the kernel sources
# (/usr/src/kernel-$version if you have a kernel-package installed)
# for more information of the values that can be defined here.

#
# Be warned that /etc/init.d/procps is executed to set the following
# variables.  However, after that, /etc/init.d/networking sets some
# network options with builtin values.  These values may be overridden
# using /etc/network/options.
#
#kernel.domainname = example.com

# Additional settings - adapted from the script contributed
# by Dariusz Puchala (see below)
# Ignore ICMP broadcasts
net/ipv4/icmp_echo_ignore_broadcasts = 1
#
# Ignore bogus ICMP errors
net/ipv4/icmp_ignore_bogus_error_responses = 1
#
# Do not accept ICMP redirects (prevent MITM attacks)
net/ipv4/conf/all/accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
# net/ipv4/conf/all/secure_redirects = 1
#
# Do not send ICMP redirects (we are not a router)
net/ipv4/conf/all/send_redirects = 0
#
# Do not forward IP packets (we are not a router)
# Note: Make sure that /etc/network/options has 'ip_forward=no'
net/ipv4/conf/all/forwarding = 0
#
# Enable TCP Syn Cookies
# Note: Make sure that /etc/network/options has 'syncookies=yes'
net/ipv4/tcp_syncookies = 1
#
# Log Martian Packets
net/ipv4/conf/all/log_martians = 1
#
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
# Note: Make sure that /etc/network/options has 'spoofprotect=yes'
net/ipv4/conf/all/rp_filter = 1
#
# Do not accept IP source route packets (we are not a router)
net/ipv4/conf/all/accept_source_route = 0
```

Per usare lo script dovete prima di tutto crearlo, per esempio, in `/etc/network/interface-secure` (il nome è solo un esempio) e chiamarlo da `/etc/network/interfaces` come mostrato qui:

```
auto eth0
iface eth0 inet static
        address xxx.xxx.xxx.xxx
        netmask 255.255.255.xxx
        broadcast xxx.xxx.xxx.xxx
        gateway xxx.xxx.xxx.xxx
        pre-up /etc/network/interface-secure
```

In questo esempio, prima di abilitare l'interfaccia eth0, lo script verrà chiamato per rendere sicure tutte le interfacce di rete come mostrato sotto.

```
#!/bin/sh -e
# Script-name: /etc/network/interface-secure
#
# Modifica alcuni comportamenti predefiniti per proteggere il
# sistema da alcuni attacchi spoofing contro il TCP/IP e da altri
# tipi di attacchi rivolti a tutte le interfacce.
#
# Contributed by Dariusz Puchalak.
#
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
                                        # Broadcast echo protection enabled.
echo 0 > /proc/sys/net/ipv4/conf/all/forwarding
                                        # IP forwarding disabled.
echo 1 > /proc/sys/net/ipv4/tcp_syncookies # TCP syn cookies protection enabled.
echo 1 >/proc/sys/net/ipv4/conf/all/log_martians # Log strange packets.
# (this includes spoofed packets, source routed packets, redirect packets)
# but be careful with this on heavy loaded web servers.
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
                                        # Bad error message protection enabled.

# IP spoofing protection.
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter

# Disable ICMP redirect acceptance.
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
echo 0 > /proc/sys/net/ipv4/conf/all/send_redirects

# Disable source routed packets.
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route

exit 0
```

È opportuno notare che si possono adottare script specifici per ogni interfaccia. Questi script abilitano le opzioni di rete relative alle diverse interfacce (qualora ne fossero presenti più di una). È sufficiente modificare il comando di pre-up come indicato di seguito:

```
pre-up /etc/network/interface-secure $IFACE
```

Questo comando lancia lo script che applicherà i cambiamenti solamente all'interfaccia di rete specificata e non a tutte le interfacce eventualmente disponibili. Notate che alcune opzioni di rete possono essere applicate solo globalmente. Di seguito un modello di script:

```
#!/bin/sh -e
# Script-name: /etc/network/interface-secure
#
# Modifica alcuni comportamenti predefiniti per proteggere il
# sistema da alcuni attacchi spoofing contro il TCP/IP e da altri
# tipi di attacchi rivolti a tutte le interfacce.
#
# Contributed by Dariusz Puchalak.
#

IFACE=$1
if [ -z "$IFACE" ] ; then
   echo "$0: Must give an interface name as argument!"
   echo "Usage: $0 <interface>"
   exit 1
fi

if [ ! -e /proc/sys/net/ipv4/conf/$IFACE/ ]; then
   echo "$0: Interface $IFACE does not exit (cannot find /proc/sys/net/ipv4/conf/)
   exit 1
fi

echo 0 > /proc/sys/net/ipv4/conf/$IFACE/forwarding  # IP forwarding disabled.
echo 1 >/proc/sys/net/ipv4/conf/$IFACE/log_martians # Log strange packets.
# (questo include spoofed packets, source routed packets, redirect packets)
# ma fate attenzione, può essere pesante per un server web.

# IP spoofing protection.
echo 1 > /proc/sys/net/ipv4/conf/$IFACE/rp_filter

# Disable ICMP redirect acceptance.
echo 0 > /proc/sys/net/ipv4/conf/$IFACE/accept_redirects
echo 0 > /proc/sys/net/ipv4/conf/$IFACE/send_redirects

# Disable source routed packets.
echo 0 > /proc/sys/net/ipv4/conf/$IFACE/accept_source_route

exit 0
```

An alternative solution is to create an `init.d` script and have it run on bootup (using **update-rc.d** to create the appropriate `rc.d` links).

Configurare le caratteristiche di un firewall

Per avere un firewall, sia per proteggere il sistema locale o tutto quello che si trova *dietro*, dovete compilare nel kernel le funzionalità del firewall. Il kernel standard per Debian 2.2 (Linux 2.2) comprende il packet filter **ipchains** come firewall, in Debian 3.0 è presente anche il kernel della serie 2.4 che è invece munito dello *stateful* packet filter, **iptables** (netfilter) come firewall.

In any case, it is pretty easy to use a kernel different from the one provided by Debian. You can find pre-compiled kernels as packages you can easily install in the Debian system. You can also download the kernel sources using the kernel-source-`X` and build custom kernel packages using **make-kpkg** from the kernel-package package.

La configurazione del firewall sarà ampiamente trattata in sezione chiamata «Aggiungere funzionalità al firewall».

Disabilitare la questione weak-end host

Systems with more than one interface on different networks can have services configured so that they will bind only to a given IP address. This usually prevents access to services when requested through any other address. However, this does not mean (although it is a common misconception) that the service is bound to a given *hardware* address (interface card). [24]

It seems, however, not to work with services bound to 127.0.0.1, you might need to write the tests using raw sockets.

This is not an ARP issue and it's not an RFC violation (it's called *weak end host* in RFC1122 [ftp://ftp.isi.edu/in-notes/rfc1122.txt], (in the section 3.3.4.2). Remember, IP addresses have nothing to do with physical interfaces.

Nelle versioni 2.2 del kernel ed anche nelle precedenti è possibile porre rimedio con:

```
# echo 1 > /proc/sys/net/ipv4/conf/all/hidden
# echo 1 > /proc/sys/net/ipv4/conf/eth0/hidden
# echo 1 > /proc/sys/net/ipv4/conf/eth1/hidden
.....
```

Nei kernel più recenti si può ottenere lo stesso con:

- Regole per iptable.
- properly configured routing. [25]
- kernel patching. [26]

Along this text there will be many occasions in which it is shown how to configure some services (sshd server, apache, printer service...) in order to have them listening on any given address, the reader should

[24] To reproduce this (example provided by Felix von Leitner on the Bugtraq mailing list):

```
   host a (eth0 connected to eth0 of host b):
     ifconfig eth0 10.0.0.1
     ifconfig eth1 23.0.0.1
     tcpserver -RHl localhost 23.0.0.1 8000 echo fnord

   host b:
     ifconfig eth0 10.0.0.2
     route add 23.0.0.1 gw 10.0.0.1
     telnet 23.0.0.1 8000
```

[25] The fact that this behavior can be changed through routing was described by Matthew G. Marsh in the Bugtraq thread:

```
eth0 = 1.1.1.1/24
eth1 = 2.2.2.2/24

ip rule add from 1.1.1.1/32 dev lo table 1 prio 15000
ip rule add from 2.2.2.2/32 dev lo table 2 prio 16000

ip route add default dev eth0 table 1
ip route add default dev eth1 table 2
```

[26] There are some patches available for this behavior as described in Bugtraq's thread at http://www.linuxvirtualserver.org/~julian/#hidden and http://www.fefe.de/linux-eth-forwarding.diff.

take into account that, without the fixes given here, the fix would not prevent accesses from within the same (local) network. [27]

FIXME: i commenti tratti da bugtraq sono metodi specifici per proteggere un data interfaccia in Linux.

FIXME: sottoporre un bug nei confronti di netbase in modo che il risultato della riparazione della tabella di routing sia il comportamento standard per Debian?

Proteggersi dagli attacchi di tipo ARP

Quando non c'è piena fiducia verso le altre postazioni sulla propria LAN (dovrebbe essere sempre così, è l'atteggiamento più sicuro), ci si dovrebbe proteggere dai diversi possibili attacchi di tipo ARP.

As you know the ARP protocol is used to link IP addresses to MAC addresses (see ftp://ftp.isi.edu/in-notes/rfc826.txt for all the details). Every time you send a packet to an IP address an ARP resolution is done (first by looking into the local ARP cache then if the IP isn't present in the cache by broadcasting an ARP query) to find the target's hardware address. All the ARP attacks aim to fool your box into thinking that box B's IP address is associated to the intruder's box's MAC address; Then every packet that you want to send to the IP associated to box B will be send to the intruder's box...

Quegli attacchi (ARP cache poisonning, ARP spoofing - avvelenamento della cache ARP, falsificazioni ARP) permettono all'attaccante di intercettare il traffico anche su reti commutate, di dirottare facilmente delle connessioni, di disconnettere un host qualunque dalla rete... gli attacchi ARP sono potenti e semplici da implementare, essendovi parecchi strumenti utili allo scopo: **arpspoof** dal pacchetto dsniff o http://arpoison.sourceforge.net/.

Tuttavia, una soluzione c'è sempre:

- Usare una cache ARP statica, nella quale impostare delle voci "statiche":

  ```
  arp -s host_name hdwr_addr
  ```

  Impostando voci statiche per ciascun host importante presente nella rete vi assicurerete che nessuno crei/modifichi per detti host la voce (la falsifichi) - le voci statiche non hanno scadenza e non possono essere modificate - cosicché le repliche ARP falsificate verranno ignorate.

- Detect suspicious ARP traffic. You can use arpwatch, karpski or more general IDS that can also detect suspicious ARP traffic (snort, http://www.prelude-ids.org...).

- Implementare un filtraggio del traffico IP in grado di convalidare gli indirizzi MAC.

Una fotografia del sistema

Prima di mettere il sistema in produzione se ne può fotografare l'attuale stato: il risultato dell'operazione può tornare utile in caso di compromissione (vedete Capitolo 11, *Dopo la compromissione (reazione agli incidenti)*). Questo procedimento dovrebbe essere ripetuto ad ogni aggiornamento del sistema, specialmente se si aggiorna ad una nuova versione di Debian.

A tale scopo si può usare un media scrivibile e rimuovibile che possa essere impostato in sola lettura, come un floppy disk, se protetto da scrittura dopo l'uso, o un CD aperto da un'unità CD-ROM - si può usare

[27] An attacker might have many problems pulling the access through after configuring the IP-address binding while not being on the same broadcast domain (same network) as the attacked host. If the attack goes through a router it might be quite difficult for the answers to return somewhere.

un CD riscrivibile, in modo da poter conservare copie di ripristino, magari con firma md5sum, create in date differenti, o un disco USB o ancora una card MMC (se il vostro sistema può accedere a tali dati e al contempo possono essere protetti in scrittura).

Il seguente script crea una fotografia del sistema:

```
#!/bin/bash
/bin/mount /dev/fd0 /mnt/floppy
trap "/bin/umount /dev/fd0" 0 1 2 3 9 13 15
if [ ! -f /usr/bin/md5sum ] ; then
 echo "Cannot find md5sum. Aborting."
 exit 1
fi
/bin/cp /usr/bin/md5sum /mnt/floppy
echo "Calculating md5 database"
>/mnt/floppy/md5checksums.txt
for dir in /bin/ /sbin/ /usr/bin/ /usr/sbin/ /lib/ /usr/lib/
do
   find $dir -type f | xargs /usr/bin/md5sum >>/mnt/floppy/md5checksums-lib.txt
done
echo "post installation md5 database calculated"
if [ ! -f /usr/bin/sha1sum ] ; then
 echo "Cannot find sha1sum"
        echo "WARNING: Only md5 database will be stored"
else
 /bin/cp /usr/bin/sha1sum /mnt/floppy
 echo "Calculating SHA-1 database"
 >/mnt/floppy/sha1checksums.txt
 for dir in /bin/ /sbin/ /usr/bin/ /usr/sbin/ /lib/ /usr/lib/
 do
    find $dir -type f | xargs /usr/bin/sha1sum >>/mnt/floppy/sha1checksums-lib.txt
 done
 echo "post installation sha1 database calculated"
fi
exit 0
```

Tenete conto che il binario di md5sum (e sha1sum se disponibile) si trova su floppy, per poter essere usato, in seguito, per controllare i binari del sistema (nel caso che abbiate preso dei trojan). Comunque, se volete essere sicuri che state facendo girare un binario legittimo, potreste sia compilare una copia statica del binario di md5sum e usare quella (per evitare che una libreria libc con trojan interferisca con il binario), oppure usare una istantanea di md5sum solamente da un ambiente sano come un CD-ROM di salvataggio o un Live-CD (per evitare che un kernel con trojan interferisca). Questo non verrà mai posto abbastanza in risalto: se siete su un sistema compromesso non potete fidarvi del suo output, vedete Capitolo 11, *Dopo la compromissione (reazione agli incidenti)*.

L'istantanea non comprende i file sotto `/var/lib/dpkg/info` che includono l'hash MD5 dei pacchetti installati (nei file che finiscono con `.md5sums`). Potete anche copiare questa informazione, ma dovreste comunque tenere in conto che:

- i file md5sum includono l'md5sum non solo dei binari di sistema, ma di tutti i file forniti dai pacchetti Debian. Una conseguenza di ciò è che il database è più grande (5 Mb contro 600 Kb in un sistema Debian GNU/Linux con sistema grafico e circa 2.5 Gb di software installato) e non entrerà in un supporto rimovibile di scarsa capienza (tipo un floppy disk, ma probabilmente entrerebbe in una memoria rimovibile USB).

- non tutti i pacchetti Debian forniscono l'md5sum per i file installati fino a che non sarà (attualmente lo è) una pratica obbligatoria. Tenete in conto che, in ogni caso, avrete la possibilità di generare l'md5sum per tutti i pacchetti usando debsums una volta terminata l'installazione del sistema:

```
# debsums --generate=missing,keep
```

Una volta fatta l'istantanea dovreste assicurarvi di impostare in sola lettura il dispositivo. Potrete poi archiviarlo per backup o metterlo in un disco ed usare **cron** ogni notte per confrontare l'md5sum originale con quella dell'istantanea.

Se non volete approntare un controllo manuale potrete sempre usare uno dei programmi di controllo dell'integrità del sistema disponibili che faranno questo e anche di più, per maggiori informazioni leggete sezione chiamata «Effettuate periodicamente dei controlli sull'integrità del sistema».

Ulteriori raccomandazioni

Non usare software che dipende dalle librerie SVGA (svgalib)

SVGAlib è molto carina per gli amanti della console come me, ma è stato provato diverse volte, in passato, che è molto insicura. Sono stati rilasciati exploit contro **zgv** ed era semplice utilizzarli per diventare root. Sarebbe meglio evitare di usare programmi che fanno uso della SVGAlib, quando possibile.

Capitolo 5. Rendere più sicuri i servizi che girano sul vostro sistema

I servizi attivi su un sistema possono essere resi più sicuri in due modi:

- Rendendoli accessibili solo dai punti di accesso (interfacce) per i quali sono necessari.
- Configurandoli opportunamente cosicché possano essere usati solo dagli utenti legittimati con un metodo di autorizzazione.

Vincolare i servizi in maniera tale da renderli accessibili solo da un luogo definito può essere fatto limitando il loro accesso a livello di kernel (per es. i firewall), configurandoli in maniera tale da ascoltare solo da una data interfaccia (alcuni servizi potrebbero non fornire questa possibilità) o impiegando qualche altro metodo, per esempio la patch linux vserver (per il 2.4.16) può essere usata per obbligare i processi ad usare una sola interfaccia.

Regarding the services running from **inetd** (**telnet**, **ftp**, **finger**, **pop3**...) it is worth noting that **inetd** can be configured so that services only listen on a given interface (using `service@ip` syntax) but that's an undocumented feature. One of its substitutes, the **xinetd** meta-daemon includes a `bind` option just for this matter. See ixnetd.conf(5) manual page.

```
service nntp
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = news
        group           = news
        server          = /usr/bin/env
        server_args     = POSTING_OK=1 PATH=/usr/sbin/:/usr/bin:/sbin/:/bin
+/usr/sbin/snntpd logger -p news.info
        bind            = 127.0.0.1
}
```

Le sezioni seguenti forniscono dettagli su come configurare opportunamente singoli e specifici servizi in funzione dell'uso previsto.

Rendere sicuro ssh

Se state ancora usando telnet invece di ssh, è meglio che passiate a ssh prima di proseguire con la lettura di questo manuale. Sarebbe bene utilizzare ssh invece di telnet per tutti i login remoti. In un'epoca in cui è facile intercettare il traffico su Internet e ottenere le password trasmesse in chiaro, bisognerebbe usare solamente protocolli che utilizzano la crittografia; per questo motivo meglio eseguire subito `apt-get install ssh` sulla propria macchina.

Incoraggiate tutti gli utenti del vostro sistema ad usare ssh invece di telnet o, ancora meglio, disinstallare telnet/telnetd. Oltre a quanto detto bisognerebbe evitare di autenticarsi nel sistema come root con ssh e utilizzare invece dei metodi alternativi per diventare root, come **su** o **sudo**. Infine, meglio modificare anche il file `sshd_config`, in `/etc/ssh`, per aumentare la sicurezza:

- `ListenAddress 192.168.0.1` Have ssh listen only on a given interface, just in case you have more than one (and do not want ssh available on it) or in the future add a new network card (and don't want ssh connections from it).
- `PermitRootLogin no` Try not to permit Root Login wherever possible. If anyone wants to become root via ssh, now two logins are needed and the root password cannot be brute forced via SSH.
- `Port 666` or `ListenAddress 192.168.0.1:666` Change the listen port, so the intruder cannot be completely sure whether a sshd daemon runs (be forewarned, this is security by obscurity).
- `PermitEmptyPasswords no` Empty passwords make a mockery of system security.
- `AllowUsers alex ref me@somewhere` Allow only certain users to have access via ssh to this machine. `user@host` can also be used to restrict a given user from accessing only at a given host.
- `AllowGroups wheel admin` Allow only certain group members to have access via ssh to this machine. AllowGroups and AllowUsers have equivalent directives for denying access to a machine. Not surprisingly they are called "DenyUsers" and "DenyGroups".
- `PasswordAuthentication yes` It is completely your choice what you want to do. It is more secure to only allow access to the machine from users with ssh-keys placed in the `~/.ssh/authorized_keys` file. If you want so, set this one to "no".
- Disable any form of authentication you do not really need, if you do not use, for example `RhostsRSAAuthentication`, `HostbasedAuthentication`, `KerberosAuthentication` or `RhostsAuthentication` you should disable them, even if they are already by default (see the manpage sshd_config(5) manual page).
- `Protocol 2` Disable the protocol version 1, since it has some design flaws that make it easier to crack passwords. For more information read http://earthops.net/ssh-timing.pdf or the http://xforce.iss.net/static/6449.php.
- `Banner /etc/`*`some_file`* Add a banner (it will be retrieved from the file) to users connecting to the ssh server. In some countries sending a warning before access to a given system about unauthorized access or user monitoring should be added to have legal protection.

Potete limitare l'accesso al server ssh anche utilizzando le direttive `pam_listfile` o `pam_wheel` nel file di configurazione PAM per ssh, in modo da limitare le autenticazioni ssh. Per esempio potreste escludere tutti gli utenti non elencati in `/etc/loginusers` aggiungendo questa riga a `/etc/pam.d/ssh`:

```
auth       required     pam_listfile.so sense=allow onerr=fail item=user file=/etc
```

Come ultima osservazione dovreste considerare che queste direttive si riferiscono ad un file di configurazione di OpenSSH. Al momento vengono utilizzati comunemente tre demoni SSH: ssh1, ssh2 e l'OpenSSH degli sviluppatori di OpenBSD. ssh1 è stato il primo demone ssh disponibile ed è ancora il più utilizzato (ci sono voci che esista anche una versione per Windows). ssh2 ha molti vantaggi rispetto a ssh1 tranne per il fatto di essere rilasciato con una licenza closed-source. OpenSSH è un demone ssh rilasciato come software libero che supporta sia ssh1 che ssh2. OpenSSH è la versione che viene installata in Debian quando selezionate il pacchetto ssh.

You can read more information on how to set up SSH with PAM support in the http://lists.debian.org/debian-security/2001/11/msg00395.html.

Ssh in chroot

Per ora OpenSSH non fornisce un modo per poter ingabbiare automaticamente, con chroot, gli utenti dopo una connessione (la versione commerciale invece fornisce questa possibilità). Ad ogni modo esiste un pro-

getto per dare questa funzionalità anche ad OpenSSH, vedete http://chrootssh.sourceforge.net, al momento credo non sia disponibile il pacchetto per Debian. Potreste usare al suo posto il modulo `pam_chroot` come descritto in sezione chiamata «Restrizioni agli utenti per l'accesso».

In sezione chiamata «Chroot environment for SSH» potete trovare numerose opzioni per creare un ambiente chroot per SSH.

Client SSH

Se usate un client SSH con un server SSH dovreste assicurarvi che supporti gli stessi protocolli impostati sul server. Per esempio, se utilizzate il pacchetto mindterm, questo supporta solo il protocollo con versione 1. Mentre, il server sshd, in modo predefinito, viene configurato per accettare solo la versione 2 (per ragioni di sicurezza).

Non permettere il trasferimento di file

Se *non* volete che gli utenti trasferiscano file da e verso il server ssh dovete limitare l'accesso all'**sftp-server** *e* l'accesso a **scp**. Potete limitare l'**sftp-server** configurando il corretto `Subsystem` in `/etc/ssh/sshd_config`.

Potete anche ingabbiare in chroot gli utenti (usando libpam-chroot) cosicché, nonostante il trasferimento di file sia permesso, gli utenti siano costretti in un ambiente limitato che non comprende file di sistema.

Limitare l'accesso al solo trasferimento di file

Potreste desiderare di limitare l'accesso agli utenti al solo trasferimento di file e non concedere loro shell interattive. A questo scopo potete anche:

- non permettere agli utenti il login per mezzo del server ssh (come descritto in precedenza, sia mediante il suo file di configurazione che tramite il file di configurazione di PAM).
- concedere agli utenti una shell limitata come scponly o rssh. Queste shell limitano i comandi disponibili agli utenti in modo da non fornire alcun privilegio di esecuzione remota.

La sicurezza in Squid

Squid is one of the most popular proxy/cache server, and there are some security issues that should be taken into account. Squid's default configuration file denies all users requests. However the Debian package allows access from 'localhost', you just need to configure your browser properly. You should configure Squid to allow access to trusted users, hosts or networks defining an Access Control List on `/etc/squid/squid.conf`, see the https://web.archive.org/web/20061206052115/http://www.deckle.co.za/squid-users-guide/Main_Page for more information about defining ACLs rules. Notice that Debian provides a minimum configuration for Squid that will prevent anything, except from *localhost* to connect to your proxy server (which will run in the default port 3128). You will need to customize your `/etc/squid/squid.conf` as needed.

The recommended minimum configuration (provided with the package) is shown below:

```
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
```

```
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443 563     # https, snews
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 901         # SWAT
acl purge method PURGE
acl CONNECT method CONNECT
(...)
# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager
# Only allow purge requests from localhost
http_access allow purge localhost
http_access deny purge
# Deny requests to unknown ports
http_access deny !Safe_ports
# Deny CONNECT to other than SSL ports
http_access deny CONNECT !SSL_ports
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
http_access allow localhost
# And finally deny all other access to this proxy
http_access deny all
#Default:
# icp_access deny all
#
#Allow ICP queries from everyone
icp_access allow all
```

Dovreste configurare Squid anche basandovi sulle vostre risorse di sistema, inclusa la memoria cache (opzione `cache_mem`), la posizione dei file nella cache e la quantità di spazio che occuperanno sull'hard disk (opzione `cache_dir`).

Da notare che, se non correttamente configurato, qualcuno potrebbe inoltrare un messaggio di posta tramite Squid, poiché i protocolli HTTP e SMTP sono abbastanza simili. La configurazione predefinita di Squid nega l'accesso alla porta 25. Se desiderate permettere connessioni alla porta 25 aggiungetela alla lista delle Safe_ports. Comunque, questo *NON* è raccomandato.

Impostare e configurare correttamente il server proxy/cache è solo una parte di lavoro per mantenere il vostro sito sicuro. Un altro compito necessario è l'analisi dei log di squid per assicurarsi che tutto vada come dovrebbe. Ci sono alcuni pacchetti in Debian GNU/Linux che potrebbero aiutare un amministratore a farlo. I seguenti sono disponibili in Debian 3.0 e Debian 3.1 (sarge):

- calamaris - Analizzatore di log per i file dei proxy Squid ed Oops.
- modlogan - Un analizzatore modulare dei file di log.
- sarg - Genera rapporti sull'analisi di Squid.

- squidtaild - Un programma per monitorare i log di Squid.

When using Squid in Accelerator Mode it acts as a web server too. Turning on this option increases code complexity, making it less reliable. By default Squid is not configured to act as a web server, so you don't need to worry about this. Note that if you want to use this feature be sure that it is really necessary. To find more information about Accelerator Mode on Squid see the https://web.archive.org/web/20070104164802/http://www.deckle.co.za/squid-users-guide/Accelerator_Mode

Rendere sicuro FTP

Se avete realmente la necessità di usare il servizio FTP (senza poterlo costringere in un tunnel SSL o SSH o tramite sslwrap), dovreste usare chroot per ingabbiare FTP nella directory home dell'utente ftp, cosicché l'utente non sia in grado di vedere niente altro se non la sua directory. Altrimenti potrebbe attraversare il filesystem principale come se avesse una shell nel sistema. Potete aggiungere la seguente riga nel vostro `proftpd.conf` nella sezione globale per abilitare la funzione chroot:

```
DefaultRoot ~
```

Fate ripartire ProFTPd con `/etc/init.d/proftpd restart` e controllate se adesso riuscite ad uscire dalla vostra directory home.

To prevent ProFTPd DoS attacks using ../../.., add the following line in `/etc/proftpd.conf`: `DenyFilter \*.*/`

Always remember that FTP sends login and authentication passwords in clear text (this is not an issue if you are providing an anonymous public service) and there are better alternatives in Debian for this. For example, **sftp** (provided by ssh). There are also free implementations of SSH for other operating systems: http://www.chiark.greenend.org.uk/~sgtatham/putty/ and http://www.cygwin.com for example.

However, if you still maintain the FTP server while making users access through SSH you might encounter a typical problem. Users accessing anonymous FTP servers inside SSH-secured systems might try to log in the *FTP server*. While the access will be refused, the password will nevertheless be sent through the net in clear form. To avoid that, ProFTPd developer TJ Saunders has created a patch that prevents users feeding the anonymous FTP server with valid SSH accounts. More information and patch available at: http://www.castaglia.org/proftpd/#Patches. This patch has been reported to Debian too, see http://bugs.debian.org/145669.

Rendere sicuro l'accesso al sistema X Window

Oggi, i terminali X vengono usati da numerose e svariate compagnie, con il sistema X Window è necessario un solo server per numerose postazioni singole. Questo potrebbe essere pericoloso perché è necessario permettere al file server di connettersi ai client (X server dal punto di vista di X. X scambia le solite definizioni di client e server). Se seguirete il (pessimo) suggerimento di numerosi documenti, digiterete `xhost +` sulla vostra macchina. Questo permette ad ogni client X di connettersi al sistema. Per una piccola miglioria di sicurezza, potrete invece usare il comando `xhost +hostname` per permettere l'accesso da specifici host.

Una soluzione più sicura, credo, è quella di usare ssh per creare un tunnel per X e cifrare l'intera sessione. Questo viene fatto automaticamente quando vi connettete via ssh ad un'altra macchina. Perché questo funzioni dovete configurare sia il client che il server ssh. Per il client ssh dovrete abilitare `ForwardX11` con un `yes` in `/etc/ssh/ssh_config`. Per il server ssh, invece, dovrete mettere un `yes` in `/etc/ssh/sshd_config` a fianco di `X11Forwarding` e il package xbase-clients dovrebbe essere instal-

lato perché il server ssh usi `/usr/X11R6/bin/xauth` (`/usr/bin/xauth` su Debian instabile) nel momento in cui configura uno pseudo display X. Con SSH, dovreste aggirare completamente il controllo dell'accesso con xhost.

Per una migliore sicurezza, se non avete bisogno di accedere ad X da altre macchine tagliate il legame alla porta TCP 6000 semplicemente digitando:

```
$ startx -- -nolisten tcp
```

Questo è il comportamento predefinito in Xfree 4.1.0 (il server X fornito da Debian 3.0). Se state eseguendo Xfree 3.3.6 (cioè se avete Debian 2.2 installata) potete modificare `/etc/X11/xinit/xserverrcc` per avere qualcosa di simile a:

```
#!/bin/sh
exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

If you are using XDM set `/etc/X11/xdm/Xservers` to: `:0 local /usr/bin/X11/X vt7 -dpi 100 -nolisten tcp`. If you are using Gdm make sure that the `DisallowTCP=true` option is set in the `/etc/gdm/gdm.conf` (which is the default in Debian). This will basically append `-nolisten tcp` to every X command line [1].

Potete anche impostare il timeout di sistema predefinito per il lock dello **xscreensaver**. Anche se l'utente può sovrascriverla, dovreste modificare la configurazione `/etc/X11/app-defaults/XScreenSaver` e cambiare la riga del lock:

```
*lock:                  False
```

(che è il predefinito di Debian) in:

```
*lock:                  True
```

FIXME: aggiungere informazioni su come disabilitare gli screensaver che mostrano il desktop dell'utente (che potrebbe avere informazioni sensibili).

Leggete di più sulla sicurezza di X Window in http://www.tldp.org/HOWTO/XWindow-User-HOWTO.html (`/usr/share/doc/HOWTO/en-txt/XWindow-User-HOWTO.txt.gz`).

FIXME: Aggiungere informazioni prese da un thread di debian-security su come fare a cambiare i file di configurazione di XFree 3.3.6.

Controllare il display manager

Se avete un display manager installato per uso locale (avete cioè un piacevole login grafico), assicuratevi che XDMCP (Protocollo di controllo per il Display Manager di X) sia disabilitato. In XDM questo si può ottenere agendo sulla riga in `/etc/X11/xdm/xdm-config`:

```
DisplayManager.requestPort:     0
```

[1] Gdm will *not* append `-nolisten tcp` if it finds a `-query` or `-indirect` on the command line since the query wouldn't work.

Per GDM ci dovrebbe essere questo nel vostro gdm.conf:

```
[xdmcp]
Enable=false
```

Di norma, in Debian, tutti i display manager vengono configurati per non lanciare servizi XDMCP.

Rendere sicuri gli accessi alla stampante (specifico per lpd ed lprng)

Immaginate di arrivare al lavoro e trovarvi una montagna di carta uscita dalla stampante perché il demone di stampa ha subito un attacco DoS. Lo trovereste piacevole?

In qualsiasi architettura per la stampa in Unix, ci deve essere un modo per ottenere dal client i dati per il server di stampa. Tradizionalmente, **lpr** ed **lp** sono i client che si occupano di fornire i comandi per copiare o creare dei link simbolici ai dati contenuti nella directory spool (che è il motivo per il quale usualmente questi programmi sono SUID o SGID).

Per evitare questi problemi, dovreste mantenere il vostro server di stampa particolarmente sicuro. Questo significa necessariamente, configurare il vostro servizio di stampa per far sì che consenta connessioni solamente dai server fidati. In merito a questo, aggiungete i server a cui volete consentire la stampa nel vostro `/etc/hosts.lpd`.

In ogni caso, il demone **lpr** accetta in ingresso connessioni sulla porta 515 da qualsiasi interfaccia. Dovreste considerare la possibilità di usare un firewall per le connessioni tra la rete e gli host a cui non è consentito stampare (così il demone **lpr** può rimanere in attesa solo da determinati indirizzi IP).

Lprng dovrebbe essere preferito a **lpr** visto che può essere configurato per avere il controllo sugli accessi IP. Potete anche specificare quale interfaccia proteggere (sebbene piuttosto bizzarro).

If you are using a printer in your system, but only locally, you will not want to share this service over a network. You can consider using other printing systems, like the one provided by cups or http://pdq.sourceforge.net/ which is based on user permissions of the `/dev/lp0` device.

In cups, la stampa dei dati viene trasferita al server mediante il protocollo http. Questo significa che il client non ha bisogno di particolari privilegi, ma richiede che il server sia in ascolto su qualche porta.

Tuttavia, se desiderate usare **cups** in locale potete configurarlo proteggendo l'interfaccia di loopback, modificando il file `/etc/cups/cupsd.conf`:

```
Listen 127.0.0.1:631
```

Vi sono molte altre opzioni per la sicurezza simili a quelle contenute nel file di configurazione hosts, che permettono di consentire o negare l'accesso dalla rete. Tuttavia, se non ne avete bisogno, una soluzione migliore potrebbe essere quella di limitare o escludere le connessioni sulle porte in attesa. **Cups** utilizza la porta di comunicazione HTTP, se desiderate non divulgare informazioni potenzialmente utili ad eventuali attaccanti, aggiungete (e chiudete verso l'esterno le porte aperte) anche:

```
<Location />
 Order Deny,Allow
```

```
 Deny From All
 Allow From 127.0.0.1
</Location>
```

This configuration file can be modified to add some more features including SSL/TLS certificates and crypto. The manuals are available at http://localhost:631/ or at http://cups.org.

FIXME: Add more content (the article on http://www.rootprompt.org provides some very interesting views).

FIXME: Controllare se PDG è disponibile in Debian e se lo fosse, suggerirlo come sistema di stampa preferito.

FIXME: Controllare se Farmer/Wietse ha rimpiazzato il demone di stampa e se è disponibile in Debian.

Rendere sicuro il servizio di posta

Se il vostro server non fornisce servizi di posta, non avete bisogno di avere un demone di posta in attesa di connessioni. Potreste aver bisogno solamente di un sistema di trasporto locale, che ad esempio riceva la posta per l'utente root e dagli altri allarmi di sistema.

Se state usando **exim** non avete bisogno di configurare il demone, in quanto, in maniera predefinita, si assume **cron** il compito di svuotare la coda di posta. Vedete sezione chiamata «Disabilitare i servizi attivi in modalità demone» per sapere come questo avviene.

Configurare un nullmailer

Potreste voler avere un demone di posta locale che possa ritrasmettere verso un altro sistema i messaggi spediti localmente. Questa è una cosa comune quando dobbiamo amministrare un certo numero di sistemi e non desideriamo connetterci ad ognuno di essi per leggere la posta spedita localmente. Proprio come la scrittura dei log di ogni singolo sistema può essere centralizzata usando un server syslog centrale, la posta può essere spedita ad un server di posta centralizzato.

Un tale sistema *solo-redirezione (relay-only)* dovrebbe essere propriamente configurato per svolgere questo compito. Il demone potrebbe anche essere configurato per rimanere in ascolto sul solo indirizzo di loopback.

I seguenti passaggi per la configurazione si rendono necessari solo per configurare il pacchetto exim nel rilascio di Debian 3.0. Nell'utilizzo di un rilascio successivo (come ad esempio per 3.1 che usa exim4) il sistema d'installazione è stato migliorato in maniera tale che se l'MTA viene configurato per smistare solamente la posta locale, automaticamente allora permetterà connessioni solo dalla macchina locale e non consentirà connessioni remote.

In un sistema Debian 3.0 che usa exim, dovrete rimuovere da **inetd** il demone SMTP.

```
$ update-inetd --disable smtp
```

e configurare il demone di posta perché rimanga in ascolto sulla sola interfaccia di loopback. In **exim** (l'MTA predefinito) lo potete fare modificando il file `/etc/exim.conf` e aggiungendo la linea seguente:

```
local_interfaces = "127.0.0.1"
```

Riavviare entrambi i demoni (inetd e exim); exim sarà in ascolto sul solo socket 127.0.0.1:25. Fate attenzione e per prima cosa disabilitate inetd, altrimenti exim non partirà poiché il demone inetd sta già gestendo le connessioni in arrivo.

Per **postfix** modificate `/etc/postfix/main.conf`:

```
inet_interfaces = localhost
```

Se volete solo posta locale, questo approccio è migliore dell'uso del tcp-wrapping sul demone di posta o dell'aggiunta di regole per il firewall per limitarne l'accesso. Tuttavia, se avete bisogno che esso resti in ascolto su altre interfacce, lo potreste lanciare da inetd ed aggiungere un tcp wrapper in modo che le connessioni in arrivo vengano controllate tramite i file `/etc/hosts.allow` e `/etc/hosts.deny`. Inoltre, configurando un'appropriata scrittura dei log per qualunque dei metodi sopra descritti, potrete sapere quando si verifica un tentativo di accesso non autorizzato al demone di posta.

In ogni caso, per respingere i tentativi di ritrasmissione della posta a livello di SMTP, potete cambiare `/etc/exim/exim.conf` in modo che contenga:

```
receiver_verify = true
```

Anche se il vostro server di posta non ritrasmetterà il messaggio, questo tipo di configurazione è necessaria al test di ritrasmissione che trovate all'indirizzo http://www.abuse.net/relay.html per determinare che il vostro server *non* sia in grado di ritrasmettere.

If you want a relay-only setup, however, you can consider changing the mailer daemon to programs that can *only* be configured to forward the mail to a remote mail server. Debian provides currently both ssmtp and nullmailer for this purpose. In any case, you can evaluate for yourself any of the mail transport agents [2] provided by Debian and see which one suits best to the system's purposes.

Fornire un accesso sicuro alle mailbox

If you want to give remote access to mailboxes there are a number of POP3 and IMAP daemons available.[3] However, if you provide IMAP access note that it is a general file access protocol, it can become the equivalent of a shell access because users might be able to retrieve any file that they can through it.

Provate, ad esempio, a configurare `{server.com}/etc/passwd` come percorso della vostra inbox. Se ci riuscite questo significa che il demone IMAP non è correttamente configurato per impedire questo tipo di accesso.

Tra i server IMAP disponibili in Debian il server **cyrus** (nel pacchetto cyrus-imapd) risolve il problema, facendo in modo che tutti gli accessi siano rivolti verso un database che risiede in una parte del file system dove l'accesso è ristretto. Inoltre, **uw-imapd** (installare **uw-imapd** o meglio, se il vostro client IMAP lo supporta, uw-imapd-ssl) può essere configurato per ottenere la cartella della posta degli utenti in chroot,

[2] To retrieve the list of mailer daemons available in Debian try:

```
$ apt-cache search mail-transport-agent
```

The list will not include **qmail**, which is distributed only as source code in the qmail-src package.

[3] A list of servers/daemons which support these protocols in Debian can be retrieved with:

```
$ apt-cache search pop3-server
$ apt-cache search imap-server
```

ma questa funzionalità non è abilitata nella configurazione predefinita. La documentazione a corredo del programma fornisce ulteriori informazioni su come configurarlo.

Inoltre, potreste voler eseguire un server IMAP che non necessiti di utenti validi creati sul sistema locale (cosa che consentirebbe anche l'accesso tramite shell); sia courier-imap (per IMAP) che courier-pop teapop (per POP3) e cyrus-imapd (per POP3 e IMAP) forniscono server con metodi di autenticazione non dipendenti dagli account degli utenti locali. **Cyrus** può usare qualunque metodo di autenticazione configurabile per mezzo di PAM, mentre **teapop** può usare dei database (come postgresql e mysql) per l'autenticazione degli utenti.

FIXME: Controllare: anche uw-imapd potrebbe essere configurato per l'autenticazione utenti mediante PAM.

Ricevere posta in sicurezza

Nella ricezione e lettura della posta viene impiegato il più comune protocollo con testo in chiaro; usando sia IMAP che POP3, si invia la propria password in chiaro, in questo modo, da quel momento in avanti, quasi chiunque può leggere la nostra posta; per evitare ciò, scaricatela usando il protocollo SSL o, in alternativa, ssh se avete un'account dotato di shell sulla postazione che funge da server POP o IMAP. Ecco un essenziale `fetchmailrc` esemplificativo:

```
poll my-imap-mailserver.org via "localhost"
  with proto IMAP port 1236
      user "ref" there with password "hackme" is alex here warnings 3600
    folders
      .Mail/debian
    preconnect 'ssh -f -P -C -L 1236:my-imap-mailserver.org:143 -l ref
     my-imap-mailserver.org sleep 15 </dev/null > /dev/null'
```

La linea che inizia con "preconnect" è importante, in quanto dà il via ad una sessione ssh e crea il tunnel necessario, che inoltra le connessioni alla porta 1236 del localhost verso il server di posta IMAP, in modo automatico, ma sottoponendole a cifratura. Un'altra possibilità sarebbe usare fetchmail con la funzionalità SSL.

Se volete fornire servizi di posta cifrata, come POP e IMAP, usate il comando `apt-get install stunnel` e attivate i demoni in questo modo:

```
stunnel -p /etc/ssl/certs/stunnel.pem -d pop3s -l /usr/sbin/popd
```

Questo ultimo comando collega il demone fornito (-l) alla porta (-d) e utilizza lo specificato modo di certificazione SSL (-p).

Rendere sicuro BIND

Ci sono diversi problemi da affrontare per rendere sicuro il demone del domain server, questi sono simili a quelli che incontrate normalmente quando vengono resi sicuri altri servizi:

- Configurare il demone stesso in modo che non possa essere possibile abusarne dall'esterno (vedete sezione chiamata «Configurazione di Bind per evitare abusi»). Questo include limitare la possibilità di richieste da parte dei client: trasferimenti di zona e richieste ricorsive.

- Limitare l'accesso del demone al server stesso di modo che, in caso di accesso indesiderato tramite il demone, i danni al sistema siano limitati. Questo include far girare il demone come utente non privile-

giato (vedete sezione chiamata «Cambiare l'utente di BIND») in un ambiente chrooted (vedete sezione chiamata «Eseguire il name server in chroot»).

Configurazione di Bind per evitare abusi

Dovreste limitare alcune delle informazioni che vengono servite tramite DNS ai client esterni, in modo che non possa essere usato per reperire importanti informazioni, che non volete far conoscere, sulla vostra organizzazione. Questo consiste nell'aggiungere le seguenti opzioni: *allow-transfer*, *allow-query*, *allow-recursive* e *version*. Potete anche limitarle nella sezione globale (in modo che vengano applicate in tutte le zone servite) oppure per zona. Questa informazione viene documentata nel pacchetto bind-doc, potrete leggere di più su questo argomento in `/usr/share/doc/bind/html/index.html` una volta che avrete installato il pacchetto.

Immaginate che il vostro server sia connesso ad Internet ed alla vostra rete interna, con vostro IP locale impostato a 192.168.1.2 (un server base con due connessioni di rete), non volete offrire nessun servizio su Internet e volete solamente abilitare il la risoluzione DNS per vostri host interni. Potete effettuare quest'operazione includendo le seguenti righe in `/etc/bind/named.conf`:

```
options {
            allow-query { 192.168.1/24; } ;
            allow-transfer { none; } ;
            allow-recursion { 192.168.1/24; } ;
            listen-on { 192.168.1.2; } ;
            forward { only; } ;
            forwarders { A.B.C.D; } ;
};
```

L'opzione *listen-on* vincola il DNS sull'interfaccia che ha l'indirizzo interno e anche se questa interfaccia è la stessa che si connette ad Internet (ad esempio se state utilizzando NAT), le richieste verranno accettate solamente se provengono dai vostri host interni. Se il sistema ha interfacce multiple e non è presente il parametro *listen-on*, solo gli utenti interni potranno effettuare richieste, anche se, visto che la porta sarebbe accessibile agli attacchi dall'esterno, qualcuno potrebbe cercare di farvi andare in crash (o provando attacchi di tipo buffer overflow) il server DNS. Potete anche fare in modo che il DNS ascolti unicamente su 127.0.0.1 se non dovete fornire il servizio ad altri sistemi tranne che a voi stessi.

