Linux Security HOWTO

Kevin Fenzi, kevin@tummy.com & Dave Wreski, dave@linuxsecurity.com v1.1.1, 17 March 2000

Traduzione a cura di Claudio Gasparotto gasp@eponet.it e Matteo Scordino felinux@siena.linux.it Questo documento è una visione d'insieme dei problemi di sicurezza che un amministratore di sistemi Linux si trova ad affrontare. Tratta una politica di sicurezza generale e una serie di esempi specifici di come mettere meglio al sicuro da intrusori un sistema Linux. Sono anche inclusi collegamenti a materiale e programmi riguardanti. Miglioramenti, critica costruttiva, aggiunte e correzioni saranno ben accette. Per favore inviate i vostri commenti ad entrambi gli autori, con il subject "Security HOWTO".

Contents

1 Introduzione **3**

1.1 Nuove versioni di questo documento 4

1.2 Commenti 4

1.3 Avvertenze 4

1.4 Copyright Information 5

2 Prologo **5**

2.1 Perché c'è bisogno di sicurezza? 5

2.2 Quanta sicurezza mi dà la sicurezza? 5

2.3 Cosa volete proteggere? 6

2.4 Sviluppare una politica di sicurezza. 7

2.5 Mezzi per proteggere il vostro sito 7

2.5.1 Sicurezza dell'Host 7

2.5.2 Sicurezza della rete locale 8

2.5.3 Sicurezza attraverso la discrezione 8

2.6 Organizzazione di questo documento 8

3 Sicurezza Concreta **8**

3.1 Serrature per Computer 9

3.2 Sicurezza del BIOS 9

3.3 Sicurezza del Boot Loader 9

3.4 xlock e vlock 10

3.5 Scoprire Infrazioni della Sicurezza Fisica 10

4 Sicurezza Locale **11**

4.1 Creare Nuovi Account 11

4.2 Sicurezza di Root 12

5 Sicurezza dei File e dei Filesystem **13**

5.1 Settaggi di Umask . . . 14

5.2 Permessi dei File . . . 15

5.3 Controllo dell'Integrità . . . 17

5.4 Cavalli di Troia . . . 18

6 Sicurezza delle Password e Crittografia **18**

6.1 PGP e Crittografia a Chiave Pubblica . . . 19

6.2 SSL, S-HTTP, HTTPS e S/MIME . . . 19

6.3 Implementazioni IPSEC per Linux . . . 20

6.4 `ssh` (Shell Sicura) e `stelnet` . . . 20

6.5 PAM - Pluggable Authentication Modules (Moduli Aggiuntivi di Autenticazione) . . . 21

6.6 Incapsulamento Crittografico IP (CIPE) . . . 22

6.7 Kerberos . . . 22

6.8 Shadow Password. . . . 23

6.9 "Crack" e "John the Ripper" . . . 23

6.10 CFS - File System Crittografico e TCFS - File System Crittografico Trasparente . . . 23

6.11 X11, SVGA e Sicurezza della GUI . . . 23

6.11.1 X11 . . . 23

6.11.2 SVGA . . . 24

6.11.3 GGI (Generic Graphics Interface project) (Progetto di Interfaccia Grafica Generica) . . . 24

7 Sicurezza nel Kernel **24**

7.1 2.0 Kernel Compile Options . . . 25

7.2 2.2 Kernel Compile Options . . . 26

7.3 Device nel Kernel . . . 27

8 Sicurezza di Rete **27**

8.1 Sniffer di Pacchetti . . . 27

8.2 Servizi di sistema e tcp_wrappers . . . 28

8.3 Verificare i vostri DNS . . . 29

8.4 `identd` . . . 29

8.5 SATAN, ISS, e altri scanner di rete . . . 29

8.5.1 Rilevare Scansioni delle Porte . . . 30

8.6 `sendmail`, `qmail` e MTA (Agenti di Trasporto di Posta) . . . 30

8.7 Attacchi di Denial of Service (Negazione di un Servizio) . . . 31

8.8 Sicurezza del NFS (Network File System). . . . 31

8.9 NIS (Network Information Service) (già YP). . . . 32

8.10 Firewall . . . 32
8.11 IP Chains - Firewall per Linux Kernel 2.2.x . . . 33
8.12 VPN - Virtual Private Networks (Reti Private Virtuali) . . . 33

9 Preparazione della Sicurezza (prima di entrare in rete) 34
9.1 Fate un backup completo della macchina. . . . 34
9.2 Scegliere una Buona Tabella di Backup . . . 34
9.3 Fate un Backup dei Vostri Database di RPM o Debian . . . 34
9.4 Tenete Nota dei Dati degli Account . . . 34
9.5 Applicate Tutti i Nuovi Aggiornamenti di Sistema. . . . 35

10 Cosa Fare Durante e Dopo un'Intrusione 35
10.1 Compromissione della Sicurezza in Corso. . . . 35
10.2 La sicurezza è già stata compromessa. . . . 36
10.2.1 Chiudere il Buco . . . 36
10.2.2 Stimare il Danno . . . 37
10.2.3 Backup, Backup, Backup! . . . 37
10.2.4 Rintracciare l'Intrusore. . . . 37

11 Documenti sulla Sicurezza 38
11.1 Siti FTP . . . 38
11.2 Siti Web . . . 38
11.3 Mailing List . . . 39
11.4 Libri - Materiale Stampato . . . 39

12 Glossario 40

13 Domande Frequenti 41

14 Conclusioni 42

15 Ringraziamenti 43

1 Introduzione

Questo documento tratta alcune delle problematiche che riguardano la sicurezza di Linux. Inoltre vengono discusse le politiche di sicurezza da seguire e le risorse intrinseche alla rete.

Molti altri HOWTO trattano le caratteristiche della sicurezza sotto Linux, e quando necessario sono presenti riferimenti a questi testi.

Questo documento *non* è una lista aggiornata di exploit. Molti nuovi exploit vengono scoperti in continuazione. Qui leggerete dove trovare questo tipo di documentazione aggiornata, e alcune metodi generali per evitare che tali exploit vengano sfruttati sul vostro sistema.

1.1 Nuove versioni di questo documento

Nuove versioni (in lingua inglese) di questo documento verranno periodicamente spedite su *comp.os.linux.answers*. Verranno anche aggiunte a vari siti che trattano documentazione del genere, incluso:

http://www.linuxdoc.org/

Inoltre, dovreste in genere poter trovare questo documento sulla pagina WWW di Linux a:

http://metalab.unc.edu/mdw/linux.html

In fine, l'ultimissima versione di questo documento dovrebbe essere disponibile in vari formati anche da:

http://scrye.com/~kevin/lsh/

o

http://www.linuxsecurity.com/Security-HOWTO

o

http://www.tummy.com/security-howto

1.2 Commenti

Tutti i commenti, eventuali errori, informazioni aggiuntive e critiche di ogni tipo dovrebbero essere dirette a:

kevin@tummy.com

e

dave@linuxsecurity.com

Nota: Per favore mandate i vostri commenti ad *entrambi* gli autori. Inoltre, siate sicuri di includere "Linux" "security", o "HOWTO" nel subject per evitare il filtro antispam di Kevin.

1.3 Avvertenze

Non mi prendo responsabilità del contenuto di questo documento. L'utilizzo di concetti, esempi o altre informazioni contenute in questo documento è a proprio rischio e pericolo. Inoltre, questa è una versione preliminare, con possibili errori e imprecisioni.

Molti degli esempi e descrizioni si riferiscono al setup di sistema della distribuzione RedHat(tm). Il vostro sistema potrebbe variare da quanto descritto.

Per quanto ne sappiamo, verranno descritti solo programmi che possono essere usati sotto certe condizioni per scopi personali o di valutazione. La maggior parte di questi programmi, completi di sorgenti, sono disponibili sotto licenza *GNU* <http://www.gnu.org/copyleft/gpl.html>.

1.4 Copyright Information

This document is copyrighted (c)1998-2000 Kevin Fenzi and Dave Wreski, and distributed under the following terms:

- Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium, physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the authors would like to be notified of any such distributions.
- All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.
- If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at

tjbynum@metalab.unc.edu

2 Prologo

Questo documento proverà a spiegare alcune procedure e software usato comunemente per aumentare la sicurezza del proprio sistema Linux. E' importante discutere prima alcuni dei concetti di base, e creare le fondamenta della sicurezza del nostro sistema, prima di cominciare.