Il record version.bind della classe chaos contiene la versione del processo bind che sta girando correntemente. Questa informazione viene spesso usata da scanner automatici o individui maliziosi con l'intento di determinare se bind è vulnerabile o meno ad un dato attacco. Non fornendo informazioni o fornendole false all'interno del record version.bind, vengono limitate le probabilità che il server venga attaccato sulla base della versione pubblicata. Per fornire la vostra versione utilizzate la direttiva *version* nella seguente maniera:

```
 options { ... various options here ...
version "Not available."; };
```

Cambiare il record version.bind non fornisce protezione a fronte di eventuali attacchi, ma può essere considerata un'utile salvaguardia.

Un esempio di configurazione del file `named.conf` potrebbe essere la seguente:

```
acl internal {
        127.0.0.1/32;           // localhost
        10.0.0.0/8;             // internal
```

```
        aa.bb.cc.dd;            // eth0 IP
};

acl friendly {
        ee.ff.gg.hh;            // slave DNS
        aa.bb.cc.dd;            // eth0 IP
        127.0.0.1/32;           // localhost
        10.0.0.0/8;             // internal
};

options {
        directory "/var/cache/bind";
        allow-query { internal; };
        allow-recursion { internal; };
        allow-transfer { none; };
};
// From here to the mysite.bogus zone
// is basically unmodified from the debian default
logging {
        category lame-servers { null; };
        category cname { null; };
};

zone "." {
        type hint;
        file "/etc/bind/db.root";
};

zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};

// zones I added myself
zone "mysite.bogus" {
        type master;
        file "/etc/bind/named.mysite";
        allow-query { any; };
        allow-transfer { friendly; };
```

```
};
```

Controllate il Bug Tracking System riguardo a Bind, specificatamente il http://bugs.debian.org/94760. Sentitevi liberi di riportare informazioni sui bug che riscontrate, se pensate di poter aggiungere informazioni utili.

Cambiare l'utente di BIND

Riguardo il come limitare i privilegi di BIND dovete sapere che se un utente normale (quindi non superutente) fa partire BIND, questo ultimo non potrà riconoscere le nuove interfacce automaticamente, ad esempio se inserite una scheda PCMCIA nel vostro portatile. Controllate il file README.Debian nella vostra documentazione di named (`/usr/share/doc/bind/README.Debian`) per ulteriori informazioni su questo problema. Ci sono stati recentemente diversi problemi di sicurezza riguardo BIND per cui cambiare utente diviene comodo quando è possibile farlo. Spiegheremo dettagliatamente i passi da compiere per realizzare questo procedimento, tuttavia, se volete farlo in modo automatizzato potreste provare lo script inserito in sezione chiamata «Script di esempio per modificare l'installazione predefinita di Bind».

Notate che, in ogni caso, ciò si applica solo alla versione 8 di BIND. Nei pacchetti Debian della versione 9 di BIND (dalla versione 9.2.1-5, disponibile a partire da *sarge*) l'utente *bind* viene creato ed usato impostando la variabile OPTIONS in `/etc/default/bind9`. Se state usando la versione 9 di BIND ed il vostro domain name server non sta girando come utente *bind*, verificate le impostazioni contenute in quel file.

Per fare girare BIND sotto un altro utente, la prima cosa da fare è creare un utente separato ed un gruppo apposito (*non è* una buona idea usare nobody e nogroup per tutti i servizi che non girano come root). In questo esempio verranno utilizzati l'utente ed il gruppo `named`. Potete crearli digitando:

```
addgroup named
adduser --system --home /home/named --no-create-home --ingroup named \
      --disabled-password --disabled-login named
```

Notate che l'utente `named` sarà parecchio limitato. Se volete, per una qualsiasi ragione, avere un utente con meno limitazioni, utilizzate:

```
adduser --system --ingroup named named
```

Adesso modificate il file `/etc/init.d/bind`, con il vostro editor preferito, la riga che inizia con:

```
start-stop-daemon --start
```

in[4]:

```
start-stop-daemon --start --quiet --exec /usr/sbin/named -- -g named -u named
```

Altrimenti potete cambiare (creare se già non esiste) il file di configurazione predefinito (`/etc/default/bind` per la versione 8 di BIND) ed inserire le seguenti voci:

[4] Notate che, a seconda della vostra versione di bind, potreste non avere l'opzione `-g`, in particolare se state usando bind 9 presente in sarge (versione 9.2.4).

```
OPTIONS="-u named -g named"
```

Cambiate i permessi dei file usati da bind, incluso `/etc/bind/rndc.key`:

```
-rw-r-----    1 root     named          77 Jan  4 01:02 rndc.key
```

ed anche dove bind crea il suo pidfile, usando, ad esempio, `/var/run/named` invece di `/var/run`:

```
$ mkdir /var/run/named
$ chown named.named /var/run/named
$ vi /etc/named.conf
[ ... update the configuration file to use this new location ...]
options { ...
        pid-file "/var/run/named/named.pid";
};
[ ... ]
```

Inoltre, per evitare di far girare qualcosa come superutente, modificate la riga di `reload` nello script init.d da così:

```
reload)
       /usr/sbin/ndc reload
```

a così:

```
reload)
        $0 stop
        sleep 1
        $0 start
```

Notate: a seconda della versione di Debian che state usando potreste dover cambiare anche la riga `restart`. Questo problema, in Debian, è stato sistemato nella versione `1:8.3.1-2` di bind.

Tutto quello che dovete fare adesso è far ripartire bind con il comando `/etc/init.d/bind restart` e controllare il vostro syslog nelle due voci:

```
Sep  4 15:11:08 nexus named[13439]: group = named
Sep  4 15:11:08 nexus named[13439]: user = named
```

Voilà! Your named now *does not* run as root. If you want to read more information on why BIND does not run as non-root user on Debian systems, please check the Bug Tracking System regarding Bind, specifically http://bugs.debian.org/50013 and http://bugs.debian.org/132582, http://bugs.debian.org/53550, http://bugs.debian.org/52745, and http://bugs.debian.org/128129. Feel free to contribute to the bug reports if you think you can add useful information.

Eseguire il name server in chroot

To achieve maximum BIND security, now build a chroot jail (see sezione chiamata «Paranoie generiche riguardo chroot e suid») around your daemon. There is an easy way to do this: the `-t` option (see the named(8) manual page or page 100 of http://www.nominum.com/content/documents/bind9arm.pdf). This

will make Bind chroot itself into the given directory without you needing to set up a chroot jail and worry about dynamic libraries. The only files that need to be in the chroot jail are:

```
dev/null
etc/bind/       - should hold named.conf and all the server zones
sbin/named-xfer - if you do name transfers
var/run/named/  - should hold the PID and the name server cache (if
                  any) this directory needs to be writable by named user
var/log/named   - if you set up logging to a file, needs to be writable
                  for the named user
dev/log         - syslogd should be listening here if named is configured to
                  log through it
```

Per far sì che il demone Bind funzioni correttamente, necessita dei permessi giusti sui file named. Questo è un compito semplice, dal momento che i file di configurazione sono sempre in `/etc/named/`. Tenete presente che necessita solo dell'accesso in lettura ai file di zona, a meno che non si tratti di un name server secondario o di cache. Se questo è il caso, dovrete concedere i permessi di lettura-scrittura alle zone necessarie (in modo che i trasferimenti di zona dal server primario funzionino).

Also, you can find more information regarding Bind chrooting in the http://www.tldp.org/HOWTO/Chroot-BIND-HOWTO.html (regarding Bind 9) and http://www.tldp.org/HOWTO/Chroot-BIND8-HOWTO.html (regarding Bind 8). This same documents should be available through the installation of the doc-linux-text (text version) or doc-linux-html (HTML version). Another useful document is http://web.archive.org/web/20011024064030/http://www.psionic.com/papers/dns/dns-linux.

Nel caso in cui stiate creando una gabbia chroot per l'esecuzione di Bind in Debian (senza usare l'opzione `-t`), assicuratevi di aver incluso i seguenti file[5]:

```
dev/log - syslogd should be listening here
dev/null
etc/bind/named.conf
etc/localtime
etc/group - with only a single line: "named:x:GID:"
etc/ld.so.cache - generated with ldconfig
lib/ld-2.3.6.so
lib/libc-2.3.6.so
lib/ld-linux.so.2 - symlinked to ld-2.3.6.so
lib/libc.so.6 - symlinked to libc-2.3.6.so
sbin/ldconfig - may be deleted after setting up the chroot
sbin/named-xfer - if you do name transfers
var/run/
```

Modificate inoltre **syslogd**, in ascolto su `$CHROOT/dev/log` in modo che il name server possa scrivere gli eventi nel syslog del sistema locale di log.

Se volete eliminare i problemi con le librerie dinamiche, potete compilare bind staticamente. Potete usare **apt-get** a questo scopo, con l'opzione `source`. Potete anche scaricare il pacchetto che si compilerà in modo opportuno. Dovrete eseguire qualcosa di simile a questo:

```
$ apt-get source bind
# apt-get build-dep bind
```

[5] Questa configurazione non è ancora stata collaudata per il nuovo rilascio di Bind.

```
$ cd bind-8.2.5-2
  (edit src/port/linux/Makefile so CFLAGS includes the '-static'
   option)
$ dpkg-buildpackage -rfakeroot -uc -us
$ cd ..
# dpkg -i bind-8.2.5-2*deb
```

Dopo l'installazione, dovrete spostare i file all'interno della gabbia chroot[6]. Potete mantenere gli script `init.d` in `/etc/init.d`, così il sistema avvierà automaticamente il name server. Però dovrete modificarlo per aggiungere `--chroot /percorso_del_chroot` nella chiamata a **start-stop-daemon** presente in quegli script, oppure potete usare l'opzione *-t* di BIND impostando l'argomento OPTIONS nel file di configurazione `/etc/default/bind` (per la versione 8) o in `/etc/default/bind9` (per la versione 9).

Per maggiori informazioni su come configurare gabbie chroot vedete sezione chiamata «Paranoie generiche riguardo chroot e suid».

FIXME, integrare con informazioni da: http://people.debian.org/~pzn/howto/chroot-bind.sh.txt, http://www.cryptio.net/~ferlatte/config/ (specifico per Debian), http://web.archive.org/web/20021216104548/http://www.psionic.com/papers/whitep01.html e http://csrc.nist.gov/fasp/FASPDocs/NISTSecuringDNS.htm.

Proteggere Apache

FIXME:Aggiungere contenuto: moduli forniti con la normale installazione di Apache (in /usr/lib/apache/X.X/mod_*) e moduli che possono essere installati separatamente dai pacchetti libapache-mod-XXX.

Potete limitare l'accesso al server Apache se volete utilizzarlo solo internamente, impedendo che vi accedano estranei (per eseguire dei test, per accedere all'archivio doc-central, ecc.). A tale proposito, usate in `/etc/apache/http.conf` le direttive `Listen` o `BindAddress`.

Se usate Listen:

```
Listen 127.0.0.1:80
```

Se usate BindAddress:

```
BindAddress 127.0.0.1
```

Quindi, riavviate Apache con `/etc/init.d/apache restart` e vedrete che ascolterà solamente sull'interfaccia di loopback.

In ogni caso, se non utilizzate tutte le funzionalità di Apache, potreste considerare altri web server forniti da Debian, come dhttpd.

La http://httpd.apache.org/docs/misc/security_tips.html informa sulle misure di sicurezza che dovreste adottare per un server di rete Apache (Debian dà questa stessa informazione con il pacchetto apache-doc).

Maggiori informazioni su come limitare ulteriormente l'accesso ad Apache, impostando una gabbia di tipo chroot, le trovate in sezione chiamata «Chroot environment for Apache».

[6] A meno che non utiliziate l'opzione `instdir` quando chiamate **dpkg**, ma in questo caso la gabbia chroot potrebbe essere leggermente più complessa.

Impedire agli utenti la divulgazione di contenuti di rete

L'installazione predefinita di Apache consente agli utenti di pubblicare contenuti in `$HOME/public_html`. Tali contenuti possono essere raggiunti in remoto impiegando un'URL come: http://your_apache_server/~user.

Se volete impedirlo, occorre modificare il file di configurazione `/etc/apache/http.conf` commentando (in Apache 1.3) la riga:

```
LoadModule userdir_module /usr/lib/apache/1.3/mod_userdir.so
```

Nel caso usiate Apache 2.0 dovrete cancellare il file `/etc/apache2/mods-enabled/userdir.load` o rendere più restrittiva la configurazione predefinita di Apache modificando il file `/etc/apache2/mods-enabled/userdir.conf`.

Però, se il modulo fosse stato compilato staticamente (cosa che si può verificare lanciando `apache -l`), dovreste aggiungere la seguente riga al file di configurazione di Apache:

```
Userdir disabled
```

Un attaccante, però, può ancora eseguire il conteggio degli utenti, dal momento che la risposta del server di rete sarà un *403 Permission Denied* e non un *404 Not available*. Tuttavia questo comportamento può essere evitato utilizzando il modulo Rewrite.

Permessi sui file di log

I file di log di Apache, dalla versione 1.3.22-1, sono proprietà dell'utente 'root' e del gruppo 'adm', con permessi 640. Questi permessi vengono cambiati a rotazione. Senza un ampliamento dei privilegi, un intruso che abbia avuto accesso al sistema attraverso il server di rete non potrebbe eliminare voci vecchie dei file di log.

Pubblicare file web

I file di Apache si trovano in `/var/www`. Subito dopo l'installazione, il file predefinito fornisce alcune informazioni sul sistema (principalmente sul fatto che si tratta di un sistema Debian su cui è attivo Apache). Le pagine web predefinite vengono impostate per definizione come proprietà dell'utente root e del gruppo root, mentre il processo Apache viene eseguito come utente www-data e gruppo www-data. Questo dovrebbe rendere più difficile agli attaccanti, che hanno compromesso il sistema mediante Apache, di defacciare il sito web. È importante sostituire le pagine predefinite mostrate da Apache con pagine bianche o comunque personalizzate perché potrebbero fornire informazioni utili agli attaccanti.

Rendere sicuro finger

Se avete intenzione di fornire il servizio finger, chiedetevi prima se serva veramente. In caso affermativo, vi renderete conto che Debian fornisce molti demoni finger (questo è l'output di **apt-cache search fingerd**):

- cfingerd - Un demone finger configurabile.
- efingerd - Un altro demone finger per unix, consente di configurare accuratamente il vostro l'output.
- ffingerd - Un demone finger sicuro.

- fingerd - Server remoto di informazioni sugli utenti.
- xfingerd - Demone finger in stile BSD con supporto per qmail.

ffingerd è il servizio finger raccomandato se avete intenzione di usarlo come servizio pubblico. In ogni caso siete invitati, quando lo configurate tramite inetd, xinetd o tcpserver, a: limitare il numero di processi che verranno eseguiti contemporaneamente, limitare gli accessi a finger da specifici host (usando i tcp wrapper) e configurarlo in modo da farlo ascoltare solo sulle interfacce desiderate.

Paranoie generiche riguardo chroot e suid

chroot is one of the most powerful possibilities to restrict a daemon or a user or another service. Just imagine a jail around your target, which the target cannot escape from (normally, but there are still a lot of conditions that allow one to escape out of such a jail). You can eventually create a modified root environment for the user or service you do not trust. This can use quite a bit of disk space as you need to copy all needed executables, as well as libraries, into the jail. But then, even if the user does something malicious, the scope of the damage is limited to the jail.

Many services running as daemons could benefit from this sort of arrangement. The daemons that you install with your Debian distribution will not come, however, chrooted[7] per default.

Questi includono: server name (come **bind**), server web (come **apache**), server di posta (come **sendmail**) e server ftp (come **wu-ftpd**). Probabilmente è giusto dire che la complessità di BIND è il motivo per cui è stato esposto a molti attacchi in questi anni (vedete in sezione chiamata «Rendere sicuro BIND»).

Comunque, Debian fornisce del software che può aiutarvi a configurare ambienti **chroot** automaticamente. Vedete sezione chiamata «Creare ambienti chroot automaticamente».

Ad ogni modo se eseguite un qualsiasi servizio su un sistema, dovreste farlo nel modo più sicuro possibile. Questo comporta: revocare i privilegi di root, eseguirli in un ambiente sottoposto a restrizioni (come una gabbia creata con chroot) o sostituirli con servizi equivalenti più sicuri.

Tuttavia tenete conto fin d'ora che un ambiente creato con **chroot** può non essere sicuro se l'utente al suo interno è il superutente, per questo è necessario eseguire il servizio come utente non privilegiato. Limitando il suo ambiente limitate i file leggibili/eseguibili globalmente a cui il servizio può accedere e con questo le possibilità di una scalata ai privilegi sfruttando vulnerabilità relative alla sicurezza del sistema locale. Anche in questa situazione non si può essere completamente sicuri che non ci siano modi per una persona capace di uscire da questa gabbia in qualche modo. Usare solo programmi server che hanno la reputazione di essere sicuri è una buona misura precauzionale aggiuntiva. Anche buchi minuscoli come file di gestione aperti, possono esser usati da una persona capace per intromettersi nel sistema. Dopotutto **chroot** non è stato progettato come strumento per la sicurezza, ma come strumento di test.

Creare ambienti chroot automaticamente

Ci sono diversi programmi per ingabbiare automaticamente server e servizi in ambienti chroot. Al momento (accettato nel maggio 2002) Debian fornisce **chrootuid** di Wietse Venema nel pacchetto **chrootuid**, così come compartment e makejail. Potete usare questi programmi per configurare un ambiente con restrizioni ed eseguire qualunque programma (**chrootuid** permette addirittura di eseguirlo come un utente con restrizioni).

Alcuni di questi strumenti possono essere usati per configurare ambienti chroot in modo semplice. Il programma **makejail**, per esempio, può creare ed aggiornare una gabbia chroot con brevi file di configurazione (ne fornisce alcuni di esempio adatti per **bind**, **apache**, **postgresql** e **mysql**). Cerca di indovinare

[7] It does try to run them under *minimum priviledge* which includes running daemons with their own users instead of having them run as root.

ed installare dentro la gabbia, tutti i file richiesti dal servizio, usando **strace**, **stat** e le dipendenze dei pacchetti Debian. Ulteriori informazioni sono reperibili presso http://www.floc.net/makejail/. **Jailer** è un programma simile che può essere scaricato da http://www.balabit.hu/downloads/jailer/ ed è anche disponibile direttamente come pacchetto Debian.

In generale, paranoia per le password in chiaro

Siete invitati a non mantenere nessun servizio di rete che spedisca e riceva le password in chiaro usando FTP/Telnet/NIS/RPC. È raccomandato l'uso di ssh invece di telnet o di ftp.

Tenete a mente che una migrazione da telnet a ssh, che utilizzi comunque altri protocolli in chiaro NON aumenta la vostra sicurezza in alcun modo! La miglior cosa sarebbe quella di eliminare servizi quali ftp, telnet, pop, imap, http e sostituirli con i rispettivi servizi cifrati. Prendete in considerazione di migrare a servizi che hanno una versione SSL quali, ad esempio: ftp-ssl, telnet-ssl, pop-ssl, https ...

La maggior parte degli esempi riportati possono essere applicati a tutti i sistemi Unix (siete invitati a cercare altri documenti relativi a Linux e ad altri Unix).

Disabilitare NIS

Se vi è possibile, non usate NIS, il servizio di informazioni di rete, perché permette la condivisione delle password. Questo rende il sistema molto insicuro, se non è configurato in modo ottimale.

If you need password sharing between machines, you might want to consider using other alternatives. For example, you can setup an LDAP server and configure PAM on your system in order to contact the LDAP server for user authentication. You can find a detailed setup in the http://www.tldp.org/HOWTO/LDAP-HOWTO.html (`/usr/share/doc/HOWTO/en-txt/LDAP-HOWTO.txt.gz`).

You can read more about NIS security in the http://www.tldp.org/HOWTO/NIS-HOWTO.html (`/usr/share/doc/HOWTO/en-txt/NIS-HOWTO.txt.gz`).

FIXME (jfs): Aggiungere informazioni su come configurarlo in Debian

Rendere sicuri i servizi RPC

Disabilitate RPC se non ne fate uso.

RPC (Remote Procedure Call) è un protocollo che i programmi possono utilizzare per richiedere servizi da altri programmi che si trovano su computer diversi da quello locale. Il servizio **portmap** controlla i servizi RPC mediante la combinazione della numerazione assegnata ai programmi RPC con i numeri delle porte assegnate dal protocollo DARPA. RPC deve essere in esecuzione per eseguire chiamate RPC.

I servizi basati su RPC hanno avuto un pessimo record circa i buchi di sicurezza, anche se non portmapper stesso (nonostante fornisca informazioni ad un attaccante remoto). Notate che alcuni degli attacchi DDoS (Distributed Denial of service) utilizzano exploit RPC per accedere al sistema ed agire come un cosiddetto agente/gestore.

RPC vi occorre solamente se state utilizzando un servizio basato su RPC. I servizi più comuni basati su RPC sono NFS (Network File System) e NIS (Network Information System). Vedete la sezione precedente per maggiori informazioni riguardo NIS. Anche il FAM (File Alteration Monitor) fornito dal pacchetto fam è un servizio RPC, che dipende da portmap.

I servizi NFS sono abbastanza importanti in alcune reti. Se è il vostro caso, allora occorre trovare un bilanciamento tra la sicurezza e l'usabilità della vostra rete (potete leggere di più riguardo la sicurezza

di NFS nel http://www.tldp.org/HOWTO/NFS-HOWTO.html (`/usr/share/doc/HOWTO/en-txt/NFS-HOWTO.txt.gz`)).

Disabilitare completamente i servizi RPC

Disabilitare portmap è abbastanza semplice, ci sono diversi metodi. Il più semplice in un sistema Debian 3.0 e successivi rilasci è quello di disinstallare il pacchetto portmap. Se state eseguendo una versione più vecchia di Debian dovrete disabilitare il servizio come avete visto in sezione chiamata «Disabilitare i servizi attivi in modalità demone», perché il programma è parte del pacchetto netbase (che non può essere disinstallato senza danneggiare il sistema).

Notate che alcuni ambienti desktop, in particolare GNOME, utilizzano i servizi RPC ed hanno bisogno di portmap per alcune delle loro funzionalità di gestione file. Se è il vostro caso, potete limitare l'accesso ai servizi RPC come descritto sotto.

Limitare l'accesso ai servizi RPC

Sfortunatamente, in alcuni casi rimuovere i servizi RPC dal sistema non rappresenta una soluzione. Alcuni servizi locali in ambito desktop (come fam di SGI) sono basati su RPC e necessitano di un portmapper locale. Questo significa che in alcune situazioni, utenti che installano un ambiente desktop (ad esempio GNOME) installerano anche portmapper.

Ci sono diversi modi per limitare l'accesso a portmapper ed ai servizi RPC:

- Bloccare l'accesso alle porte utilizzate da questi servizi con un firewall locale (consultate sezione chiamata «Aggiungere funzionalità al firewall»).
- Bloccare l'accesso a questi servizi utilizzando dei wrapper TCP, da quando portmapper (ed alcuni servizi RPC) vengono compilati con `libwrap` (consultate sezione chiamata «Usare i tcpwrapper»). Questo significa che è possibile bloccare l'accesso ai servizi tramite la configurazione dei wrapper tcp `hosts.allow` e `hosts.deny`.
- Dalla versione 5-5, il pacchetto portmap può essere configurato per rimanere in ascolto solamente sull'interfaccia di loopback. Per fare questo, occorre modificare `/etc/default/portmap`, decommentando la riga seguente: `#OPTIONS="-i 127.0.0.1"` e fare un restart del portmapper. È sufficiente questo per permettere ai servizi locali RPC di essere in esecuzione ed allo stesso tempo impedire l'accesso remoto al sistema (consultate, comunque, sezione chiamata «Disabilitare la questione weak-end host»).

Aggiungere funzionalità al firewall

The Debian GNU/Linux operating system has the built-in capabilities provided by the Linux kernel. If you install a recent Debian release (default kernel installed is 2.6) you will have **iptables** (netfilter) firewalling available[8].

Proteggere il sistema locale con un firewall

Potete usare le regole di un firewall per rendere più sicuro l'accesso al sistema ed anche per limitare le comunicazioni in uscita. Le regole del firewall possono anche essere utilizzate per proteggere i processi che non possono essere adeguatamente configurati per *non* fornire servizi ad alcune reti, indirizzi IP, etc..

[8] Available since the kernel version 2.4 (which was the default kernel in Debian 3.0). Previous kernel versions (2.2, available in even older Debian releases) used **ipchains**. The main difference between **ipchains** and **iptables** is that the latter is based on *stateful packet inspection* which provides for more secure (and easier to build) filtering configurations. Older (and now unsupported) Debian distributions using the 2.0 kernel series needed the appropriate kernel patch.

Comunque, questo passaggio viene presentato per ultimo in questo manuale essenzialmente perché è *molto* meglio non dipendere unicamente dalle capacità di un firewall per proteggere un determinato sistema. La sicurezza in un sistema è data da livelli, in cui il firewall è l'ultimo da includere, una volta che tutti i processi sono stati adeguatamente irrobustiti. Potete facilmente immaginare un'installazione in cui un sistema è protetto solo dal firewall magari incorporato e un amministratore che incautamente rimuove le regole del firewall per qualche motivo (problemi con l'installazione, fastidio, errore umano...), questo sistema sarebbe parecchio esposto ad un attacco se non ci fosse nessun'altra protezione a garantirlo.

D'altra parte, impostare le regole di un firewall sul sistema può anche prevenire il verificarsi di alcuni fatti spiacevoli. Anche se i servizi forniti vengono configurati in modo sicuro, un firewall può proteggere da errate configurazioni o da servizi appena installati che non sono ancora stati configurati. Inoltre, una configurazione sicura previene il funzionamento di trojans *che chiamano casa*, finché il codice del firewall non viene rimosso. Notate che un intruso *non* necessita di un accesso superutente per installare un trojan locale che possa essere controllato da remoto (dato che il binding sulle porte è consentito se non si tratta di porte privilegiate e se queste funzionalità non sono state rimosse).

Quindi, il firewall, se impostato correttamente, dovrebbe avere una politica di negazione predefinita, secondo cui:

- Le connessioni in ingresso vengono consentite solo ai servizi locali da macchine autorizzate.
- outgoing connections are only allowed to services used by your system (DNS, web browsing, POP, email...).[9]
- La regola di inoltro nega qualsiasi cosa (a meno che non stiate proteggendo altri sistemi, vedete più in basso).
- Tutte le altre connessioni in ingresso o in uscita vengono negate.

Utilizzare un firewall per proteggere altri sistemi

Un firewall Debian può anche essere installato per proteggere, con regole di filtraggio, accessi a sistemi posti *dietro* ad esso, limitando la loro esposizione su Internet. Il firewall può essere configurato per proteggere i sistemi dall'esterno verso la rete locale per l'accesso a determinati servizi (porte) che non sono pubblici. Per esempio, su un server di posta, solo la porta 25 (che fornisce il servizio di posta) ha bisogno di essere accessibile dall'esterno. Un firewall può essere configurato, anche se ci sono altri servizi di rete oltre a quelli pubblici, per respingere pacchetti (questo è conosciuto come *filtraggio*) diretti verso di sé.

Potete anche configurare un sistema Debian GNU/Linux come bridge firewall, ad esempio un firewall filtrante privo di indirizzo IP, completamente trasparente alla rete e che quindi non può essere attaccato direttamente. A seconda del kernel che avete installato potreste aver bisogno di installare la patch per il bridge firewall; andate quindi su *802.1d Ethernet Bridging* durante la configurazione del kernel e abilitate la nuova opzione *netfilter (firewalling) support*. Leggete sezione chiamata «Impostare un bridge firewall» per ulteriori informazioni su come meglio sfruttare questa funzionalità sul vostro sistema Debian GNU/Linux.

Configurare il firewall

L'installazione Debian predefinita, contrariamente ad altre distribuzioni Linux, non fornisce ancora all'amministratore un modo per impostare una configurazione per il firewall durante l'installazione, ma è possibile installare alcuni pacchetti per configurare il firewall (vedete sezione chiamata «Uso dei pacchetti firewall»).

[9] Unlike personal firewalls in other operating systems, Debian GNU/Linux does not (yet) provide firewall generation interfaces that can make rules limiting them per process or user. However, the iptables code can be configured to do this (see the owner module in the iptables(8) manual page).

Of course, the configuration of the firewall is always system and network dependant. An administrator must know beforehand what is the network layout and the systems to protect, the services that need to be accessed, and whether or not other network considerations (like NAT or routing) need to be taken into account. Be careful when configuring your firewall, as Laurence J. Lane says in the iptables package:

The tools can easily be misused, causing enormous amounts of grief by completely crippling network access to a system. It is not terribly uncommon for a remote system administrator to accidentally get locked out of a system hundreds or thousands of miles away. You can even manage to get locked out of a computer who's keyboard is under your own fingers. Please, use due caution.

Ricordate: la semplice installazione di iptables (o di altri programmi per gestire firewall) non dà nessuna protezione, fornisce solamente il software. Per poter avere un firewall è necessario *configurarlo*!

Se non avete idea di come impostare delle regole per un firewall, è opportuno che consultiate il *Packet Filtering HOWTO* ed il *NAT HOWTO*, forniti da iptables per la lettura non in linea in `/usr/share/doc/iptables/html/`.

If you do not know much about firewalling you should start by reading the http://www.tldp.org/HOWTO/Firewall-HOWTO.html, install the doc-linux-text package if you want to read it offline. If you want to ask questions or need help setting up a firewall you can use the debian-firewall mailing list, see http://lists.debian.org/debian-firewall. Also see sezione chiamata «Conoscenze preliminari» for more (general) pointers on firewalls. Another good iptables tutorial is http://iptables-tutorial.frozentux.net/iptables-tutorial.html.

Uso dei pacchetti firewall

L'impostazione manuale di un firewall può essere complicata per gli amministratori novizi (e talvolta anche per quelli esperti). Comunque, la comunità del free software ha creato un buon numero di strumenti che possono essere usati per configurare facilmente un firewall locale. Sappiate però che alcuni di questi strumenti sono più orientati verso una protezione esclusivamente locale (anche conosciuta come *personal firewall*) mentre alcuni sono molto più versatili e possono essere usati per configurare regole complesse per proteggere intere reti.

Alcuni software che possono essere usati per impostare regole di firewall in sistemi Debian sono:

- Per sistemi desktop:
  - firestarter, un'applicazione GNOME orientata all'utente finale che include un'utile wizard per impostare velocemente le regole del firewall. L'applicazione comprende un'interfaccia grafica capace di controllare quando una regola del firewall blocca il traffico.
  - guarddog, un pacchetto di configurazione basato su KDE e rivolto sia ad utenti principianti che avanzati.
  - knetfilter, un'interfaccia grafica per KDE in grado di gestire le regole firewall e NAT di iptables (è un'alternativa, o concorrente che dir si voglia, del programma guarddog, anche se tende a soddisfare maggiormente le necessità di un utente esperto).
  - fireflier, un programma interattivo in grado di creare regole per iptables in base al traffico rilevato sul sistema e alle applicazioni presenti sul sistema stesso. È composto da un server e da un client, che vanno installati entrambi; oltre al server (fireflier-server) bisogna quindi anche scegliere uno dei vari client disponibili, ve ne sono per vari ambienti grafici: fireflier-client-gtk (basato sulle Gtk+), fireflier-client-kde (per ambiente KDE) e fireflier-client-qt (basato sulle QT).
- Per sistemi ad uso server (senza schermo):

- fwbuilder, è un interfaccia grafica, scritta usando la programmazione orientata agli oggetti, che include dei generatori di regole per vari firewall, tra cui netfilter per Linux, pf (usato da OpenBSD, NetBSD, FreeBSD e MacOS X) ed un generatore di liste di accesso per i router. Le sue funzionalità sono paragonabili ai software per la gestione dei firewall comunemente usati dalle aziende. Include anche un'interfaccia a riga di comando che permette di sfruttare tutte le funzionalità del programma.

- shorewall, un tool di configurazione per i firewall che fornisce supporto per IPsec così come un supporto limitato per il traffic shaping nonché per le definizioni delle regole di un firewall. La configurazione viene effettuata mediante un semplice insieme di file che vengono usati per generare le regole di iptables.

- bastille, questa applicazione di irrobustimento viene descritta in Capitolo 6, *Irrobustimento automatico di un sistema Debian*. Uno dei passi di irrobustimento configurabile dall'amministratore è la definizione del traffico di rete ammesso e vietato, che viene usato per generare un insieme di regole del firewall che il sistema eseguirà all'avvio.

In Debian vengono forniti diversi frontend a iptables; un elenco esaustivo che raffronta i differenti pacchetti software presenti viene mantenuta nella pagina http://wiki.debian.org/Firewalls del wiki di Debian.

Da notare che alcuni dei pacchetti presentati precedentemente, installeranno degli script di firewalling che verranno eseguiti all'avvio del sistema. Testate gli script in maniera approfondita prima di riavviare, altrimenti potreste rimanere chiusi fuori dal sistema. Di solito, mescolando diversi pacchetti di firewalling, si possono ottenere effetti indesiderati, tipo che l'ultimo script ad essere eseguito sarà quello che configura il sistema (comportamento che potrebbe non essere quello desiderato). Consultate la documentazione del pacchetto e usate solo uno di questi setup.

Come già detto, alcuni programmi, come firestarter, guarddog e knetfilter sono interfacce grafiche di amministrazione che usano GNOME o (le ultime due) KDE. Queste applicazioni sono rivolte in particolare all'utente (ossia all'utente comune), a differenza di alcuni degli altri pacchetti dell'elenco, che invece potrebbero essere pensati più per gli amministratori. Alcuni dei programmi già menzionati (come **bastille**) sono specializzati nell'impostare le regole del firewall che protegge l'host su cui girano ma non necessariamente per impostare regole di firewall per host che proteggono una rete (come invece fanno **shorewall** o **fwbuilder**).

Esiste poi un altro tipo di applicazione per firewall, ossia gli application proxy. Se cercate di configurare un firewall aziendale che faccia packet filtering e fornisca un certo numero di transparent proxy che analizzino il traffico in modo dettagliato, potete considerare l'utilizzo di zorp, che fa tutto questo in un unico programma. Potete anche impostare manualmente questo tipo di host per il firewall usando i proxy disponibili in Debian per differenti servizi; per il DNS usando bind (adeguatamente configurato), dnsmasq, pdnsd o totd, per l'FTP usando frox o ftp-proxy, per X11 usando xfwp, per IMAP usando imapproxy, per la mail usando smtpd, o per il POP3 usando p3scan. Per altri protocolli potete usare un proxy TCP generico come simpleproxy o un proxy SOCKS generico come dante-server, tsocks o socks4-server. Nella maggior parte dei casi, bisognerà anche usare un sistema di web caching (come squid) e di filtraggio web (come squidguard o dansguardian).

Configurazione manuale di init.d

Un'altra possibilità è quella di configurare manualmente le regole del firewall attraverso uno script in init.d che lanci tutti i comandi **iptables**. È sufficiente seguire la seguente procedura:

- Revisionare lo script che segue ed adattarlo alle proprie esigenze.

- Collaudare lo script ed esaminare i log di sistema per vedere il tipo di traffico che viene eliminato. Se lo state collaudando in remoto, vorrete lanciare la snippet shell di esempio per rimuovere il firewall (se

non digitate nulla per 20 secondi), oppure potreste voler decommentare le definizioni della procedura di *default deny* (*-P INPUT DROP* e *-P OUTPUT DROP*) e verificare poi che il sistema non elimini alcun traffico legittimo.

- Spostare lo script in `/etc/init.d/myfirewall`.
- The below script takes advantage of Debian's use (since Squeeze) of dependency based boot sequencing. For more information see: https://wiki.debian.org/LSBInitScripts/DependencyBasedBoot and https://wiki.debian.org/LSBInitScripts. With the LSB headers set as they are in the script, insserv will automatically configure the system to start the firewall before any network is brought up, and stop the firewall after any network is brought down.

```
# insserv myfirewall
```

Questo e' un esempio di uno script per un firewall:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          myfirewall
# Required-Start:    $local_fs
# Required-Stop:     $local_fs
# Default-Start:     S
# Default-Stop:      0 6
# X-Start-Before:    $network
# X-Stop-After:      $network
# Short-Description: My custom firewall.
### END INIT INFO
#
# Simple example firewall configuration.
#
# Caveats:
# - This configuration applies to all network interfaces
#   if you want to restrict this to only a given interface use
#   '-i INTERFACE' in the iptables calls.
# - Remote access for TCP/UDP services is granted to any host,
#   you probably will want to restrict this using '--source'.
#
# chkconfig: 2345 9 91
# description: Activates/Deactivates the firewall at boot time
#
# You can test this script before applying with the following shell
# snippet, if you do not type anything in 10 seconds the firewall
# rules will be cleared.
#------------------------------------------------------------------
#  while true; do test=""; read  -t 20 -p "OK? " test ; \
#  [ -z "$test" ] && /etc/init.d/myfirewall clear ; done
#------------------------------------------------------------------

PATH=/bin:/sbin:/usr/bin:/usr/sbin

# Services that the system will offer to the network
TCP_SERVICES="22" # SSH only
UDP_SERVICES=""
```

```
# Services the system will use from the network
REMOTE_TCP_SERVICES="80" # web browsing
REMOTE_UDP_SERVICES="53" # DNS
# Network that will be used for remote mgmt
# (if undefined, no rules will be setup)
# NETWORK_MGMT=192.168.0.0/24
# If you want to setup a management network (i.e. you've uncommented
# the above line) you will need to define the SSH port as well (i.e.
# uncomment the below line.) Remember to remove the SSH port from the
# TCP_SERVICES string.
# SSH_PORT="22"

if ! [ -x /sbin/iptables ]; then
    exit 0
fi

fw_start () {

  # Input traffic:
  /sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
  # Services
  if [ -n "$TCP_SERVICES" ] ; then
  for PORT in $TCP_SERVICES; do
    /sbin/iptables -A INPUT -p tcp --dport ${PORT} -j ACCEPT
  done
  fi
  if [ -n "$UDP_SERVICES" ] ; then
  for PORT in $UDP_SERVICES; do
    /sbin/iptables -A INPUT -p udp --dport ${PORT} -j ACCEPT
  done
  fi
  # Remote management
  if [ -n "$NETWORK_MGMT" ] ; then
    /sbin/iptables -A INPUT -p tcp --src ${NETWORK_MGMT} --dport ${SSH_PORT} -j AC
  fi
  # Remote testing
  /sbin/iptables -A INPUT -p icmp -j ACCEPT
  /sbin/iptables -A INPUT -i lo -j ACCEPT
  /sbin/iptables -P INPUT DROP
  /sbin/iptables -A INPUT -j LOG

  # Output:
  /sbin/iptables -A OUTPUT -j ACCEPT -o lo
  /sbin/iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
  # ICMP is permitted:
  /sbin/iptables -A OUTPUT -p icmp -j ACCEPT
  # So are security package updates:
  # Note: You can hardcode the IP address here to prevent DNS spoofing
  # and to setup the rules even if DNS does not work but then you
  # will not "see" IP changes for this service:
  /sbin/iptables -A OUTPUT -p tcp -d security.debian.org --dport 80 -j ACCEPT
  # As well as the services we have defined:
  if [ -n "$REMOTE_TCP_SERVICES" ] ; then
  for PORT in $REMOTE_TCP_SERVICES; do
```

```
    /sbin/iptables -A OUTPUT -p tcp --dport ${PORT} -j ACCEPT
  done
  fi
  if [ -n "$REMOTE_UDP_SERVICES" ] ; then
  for PORT in $REMOTE_UDP_SERVICES; do
    /sbin/iptables -A OUTPUT -p udp --dport ${PORT} -j ACCEPT
  done
  fi
  # All other connections are registered in syslog
  /sbin/iptables -A OUTPUT -j LOG
  /sbin/iptables -A OUTPUT -j REJECT
  /sbin/iptables -P OUTPUT DROP
  # Other network protections
  # (some will only work with some kernel versions)
  echo 1 > /proc/sys/net/ipv4/tcp_syncookies
  echo 0 > /proc/sys/net/ipv4/ip_forward
  echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
  echo 1 > /proc/sys/net/ipv4/conf/all/log_martians
  echo 1 > /proc/sys/net/ipv4/ip_always_defrag
  echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
  echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
  echo 0 > /proc/sys/net/ipv4/conf/all/send_redirects
  echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route

}

fw_stop () {
  /sbin/iptables -F
  /sbin/iptables -t nat -F
  /sbin/iptables -t mangle -F
  /sbin/iptables -P INPUT DROP
  /sbin/iptables -P FORWARD DROP
  /sbin/iptables -P OUTPUT ACCEPT
}

fw_clear () {
  /sbin/iptables -F
  /sbin/iptables -t nat -F
  /sbin/iptables -t mangle -F
  /sbin/iptables -P INPUT ACCEPT
  /sbin/iptables -P FORWARD ACCEPT
  /sbin/iptables -P OUTPUT ACCEPT
}


case "$1" in
  start|restart)
    echo -n "Starting firewall.."
    fw_stop
    fw_start
    echo "done."
    ;;
  stop)
    echo -n "Stopping firewall.."
```

```
    fw_stop
    echo "done."
    ;;
  clear)
    echo -n "Clearing firewall rules.."
    fw_clear
    echo "done."
    ;;
  *)
    echo "Usage: $0 {start|stop|restart|clear}"
    exit 1
    ;;
  esac
exit 0
```

Invece di includere tutte le regole di iptables negli script in init.d, potete usare il programma **iptables-restore** per ripristinare le regole salvate usando **iptables-save**. Per fare questo dovrete definire le vostre regole e salvare il file delle regole creato in una posizione statica (come ad esempio in `/etc/default/firewall`).

Configurare le regole del firewall tramite ifup

Potete utilizzare anche la configurazione di rete in `/etc/network/interfaces` per determinare le regole del firewall. Per fare questo:

- Create il vostro insieme di regole del firewall per quando l'interfaccia è attiva.
- Salvate l'insieme di regole con **iptables-save** in un file in `/etc`, ad esempio `/etc/iptables.up.rules`
- Configurate `/etc/network/interfaces` ed usate l'insieme delle regole appena impostate:

```
iface eth0 inet static
        address x.x.x.x
        [.. interface configuration ..]
        pre-up iptables-restore < /etc/iptables.up.rules
```

In aggiunta potete anche configurare un insieme di regole da applicare per quando l'interfaccia di rete è *down*, mediante la creazione del suddetto insieme di regole, salvandolo in `/etc/iptables.down.rules` e aggiungendo questa direttiva alla configurazione dell'interfaccia:

```
    post-down iptables-restore < /etc/iptables.down.rules
```

For more advanced firewall configuration scripts through ifupdown you can use the hooks available to each interface as in the `*.d/` directories called with **run-parts** (see run-parts(8) manual page).

Collaudare la configurazione del firewall

Collaudare la configurazione del firewall è molto facile, e pericoloso, infatti basta eseguire lo script del firewall (o abilitare la configurazione definita nella propria applicazione per la configurazione del firewall). Comunque, se non siete abbastanza cauti e state configurando il vostro firewall da remoto (ad esempio mediante una connessione SSH), potreste trovarvi tagliati fuori.

Ci sono parecchi modi per impedire che questo avvenga. Uno di questi è eseguire uno script in un terminale separato che rimuoverà la configurazione del firewall se non viene recepito un input. Un esempio di questo è:

```
$  while true; do test=""; read  -t 20 -p "OK? " test ; \
  [ -z "$test" ] && /etc/init.d/firewall clear ; done
```

Un'altra strada è quella di introdurre una backdoor nel sistema mediante un meccanismo alternativo che permetta di disattivare il firewall o bucarlo nel caso in cui qualcosa andasse storto. Per fare questo potete usare knockd e configurarlo in modo che una determinata connessione ad una porta disattivi il firewall (o aggiunga una regola temporanea). Anche se i pacchetti verranno rifiutati dal firewall, finché knockd impegna l'interfaccia e *vede*, potrete aggirare il problema.

Collaudare un firewall che sta proteggendo una rete interna è una questione differente, si può voler conoscere alcuni dei tool usati nella valutazione delle vulnerabilità remote (vedete in sezione chiamata «Strumenti per la valutazione delle vulnerabilità da remoto») per sondare la rete dall'esterno all'interno (o da qualunque altra direzione) per verificare l'efficacia della configurazione del firewall.

Capitolo 6. Irrobustimento automatico di un sistema Debian

Dopo aver letto tutte le informazioni dei capitoli precedenti potreste chiedervi "Devo fare molte cose per rendere sicuro il mio sistema, non potrebbero essere automatizzate?". La risposta è sì, ma fate attenzione agli strumenti automatizzati. Alcune persone credono che uno strumento per l'irrobustimento non elimini il bisogno di una buona amministrazione. Perciò non pensate stupidamente che si possa automatizzare l'intero processo e risolvere i problemi correlati. La sicurezza è un processo continuo al quale l'amministratore deve partecipare e al quale non può solo assistere e lasciare che gli strumenti svolgano il loro lavoro. Non esiste un solo strumento che possa risolvere i problemi per tutte le possibili implementazioni delle politiche di sicurezza, tutti i tipi di attacchi e tutti gli ambienti.

Since woody (Debian 3.0) there are two specific packages that are useful for security hardening. The harden package which takes an approach based on the package dependencies to quickly install valuable security packages and remove those with flaws, configuration of the packages must be done by the administrator. The bastille package that implements a given security policy on the local system based on previous configuration by the administrator (the building of the configuration can be a guided process done with simple yes/no questions).

Harden

The harden package tries to make it more easy to install and administer hosts that need good security. This package should be used by people that want some quick help to enhance the security of the system. It automatically installs some tools that should enhance security in some way: intrusion detection tools, security analysis tools, etc. Harden installs the following *virtual* packages (i.e. no contents, just dependencies or recommendations on others):

- harden-tools: tools to enhance system security (integrity checkers, intrusion detection, kernel patches...)
- harden-environment: helps configure a hardened environment (currently empty).
- harden-servers: removes servers considered insecure for some reason.
- harden-clients: removes clients considered insecure for some reason.
- harden-remoteaudit: tools to remotely audit a system.
- harden-nids: helps to install a network intrusion detection system.
- harden-surveillance: helps to install tools for monitoring of networks and services.

Package utili che non costituiscono dipendenze:

- harden-doc: provides this same manual and other security-related documentation packages.
- harden-development: development tools for creating more secure programs.

Be careful because if you have software you need (and which you do not wish to uninstall for some reason) and it conflicts with some of the packages above you might not be able to fully use harden. The harden packages do not (directly) do a thing. They do have, however, intentional package conflicts with known non-secure packages. This way, the Debian packaging system will not approve the installation of these packages. For example, when you try to install a telnet daemon with harden-servers, apt will say:

```
# apt-get install telnetd
The following packages will be REMOVED:
  harden-servers
The following NEW packages will be installed:
  telnetd
Do you want to continue? [Y/n]
```

Questo dovrebbe far nascere qualche dubbio nella testa dell'amministratore che, a questo punto, dovrebbe riconsiderare questa azione.

Bastille Linux

http://bastille-linux.sourceforge.net/ is an automatic hardening tool originally oriented towards the Red Hat and Mandrake Linux distributions. However, the bastille package provided in Debian (since woody) is patched in order to provide the same functionality for Debian GNU/Linux systems.

Bastille può essere utilizzato con diversi frontend (sono tutti documentati nella rispettiva pagina man inclusa nel pacchetto Debian) e questo permette all'amministratore di:

- Answer questions step by step regarding the desired security of your system (using InteractuveBastille(8)
- Use a default setting for security (amongst three: Lax, Moderate or Paranoia) in a given setup (server or workstation) and let Bastille decide which security policy to implement (using BastilleChooser(8)).
- Take a predefined configuration file (could be provided by Bastille or made by the administrator) and implement a given security policy (using AutomatedBastille(8)).

Capitolo 7. Infrastrutture per la sicurezza in Debian

Il Team Debian per la Sicurezza

Debian has a Security Team, that handles security in the *stable* distribution. Handling security means they keep track of vulnerabilities that arise in software (watching forums such as Bugtraq, or vuln-dev) and determine if the *stable* distribution is affected by it.

Also, the Debian Security Team is the contact point for problems that are coordinated by upstream developers or organizations such as http://www.cert.org which might affect multiple vendors. That is, when problems are not Debian-specific. The contact point of the Security Team is mailto:team@security.debian.org which only the members of the security team read.

Sensitive information should be sent to the first address and, in some cases, should be encrypted with the Debian Security Contact key (as found in the Debian keyring).

Once a probable problem is received by the Security Team it will investigate if the *stable* distribution is affected and if it is, a fix is made for the source code base. This fix will sometimes include backporting the patch made upstream (which usually is some versions ahead of the one distributed by Debian). After testing of the fix is done, new packages are prepared and published in the http://security.debian.org site so they can be retrieved through **apt** (see sezione chiamata «Eseguire un aggiornamento per la sicurezza»). At the same time a *Debian Security Advisory* (DSA) is published on the web site and sent to public mailing lists including http://lists.debian.org/debian-security-announce and Bugtraq.

Potete trovare altre "risposte a domande frequenti" in sezione chiamata «Domande sul Team per la sicurezza di Debian».

Avvisi di sicurezza Debian

Debian Security Advisories (DSAs) are made whenever a security vulnerability is discovered that affects a Debian package. These advisories, signed by one of the Security Team members, include information of the versions affected as well as the location of the updates. This information is:

- Numero di versione della correzione.
- Tipo di problema.
- Se la vulnerabilità è sfruttabile da remoto o solo localmente.
- Breve descrizione del pacchetto.
- Descrizione del problema.
- Descrizione dell'exploit.
- Descrizione della correzione.

DSAs are published both on http://www.debian.org/ and in the http://www.debian.org/security/. Usually this does not happen until the website is rebuilt (every four hours) so they might not be present immediately. The preferred channel is the debian-security-announce mailing list.

Gli utenti interessati possono comunque (e questo viene fatto in alcuni portali relativi a Debian) utilizzare il canale RDF per scaricare automaticamente gli avvisi di sicurezza sul loro desktop. Alcune applicazioni,

come **Evolution** (un client e-mail ed assistente per informazioni personali) e **Multiticker** (un'applet di GNOME), possono essere usate per reperire gli avvisi automaticamente. Il canale RDF è disponibile su http://www.debian.org/security/dsa.rdf.

I DSA pubblicati sul sito potrebbero essere aggiornati dopo essere stati spediti alle mailing-list pubbliche. Un aggiornamento tipico è aggiungere riferimenti incrociati ai database delle vulnerabilità di sicurezza. Inoltre, le traduzioni[1] dei DSA non vengono spedite alle mailing list di sicurezza ma incluse direttamente nel sito.

Riferimenti incrociati sulle vulnerabilità

Debian fornisce una http://www.debian.org/security/crossreferences che include tutti i riferimenti disponibili per tutti gli avvisi pubblicati dal 1998. Questa tabella viene fornita per completare la http://cve.mitre.org/cve/refs/refmap/source-DEBIAN.html.

You will notice that this table provides references to security databases such as http://www.securityfocus.com/bid, http://www.cert.org/advisories/ and http://www.kb.cert.org/vuls as well as CVE names (see below). These references are provided for convenience use, but only CVE references are periodically reviewed and included.

Advantages of adding cross references to these vulnerability databases are:

- Rende agli utenti Debian più facile vedere e tenere traccia di quali avvisi pubblicati sono già stati coperti da Debian.
- Gli amministratori di sistema possono imparare di più sulle vulnerabilità ed il loro impatto seguendo i riferimenti incrociati.
- Queste informazioni possono essere usate per fare controlli incrociati con l'output degli scanner di vulnerabilità che includono riferimenti a CVE per rimuovere falsi allarmi (vedete in sezione chiamata «Sa scansione per la ricerca delle vulnerabilità risulta che il mio sistema Debian è vulnerabile!»).

Compatibilità con CVE

Debian Security Advisories were http://www.debian.org/security/CVE-certificate.jpg[2] in February 24, 2004.

Debian developers understand the need to provide accurate and up to date information of the security status of the Debian distribution, allowing users to manage the risk associated with new security vulnerabilities. CVE enables us to provide standardized references that allow users to develop a https://cve.mitre.org/compatible/enterprise.html.

Il progetto http://cve.mitre.org viene mantenuto dalla MITRE Corporation e fornisce un elenco standardizzato per le vulnerabilità ed i problemi di sicurezza.

Debian crede che fornire agli utenti informazioni aggiuntive relative alle questioni di sicurezza della distribuzione Debian sia estremamente importante. L'inclusione dei nomi CVE negli avvisi aiuta gli utenti ad associare delle generiche vulnerabilità con gli aggiornamenti Debian, il che riduce il tempo necessario per gestire le vulnerabilità che affliggono i nostri utenti. Inoltre, facilita la gestione della sicurezza in un ambiente dove strumenti di sicurezza con CVE abilitato - come sistemi di rilevamento d'intrusione su rete o su di un host, o strumenti per la valutazione di vulnerabilità - sono già in uso indipendentemente dal fatto che siano basati sulla distribuzione Debian.

[1] Le traduzioni sono disponibili in una decina di lingue diverse.
[2] The full http://cve.mitre.org/compatible/phase2/SPI_Debian.html is available at CVE

Debian provides CVE names for all DSAs released since September 1998. All of the advisories can be retrieved on the Debian web site, and announcements related to new vulnerabilities include CVE names if available at the time of their release. Advisories associated with a given CVE name can be searched directly through the Debian Security Tracker (see below).

In alcuni casi potreste non trovare un dato nome CVE tra gli avvisi pubblicati, per esempio perché:

- Non ci sono prodotti Debian affetti da quella specifica vulnerabilità.
- Non c'è ancora un avviso che copre quella vulnerabilità (il problema di sicurezza potrebbe essere stato riportato come http://bugs.debian.org/cgi-bin/pkgreport.cgi?tag=security ma non è ancora stata caricata o collaudata un'apposita patch).
- Un avviso è stato pubblicato prima che il nome CVE sia stato assegnato ad una data vulnerabilità (cercate un aggiornamento sul sito web).

Security Tracker

The central database of what the Debian security teams know about vulnerabilities is the http://security-tracker.debian.org. It cross references packages, vulnerable and fixed versions for different suites, CVE names, Debian bug numbers, DSA's and miscellaneous notes. It can be searched, e.g. by CVE name to see which Debian packages are affected or fixed, or by package to show unresolved security issues. The only information missing from the tracker is confidential information that the security team received under embargo.

The package **debsecan** uses the information in the tracker to report to the administrator of a system which of the installed packages are vulnerable, and for which updates are available to fix security issues.

La costruzione dell'infrastruttura di sicurezza in Debian

Poiché Debian supporta attualmente un gran numero di architetture, gli amministratori a volte si sorprendono se una determinata architettura impiega più tempo per ricevere gli aggiornamenti per la sicurezza rispetto ad un'altra. È un dato di fatto, tranne in rare circostanze, che gli aggiornamenti siano disponibili per tutte le architetture allo stesso tempo.

Packages in the security archive are autobuilt, just like the regular archive. However, security updates are a little more different than normal uploads sent by package maintainers since, in some cases, before being published they need to wait until they can be tested further, an advisory written, or need to wait for a week or more to avoid publicizing the flaw until all vendors have had a reasonable chance to fix it.

Thus, the security upload archive works with the following procedure:

- Qualcuno scopre un problema di sicurezza.
- Qualcuno risolve il problema e comincia un aggiornamento su security-master.debian.org (questo *qualcuno* è solitamente un membro del Security Team ma può anche essere un manutentore di un pacchetto con una correzione appropriata che ha contattato il Security Team in precedenza). Il Changelog include un *testing-security* o *stable-security* come distribuzione finale.
- L'aggiornamento viene controllato e verificato dal sistema Debian e spostato in coda/accettato e quindi notificato al sistema buildd. Ai file che si trovano qui accede il Security Team e (a volte indirettamente) il sistema buildd.

- Il sistema buildd, progettato tenendo anche conto dei problemi di sicurezza, prende il pacchetto sorgente (con una priorità maggiore delle normali costruzioni di pacchetti), lo costruisce ed invia i log al Security Team.
- Il Security Team risponde ai log e i nuovi pacchetti costruiti vengono caricati in coda/non_verificata, dove vengono controllati da un sistema Debian e spostati in in_coda/accettati.
- Quando il Security Team trova il pacchetto sorgente accettabile (ovvero, che è stato correttamente costruito per tutte le architetture supportate e che corregge i buchi di sicurezza senza introdurne di nuovi) lancia uno script che:
  - Installa il pacchetto nell'archivio di sicurezza.
  - Aggiorna i file `Packages`, `Sources` e `Release` di security.debian.org nel solito modo (**dpkg-scanpackages**, **dpkg-scansources** ...).
  - Imposta un template dell'avviso che il Security Team ha finito il suo compito.
  - forwards the packages to the appropriate proposed-updates so that it can be included in the real archive as soon as possible.

Questa procedura, precedentemente fatta a mano, è stata collaudata ed immessa nello stadio di congelamento di Debian 3.0 woody (luglio 2002). Grazie a questa infrastruttura, il Security Team è stato in grado di avere pacchetti aggiornati pronti per le uscite di Apache ed OpenSSH per tutte le architetture supportate (quasi venti), in meno di un giorno.

Guida degli sviluppatori agli aggiornamenti sulla sicurezza

Debian developers that need to coordinate with the security team on fixing in issue in their packages, can refer to the Developer's Reference section http://www.debian.org/doc/manuals/developers-reference/pkgs.html#bug-security.

Firma dei pacchetti in Debian

This section could also be titled "how to upgrade/update safely your Debian GNU/Linux system" and it deserves its own section basically because it is an important part of the Security Infrastructure. Package signing is an important issue since it avoids tampering of packages distributed in mirrors and of downloads with man-in-the-middle attacks. Automatic software update is an important feature but it's also important to remove security threats that could help the distribution of trojans and the compromise of systems during updates [3]

FIXME: probably the Internet Explorer vulnerability handling. certificate chains has an impact on security updates on Microsoft Windows.

Debian does not provide signed packages but provides a mechanism available since Debian 4.0 (codename *etch*) to check for downloaded package's integrity[4]. For more information, see sezione chiamata «Apt sicuro».

Questo problema viene discusso più approfonditamente nello http://www.cryptnet.net/fdp/crypto/strong_distro.html di V. Alex Brennen.

[3] Some operating systems have already been plagued with automatic-updates problems such as the http://www.cunap.com/~hardingr/projects/osx/exploit.html.
[4] Older releases, such as Debian 3.1 *sarge* can use this feature by using backported versions of this package management tool

The current scheme for package signature checks

L'attuale (non implementato) schema di controllo della firma dei pacchetti usando **apt** è il seguente:

- Il file `Release` include la somma MD5 di Packages.gz (che contiene le somme MD5 dei pacchetti) e verrà firmato. La firma è una delle fonti certe.
- Questo file `Release` firmato viene scaricato con 'apt-get update' e conservato insieme a `Packages.gz`.
- Quando un pacchetto sta per essere installato, viene prima scaricato, successivamente viene generata la somma MD5.
- Il file `Release` firmato viene controllato (la firma è ok) e viene estratto da questo la somma MD5 per il file `Packages.gz`, il checksum di `Packages.gz` viene generato e (se ok) la somma MD5 del pacchetto scaricato viene calcolata direttamente da sé stesso.
- Se la somma MD5 del pacchetto scaricato è la stessa di quella nel file `Packages.gz` allora verrà installato, altrimenti l'amministratore verrà avvisato ed il pacchetto verrà lasciato in cache (così l'amministratore potrà decidere se installarlo o meno). Se il pacchetto non è in `Packages.gz` e l'amministratore ha configurato il sistema per installare solo pacchetti firmati non sarà nemmeno installabile.

Seguendo la catena delle somme MD5 **apt** è in grado di verificare se un pacchetto proviene da una determinata distribuzione rilasciata da Debian. Questo è meno flessibile che firmare ogni singolo pacchetto, uno ad uno, ma può essere combinato anche con questo schema (vedete più sotto).

This scheme is http://lists.debian.org/debian-devel/2003/12/msg01986.html in apt 0.6 and is available since the Debian 4.0 release. For more information see sezione chiamata «Apt sicuro». Packages that provide a front-end to apt need to be modified to adapt to this new feature; this is the case of **aptitude** which was http://lists.debian.org/debian-devel/2005/03/msg02641.html to adapt to this scheme. Front-ends currently known to work properly with this feature include **aptitude** and **synaptic**.

La firma dei pacchetti è stata discussa in debian per molto tempo, per altre informazioni leggete: http://www.debian.org/News/weekly/2001/8/ e http://www.debian.org/News/weekly/2000/11/.

Apt sicuro

The apt 0.6 release, available since Debian 4.0 *etch* and later releases, includes *apt-secure* (also known as *secure apt*) which is a tool that will allow a system administrator to test the integrity of the packages downloaded through the above scheme. This release includes the tool **apt-key** for adding new keys to apt's keyring, which by default includes only the current Debian archive signing key.

These changes are based on the patch for **apt** (available in http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=203741) which provides this implementation.

Apt sicuro funziona controllando la distribuzione attraverso il file `Release`, come discusso in sezione chiamata «Controllo di rilascio per ogni distribuzione». Di solito, questo processo è trasparente all'amministratore anche se è necessario un intervento ogni anno[5] per aggiungere la nuova chiave all'archivio quando vengono ruotati i log, per maggiori informazioni sui passi che un amministratore deve compiere vedete sezione chiamata «Come aggiungere una chiave in maniera sicura».

This feature is still under development, if you believe you find bugs in it, please, make first sure you are using the latest version (as this package might change quite a bit before it is finally released) and, if running the latest version, submit a bug against the apt package.

[5] Fino a quando non verrà sviluppato un processo automatico.

You can find more information at http://wiki.debian.org/SecureApt and the official documentation: http://www.enyo.de/fw/software/apt-secure/ and https://web.archive.org/web/20070206063141/http://www.syntaxpolice.org/apt-secure/.

Controllo di rilascio per ogni distribuzione

Questa sezione descrive come il meccanismo di controllo del rilascio della distribuzione funzioni, fu scritto da Joey Hess ed è disponibile anche presso il http://wiki.debian.org/SecureApt.

Concetti di base

Qui diamo alcuni concetti di base necessari per capire il resto di questa sezione.

Un checksum è un metodo che consiste nel prendere un file e manipolarlo per tirarne fuori un numero di una lunghezza ragionevole che identifichi in maniera univoca il contenuto. Ciò è un po' più complicato di quanto possa sembrare, ed il tipo di checksum più usato, la somma MD5, è stato completamente violato.

La crittografia a chiave pubblica è basata su una coppia di chiavi, una pubblica ed una privata. La chiave pubblica deve essere diffusa il più possibile; la chiave privata deve invece rimanere segreta. Chiunque possegga la chiave pubblica può cifrare un messaggio che possa essere letto solo da chi possiede la chiave privata corrispondente. È anche possibile usare la chiave privata per firmare un file, ma non per cifrarlo. Se una chiave privata viene usata per firmare un file, chiunque in possesso della chiave pubblica corrispondente può verificare se il file è stato firmato da quella chiave. Nessuno che non possegga la chiave privata può creare una tale firma.

Queste chiavi sono dei numeri abbastanza lunghi (da 1024 o 2048 cifre binarie), e perché sia più semplice lavorarci, hanno un identificativo di chiave, che è un numero più corto, di 8 o 16 cifre binarie.

gpg è lo strumento usato nella versione sicura di apt per firmare i file e verificarne le firme.

apt-key è un programma che viene usato per gestire un portachiavi di chiavi gpg per apt sicuro. Il portachiavi viene mantenuto nel file `/etc/apt/trusted.gpg` (da non confondersi con file `/etc/apt/trustdb.gpg` sempre legato ad apt, ma non troppo interessante). **apt-key** può essere usato per mostrare le chiavi del portachiavi, per aggiungerne e per rimuoverne altre.

Checksum di `Release`

Un archivio Debian contiene un file `Release`, che viene aggiornato ogni volta che un pacchetto dell'archivio cambia. Tra l'altro, il file `Release` contiene le somme MD5 di altri file presenti nell'archivio. Un estratto di un file `Release` di esempio:

```
MD5Sum:
 6b05b392f792ba5a436d590c129de21f            3453 Packages
 1356479a23edda7a69f24eb8d6f4a14b            1131 Packages.gz
 2a5167881adc9ad1a8864f281b1eb959            1715 Sources
 88de3533bf6e054d1799f8e49b6aed8b             658 Sources.gz
```

I file `Release` comprendono anche un checksum SHA-1, che sarà utile una volta che i checksum MD5 verranno completamente rotti, ad ogni modo apt per ora non li usa.

Adesso, se guardiamo in un file `Packages`, troveremo diversi checksum MD5, uno per ogni pacchetto elencato. Ad esempio:

```
Package: uqm
Priority: optional
```

```
...
Filename: unstable/uqm_0.4.0-1_i386.deb
Size: 580558
MD5sum: 864ec6157c1eea88acfef44d0f34d219
```

Questi due checksum possono essere usati per verificare che si sia scaricata una versione corretta del file `Packages`, con un md5sum che corrisponde con quello del file `Release`. Inoltre, quando scaricate un singolo pacchetto, potete verificare il suo md5sum con quello contenuto nel file `Packages`. Se apt fallisce entrambi questi passi, abortisce.

Nulla di questo è nuovo nella versione sicura di apt, ma ne fornisce le fondamenta. Notate che ad ogni modo c'è un file che apt non ha modo di verificare: il file `Release`. Questa versione sicura di apt ruota tutto attorno alla verifica del file `Release` prima di intraprendere qualunque azione, in modo che ci sia una catena ininterrotta di verifica, dal pacchetto che state per installare fino al al fornitore del pacchetto stesso.

Verifica del file `Release`

Per verificare il file `Release`, viene aggiunta al file una firma gpg. Questa viene messa in un file chiamato `Release.gpg` che viene fornito assieme al file `Release`. Assomiglia a qualcosa del genere[6] , anche se di solito solo gpg guarda il suo contenuto:

```
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.1 (GNU/Linux)

iD8DBQBCqKO1nukh8wJbxY8RAsfHAJ9hu8oGNRAl2MSmP5+z2RZb6FJ8kACfWvEx
UBGPVc7jbHHsg78EhMBlV/U=
=x6og
-----END PGP SIGNATURE-----
```

Verifica del file `Release.gpg` da parte di apt

La versione sicura di Apt scarica sempre i file `Release.gpg` quando scarica i file `Release` e se non riesce a farlo, o se la firma non è valida, si lamenterà ed annoterà che il file `Packages` a cui il file `Release` punta, con tutti i pacchetti elencati al suo interno, provengono da una sorgente non fidata. Ecco a cosa somiglia un'operazione come **apt-get update**:

```
W: GPG error: http://ftp.us.debian.org testing Release: The following signatures
 couldn't be verified because the public key is not available: NO_PUBKEY 010908312
```

Notate che la seconda metà del numero "long" è l'identificativo della chiave che apt non conosce, e in questo caso è 2D230C5F.

Se ignorerete l'avvertimento e tenterete di installare un pacchetto più avanti, apt avvertirà nuovamente:

```
WARNING: The following packages cannot be authenticated!
  libglib-perl libgtk2-perl
Install these packages without verification [y/N]?
```

Se risponderete Y non avrete alcun modo di sapere se il file che state per ricevere è il pacchetto che intendete installare o se è qualcos'altro che qualcuno può aver deciso di spedirvi avendolo preparato appositamente con all'interno una cattiva sorpresa, dopo aver intercettato la comunicazione verso il server[7].

[6] Parlando tecnicamente, questa è una firma gpg ASCII.
[7] O abbia contraffatto il vostro DNS, o si stia fingendo il server, o abbia rimpiazzato il file nel mirror che state usando, etc..

Notate che potete disabilitare questi controlli eseguendo apt con l'opzione --allow-unauthenticated.

Vale anche la pena notare che le nuove versioni dell'installatore Debian usano lo stesso meccanismo del file `Release` firmato durante il debootstrap del sistema base Debian, prima che apt sia disponibile. Perfino l'installer usa questo sistema per verificare pezzi di se stesso che scarica dalla rete. Inoltre, Debian attualmente non firma i suoi file `Release` nei CD; apt può essere comunque configurato per fidarsi sempre dei pacchetti dai CD così che questo non sia un grosso problema.

Come dire ad apt di cosa fidarsi

Quindi la sicurezza dell'intero sistema dipende dalla presenza di un file `Release.gpg`, che firma un file `Release` e da **apt** che verifica tale firma usando gpg. Per verificare la firma dovete conoscere la chiave pubblica del firmatario. Queste chiavi vengono conservate proprio nel portachiavi di apt (`/etc/apt/trusted.gpg`) e con la gestione delle chiavi fa il suo ingresso la versione sicura di apt.

I sistemi Debian vengono configurati, in modo predefinito, con la chiave dell'archivio Debian nel portachiavi.

```
# apt-key list
/etc/apt/trusted.gpg
--------------------
pub   1024D/4F368D5D 2005-01-31 [expires: 2006-01-31]
uid                  Debian Archive Automatic Signing Key (2005) <ftpmaster@debian
```

In questo caso 4F368D5D è l'id della chiave e notate che tale chiave rimane valida solo per un anno. Debian cambia regolarmente queste chiavi come ultima misura di sicurezza nel caso in cui le chiavi vengano compromesse.

That will make **apt** trust the official Debian archive, but if you add some other apt repository to `/etc/apt/sources.list`, you'll also have to give **apt** its key if you want apt to trust it. Once you have the key and have verified it, it's a simple matter of running **apt-key add** *`file`* to add it. Getting the key and verifying it are the trickier parts.

Come ottenere la chiave di un repository

Il pacchetto debian-archive-keyring viene usato per distribuire le chiavi tramite **apt**. Aggiornate questo pacchetto per aggiungere (o rimuovere) chiavi gpg relative all'archivio principale di Debian.

Per quanto riguarda gli altri archivi, non esiste ancora un sito standard dove trovare le chiavi di un qualsiasi repository apt. Di solito la chiave viene messa sulla pagina web del repository o in un file nel repository stesso, ma non esiste uno standard ben definito e potreste doverla cercare da soli.

The Debian archive signing key is available at https://ftp-master.debian.org/keys.html.[8]

Lo stesso programma **gpg** ha una procedura standard per distribuire le chiavi, scaricando le chiavi da un server apposito ed aggiungendole al suo portachiavi. Per esempio:

```
$ gpg --keyserver pgpkeys.mit.edu --recv-key 2D230C5F
gpg: requesting key 2D230C5F from hkp server pgpkeys.mit.edu
gpg: key 2D230C5F: public key "Debian Archive Automatic Signing Key (2006) <ftpm
aster@debian.org>" imported
```

[8] "ziyi" is the name of the tool used for signing on the Debian servers, the name is based on the name of a http://en.wikipedia.org/wiki/Zhang_Ziyi.

```
gpg: Total number processed: 1
gpg:               imported: 1
```

Potete poi esportare tale chiave dal vostro portachiavi e fornirla al programma **apt-key**:

```
$ gpg -a --export 2D230C5F | sudo apt-key add -
gpg: no ultimately trusted keys found
OK
```

L'avviso "gpg: no ultimately trusted keys found" indica che gpg non è stato configurato per dare fiducia indiscussa a una delle chiavi presenti nel vostro portachiavi. Il livello di fiducia nelle singole chiavi dipende dalla http://it.wikipedia.org/wiki/Web_of_trust usata da OpenPGP ed in questo caso potete ignorarlo. In una configurazione tipica, viene data fiducia indiscussa alla chiave dell'utente.

Come aggiungere una chiave in maniera sicura

By adding a key to apt's keyring, you're telling apt to trust everything signed by the key, and this lets you know for sure that apt won't install anything not signed by the person who possesses the private key. But if you're sufficiently paranoid, you can see that this just pushes things up a level, now instead of having to worry if a package, or a `Release` file is valid, you can worry about whether you've actually gotten the right key. Is the key file from https://ftp-master.debian.org/keys.html mentioned above really Debian's archive signing key, or has it been modified (or this document lies).

It's good to be paranoid in security, but verifying things from here is harder. **gpg** has the concept of a chain of trust, which can start at someone you're sure of, who signs someone's key, who signs some other key, etc., until you get to the archive key. If you're sufficiently paranoid you'll want to check that your archive key is signed by a key that you can trust, with a trust chain that goes back to someone you know personally. If you want to do this, visit a Debian conference or perhaps a local LUG for a key signing [9].

Se non siete così paranoici, potete fare quello che ritenete più opportuno ogni volta che usate un nuovo archivio di pacchetti e dovrete quindi aggiungere una chiave ad apt. Potreste inviare una email alla persona che vi ha inviato la chiave chiedendo conferma della bontà della chiave, o potreste semplicemente scaricare la chiave ed assumere che sia quella giusta. Il vantaggio principale di questa versione sicura di apt è che, visto che ogni pacchetto viene firmato da una chiave, il problema dell'autenticità dei pacchetti si riduce al problema dell'autenticità della chiave che li firmano; perciò, potete scegliere il numero e la qualità dei controlli sull'autenticità della chiave a vostro piacimento.

Verifica dell'integrità della chiave

You can verify the fingerprint as well as the signatures on the key. Retrieving the fingerprint can be done for multiple sources, you can talk to Debian Developers on IRC, read the mailing list where the key change will be announced or any other additional means to verify the fingerprint. For example you can do this:

```
$ GET http://ftp-master.debian.org/ziyi_key_2006.asc | gpg --import
gpg: key 2D230C5F: public key "Debian Archive Automatic Signing Key (2006)
  <ftpmaster&debian.org>" imported
gpg: Total number processed: 1
gpg:               imported: 1
$ gpg --check-sigs --fingerprint 2D230C5F
```

[9] Not all apt repository keys are signed at all by another key. Maybe the person setting up the repository doesn't have another key, or maybe they don't feel comfortable signing such a role key with their main key. For information on setting up a key for a repository see sezione chiamata «Controllo della versione su fonti esterne a Debian».

```
pub   1024D/2D230C5F 2006-01-03 [expires: 2007-02-07]
      Key fingerprint = 0847 50FC 01A6 D388 A643  D869 0109 0831 2D23 0C5F
uid   Debian Archive Automatic Signing Key (2006) <ftpmaster@debian.org>
sig!3        2D230C5F 2006-01-03  Debian Archive Automatic Signing Key
                                  (2006) <ftpmaster@debian.org>
sig!         2A4E3EAA 2006-01-03  Anthony Towns <aj@azure.humbug.org.au>
sig!         4F368D5D 2006-01-03  Debian Archive Automatic Signing Key
                                  (2005) <ftpmaster@debian.org>
sig!         29982E5A 2006-01-04  Steve Langasek <vorlon@dodds.net>
sig!         FD6645AB 2006-01-04  Ryan Murray <rmurray@cyberhqz.com>
sig!         AB2A91F5 2006-01-04  James Troup <james@nocrew.org>
```

and then as in sezione chiamata «Firma dei pacchetti in Debian» check the trust path from your key (or a key you trust) to at least one of the keys used to sign the archive key. If you are sufficiently paranoid you will tell apt to trust the key only if you find an acceptable path:

```
$ gpg --export -a 2D230C5F | sudo apt-key add -
Ok
```

Notate che la nuova chiave è firmata con la chiave precedente dell'archivio, pertanto teoricamente potreste limitarvi ad aggiungere un anello alla catena di fiducia costruita in precedenza.

Rinnovo annuale delle chiavi degli archivi Debian

Come summenzionato, la chiave utilizzata per firmare gli archivi Debian viene cambiata ogni anno, a gennaio. Poiché la versione sicura di apt è un programma giovane, i suoi sviluppatori non hanno ancora ben collaudato le procedure da seguire per il rinnovo della chiave e ci potrebbero essere dei passi da compiere ancora non ben definiti.

Nel gennaio 2006 il file `Release` venne firmato dalla nuova chiave ma, per evitare di danneggiare i sistemi che usavano ancora la chiave del 2005, tale file venne firmato anche dalla vecchia chiave. Si pensava che apt avrebbe considerato validi i file firmati da almeno una chiave che apt considerasse fidata, ma apt aveva un difetto per cui considerava validi solo i file firmati esclusivamente da chiavi che apt considerasse fidate. Questo difetto fu corretto nella versione 0.6.43.1. di apt. Si creò un po' di confusione anche sulla distribuzione della nuova chiave agli utenti che già usavano la versione sicura di apt; in un primo momento venne distribuita sul sito web senza un annuncio specifico e senza che gli utenti fossero in grado di verificarla; gli utenti furono costretti a scaricarla ed installarla a mano.

Nel gennaio 2006, fu creata una nuova chiave per il 2006 e si cominciò a firmare con essa il file `Release`, ma per cercare di non rompere la compatibilità con i sistemi che avevano ancora la vecchia chiave del 2005, il file `Release` venne firmato anche con quella vecchia. Per evitare confusione sul miglior meccanismo di distribuzione per gli utenti che già avevano sistemi che usavano la versione sicura di apt, fu introdotto il pacchetto debian-archive-keyring, che gestisce gli aggiornamenti del portachiavi di apt.

Problemi noti sul controllo della versione

Un problema non molto ovvio è che se l'orologio di sistema è sballato di molto, la versione sicura di apt non potrà funzionare. Se l'orologio segna una data del passato, ad esempio il 1999, apt se ne uscirà con un messaggio d'errore poco utile come questo:

```
W: GPG error: http://archive.progeny.com sid Release: Unknown error executing gpg
```

In ogni caso **apt-key** renderà chiaro il problema:

```
gpg: key 2D230C5F was created 192324901 seconds in the future (time warp or clock
gpg: key 2D230C5F was created 192324901 seconds in the future (time warp or clock
pub   1024D/2D230C5F 2006-01-03
uid                  Debian Archive Automatic Signing Key (2006) <ftpmaster@debian
```

Se invece segna una data troppo distante nel futuro, apt considererà le chiavi come scadute.

Un altro problema in cui vi potete imbattere usando testing o unstable è che, se non avete eseguito recentemente **apt-get update** prima di **apt-get install** su di un pacchetto che volete installare, apt potrebbe lamentarsi di non potersi autenticare (perché lo fa?). **apt-get update** correggerà il problema.

Controllare manualmente i rilasci della distribuzione

Nel caso in cui vogliate aggiungere adesso i controlli di sicurezza aggiuntivi e non vogliate o possiate eseguire l'ultima versione di apt[10] , potete usare lo script qui sotto, fornito da Anthony Towns. Questo script può eseguire automaticamente dei nuovi controlli di sicurezza per far sì che l'utente sia certo che il software che sta scaricando sia lo stesso che Debian distribuisce. Ciò impedisce agli sviluppatori Debian di violare il sistema di qualcuno senza la garanzia di autenticità derivante dall'inclusione nell'archivio principale, ai mirror di fornire qualcosa di molto simile a Debian ma non esattamente uguale ad essa, o copie non aggiornate di unstable con problemi di sicurezza noti.

Questo semplice codice, rinominato come **apt-check-sigs**, dovrebbe essere usato nel modo seguente:

```
# apt-get update
# apt-check-sigs
(...results...)
# apt-get dist-upgrade
```

Per prima cosa avete bisogno di:

- get the keys the archive software uses to sign `Release` files from https://ftp-master.debian.org/keys.html and add them to `~/.gnupg/trustedkeys.gpg` (which is what **gpgv** uses by default).

```
gpg --no-default-keyring --keyring trustedkeys.gpg --import ziyi_key_2006.asc
```

- Rimuovere tutte le righe di `/etc/apt/sources.list` che non usano la normale struttura "dists" o modificare lo script in modo che funzioni con esse.
- Essere preparati ad ignorare il fatto che gli aggiornamenti di sicurezza di Debian non hanno file `Release` firmati e che i file `Sources` non hanno (ancora) un checksum appropriato nel file `Release`.
- Essere pronti a verificare che i sorgenti siano firmati da chiavi appropriate.

This is the example code for **apt-check-sigs**, the latest version can be retrieved from http://people.debian.org/~ajt/apt-check-sigs. This code is currently in beta, for more information read http://lists.debian.org/debian-devel/2002/07/msg00421.html.

```
#!/bin/bash
```

[10] O perché state usando la versione stabile, *sarge*, o una versione più vecchia o perché non volete usare l'ultima versione di apt, anche se apprezzeremmo molto che gli utenti la collaudassero.

```
# Copyright (c) 2001 Anthony Towns <ajt@debian.org>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.

rm -rf /tmp/apt-release-check
mkdir /tmp/apt-release-check || exit 1
cd /tmp/apt-release-check

>OK
>MISSING
>NOCHECK
>BAD

arch=`dpkg --print-installation-architecture`

am_root () {
        [ `id -u` -eq 0 ]
}

get_md5sumsize () {
        cat "$1" | awk '/^MD5Sum:/,/^SHA1:/' |
          MYARG="$2" perl -ne '@f = split /\s+/; if ($f[3] eq $ENV{"MYARG"}) {
print "$f[1] $f[2]\n"; exit(0); }'
}

checkit () {
        local FILE="$1"
        local LOOKUP="$2"

        Y="`get_md5sumsize Release "$LOOKUP"`"
        Y="`echo "$Y" | sed 's/^ *//;s/  */ /g'`"

        if [ ! -e "/var/lib/apt/lists/$FILE" ]; then
                if [ "$Y" = "" ]; then
                        # No file, but not needed anyway
                        echo "OK"
                        return
                fi
                echo "$FILE" >>MISSING
                echo "MISSING $Y"
                return
        fi
        if [ "$Y" = "" ]; then
                echo "$FILE" >>NOCHECK
                echo "NOCHECK"
                return
```

```
        fi
        X="`md5sum < /var/lib/apt/lists/$FILE | cut -d\  -f1` `wc -c < /var/lib
/apt/lists/$FILE`"
        X="`echo "$X" | sed 's/^ *//;s/  */ /g'`"
        if [ "$X" != "$Y" ]; then
                echo "$FILE" >>BAD
                echo "BAD"
                return
        fi
        echo "$FILE" >>OK
        echo "OK"
}

echo
echo "Checking sources in /etc/apt/sources.list:"
echo "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
echo
(echo "You should take care to ensure that the distributions you're downloading
"
echo "are the ones you think you are downloading, and that they are as up to"
echo "date as you would expect (testing and unstable should be no more than"
echo "two or three days out of date, stable-updates no more than a few weeks"
echo "or a month)."
) | fmt
echo

cat /etc/apt/sources.list |
  sed 's/^ *//' | grep '^[^#]' |
  while read ty url dist comps; do
        if [ "${url%%:*}" = "http" -o "${url%%:*}" = "ftp" ]; then
                baseurl="${url#*://}"
        else
                continue
        fi

        echo "Source: ${ty} ${url} ${dist} ${comps}"

        rm -f Release Release.gpg
        lynx -reload -dump "${url}/dists/${dist}/Release" >/dev/null 2>&1
        wget -q -O Release "${url}/dists/${dist}/Release"

        if ! grep -q '^' Release; then
                echo "  * NO TOP-LEVEL Release FILE"
                >Release
        else
                origline=`sed -n 's/^Origin: *//p' Release | head -1`
                lablline=`sed -n 's/^Label: *//p' Release | head -1`
                suitline=`sed -n 's/^Suite: *//p' Release | head -1`
                codeline=`sed -n 's/^Codename: *//p' Release | head -1`
                dateline=`grep "^Date:" Release | head -1`
                dscrline=`grep "^Description:" Release | head -1`
                echo "  o Origin: $origline/$lablline"
                echo "  o Suite: $suitline/$codeline"
                echo "  o $dateline"
```

```
                echo "  o $dscrline"

                if [ "${dist%%/*}" != "$suitline" -a "${dist%%/*}" != "$codeline"
                        echo "  * WARNING: asked for $dist, got $suitline/$codelin
                fi

                lynx -reload -dump "${url}/dists/${dist}/Release.gpg" >/dev/null 2
                wget -q -O Release.gpg "${url}/dists/${dist}/Release.gpg"

                gpgv --status-fd 3 Release.gpg Release 3>&1 >/dev/null 2>&1 | sed
                        if [ "$gpgcode" = "GOODSIG" ]; then
                            if [ "$err" != "" ]; then
                                echo "  * Signed by ${err# } key: ${rest#* }"
                            else
                                echo "  o Signed by: ${rest#* }"
                                okay=1
                            fi
                            err=""
                        elif [ "$gpgcode" = "BADSIG" ]; then
                            echo "  * BAD SIGNATURE BY: ${rest#* }"
                            err=""
                        elif [ "$gpgcode" = "ERRSIG" ]; then
                            echo "  * COULDN'T CHECK SIGNATURE BY KEYID: ${rest %%
                            err=""
                        elif [ "$gpgcode" = "SIGREVOKED" ]; then
                            err="$err REVOKED"
                        elif [ "$gpgcode" = "SIGEXPIRED" ]; then
                            err="$err EXPIRED"
                        fi
                    done
                    if [ "$okay" != 1 ]; then
                        echo "  * NO VALID SIGNATURE"
                        >Release
                    fi)
        fi
        okaycomps=""
        for comp in $comps; do
                if [ "$ty" = "deb" ]; then
                        X=$(checkit "`echo "${baseurl}/dists/${dist}/${comp}/binar
                        Y=$(checkit "`echo "${baseurl}/dists/${dist}/${comp}/binar
                        if [ "$X $Y" = "OK OK" ]; then
                                okaycomps="$okaycomps $comp"
                        else
                                echo "  * PROBLEMS WITH $comp ($X, $Y)"
                        fi
                elif [ "$ty" = "deb-src" ]; then
                        X=$(checkit "`echo "${baseurl}/dists/${dist}/${comp}/sourc
                        Y=$(checkit "`echo "${baseurl}/dists/${dist}/${comp}/sourc
                        if [ "$X $Y" = "OK OK" ]; then
                                okaycomps="$okaycomps $comp"
                        else
                                echo "  * PROBLEMS WITH component $comp ($X, $Y)"
                        fi
                fi
```

```
        done
        [ "$okaycomps" = "" ] || echo "  o Okay:$okaycomps"
        echo
  done

echo "Results"
echo "~~~~~~~"
echo

allokay=true

cd /tmp/apt-release-check
diff <(cat BAD MISSING NOCHECK OK | sort) <(cd /var/lib/apt/lists && find . -type

cd /tmp/apt-release-check
if grep -q ^ UNVALIDATED; then
    allokay=false
    (echo "The following files in /var/lib/apt/lists have not been validated."
    echo "This could turn out to be a harmless indication that this script"
    echo "is buggy or out of date, or it could let trojaned packages get onto"
    echo "your system."
    ) | fmt
    echo
    sed 's/^/    /' < UNVALIDATED
    echo
fi

if grep -q ^ BAD; then
    allokay=false
    (echo "The contents of the following files in /var/lib/apt/lists does not"
    echo "match what was expected. This may mean these sources are out of date,"
    echo "that the archive is having problems, or that someone is actively"
    echo "using your mirror to distribute trojans."
    if am_root; then
        echo "The files have been renamed to have the extension .FAILED and"
        echo "will be ignored by apt."
        cat BAD | while read a; do
            mv /var/lib/apt/lists/$a /var/lib/apt/lists/${a}.FAILED
        done
    fi) | fmt
    echo
    sed 's/^/    /' < BAD
    echo
fi

if grep -q ^ MISSING; then
    allokay=false
    (echo "The following files from /var/lib/apt/lists were missing. This"
    echo "may cause you to miss out on updates to some vulnerable packages."
    ) | fmt
    echo
    sed 's/^/    /' > MISSING
    echo
fi
```

```
if grep -q ^ NOCHECK; then
    allokay=false
    (echo "The contents of the following files in /var/lib/apt/lists could not"
    echo "be validated due to the lack of a signed Release file, or the lack"
    echo "of an appropriate entry in a signed Release file. This probably"
    echo "means that the maintainers of these sources are slack, but may mean"
    echo "these sources are being actively used to distribute trojans."
    if am_root; then
        echo "The files have been renamed to have the extension .FAILED and"
        echo "will be ignored by apt."
        cat NOCHECK | while read a; do
            mv /var/lib/apt/lists/$a /var/lib/apt/lists/${a}.FAILED
        done
    fi) | fmt
    echo
    sed 's/^/    /' > NOCHECK
    echo
fi

if $allokay; then
    echo 'Everything seems okay!'
    echo
fi

rm -rf /tmp/apt-release-check
```

Potrebbe essere necessario applicare questa patch per *sid* perché **md5sum** aggiunge un '-' dopo la somma quando l'input è stdin:

```
@@ -37,7 +37,7 @@
        local LOOKUP="$2"