2.1 Perché c'è bisogno di sicurezza?

Nel mondo continuamente in movimento delle comunicazioni globali, delle connessioni ad Internet gratuite, e del veloce sviluppo del software, la sicurezza sta diventando sempre più un fattore da considerare. La sicurezza è un requisito basilare perché l'informatizzazione globale è intrinsecamente inaffidabile. Mentre i dati vanno dal punto A al punto B su Internet, per esempio, possono passare attraverso molti altri punti lungo il tragitto, lasciando ad altri la possibilità di intercettarli, e persino alterarli. Anche altri utenti sul vostro sistema possono maliziosamente trasformare i vostri dati in qualcosa che non vorreste. Degli intrusori, detti "cracker" possono ottenere l'accesso al vostro sistema, e poi usare le loro approfondite conoscenze per impersonarvi, rubarvi informazioni o anche negarvi l'accesso ai vostri stessi dati. Se vi state chiedendo qual è la differenza fra un "Hacker" e un "Cracker", leggete il documento di Eric Raymond "How to Become A Hacker" ("Come Diventare Un Hacker"), reperibile a *http://www.netaxs.com/~esr/faqs/hacker-howto.html*.

2.2 Quanta sicurezza mi dà la sicurezza?

Primo, ricordatevi che nessun sistema informatico potrà mai essere completamente sicuro. Tutto quello che potete fare è rendere sempre più difficile per gli altri compromettere il vostro sistema. Per l'utente medio di un sistema Linux casalingo non serve molto per tenere a bada un cracker occasionale. Al contrario, per sistemi Linux di alto profilo (banche, compagnie di telecomunicazioni, etc), è necessario molto più impegno.

Un altro fattore da tenere in considerazione è che più il vostro sistema è sicuro, più la sicurezza diventa intrusiva. Dovete decidere a quale punto di equilibrio il vostro sistema sarà ancora utilizzabile ma sicuro, relativamente all'uso che ne fate. Per esempio, potreste esigere che chiunque tenti di fare un login via telefono col vostro sistema usi un modem che lo richiami a casa, stabilendo così la vera connessione. Questo dà più

sicurezza, ma se qualcuno non è a casa sarà difficile che possa connettersi. Potreste anche configurare il vostro sistema Linux senza connessione a Internet, ma questo ne limita l'utilità.

Se avete un sito medio o grande, dovreste stabilire una politica di sicurezza che definisca quanto deve essere sicuro il vostro sito e quale utenza è autorizzata all'accesso. Potete trovare un famoso esempio di politica di sicurezza a *http://www.faqs.org/rfcs/rfc2196.html*. E' stato recentemente aggiornato, e contiene un'ottima guida per stabilire una politica di sicurezza per la vostra compagnia.

2.3 Cosa volete proteggere?

Prima di tentare di rendere sicuro il vostro sistema, dovete decidere contro che tipo di attacco proteggervi, quali rischi dovreste o non dovreste correre, e quanto vulnerabile ne risulta il sistema. Dovreste analizzare il sistema per sapere cosa state proteggendo, perché lo state proteggendo, quanto vale e chi ne ha la responsabilità.

- Il *rischio* è la possibilità che un intrusore riesca nel tentativo di accedere al vostro computer. Un intrusore può leggere o scrivere file, o eseguire programmi che potrebbero fare danno? Può cancellare dei dati? Può bloccare il lavoro vostro e della vostra compagnia? Non dimenticate: qualcuno che ha accesso al vostro account, o al vostro sistema, può anche impersonarvi. Inoltre, avere un account insicuro nel vostro sistema può compromettere l'intera rete. Se permettete anche ad un solo utente di fare un login usando un file `.rhosts`, o di usare un servizio insicuro come `tftp`, rischiate di permettere ad un intrusore di "aprire una breccia". Una volta che l'intrusore ha un account sul vostro sistema, o sul sistema di qualcun altro, può usarlo per avere accesso ad un altro sistema, o un altro account.
- Una *minaccia* viene tipicamente da qualcuno con un movente per cercare di avere accesso non autorizzato alla vostra rete o computer. Dovete decidere di chi potete fidarvi, e che tipo di minaccia potrebbe rappresentare. Ci sono diversi tipi di intrusori, ed è utile tenere a mente le loro caratteristiche quando configurate le difese del vostro sistema.
  - **Il Curioso** - Questo tipo di intrusore è principalmente interessato nel capire che tipo di dati e di sistema avete.
  - **Il Malizioso** - Questo tipo di intrusore in genere tenta di mettere fuori uso il vostro sistema, o sfregiare la vostra pagina web, o altrimenti di farvi perdere tempo e denaro per riparare i danni che causa.
  - **L'Intrusore di Alto Profilo** - Questo tipo di intrusore tenta di usare il vostro sistema per guadagnare popolarità o fama. Potrebbe usare il vostro sistema di alto profilo per pubblicizzare le sue abilità
  - **La Concorrenza** - Questo tipo di intrusore ha interesse nei dati che avete sul vostro sistema. Potrebbe essere qualcuno che pensa che abbiate qualcosa di cui potrebbe beneficiare, finanziariamente o in altro modo.
  - **Lo Sfruttatore** - Questo tipo di intrusore è interessato a sfruttare il vostro sistema e ad usare le sue risorse per i propri fini. In genere eseguirà server di chat o irc, siti con archivi porno o addirittura server DNS.
  - **Il Clandestino** - Questo tipo di intrusore è interessato del vostro computer come trampolino per entrare in altri sistemi. Se il vostro sistema ha una buona connessione o è il passaggio verso molti altri host, è probabile che vi troviate ad affrontare questo tipo di intrusore.
- La vulnerabilità esprime quanto bene il vostro computer è protetto dalle altre reti, e la possibilità che qualcuno ottenga senza permesso l'accesso. Cosa è in gioco se qualcuno entra nel vostro sistema? Ovviamente le preoccupazioni di un utente casalingo con un IP dinamico saranno diverse da quelle di una compagnia che si connetta ad Internet o ad un'altra grande rete.