        Y="`get_md5sumsize Release "$LOOKUP"`"
-       Y="`echo "$Y" | sed 's/^ *//;s/  */ /g'`"
+       Y="`echo "$Y" | sed 's/-//;s/^ *//;s/  */ /g'`"

        if [ ! -e "/var/lib/apt/lists/$FILE" ]; then
                if [ "$Y" = "" ]; then
@@ -55,7 +55,7 @@
                return
        fi
        X="`md5sum < /var/lib/apt/lists/$FILE` `wc -c < /var/lib/apt/lists/$FILE`"
-       X="`echo "$X" | sed 's/^ *//;s/  */ /g'`"
+       X="`echo "$X" | sed 's/-//;s/^ *//;s/  */ /g'`"
        if [ "$X" != "$Y" ]; then
                echo "$FILE" >>BAD
                echo "BAD"
```

Controllo della versione su fonti esterne a Debian

Notice that, when using the latest apt version (with *secure apt*) no extra effort should be required on your part unless you use non-Debian sources, in which case an extra confirmation step will be required by apt-

get. This is avoided by providing `Release` and `Release.gpg` files in the non-Debian sources. The `Release` file can be generated with **apt-ftparchive** (available in apt-utils 0.5.0 and later), the `Release.gpg` is just a detached signature. To generate both follow this simple procedure:

```
$ rm -f dists/unstable/Release
$ apt-ftparchive release dists/unstable > dists/unstable/Release
$ gpg --sign -ba -o dists/unstable/Release.gpg dists/unstable/Release
```

Un modello alternativo di firma per ciascun pacchetto

L'ulteriore modello per firmare ogni pacchetto, ne permette il controllo quando questi non hanno più riferimenti in un file `Packages` già esistente. Così facendo, anche se i pacchetti sono provenienti da terze parti di Debian, potranno comunque essere usati in Debian, ma non con lo schema predefinito.

This package signing scheme can be implemented using debsig-verify and debsigs. These two packages can sign and verify embedded signatures in the .deb itself. Debian already has the capability to do this now, but there is no feature plan to implement the policy or other tools since the archive signing scheme is prefered. These tools are available for users and archive administrators that would rather use this scheme instead.

Latest **dpkg** versions (since 1.9.21) incorporate a http://lists.debian.org/debian-dpkg/2001/03/msg00024.html that provides this functionality as soon as debsig-verify is installed.

NOTE: Attualmente `/etc/dpkg/dpkg.cfg` nasce con "no-debsig" come valore predefinito.

NOTE2: Le firme degli sviluppatori attualmente vengono eliminate quando depositano il pacchetto nell'archivio poiché lo stile attualmente in voga consiste nel controllare la release come descritto precedentemente.

Capitolo 8. Strumenti per la sicurezza in Debian

FIXME: Trattazione da integrare

Debian offre anche molti strumenti per la sicurezza, tali da garantire una configurazione ottimale della postazione. Questi comprendono la protezione dei sistemi di informazione mediante firewall (a livello sia di pacchetti che di applicazioni), la ricerca delle intrusioni (basata sia su rete che su singoli host), la valutazione dei punti deboli e poi, antivirus, reti private, etc..

A partire da Debian 3.0 (*woody*), la distribuzione è caratterizzata da software specifico per la crittografia integrato nella distribuzione principale. OpenSSH e GNU Privacy Guard vengono inclusi nell'installazione predefinita e già adesso i browser, i server di rete, i database e via dicendo, hanno un robusto sistema di cifratura. Per le future versioni è già pianificata un'ulteriore integrazione della crittografia. Questo tipo di software, per essere esportato, viste alcune limitazioni negli Stati Uniti, non è stato distribuito con la distribuzione principale, ma è stato incluso nei siti non-US.

Strumenti per la valutazione delle vulnerabilità da remoto

The tools provided by Debian to perform remote vulnerability assessment are: [1]

- nessus
- raccess
- nikto (sostituto di whisker's)

nessus - composto da un client (nessus), usato come interfaccia GUI e da un server (nessusd), che lancia gli attacchi programmati è, di gran lunga, il più completo e aggiornato. Conosce i punti deboli di un numero consistente di sistemi contenenti applicazioni di rete, server ftp, server www, etc.. Le ultime versioni sono in grado di analizzare un sito web e provare a scoprire quali, fra le pagine disponibili, potrebbero essere attaccate. Ci sono anche dei client Java e Win32 (non inclusi in Debian), utilizzabili per contattare il server dell'organizzazione aziendale.

nikto - scansionatore per la valutazione di vulnerabilità dedicato solo al web, presenta tattiche anti-IDS (la maggior parte delle quali non è più solamente *anti-IDS*) ed è uno dei migliori scansionatori cgi: può scoprire un server WWW e lanciarvi contro solo una determinata tipologia di attacchi. Il database usato per la scansione è facilmente modificabile, al fine di ottenere nuove informazioni.

Strumenti per effettuare scansioni di rete

Debian fornisce alcuni mezzi dedicati alla scansione di host da remoto (ma non anche della verifica delle vulnerabilità) che, in certi casi, possono essere usati come scansionatori per un primo tipo di "attacco", volto a determinare i servizi dell'host remoto disponibili. Al momento Debian fornisce:

- nmap

[1] Some of them are provided when installing the harden-remoteaudit package.

- xprobe
- p0f
- knocker
- isic
- hping2
- icmpush
- nbtscan (per verifiche su SMB /NetBIOS)
- fragrouter
- **strobe** (nel pacchetto netdiag)
- irpas

Mentre xprobe offre solamente l'esame da remoto del sistema operativo (usando TCP/IP fingerprinting), nmap e knocker svolgono sia l'esame del sistema operativo, sia la scansione delle porte dell'host remoto. D'altra parte, per le tecniche di attacco basate su ICMP da remoto potete usare hping2 e icmpush.

Progettato specificamente per reti di tipo SMB /NetBIOS, nbtscan viene impiegato per scansionare le reti IP e per recuperare informazioni riguardo ai nomi (fra cui: nomi degli utenti, nomi della rete, indirizzi di tipo MAC...) dai server su cui sia abilitato SMB.

Infine, per rilevare intrusioni nella rete e verificare se i NIDS possano essere aggirati da attacchi mediante frammentazione, si può usare fragrouter.

FIXME: Controllare se il http://bugs.debian.org/153117 (ITP fragrouter) sia incluso.

FIXME add information based on https://web.archive.org/web/20040725013857/http://www.giac.org/practical/gcux/Stephanie_Thomas_GCUX.pdf which describes how to use Debian and a laptop to scan for wireless (803.1) networks (link not there any more).

Controlli interni

Al momento solamente il tool chiamato tiger può essere usato per effettuare un controllo interno (chiamato anche white box (NdT:cassetta bianca)) degli host per controllare se il file system è configurato in modo appropriato, quali processi sono in ascolto nell'host etc.

Revisione del codice sorgente

Debian mette a disposizione vari pacchetti che potete usare nella revisione dei sorgenti C/C++ per ricercare errori di programmazione da considerarsi potenziali problemi per la sicurezza:

- flawfinder
- rats
- splint
- pscan

Rete privata virtuale (VPN)

Un network privato virtuale (VPN) è un gruppo di due o più computer che hanno tra di loro una connessione privata con un accesso limitato dall'esterno, in pratica permette una comunicazione sicura su rete pubblica. VPN permette la connessione tra un singolo computer e una rete privata (client-server), oppure tra una LAN remota ed una rete privata (server-server). VPN include la possibilità di cifrare ed autenticare in modalità sicura utenti remoti oppure host ed inoltre anche metodi per nascondere la tipologia della rete.

Debian mette a disposizione alcuni pacchetti per configurare correttamente una rete privata cifrata:

- vtun
- tunnelv (sezione non-US)
- cipe-source, cipe-common
- tinc
- secvpn
- pptpd
- openvpn
- openswan (http://www.openswan.org/)

FIXME: aggiornare qui le informazioni fino a quando non saranno scritte con FreeSWAN in testa. Controllare l'errore #237764 e l'id del messaggio: <200412101215.04040.rmayr@debian.org>.

Il pacchetto FreeSWAN probabilmente è la miglior scelta tra tutte quelle proposte, può essere usato da chiunque usi il protocollo di sicurezza IPsec (RFC 2411). Comunque gli altri pacchetti presenti nella lista possono aiutare a creare facilmente un tunnel. Il tunnel punto punto (PPTP) è un protocollo per VPN di proprietà della Microsoft. Viene supportato da Linux, ma sono noti seri problemi concernenti la sicurezza.

For more information see the http://www.tldp.org/HOWTO/VPN-Masquerade-HOWTO.html (covers IPsec and PPTP), http://www.tldp.org/HOWTO/VPN-HOWTO.html (covers PPP over SSH), http://www.tldp.org/HOWTO/mini/Cipe+Masq.html, and http://www.tldp.org/HOWTO/mini/ppp-ssh/index.html.

Also worth checking out is http://yavipin.sourceforge.net/, but no Debian packages seem to be available yet.

Tunneling punto punto

Se volete avere un tunnel sicuro tra sistemi operativi differenti (nello stesso tempo postazioni con sistema operativo Microsoft e con Linux) e che IPsec non sia un opzione (è il solo fornito per Windows 2000 e Windows XP), dovrete usare *PoPToP* (server tunneling punto punto), fornito dal pacchetto pptpd.

Se volete usare l'autenticazione e criptazione di Microsoft, mediante il server fornito dal pacchetto ppp, leggete questo estratto tratto dalle FAQ:

```
It is only necessary to use PPP 2.3.8 if you want Microsoft compatible
MSCHAPv2/MPPE authentication and encryption. The reason for this is that
the MSCHAPv2/MPPE patch currently supplied (19990813) is against PPP
2.3.8. If you don't need Microsoft compatible authentication/encryption
```

```
any 2.3.x PPP source will be fine.
```

Comunque, dovete anche applicare una patch al kernel, presente nel pacchetto kernel-patch-mppe, che mette a disposizione il modulo pp-mppe per pppd.

Avere un account cifrato con ppptp vi obbliga a conservare le password in chiaro e senza cifratura, inoltre nel protocollo MS_CHAPv2 sono presenti http://mopo.informatik.uni-freiburg.de/pptp_mschapv2/.

Infrastruttura a chiave pubblica (PKI)

L'infrastruttura a chiave pubblica (PKI) è un'architettura di sicurezza introdotta per fornire un maggior livello di confidenza nello scambiarsi informazioni tramite una rete insicura. Fa uso del concetto di chiavi pubbliche e private per verificare l'identità del mittente (firma) e per assicurare la privacy (crittografia).

Nel prendere in considerazione una PKI ci si trova davanti a una serie di questioni:

- Un'Autorità Certificatrice (CA) che possa creare e verificare certificati e che funzioni sotto una determinata gerarchia.
- Una directory che contenga i certificati pubblici degli utenti.
- Un database (?) per mantenere le Liste di Revoca dei Certificati (CRL).
- Dispositivi che interoperino con la CA al fine di produrre smart card/memorie-USB/qualsiasi altra cosa in cui poter racchiudere i certificati con sicurezza.
- Applicazioni che individuino i certificati, che possano usare quelle prodotte da una CA nell'ambito di una comunicazione cifrata e che controllino i certificati ricevuti rispetto ad una CRL (per l'autenticazione e le soluzioni Single Sign On).
- Un'autorità che segni la data e quindi la scadenza, per la firma digitale dei documenti.
- Una console dalla quale possa essere utilizzato tutto ciò (generazione dei certificati, controllo delle liste di revoca, etc...).

Debian GNU/Linux has software packages to help you with some of these PKI issues. They include **OpenSSL** (for certificate generation), **OpenLDAP** (as a directory to hold the certificates), **gnupg** and **openswan** (with X.509 standard support). However, as of the Woody release (Debian 3.0), Debian does not have any of the freely available Certificate Authorities such as pyCA, http://www.openca.org or the CA samples from OpenSSL. For more information read the http://ospkibook.sourceforge.net/.

Infrastruttura SSL

Debian fornisce alcune certificazioni SSL con la propria distribuzione, in modo da poterle installare localmente. Si trovano nel pacchetto ca-certificates. Questo pacchetto fornisce un deposito locale di certificazioni che sono state inviate a Debian e approvate (ossia verificate) dal manutentore del pacchetto, utili per applicazioni OpenSSL che verifichino le connessioni SSL.

FIXME: leggere debian-devel per vedere se ci sono aggiunte.

Antivirus

Non ci sono molti antivirus in Debian GNU/Linux, probabilmente perché gli utenti GNU/Linux non vengono danneggiati dai virus. Il modello di sicurezza degli Unix fa distinzione tra i processi privilegiati (root) ed i processi utente, quindi un eseguibile "ostile" che un utente non-root riceve o crea ed esegue non può

"infettare" o manipolare l'intero sistema. Comunque, i worms per GNU/Linux ed i virus esistono, anche se non ce n'è nessuno (beh, speriamo) che si sia diffuso a macchia d'olio in una distribuzione Debian. In ogni caso, gli amministratori potrebbero voler mettere su dei gateway antivirus che proteggano dai virus altri sistemi più vulnerabili che fanno parte della loro rete.

Attualmente Debian GNU/Linux fornisce i seguenti strumenti per realizzare ambienti antivirus:

- http://www.clamav.net, provided since Debian *sarge* (3.1 release). Packages are provided both for the virus scanner (clamav) for the scanner daemon (clamav-daemon) and for the data files needed for the scanner. Since keeping an antivirus up-to-date is critical for it to work properly there are two different ways to get this data: clamav-freshclam provides a way to update the database through the Internet automatically and clamav-data which provides the data files directly. [2]

- mailscanner è un e-mail gateway con virus scanner e controllo dello spam. Usando sendmail o exim come sua base, può usare fino a 17 diversi motori di ricerca di virus (incluso clamav).

- libfile-scan-perl che fornisce File::Scan, un'estensione Perl in grado di effettuare una scansione dei file per ricercare virus. Questo modulo può essere usato per eseguire scansioni indipendentemente dalla piattaforma adottata.

- http://www.sourceforge.net/projects/amavis, provided in the package amavis-ng and available in *sarge*, which is a mail virus scanner which integrates with different MTA (Exim, Sendmail, Postfix, or Qmail) and supports over 15 virus scanning engines (including clamav, File::Scan and openantivirus).

- http://packages.debian.org/sanitizer, a tool that uses the procmail package, which can scan email attachments for viruses, block attachments based on their filenames, and more.

- http://packages.debian.org/amavis-postfix, a script that provides an interface from a mail transport agent to one or more commercial virus scanners (this package is built with support for the **postfix** MTA only).

- exiscan, un antivirus per e-mail scritto in Perl che lavora con Exim.

- blackhole-qmail un filtro di spam per Qmail con supporto integrato per Clamav.

Alcuni demoni gateway con supporto per strumenti esterni per costruire un ambiente antivirus sono: exim4-daemon-heavy (la versione *pesante* dell'MTA Exim), frox (un server proxy ftp trasparente con cache), messagewall (un demone proxy SMTP) e pop3vscan (un proxy POP3 trasparente).

Attualmente Debian fornisce **clamav** come unico software di scansione antivirus nella distribuzione stabile ufficiale e fornisce anche interfacce multiple per generare gateway con funzionalità di antivirus per vari protocolli.

Some other free software antivirus projects which might be included in future Debian GNU/Linux releases:http://sourceforge.net/projects/openantivirus/ (see http://bugs.debian.org/150698 and http://bugs.debian.org/150695).

FIXME: È questo il pacchetto che fornisce lo script per scaricare gli ultimi virus firmati da http://www.openantivirus.org/latest.php?

FIXME: Controllare se scannerdaemon non sia in realtà lo scannerdaemon dell'open antivirus (leggere ITPs).

[2] If you use this last package and are running an official Debian, the database will not be updated with security updates. You should either use clamav-freshclam, **clamav-getfiles** to generate new clamav-data packages or update from the maintainers location:

```
deb http://people.debian.org/~zugschlus/clamav-data/ /
deb-src http://people.debian.org/~zugschlus/clamav-data/ /
```

However, Debian will *never* provide propietary (non-free and undistributable) antivirus software such as: Panda Antivirus, NAI Netshield, http://www.sophos.com/, http://www.antivirus.com, or http://www.ravantivirus.com. For more pointers see the http://www.computer-networking.de/~link/security/av-linux_e.txt. This does not mean that this software cannot be installed properly in a Debian system[3].

For more information on how to set up a virus detection system read Dave Jones' article https://web.archive.org/web/20120509212938/http://www.linuxjournal.com/article/4882.

GPG

Oggigiorno la firma digitale e a volte, la cifratura delle e-mail, sono molto diffuse: per esempio, molte persone che partecipano a mailing list (gruppi di discussione a mezzo e-mail) firmano le proprie mail verso la lista. Le firme a chiave pubblica sono gli unici mezzi per verificare che un'e-mail sia stata spedita proprio dal mittente e non da qualcun altro.

Debian GNU/Linux offre un certo numero di client e-mail, dotati di funzioni integrate per la firma delle e-mail, che interagiscono sia con gnupg, che con pgp:

- evolution.
- mutt.
- kmail.
- icedove (rebranded version of Mozilla's Thunderbird) through the http://enigmail.mozdev.org/ plugin. This plugin is provided by the enigmail package.
- sylpheed: a seconda di come si evolve la versione stabile di questo pacchetto, potrebbe essere necessario usare sylpheed-claws, ovvero la sua *versione in sviluppo*.
- gnus, che, se installato con il pacchetto mailcrypt, è un'interfaccia di **emacs** per **gnupg**.
- kuvert, che offre questa funzionalità, indipendentemente dall'agente di posta dell'utente (MUA), mediante l'interazione con l'agente di trasporto della posta (MTA).

Key servers allow you to download published public keys so that you may verify signatures. One such key server is http://wwwkeys.pgp.net. gnupg can automatically fetch public keys that are not already in your public keyring. For example, to configure **gnupg** to use the above key server, edit the file `~/.gnupg/options` and add the following line: [4]

```
keyserver wwwkeys.pgp.net
```

La maggior parte dei server di chiavi è collegata, in modo che quando viene aggiunta su uno di essi una chiave pubblica, tale aggiunta viene riprodotta su tutti gli altri. C'è anche un pacchetto Debian GNU/Linux, debian-keyring, che fornisce tutte le chiavi pubbliche degli sviluppatori Debian. I portachiavi di **gnupg** vengono installati nella cartella `/usr/share/keyrings/`.

Per maggiori informazioni:

- http://www.gnupg.org/faq.html.
- http://www.gnupg.org/gph/en/manual.html.

[3] Actually, there is an installer package for the *F-prot* antivirus, which is non-free but *gratis* for home users, called **f-prot-installer**. This installer, however, just downloads http://www.f-prot.com/products/home_use/linux/ and installs it in the system.
[4] For more examples of how to configure **gnupg** check `/usr/share/doc/mutt/examples/gpg.rc`.

- https://web.archive.org/web/20080201103530/http://www.dewinter.com/gnupg_howto/english/GPGMiniHowto.html.
- https://web.archive.org/web/20080513095235/http://www.uk.pgp.net/pgpnet/pgp-faq/.
- https://web.archive.org/web/20060222110131/http://www.cryptnet.net/fdp/crypto/gpg-party.html.

Capitolo 9. Linee guida consigliate agli sviluppatori per la sicurezza del sistema operativo

This chapter introduces some best secure coding practices for developers writing Debian packages. If you are really interested in secure coding I recommend you read David Wheeler's http://www.dwheeler.com/secure-programs/ and http://www.securecoding.org by Mark G. Graff and Kenneth R. van Wyk (O'Reilly, 2003).

Tecniche raccomandate per i controlli sulla sicurezza del software e sulla progettazione di software sicuro

Gli sviluppatori che creano dei pacchetti debian dovrebbero fare del loro meglio per assicurarsi che l'uso o l'installazione del software di cui stanno creando il pacchetto Debian non introduca delle falle di sicurezza sul sistema in cui viene installato e non comporti dei rischi di sicurezza per i suoi utenti.

In order to do so, they should make their best to review the source code of the package and detect any flaws that might introduce security bugs before releasing the software or distributing a new version. It is acknowledged that the cost of fixing bugs grows for different stages of its development, so it is easier (and cheaper) to fix bugs when designing than when the software has been deployed and is in maintenance mode (some studies say that the cost in this later phase is *sixty* times higher). Although there are some tools that try to automatically detect these flaws, developers should strive to learn about the different kind of security flaws in order to understand them and be able to spot them in the code they (or others) have written.

The programming bugs which lead to security bugs typically include: http://en.wikipedia.org/wiki/Buffer_overflow, format string overflows, heap overflows and integer overflows (in C/C++ programs), temporary http://en.wikipedia.org/wiki/Symlink_race (in scripts), http://en.wikipedia.org/wiki/Directory_traversal and command injection (in servers) and http://en.wikipedia.org/wiki/Cross_site_scripting, and http://en.wikipedia.org/wiki/SQL_injection (in the case of web-oriented applications). For a more complete information on security bugs review Fortify's http://vulncat.fortifysoftware.com/.

Some of these issues might not be easy to spot unless you are an expert in the programming language the software uses, but some security problems are easy to detect and fix. For example, finding temporary race conditions due to misuse of temporary directories can easily be done just by running `grep -r "/tmp/" .`. Those calls can be reviewed and replace the hardcoded filenames using temporary directories to calls to either **mktemp** or **tempfile** in shell scripts, File::Temp(3perl) in Perl scripts, or tmpfile(3) in C/C++.

There are a set of tools available to assist to the security code review phase. These include rats, flawfinder and pscan. For more information, read the http://www.debian.org/security/audit/tools.

Se dovete pacchettizzare un software sarà bene che vi ricordiate di seguire le più comuni linee guida sulla sicurezza, tra cui:

- Affinché il software venga eseguito esclusivamente con i privilegi minimi indispensabili è necessario che:

    - The package does install binaries setuid or setgid. **Lintian** will warn of http://lintian.debian.org/reports/Tsetuid-binary.html, http://lintian.debian.org/reports/Tsetgid-binary.html and http://lintian.debian.org/reports/Tsetuid-gid-binary.html binaries.

    - I demoni contenuti nel pacchetto vengano eseguiti da un utente con scarsi privilegi (vedete sezione chiamata «Creazione di utenti e gruppi che verranno usati dai demoni»).

- Le attività programmate (cioè avviate da **cron**) non vengano eseguite dall'utente root o, se lo sono, non svolgano compiti complessi.

If you have to do any of the above make sure the programs that might run with higher privileges have been audited for security bugs. If you are unsure, or need help, contact the http://www.debian.org/security/audit/. In the case of setuid/setgid binaries, follow the Debian policy section regarding http://www.debian.org/doc/debian-policy/ch-files.html#s10.9

For more information, specific to secure programming, make sure you read (or point your upstream to) http://www.dwheeler.com/secure-programs/ and the https://buildsecurityin.us-cert.gov/portal/ portal.

Creazione di utenti e gruppi che verranno usati dai demoni

If your software runs a daemon that does not need root privileges, you need to create a user for it. There are two kind of Debian users that can be used by packages: static uids (assigned by base-passwd, for a list of static users in Debian see sezione chiamata «Utenti e gruppi del sistema operativo») and dynamic uids in the range assigned to system users.

In the first case, you need to ask for a user or group id to the base-passwd. Once the user is available there the package needs to be distributed including a proper versioned depends to the base-passwd package.

In the second case, you need to create the system user either in the *preinst* or in the *postinst* and make the package depend on `adduser (>= 3.11)`.

Il seguente codice di esempio crea l'utente ed il gruppo che verranno usati dal demone quando il pacchetto verrà installato o aggiornato:

```
[...]
case "$1" in
  install|upgrade)

  # If the package has default file it could be sourced, so that
  # the local admin can overwrite the defaults

  [ -f "/etc/default/packagename" ] && . /etc/default/packagename

  # Sane defaults:

  [ -z "$SERVER_HOME" ] && SERVER_HOME=server_dir
  [ -z "$SERVER_USER" ] && SERVER_USER=server_user
  [ -z "$SERVER_NAME" ] && SERVER_NAME="Server description"
  [ -z "$SERVER_GROUP" ] && SERVER_GROUP=server_group

  # Groups that the user will be added to, if undefined, then none.
```

```
  ADDGROUP=""

  # create user to avoid running server as root
  # 1. create group if not existing
  if ! getent group | grep -q "^$SERVER_GROUP:" ; then
     echo -n "Adding group $SERVER_GROUP.."
     addgroup --quiet --system $SERVER_GROUP 2>/dev/null ||true
     echo "..done"
  fi
  # 2. create homedir if not existing
  test -d $SERVER_HOME || mkdir $SERVER_HOME
  # 3. create user if not existing
  if ! getent passwd | grep -q "^$SERVER_USER:"; then
    echo -n "Adding system user $SERVER_USER.."
    adduser --quiet \
            --system \
            --ingroup $SERVER_GROUP \
            --no-create-home \
            --disabled-password \
            $SERVER_USER 2>/dev/null || true
    echo "..done"
  fi
  # 4. adjust passwd entry
  usermod -c "$SERVER_NAME" \
          -d $SERVER_HOME   \
          -g $SERVER_GROUP  \
             $SERVER_USER
  # 5. adjust file and directory permissions
  if ! dpkg-statoverride --list $SERVER_HOME >/dev/null
  then
      chown -R $SERVER_USER:adm $SERVER_HOME
      chmod u=rwx,g=rxs,o= $SERVER_HOME
  fi
  # 6. Add the user to the ADDGROUP group
  if test -n $ADDGROUP
  then
      if ! groups $SERVER_USER | cut -d: -f2 | \
         grep -qw $ADDGROUP; then
           adduser $SERVER_USER $ADDGROUP
      fi
  fi
  ;;
  configure)

[...]
```

Dovete assicurarvi che il file di script init.d:

- Starts the daemon dropping privileges: if the software does not do the setuid(2) or seteuid(2) call itself, you can use the `--chuid` call of **start-stop-daemon**.
- Fermi il demone solo se viene riscontrata una corrispondenza con l'id dell'utente, a questo scopo, è possibile utilizzare l' opzione **start-stop-daemon** `--user`.
- Non venga eseguito se l'utente o il gruppo non esistono:

```
if ! getent passwd | grep -q "^server_user:"; then
   echo "Server user does not exist. Aborting" >&2
   exit 1
fi
if ! getent group | grep -q "^server_group:" ; then
   echo "Server group does not exist. Aborting" >&2
   exit 1
fi
```

If the package creates the system user it can remove it when it is purged in its *postrm*. This has some drawbacks, however. For example, files created by it will be orphaned and might be taken over by a new system user in the future if it is assigned the same uid[1]. Consequently, removing system users on purge is not yet mandatory and depends on the package needs. If unsure, this action could be handled by asking the administrator for the prefered action when the package is installed (i.e. through **debconf**).

Maintainers that want to remove users in their *postrm* scripts are referred to the **deluser**/**deluser** `--system` option.

Running programs with a user with limited privileges makes sure that any security issue will not be able to damage the full system. It also follows the principle of *least privilege*. Also consider you can limit privileges in programs through other mechanisms besides running as non-root[2]. For more information, read the http://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/minimize-privileges.html chapter of the *Secure Programming for Linux and Unix HOWTO* book.

[1] Some relevant threads discussing these drawbacks include http://lists.debian.org/debian-mentors/2004/10/msg00338.html and http://lists.debian.org/debian-devel/2004/05/msg01156.html
[2] You can even provide a SELinux policy for it

Capitolo 10. Prima della compromissione

Mantenere sicuro il proprio sistema

Dovreste sempre sforzarvi di mantenere il vostro sistema sicuro controllandone l'uso che ne viene fatto e le sue possibili vulnerabilità; tali vulnerabilità andrebbero poi eliminate installando gli aggiornamenti di sicurezza non appena sono disponibili. Anche se siete convinti della robustezza del vostro sistema appena installato, va ricordato che la sicurezza di un sistema si degrada nel tempo, poiché potrebbero venire scoperte nuove vulnerabilità nei servizi offerti dal proprio sistema agli utenti, anche questi ultimi potrebbero compromettere la sicurezza del sistema per scarse cognizioni tecniche (ad esempio, potrebbero accedere al sistema da remoto usando un protocollo di comunicazione non cifrato, o usando delle password banali e dunque facili da indovinare) o perché mirano esplicitamente a compromettere la sicurezza del sistema (per esempio, installano dei servizi aggiuntivi sul sistema usando le risorse del sistema che sono state loro concesse).

Mantenersi aggiornati sulle vulnerabilità di sicurezza

Sebbene molti amministratori di sistema vengano a conoscenza di una vulnerabilità del sistema solo quando un aggiornamento diventa disponibile, sarebbe meglio venire a conoscenza delle vulnerabilità di sicurezza il prima possibile, per prevenire gli attacchi, introducendo delle contromisure di emergenza. Ciò è particolarmente utile nel caso in cui stiate amministrando un sistema esposto (ossia connesso ad Internet) che fornisce un servizio. In tal caso gli amministratori di sistema dovrebbero controllare regolarmente delle fonti di informazioni fidate relative alla sicurezza, così da venire a conoscenza il prima possibile se un servizio critico del sistema (o che comunque potrebbe avere degli effetti indesiderati su di esso) è affetto da vulnerabilità.

This typically includes subscribing to the announcement mailing lists, project websites or bug tracking systems provided by the software developers for a specific piece of code. For example, Apache users should regularly review Apache's http://httpd.apache.org/security_report.html and subscribe to the http://httpd.apache.org/lists.html#http-announce mailing list.

In order to track known vulnerabilities affecting the Debian distribution, the Debian Testing Security Team provides a https://security-tracker.debian.org/ that lists all the known vulnerabilities which have not been yet fixed in Debian packages. The information in that tracker is obtained through different public channels and includes known vulnerabilities which are available either through security vulnerability databases or http://www.debian.org/Bugs/. Administrators can search for the known security issues being tracked for https://security-tracker.debian.org/tracker/status/release/stable, https://security-tracker.debian.org/tracker/status/release/oldstable, https://security-tracker.debian.org/tracker/status/release/testing, or https://security-tracker.debian.org/tracker/status/release/unstable.

The tracker has searchable interfaces (by http://cve.mitre.org/ name and package name) and some tools (such as debsecan, see sezione chiamata «Individuazione automatica dei problemi di sicurezza con debsecan») use that database to provide information of vulnerabilities affecting a given system which have not yet been addressed (i.e. those who are pending a fix).

Un amministratore di sistema scrupoloso dovrebbe usare tali informazioni per determinare le falle nella sicurezza del sistema che sta amministrando, determinare la pericolosità di tali falle ed applicare (se possibile) delle contromisure temporanee fintantoché non venga rilasciato un aggiornamento di sicurezza che risolva il problema.

I problemi relativi alla sicurezza, individuati per le versioni di Debian di cui il Team Debian per la Sicurezza si occupa ancora, verranno pubblicati negli avvisi Debian per la sicurezza e i conseguenti aggiornamenti saranno disponibili per tutti gli utenti (vedete al riguardo sezione chiamata «Aggiornare continuamente il sistema»). Una volta che gli aggiornamenti di sicurezza vengono pubblicati, l'avviso corrispondente viene cancellato dall'archivio, ma è ancora possibile ottenere dettagli sulla vulnerabilità di sicurezza, identificata col suo nome CVE, usando la http://www.debian.org/security/crossreferences che collega gli avvisi Debian ai dati sulle vulnerabilità forniti dal CVE.

Notate, comunque, che il Team per la Sicurezza di Debian Testing gestisce solo vulnerabilità note al pubblico. In alcune occasione invece il Team Debian per la Sicurezza (fate attenzione che sono due entità separate) potrebbe preparare degli Avvisi Debian sulla Sicurezza (l'acronimo inglese di tali documenti è DSA) in base ad informazioni riservate (che gli sono state fornite ad esempio da mailing list private del produttore o da alcuni dei manutentori del software in via confidenziale). Pertanto alcuni problemi di sicurezza, sebbene descritti e risolti tramite un DSA, potrebbero non essere presenti nell'archivio.

Aggiornare continuamente il sistema

You should conduct security updates frequently. The vast majority of exploits result from known vulnerabilities that have not been patched in time, as this http://www.cs.umd.edu/~waa/vulnerability.html (presented at the 2001 IEEE Symposium on Security and Privacy) explains. Updates are described under sezione chiamata «Eseguire un aggiornamento per la sicurezza».

Controllo manuale degli aggiornamenti di sicurezza disponibili

Debian ha uno strumento apposito per verificare se un sistema deve essere aggiornato, ma molti utenti vorranno semplicemente controllare manualmente se sono disponibili aggiornamenti di sicurezza per il loro sistema.

Se avrete configurato il sistema come descritto in sezione chiamata «Eseguire un aggiornamento per la sicurezza», basterà dare il comando:

```
# apt-get update
# apt-get upgrade -s
[ ... review packages to be upgraded ... ]
# apt-get upgrade
# checkrestart
[ ... restart services that need to be restarted ... ]
```

e riavviare quei servizi le cui librerie sono state aggiornate. Notate: leggete in sezione chiamata «Eseguire un aggiornamento per la sicurezza» per altre informazioni sugli aggiornamenti delle librerie (e del kernel).

La prima riga scaricherà l'elenco dei pacchetti disponibili tra quelli presenti sul sistema e configurati. L'opzione `-s` simulerà l'esecuzione, cioè *non* scaricherà o installerà i pacchetti, ma piuttosto, comunicherà quali dovrebbero essere scaricati/installati. Dal risultato si potrà dedurre quali pacchetti siano stati corretti da Debian e siano disponibili come aggiornamento di sicurezza. Per esempio:

```
# apt-get upgrade -s
Reading Package Lists... Done
Building Dependency Tree... Done
2 packages upgraded, 0 newly installed, 0 to remove and 0  not upgraded.
Inst cvs (1.11.1p1debian-8.1 Debian-Security:3.0/stable)
Inst libcupsys2 (1.1.14-4.4 Debian-Security:3.0/stable)
Conf cvs (1.11.1p1debian-8.1 Debian-Security:3.0/stable)
```

```
Conf libcupsys2 (1.1.14-4.4 Debian-Security:3.0/stable)
```

In this example, you can see that the system needs to be updated with new cvs and cupsys packages which are being retrieved from *woody's* security update archive. If you want to understand why these packages are needed, you should go to http://security.debian.org and check which recent Debian Security Advisories have been published related to these packages. In this case, the related DSAs are https://lists.debian.org/debian-security-announce/2003/msg00014.html (for cvs) and https://lists.debian.org/debian-security-announce/2003/msg00013.html (for cupsys).

Notate che il sistema dovrà essere riavviato nel caso vi sia stato un aggiornamento del kernel.

Controllo degli aggiornamenti dal Desktop

Since Debian 4.0 *lenny* Debian provides and installs in a default installation update-notifier. This is a GNOME application that will startup when you enter your Desktop and can be used to keep track of updates available for your system and install them. It uses update-manager for this.

In un sistema stabile gli aggiornamenti sono disponibili solo quando viene rilasciata una patch di sicurezza o al momento dei nuovi rilasci. Di conseguenza, se il sistema è configurato per ricevere gli aggiornamenti di sicurezza, come descritto in sezione chiamata «Eseguire un aggiornamento per la sicurezza» e c'è un'istanza di cron che aggiorna le informazioni dei pacchetti, verrete informati attraverso un'icona dell'area di notifica del desktop.

La notifica non è intrusiva e gli utenti non sono obbligati ad installare gli aggiornamenti. Dall'icona di notifica un utente del desktop (con la password di amministratore) può accedere ad una semplice interfaccia grafica per vedere quali sono gli aggiornamenti disponibili ed installarli.

Questa applicazione funziona controllando il database dei pacchetti e comparando il sistema con il suo contenuto. Se il database dei pacchetti viene aggiornato periodicamente attraverso un'istanza di **cron**, allora il contenuto del database sarà più nuovo dei pacchetti installati nel sistema e quindi una applicazione lo notificherà.

Apt installa un compito (`/etc/cron.d/apt`) che verrà eseguito a seconda della configurazione di Apt (più nello specifico *APT::Periodic*). Nell'ambiente GNOME questo valore di configurazione può essere modificato in Sistema > Amministrazione > Sorgenti Software > Aggiornamenti, o eseguendo **/usr/bin/software-properties**.

Se il sistema è configurato per scaricare la lista dei pacchetti quotidianamente ma non per scaricare i pacchetti stessi, il file `/etc/apt/apt.conf.d/10periodic` dovrebbe avere questo aspetto:

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "0";
```

You can use a different cron task, such as the one installed by cron-apt (see sezione chiamata «Controllo automatico degli aggiornamenti con cron-apt»). You can also just manually check for upgrades using this application.

Users of the KDE desktop environment will probably prefer to install adept and adept-notifier instead which offers a similar functionality but is not part of the standard installation.

Controllo automatico degli aggiornamenti con cron-apt

Another method for automatic security updates is the use of cron-apt. This package provides a tool to update the system at regular intervals (using a cron job), and can also be configured to send mails to the

system administrator using the local mail transport agent. It will just update the package list and download new packages by default but it can be configured to automatically install new updates.

Se volete aggiornare automaticamente il vostro sistema (anche solo scaricandone i pacchetti), controllate la versione della distribuzione, come descritto in sezione chiamata «Controllo di rilascio per ogni distribuzione»; in mancanza di questo controllo, non potrete essere sicuri che i pacchetti provengano da una fonte fidata.

Ulteriori informazioni sono disponibili presso il sito http://www.debian-administration.org/articles/162.

Individuazione automatica dei problemi di sicurezza con debsecan

The **debsecan** program evaluates the security status of by reporting both missing security updates and security vulnerabilities. Unlike cron-apt, which only provides information related to security updates available, but this tool obtains information from the security vulnerability database maintained by the Debian Security Team which includes also information on vulnerabilities which are not yet fixed through a security update. Consequently, it is more efficient at helping administrators track security vulnerabilities (as described in sezione chiamata «Mantenersi aggiornati sulle vulnerabilità di sicurezza»).

Upon installing the Debian package debsecan, and if the administrator consents to it, it will generate a cron task that will make it run and send the output to a specific user whenever it finds a vulnerable package. It will also download the information from the Internet. The location of the security database is also part of the questions ask on installation and are later defined `/etc/default/debsecan`, it can be easily adjusted for systems that do not have Internet access so that they all pull from a local mirror so that there is a single point that access the vulnerability database.

Notate, comunque, che il Team Debian per la sicurezza esamina molte vulnerabilità, incluse quelle a basso rischio che potrebbero essere ignorate e non essere mai risolte tramite un aggiornamento di sicurezza; alcune vulnerabilità che inizialmente sono state identificate come riguardanti anche Debian, potrebbero, in futuro, dopo ulteriori accertamenti, essere invece completamente irrilevanti per Debian. **Debsecan** elenca tutte queste vulnerabilità, tendendo a fornire dei risultati molto più lunghi da leggere degli altri strumenti descritti in precedenza.

More information is available at the http://www.enyo.de/fw/software/debsecan/.

Altri metodi per effettuare aggiornamenti per la sicurezza

There is also the apticron, which, similarly to cron-apt will check for updates and send mails to the administrator. More information on apticron is available at the http://www.debian-administration.org/articles/491.

You might also want to take a look at http://clemens.endorphin.org/secpack/ which is an unofficial program to do security updates from security.debian.org with signature checking written by Fruhwirth Clemens. Or to the Nagios Plugin http://www.unixdaemon.net/nagios_plugins.html#check_debian_packages written by Dean Wilson.

Evitare di usare il ramo instabile

Unless you want to dedicate time to patch packages yourself when a vulnerability arises, you should *not* use Debian's unstable branch for production-level systems. The main reason for this is that there are no security updates for *unstable*.

Il fatto è che alcuni problemi di sicurezza potrebbero apparire nella distribuzione *unstable* e *non* nella *stable*. Questo è dovuto alle nuove funzionalità costantemente aggiunte alle applicazioni lì fornite, come anche a nuove applicazioni che vengono incluse senza aver passato un collaudo approfondito.

Per quanto riguarda l'eseguire aggiornamenti di sicurezza nel ramo *unstable*, dovreste aggiornare completamente alle nuove versioni (che vengono aggiornate più dei pacchetti in questione). Sebbene ci siano alcune eccezioni, solamente le patch di sicurezza vengono riportate nel ramo *stable*. L'idea di fondo è che tra gli aggiornamenti non venga aggiunto *nuovo codice*, ma che vengano solamente risolti i problemi importanti per la sicurezza.

Notate che, comunque potete ancora usare l'archivio delle vulnerabilità di sicurezza (come descritto in sezione chiamata «Mantenersi aggiornati sulle vulnerabilità di sicurezza») per scoprire quali vulnerabilità riguardino la versione di Debian che state usando.

Supporto alla sicurezza per il ramo testing

Se usate il ramo *testing*, ci sono alcuni aspetti da tenere in considerazione sulla disponibilità degli aggiornamenti di sicurezza:

- Quando è pronta una correzione di sicurezza, il Team per la Sicurezza appronta il backport della patch a *stable* (in quanto stable di solito è indietro di qualche versione, minore o maggiore). I manutentori del pacchetto sono responsabili della preparazione dei pacchetti per il ramo *unstable*, di solito basati su una letteralmente nuova versione. A volte le modifiche avvengono quasi in contemporanea e altre volte uno dei due rami riceve la correzione di sicurezza prima dell'altro. I pacchetti per la distribuzione *stable* sono soggetti ad un controllo più approfondito di quelli per *unstable*, in quanto quest'ultima contiene perlopiù la versione più recente (che potrebbe includere nuovi bug ancora sconosciuti).

- Gli aggiornamenti di sicurezza sono normalmente disponibili per il ramo *unstable* quando il manutentore del pacchetto crea un nuovo pacchetto e per il branch *stable* quando il Team per la Sicurezza fa un nuovo caricamento e pubblica un DSA. Notate che nessuno di questi due cambia il ramo di *testing*.

- Se nessun (nuovo) errore viene scoperto nella versione *unstable* del pacchetto, questo si sposta nel ramo *testing* dopo diversi giorni. Il tempo che occorre per questo procedimento di solito è dieci giorni, anche se dipende dalla priorità di caricamento del cambiamento e se il pacchetto viene bloccato dall'ingresso nel ramo *testing* dalle sue relazioni di dipendenza. Notate che se il pacchetto viene bloccato dall'entrata in *testing* la priorità di caricamento non modificherà il tempo che gli occorre per entrare.

Questo comportamento può cambiare in base allo stato di rilascio della distribuzione. Quando un rilascio è quasi imminente, il Team per la Sicurezza o i manutentori del pacchetto possono fornire degli aggiornamenti direttamente al ramo *testing*.

Additionally, the http://secure-testing-master.debian.net can issue Debian Testing Security Advisories (DTSAs) for packages in the *testing* branch if there is an immediate need to fix a security issue in that branch and cannot wait for the normal procedure (or the normal procedure is being blocked by some other packages).

Gli utenti che volessero usufruire di questo supporto devono aggiungere le seguenti righe al loro `/etc/apt/sources.list` (invece delle righe descritte in sezione chiamata «Eseguire un aggiornamento per la sicurezza»):

```
    deb http://security.debian.org testing/updates main contrib non-free
# This line makes it possible to donwload source packages too
    deb-src  http://security.debian.org testing/updates main contrib non-free
```

For additional information on this support please read the http://lists.debian.org/debian-devel-announce/2006/05/msg00006.html. This support officially started in http://lists.debian.org/debian-devel-announce/2005/09/msg00006.html in a separate repository and was later integrated into the main security archive.

Aggiornamento automatico in un sistema Debian GNU/Linux

Per cominciare, gli aggiornamenti automatici non sono del tutto consigliabili, visto che gli amministratori dovrebbero leggere gli annunci DSA e comprendere l'impatto di ogni aggiornamento di sicurezza.

Se volete aggiornare automaticamente il vostro sistema occorre:

- Configure **apt** so that those packages that you do not want to update stay at their current version, either with **apt**'s *pinning* feature or marking them as *hold* with **aptitude** or **dpkg**.

  To pin the packages under a given release, you must edit `/etc/apt/preferences` (see apt_preferences(5)) and add:

  ```
  Package: *
  Pin: release a=stable
  Pin-Priority: 100
  ```

  FIXME: verificare se questa configurazione è corretta.

- Either use cron-apt as described in sezione chiamata «Controllo automatico degli aggiornamenti con cron-apt» and enable it to install downloaded packages or add a **cron** entry yourself so that the update is run daily, for example:

  ```
  apt-get update && apt-get -y upgrade
  ```

  L'opzione `-y` farà in modo che **apt** risponda automaticamente 'yes' a tutte le domande che possono essere poste durante l'aggiornamento. In alcuni casi può essere preferibile usare l'opzione `--trivial-only` invece di quella `--assume-yes` (equivalente a `-y`)[1].

- Configurare **debconf** in modo che non ponga nessuna domanda durante l'aggiornamento; in questo modo l'aggiornamento non è interattivo[2].