Quanto tempo richiederebbe recuperare/ricreare dei dati andati persi? Un investimento di tempo ora può risparmiare dieci volte tanto in seguito, se doveste recuperare dati perduti. Avete controllato il vostro sistema di backup recentemente e avete verificato i vostri dati?

2.4 Sviluppare una politica di sicurezza.

Create una strategia semplice e generica per il vostro sistema che gli utenti possano facilmente capire e seguire. Dovrebbe proteggere, oltre ai dati a cui tenete, anche la privacy degli utenti. Alcune cose che potrete aggiungere sono: chi ha accesso al sistema (Il mio amico può usare il mio account?), a chi ѐpermesso di installare programmi, chi possiede quali dati, recupero di "disastri" e uso appropriato del sistema.

Una strategia di sicurezza generalmente approvata comincia col motto:

Quel che non è permesso è proibito

Questo significa che almeno che non abbiate permesso a un utente l'accesso a un servizio, quell'utente non dovrebbe usare quel servizio finché non lo permettete. Assicurate che la vostra strategia funzioni davvero sull'account dell'utente normale. Dire, "Ah, non riesco a risolvere questo problema di privilegi, lo farò da root." può portare a buchi nella sicurezza evidenti, o persino ad altri che non sono ancora stati scoperti.

rfc1244 è un documento che spiega come creare una vostra strategia di sicurezza di rete.

rfc1281 è un documento che mostra una strategia di esempio con descrizioni dettagliate di ogni passo.

In fine potreste dare un'occhiata all'archivio COAST presso *ftp://coast.cs.purdue.edu/pub/doc/policy* come sono delle strategie applicabili alla pratica.

2.5 Mezzi per proteggere il vostro sito

Questo documento discuterà diversi mezzi con cui potete mettere al sicuro la struttura per cui avete lavorato duro: la vostra macchina locale, i vostri dati, i vostri utenti, la vostra rete, persino la vostra reputazione. Cosa succederebbe alla vostra reputazione se un intrusore cancellasse dati dei vostri utenti? O sfregiasse il vostro sito? O pubblicasse il piano corporativo della vostra società per la prossima stagione? Se state pianificando l'installazione di una rete, ci sono diversi fattori da considerare prima di aggiungere una singola macchina.

Anche se avete un solo account PPP, o appena un piccolo sito, questo non significa che gli intrusori non saranno interessati a voi. Grandi sistemi di alto profilo non sono i soli bersagli – molti intrusori vogliono solo passare la sicurezza di più siti possibile, ignorando le loro dimensioni. Inoltre, possono usare un buco nella sicurezza del vostro sistema per riuscire ad entrare in altri siti a cui siete connessi.

Questi individui hanno tutto il tempo che vogliono, e possono evitare di capire come avete protetto il vostro sistema semplicemente provando tutte le possibilità. C'è inoltre una serie di ragioni per cui un intrusore può interessarsi ai vostri sistemi, che discuteremo più avanti.

2.5.1 Sicurezza dell'Host

Forse l'area di sicurezza su cui gli amministratori si concentrano di più è la sicurezza dei singoli host. Questo in genere implica di assicurarsi che il proprio sistema sia sicuro, e di sperare che tutti gli altri nella rete faccia lo stesso. Scegliere buone password, rendere sicuri i sevizi di rete del vostro host locale, tenere buoni log degli account, e aggiornare i programmi che notoriamente hanno buchi di sicurezza sono fra le cose di cui un amministratore è responsabile. Anche se questo è assolutamente necessario, può diventare un compito pesante quando la vostra rete diventa più estesa di qualche macchina.