- Controllare i risultati dell'esecuzione di **cron**, che verranno spediti al superutente (a meno che **cron** non sia stato configurato diversamente con la variabile `MAILTO` nell'apposito script).

Un'alternativa più sicura potrebbe essere usare l'opzione `-d` (o `--download-only`), che scaricherà ma non installerà i pacchetti necessari. L'aggiornamento verrà eseguito manualmente se l'esecuzione di **cron** mostrerà che il sistema deve essere aggiornato.

Per eseguire questi compiti, il sistema deve essere propriamente configurato per scaricare gli aggiornamenti di sicurezza come visto in sezione chiamata «Eseguire un aggiornamento per la sicurezza».

Ad ogni modo questo procedimento non è consigliabile per *unstable*, senza prima aver effettuato un'accurata analisi, perché potrebbe rendere il sistema inusabile se qualche bug pericoloso si insinuasse in un pacchetto importante e venisse installato nel sistema. *Testing* è un po' più *sicura* da questo punto di vista, dal momento che le possibilità di scoprire i bug più gravi prima che il pacchetto sia inserito in testing sono maggiori (tuttavia potreste *non* avere alcun aggiornamento di sicurezza disponibile, in questo caso).

[1] Potreste anche utilizzare l'opzione `--quiet` (`-q`) per ridurre l'output di **apt-get**, in modo da non mostrare alcun output se non vengono installati pacchetti.

[2] Bisogna ricordare che alcuni pacchetti potrebbero *non* utilizzare **debconf** e l'aggiornamento potrebbe bloccarsi a causa dei pacchetti che richiedono un input da parte dell'utente durante la configurazione.

Se avete una distribuzione mista, cioè una distribuzione *stable* con alcuni pacchetti presi da *testing* o *unstable*, potete utilizzare il sistema del pinning o l'opzione `--target-release` di **apt-get** per aggiornare *solo* quei pacchetti che hanno subito variazioni[3].

Effettuate periodicamente dei controlli sull'integrità del sistema

Basandovi sulle informazioni che avete generato in fase di installazione come metro di paragone, (ovvero l'istantanea descritta in sezione chiamata «Una fotografia del sistema») dovreste essere in grado di controllare l'integrità del sistema quando volete. Un controllo sull'integrità del sistema sarà in grado di rilevare le modifiche su disco, effettuate da un intruso o generate da errori dell'amministratore di sistema.

Integrity checks should be, if possible, done offline.[4] That is, without using the operating system of the system to review, in order to avoid a false sense of security (i.e. false negatives) produced by, for example, installed rootkits. The integrity database that the system is checked against should also be used from read-only media.

Potreste pensare di voler fare dei controlli d'integrità a sistema avviato (ossia, utilizzando il sistema operativo che state controllando) usando uno qualsiasi dei sistemi di controllo dell'integrità del filesystem disponibili, (descritti in sezione chiamata «Controllare l'integrità del file system») se non avete la possibilità di spengere il sistema. Comunque sia, dovreste assicurarvi che la banca dati usata nel controllo dell'integrità del filesystem non sia scrivibile e che gli strumenti di analisi che state usando (incluso il nucleo del sistema operativo) non siano stati manomessi.

Alcuni degli strumenti sono stati menzionati nella sezione sugli strumenti per il controllo dell'integrità, ad esempio **aide**, **integrit** o **samhain** sono già predisposti per effettuare periodicamente dei controlli (nei primi due casi si usa il crontab mentre **samhain** utilizza un demone indipendente) e possono avvisare l'amministratore in vari modi (in genere tramite email, ma **samhain** può anche inviare pagine web, avvisi mediante syslog o notifiche SNMP) qualora il filesystem sia stato alterato.

Naturalmente, se eseguite un aggiornamento per la sicurezza, dovrete effettuare una nuova istantanea del sistema che fotografi anche i cambiamenti legittimi avvenuti in seguito ad un aggiornamento del sistema per la sicurezza.

Pianificare la ricerca di intrusi

In Debian GNU/Linux sono presenti molti programmi che servono ad individuare intrusi nel sistema, possono scovare delle attività malevole sul vostro sistema personale, oppure negli altri sistemi della vostra rete. Questo tipo di difesa è importante sia che nel sistema siano residenti informazioni riservate, sia che voi siate veramente paranoici in fatto di sicurezza. I più comuni metodi per individuare degli intrusi sono l'individuazione di anomalie e la ricerca mediante l'uso di espressioni regolari.

Dovete essere consapevoli che la sicurezza del sistema viene migliorata con l'introduzione di questi programmi, avrete bisogno di avere un meccanismo di allerta e risposta configurato correttamente. La ricerca di intrusi senza un valido sistema di allerta diviene completamente inutile.

Quando viene scoperto un particolare attacco, molti di questi programmi vengono configurati per inviare un log con **syslogd** oppure per inviare una email all'amministratore (le intestazioni delle email sono soli-

[3] Questo è un problema comune visto che molti utenti vogliono utilizzare un sistema stable e prendere solo alcuni pacchetti da *unstable* per disporre di funzionalità più recenti. Questo bisogno nasce dal fatto che alcuni progetti evolvono più rapidamente del tempo che passa tra due versioni *stable* di Debian.

[4] An easy way to do this is using a Live CD, such as http://www.knoppix-std.org/ which includes both the file integrity tools and the integrity database for your system.

tamente configurabili). Un amministratore può accuratamente configurare questi strumenti evitando così di ricevere allarmi per falsi positivi. Inoltre è necessario fare attenzione ai sistemi di allarme dei tentativi di intrusione, potrebbero rivelarsi inutili se vengono generati il giorno dopo che l'attacco è avvenuto. Siamo sicuri che questa sia la politica di sicurezza migliore, è però importante che gli strumenti per migliorare questa politica siano implementati.

An interesting source of information is http://www.cert.org/tech_tips/intruder_detection_checklist.html

Individuazione delle intrusioni sulla rete

Gli strumenti che controllano le intrusioni lo fanno sul traffico di un segmento di rete e usano le informazioni come una sorgente di dati. Specificatamente, vengono esaminati i pacchetti in rete e viene controllato che abbiano un certificato valido.

snort is a flexible packet sniffer or logger that detects attacks using an attack signature dictionary. It detects a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, and much more. **snort** also has real-time alerting capability. You can use **snort** for a range of hosts on your network as well as for your own host. This is a tool which should be installed on every router to keep an eye on your network. Just install it with `apt-get install snort`, follow the questions, and watch it log. For a little broader security framework, see http://www.prelude-ids.org.

Debian's snort package has many security checks enabled by default. However, you should customize the setup to take into account the particular services you run on your system. You may also want to seek additional checks specific to these services.

There are other, simpler tools that can be used to detect network attacks. portsentry is an interesting package that can tip you off to port scans against your hosts. Other tools like ippl or iplogger will also detect some IP (TCP and ICMP) attacks, even if they do not provide the kind of advanced techniques **snort** does.

You can test any of these tools with the Debian package idswakeup, a shell script which generates false alarms, and includes many common attack signatures.

Sistemi per individuare gli intrusi

I sistemi per individuare gli intrusi controllano chi usa i file di log e/o i sistemi di verifica come se fossero una sorgente dati. Controllano i processi sospetti, l'accesso al sistema e possono riportare dei cambiamenti ai file fondamentali per il sistema.

tiger is an older intrusion detection tool which has been ported to Debian since the Woody branch. **tiger** provides checks of common issues related to security break-ins, like password strength, file system problems, communicating processes, and other ways root might be compromised. This package includes new Debian-specific security checks including: MD5sums checks of installed files, locations of files not belonging to packages, and analysis of local listening processes. The default installation sets up **tiger** to run each day, generating a report that is sent to the superuser about possible compromises of the system.

Log analysis tools, such as logcheck can also be used to detect intrusion attempts. See sezione chiamata «Usare e personalizzare **logcheck**».

Inoltre esistono programmi che controllano l'integrita dei filesystem (vedete in sezione chiamata «Controllare l'integrità del file system») che sono abbastanza utili nella ricerca di anomalie in un ambiente sicuro. È molto probabile che un vero intruso modifichi alcuni file nel filesystem locale, allo scopo di aggirare la politica di sicurezza, installare dei cavalli di Troia, oppure creare utenti. Questi eventi vengono ricercati dai programmi atti a controllare l'integrità dei filesystem.

Evitare i root-kit

Moduli del kernel caricabili (LKM)

I moduli caricabili del kernel sono file che contengono componenti del kernel per espanderne le funzionalità, caricabili in modo dinamico. Il principale vantaggio nel loro impiego sta nella possibilità di aggiungere ulteriori dispositivi, come per una scheda sonora o una Ethernet, senza apportare correzioni al sorgente del kernel e senza ricompilarlo interamente. Però, al momento, i cracker li sfruttano per i loro root-kit (usurpando l'account di root (knark e adore)) per aprire porte all'insaputa dell'amministratore (le cosiddette "back door") nei sistemi GNU/Linux.

Le porte segrete aperte tramite LKM sono più sofisticate e meno rilevabili rispetto ai tradizionali root-kit. Possono nascondere processi, file, cartelle e perfino connessioni, senza modificare il codice sorgente dei binari. Per esempio, un LKM maligno può obbligare il kernel a nascondere processi specifici da `procfs`, cosicché una copia del binario **ps**, ritenuta fedele, non fornirebbe, invece, informazioni precise sugli attuali processi in atto nel sistema.

Scoprire i root-kit

Ci sono due approcci per difendere il sistema contro i root-kit a mezzo LKM: la difesa preventiva e quella reattiva. Il lavoro di ricerca può essere semplice e indolore, o difficile e faticoso, a seconda dell'approccio.

Difesa preventiva

Il vantaggio di questo tipo di difesa è che in primo luogo previene danni al sistema. Una siffatta strategia consiste nel motto *arrivateci per primi*, cioè, caricare un LKM atto a proteggere il sistema da altri LKM malevoli. Una seconda strategia è quella di rimuovere completamente la possibilità che il kernel possa caricare dei moduli. Notate, comunque, che esistono rootkit che potrebbero funzionare anche in questo caso, ce ne sono alcuni che manomettono direttamente `/dev/kmem` (la memoria del kernel) per non essere scoperti.

Debian GNU/Linux ha alcuni pacchetti che possono essere utilizzati per realizzare una difesa preventiva:

lcap - A user friendly interface to remove *capabilities* (kernel-based access control) in the kernel, making the system more secure. For example, executing `lcap CAP_SYS_MODULE`[5] will remove module loading capabilities (even for the root user).[6] There is some (old) information on capabilities at Jon Corbet's http://lwn.net/1999/1202/kernel.php3 section on LWN (dated December 1999).

Se non avete affatto bisogno di molte delle caratteristiche del kernel sul vostro sistema GNU/Linux, potete disabilitare il supporto ai moduli caricabili durante la fase di configurazione del kernel stesso. Per disabilitare questo supporto, impostate CONFIG_MODULES=n durante la fase di configurazione per la compilazione del vostro kernel, oppure nel file `.config`. Questo annullerà i tentativi dei root-kit LKM, ma perderete questa potente caratteristica del kernel Linux. Inoltre, disabilitare il supporto per i moduli caricabili a volte potrebbe appesantire troppo il kernel, rendendo il supporto ai moduli indispensabile.

[5] There are over 28 capabilities including: `CAP_BSET`, `CAP_CHOWN`, `CAP_FOWNER`, `CAP_FSETID`, `CAP_FS_MASK`, `CAP_FULL_SET`, `CAP_INIT_EFF_SET`, `CAP_INIT_INH_SET`, `CAP_IPC_LOCK`, `CAP_IPC_OWNER`, `CAP_KILL`, `CAP_LEASE`, `CAP_LINUX_IMMUTABLE`, `CAP_MKNOD`, `CAP_NET_ADMIN`, `CAP_NET_BIND_SERVICE`, `CAP_NET_RAW`, `CAP_SETGID`, `CAP_SETPCAP`, `CAP_SETUID`, `CAP_SYS_ADMIN`, `CAP_SYS_BOOT`, `CAP_SYS_CHROOT`, `CAP_SYS_MODULE`, `CAP_SYS_NICE`, `CAP_SYS_PACCT`, `CAP_SYS_PTRACE`, `CAP_SYS_RAWIO`, `CAP_SYS_RESOURCE`, `CAP_SYS_TIME`, and `CAP_SYS_TTY_CONFIG`. All of them can be de-activated to harden your kernel.

[6] You don't need to install lcap to do this, but it's easier than setting `/proc/sys/kernel/cap-bound` by hand.

Difesa reattiva

Il vantaggio di una difesa reattiva è che non sovraccarica le risorse del sistema. Funziona confrontando la tabella delle chiamate di sistema con una copia sicura in un file del disco, `System.map`. Ovviamente, una difesa reattiva si limiterà ad avvisare l'amministratore di sistema dopo che il sistema è già stato compromesso.

Detection of some root-kits in Debian can be accomplished with the chkrootkit package. The http://www.chkrootkit.org program checks for signs of several known root-kits on the target system, but is not a definitive test.

Genius/Paranoia Ideas - what you could do

This is probably the most unstable and funny section, since I hope that some of the "duh, that sounds crazy" ideas might be realized. The following are just some ideas for increasing security - maybe genius, paranoid, crazy or inspired depending on your point of view.

- Divertirsi con i Pluggable Authentication Modules (PAM - moduli per l'autenticazione inseribili): come riportato nell'articolo su PAM, in Phrack 56, il loro aspetto più simpatico è che "l'unico limite è ciò che si riesce a pensare". Ed è vero! Immaginate un'autenticazione di root possibile solo mediante impronta digitale o scansione oculare o tessera magnetica cifrata (ma perché usare la congiunzione "O", invece che la "E"?).

- Registrazione dei log "fascista" - per contrasto verso tutte le precedenti discussioni sulla "registrazione dei log leggera": se si vuole una registrazione dei log degna di tal nome, basta spedire tutti i log ad una stampante con carta a modulo continuo. Sembra un espediente buffo, ma è sicuro ed al riparo da manomissioni e cancellazioni.

- Distribuzione su CD: idea davvero semplicissima da realizzarsi, dà una sicurezza abbastanza buona; basta creare una distribuzione Debian ben corazzata, con regole di firewall appropriate, convertirla in immagine ISO inizializzabile e schiaffarla su un CD-Rom, così da avere una distribuzione in sola lettura, con circa 600 MB di spazio per i vari servizi. Per gli intrusi è impossibile ottenere l'accesso al sistema in lettura e scrittura ma qualsiasi cambiamento essi riescano ad operare può essere annullato con la reinizializzazione del sistema.

- Disabilitare le funzionalità tramite modulo: come già visto, quando, durante la sua compilazione, si impedisce l'uso dei moduli del kernel, molte porte segrete basate su di essi non possono essere sfruttate, perché, nella maggioranza dei casi, si fondano sull'installazione di moduli del kernel modificati.

- Registrazione dei log tramite cavo seriale (collaborazione di Gaby Schilders): finché i server avranno porte seriali, si potrà dedicare un sistema per la registrazione dei log di un certo numero di server. Tale sistema sarà fuori dalla rete, ma connesso ai server per mezzo di una multipresa per porte seriali (Cyclades o simili). Ora che tutti i server registreranno verso le loro porte seriali, in sola scrittura, basterà connettere un masterizzatore di CD/DVD, sul quale trasferire i file di registrazione, fintantoché non giungano a riempire la capacità del supporto. Se solo facessero dei masterizzatori con autocommutatori...! Non è un modo di fare copie "dure" come la registrazione diretta a mezzo stampante, ma può gestirne volumi più consistenti - e stoccare CD-Rom richiede meno spazio.

- Cambiare gli attributi dei file usando **chattr** (tratto dal Tips-HOWTO, di Jim Dennis): dopo una semplice installazione ed una configurazione iniziale, potete usare il programma **chattr** con l'attributo `+i`, per far si che i file non possano essere modificati (ossia, cancellati, rinominati, collegati o riscritti). Da considerare la possibilità di impostare questo attributo su tutti i file delle cartelle `/bin`, `/sbin/`, `/usr/bin`, `/usr/sbin`, `/usr/lib` e sui file del kernel in root. Potreste anche copiare tutti i file in `/etc/`, con **tar** od un altro programma simile, e classificare l'archivio come immutabile.

This strategy will help limit the damage that you can do when logged in as root. You won't overwrite files with a stray redirection operator, and you won't make the system unusable with a stray space in a **rm -fr** command (you might still do plenty of damage to your data - but your libraries and binaries will be safer).

Inoltre questa strategia rende quasi impossibile, o perlomeno più difficile, sfruttare attacchi alla sicurezza del tipo denial of service (DoS - rifiuto di fornire servizi), miranti a sovrascrivere un file attraverso l'azione di qualche programma con SETUID attivo, perché *fornisca un'arbitraria shell di root*.

One inconvenience of this strategy arises during building and installing various system binaries. On the other hand, it prevents the **make install** from over-writing the files. When you forget to read the Makefile and **chattr -i** the files that are to be overwritten, (and the directories to which you want to add files) - the make command fails, and you just use the **chattr** command and rerun it. You can also take that opportunity to move your old bin's and libs out of the way, into a .old/ directory or tar archive for example.

Notate che questa strategia impedisce anche di aggiornare i pacchetti del proprio sistema, dal momento che i file forniti dai pacchetti dell'ultima versione non possono essere sovrascritti. A questo proposito, potreste creare uno script o un altro meccanismo per disabilitare il flag "immutabile" su tutti i binari, giusto prima di fare un **apt-get update**.

- Potreste voler alterare un cavo UTP tagliando 2 o 4 fili in maniera che il traffico di rete possa scorrere in una sola direzione. Successivamente potreste inviare ad un'altra macchina, tramite dei pacchetti UDP, i log di sistema.

Costruirsi una honeypot ("trappola al miele")

Una honeypot ("trappola al miele") è un sistema progettato per insegnare agli amministratori come i cracker sondano e sfruttano il sistema; è un modo di impostare un sistema con l'aspettativa e l'obiettivo che sia sondato, attaccato e potenzialmente, sfruttato. Apprendendo gli strumenti e le metodiche dei cracker, un amministratore di sistema impara a proteggere meglio i sistemi e le reti di cui si occupa.

Un sistema Debian GNU/Linux può essere agevolmente configurato come "trappola al miele", dedicando un po' di tempo ad implementare e controllare: basta impostare il falso server con un firewall[7] ed un qualsiasi rilevatore di intrusioni nella rete, collegarlo ad Internet e aspettare. In caso di sfruttamento del sistema, occorre essere ben certi di venire avvisati per tempo (vedete in sezione chiamata «L'importanza di log e avvisi»), sì da poter assumere le opportune contromisure e bloccare la compromissione quando se ne conosca abbastanza. Questo è un elenco di pacchetti e di aspetti da valutare durante l'impostazione di una honeypot:

- La tecnologia di firewall che si impiegherà (fornita dal kernel Linux).
- syslog-ng, useful for sending logs from the honeypot to a remote syslog server.
- snort, to set up capture of all the incoming network traffic to the honeypot and detect the attacks.
- osh, a SETUID root, security enhanced, restricted shell with logging (see Lance Spitzner's article below).
- Naturalmente, tutti i demoni che si useranno per il server-trappola. Analizzando la situazione, a seconda del tipo di attaccante, dovrete decidere se irrobustire la vostra trappola, se mantenere gli aggiornamenti con le patch per la sicurezza o se *non farlo*.
- Integrity checkers (see sezione chiamata «Controllare l'integrità del file system») and The Coroner's Toolkit (tct) to do post-attack audits.

[7] Di solito si usa un bridge firewall in modo tale che il firewall stesso non sia rilevabile, vedete in sezione chiamata «Impostare un bridge firewall».

- honeyd and farpd to setup a honeypot that will listen to connections to unused IP addresses and forward them to scripts simulating live services. Also check out iisemulator.

- tinyhoneypot to setup a simple honeypot server with fake services.

If you cannot use spare systems to build up the honeypots and the network systems to protect and control it you can use the virtualisation technology available in **xen** or **uml** (User-Mode-Linux). If you take this route you will need to patch your kernel with either kernel-patch-xen or kernel-patch-uml.

Maggiori informazioni sulla costruzione di honeypot si possono trovare nell'eccellente articolo di Lanze Spitzner http://www.net-security.org/text/articles/spitzner/honeypot.shtml, (costruire una honeypot), - della serie "Conosci i tuoi nemici". Inoltre, l'http://project.honeynet.org/ offre informazioni preziose sul modo in cui progettare queste trappole ed analizzare gli attacchi rivolti contro di esse.

Capitolo 11. Dopo la compromissione (reazione agli incidenti)

Come comportarsi, in generale

Se si è fisicamente presenti durante l'attacco, la prima risposta dovrebbe essere rimuovere la macchina dalla rete, estraendo la scheda di rete (sempre che ciò non danneggi transazioni commerciali in atto). Disabilitare la rete a basso livello è l'unico vero modo di allontanare l'attaccante dalla postazione presa di mira (saggio consiglio di Philip Hofmeister).

However, some tools installed by rootkits, trojans and, even, a rogue user connected through a back door, might be capable of detecting this event and react to it. Seeing a `rm -rf /` executed when you unplug the network from the system is not really much fun. If you are unwilling to take the risk, and you are sure that the system is compromised, you should *unplug the power cable* (all of them if more than one) and cross your fingers. This may be extreme but, in fact, will avoid any logic-bomb that the intruder might have programmed. In this case, the compromised system *should not be re-booted*. Either the hard disks should be moved to another system for analysis, or you should use other media (a CD-ROM) to boot the system and analyze it. You should *not* use Debian's rescue disks to boot the system, but you *can* use the shell provided by the installation disks (remember, Alt+F2 will take you to it) to analyze [1] the system.

The most recommended method for recovering a compromised system is to use a live-filesystem on CD-ROM with all the tools (and kernel modules) you might need to access the compromised system. You can use the mkinitrd-cd package to build such a CD-ROM[2]. You might find the http://www.caine-live.net/ (Computer Aided Investigative Environment) CD-ROM useful here too, since it's also a live CD-ROM under active development with forensic tools useful in these situations. There is not (yet) a Debian-based tool such as this, nor an easy way to build the CD-ROM using your own selection of Debian packages and mkinitrd-cd (so you'll have to read the documentation provided with it to make your own CD-ROMs).

If you really want to fix the compromise quickly, you should remove the compromised host from your network and re-install the operating system from scratch. Of course, this may not be effective because you will not learn how the intruder got root in the first place. For that case, you must check everything: firewall, file integrity, log host, log files and so on. For more information on what to do following a break-in, see http://www.cert.org/tech_tips/root_compromise.html or SANS's https://www.sans.org/white-papers/.

Some common questions on how to handle a compromised Debian GNU/Linux system are also available in.

Fare una copia di ripristino del sistema

Ricordate che, se siete sicuri che il sistema sia stato compromesso, non potete fare affidamento sui programmi installati né sulle informazioni restituite. Le applicazioni potrebbero essere state infettate da trojan, altri moduli del kernel potrebbero essere stati installati, etc.

La miglior cosa da farsi è una copia completa del file system (mediante **dd**), dopo la reinizializzazione da un supporto sicuro. A tal fine, possono risultare comodi i CD-ROM di Debian GNU/Linux, dal momento che, iniziata l'installazione, forniscono, nella console 2, una shell raggiungibile con Alt+F2 e Invio. Da questa,

[1] >If you are adventurous, you can login to the system and save information on all running processes (you'll get a lot from /proc/nnn/). It is possible to get the whole executable code from memory, even if the attacker has deleted the executable files from disk. Then pull the power cord.
[2] >In fact, this is the tool used to build the CD-ROMs for the http://www.gibraltar.at/ project (a firewall on a live CD-ROM based on the Debian distribution).

si salvano le informazioni su un'altra postazione, se possibile (magari con un server di file di rete, tramite NFS/FTP), sì da poter svolgere, a sistema interessato non in linea, l'analisi del danno o la reinstallazione.

Se siete sicuri che la compromissione è avvenuta tramite un trojan installato in un modulo del kernel, potete avviare tramite l'immagine del kernel denominata *rescue*. Assicuratevi di avviare in modalità *single user*, per evitare che altri processi trojanizzati si avviino dopo il kernel.

Contattate il vostro CERT locale

The CERT (Computer and Emergency Response Team) is an organization that can help you recover from a system compromise. There are CERTs worldwide [3] and you should contact your local CERT in the event of a security incident which has lead to a system compromise. The people at your local CERT can help you recover from it.

Providing your local CERT (or the CERT coordination center) with information on the compromise even if you do not seek assistance can also help others since the aggregate information of reported incidents is used in order to determine if a given vulnerability is in wide spread use, if there is a new worm aloft, which new attack tools are being used. This information is used in order to provide the Internet community with information on the http://www.cert.org/current/, and to publish http://www.cert.org/incident_notes/ and even http://www.cert.org/advisories/. For more detailed information read on how (and why) to report an incident read http://www.cert.org/tech_tips/incident_reporting.html.

You can also use less formal mechanisms if you need help for recovering from a compromise or want to discuss incident information. This includes the http://marc.theaimsgroup.com/?l=incidents and the http://marc.theaimsgroup.com/?l=intrusions.

Analisi "patologica"

If you wish to gather more information, the tct (The Coroner's Toolkit from Dan Farmer and Wietse Venema) package contains utilities which perform a *post mortem* analysis of a system. tct allows the user to collect information about deleted files, running processes and more. See the included documentation for more information. These same utilities and some others can be found in http://www.sleuthkit.org/ by Brian Carrier, which provides a web front-end for forensic analysis of disk images. In Debian you can find both sleuthkit (the tools) and autopsy (the graphical front-end).

Ricordate di condurre sempre l'analisi patologica sulla copia di ripristino dei dati, *mai* direttamente sui dati stessi: in caso di alterazione durante l'analisi, ogni tipo di prova andrebbe persa!

You will find more information on forensic analysis in Dan Farmer's and Wietse Venema's http://www.porcupine.org/forensics/forensic-discovery/ book (available online), as well as in their http://www.porcupine.org/forensics/column.html and their http://www.porcupine.org/forensics/handouts.html. Brian Carrier's newsletter http://www.sleuthkit.org/informer/index.php is also a very good resource on forensic analysis tips. Finally, the http://www.honeynet.org/misc/chall.html are an excellent way to hone your forensic analysis skills as they include real attacks against honeypot systems and provide challenges that vary from forensic analysis of disks to firewall logs and packet captures. For information about available forensics packages in Debian visit https://salsa.debian.org and search for *forensic*.

FIXME: Se tutto va bene, in futuro questo paragrafo dovrebbe contenere maggiori informazioni sui metodi di diagnosi in un sistema Debian che ha subito un attacco.

[3] > This is a list of some CERTs, for a full list look at the http://www.first.org/about/organization/teams/index.html (FIRST is the Forum of Incident Response and Security Teams): http://www.auscert.org.au (Australia), http://www.unam-cert.unam.mx/ (Mexico) http://www.cert.funet.fi (Finland), http://www.dfn-cert.de (Germany), http://cert.uni-stuttgart.de/ (Germany), http://security.dico.unimi.it/ (Italy), http://www.jpcert.or.jp/ (Japan), http://cert.uninett.no (Norway), http://www.cert.hr (Croatia) http://www.cert.pl (Poland), http://www.cert.ru (Russia), http://www.arnes.si/si-cert/ (Slovenia) http://www.rediris.es/cert/ (Spain), http://www.switch.ch/cert/ (Switzerland), http://www.cert.org.tw (Taiwan), and http://www.cert.org (US).

FIXME: Bisognerebbe parlare di come creare un archivio di tipo debsum su un sistema stabile, salvando i file MD5sum su CD e ripristinando su una partizione distinta il sistema recuperato.

FIXME: Add pointers to forensic analysis papers (like the Honeynet's reverse challenge or http://staff.washington.edu/dittrich/).

Analisi di codice malevolo

Some other tools that can be used for forensic analysis provided in the Debian distribution are: strace and ltrace

Any of these packages can be used to analyze rogue binaries (such as back doors), in order to determine how they work and what they do to the system. Some other common tools include **ldd** (in libc6), **strings** and **objdump** (both in binutils).

If you try to do forensic analysis with back doors or suspected binaries retrieved from compromised systems, you should do so in a secure environment (for example in a bochs or xen image or a **chroot**'ed environment using a user with low privileges[4]). Otherwise your own system can be back doored/r00ted too!

Se siete interessati nell'analisi dei malware, allora dovreste leggere il capitolo http://www.porcupine.org/forensics/forensic-discovery/chapter6.html del libro sull'analisi forense di Dan Farmer e Wietse Venema.

[4] >Be *very* careful if using chroots, since if the binary uses a kernel-level exploit to increase its privileges it might still be able to infect your system

Capitolo 12. Domande frequenti (FAQ)

Questo capitolo introduce alcune delle domande più frequenti nella lista di discussione Debian sulla sicurezza (Debian security mailing list): bisogna leggerle prima di spedire domande e rischiare la risposta, RTFM ("leggiti il fottuto manuale!").

La sicurezza nel sistema operativo Debian

Debian è più sicura di quella X?

A system is only as secure as its administrator is capable of making it. Debian's default installation of services aims to be *secure*, but may not be as paranoid as some other operating systems which install all services *disabled by default*. In any case, the system administrator needs to adapt the security of the system to the local security policy.

For a collection of data regarding security vulnerabilities for many operating systems, see the http://www.cert.org/stats/cert_stats.html or generate stats using the http://nvd.nist.gov/statistics.cfm (formerly ICAT) Is this data useful? There are several factors to consider when interpreting the data, and it is worth noticing that the data cannot be used to compare the vulnerabilities of one operating system versus another.[1] Also, keep in mind that some reported vulnerabilities regarding Debian apply only to the *unstable* (i.e. unreleased) branch.

Debian è più sicura di altre distribuzioni Linux (Red Hat, SuSe...)?

Non ci sono davvero molte differenze fra le varie distribuzioni Linux, al di là dell'installazione di base e del sistema di gestione dei pacchetti; in genere, esse hanno in comune molti applicativi (ad esempio, il kernel, Bind, Apache, OpenSSH, XFree, gcc, zlib, etc.), che si differenziano principalmente per la versione inclusa nella distribuzione stabile.

Red Hat è stata sfortunata nel rilasciare una versione contenente l'allora attuale foo 1.2.3., che aveva un difetto nella sicurezza, come si scoprì in un secondo momento; invece, Debian ha avuto miglior sorte nel rilasciare la propria con foo 1.2.4, in cui quel difetto era stato eliminato. L'identico caso si ripeté con http://www.cert.org/advisories/CA-2000-17.html, un paio di anni fa.

Fra i gruppi per la sicurezza delle maggiori distribuzioni Linux c'è molta collaborazione: raramente un distributore trascura di sistemare gli aggiornamenti in materia di sicurezza e le cognizioni sulla sua vulnerabilità non vengono mai nascoste agli altri, dato che le correzioni vengono coordinate già in fase di sviluppo, oppure dal http://cert.org. Ne consegue che gli aggiornamenti necessari vengono diffusi nello stesso momento e la relativa sicurezza delle diverse distribuzioni è molto simile.

Riguardo ad essa, uno dei maggiori vantaggi di Debian è il semplice sistema di aggiornamento mediante l'uso di **apt**; ma ecco altri aspetti da considerare:

- Rispetto ad altre distribuzioni, Debian offre maggiori strumenti per la sicurezza, vedete in Capitolo 8, *Strumenti per la sicurezza in Debian*.

- Debian's standard installation is smaller (less functionality), and thus more secure. Other distributions, in the name of usability, tend to install many services by default, and sometimes they are not proper-

[1] For example, based on some data, it might seem that Windows NT is more secure than Linux, which is a questionable assertion. After all, Linux distributions usually provide many more applications compared to Microsoft's Windows NT. This *counting vulnerabilities* issues are better described in http://www.dwheeler.com/oss_fs_why.html#security by David A. Wheeler

ly configured (remember the http://www.sophos.com/virusinfo/analyses/linuxlion.html http://www.sophos.com/virusinfo/analyses/linuxramen.html). Debian's installation is not as limited as OpenBSD (no daemons are active per default), but it's a good compromise. [2]

- Debian documenta al meglio le pratiche di sicurezza, in scritti come questo.

Bugtraq cita molti difetti di Debian: è forse molto vulnerabile?

La distribuzione Debian vanta un grande e crescente numero di pacchetti software, probabilmente più di quelli offerti da molti sistemi operativi proprietari. Maggiore è il numero dei pacchetti installati, maggiore il rischio di problemi di sicurezza in qualunque dato sistema.

Aumenta il numero di esaminatori del codice sorgente per cercare imperfezioni. Ci sono molti resoconti circa i controlli sul codice sorgente dei principali componenti software inclusi in Debian: qualora un controllo riveli dei difetti per la sicurezza, vi si ovvia e se ne manda un resoconto a liste come Bugtraq.

Di solito, i difetti presenti in Debian affliggono anche le altre distribuzioni. Controllate la sezione "Specifico di Debian: sì/no" all'inizio di ogni resoconto DSA (Debian Specific Advisory).

Debian ha certificazioni di sicurezza?

Risposta concisa: no.

Risposta lunga: la certificazione costa soldi (specialmente una certificazione di sicurezza *seria*), nessuno ha dedicato le risorse allo scopo di certificare Debian GNU/Linux ad alcun livello come, ad esempio, il http://niap.nist.gov/cc-scheme/st/. Se siete interessati ad avere una distribuzione GNU/Linux certificata in ambito sicurezza, provate a fornire prima di tutto le risorse per rendere possibile ciò.

There are currently at least two linux distributions certified at different http://en.wikipedia.org/wiki/Evaluation_Assurance_Level levels. Notice that some of the CC tests are being integrated into the http://ltp.sourceforge.net which is available in Debian in the ltp.

Ci sono programmi per rendere Debian più sicura?

Yes. http://bastille-linux.sourceforge.net/, originally oriented toward other Linux distributions (Red Hat and Mandrake), it currently works also for Debian. Steps are being taken to integrate the changes made to the upstream version into the Debian package, named bastille.

Alcuni, tuttavia, ritengono che gli strumenti per avere maggiore sicurezza non possano eliminare l'esigenza di una buona amministrazione.

Ho necessità di rendere disponibile il servizio XYZ, come sceglierlo?

Un punto di forza di Debian è l'ampia varietà di scelta fra pacchetti che forniscono le stesse funzionalità (serventi di tipo DNS, ftp, di posta, web, etc.). Questo può confondere un amministratore inesperto nel determinare il pacchetto più adatto alle sue esigenze. La soluzione migliore dipende da un compromesso fra le esigenze di funzionalità e quelle della sicurezza. Per decidere fra pacchetti simili, bisogna rispondere ad alcune domande, come:

- Il software continua a essere sviluppato? Quand'è uscita l'ultima versione?

[2] >Without diminishing the fact that some distributions, such as Red Hat or Mandrake, are also taking into account security in their standard installations by having the user select *security profiles*, or using wizards to help with configuration of *personal firewalls*.

- Il pacchetto è obsoleto? Il numero di versione *non* ne indica l'età: cercate di tracciarne la storia.
- Il programma è esente da difetti? Ci sono avvisi sulla sua sicurezza?
- Il programma ha tutte le funzionalità cercate? Ne ha addirittura più di quelle necessarie?

Come rendere il servizio XYZ più sicuro in Debian?

In questo documento si troveranno informazioni sul modo per rendere alcuni servizi (FTP, Bind) più sicuri in Debian GNU/Linux. Per quelli non trattati qui, vedete la documentazione del programma o in generale le informazioni per quel software in GNU/Linux. La maggior parte delle linee guida per la sicurezza di Unix si applicano anche a Debian. Proteggere un servizio in Debian è come farlo in qualunque altra distribuzione Linux (o Un*x, per quell'argomento).

Come posso rimuovere tutte le etichette per i servizi?

If you do not like users connecting to your POP3 daemon, for example, and retrieving information about your system, you might want to remove (or change) the banner the service shows to users. [3] Doing so depends on the software you are running for a given service. For example, in **postfix**, you can set your SMTP banner in `/etc/postfix/main.cf`:

```
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
```

Other software is not as easy to change. ssh will need to be recompiled in order to change the version that it prints. Take care not to remove the first part (`SSH-2.0`) of the banner, which clients use to identify which protocol(s) is supported by your package.

Sono sicuri tutti i pacchetti Debian?

Il Team Debian per la Sicurezza non può analizzare i potenziali punti deboli di tutti i pacchetti, perché mancano le risorse per revisionare l'intero codice sorgente; Debian beneficia, però, delle revisioni svolte dagli sviluppatori.

In effetti, uno sviluppatore Debian potrebbe benissimo distribuire un virus di tipo trojan in un pacchetto, senza che venga scoperto: anche se esperti in un ramo di Debian, sarebbe impossibile trattare tutte le possibili situazioni in cui il trojan potrebbe agire. Perciò, Debian ha una licenza *"no guarantees"* (*nessuna garanzia*).

Tuttavia, gli utenti Debian possono fidare nel fatto che il codice stabile abbia un vasto pubblico: la maggior parte dei problemi si manifesterebbe con l'uso. In un sistema molto importante, non è indicato installare software che non sia stato sottoposto ad un processo di revisione del codice sorgente. In ogni caso, quand'anche nella distribuzione sia introdotta una vulnerabilità della sicurezza, il processo di inclusione dei pacchetti assicura la riconducibilità finale allo sviluppatore (mediante le firme digitali). Il progetto Debian non sottovaluta il problema.

Chiunque può leggere certi file di log/configurazione: non è insicuro?

Naturalmente, nel proprio sistema, ognuno può cambiare i permessi predefiniti di Debian. Con l'attuale codice di condotta sui file di log e configurazione, ognuno li può leggere, *salvo che* contengano informazioni sensibili.

[3] >Note that this is 'security by obscurity', and will probably not be worth the effort in the long term.

Si raccomanda prudenza, nel fare cambiamenti, dal momento che:

- Se si limitano i permessi, certi processi potrebbero non riuscire a scrivere sui file di log.
- Alcune applicazioni potrebbero non funzionare se non fosse leggibile il file di configurazione da cui dipendono. Per esempio, se viene rimosso il permesso di libera leggibilità per `/etc/samba/smb.conf`, il programma **smbclient** non funzionerà, se attivato da un comune utente.

FIXME: Controllare se questo sia scritto nella Policy. Alcuni pacchetti (ad esempio i demoni ftp) sembrano imporre permessi differenti.

Perché /root/ (o User X) ha i permessi impostati a 755?

In effetti, l'identico problema riguarda qualunque altro utente. Siccome l'installazione di Debian non mette *alcun* file sotto quella directory, non vi sono informazioni importanti da proteggere; se ritenete che tali permessi siano troppo laschi per il sistema, li potete portare a 750. Per gli utenti, leggete in sezione chiamata «Limitare l'accesso alle informazioni di altri utenti».

This Debian security mailing list http://lists.debian.org/debian-devel/2000/11/msg00783.html has more on this issue.

Rimozione dei messaggi che arrivano in console dopo l'installazione di un grsec/firewall

Se riceveste messaggi in console ed aveste configurato `/etc/syslog.conf`, in modo da trasmetterli a dei file o ad uno speciale TTY, potreste vedere i messaggi direttamente sulla console.

Per ogni kernel, il livello predefinito per la trasmissione dei messaggi di log è 7 e quelli con priorità inferiore appariranno in console. Di solito i firewall (il regno dei LOG) ed altri strumenti per la sicurezza registrano log con priorità inferiore, che, quindi, vengono spediti direttamente in console.

To reduce messages sent to the console, you can use **dmesg** (`-n` option, see dmseg(8)), which examines and *controls* the kernel ring buffer. To fix this after the next reboot, change `/etc/init.d/klogd` from:

```
KLOGD=""
```

a:

```
KLOGD="-c 4"
```

Usate un numero inferiore per `-c`, se continuate a vederli; una descrizione dei vari livelli dei messaggi di log si trova in `/usr/include/sys/syslog.h`:

```
#define LOG_EMERG       0       /* system is unusable */
#define LOG_ALERT       1       /* action must be taken immediately */
#define LOG_CRIT        2       /* critical conditions */
#define LOG_ERR         3       /* error conditions */
#define LOG_WARNING     4       /* warning conditions */
#define LOG_NOTICE      5       /* normal but significant condition */
#define LOG_INFO        6       /* informational */
#define LOG_DEBUG       7       /* debug-level messages */
```

Utenti e gruppi del sistema operativo

Tutti gli utenti di sistema sono necessari?

Yes and no. Debian comes with some predefined users (user id (UID) < 99 as described in http://www.debian.org/doc/debian-policy/ or `/usr/share/doc/base-passwd/README`) to ease the installation of some services that require that they run under an appropriate user/UID. If you do not intend to install new services, you can safely remove those users who do not own any files in your system and do not run any services. In any case, the default behavior is that UID's from 0 to 99 are reserved in Debian, and UID's from 100 to 999 are created by packages on install (and deleted when the package is purged).

Per scoprire facilmente gli utenti che non possiedono file, è possibile eseguire (da root, dato che agli utenti comuni potrebbero mancare i permessi necessari per potere esaminare alcune cartelle "delicate") il seguente comando[4]:

```
cut -f 1 -d : /etc/passwd | \
while read i; do find / -user "$i" | grep -q . || echo "$i"; done
```

These users are provided by base-passwd. Look in its documentation for more information on how these users are handled in Debian. The list of default users (with a corresponding group) follows:

- root: root è (tipicamente) il superutente.
- daemon: alcuni demoni sprovvisti di privilegi di sorta che devono scrivere su file sono attivi come daemon.daemon (per esempio, **portmap**, **atd** e probabilmente, altri), invece, i demoni che non hanno bisogno di possedere file, possono attivarsi come nobody.nogroup, mentre altri, più rispettosi della sicurezza, lo fanno come utenti dedicati. Il demone utente è utile anche per i demoni installati localmente.
- bin: mantenuto per ragioni storiche.
- sys: idem come sopra; /dev/vcs* e `/var/spool/cups`, però, sono di proprietà del gruppo sys.
- sync: la shell dell'utente sync è `/bin/sync`, così, se la sua parola d'ordine è semplice da indovinare (una cosa tipo ""), chiunque può sincronizzare il sistema da console, anche senza avere un account.
- games: molti giochi vengono impostati per il gruppo (SETGID) games , sì da poter scrivere file coi migliori punteggi. Spiegazione nelle Linee Guida.
- man: il programma man (a volte) è attivo come utente man, in modo da poter scrivere pagine in `/var/cache/man`.
- lp: usato dai demoni di stampa.
- mail: le caselle in `/var/mail` appartengono al gruppo mail, come descritto nelle Linee Guida. Utente e gruppo vengono usati, ad altri fini, anche da vari MTA.
- news: diversi serventi di news ed altri programmi associati (come **suck**) impiegano l'utente ed il gruppo in vari modi: spesso, i file nello spool news appartengono ad entrambi. Programmi come **inews**, usati per spedire news, vengono tipicamente impostati per il gruppo (SETGID) news.
- uucp: l'utente e il gruppo UUCP sono usati dal sottosistema UUCP, cui appartengono i file di spool e di configurazione. Gli utenti del gruppo uucp possono eseguire uucico.

[4] Attenzione, perché questo metterà in pericolo l'intero sistema. Se avete parecchio spazio disco e molte partizioni, potreste voler ridurre il campo di applicazione.

- proxy: come demone, questo utente e gruppo possono essere usati da alcuni demoni (di tipo proxy, in particolare) sprovvisti di utente dedicato ma che debbono essere proprietari di quei file. Per esempio, il gruppo proxy è usato da **pdnsd**, mentre **squid** è attivo come utente proxy.

- majordom: nei sistemi Debian, **Majordomo** aveva un UID allocato staticamente, per ragioni storiche, adesso non è più installato.

- postgres: i database **Postgresql** appartengono a questo utente e gruppo. Tutti i file di `/var/lib/postgresql` appartengono a questo utente per rafforzare la sicurezza.

- www-data: alcuni browser vengono eseguiti come www-data. Il contenuto di rete *non* dovrebbe essere posseduto da questo utente, altrimenti un server compromesso potrebbe riscrivere un sito web. I dati restituiti da un server web, file di log compresi, appartengono a www-data.

- backup: così, le responsabilità del ripristino/recupero possono essere date, localmente, ad una persona che non abbia i pieni permessi di root.

- operator: storicamente (e praticamente) operator è il solo accredito a livello utente che possa autenticarsi da remoto senza dipendere da NIS/NFS.

- list: gli archivi dei gruppi di discussione (mailing list) appartengono a questo utente e gruppo; alcuni appositi programmi girano come utente list.

- irc: usato dai demoni irc. Un utente allocato staticamente occorre solo per via di un baco in **ircd**, che imposta con SETUID() un determinato UID all'avvio.

- gnats.

- nobody.nogroup: demoni che non fanno uso di alcun file vengono eseguiti come utente e gruppo nobody; in tal modo, nessun file nel sistema è di loro proprietà.

Altri gruppi senza utenti associati:

- adm: gruppo usato per compiti di monitoraggio del sistema, i suoi membri possono leggere i file di log della cartella `/var/log` e usare xconsole. Storicamente, `/var/log` era `/usr/adm` (e poi, `/var/adm`), da cui il nome del gruppo.

- tty: gruppo titolare dei dispositivi TTY (TeleTYpe, telescrivente), usati come strumento di comunicazione per scrivere su TTY altrui.

- disk: accesso "grezzo" ai dischi, perlopiù equivalente all'accesso come root.

- kmem: questo gruppo legge /dev/kmem e file similari: è principalmente una reliquia di BSD, ma se i programmi richiedono un accesso diretto, in lettura, alla memoria di sistema si può impostare kmem SETGID.

- dialout: i suoi membri hanno pieno e diretto accesso alle porte seriali e possono riconfigurare il modem, fare chiamate verso qualunque posto, etc. ...

- dip: acronimo di "Dial-up IP", i suoi membri possono usare strumenti come **ppp**, **dip**, **wvdial**, etc. per avviare una connessione ed eseguire i programmi che utilizzano il modem, ma non possono configurarlo.

- fax: consente ai membri l'uso del fax in emissione/ricezione.

- voice: posta vocale, utile per sistemi che usano il modem come segreteria.

- cdrom: usato localmente per dare agli utenti un accesso all'unità CDROM.

- floppy: usato localmente per dare agli utenti un accesso all'unità floppy.
- tape: usato localmente per dare agli utenti un accesso alle unità a nastro.
- sudo: i membri di questo gruppo non devono digitare la password per usare **sudo**. Vedete in `/usr/share/doc/sudo/OPTIONS`.
- audio: usato localmente per dare agli utenti l'accesso ai dispositivi audio.
- src: proprietario di codice sorgente, file in `/usr/src` compresi, viene usato per dare agli utenti la facoltà di gestire il codice sorgente presente sul sistema.
- shadow: i programmi che necessitano di accedere al file `/etc/shadow`, devono avere impostato il SETGID per il gruppo shadow, che può leggerlo.
- utmp: può scrivere sul file `/var/run/utmp` e simili: programmi che devono poter scrivere su di esso, devono avere impostato il SETGID per il gruppo utmp.
- video: usato localmente per dare agli utenti l'accesso ai dispositivi video.
- staff: consente agli utenti modifiche locali del sistema (in `/usr/local`, `/home`) anche senza i privilegi di root. Confrontatelo con il gruppo "adm", volto maggiormente a compiti di monitoraggio e sicurezza.
- users: mentre i sistemi Debian usano il sistema predefinito di creare un gruppo personale per ogni utente, alcuni preferiscono un sistema di gruppi più tradizionale, in cui ogni utente è membro del gruppo "users".

Ho rimosso un utente di sistema! Come posso rimediare?

If you have removed a system user and have not made a backup of your `password` and `group` files you can try recovering from this issue using **update-passwd** (see update-passwd(8)).

Qual è la differenza fra i gruppi adm e staff?

Gli appartenenti al gruppo "adm" sono, di solito, amministratori ai quali i permessi del gruppo consentono la lettura dei log senza dare il comando **su**. Al gruppo "staff", invece, appartengono i coadiutori/amministratori novizi, cui è permesso di lavorare in `/usr/local` e di creare cartelle in `/home`.

Perché c'è un nuovo gruppo per ogni nuovo utente? (Ovvero: perché Debian assegna un gruppo ad ogni utente?)

La linea di condotta di Debian prevede di assegnare ad ogni utente un proprio gruppo. Il tradizionale schema UN*X assegna tutti gli utenti al gruppo *users*. Ulteriori gruppi vengono creati ed utilizzati per limitare l'accesso a file condivisi associati a cartelle di progetti diverse. Siccome, all'atto della creazione, i file vengono associati al gruppo di prima appartenenza dell'autore, la loro gestione diventa difficile, se l'utente lavora su più progetti.

Lo schema Debian risolve questo problema, assegnando ad ogni utente un gruppo personale, sicché, con una corretta umask (0002) e con il bit per l'assegnazione del gruppo (SETGID) impostato su una data cartella di progetto, il gruppo viene assegnato automaticamente ed i file vengono creati in quella cartella. Ciò agevola chi lavora su più progetti, evitandogli di cambiare gruppo o umask quando lavora su file condivisi.

Tuttavia, dando il valore "no" alla variabile *USERGROUPS*, nel file `/etc/adduser.conf`, si può cambiare questo comportamento. In tal modo, quando si crea un nuovo utente, non si crea anche un nuovo

gruppo. Le stesse considerazioni valgono per l'impostazione di *USERS_GID* al GID cui tutti gli utenti appartengono.

Domande riguardanti i servizi e le porte aperte

Perché in installazione vengono attivati tutti i servizi?

È soltanto un compromesso tra l'essere da un lato attenti alla sicurezza e dall'altro user-friendly. Diversamente da OpenBSD, che disabilita tutti i servizi a meno che non siano esplicitamente attivati dall'amministratore, Debian GNU/Linux attiva tutti i servizi installati a meno che non vengano disattivati (vedete in sezione chiamata «Disabilitare i servizi attivi in modalità demone» per maggiori informazioni). Dopo tutto siete stati voi ad installare il servizio, o no?

Ci sono state molte discussioni sulle mailing list Debian (sia su debian-devel che su debian-security) riguardo a quale sia il miglior compromesso per un'installazione standard. Comunque, al momento della stesura (marzo 2002) non si è ancora arrivati ad un accordo.

Posso rimuovere inetd?

Inetd is not easy to remove since netbase depends on the package that provides it (netkit-inetd). If you want to remove it, you can either disable it (see sezione chiamata «Disabilitare i servizi attivi in modalità demone») or remove the package by using the equivs package.

Perché ho la porta 111 aperta?

La porta 111 è quella del portmapper sunrpc e viene installata in maniera predefinita come parte dell'installazione Debian poiché non c'è bisogno di sapere quando un programma utente potrebbe aver bisogno che l'RPC che funzioni correttamente. In ogni caso, si usa principalmente per l'NFS, se non vi serve, rimuovetelo com'è spiegato in sezione chiamata «Rendere sicuri i servizi RPC».

In versions of the portmap package later than 5-5 you can actually have the portmapper installed but listening only on localhost (by modifying `/etc/default/portmap`)

A cosa serve identd (porta 113) ?

Il servizio identd serve per l'autenticazione, che tra l'altro identifica il proprietario di una specifica connessione TCP/IP al server remoto che accetta la connessione. Tipicamente, quando un utente si connette a un host remoto, l'**inetd** dell'host remoto manda una query alla porta 113 per ottenere informazioni sull'utente. Spesso viene usato per la posta, i server FTP e IRC e può essere usato anche per tracciare chi, degli utenti del vostro sistema, stia tentando di attaccare un sistema remoto.

There has been extensive discussion on the security of **identd** (See http://lists.debian.org/debian-security/2001/08/msg00297.html). In general, **identd** is more helpful on a multi-user system than on a single user workstation. If you don't have a use for it, disable it, so that you are not leaving a service open to the outside world. If you decide to firewall the identd port, *please* use a reject policy and not a deny policy, otherwise a connection to a server utilizing **identd** will hang until a timeout expires (see http://logi.cc/linux/reject_or_deny.php3).

Ho dei servizi che usano le porte 1 e 6, cosa sono e come posso rimuoverli?

Se avete lanciato `netstat -an` e vi ha restituito:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
```

```
PID/Program name
raw        0      0 0.0.0.0:1               0.0.0.0:*               7
-
raw        0      0 0.0.0.0:6               0.0.0.0:*               7
-
```

You are *not* seeing processes listening on TCP/UDP port 1 and 6. In fact, you are seeing a process listening on a *raw* socket for protocols 1 (ICMP) and 6 (TCP). Such behavior is common to both legitimate software like intrustion detection systems, such as iplogger and portsentry, but some trojans have also been known yo use them. If you have the mentioned packages simply remove them to close the port. If you do not, try netstat's `-p` (process) option to see which process is running these listeners.

Ho trovato la porta XYZ aperta, posso chiuderla?

Naturalmente! Le porte che lasciate aperte debbono essere concordi con le linee guida del vostro host per quanto riguarda i servizi pubblici disponibili per le altre reti. Controllate se vengono avviati da **inetd** (vedete in sezione chiamata «Disabilitare i servizi gestiti da **inetd**») o da altri pacchetti installati e prendete le misure appropriate (es., configurate inetd, rimuovete il pacchetto, evitate di lanciarlo al boot).

Rimuovere dei servizi da `/etc/services` può aiutarmi a rendere sicura la mia macchina?

No, `/etc/services` fornisce solo una mappatura tra un nome virtuale ed un dato numero di porta. Rimuovere nomi da questo file (solitamente) non eviterà che questi servizi vengano lanciati. Alcuni demoni potrebbero non partire se `/etc/services` viene modificato, ma non è la norma. Per disabilitare correttamente il servizio, vedete in sezione chiamata «Disabilitare i servizi attivi in modalità demone».

Problemi comuni di sicurezza

Ho dimenticato la password e non posso accedere al sistema!

I passi che dovete fare per sistemare questo problema dipende dall'avere o meno applicato quanto suggerito per limitare l'accesso a **lilo** e al BIOS del vostro sistema.

Se avete limitato entrambi, dovete disabilitare l'impostazione del BIOS che permette il boot solo dal disco rigido prima di procedere. Se avete dimenticato anche la password del BIOS, dovrete annullare il BIOS aprendo la macchina e rimuovendo la batteria del BIOS.

Una volta che avete abilitato il boot da CD-ROM o da dischetto, tentate i passi seguenti:

- Eseguite il boot da un disco di salvataggio e fate partire il kernel.
- Passate sulla console virtuale (Alt+F2).
- Montate il disco dove risiede la partizione /root.
- Modificate (il disco di rescue di Debian 2.2 contiene l'editor **ae** e Debian 3.0 con **nano-tiny** che è simile a **vi**) `/etc/shadow` e cambiate la riga:

  ```
  root:asdfjl290341274075:XXXX:X:XXXX:X::: (X=any number)
  ```

  a:

```
root::XXXX:X:XXXX:X:::
```

Questo eliminerà la password di root dimenticata, contenuta nel primo campo separato dai due punti dopo il nome dell'utente. Salvate il file, riavviate il sistema ed effettuate il login da root usando una password vuota. Ricordate di annullare la password. Questo funzionerà, a meno che non abbiate configurato il sistema in maniera più attenta, ovvero se avete impedito agli utenti di avere password vuote o a root di effettuare il login dalla console.

Se avete introdotto queste funzionalità, dovrete entrare nel sistema in modalità utente singolo. Se LILO è stato limitato, dovrete eseguire **lilo** subito dopo il reset di root di cui sopra. Questo è abbastanza furbo poiché il vostro `/etc/lilo.conf` dovrà essere configurato alla directory radice (/) del sistema essendo un ramdisk e non un disco rigido reale.

Una volta che LILO è senza restrizioni, provate i seguenti:

- Premete i tasti Alt, Shift o Control appena prima che il vostro BIOS finisca, dovreste ottenere un prompt di LILO.
- Digitate `linux single`, `linux init=/bin/sh` o `linux 1` al prompt.
- Questo vi farà ottenere un prompt di shell in modalità utente singolo e vi chiederà una password, ma la conoscete già.
- Rimontate in lettura/scrittura la partizione di root (/), usando il comando mount:

```
# mount -o remount,rw /
```

- Cambiate la password del superuser con **passwd** (poiché siete il superuser non vi chiederà la precedente).

Come configurare un servizio per i miei utenti senza dare loro una shell?

For example, if you want to set up a POP service, you don't need to set up a user account for each user accessing it. It's best to set up directory-based authentication through an external service (like Radius, LDAP or an SQL database). Just install the appropriate PAM library (libpam-radius-auth, libpam-ldap, libpam-pgsql or libpam-mysql), read the documentation (for starters, see sezione chiamata «Autenticazione degli utenti: PAM») and configure the PAM-enabled service to use the back end you have chosen. This is done by editing the files under `/etc/pam.d/` for your service and modifying the

```
auth   required    pam_unix_auth.so shadow nullok use_first_pass
```

in, per esempio, ldap:

```
auth   required    pam_ldap.so
```

In caso di directory LDAP, alcuni servizi forniscono uno schema LDAP da includere nella directory allo scopo di usare l'autenticazione via LDAP. Se state usando un database relazionale, un trucco utile è usare l'espressione *where* quando configurate il modulo PAM. Per esempio, se avete un database con i seguenti campi:

```
(user_id, user_name, realname, shell, password, UID, GID, homedir, sys, pop, ima
```

Prendendo i campi booleani degli attributi dei servizi, potrete usarli per abilitare o disabilitare i diversi servizi solamente inserendo la riga appropriata nei seguenti file:

- `/etc/pam.d/imap`:`where=imap=1`.
- `/etc/pam.d/qpopper`:`where=pop=1`.
- `/etc/nss-mysql*.conf`:`users.where_clause = user.sys = 1;`.
- `/etc/proftpd.conf`: `SQLWhereClause "ftp=1"`.

Il mio sistema è vulnerabile! (Ne sei sicuro?)

Sa scansione per la ricerca delle vulnerabilità risulta che il mio sistema Debian è vulnerabile!

Molti scanner per ricercare vulnerabilità trovano falsi positivi, quando vengono usati in sistemi Debian, usano solamente il controllo di versione per determinare se un dato pacchetto sia vulnerabile, ma non ne testano la vulnerabilità. Poiché Debian non cambia la versione del software quando corregge un pacchetto (molte volte una correzione rilasciata recentemente pare una vecchia versione) molti programmi tendono a "pensare" che un sistema Debian aggiornato sia vulnerabile, invece è vero il contrario.

Se pensate che il vostro sistema sia sicuro usando le patch di sicurezza, vi invito ad incrociare i vostri riferimenti con il database sulla sicurezza pubblicato con il DSAs (leggete in sezione chiamata «Avvisi di sicurezza Debian») per eliminare false sicurezze, sempre se il tool che usate include i riferimenti per CVE.

Ho notato traccia di un attacco nei miei log di sistema. Il mio sistema è compromesso?

Un traccia di attacco non significa necessariamente che il sistema sia stato compromesso, dovreste compiere i comuni passaggi per determinare se il sistema è stato realmente compromesso (leggete in Capitolo 11, *Dopo la compromissione (reazione agli incidenti)*). Altresì, se avete visto dai log che è avvenuto un attacco, è possibile che il vostro sistema sia vulnerabile allo stesso tipo di attacco (un attaccante molto determinato può aver sfruttato molte altre falle di sicurezza).

Nei miei log ho trovato una strana riga "MARK": sono stato attaccato?

Dovete ricercare le seguenti righe nei vostri log di sistema:

```
Dec 30 07:33:36 debian -- MARK --
Dec 30 07:53:36 debian -- MARK --
Dec 30 08:13:36 debian -- MARK --
```

This does not indicate any kind of compromise, and users changing between Debian releases might find it strange. If your system does not have high loads (or many active services), these lines might appear throughout your logs. This is an indication that your **syslogd** daemon is running properly. From syslogd(8):

```
-m interval
       The syslogd logs a mark timestamp  regularly.   The
       default interval between two -- MARK -- lines is 20
       minutes.  This can be  changed  with  this  option.
```

```
                Setting the interval to zero turns it off entirely.
```

Ho trovato nei miei log un utente che può usare il comando "su": sono compromesso?

Dovete ricercare delle righe simili alle seguenti nei vostri log:

```
Apr  1 09:25:01 server su[30315]: + ??? root-nobody
Apr  1 09:25:01 server PAM_unix[30315]: (su) session opened for user nobody by (
```

Non preoccupatevi. Controllate se queste righe sono presenti nei job di **cron** (solitamente si trovano in `/etc/cron.daily/find` oppure **logrotate**):

```
$ grep 25 /etc/crontab
25 9    * * *   root    test -e /usr/sbin/anacron || run-parts --report
/etc/cron.daily
$ grep nobody /etc/cron.daily/*
find:cd / && updatedb --localuser=nobody 2>/dev/null
```

Nei miei log ho trovato un possibile "SYN flooding":sono sotto attacco?

Se nei vostri log trovate delle righe come le seguenti:

```
May 1 12:35:25 linux kernel: possible SYN flooding on port X. Sending cookies.
May 1 12:36:25 linux kernel: possible SYN flooding on port X. Sending cookies.
May 1 12:37:25 linux kernel: possible SYN flooding on port X. Sending cookies.
May 1 13:43:11 linux kernel: possible SYN flooding on port X. Sending cookies.
```

Controllate se sono presenti molte connessioni attive sul server usando **netstat**, ad esempio:

```
linux:~# netstat -ant | grep SYN_RECV | wc -l
   9000
```

Questo è un indice certo di un attacco denial of service (DoS) contro la porta X (molto spesso avvengono contro dei servizi pubblici come i server web o i server di posta). Vi invito ad attivare nel vostro kernel l'opzione TCD syncookies, leggete in sezione chiamata «Configurare i Syncookies». Attenzione comunque, come un attacco DoS può saturare la vostra rete, al tempo stesso potete fermarlo mandando in crash il sistema (saturando i file descrittori, il sistema rimane inerte finché la connessione TCP non viene interrotta). Il solo modo efficace per fermare questo tipo di attacco è quello di contattare il vostro provider.

Ho trovato strane sessioni di root nei miei log: sono compromesso?

Potreste vedere queste righe nel vostro file `/var/log/auth.log`:

```
May 2 11:55:02 linux PAM_unix[1477]: (cron) session closed for user root
May 2 11:55:02 linux PAM_unix[1476]: (cron) session closed for user root
May 2 12:00:01 linux PAM_unix[1536]: (cron) session opened for user root by
(UID=0)
```

```
May 2 12:00:02 linux PAM_unix[1536]: (cron) session closed for user root
```

Sono dovuti all'esecuzione di un processo **cron** (nell'esempio, ogni cinque minuti). Per stabilire quale programma è responsabile dei processi, controllare le immissioni sotto: `/etc/crontab`, `/etc/cron.d`, `/etc/crond.daily` ed il `crontab` di root sotto `/var/spool/cron/crontabs`.

Ho subito un'intrusione, cosa devo fare?

Esistono diversi passaggi che potreste seguire in caso di intrusione:

- Controllare che il vostro sistema sia aggiornato con patch di sicurezza per le vulnerabilità pubbliche. Se il sistema è vulnerabile, le possibilità che sia di fatto compromesso sono incrementate. Le possibilità aumentano ancora se la vulnerabilità è nota da tempo, dato che esiste solitamente più attività relativa alle vulnerabilità datate. Qui c'è un link alle http://www.sans.org/top20.htm.

- Leggere questo documento, specialmente il Capitolo 11, *Dopo la compromissione (reazione agli incidenti)*.

- Chiedere assistenza. Potete usare la mailing list debian-security e chiedere consigli su come ripristinare/riparare il vostro sistema.

- Segnalare al http://www.cert.org locale (se esiste, altrimenti potreste contattare il CERT direttamente). Questo potrebbe o non potrebbe aiutarvi ma, almeno, informate il CERT degli attacchi in corso. Questa informazione è preziosa per determinare quali strumenti ed attacchi vengono utilizzati dalla comunità *blackhat*.

Come posso tracciare un attacco?

Osservando i log (se non sono stati alterati), utilizzando sistemi di rilevamento di intrusioni (vedete in sezione chiamata «Pianificare la ricerca di intrusi»), **traceroute**, **whois** e strumenti simili (inclusa analisi forense), potreste essere in grado di tracciare un attacco fino alla fonte. Il modo in cui reagite a questa informazione dipende unicamente dalla vostra politica di sicurezza, e da cosa *voi* considerate un attacco. Uno scan remoto è un attacco? Un rilevamento di vulnerabilità è un attacco?

Il programma X in Debian è vulnerabile, cosa devo fare?

Per prima cosa, controllate se la vulnerabilità è stata resa pubblica sulle mailing list di sicurezza (come Bugtraq) o altri forum. Il Team per la sicurezza di Debian si tiene in contatto con queste liste, quindi potrebbero essere a conoscenza del problema. Non procedete in modo diverso se vedete un annuncio su http://security.debian.org, da quanto suggerito per correggere la vulnerabilità.

If no information seems to be published, please send e-mail about the affected package(s), as well as a detailed description of the vulnerability (proof of concept code is also OK), to mailto:team@security.debian.org. This will get you in touch with Debian's security team.

Il numero di versione di un pacchetto indica che sta girando una versione vulnerabile!

Invece di aggiornare ad un nuovo rilascio, Debian riporta gli aggiornamenti di sicurezza alla versione che è stata distribuita con la versione stabile. Il motivo è assicurarsi che i rilasci stabili cambino il meno possibile, così che le cose non cambino o si interrompano inaspettatamente a causa di un aggiornamento di sicurezza. Potete controllare se sta girando una versione sicura di un pacchetto dal changelog package, o confrontando il numero di versione (versione corrente -slash- rilascio Debian) con la versione indicata nel Debian Security Advisory.

Software specifico

proftpd is vulnerable to a Denial of Service attack.

Aggiungere `DenyFilter \*.*/` al vostro file di configurazione e per ulteriori informazioni vedete http://www.proftpd.org/critbugs.html.

After installing portsentry, there are a lot of ports open.

È solo il modo in cui funziona **portsentry**. Apre circa venti porte inutilizzate per rilevare i port scan.

Domande sul Team per la sicurezza di Debian

The security team keeps its list of Frequently Asked Questions at the http://www.debian.org/security/faq. Please refer to that web page for up to date information.

Appendice A. Diario delle Revisioni

Diario delle Revisioni

Revisione 3-19.2	Sun May 19 2024	HolgerWansing<hwansing@mailbox.org>

Translation files synchronised with XML sources 3-19

Revisione 3-19.1	Mon May 1 2017	MarcosFouces<marcos.fouces@gmail.com>

Translation files synchronised with XML sources 3-19

Revisione 3-19	April 2017	Marcos Fouces<marcos.fouces@gmail.com>

Migrate to Docbook XML.
Build with Publican. No longer use custom Makefile.
Migrate svn repository to git.
Import chinese, italian, spanish, portuguese, japanese, russian, french and german translations to PO format.

Revisione 3-18	February 2015	ThijsKinkhorst<thijs@debian.org>

Clarify FAQ on raw sockets.
Update section 4.5 on GRUB2.
Replace example postrm user removal code with advice to use deluser/delgroup --system

Revisione 3-17	January 2015	Thijs Kinkhorst<thijs@debian.org>

Remove mention of MD5 shadow passwords.
Do not recommend dselect for holding packages.
No longer include the Security Team FAQ verbatim, because it duplicates information documented elsewhere and is hence perpetually out of date.
Update section on restart after library upgrades to mention needrestart.
Avoid gender-specific language. Patch by Myriam.
Use LSB headers for firewall script. Patch by Dominic Walden.

Revisione 3-16	January 2013	JavierFernández-Sanguino Peña.<jfs@debian.org>

Indicate that the document is not updated with latest versions.
Update pointers to current location of sources.
Update information on security updates for newer releases.
Point information for Developers to online sources instead of keeping the information in the document, to prevent duplication.
Extend the information regarding securing console access, including limiting the Magic SysRq key.
Update the information related to PAM modules including how to restrict console logins, use cracklib and use the features avialable in /etc/pam.d/login. Remove the references to obsolete variables in /etc/login.defs.
Reference some of the PAM modules available to use double factor authentication, for administrators that want to stop using passwords altogether.
Fix shell script example in Appendix.
Fix reference errors.
Point to the Basille sourceforge project instead of the bastille-unix.org site as it is not responding.

Revisione 3-15	December 2010	JavierFernández-Sanguino Peña<jfs@debian.org>

Change reference to Log Analysis' website as this is no longer available.

Revisione 3-14	March 2009	JavierFernández-Sanguino Peña<jfs@debian.org>

Change the section related to choosing a filesystem: note that ext3 is now the default.

Change the name of the packages related to enigmail to reflect naming changes introduced in Debian.

Revisione 3-13 February 2008 JavierFernández-Sanguino Peña<jfs@debian.org>

Change URLs pointing to Bastille Linux to www.Bastille-UNIX.org since the domain has been http://bastille-linux.sourceforge.net/press-release-newname.html.
Fix pointers to Linux Ramen and Lion worms.
Use linux-image in the examples instead of the (old) kernel-image packages.
Fix typos spotted by Francesco Poli.

Revisione 3-12 August 2007 JavierFernández-Sanguino Peña<jfs@debian.org>

Update the information related to security updates. Drop the text talking about Tiger and include information on the update-notifier and adept tools (for Desktops) as well as debsecan. Also include some pointers to other tools available.
Divide the firewall applications based on target users and add fireflier to the Desktop firewall applications list.
Remove references to libsafe, it's not in the archive any longer (was removed January 2006).
Fix the location of syslog's configuration, thanks to John Talbut.

Revisione 3-11 January 2007 JavierFernández-Sanguino Peña<jfs@debian.org>

Thanks go to Francesco Poli for his extensive review of the document.
Remove most references to the woody release as it is no longer available (in the archive) and security support for it is no longer available.
Describe how to restrict users so that they can only do file transfers.
Added a note regarding the debian-private declasiffication decision.
Updated link of incident handling guides.
Added a note saying that development tools (compilers, etc.) are not installed now in the default 'etch' installation.
Added a note saying that development tools (compilers, etc.) are not installed now in the default 'etch' installation.
Fix references to the master security server.
Add pointers to additional APT-secure documentation.
Improve the description of APT signatures.
Comment out some things which are not yet final related to the mirror's official public keys.
Fixed name of the Debian Testing Security Team.
Remove reference to sarge in an example.
Update the antivirus section, clamav is now available on the release. Also mention the f-prot installer.
Removes all references to freeswan as it is obsolete.
Describe issues related to ruleset changes to the firewall if done remotely and provide some tips (in footnotes).
Update the information related to the IDS installation, mention BASE and the need to setup a logging database.
Rewrite the "running bind as a non-root user" section as this no longer applies to Bind9. Also remove the reference to the init.d script since the changes need to be done through /etc/default.
Remove the obsolete way to setup iptables rulesets as woody is no longer supported.
Revert the advice regarding LOG_UNKFAIL_ENAB it should be set to 'no' (as per default).
Added more information related to updating the system with desktop tools (including update-notifier) and describe aptitude usage to update the system. Also note that dselect is deprecated.
Updated the contents of the FAQ and remove redundant paragraphs.
Review and update the section related to forensic analysis of malware.
Remove or fix some dead links.
Fix many typos and gramatical errors reported by Francesco Poli.

Revisione 3-10 November 2006 JavierFernández-Sanguino Peña<jfs@debian.org>

Provide examples using apt-cache's rdepends as suggested by Ozer Sarilar.

Fix location of Squid's user's manual because of its relocation as notified by Oskar Pearson (its maintainer).
Fix information regarding umask, it's logins.defs (and not limits.conf) where this can be configured for all login connections. Also state what is Debian's default and what would be a more restrictive value for both users and root. Thanks to Reinhard Tartler for spotting the bug.

Revisione 3-9 October 2006 JavierFernández-Sanguino Peña<jfs@debian.org>

Add information on how to track security vulnerabilities and add references to the Debian Testing Security Tracker.
Add more information on the security support for testing.
Fix a large number of typos with a patch provided by Simon Brandmair.
Added section on how to disable root prompt on initramfs provided by Max Attems.
Remove references to queso.
Note that testing is now security-supported in the introduction.

Revisione 3-8 July 2006 JavierFernández-Sanguino Peña<jfs@debian.org>

Rewrote the information on how to setup ssh chroots to clarify the different options available, thank to Bruce Park for bringing up the different mistakes in this appendix.
Fix lsof call as suggested by Christophe Sahut.
Include patches for typo fixes from Uwe Hermann.
Fix typo in reference spotted by Moritz Naumann.

Revisione 3-7 April 2006 JavierFernández-Sanguino Peña<jfs@debian.org>

Add a section on Debian Developer's best practices for security.
Ammended firewall script with comments from WhiteGhost.

Revisione 3-6 March 2006 JavierFernández-Sanguino Peña<jfs@debian.org>

Included a patch from Thomas Sjögren which describes that `noexec` works as expected with "new" kernels, adds information regarding tempfile handling, and some new pointers to external documentation.
Add a pointer to Dan Farmer's and Wietse Venema's forensic discovery web site, as suggested by Freek Dijkstra, and expanded a little bit the forensic analysis section with more pointers.
Fixed URL of Italy's CERT, thanks to Christoph Auer.
Reuse Joey Hess' information at the wiki on secure apt and introduce it in the infrastructure section.
Review sections referring to old versions (woody or potato).
Fix some cosmetic issues with patch from Simon Brandmair.
Included patches from Carlo Perassi: acl patches are obsolete, openwall patches are obsolete too, removed fixme notes about 2.2 and 2.4 series kernels, hap is obsolete (and not present in WNPP), remove references to Immunix (StackGuard is now in Novell's hands), and fix a FIXME about the use of bsign or elfsign.
Updated references to SElinux web pages to point to the Wiki (currently the most up to date source of information).
Include file tags and make a more consistent use of "MD5 sum" with a patch from Jens Seidel.
Patch from Joost van Baal improving the information on the firewall section (pointing to the wiki instead of listing all firewall packages available) (Closes: #339865).
Review the FAQ section on vulnerability stats, thanks to Carlos Galisteo de Cabo for pointing out that it was out of date.
Use the quote from the Social Contract 1.1 instead of 1.0 as suggested by Francesco Poli.

Revisione 3-5 November 2005 JavierFernández-Sanguino Peña<jfs@debian.org>

Note on the SSH section that the chroot will not work if using the nodev option in the partition and point to the latest ssh packages with the chroot patch, thanks to Lutz Broedel for pointing these issues out.
Fix typo spotted by Marcos Roberto Greiner (md5sum should be sha1sum in code snippet).
Included Jens Seidel's patch fixing a number of package names and typos.
Slightly update of the tools section, removed tools no longer available and added some new ones.

Rewrite parts of the section related to where to find this document and what formats are available (the website does provide a PDF version). Also note that copies on other sites and translations might be obsolete (many of the Google hits for the manual in other sites are actually out of date).

Revisione 3-4 August-September 2005 JavierFernández-Sanguino Peña<jfs@debian.org>

Improved the after installation security enhancements related to kernel configuration for network level protection with a sysctl.conf file provided by Will Moy.
Improved the gdm section, thanks to Simon Brandmair.
Typo fixes from Frédéric Bothamy and Simon Brandmair.
Improvements in the after installation sections related to how to generate the MD5 (or SHA-1) sums of binaries for periodic review.
Updated the after installation sections regarding checksecurity configuration (was out of date).

Revisione 3-3 June 2005 JavierFernández-Sanguino Peña<jfs@debian.org>

Added a code snippet to use grep-available to generate the list of packages depending on Perl. As requested in #302470.
Rewrite of the section on network services (which ones are installed and how to disable them).
Added more information to the honeypot deployment section mentioning useful Debian packages.

Revisione 3-2 March 2005 JavierFernández-Sanguino Peña<jfs@debian.org>

Expanded the PAM configuration limits section.
Added information on how to use pam_chroot for openssh (based on pam_chroot's README).
Fixed some minor issues reported by Dan Jacobson.
Updated the kernel patches information partially based on a patch from Carlo Perassi and also by adding deprecation notes and new kernel patches available (adamantix).
Included patch from Simon Brandmair that fixes a sentence related to login failures in terminal.
Added Mozilla/Thunderbird to the valid GPG agents as suggested by Kapolnai Richard.
Expanded the section on security updates mentioning library and kernel updates and how to detect when services need to be restarted.
Rewrote the firewall section, moved the information that applies to woody down and expand the other sections including some information on how to manually set the firewall (with a sample script) and how to test the firewall configuration.
Added some information preparing for the 3.1 release.
Added more detailed information on kernel upgrades, specifically targeted at those that used the old installation system.
Added a small section on the experimental apt 0.6 release which provides package signing checks. Moved old content to the section and also added a pointer to changes made in aptitude.
Typo fixes spotted by Frédéric Bothamy.

Revisione 3-1 January 2005 JavierFernández-Sanguino Peña<jfs@debian.org>

Added clarification to ro /usr with patch from Joost van Baal.
Apply patch from Jens Seidel fixing many typos.
FreeSWAN is dead, long live OpenSWAN.
Added information on restricting access to RPC services (when they cannot be disabled) also included patch provided by Aarre Laakso.
Update aj's apt-check-sigs script.
Apply patch Carlo Perassi fixing URLs.
Apply patch from Davor Ocelic fixing many errors, typos, urls, grammar and FIXMEs. Also adds some additional information to some sections.
Rewrote the section on user auditing, highlight the usage of script which does not have some of the issues associated to shell history.

Revisione 3-0 December 2004 JavierFernández-Sanguino Peña<jfs@debian.org>

Rewrote the user-auditing information and include examples on how to use script.

Revisione 2-99 March 2004 JavierFernández-Sanguino Peña<jfs@debian.org>
Added information on references in DSAs and CVE-Compatibility.
Added information on apt 0.6 (apt-secure merge in experimental).
Fixed location of Chroot daemons HOWTO as suggested by Shuying Wang.
Changed APACHECTL line in the Apache chroot example (even if its not used at all) as suggested by Leonard Norrgard.
Added a footnote regarding hardlink attacks if partitions are not setup properly.
Added some missing steps in order to run bind as named as provided by Jeffrey Prosa.
Added notes about Nessus and Snort out-of-dateness in woody and availability of backported packages.
Added a chapter regarding periodic integrity test checks.
Clarified the status of testing regarding security updates (Debian bug 233955).
Added more information regarding expected contents in securetty (since it's kernel specific).
Added pointer to snoopylogger (Debian bug 179409).
Added reference to guarddog (Debian bug 170710).
apt-ftparchive is in apt-utils, not in apt (thanks to Emmanuel Chantreau for pointing this out).
Removed jvirus from AV list.
Revisione 2-98 JavierFernández-Sanguino Peña<jfs@debian.org>
Fixed URL as suggested by Frank Lichtenheld.
Fixed PermitRootLogin typo as suggested by Stefan Lindenau.
Revisione 2-97 September 2003 JavierFernández-Sanguino Peña<jfs@debian.org>
Added those that have made the most significant contributions to this manual (please mail me if you think you should be in the list and are not).
Added some blurb about FIXME/TODOs.
Moved the information on security updates to the beginning of the section as suggested by Elliott Mitchell.
Added grsecurity to the list of kernel-patches for security but added a footnote on the current issues with it as suggested by Elliott Mitchell.
Removed loops (echo to 'all') in the kernel's network security script as suggested by Elliott Mitchell.
Added more (up-to-date) information in the antivirus section.
Rewrote the buffer overflow protection section and added more information on patches to the compiler to enable this kind of protection.
Revisione 2-96 August 2003 JavierFernández-Sanguino Peña<jfs@debian.org>
Removed (and then re-added) appendix on chrooting Apache. The appendix is now dual-licensed.
Revisione 2-95 June 2003 JavierFernández-Sanguino Peña<jfs@debian.org>
Fixed typos spotted by Leonard Norrgard.
Added a section on how to contact CERT for incident handling (Capitolo 11, *Dopo la compromissione (reazione agli incidenti)*).
More information on setting up a Squid proxy.
Added a pointer and removed a FIXME thanks to Helge H. F.
Fixed a typo (save_inactive) spotted by Philippe Faes.
Fixed several typos spotted by Jaime Robles.
Revisione 2-94 April 2003 JavierFernández-Sanguino Peña<jfs@debian.org>
Following Maciej Stachura's suggestions I've expanded the section on limiting users.
Fixed typo spotted by Wolfgang Nolte.
Fixed links with patch contributed by Ruben Leote Mendes
Added a link to David Wheeler's excellent document on the footnote about counting security vulnerabilities.
Revisione 2-93 March 2003 FrédéricSchütz<schutz@math-gen.ch>

rewrote entirely the section of ext2 attributes (lsattr/chattr)

Revisione 2-92 — February 2003 — JavierFernández-Sanguino Peña<jfs@debian.org>, FrédericSchütz<schutz@math-gen.ch>

Merge section 9.3 ("useful kernel patches") into section 4.13 ("Adding kernel patches"), and added some content.
Added a few more TODOs.
Added information on how to manually check for updates and also about cron-apt. That way Tiger is not perceived as the only way to do automatic update checks.
Slightly rewrite of the section on executing a security updates due to Jean-Marc Ranger comments.
Added a note on Debian's installation (which will suggest the user to execute a security update right after installation).

Revisione 2-91 — January/February 2003 — JavierFernández-Sanguino Peña<jfs@debian.org>

Added a patch contributed by Frédéric Schütz.
Added a few more references on capabilities thanks to Frédéric.
Slight changes in the bind section adding a reference to BIND's 9 online documentation and proper references in the first area (Hi Pedro!).
Fixed the changelog date - new year :-).
Added a reference to Colin's articles for the TODOs.
Removed reference to old ssh+chroot patches.
More patches from Carlo Perassi.
Typo fixes (recursive in Bind is recursion), pointed out by Maik Holtkamp.

Revisione 2-9 — December 2002 — JavierFernández-Sanguino Peña<jfs@debian.org>

Reorganized the information on chroot (merged two sections, it didn't make much sense to have them separated).
Added the notes on chrooting Apache provided by Alexandre Ratti.
Applied patches contributed by Guillermo Jover.

Revisione 2-8 — JavierFernández-Sanguino Peña<jfs@debian.org>

Applied patches from Carlo Perassi, fixes include: re-wrapping the lines, URL fixes, and fixed some FIXMEs.
Updated the contents of the Debian security team FAQ.
Added a link to the Debian security team FAQ and the Debian Developer's reference, the duplicated sections might (just might) be removed in the future.
Fixed the hand-made auditing section with comments from Michal Zielinski.
Added links to wordlists (contributed by Carlo Perassi).
Fixed some typos (still many around).
Fixed TDP links as suggested by John Summerfield.

Revisione 2-7 — JavierFernández-Sanguino Peña<jfs@debian.org>

Some typo fixes contributed by Tuyen Dinh, Bartek Golenko and Daniel K. Gebhart.
Note regarding /dev/kmem rootkits contributed by Laurent Bonnaud.
Fixed typos and FIXMEs contributed by Carlo Perassi.

Revisione 2-6 — September 2002 — CrisTillman<tillman@voice-trak.com>

Changed around to improve grammar/spelling.
s/host.deny/hosts.deny/ (1 place).
Applied Larry Holish's patch (quite big, fixes a lot of FIXMEs).

Revisione 2-5.1 — September 2002 — JavierFernández-Sanguino Peña<jfs@debian.org>

Fixed minor typos submitted by Thiemo Nagel.

Added a footnote suggested by Thiemo Nagel.
Fixed an URL link.

Revisione 2-5.0 August 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Applied a patch contributed by Philipe Gaspar regarding the Squid which also kills a FIXME.
Yet another FAQ item regarding service banners taken from the debian-security mailing list (thread "Telnet information" started 26th July 2002).
Added a note regarding use of CVE cross references in the *How much time does the Debian security team...* FAQ item.
Added a new section regarding ARP attacks contributed by Arnaud "Arhuman" Assad.
New FAQ item regarding dmesg and console login by the kernel.
Small tidbits of information to the signature-checking issues in packages (it seems to not have gotten past beta release).
New FAQ item regarding vulnerability assessment tools false positives.
Added new sections to the chapter that contains information on package signatures and reorganized it as a new *Debian Security Infrastructure* chapter.
New FAQ item regarding Debian vs. other Linux distributions.
New section on mail user agents with GPG/PGP functionality in the security tools chapter.
Clarified how to enable MD5 passwords in woody, added a pointer to PAM as well as a note regarding the max definition in PAM.
Added a new appendix on how to create chroot environments (after fiddling a bit with makejail and fixing, as well, some of its bugs), integrated duplicate information in all the appendix.
Added some more information regarding SSH chrooting and its impact on secure file transfers. Some information has been retrieved from the debian-security mailing list (June 2002 thread: *secure file transfers*).
New sections on how to do automatic updates on Debian systems as well as the caveats of using testing or unstable regarding security updates.
New section regarding keeping up to date with security patches in the *Before compromise* section as well as a new section about the debian-security-announce mailing list.
Added information on how to automatically generate strong passwords.
New section regarding login of idle users.
Reorganized the securing mail server section based on the *Secure/hardened/minimal Debian (or "Why is the base system the way it is?")* thread on the debian-security mailing list (May 2002).
Reorganized the section on kernel network parameters, with information provided in the debian-security mailing list (May 2002, *syn flood attacked?* thread) and added a new FAQ item as well.
New section on how to check users passwords and which packages to install for this.
New section on PPTP encryption with Microsoft clients discussed in the debian-security mailing list (April 2002).
Added a new section describing what problems are there when binding any given service to a specific IP address, this information was written based on the Bugtraq mailing list in the thread: *Linux kernel 2.4 "weak end host" issue (previously discussed on debian-security as "arp problem")* (started on May 9th 2002 by Felix von Leitner).
Added information on ssh protocol version 2.
Added two subsections related to Apache secure configuration (the things specific to Debian, that is).
Added a new FAQ related to raw sockets, one related to /root, an item related to users' groups and another one related to log and configuration files permissions.
Added a pointer to a bug in libpam-cracklib that might still be open... (need to check).
Added more information regarding forensics analysis (pending more information on packet inspection tools such as tcpflow).
Changed the "what should I do regarding compromise" into a bullet list and included some more stuff.
Added some information on how to set up the Xscreensaver to lock the screen automatically after the configured timeout.
Added a note related to the utilities you should not install in the system. Included a note regarding Perl and why it cannot be easily removed in Debian. The idea came after reading Intersect's documents regarding Linux hardening.

Added information on lvm and journalling file systems, ext3 recommended. The information there might be too generic, however.
Added a link to the online text version (check).
Added some more stuff to the information on firewalling the local system, triggered by a comment made by Hubert Chan in the mailing list.
Added more information on PAM limits and pointers to Kurt Seifried's documents (related to a post by him to Bugtraq on April 4th 2002 answering a person that had ``discovered'' a vulnerability in Debian GNU/Linux related to resource starvation).
As suggested by Julián Muñoz, provided more information on the default Debian umask and what a user can access if given a shell in the system (scary, huh?).
Included a note in the BIOS password section due to a comment from Andreas Wohlfeld.
Included patches provided by Alfred E. Heggestad fixing many of the typos still present in the document.
Added a pointer to the changelog in the Credits section since most people who contribute are listed here (and not there).
Added a few more notes to the chattr section and a new section after installation talking about system snapshots. Both ideas were contributed by Kurt Pomeroy.
Added a new section after installation just to remind users to change the boot-up sequence.
Added some more TODO items provided by Korn Andras.
Added a pointer to the NIST's guidelines on how to secure DNS provided by Daniel Quinlan.
Added a small paragraph regarding Debian's SSL certificates infrastructure.
Added Daniel Quinlan's suggestions regarding ssh authentication and exim's relay configuration.
Added more information regarding securing bind including changes suggested by Daniel Quinlan and an appendix with a script to make some of the changes commented on in that section.
Added a pointer to another item regarding Bind chrooting (needs to be merged).
Added a one liner contributed by Cristian Ionescu-Idbohrn to retrieve packages with tcpwrappers support.
Added a little bit more info on Debian's default PAM setup.
Included a FAQ question about using PAM to provide services without shell accounts.
Moved two FAQ items to another section and added a new FAQ regarding attack detection (and compromised systems).
Included information on how to set up a bridge firewall (including a sample Appendix). Thanks to Francois Bayart who sent this to me in March.
Added a FAQ regarding the syslogd's *MARK heartbeat* from a question answered by Noah Meyerhans and Alain Tesio in December 2001.
Included information on buffer overflow protection as well as some information on kernel patches.
Added more information (and reorganized) the firewall section. Updated the information regarding the iptables package and the firewall generators available.
Reorganized the information regarding log checking, moved logcheck information from host intrusion detection to that section.
Added some information on how to prepare a static package for bind for chrooting (untested).
Added a FAQ item regarding some specific servers/services (could be expanded with some of the recommendations from the debian-security list).
Added some information on RPC services (and when it's necessary).
Added some more information on capabilities (and what lcap does). Is there any good documentation on this? I haven't found any documentation on my 2.4 kernel.
Fixed some typos.

Revisione 2-4	June 2002	JavierFernández-Sanguino Peña<jfs@debian.org>

Rewritten part of the BIOS section.

Revisione 2-3.1	April 2002	JavierFernández-Sanguino Peña<jfs@debian.org>

Wrapped most file locations with the file tag.
Fixed typo noticed by Edi Stojicevi.
Slightly changed the remote audit tools section.
Added some todo items.

Added more information regarding printers and cups config file (taken from a thread on debian-security).
Added a patch submitted by Jesus Climent regarding access of valid system users to Proftpd when configured as anonymous server.
Small change on partition schemes for the special case of mail servers.
Added Hacking Linux Exposed to the books section.
Fixed directory typo noticed by Eduardo Pérez Ureta.
Fixed /etc/ssh typo in checklist noticed by Edi Stojicevi.

Revisione 2-3.0 April 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Fixed location of dpkg conffile.
Remove Alexander from contact information.
Added alternate mail address.
Fixed Alexander mail address (even if commented out).
Fixed location of release keys (thanks to Pedro Zorzenon for pointing this out).

Revisione 2-2 April 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Fixed typos, thanks to Jamin W. Collins.
Added a reference to apt-extracttemplate manpage (documents the APT::ExtractTemplate config).
Added section about restricted SSH. Information based on that posted by Mark Janssen, Christian G. Warden and Emmanuel Lacour on the debian-security mailing list.
Added information on antivirus software.
Added a FAQ: su logs due to the cron running as root.

Revisione 2-1 April 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Changed FIXME from lshell thanks to Oohara Yuuma.
Added package to sXid and removed comment since it *is* available.
Fixed a number of typos discovered by Oohara Yuuma.
ACID is now available in Debian (in the acidlab package) thanks to Oohara Yuuma for noticing.
Fixed LinuxSecurity links (thanks to Dave Wreski for telling).

Revisione 2-0 March 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Converted the HOWTO into a Manual (now I can properly say RTFM).
Added more information regarding tcp wrappers and Debian (now many services are compiled with support for them so it's no longer an inetd issue).
Clarified the information on disabling services to make it more consistent (rpc info still referred to update-rc.d).
Added small note on lprng.
Added some more info on compromised servers (still very rough).
Fixed typos reported by Mark Bucciarelli.
Added some more steps in password recovery to cover the cases when the admin has set paranoid-mode=on.
Added some information to set paranoid-mode=on when login in console.
New paragraph to introduce service configuration.
Reorganized the *After installation* section so it is more broken up into several issues and it's easier to read.
Wrote information on how to set up firewalls with the standard Debian 3.0 setup (iptables package).
Small paragraph explaining why installing connected to the Internet is not a good idea and how to avoid this using Debian tools.
Small paragraph on timely patching referencing to IEEE paper.
Appendix on how to set up a Debian snort box, based on what Vladimir sent to the debian-security mailing list (September 3rd 2001).
Information on how logcheck is set up in Debian and how it can be used to set up HIDS.
Information on user accounting and profile analysis.
Included apt.conf configuration for read-only /usr copied from Olaf Meeuwissen's post to the debian-security mailing list.

New section on VPN with some pointers and the packages available in Debian (needs content on how to set up the VPNs and Debian-specific issues), based on Jaroslaw Tabor's and Samuli Suonpaa's post to debian-security.
Small note regarding some programs to automatically build chroot jails.
New FAQ item regarding identd based on a discussion in the debian-security mailing list (February 2002, started by Johannes Weiss).
New FAQ item regarding inetd based on a discussion in the debian-security mailing list (February 2002).
Introduced note on rcconf in the "disabling services" section.
Varied the approach regarding LKM, thanks to Philipe Gaspar.
Added pointers to CERT documents and Counterpane resources.

Revisione 1-99 January 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Added a new FAQ item regarding time to fix security vulnerabilities.
Reorganized FAQ sections.
Started writing a section regarding firewalling in Debian GNU/Linux (could be broadened a bit).
Fixed typos sent by Matt Kraai.
Fixed DNS information.
Added information on whisker and nbtscan to the auditing section.
Fixed some wrong URLs.

Revisione 1-98 January 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Added a new section regarding auditing using Debian GNU/Linux.
Added info regarding finger daemon taken from the security mailing list.

Revisione 1-97 January 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Fixed link for Linux Trustees.
Fixed typos (patches from Oohara Yuuma and Pedro Zorzenon).

Revisione 1-96 December 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Reorganized service installation and removal and added some new notes.
Added some notes regarding using integrity checkers as intrusion detection tools.
Added a chapter regarding package signatures.

Revisione 1-95 December 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Added notes regarding Squid security sent by Philipe Gaspar.
Fixed rootkit links thanks to Philipe Gaspar.

Revisione 1-94 November 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Added some notes regarding Apache and Lpr/lpng.
Added some information regarding noexec and read-only partitions.
Rewrote how users can help in Debian security issues (FAQ item).

Revisione 1-93 November 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Fixed location of mail program.
Added some new items to the FAQ.

Revisione 1-92 October 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Added a small section on how Debian handles security.
Clarified MD5 passwords (thanks to `rocky').
Added some more information regarding harden-X from Stephen van Egmond.
Added some new items to the FAQ.

Revisione 1-91 October 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Added some forensics information sent by Yotam Rubin.

Added information on how to build a honeynet using Debian GNU/Linux.
Added some more TODOS.
Fixed more typos (thanks Yotam!).

Revisione 1-9 — October 2001 — JavierFernández-Sanguino Peña<jfs@debian.org>

Added patch to fix misspellings and some new information (contributed by Yotam Rubin).
Added references to other online (and offline) documentation both in a section (see sezione chiamata «Conoscere i problemi generali di sicurezza») by itself and inline in some sections.
Added some information on configuring Bind options to restrict access to the DNS server.
Added information on how to automatically harden a Debian system (regarding the harden package and bastille).
Removed some done TODOs and added some new ones.

Revisione 1-8 — October 2001 — JavierFernández-Sanguino Peña<jfs@debian.org>

Added the default user/group list provided by Joey Hess to the debian-security mailing list.
Added information on LKM root-kits (sezione chiamata «Moduli del kernel caricabili (LKM)») contributed by Philipe Gaspar.
Added information on Proftp contributed by Emmanuel Lacour.
Recovered the checklist Appendix from Era Eriksson.
Added some new TODO items and removed other fixed ones.
Manually included Era's patches since they were not all included in the previous version.

Revisione 1-7 — September 2001 — JavierFernández-Sanguino Peña<jfs@debian.org>, EraEriksson<era@iki.fi>

Typo fixes and wording changes.
Minor changes to tags in order to keep on removing the tt tags and substitute prgn/package tags for them.

Revisione 1-6 — August 2001 — JavierFernández-Sanguino Peña<jfs@debian.org>

Added pointer to document as published in the DDP (should supersede the original in the near future).
Started a mini-FAQ (should be expanded) with some questions recovered from my mailbox.
Added general information to consider while securing.
Added a paragraph regarding local (incoming) mail delivery.
Added some pointers to more information.
Added information regarding the printing service.
Added a security hardening checklist.
Reorganized NIS and RPC information.
Added some notes taken while reading this document on my new Visor :).
Fixed some badly formatted lines.
Fixed some typos.
Added a Genius/Paranoia idea contributed by Gaby Schilders.

Revisione 1-5 — May 2001 — JavierFernández-Sanguino Peña<jfs@debian.org>, JosipRodin<joy@debian.org>

Added paragraphs related to BIND and some FIXMEs.

Revisione 1-4 — May 2001 — JavierFernández-Sanguino Peña<jfs@debian.org>

Small setuid check paragraph
Various minor cleanups.
Found out how to use `sgml2txt -f` for the txt version.

Revisione 1-3 — March 2001 — JavierFernández-Sanguino Peña<jfs@debian.org>

Added a security update after installation paragraph.
Added a proftpd paragraph.
This time really wrote something about XDM, sorry for last time.

Revisione 1-2	December 2000	JavierFernández-Sanguino Peña<jfs@debian.org>
Lots of grammar corrections by James Treacy, new XDM paragraph.		
Revisione 1-1	December 2000	JavierFernández-Sanguino Peña<jfs@debian.org>
Typo fixes, miscellaneous additions.		
Revisione 1-0	December 2000	JavierFernández-Sanguino Peña<jfs@debian.org>
Initial release.		

Appendice B. Appendix

Il processo di blindatura passo-passo

Di seguito viene riportata una procedura passo-passo, post-installazione, per blindare un sistema Debian GNU/linux 2.2. Questo è un possibile approccio a tale procedura ed è orientato a blindare i servizi di rete. Viene incluso per mostrare l'intero processo che potrebbe essere usato durante la configurazione. Vedete in sezione chiamata «Verifica della configurazione».

- Install the system, taking into account the information regarding partitioning included earlier in this document. After base installation, go into custom install. Do not select task packages.
- Utilizzando **dselect**, rimuovete tutti i pacchetti selezionati ma non necessari prima di effettuare l'installazione con il comando [I]nstall. Mantenete il minimo numero di pacchetti per il sistema.
- Aggiornate tutti i programmi dal più recente pacchetto disponibile da security.debian.org come spiegato precedentemente in sezione chiamata «Eseguire un aggiornamento per la sicurezza».
- Applicate i suggerimenti presentati in questo manuale al riguardo delle quote utente, definizioni di login e **lilo**.
- Compilate una lista di servizi attivi al momento sul sistema. Eseguite:

  ```
  $ ps aux
  $ netstat -pn -l -A inet
  # /usr/sbin/lsof -i | grep LISTEN
  ```

  You will need to install lsof-2.2 for the third command to work (run it as root). You should be aware that **lsof** can translate the word LISTEN to your locale settings.
- Per rimuovere i servizi non necessari, determinate inizialmente quale pacchetto fornisca il servizio e come si avvii. Questo può essere ottenuto controllando quali programmi ascoltano su di un socket. Il seguente script di shell, che utilizza i programmi **lsof** e **dpkg**, fa proprio questo:

  ```
  #!/bin/sh
  # FIXME: this is quick and dirty; replace with a more robust script snippet
  for i in `sudo lsof -i | grep LISTEN | cut -d " " -f 1 |sort -u` ; do
    pack=`dpkg -S $i |grep bin |cut -f 1 -d : | uniq`
    echo "Service $i is installed by $pack";
    init=`dpkg -L $pack |grep init.d/ `
    if [ ! -z "$init" ]; then
      echo "and is run by $init"
    fi
  done
  ```
- Una volta che avete individuato un qualsiasi servizio che non volete fornire, rimuovete il pacchetto che lo genera (con **dpkg --purge**), o disabilitate la partenza automatica del servizio all'avvio del sistema utilizzando **update-rc.d** (vedete in sezione chiamata «Disabilitare i servizi attivi in modalità demone»).
- Per i servizi avviati da inetd (lanciati tramite il superdemone), verificate che i servizi siano abilitati in `/etc/inetd.conf`, utilizzando:

```
$ grep -v "^#" /etc/inetd.conf | sort -u
```

Per disabilitare quei servizi che non desiderate è necessario commentare le righe dello script che li avviano in `/etc/inetd.conf`, altrimenti, rimuovere il pacchetto che fornisce il servizio, o utilizzare **update-inetd**.

- Se avete servizi wrapped (quelli che usano **/usr/sbin/tcpd**), verificate che i file `/etc/hosts.allow` e `/etc/hosts.deny` siano configurati in accordo con la vostra politica di sicurezza.

- Se il server usa più di una interfaccia con l'esterno, in funzione del servizio, potreste voler limitare il servizio all'ascolto su una specifica interfaccia. Per esempio, se si volesse un accesso FTP esclusivamente dall'interno, fate in modo che il demone FTP sia in ascolto solo sull'interfaccia di gestione e non su tutte le interfacce (p.e. 0.0.0.0:21).

- Riavviate la macchina, o commutate sulla modalità monoutente e quindi tornate a quella multiutente, utilizzando i comandi:

```
# init 1
(....)
# init 2
```

- Controllate i servizi adesso disponibili e se necessario, ripetete i passi appena esposti.

- Ora installate i servizi che ritenete necessari, se non avete già agito così e configurateli appropriatamente.

- Usate il seguente comando da shell, per determinare con quale identità ogni servizio disponibile sta girando:

```
# for i in `/usr/sbin/lsof -i |grep LISTEN |cut -d " " -f 1 |sort -u`; \
> do user=`ps ef |grep $i |grep -v grep |cut -f 1 -d " "` ; \
> echo "Service $i is running as user $user"; done
```

Valutate l'opportunità di modificare questi servizi associandoli a specifici utenti/gruppi mediante gabbie **chroot** per avere maggior sicurezza. Potete farlo modificando lo script `/etc/init.d` che avvia il servizio. La maggior parte dei servizi in Debian usano **start-stop-daemon** che ha l'opzione (`--change-uid` e `--chroot`) per apportare le modifiche di cui sopra. Alcuni avvertimenti al riguardo dei servizi in **chroot**: potrebbe essere necessario mettere tutti i file installati dal pacchetto (usando dpkg -L) che fornisce il servizio, così come tutti i pacchetti da cui esso dipende, in un ambiente di tipo **chroot**. Le informazioni per configurare un ambiente **chroot** per il programma **ssh** possono essere trovate in sezione chiamata «Chroot environment for SSH».

- Ripetete i passi summenzionati al fine di verificare che girino i soli servizi desiderati e che essi stiano girando con la desiderata combinazione utente/gruppo.

- Verificate i servizi installati per controllare che funzionino come avete previsto.

- Check the system using a vulnerability assessment scanner (like nessus), in order to determine vulnerabilities in the system (i.e., misconfiguration, old services or unneeded services).

- Install network and host intrusion measures like snort and logcheck.

- Ripetete la sequenza dell'esame della rete e verificate che i sistemi di rilevamento delle intrusioni stiano funzionando correttamente.

Se siete persone realmente paranoiche dovreste valutare anche quanto segue:

- Aggiungere funzionalità "firewall" al sistema, accettando connessioni in ingresso solo per servizi offerti e limitando le connessioni uscenti alle sole autorizzate.
- Ricontrollare l'installazione con una nuova verifica di vulnerabilità usando uno scanner di rete.
- Usando un rilevatore di rete, controllare le connessioni uscenti dal sistema verso una macchina esterna e verificare che nessuna connessione trovi il modo di uscire.

FIXME: questa procedura si occupa della blindatura dei servizi, non della blindatura di sistemi a livello utente, includendo le informazioni per controllare i permessi utente, i file SETUID ed il congelamento dei cambiamenti del sistema utilizzando il filesystem ext2.

Verifica della configurazione

This appendix briefly reiterates points from other sections in this manual in a condensed checklist format. This is intended as a quick summary for someone who has already read the manual. There are other good checklists available, including Kurt Seifried's http://seifried.org/security/os/linux/20020324-securing-linux-step-by-step.html and http://www.cert.org/tech_tips/usc20_full.html.

FIXME: Questa checklist è basata sulla versione 1.4 del manuale e potrebbe aver bisogno di essere aggiornata.

- Limitate le capacità di avvio e di accesso fisico
  - Abilitate la password del BIOS
  - Disabilitate l'avvio da floppy/cdrom/ ...
  - Impostate una password per GRUB o LILO (rispettivamente `/etc/lilo.conf` o `/boot/grub/menu.lst`); verificate che i file di configurazione di LILO o GRUB siano protetti da scrittura.
- Partizionamento
  - Separate i dati scrivibili dagli utenti, i dati non di sistema e i dati run-time che cambiano velocemente, in partizioni distinte.
  - Impostate le opzioni di mount `nosuid,noexec,nodev` in `/etc/fstab` per partizioni ext2/3 che non dovrebbero bloccare binari come `/home` o `/tmp`.
- Accuratezza delle password e sicurezza per il login
  - Assegnate a root una buona password.
  - Installate ed utilizzate PAM.
    - Aggiungete il supporto per MD5 a PAM e assicuratevi che (in generale) le voci nel file `/etc/pam.d/`, che garantiscono l'accesso alla macchina, abbiano il secondo campo nel file pam.d impostato a `requisite` o `required`.
    - Modificate `/etc/pam.d/login` in modo che permetta il login solamente a root solamente in locale.
    - Inoltre segnate le tty:s autorizzate nel file `/etc/security/access.conf` e in generale configurate il file affinché le login da root siano limitate il più possibile.
    - Aggiungete pam_limits.so se volete impostare un limite per ciascun utente.

    - Modificate `/etc/pam.d/passwd`: aumentate la lunghezza minima delle passwords (6 caratteri probabilmente) e abilitate l'MD5.
    - Aggiungete il gruppo wheel a `/etc/group` se lo desiderate; aggiungete la voce pam_wheel.so group=wheel in `/etc/pam.d/su`.
    - Per personalizzare i controlli relativi a ciascun utente, usate un'apposita voce pam_listfile.so dove appropriato.
    - Create un file `/etc/pam.d/other` e configuratelo con un'alta sicurezza.
  - Impostate i limiti `/etc/security/limits.conf` (notate che `/etc/limits` non viene utilizzato se usate PAM).
  - Limitate i permessi al file `/etc/login.defs`; inoltre, se avete abilitato MD5 e/o PAM, assicuratevi di fare i cambiamenti corrispondenti.
  - Tighten up `/etc/pam.d/login`
  - Disabilitate l'accesso root via ftp in `/etc/ftpusers`.
  - Disable network root login; use su(1) or sudo(1). (consider installing sudo)
  - Usate PAM per rendere più sicuri gli accessi con login?
- Altri suggerimenti sulla sicurezza locale
  - Adattamenti del kernel (vedete in sezione chiamata «Configurare le caratteristiche di rete del kernel»).
  - Patch per il kernel (vedete in sezione chiamata «Includere le patch nel kernel»).
  - Rendete più restrittivi i permessi ai file di log (`/var/log/{last,fail}log`, i log di Apache)
  - Verificate che il controllo SETUID sia abilitato in `/etc/checksecurity.conf`.
  - Considerate la possibilità di rendere alcuni file di log append-only e alcuni file di configurazione immutabili con il comando chattr (solo per il file system ext2/3).
  - Set up file integrity (see sezione chiamata «Controllare l'integrità del file system»). Install debsums
  - Loggare tutto su una stampante locale?
  - Scrivere su un CD avviabile la propria configurazione e fare il boot da CD?
  - Disabilitare i moduli del kernel?
- Limitate l'accesso alla rete
  - Installate e configurate **ssh** (si suggerisce di impostare a No le voci PermitRootLogin e PermitEmptyPasswords nel file `/etc/ssh/sshd_config`; notate anche gli altri suggerimenti nel testo).
  - Considerate la possibilità di disabilitare o rimuovere **in.telnetd**, se installato.
  - In linea di massima, disabilitate i servizi inutili in `/etc/inetd.conf` usando **update-inetd --disable** (oppure disabilitate **inetd** completamente, o ancora usate un sostituto come ad esempio **xinetd** o **rlinetd**).

    - Disabilitate altri servizi di rete inutili; ftp, DNS, WWW etc. non dovrebbero essere eseguiti se non sono necessari e peraltro, tenuti regolarmente sotto controllo.

    - Per i servizi di cui avete bisogno, non vi limitate ad usare i programmi più comuni ma cercatene versioni più sicure contenute all'interno di Debian (o di altre fonti). Qualsiasi cosa finiate per eseguire, assicuratevi di capirne i rischi.

    - Impostate gabbie **chroot** per utenti e demoni esterni.

    - Configure firewall and tcpwrappers (i.e. hosts_access(5)); note trick for `/etc/hosts.deny` in text.

    - Se eseguite ftp, impostate il server ftpd per essere eseguito sempre in **chroot** nella home directory dell'utente.

    - Se usate X, disabilitate l'autenticazione xhost ed usate **ssh** come sostituto; ancora meglio, disabilitate, se possibile, la possibilità di autenticarsi in X da remoto (aggiungete -nolisten tcp alla riga di comando di X e disabilitate XDMCP nel file `/etc/X11/xdm/xdm-config` impostando a 0 la requestPort).

    - Disabilitate l'accesso dall'esterno alle stampanti.

    - Effettuate il tunneling di qualsiasi sezione POP o IMAP attraverso SSL o **ssh**; installate stunnel se volete fornire questo servizio agli utenti remoti del servizio di posta.

    - Impostate un log host e configurate tutti gli altri host a spedire i log a questo host (`/etc/syslog.conf`).

    - Rendete sicuri BIND, Sendmail e altri demoni complessi (eseguiteli in una gabbia **chroot**; eseguiteli come pseudo-utente non-root).

    - Installate tiger o un simile strumento di logging.

    - Install snort or a similar network intrusion detection tool.v

    - Se possibile, fate a meno di NIS ed RPC (disabilitate portmap).

- Documenti sulle politiche adottate

    - Educate gli utenti a comprendere i perché ed i come delle vostre politiche. Quando proibite qualcosa che è regolarmente disponibile su altri sistemi, fornite documentazione che spieghi come raggiungere i risultati voluti utilizzando altri mezzi più sicuri.

    - Proibite l'uso di protocolli che usano password in chiaro (**telnet**, **rsh** e simili; ftp, imap, http, ...).

    - Proibite i programmi che usano SVGAlib.

    - Usate la gestione delle quote disco.

- Tenetevi informati circa le notizie riguardanti la sicurezza

    - Iscrivetevi a mailing list di sicurezza.

    - Configure apt for security updates -- add to `/etc/apt/sources.list` an entry (or entries) for http://security.debian.org/

- Ricordatevi anche di eseguire periodicamente **apt-get update; apt-get upgrade** (potreste farlo eseguire ad un job **cron**?) come spiegato in sezione chiamata «Eseguire un aggiornamento per la sicurezza».

Configurazione ed installazione di un sistema autonomo IDS

You can easily set up a dedicated Debian system as a stand-alone Intrusion Detection System using snort and a web-based interface to analyse the intrusion detection alerts:

- Installare un sistema di base Debian senza selezionare pacchetti aggiuntivi.
- Installare una versione di Snort con supporto ai database e configurare l'IDS per registrare gli avvisi in un database.
- Scaricare ed installare BASE (Basic Analysis and Security Engine), o ACID (Analysis Console for Intrusion Databases). Configurare il software scelto per utilizzare lo stesso database usato da Snort.
- Scaricare ed installare i pacchetti necessari[1].

BASE is currently packaged for Debian in acidbase and ACID is packaged as acidlab[2]. Both provide a graphical WWW interface to Snort's output.

Besides the base installation you will also need a web server (such as apache), a **PHP** interpreter and a relational database (such postgresql or mysql) where Snort will store its alerts.

Questo sistema dovrebbe essere configurato con almeno due interfacce: una connessa ad una LAN gestionale (per avere accesso ai risultati e gestire il sistema) e l'altra senza indirizzo IP collegata al segmento di rete che state analizzando. Dovreste configurare il server web per rimanere in ascolto solo sull'interfaccia connessa alla LAN gestionale.

Dovreste configurare entrambe le interfacce nel file di configurazione standard di Debian `/etc/network/interfaces`. Un indirizzo (quello di gestione della LAN) può essere configurato come fareste solitamente. L'altra interfaccia dovrebbe essere inizializzata all'avvio del sistema, ma senza indirizzo d'interfaccia. Potete usare la seguente definizione d'interfaccia:

```
auto eth0
iface eth0 inet manual
      up ifconfig $IFACE 0.0.0.0 up
      up ip link set $IFACE promisc on
      down ip link set $IFACE promisc off
      down ifconfig $IFACE down
```

The above configures an interface to read all the traffic on the network in a *stealth*-type configuration. This prevents the NIDS system to be a direct target in a hostile network since the sensors have no IP address on the network. Notice, however, that there have been known bugs over time in sensors part of NIDS (for example see https://lists.debian.org/debian-security-announce/2003/msg00087.html related to Snort) and remote buffer overflows might even be triggered by network packet processing.

You might also want to read the http://www.faqs.org/docs/Linux-HOWTO/Snort-Statistics-HOWTO.html and the documentation available at the https://www.snort.org/#documents.

[1] Solitamente i pacchetti di cui avete bisogno vengono installati come dipendenza.
[2] It can also be downloaded from http://www.cert.org/kb/acid/, http://acidlab.sourceforge.net or http://www.andrew.cmu.edu/~rdanyliw/snort/.

Impostare un bridge firewall

This information was contributed by Francois Bayart in order to help users set up a Linux bridge/firewall with the 2.4.x kernel and iptables. Kernel patches are no more needed as the code was made standard part of the Linux kernel distribution.

Per configurare il kernel con il necessario supporto, eseguite `make menuconfig` o `make xconfig`. Nella sezione *Networking options*, abilitate le seguenti opzioni:

```
[*] Network packet filtering (replaces ipchains)
[ ]   Network packet filtering debugging (NEW)
<*> 802.1d Ethernet Bridging
[*]   netfilter (firewalling) support (NEW)
```

Attenzione: dovete disabilitare questa opzione se volete applicare delle regole di firewall, altrimenti **iptables** non funzionerà.

```
[ ]   Network packet filtering debugging (NEW)
```

Next, add the correct options in the section *IP: Netfilter Configuration*. Then, compile and install the kernel. If you want to do it the *Debian way*, install kernel-package and run **make-kpkg** to create a custom Debian kernel package you can install on your server using dpkg. Once the new kernel is compiled and installed, install the bridge-utils package.

Una volta completati questi passaggi, potete completare la configurazione del vostro bridge. La sezione successiva mostra due diverse possibili configurazioni per il bridge, ognuna con un'ipotetica mappa di rete ed i comandi necessari.

Un bridge con funzionalità NAT e firewall

La prima configurazione utilizza il bridge come un firewall con traslazione degli indirizzi di rete (NAT), che protegge un server ed i client della LAN interna. Un diagramma della configurazione di rete viene mostrato qui sotto:

```
Internet ---- router ( 62.3.3.25 ) ---- bridge (62.3.3.26 gw 62.3.3.25 / 192.168.0
                                          |
                                          |
                                          |---- WWW Server (62.3.3.27 gw 62.3.3.25
                                          |
                                          |
                                        LAN --- Zipowz (192.168.0.2 gw 192.168.0.
```

Le seguenti istruzioni mostrano come sia possibile configurare questo bridge.

```
# Create the interface br0
/usr/sbin/brctl addbr br0

# Add the Ethernet interface to use with the bridge
/usr/sbin/brctl addif br0 eth0
/usr/sbin/brctl addif br0 eth1
```

```
# Start up the Ethernet interface
/sbin/ifconfig eth0 0.0.0.0
/sbin/ifconfig eth1 0.0.0.0

# Configure the bridge ethernet
# The bridge will be correct and invisible ( transparent firewall ).
# It's hidden in a traceroute and you keep your real gateway on the
# other computers. Now if you want you can config a gateway on your
# bridge and choose it as your new gateway for the other computers.

/sbin/ifconfig br0 62.3.3.26 netmask 255.255.255.248 broadcast 62.3.3.31

# I have added this internal IP to create my NAT
ip addr add 192.168.0.1/24 dev br0
/sbin/route add default gw 62.3.3.25
```

Bridge con funzionalità di firewall

Una seconda configurazione possibile è un sistema configurato come un firewall trasparente per una LAN con spazio di indirizzi IP pubblici.

```
Internet ---- router (62.3.3.25) ---- bridge (62.3.3.26)
                                          |
                                          |
                                          |---- WWW Server (62.3.3.28 gw 62.3.3.25)
                                          |
                                          |
                                          |---- Mail Server (62.3.3.27 gw 62.3.3.25)
```

Le seguenti istruzioni mostrano come sia possibile configurare questo bridge.

```
# Create the interface br0
/usr/sbin/brctl addbr br0

# Add the Ethernet interface to use with the bridge
/usr/sbin/brctl addif br0 eth0
/usr/sbin/brctl addif br0 eth1

# Start up the Ethernet interface
/sbin/ifconfig eth0 0.0.0.0
/sbin/ifconfig eth1 0.0.0.0

# Configure the bridge Ethernet
# The bridge will be correct and invisible ( transparent firewall ).
# It's hidden in a traceroute and you keep your real gateway on the
# other computers. Now if you want you can config a gateway on your
# bridge and choose it as your new gateway for the other computers.

/sbin/ifconfig br0 62.3.3.26 netmask 255.255.255.248 broadcast 62.3.3.31
```

Se eseguite un traceroute verso il Linux Mail Server, non vedrete il bridge. Se volete accedere al bridge con **ssh**, dovete avere un gateway, altrimenti dovreste prima connettervi ad un altro server, come il "Mail Server" ed in seguito connettervi al bridge tramite la scheda di rete interna.

Regole base di IPtables

Questo è un esempio delle regole base che potreste usare indistintamente per queste due configurazioni.

Esempio B.1. Regole base di IPtables

```
iptables -F FORWARD
iptables -P FORWARD DROP
iptables -A FORWARD -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 -m state --state INVALID
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Some funny rules but not in a classic Iptables sorry ...
# Limit ICMP
# iptables -A FORWARD -p icmp -m limit --limit 4/s -j ACCEPT
# Match string, a good simple method to block some VIRUS very quickly
# iptables -I FORWARD -j DROP -p tcp -s 0.0.0.0/0 -m string --string "cmd.exe"

# Block all MySQL connection just to be sure
iptables -A FORWARD -p tcp -s 0/0 -d 62.3.3.0/24 --dport 3306 -j DROP

# Linux Mail Server Rules

# Allow FTP-DATA (20), FTP (21), SSH (22)
iptables -A FORWARD -p tcp -s 0.0.0.0/0 -d 62.3.3.27/32 --dport 20:22 -j ACCEPT

# Allow the Mail Server to connect to the outside
# Note: This is *not* needed for the previous connections
# (remember: stateful filtering) and could be removed.
iptables -A FORWARD -p tcp -s 62.3.3.27/32 -d 0/0 -j ACCEPT

# WWW Server Rules

# Allow HTTP ( 80 ) connections with the WWW server
iptables -A FORWARD -p tcp -s 0.0.0.0/0 -d 62.3.3.28/32 --dport 80 -j ACCEPT

# Allow HTTPS ( 443 ) connections with the WWW server
iptables -A FORWARD -p tcp -s 0.0.0.0/0 -d 62.3.3.28/32 --dport 443 -j ACCEPT

# Allow the WWW server to go out
# Note: This is *not* needed for the previous connections
# (remember: stateful filtering) and could be removed.
iptables -A FORWARD -p tcp -s 62.3.3.28/32 -d 0/0 -j ACCEPT
```

Script di esempio per modificare l'installazione predefinita di Bind

This script automates the procedure for changing the **bind** version 8 name server's default installation so that it does *not* run as the superuser. Notice that **bind** version 9 in Debian already does this by default [3] , and you are much better using that version than **bind** version 8.

[3] Since version 9.2.1-5. That is, since Debian release *sarge*.

Questo script viene qui riportato per ragioni storiche e per mostrare come si possa automatizzare questo tipo di cambiamenti per tutto il sistema. Lo script creerà l'utente ed i gruppi definiti dal server dei nomi e modificherà entrambi i file `/etc/default/bind` e `/etc/init.d/bind`, così il programma verrà eseguito con quell'utente. Usate estrema cautela poiché non è stato collaudato estensivamente.

You can also create the users manually and use the patch available for the default init.d script attached to http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=157245.

```
#!/bin/sh
# Change the default Debian bind v8 configuration to have it run
# with a non-root user and group.
#
# DO NOT USER this with version 9, use debconf for configure this instead
#
# WARN: This script has not been tested thoroughly, please
# verify the changes made to the INITD script

# (c) 2002 Javier Fernandez-Sanguino Pena
#
#    This program is free software; you can redistribute it and/or modify
#    it under the terms of the GNU General Public License as published by
#    the Free Software Foundation; either version 1, or (at your option)
#    any later version.
#
#    This program is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.
#
#     Please see the file `COPYING' for the complete copyright notice.
#

restore() {
# Just in case, restore the system if the changes fail
  echo "WARN: Restoring to the previous setup since I'm unable to properly chang
  echo "WARN: Please check the $INITDERR script."
  mv $INITD $INITDERR
  cp $INITDBAK $INITD
}


USER=named
GROUP=named
INITD=/etc/init.d/bind
DEFAULT=/etc/default/bind
INITDBAK=$INITD.preuserchange
INITDERR=$INITD.changeerror
AWKS="awk ' /\/usr\/sbin\/ndc reload/ { print \"stop; sleep 2; start;\"; noprint

[ `id -u` -ne 0 ] && {
  echo "This program must be run by the root user"
  exit 1
}
```

```
RUNUSER=`ps eo user,fname |grep named |cut -f 1 -d " "`

if [ "$RUNUSER" = "$USER" ]
then
  echo "WARN: The name server running daemon is already running as $USER"
  echo "ERR:  This script will not do any changes to your setup."
  exit 1
fi
if [ ! -f "$INITD" ]
then
  echo "ERR:  This system does not have $INITD (which this script tries to chang
  RUNNING=`ps eo fname |grep named`
  [ -z "$RUNNING" ] && \
    echo "ERR:  In fact the name server daemon is not even running (is it instal
  echo "ERR:  No changes will be made to your system"
  exit 1
fi

# Check if there are options already setup
if [ -e "$DEFAULT" ]
then
  if grep -q ^OPTIONS $DEFAULT; then
    echo "ERR: The $DEFAULT file already has options set."
    echo "ERR:  No changes will be made to your system"
  fi
fi

# Check if named group exists
if [ -z "`grep $GROUP /etc/group`" ]
then
  echo "Creating group $GROUP:"
  addgroup $GROUP
else
  echo "WARN: Group $GROUP already exists. Will not create it"
fi
# Same for the user
if [ -z "`grep $USER /etc/passwd`" ]
then
  echo "Creating user $USER:"
  adduser --system --home /home/$USER \
    --no-create-home --ingroup $GROUP \
    --disabled-password --disabled-login $USER
else
  echo "WARN: The user $USER already exists. Will not create it"
fi

# Change the init.d script

# First make a backup (check that there is not already
# one there first)
if [ ! -f $INITDBAK ]
then
  cp $INITD $INITDBAK
```

```
  fi

  # Then use it to change it
  cat $INITDBAK |
  eval $AWKS > $INITD

  # Now put the options in the /etc/default/bind file:
  cat >>$DEFAULT <<EOF
# Make bind run with the user we defined
OPTIONS="-u $USER -g $GROUP"
EOF

  echo "WARN: The script $INITD has been changed, trying to test the changes."
  echo "Restarting the named daemon (check for errors here)."

  $INITD restart
  if [ $? -ne 0 ]
  then
    echo "ERR:  Failed to restart the daemon."
    restore
    exit 1
  fi

  RUNNING=`ps eo fname |grep named`
  if [ -z "$RUNNING" ]
  then
    echo "ERR:  Named is not running, probably due to a problem with the changes."
    restore
    exit 1
  fi

  # Check if it's running as expected
  RUNUSER=`ps eo user,fname |grep named |cut -f 1 -d " "`

  if [ "$RUNUSER" = "$USER" ]
  then
    echo "All has gone well, named seems to be running now as $USER."
  else
    echo "ERR:  The script failed to automatically change the system."
    echo "ERR:  Named is currently running as $RUNUSER."
    restore
    exit 1
  fi

  exit 0
```

Lo script precedente, che funziona sulla versione 8 di **bind** per Woody (Debian 3.0), modificherà il file in init.d dopo aver creato l'utente ed il gruppo 'named'.

Aggiornamenti di sicurezza protetti da un firewall

Dopo un'installazione standard, la sicurezza di un sistema potrebbe presentare ancora delle vulnerabilità. A meno che voi non scarichiate gli aggiornamenti da un altro sistema (o abbiate fatto il mirror di security.debian.org per un uso locale), il sistema dovrà essere collegato ad internet per i download.

However, as soon as you connect to the Internet you are exposing this system. If one of your local services is vulnerable, you might be compromised even before the update is finished! This may seem paranoid but, in fact, analysis from the http://www.honeynet.org has shown that systems can be compromised in less than three days, even if the system is not publicly known (i.e., not published in DNS records).

Quando si esegue un aggiornamento, su di un sistema non protetto da un altro sistema esterno, come un firewall, è possibile configurare adeguatamente il vostro firewall locale per limitare le connessioni alle sole riguardanti gli aggiornamenti per la sicurezza. L'esempio qui di seguito mostra come configurare un tale firewall, per autorizzare solo le connessioni da security.debian.org e registrare tutte le altre.

Il seguente esempio può essere utilizzato per impostare un insieme di regole restrittive del firewall. Eseguite questi comandi da una console locale (non da remoto) per ridurre il rischio di rimanere tagliati fuori da un eventuale blocco del sistema.

```
# iptables -F
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
# iptables -A OUTPUT -d security.debian.org --dport 80 -j ACCEPT
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# iptables -A INPUT -p icmp -j ACCEPT
# iptables -A INPUT -j LOG
# iptables -A OUTPUT -j LOG
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
# iptables -P OUTPUT DROP
# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0          state RELATED,ESTABL
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
LOG        all  --  anywhere             anywhere           LOG level warning

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     80   --  anywhere             security.debian.org
```

```
LOG        all  --  anywhere             anywhere           LOG level warning
```

Note: Using a *DROP* policy in the INPUT chain is the most correct thing to do, but be *very* careful when doing this after flushing the chain from a remote connection. When testing firewall rulesets from a remote location it is best if you run a script with the firewall ruleset (instead of introducing the ruleset line by line through the command line) and, as a precaution, keep a backdoor[4]

Of course, you should disable any backdoors before getting the system into production. configured so that you can re-enable access to the system if you make a mistake. That way there would be no need to go to a remote location to fix a firewall ruleset that blocks you.

Questo richiede che il DNS funzioni correttamente, dato che serve per far funzionare security.debian.org. Potete aggiungere security.debian.org ad /etc/hosts ma attualmente è un CNAME a svariati host (c'è più di un mirror di sicurezza).

FIXME: ciò funzionerà solo con le URL HTTP poiché ftp potrebbe richiedere il modulo ip_conntrack_ftp module, oppure utilizzare la modalità passiva.

Chroot environment for SSH

Creating a restricted environment for SSH is a tough job due to its dependencies and the fact that, unlike other servers, SSH provides a remote shell to users. Thus, you will also have to consider the applications users will be allowed to use in the environment.

Avete due opzioni per impostare una shell remota con restrizioni:

- Confinare gli utenti ssh in un ambiente chroot; configurando correttamente il demone ssh potete chiedergli di confinare un utente in un ambiente chroot dopo l'autenticazione, subito prima che gli venga fornita una shell. Ogni utente può avere il proprio ambiente.
- Mettere il server ssh in chroot poiché, se l'applicazione ssh stessa viene messa in chroot tutti gli utenti sono in chroot all'interno dell'ambiente definito.

La prima opzione ha il vantaggio di rendere possibile la compresenza di utenti non in chroot ed utenti in chroot; se non vengono introdotte applicazioni setuid negli ambienti chroot dell'utente, è più difficile evaderne. Comunque, potrebbe essere necessario impostare ambienti chroot individuali per ciascun utente e

[4] Such as *knockd*. Alternatively, you can open a different console and have the system ask for confirmation that there is somebody on the other side, and reset the firewall chain if no confirmation is given. The following test script could be of use:

```
#!/bin/bash

while true; do
    read -n 1 -p "Are you there? " -t 30 ayt
    if [ -z "$ayt" ] ; then
        break
    fi
done

# Reset the firewall chain, user is not available
echo
echo "Resetting firewall chain!"
iptables -F
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
exit 1
```

quindi sarebbe più difficile da configurare (perché richiede cooperazione da parte del server SSH). La seconda opzione è più facile da configurare e protegge dallo sfruttamento di un possibile baco nel server ssh stesso (dal momento che anch'esso è all'interno del chroot) ma avrà una limitazione che consiste nell'obbligo della condivisione, da parte di tutti gli utenti, del medesimo ambiente chroot (non si può configurare un ambiente chroot utente per utente).

Mettere gli utenti ssh in chroot

È possibile impostare il server ssh in modo che metta in chroot un certo insieme di utenti dentro una shell con una scelta limitata di applicazioni disponibili.

Using libpam-chroot

Probably the easiest way is to use the libpam-chroot package provided in Debian. Once you install it you need to:

- Modificare `/etc/pam.d/ssh` per usare questo modulo PAM, ed aggiungere come ultima riga[5]:

```
session    required   pam_chroot.so
```

- set a proper chroot environment for the user. You can try using the scripts available at `/usr/share/doc/libpam-chroot/examples/`, use the makejail[6] program or setup a minimum Debian environment with debootstrap. Make sure the environment includes the needed devices [7].

- Configurate `/etc/security/chroot.conf`, così che determinati utenti siano in chroot nella directory che avete configurato precedentemente. Potreste anche voler avere diverse directory indipendenti per utenti diversi così che non siano in grado di vedere né il sistema nella sua interezza né i file l'uno dell'altro.

- Configurate SSH: a seconda della versione di OpenSSH, l'ambiente in chroot potrebbe funzionare direttamente appena installato oppure no. Dalla versione 3.6.1p2 la funzione *do_pam_session()* viene chiamata dopo che sshd ha ridotto i privilegi, poiché, siccome chroot() necessita dei privilegi di root, naturalmente non funzionerà con l'opzione Privilege separation on. In versioni più recenti di OpenSSH, invece, il codice PAM è stato modificato e *do_pam_session()* viene chiamato prima di ridurre i privilegi, cosi funzionerà anche con Privilege separation on. Se dovete disabilitarlo, modificate come segue `/etc/ssh/sshd_config`:

```
UsePrivilegeSeparation no
```

[5] Potete usare l'opzione *debug* per fare in modo che spedisca il progresso raggiunto dal modulo al sistema *authpriv.notice*.

[6] You can create a very limited bash environment with the following python definition for makejail, just create the directory `/var/chroots/users/foo` and a file with the following contents and call it `bash.py`:

```
chroot="/var/chroots/users/foo"
cleanJailFirst=1
testCommandsInsideJail=["bash ls"]
```

And then run *makejail bash.py* to create the user environment at `/var/chroots/users/foo`. To test the environment run:

```
# chroot /var/chroots/users/foo/ ls
bin  dev  etc  lib  proc  sbin  usr
```

[7] In some occasions you might need the `/dev/ptmx` and `/dev/pty*` devices and the `/dev/pts/` subdirectory. Running MAKEDEV in the `/dev` directory of the chrooted environment should be sufficient to create them if they do not exist. If you are using kernels (version 2.6) which dynamically create device files you will need to create the /dev/pts/ files yourself and grant them the proper privileges.

Notice that this will lower the security of your system since the OpenSSH server will then run as *root* user. This means that if a remote attack is found against OpenSSH an attacker will get *root* privileges instead of *sshd*, thus compromising the whole system. [8]

Se non disabilitate l'opzione *Privilege Separation* avrete bisogno di un `/etc/passwd` che includa l'UID dell'utente all'interno del chroot perché *Privilege Separation* funzioni correttamente.

If you have *Privilege Separation* set to *yes* and your OpenSSH version does not behave properly you will need to disable it. If you don't, users that try to connect to your server and would be chrooted by this module will see this:

```
$ ssh -l user server
user@server's password:
Connection to server closed by remote host.
Connection to server closed.
```

Questo perché il demone ssh, che viene eseguito come 'sshd', non è in grado di eseguire la chiamata di sistema chroot(). Per disabilitare l'opzione Privilege separation, dovrete modificare il file di configurazione `/etc/ssh/sshd_config`, come descritto in precedenza.

Notate che se uno dei seguenti manca, gli utenti non saranno in grado di autenticarsi in chroot:

- Il filesystem `/proc` dev'essere montato nel chroot dell'utente.
- I file dispositivo necessari in `/dev/pts/` devono esistere. Se i file vengono generati automaticamente dal kernel in esecuzione allora devono esser creati manualmente nella `/dev/` del chroot.
- La home directory dell'utente deve esistere nel chroot, altrimenti il demone ssh smetterà di funzionare.

Si può correggere dai vari errori ognuna di queste questioni se usate la parola chiave *debug* nella definizione PAM di */etc/pam.d/ssh*. Se incontrate altre questioni potreste trovare interessante abilitare il modo di debug anche nel client ssh.

Notate che queste informazioni sono disponibili (e probabilmente anche più aggiornate) nel file `/usr/share/doc/libpam-chroot/README.Debian.gz`, per favore controllatelo per aggiornarvi sulle ultime novità prima di eseguire la suddetta procedura.

Applicare una patch al server SSH

Debian's **sshd** does not allow restriction of a user's movement through the server, since it lacks the **chroot** function that the commercial program **sshd2** includes (using 'ChrootGroups' or 'ChrootUsers', see sshd2_config(5)). However, there is a patch available to add this functionality available from http://chroot-tssh.sourceforge.net (requested and available in http://bugs.debian.org/139047 in Debian). The patch may be included in future releases of the OpenSSH package. Emmanuel Lacour has **ssh** deb packages for *sarge* with this feature. They are available at http://debian.home-dn.net/sarge/ssh/. Notice that those might not be up to date so completing the compilation step is recommended.

Dopo aver applicato la patch, modificate il file `/etc/passwd`, cambiando il percorso della directory home degli utenti (utilizzando la combinazione di caratteri speciale `/./`):