2.5.2 Sicurezza della rete locale

La sicurezza della rete è necessaria quanto quella dei singoli host. Con centinaia, migliaia, o addirittura più computer in rete, non potete sperare che ognuno di essi sia sicuro. Assicurare che solo utenti autorizzati possano accedere al sistema, alzare firewall, usare molto la crittografia, e evitare che non ci siano macchine pericolose (cioè insicure) nella vostra rete sono tutti compiti dell'amministratore della sicurezza di rete.

Questo documento discuterà le tecniche usate per rendere sicuro il vostro sito e mostrerà alcuni modi per impedire che un intrusore riesca ad accedere a quello che state tentando di proteggere.

2.5.3 Sicurezza attraverso la discrezione

Un tipo di sicurezza che deve essere discusso è la "sicurezza attraverso la discrezione". Questo significa, per esempio, spostare un servizio che è notoriamente "fragile" per la sicurezza verso una porta non standard nella speranza che chi vi attacca non noti che è lì e non lo sfrutti. State certi che lo possono trovare e sfruttare. La sicurezza attraverso la discrezione non è assolutamente sicurezza. Semplicemente perché avete un piccolo sito, o di profilo relativamente basso, non significa che un intrusore non sarà interessato da quello che avete. Discuteremo cosa proteggere nelle prossime sezioni.

2.6 Organizzazione di questo documento

Questo documento è stato diviso in una serie di sezioni. Trattano diversi argomenti riguardanti la sicurezza. Il primo, 3 (Sicurezza concreta), tratta della protezione dalle intrusioni. Il secondo, 4 (Sicurezza Locale), descrive come proteggere il sistema dagli utenti locali. Il terzo, 5 (Sicurezza dei File e dei Filesystem), mostra come configurare il filesystem e i privilegi sui file. Il successivo, 6 (Sicurezza delle Password e Crittografia), discute come usare la crittografia per rendere più sicura la vostra macchina e la vostra rete. 7 (Sicurezza del Kernel) discute quali opzioni del kernel dovreste attivare o conoscere per un sistema più sicuro. 8 (Sicurezza della Rete), descrive come rendere il vostro sistema Linux più difficile da attaccare via rete. 9 (Preparare la Sicurezza), discute come preparare le vostre macchine prima di metterle in rete. Quindi, 10 (Cosa Fare Durante e Dopo un'Intrusione), tratta cosa fare quando scoprite una compromissione in atto o una successa recentemente. In 11 (Risorse per la Sicurezza), sono citate alcune importanti risorse per la sicurezza. La sezione D e R 13 (Domande Frequenti), risponde ad alcune domande frequenti, e in fine troverete una conclusione in 14 (Conclusione).

I due problemi da tenere a mente leggendo questo documento sono:

- Conoscere il vostro sistema. Controllate i log di sistema, come `/var/log/messages`, tenete d'occhio il vostro sistema, e
- Tenete aggiornato il vostro sistema assicurandovi di avere installato le versioni aggiornate del software. Già questi accorgimenti renderanno decisamente più sicuro il vostro sistema.

3 Sicurezza Concreta

Il primo grado di sicurezza che dovete considerare è la sicurezza pratica dei vostri sistemi. Chi ha fisicamente accesso ai computer? Dovrebbe averlo? Riuscite a proteggere la vostra macchina da eventuali intrusioni?

Di quanta sicurezza fisica avete bisogno sul vostro sistema dipende molto dalla situazione e/o dal budget.

Se siete un utente casalingo, probabilmente non è molto importante (anche se potreste aver bisogno di proteggervi dalla curiosità di bambini o parenti). Se si tratta di un laboratorio, ne avrete più bisogno, ma gli utenti avranno comunque bisogno di non essere limitati nel lavoro. Molte delle seguenti sezioni saranno

d'aiuto. Se siete in un ufficio, potreste avere bisogno o no di tenere al sicuro le vostre macchine fuori dalle ore di lavoro o quando non siete presenti. In certe compagnie, lasciare incustodita la vostra macchina è un errore da licenziamento.