```
joeuser:x:1099:1099:Joe Random User:/home/joe/./:/bin/bash
```

In questo modo viene limitato *sia* l'accesso remoto alla shell *che* la copia remota attraverso il canale **ssh**.

[8] If you are using a kernel that implements Mandatory Access Control (RSBAC/SElinux) you can avoid changing this configuration just by granting the *sshd* user privileges to make the chroot() system call.

Bisogna accertarsi di avere tutti i binari e le librerie necessarie nel path soggetto a **chroot** per gli utenti. Questi file dovrebbero essere di proprietà di root per evitare che gli utenti li manomettano (per sfuggire dalla gabbia **chroot**). Un esempio potrebbe comprendere:

```
./bin:
total 660
drwxr-xr-x    2 root     root         4096 Mar 18 13:36 .
drwxr-xr-x    8 guest    guest        4096 Mar 15 16:53 ..
-r-xr-xr-x    1 root     root       531160 Feb  6 22:36 bash
-r-xr-xr-x    1 root     root        43916 Nov 29 13:19 ls
-r-xr-xr-x    1 root     root        16684 Nov 29 13:19 mkdir
-rwxr-xr-x    1 root     root        23960 Mar 18 13:36 more
-r-xr-xr-x    1 root     root         9916 Jul 26  2001 pwd
-r-xr-xr-x    1 root     root        24780 Nov 29 13:19 rm
lrwxrwxrwx    1 root     root            4 Mar 30 16:29 sh -> bash


./etc:
total 24
drwxr-xr-x    2 root     root         4096 Mar 15 16:13 .
drwxr-xr-x    8 guest    guest        4096 Mar 15 16:53 ..
-rw-r--r--    1 root     root           54 Mar 15 13:23 group
-rw-r--r--    1 root     root          428 Mar 15 15:56 hosts
-rw-r--r--    1 root     root           44 Mar 15 15:53 passwd
-rw-r--r--    1 root     root           52 Mar 15 13:23 shells


./lib:
total 1848
drwxr-xr-x    2 root     root         4096 Mar 18 13:37 .
drwxr-xr-x    8 guest    guest        4096 Mar 15 16:53 ..
-rwxr-xr-x    1 root     root        92511 Mar 15 12:49 ld-linux.so.2
-rwxr-xr-x    1 root     root      1170812 Mar 15 12:49 libc.so.6
-rw-r--r--    1 root     root        20900 Mar 15 13:01 libcrypt.so.1
-rw-r--r--    1 root     root         9436 Mar 15 12:49 libdl.so.2
-rw-r--r--    1 root     root       248132 Mar 15 12:48 libncurses.so.5
-rw-r--r--    1 root     root        71332 Mar 15 13:00 libnsl.so.1
-rw-r--r--    1 root     root        34144 Mar 15 16:10
libnss_files.so.2
-rw-r--r--    1 root     root        29420 Mar 15 12:57 libpam.so.0
-rw-r--r--    1 root     root       105498 Mar 15 12:51 libpthread.so.0
-rw-r--r--    1 root     root        25596 Mar 15 12:51 librt.so.1
-rw-r--r--    1 root     root         7760 Mar 15 12:59 libutil.so.1
-rw-r--r--    1 root     root        24328 Mar 15 12:57 libwrap.so.0


./usr:
total 16
drwxr-xr-x    4 root     root         4096 Mar 15 13:00 .
drwxr-xr-x    8 guest    guest        4096 Mar 15 16:53 ..
drwxr-xr-x    2 root     root         4096 Mar 15 15:55 bin
drwxr-xr-x    2 root     root         4096 Mar 15 15:37 lib
```

```
./usr/bin:
total 340
drwxr-xr-x    2 root     root         4096 Mar 15 15:55 .
drwxr-xr-x    4 root     root         4096 Mar 15 13:00 ..
-rwxr-xr-x    1 root     root        10332 Mar 15 15:55 env
-rwxr-xr-x    1 root     root        13052 Mar 15 13:13 id
-r-xr-xr-x    1 root     root        25432 Mar 15 12:40 scp
-rwxr-xr-x    1 root     root        43768 Mar 15 15:15 sftp
-r-sr-xr-x    1 root     root       218456 Mar 15 12:40 ssh
-rwxr-xr-x    1 root     root         9692 Mar 15 13:17 tty


./usr/lib:
total 852
drwxr-xr-x    2 root     root         4096 Mar 15 15:37 .
drwxr-xr-x    4 root     root         4096 Mar 15 13:00 ..
-rw-r--r--    1 root     root       771088 Mar 15 13:01
libcrypto.so.0.9.6
-rw-r--r--    1 root     root        54548 Mar 15 13:00 libz.so.1
-rwxr-xr-x    1 root     root        23096 Mar 15 15:37 sftp-server
```

Eseguire un chroot del server ssh

Se create un ambiente chroot che contiene al suo interno i file usati dal server ssh, per esempio `/var/chroot/ssh`, potrete avviare il server ssh contenuto in tale ambiente con il seguente comando:

```
# chroot /var/chroot/ssh /sbin/sshd -f /etc/sshd_config
```

Questo avvierebbe il demone **sshd** all'interno dell'ambiente chroot. A tal fine dovrete assicurarvi che la directory `/var/chroot/ssh` contenga il server SSH e tutti i programmi che gli utenti che si collegheranno tramite ssh potrebbero voler usare. In tal caso, dovrete anche controllare che SSH usi la *separazione dei privilegi* (normalmente nella configurazione base è così) e che la seguente riga sia presente nel file `/etc/ssh/sshd_config`:

```
UsePrivilegeSeparation yes
```

In tal modo il server SSH eseguirà come utente root solo le operazioni strettamente indispensabili; conseguentemente, verranno minimizzate le possibilità che un baco nel server SSH permetta di compromettere l'intero sistema. Notate che, a differenza della configurazione in cui assegnate ad ogni utente il proprio ambiente chroot, in questa configurazione il demone ssh condivide l'ambiente di chroot con altri utenti, perciò vi è almeno un processo con privilegi di root che, se compromesso, potrebbe essere usato per uscire fuori dalla gabbia chroot.

Notate inoltre che, per assicurare il corretto funzionamento del demone ssh nell'ambiente chroot, la partizione dove risiede la directory usata dall'ambiente chroot non può essere montata con l'opzione *nodev* attiva: in tal caso, poiché `/dev/urandom` non funziona in chroot, otterreste il seguente errore: *PRNG is not seeded* (NdT: il generatore di numeri pseudo-casuali non riesce a ottenere un punto da cui partire).

Come creare un sistema minimale (nella maniera più semplice possibile)

You can use debootstrap to setup a minimal environment that just includes the ssh server. In order to do this you just have to create a chroot as described in the http://www.debian.org/doc/manuals/referen-

ce/ch09#_chroot_system document. This method is bound to work (you will get all the necessary componentes for the chroot) but at the cost of disk space (a minimal installation of Debian will amount to several hundred megabytes). This minimal system might also include setuid files that a user in the chroot could use to break out of the chroot if any of those could be use for a privilege escalation.

Creazione automatica dell'ambiente chroot (nella maniera più semplice possibile)

You can easily create a restricted environment with the makejail package, since it automatically takes care of tracing the server daemon (with **strace**), and makes it run under the restricted environment.

Il vantaggio dei programmi che sono in grado di generare automaticamente un ambiente **chroot** risiede nella loro capacità di inserire un qualsiasi pacchetto (ed eventualmente le dipendenze di tale pacchetto) automaticamente nell'ambiente **chroot**. In questo modo, fornire al demone i pacchetti di cui ha bisogno risulta molto più semplice.

Per creare un ambiente chroot per il demone ssh sfruttando gli esempi forniti con **makejail**, create la directory `/var/chroot/sshd` ed usate il comando :

```
# makejail /usr/share/doc/makejail/examples/sshd.py
```

Tale comando creerà l'ambiente chroot richiesto nella directory `/var/chroot/sshd`. Notate che affinché il tutto funzioni correttamente è necessario che voi:

- Montiate il filesystem *procfs* nella directory `/var/chroot/sshd/proc`. Il programma **Makejail** effettua automaticamente tale operazione, ma in caso di riavvio del sistema il filesystem dovrà essere rimontato usando il comando :

  ```
  # mount -t proc proc /var/chroot/sshd/proc
  ```

  Si può anche farlo montare in automatico modificando il file `/etc/fstab` e aggiungendo questa riga:

  ```
  proc-ssh /var/chroot/sshd/proc  proc none 0 0
  ```

- Have syslog listen to the device `/dev/log` inside the chroot. In order to do this you have modify `/etc/default/syslogd` and add *-a /var/chroot/sshd/dev/log* to the `SYSLOGD` variable definition.

Leggete il file di esempio per vedere quali altre modifiche all'ambiente devono essere fatte. Alcuni di questi cambiamenti, come la copia della directory home dell'utente, non possono essere fatti automaticamente. Inoltre, limitate l'esposizione di informazioni sensibili copiando solo i dati di un dato numero di utenti dai file `/etc/shadow` o `/etc/group`. Notate che se state usando Privilege Separation l'utente *sshd* deve esistere in questi file.

The following sample environment has been (slightly) tested in Debian 3.0 and is built with the configuration file provided in the package and includes the fileutils package:

```
.
|-- bin
|   |-- ash
```

```
|   |-- bash
|   |-- chgrp
|   |-- chmod
|   |-- chown
|   |-- cp
|   |-- csh -> /etc/alternatives/csh
|   |-- dd
|   |-- df
|   |-- dir
|   |-- fdflush
|   |-- ksh
|   |-- ln
|   |-- ls
|   |-- mkdir
|   |-- mknod
|   |-- mv
|   |-- rbash -> bash
|   |-- rm
|   |-- rmdir
|   |-- sh -> bash
|   |-- sync
|   |-- tcsh
|   |-- touch
|   |-- vdir
|   |-- zsh -> /etc/alternatives/zsh
|   `-- zsh4
|-- dev
|   |-- null
|   |-- ptmx
|   |-- pts
|   |-- ptya0
(...)
|   |-- tty
|   |-- tty0
(...)
|   `-- urandom
|-- etc
|   |-- alternatives
|   |   |-- csh -> /bin/tcsh
|   |   `-- zsh -> /bin/zsh4
|   |-- environment
|   |-- hosts
|   |-- hosts.allow
|   |-- hosts.deny
|   |-- ld.so.conf
|   |-- localtime -> /usr/share/zoneinfo/Europe/Madrid
|   |-- motd
|   |-- nsswitch.conf
|   |-- pam.conf
|   |-- pam.d
|   |   |-- other
|   |   `-- ssh
|   |-- passwd
|   |-- resolv.conf
```

```
|   |-- security
|   |   |-- access.conf
|   |   |-- chroot.conf
|   |   |-- group.conf
|   |   |-- limits.conf
|   |   |-- pam_env.conf
|   |   `-- time.conf
|   |-- shadow
|   |-- shells
|   `-- ssh
|       |-- moduli
|       |-- ssh_host_dsa_key
|       |-- ssh_host_dsa_key.pub
|       |-- ssh_host_rsa_key
|       |-- ssh_host_rsa_key.pub
|       `-- sshd_config
|-- home
|   `-- userX
|-- lib
|   |-- ld-2.2.5.so
|   |-- ld-linux.so.2 -> ld-2.2.5.so
|   |-- libc-2.2.5.so
|   |-- libc.so.6 -> libc-2.2.5.so
|   |-- libcap.so.1 -> libcap.so.1.10
|   |-- libcap.so.1.10
|   |-- libcrypt-2.2.5.so
|   |-- libcrypt.so.1 -> libcrypt-2.2.5.so
|   |-- libdl-2.2.5.so
|   |-- libdl.so.2 -> libdl-2.2.5.so
|   |-- libm-2.2.5.so
|   |-- libm.so.6 -> libm-2.2.5.so
|   |-- libncurses.so.5 -> libncurses.so.5.2
|   |-- libncurses.so.5.2
|   |-- libnsl-2.2.5.so
|   |-- libnsl.so.1 -> libnsl-2.2.5.so
|   |-- libnss_compat-2.2.5.so
|   |-- libnss_compat.so.2 -> libnss_compat-2.2.5.so
|   |-- libnss_db-2.2.so
|   |-- libnss_db.so.2 -> libnss_db-2.2.so
|   |-- libnss_dns-2.2.5.so
|   |-- libnss_dns.so.2 -> libnss_dns-2.2.5.so
|   |-- libnss_files-2.2.5.so
|   |-- libnss_files.so.2 -> libnss_files-2.2.5.so
|   |-- libnss_hesiod-2.2.5.so
|   |-- libnss_hesiod.so.2 -> libnss_hesiod-2.2.5.so
|   |-- libnss_nis-2.2.5.so
|   |-- libnss_nis.so.2 -> libnss_nis-2.2.5.so
|   |-- libnss_nisplus-2.2.5.so
|   |-- libnss_nisplus.so.2 -> libnss_nisplus-2.2.5.so
|   |-- libpam.so.0 -> libpam.so.0.72
|   |-- libpam.so.0.72
|   |-- libpthread-0.9.so
|   |-- libpthread.so.0 -> libpthread-0.9.so
|   |-- libresolv-2.2.5.so
```

```
|   |-- libresolv.so.2 -> libresolv-2.2.5.so
|   |-- librt-2.2.5.so
|   |-- librt.so.1 -> librt-2.2.5.so
|   |-- libutil-2.2.5.so
|   |-- libutil.so.1 -> libutil-2.2.5.so
|   |-- libwrap.so.0 -> libwrap.so.0.7.6
|   |-- libwrap.so.0.7.6
|   `-- security
|       |-- pam_access.so
|       |-- pam_chroot.so
|       |-- pam_deny.so
|       |-- pam_env.so
|       |-- pam_filter.so
|       |-- pam_ftp.so
|       |-- pam_group.so
|       |-- pam_issue.so
|       |-- pam_lastlog.so
|       |-- pam_limits.so
|       |-- pam_listfile.so
|       |-- pam_mail.so
|       |-- pam_mkhomedir.so
|       |-- pam_motd.so
|       |-- pam_nologin.so
|       |-- pam_permit.so
|       |-- pam_rhosts_auth.so
|       |-- pam_rootok.so
|       |-- pam_securetty.so
|       |-- pam_shells.so
|       |-- pam_stress.so
|       |-- pam_tally.so
|       |-- pam_time.so
|       |-- pam_unix.so
|       |-- pam_unix_acct.so -> pam_unix.so
|       |-- pam_unix_auth.so -> pam_unix.so
|       |-- pam_unix_passwd.so -> pam_unix.so
|       |-- pam_unix_session.so -> pam_unix.so
|       |-- pam_userdb.so
|       |-- pam_warn.so
|       `-- pam_wheel.so
|-- sbin
|   `-- start-stop-daemon
|-- usr
|   |-- bin
|   |   |-- dircolors
|   |   |-- du
|   |   |-- install
|   |   |-- link
|   |   |-- mkfifo
|   |   |-- shred
|   |   |-- touch -> /bin/touch
|   |   `-- unlink
|   |-- lib
|   |   |-- libcrypto.so.0.9.6
|   |   |-- libdb3.so.3 -> libdb3.so.3.0.2
```

```
|   |   |-- libdb3.so.3.0.2
|   |   |-- libz.so.1 -> libz.so.1.1.4
|   |   `-- libz.so.1.1.4
|   |-- sbin
|   |   `-- sshd
|   `-- share
|       |-- locale
|       |   `-- es
|       |       |-- LC_MESSAGES
|       |       |   |-- fileutils.mo
|       |       |   |-- libc.mo
|       |       |   `-- sh-utils.mo
|       |       `-- LC_TIME -> LC_MESSAGES
|       `-- zoneinfo
|           `-- Europe
|               `-- Madrid
`-- var
    `-- run
        |-- sshd
        `-- sshd.pid

27 directories, 733 files
```

Per la distribuzione di Debian 3.1, assicuratevi che l'ambiente includa anche i file comuni per PAM. I seguenti file devono essere copiati all'interno del chroot se **makejail** non lo ha fatto per voi:

```
$ ls /etc/pam.d/common-*
/etc/pam.d/common-account  /etc/pam.d/common-password
/etc/pam.d/common-auth     /etc/pam.d/common-session
```

Ambiente fatto a mano (nella maniera più brutale)

It is possible to create an environment, using a trial-and-error method, by monitoring the **sshd** server traces and log files in order to determine the necessary files. The following environment, contributed by José Luis Ledesma, is a sample listing of files in a **chroot** environment for **ssh** in Debian woody (3.0): [9]

```
.:
total 36
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ./
drwxr-xr-x 11 root root 4096 Jun 3 13:43 ../
drwxr-xr-x 2 root root 4096 Jun 4 12:13 bin/
drwxr-xr-x 2 root root 4096 Jun 4 12:16 dev/
drwxr-xr-x 4 root root 4096 Jun 4 12:35 etc/
drwxr-xr-x 3 root root 4096 Jun 4 12:13 lib/
drwxr-xr-x 2 root root 4096 Jun 4 12:35 sbin/
drwxr-xr-x 2 root root 4096 Jun 4 12:32 tmp/
drwxr-xr-x 2 root root 4096 Jun 4 12:16 usr/
./bin:
total 8368
drwxr-xr-x 2 root root 4096 Jun 4 12:13 ./
```

[9] Notice that there are no SETUID files. This makes it more difficult for remote users to escape the **chroot** environment. However, it also prevents users from changing their passwords, since the **passwd** program cannot modify the files `/etc/passwd` or `/etc/shadow`.

```
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
-rwxr-xr-x 1 root root 109855 Jun 3 13:45 a2p*
-rwxr-xr-x 1 root root 387764 Jun 3 13:45 bash*
-rwxr-xr-x 1 root root 36365 Jun 3 13:45 c2ph*
-rwxr-xr-x 1 root root 20629 Jun 3 13:45 dprofpp*
-rwxr-xr-x 1 root root 6956 Jun 3 13:46 env*
-rwxr-xr-x 1 root root 158116 Jun 3 13:45 fax2ps*
-rwxr-xr-x 1 root root 104008 Jun 3 13:45 faxalter*
-rwxr-xr-x 1 root root 89340 Jun 3 13:45 faxcover*
-rwxr-xr-x 1 root root 441584 Jun 3 13:45 faxmail*
-rwxr-xr-x 1 root root 96036 Jun 3 13:45 faxrm*
-rwxr-xr-x 1 root root 107000 Jun 3 13:45 faxstat*
-rwxr-xr-x 1 root root 77832 Jun 4 11:46 grep*
-rwxr-xr-x 1 root root 19597 Jun 3 13:45 h2ph*
-rwxr-xr-x 1 root root 46979 Jun 3 13:45 h2xs*
-rwxr-xr-x 1 root root 10420 Jun 3 13:46 id*
-rwxr-xr-x 1 root root 4528 Jun 3 13:46 ldd*
-rwxr-xr-x 1 root root 111386 Jun 4 11:46 less*
-r-xr-xr-x 1 root root 26168 Jun 3 13:45 login*
-rwxr-xr-x 1 root root 49164 Jun 3 13:45 ls*
-rwxr-xr-x 1 root root 11600 Jun 3 13:45 mkdir*
-rwxr-xr-x 1 root root 24780 Jun 3 13:45 more*
-rwxr-xr-x 1 root root 154980 Jun 3 13:45 pal2rgb*
-rwxr-xr-x 1 root root 27920 Jun 3 13:46 passwd*
-rwxr-xr-x 1 root root 4241 Jun 3 13:45 pl2pm*
-rwxr-xr-x 1 root root 2350 Jun 3 13:45 pod2html*
-rwxr-xr-x 1 root root 7875 Jun 3 13:45 pod2latex*
-rwxr-xr-x 1 root root 17587 Jun 3 13:45 pod2man*
-rwxr-xr-x 1 root root 6877 Jun 3 13:45 pod2text*
-rwxr-xr-x 1 root root 3300 Jun 3 13:45 pod2usage*
-rwxr-xr-x 1 root root 3341 Jun 3 13:45 podchecker*
-rwxr-xr-x 1 root root 2483 Jun 3 13:45 podselect*
-r-xr-xr-x 1 root root 82412 Jun 4 11:46 ps*
-rwxr-xr-x 1 root root 36365 Jun 3 13:45 pstruct*
-rwxr-xr-x 1 root root 7120 Jun 3 13:45 pwd*
-rwxr-xr-x 1 root root 179884 Jun 3 13:45 rgb2ycbcr*
-rwxr-xr-x 1 root root 20532 Jun 3 13:45 rm*
-rwxr-xr-x 1 root root 6720 Jun 4 10:15 rmdir*
-rwxr-xr-x 1 root root 14705 Jun 3 13:45 s2p*
-rwxr-xr-x 1 root root 28764 Jun 3 13:46 scp*
-rwxr-xr-x 1 root root 385000 Jun 3 13:45 sendfax*
-rwxr-xr-x 1 root root 67548 Jun 3 13:45 sendpage*
-rwxr-xr-x 1 root root 88632 Jun 3 13:46 sftp*
-rwxr-xr-x 1 root root 387764 Jun 3 13:45 sh*
-rws--x--x 1 root root 744500 Jun 3 13:46 slogin*
-rwxr-xr-x 1 root root 14523 Jun 3 13:46 splain*
-rws--x--x 1 root root 744500 Jun 3 13:46 ssh*
-rwxr-xr-x 1 root root 570960 Jun 3 13:46 ssh-add*
-rwxr-xr-x 1 root root 502952 Jun 3 13:46 ssh-agent*
-rwxr-xr-x 1 root root 575740 Jun 3 13:46 ssh-keygen*
-rwxr-xr-x 1 root root 383480 Jun 3 13:46 ssh-keyscan*
-rwxr-xr-x 1 root root 39 Jun 3 13:46 ssh_europa*
-rwxr-xr-x 1 root root 107252 Jun 4 10:14 strace*
-rwxr-xr-x 1 root root 8323 Jun 4 10:14 strace-graph*
```

```
-rwxr-xr-x 1 root root 158088 Jun 3 13:46 thumbnail*
-rwxr-xr-x 1 root root 6312 Jun 3 13:46 tty*
-rwxr-xr-x 1 root root 55904 Jun 4 11:46 useradd*
-rwxr-xr-x 1 root root 585656 Jun 4 11:47 vi*
-rwxr-xr-x 1 root root 6444 Jun 4 11:45 whoami*
./dev:
total 8
drwxr-xr-x 2 root root 4096 Jun 4 12:16 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
crw-r--r-- 1 root root 1, 9 Jun 3 13:43 urandom
./etc:
total 208
drwxr-xr-x 4 root root 4096 Jun 4 12:35 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
-rw------- 1 root root 0 Jun 4 11:46 .pwd.lock
-rw-r--r-- 1 root root 653 Jun 3 13:46 group
-rw-r--r-- 1 root root 242 Jun 4 11:33 host.conf
-rw-r--r-- 1 root root 857 Jun 4 12:04 hosts
-rw-r--r-- 1 root root 1050 Jun 4 11:29 ld.so.cache
-rw-r--r-- 1 root root 304 Jun 4 11:28 ld.so.conf
-rw-r--r-- 1 root root 235 Jun 4 11:27 ld.so.conf~
-rw-r--r-- 1 root root 88039 Jun 3 13:46 moduli
-rw-r--r-- 1 root root 1342 Jun 4 11:34 nsswitch.conf
drwxr-xr-x 2 root root 4096 Jun 4 12:02 pam.d/
-rw-r--r-- 1 root root 28 Jun 4 12:00 pam_smb.conf
-rw-r--r-- 1 root root 2520 Jun 4 11:57 passwd
-rw-r--r-- 1 root root 7228 Jun 3 13:48 profile
-rw-r--r-- 1 root root 1339 Jun 4 11:33 protocols
-rw-r--r-- 1 root root 274 Jun 4 11:44 resolv.conf
drwxr-xr-x 2 root root 4096 Jun 3 13:43 security/
-rw-r----- 1 root root 1178 Jun 4 11:51 shadow
-rw------- 1 root root 80 Jun 4 11:45 shadow-
-rw-r----- 1 root root 1178 Jun 4 11:48 shadow.old
-rw-r--r-- 1 root root 161 Jun 3 13:46 shells
-rw-r--r-- 1 root root 1144 Jun 3 13:46 ssh_config
-rw------- 1 root root 668 Jun 3 13:46 ssh_host_dsa_key
-rw-r--r-- 1 root root 602 Jun 3 13:46 ssh_host_dsa_key.pub
-rw------- 1 root root 527 Jun 3 13:46 ssh_host_key
-rw-r--r-- 1 root root 331 Jun 3 13:46 ssh_host_key.pub
-rw------- 1 root root 883 Jun 3 13:46 ssh_host_rsa_key
-rw-r--r-- 1 root root 222 Jun 3 13:46 ssh_host_rsa_key.pub
-rw-r--r-- 1 root root 2471 Jun 4 12:15 sshd_config
./etc/pam.d:
total 24
drwxr-xr-x 2 root root 4096 Jun 4 12:02 ./
drwxr-xr-x 4 root root 4096 Jun 4 12:35 ../
lrwxrwxrwx 1 root root 4 Jun 4 12:02 other -> sshd
-rw-r--r-- 1 root root 318 Jun 3 13:46 passwd
-rw-r--r-- 1 root root 546 Jun 4 11:36 ssh
-rw-r--r-- 1 root root 479 Jun 4 12:02 sshd
-rw-r--r-- 1 root root 370 Jun 3 13:46 su
./etc/security:
total 32
drwxr-xr-x 2 root root 4096 Jun 3 13:43 ./
```

```
drwxr-xr-x 4 root root 4096 Jun 4 12:35 ../
-rw-r--r-- 1 root root 1971 Jun 3 13:46 access.conf
-rw-r--r-- 1 root root 184 Jun 3 13:46 chroot.conf
-rw-r--r-- 1 root root 2145 Jun 3 13:46 group.conf
-rw-r--r-- 1 root root 1356 Jun 3 13:46 limits.conf
-rw-r--r-- 1 root root 2858 Jun 3 13:46 pam_env.conf
-rw-r--r-- 1 root root 2154 Jun 3 13:46 time.conf
./lib:
total 8316
drwxr-xr-x 3 root root 4096 Jun 4 12:13 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
-rw-r--r-- 1 root root 1024 Jun 4 11:51 cracklib_dict.hwm
-rw-r--r-- 1 root root 214324 Jun 4 11:51 cracklib_dict.pwd
-rw-r--r-- 1 root root 11360 Jun 4 11:51 cracklib_dict.pwi
-rwxr-xr-x 1 root root 342427 Jun 3 13:46 ld-linux.so.2*
-rwxr-xr-x 1 root root 4061504 Jun 3 13:46 libc.so.6*
lrwxrwxrwx 1 root root 15 Jun 4 12:11 libcrack.so -> libcrack.so.2.7*
lrwxrwxrwx 1 root root 15 Jun 4 12:11 libcrack.so.2 -> libcrack.so.2.7*
-rwxr-xr-x 1 root root 33291 Jun 4 11:39 libcrack.so.2.7*
-rwxr-xr-x 1 root root 60988 Jun 3 13:46 libcrypt.so.1*
-rwxr-xr-x 1 root root 71846 Jun 3 13:46 libdl.so.2*
-rwxr-xr-x 1 root root 27762 Jun 3 13:46 libhistory.so.4.0*
lrwxrwxrwx 1 root root 17 Jun 4 12:12 libncurses.so.4 -> libncurses.so.4.2*
-rwxr-xr-x 1 root root 503903 Jun 3 13:46 libncurses.so.4.2*
lrwxrwxrwx 1 root root 17 Jun 4 12:12 libncurses.so.5 -> libncurses.so.5.0*
-rwxr-xr-x 1 root root 549429 Jun 3 13:46 libncurses.so.5.0*
-rwxr-xr-x 1 root root 369801 Jun 3 13:46 libnsl.so.1*
-rwxr-xr-x 1 root root 142563 Jun 4 11:49 libnss_compat.so.1*
-rwxr-xr-x 1 root root 215569 Jun 4 11:49 libnss_compat.so.2*
-rwxr-xr-x 1 root root 61648 Jun 4 11:34 libnss_dns.so.1*
-rwxr-xr-x 1 root root 63453 Jun 4 11:34 libnss_dns.so.2*
-rwxr-xr-x 1 root root 63782 Jun 4 11:34 libnss_dns6.so.2*
-rwxr-xr-x 1 root root 205715 Jun 3 13:46 libnss_files.so.1*
-rwxr-xr-x 1 root root 235932 Jun 3 13:49 libnss_files.so.2*
-rwxr-xr-x 1 root root 204383 Jun 4 11:33 libnss_nis.so.1*
-rwxr-xr-x 1 root root 254023 Jun 4 11:33 libnss_nis.so.2*
-rwxr-xr-x 1 root root 256465 Jun 4 11:33 libnss_nisplus.so.2*
lrwxrwxrwx 1 root root 14 Jun 4 12:12 libpam.so.0 -> libpam.so.0.72*
-rwxr-xr-x 1 root root 31449 Jun 3 13:46 libpam.so.0.72*
lrwxrwxrwx 1 root root 19 Jun 4 12:12 libpam_misc.so.0 ->
libpam_misc.so.0.72*
-rwxr-xr-x 1 root root 8125 Jun 3 13:46 libpam_misc.so.0.72*
lrwxrwxrwx 1 root root 15 Jun 4 12:12 libpamc.so.0 -> libpamc.so.0.72*
-rwxr-xr-x 1 root root 10499 Jun 3 13:46 libpamc.so.0.72*
-rwxr-xr-x 1 root root 176427 Jun 3 13:46 libreadline.so.4.0*
-rwxr-xr-x 1 root root 44729 Jun 3 13:46 libutil.so.1*
-rwxr-xr-x 1 root root 70254 Jun 3 13:46 libz.a*
lrwxrwxrwx 1 root root 13 Jun 4 12:13 libz.so -> libz.so.1.1.3*
lrwxrwxrwx 1 root root 13 Jun 4 12:13 libz.so.1 -> libz.so.1.1.3*
-rwxr-xr-x 1 root root 63312 Jun 3 13:46 libz.so.1.1.3*
drwxr-xr-x 2 root root 4096 Jun 4 12:00 security/
./lib/security:
total 668
drwxr-xr-x 2 root root 4096 Jun 4 12:00 ./
```

```
drwxr-xr-x 3 root root 4096 Jun 4 12:13 ../
-rwxr-xr-x 1 root root 10067 Jun 3 13:46 pam_access.so*
-rwxr-xr-x 1 root root 8300 Jun 3 13:46 pam_chroot.so*
-rwxr-xr-x 1 root root 14397 Jun 3 13:46 pam_cracklib.so*
-rwxr-xr-x 1 root root 5082 Jun 3 13:46 pam_deny.so*
-rwxr-xr-x 1 root root 13153 Jun 3 13:46 pam_env.so*
-rwxr-xr-x 1 root root 13371 Jun 3 13:46 pam_filter.so*
-rwxr-xr-x 1 root root 7957 Jun 3 13:46 pam_ftp.so*
-rwxr-xr-x 1 root root 12771 Jun 3 13:46 pam_group.so*
-rwxr-xr-x 1 root root 10174 Jun 3 13:46 pam_issue.so*
-rwxr-xr-x 1 root root 9774 Jun 3 13:46 pam_lastlog.so*
-rwxr-xr-x 1 root root 13591 Jun 3 13:46 pam_limits.so*
-rwxr-xr-x 1 root root 11268 Jun 3 13:46 pam_listfile.so*
-rwxr-xr-x 1 root root 11182 Jun 3 13:46 pam_mail.so*
-rwxr-xr-x 1 root root 5923 Jun 3 13:46 pam_nologin.so*
-rwxr-xr-x 1 root root 5460 Jun 3 13:46 pam_permit.so*
-rwxr-xr-x 1 root root 18226 Jun 3 13:46 pam_pwcheck.so*
-rwxr-xr-x 1 root root 12590 Jun 3 13:46 pam_rhosts_auth.so*
-rwxr-xr-x 1 root root 5551 Jun 3 13:46 pam_rootok.so*
-rwxr-xr-x 1 root root 7239 Jun 3 13:46 pam_securetty.so*
-rwxr-xr-x 1 root root 6551 Jun 3 13:46 pam_shells.so*
-rwxr-xr-x 1 root root 55925 Jun 4 12:00 pam_smb_auth.so*
-rwxr-xr-x 1 root root 12678 Jun 3 13:46 pam_stress.so*
-rwxr-xr-x 1 root root 11170 Jun 3 13:46 pam_tally.so*
-rwxr-xr-x 1 root root 11124 Jun 3 13:46 pam_time.so*
-rwxr-xr-x 1 root root 45703 Jun 3 13:46 pam_unix.so*
-rwxr-xr-x 1 root root 45703 Jun 3 13:46 pam_unix2.so*
-rwxr-xr-x 1 root root 45386 Jun 3 13:46 pam_unix_acct.so*
-rwxr-xr-x 1 root root 45386 Jun 3 13:46 pam_unix_auth.so*
-rwxr-xr-x 1 root root 45386 Jun 3 13:46 pam_unix_passwd.so*
-rwxr-xr-x 1 root root 45386 Jun 3 13:46 pam_unix_session.so*
-rwxr-xr-x 1 root root 9726 Jun 3 13:46 pam_userdb.so*
-rwxr-xr-x 1 root root 6424 Jun 3 13:46 pam_warn.so*
-rwxr-xr-x 1 root root 7460 Jun 3 13:46 pam_wheel.so*
./sbin:
total 3132
drwxr-xr-x 2 root root 4096 Jun 4 12:35 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
-rwxr-xr-x 1 root root 178256 Jun 3 13:46 choptest*
-rwxr-xr-x 1 root root 184032 Jun 3 13:46 cqtest*
-rwxr-xr-x 1 root root 81096 Jun 3 13:46 dialtest*
-rwxr-xr-x 1 root root 1142128 Jun 4 11:28 ldconfig*
-rwxr-xr-x 1 root root 2868 Jun 3 13:46 lockname*
-rwxr-xr-x 1 root root 3340 Jun 3 13:46 ondelay*
-rwxr-xr-x 1 root root 376796 Jun 3 13:46 pagesend*
-rwxr-xr-x 1 root root 13950 Jun 3 13:46 probemodem*
-rwxr-xr-x 1 root root 9234 Jun 3 13:46 recvstats*
-rwxr-xr-x 1 root root 64480 Jun 3 13:46 sftp-server*
-rwxr-xr-x 1 root root 744412 Jun 3 13:46 sshd*
-rwxr-xr-x 1 root root 30750 Jun 4 11:46 su*
-rwxr-xr-x 1 root root 194632 Jun 3 13:46 tagtest*
-rwxr-xr-x 1 root root 69892 Jun 3 13:46 tsitest*
-rwxr-xr-x 1 root root 43792 Jun 3 13:46 typetest*
./tmp:
```

```
total 8
drwxr-xr-x 2 root root 4096 Jun 4 12:32 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
./usr:
total 8
drwxr-xr-x 2 root root 4096 Jun 4 12:16 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
lrwxrwxrwx 1 root root 7 Jun 4 12:14 bin -> ../bin//
lrwxrwxrwx 1 root root 7 Jun 4 11:33 lib -> ../lib//
lrwxrwxrwx 1 root root 8 Jun 4 12:13 sbin -> ../sbin//
```

Chroot environment for Apache

Introduzione

L'utilità **chroot** viene usata spesso per imprigionare un demone in un settore sottoposto a limitazioni, per isolare reciprocamente dei servizi, in modo che problemi di sicurezza in un pacchetto software non mettano in pericolo l'intero server. Usando lo script **makejail**, impostare ed aggiornare la zona sottoposta a chroot è più facile.

FIXME: Apache can also be chrooted using http://www.modsecurity.org which is available in libapache-mod-security (for Apache 1.x) and libapache2-mod-security (for Apache 2.x).

Licenza

Installare il server

This procedure was tested on Debian GNU/Linux 3.0 (Woody) with **makejail** 0.0.4-1 (in Debian/testing).

- Autenticatevi come **root** e creare una nuova directory-gabbia:

  ```
  $ mkdir -p /var/chroot/apache
  ```

- Create un nuovo utente ed un nuovo gruppo. Il server Apache sottoposto a "chroot" sarà attivo in qualità di nuovo utente/gruppo, usato, nel sistema, solo a questo fine. In questo esempio, entrambi vengono chiamati **chrapach**.

  ```
  $ adduser --home /var/chroot/apache --shell /bin/false \
  --no-create-home --system --group chrapach
  ```

  FIXME: Occorre un nuovo utente? Apache è già attivo come utente Apache.

- Installate Apache come d'uso, su Debian: `apt-get install apache`.

- Impostate Apache (definite i vostri sottodomini, etc.) e, nel file di configurazione `/etc/apache/httpd.conf`, impostate *Group* ed *User* come `chrapach`. Riavviate Apache ed assicuratevi che funzioni correttamente. A questo punto, fermate il demone Apache.

- Installate **makejail** (disponibile, per ora, in Debian/testing) ed anche **wget** e **lynx**, giacché vengono usati da **makejail** per collaudare il server sottoposto a "chroot": `apt-get install makejail wget lynx`.

- Copiate il file della configurazione esemplificativa per Apache nella cartella `/etc/makejail`:

```
# cp /usr/share/doc/makejail/examples/apache.py /etc/makejail/
```

- Modificate `/etc/makejail/apache.py`, cambiando le opzioni di *chroot*, *users* e *groups*. Per eseguire questa versione di **makejail**, potete aggiungere anche l'opzione **packages**. Vedete nella http://www.floc.net/makejail/current/doc/. Un esempio potrebbe essere questo:

```
chroot="/var/chroot/apache"
testCommandsInsideJail=["/usr/sbin/apachectl start"]
processNames=["apache"]
testCommandsOutsideJail=["wget -r --spider http://localhost/",
                         "lynx --source https://localhost/"]
preserve=["/var/www",
          "/var/log/apache",
          "/dev/log"]
users=["chrapach"]
groups=["chrapach"]
packages=["apache", "apache-common"]
userFiles=["/etc/password",
           "/etc/shadow"]
groupFiles=["/etc/group",
            "/etc/gshadow"]
forceCopy=["/etc/hosts",
           "/etc/mime.types"]
```

  FIXME: pare che alcune opzioni non funzionino correttamente. Per esempio, `/etc/shadow` e `/etc/gshadow` non vengono copiate, mentre `/etc/password` e `/etc/group` sono state copiate completamente, invece di essere filtrate.

- Create il sottoalbero sottoposto a chroot: `makejail /etc/makejail/apache.py`

- Qualora `/etc/password` ed `/etc/group` siano stati copiati completamente, per sostituirli con copie filtrate, digitate:

```
$ grep chrapach /etc/passwd > /var/chroot/apache/etc/passwd
$ grep chrapach /etc/group > /var/chroot/apache/etc/group
```

  to replace them with filtered copies.

- Copy the Web site pages and the logs into the jail. These files are not copied automatically (see the *preserve* option in **makejail**'s configuration file).

```
# cp -Rp /var/www /var/chroot/apache/var
# cp -Rp /var/log/apache/*.log /var/chroot/apache/var/log/apache
```

- Edit the startup script for the system logging daemon so that it also listen to the `/var/chroot/apache/dev/log` socket. In `/etc/default/syslogd`, replace: `SYSLOGD=""` with `SYSLOGD=" -`

a /var/chroot/apache/dev/log" and restart the daemon (/etc/init.d/sysklogd restart).

- Edit the Apache startup script (/etc/init.d/apache). You might need to make some changes to the default startup script for it to run properly with a chrooted tree. Such as:
  - set a new *CHRDIR* variable at the top of the file;
  - edit the *start*, *stop*, *reload*, etc. sections;
  - add a line to mount and unmount the /proc filesystem within the jail.

```
#! /bin/bash
#
# apache        Start the apache HTTP server.
#

CHRDIR=/var/chroot/apache

NAME=apache
PATH=/bin:/usr/bin:/sbin:/usr/sbin
DAEMON=/usr/sbin/apache
SUEXEC=/usr/lib/apache/suexec
PIDFILE=/var/run/$NAME.pid
CONF=/etc/apache/httpd.conf
APACHECTL=/usr/sbin/apachectl

trap "" 1
export LANG=C
export PATH

test -f $DAEMON || exit 0
test -f $APACHECTL || exit 0

# ensure we don't leak environment vars into apachectl
APACHECTL="env -i LANG=${LANG} PATH=${PATH} chroot $CHRDIR $APACHECTL"

if egrep -q -i "^[[:space:]]*ServerType[[:space:]]+inet" $CONF
then
    exit 0
fi

case "$1" in
  start)
    echo -n "Starting web server: $NAME"
    mount -t proc proc /var/chroot/apache/proc
    start-stop-daemon --start --pidfile $PIDFILE --exec $DAEMON \
      --chroot $CHRDIR
    ;;

  stop)
    echo -n "Stopping web server: $NAME"
    start-stop-daemon --stop --pidfile "$CHRDIR/$PIDFILE" --oknodo
    umount /var/chroot/apache/proc
```

```
    ;;

  reload)
    echo -n "Reloading $NAME configuration"
    start-stop-daemon --stop --pidfile "$CHRDIR/$PIDFILE" \
      --signal USR1 --startas $DAEMON --chroot $CHRDIR
    ;;

  reload-modules)
    echo -n "Reloading $NAME modules"
    start-stop-daemon --stop --pidfile "$CHRDIR/$PIDFILE" --oknodo \
      --retry 30
    start-stop-daemon --start --pidfile $PIDFILE \
      --exec $DAEMON --chroot $CHRDIR
    ;;

  restart)
    $0 reload-modules
    exit $?
    ;;

  force-reload)
    $0 reload-modules
    exit $?
    ;;

  *)
    echo "Usage: /etc/init.d/$NAME {start|stop|reload|reload-modules|force-reloa
    exit 1
    ;;
esac

if [ $? == 0 ]; then
  echo .
  exit 0
else
  echo failed
  exit 1
fi
```

FIXME: should the first Apache process be run as another user than root (i.e. add --chuid chrapach:chrapach)? Cons: chrapach will need write access to the logs, which is awkward.

- Replace in `/etc/logrotate.d/apache/var/log/apache/*.log` with `/var/chroot/apache/var/log/apache/*.log`
- Start Apache (**/etc/init.d/apache start**) and check what is it reported in the jail log (`/var/chroot/apache/var/log/apache/error.log`). If your setup is more complex, (e.g. if you also use PHP and MySQL), files will probably be missing. if some files are not copied automatically by **makejail**, you can list them in the *forceCopy* (to copy files directly) or *packages* (to copy full packages and their dependencies) option the `/etc/makejail/apache.py` configuration file.
- Type `ps aux | grep apache` to make sure Apache is running. You should see something like:

```
root 180 0.0 1.1 2936 1436 ? S 04:03 0:00 /usr/sbin/apache
chrapach 189 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
chrapach 190 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
chrapach 191 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
chrapach 192 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
chrapach 193 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
```

- Make sure the Apache processes are running chrooted by looking in the `/proc` filesystem: `ls -la /proc/process_number/root/.` where `process_number` is one of the PID numbers listed above (2nd column; 189 for instance). The entries for a restricted tree should be listed:

```
drwxr-sr-x 10 root staff 240 Dec 2 16:06 .
drwxrwsr-x 4 root staff 72 Dec 2 08:07 ..
drwxr-xr-x 2 root root 144 Dec 2 16:05 bin
drwxr-xr-x 2 root root 120 Dec 3 04:03 dev
drwxr-xr-x 5 root root 408 Dec 3 04:03 etc
drwxr-xr-x 2 root root 800 Dec 2 16:06 lib
dr-xr-xr-x 43 root root 0 Dec 3 05:03 proc
drwxr-xr-x 2 root root 48 Dec 2 16:06 sbin
drwxr-xr-x 6 root root 144 Dec 2 16:04 usr
drwxr-xr-x 7 root root 168 Dec 2 16:06 var
```

  To automate this test, you can type:`ls -la /proc/`cat /var/chroot/apache/var/run/apache.pid`/root/.`

  FIXME: Add other tests that can be run to make sure the jail is closed?

The reason I like this is because setting up the jail is not very difficult and the server can be updated in just two lines:

```
apt-get update && apt-get install apache
makejail /etc/makejail/apache.py
```

See also

If you are looking for more information you can consider these sources of information in which the information presented is based: http://www.floc.net/makejail/, this program was written by Alain Tesio