Ovvi metodi di sicurezza come lucchetti, macchine chiuse a chiave e video sorveglianza sono buone idee, ma vanno oltre il tema di questo documento. :)

3.1 Serrature per Computer

Molti case dei moderni PC includono la possibilità di essere chiusi. In genere hanno una toppa sulla parte frontale del case che può essere alternate fra le posizioni "chiuso" e "aperto con una chiave. Queste serrature possono aiutare ad evitare che qualcuno rubi il vostro PC, o che apra il case e manipoli o rubi i componenti. In alcuni casi possono anche impedire che qualcuno riavvii il computer con un proprio floppy o con un componenti modificati.

Queste serrature hanno funzioni diverse a seconda che siano riconosciute dalla scheda madre e di come è costruito il case. Su molti PC fanno in modo di obbligarvi a rompere il case per aprirlo. In altri, non vi permetteranno neanche di inserire nuove tastiere o mouse. Controllate le istruzioni della vostra scheda madre o del case per maggiori informazioni. A volte questa può essere una caratteristica molto utile, anche se le serrature sono in genere di bassissima qualità e possono essere facilmente essere passate con un po' di scasso.

Alcune macchine (soprattutto SPARC e Mac) hanno un gancio sul retro in cui far passare una catena che un ipotetico intrusore dovrebbe rompere per aprire il case. Semplicemente metterci un lucchetto può essere un deterrente contro i furti.

3.2 Sicurezza del BIOS

Il BIOS è il più basso livello software per configurare il vostro hardware basato su tecnologia x86. LILO e altri metodi di boot di Linux accedono al BIOS per capire come avviare la macchina. Altro hardware su cui gira Linux ha un software simile (OpenFirmware sui Mac e i nuovi Sun, la PROM di avvio di Sun, ecc...). Potete usare il vostro BIOS per evitare che un intrusore riavvii la vostra macchina e manipoli il vostro sistema Linux.

Molti BIOS di PC vi permettono di usare una password di avvio. Questo non dà molta sicurezza (il BIOS può essere resettato o rimosso se qualcuno può aprire il case), ma potrebbe essere un buon deterrente (cioè prenderà del tempo e lascerà delle tracce). Allo stesso modo, su S/Linux (Linux per macchine con processore SPARC(tm)), la vostra EEPROM può essere configurata per richiedere una password di avvio. Questo può rallentare l'intrusore.

Molti BIOS x86 vi permettono anche di usare diversi altri utili settaggi di sicurezza. Leggete il manuale del vostro BIOS o dategli un'occhiata quando riavviate la prossima volta. Per esempio alcuni BIOS bloccano l'avvio da floppy e altri richiedono una password per modificare certe caratteristiche del BIOS.

Nota: se avete una macchina server e inserite una password di avvio, questa non ripartirà da sola. Ricordate che dovrete andare a mettere la password se mancasse la corrente. ;(

3.3 Sicurezza del Boot Loader

Anche i vari boot loader per Linux possono avere una password. LILO, per esempio, ha le opzioni `password` e `restricted`; `password` richiede una password all'avvio, mentre `restricted` richiede una password solo se specificate opzioni (come `single`) al prompt di `LILO`.

Dalla pagina man di lilo.conf:

```
password=password
              L'opzione 'password=...' (vedi sotto) ha effetto su ogni partizione.

restricted
              L'opzione 'restricted' (vedi sotto) ha effetto su ogni partizione.

       password=password
              proteggi l'immagine con una password.

       restricted
              Serve una password per avviare l'immagine solo se vengono
              specificati parametri alla linea di comando (per es. single).
```

Tenete presente quando settate tutte queste password che poi dovrete ricordarvele. :) Ricordatevi anche che tutte queste password si limiteranno a rallentare un intrusore ben determinato. Non impediranno di avviare da un floppy e quindi montare la partizione di root. Se state usando questo tipo di sicurezza dovreste anche disabilitare il boot da floppy nel BIOS del computer, e quindi proteggerlo con una password.

Se qualcuno avesse informazioni sulla sicurezza dei boot loader, ci piacerebbe averle. (`grub`, `silo`, `milo`, `linload`, etc).

Nota: Se avete una macchina server e inserite una password di boot, la vostra macchina *non* si riavvierà da sola. Ricordate che dovrete andare a mettere la password se andasse via la corrente. ;(

3.4 xlock e vlock

Se andate via dal computer ogni tanto, torna utile poter "bloccare" la macchina così che nessuno possa manomettere, o vedere, il vostro lavoro. Due programmi che lo fanno sono: `xlock` e `vlock`.

`xlock` blocca il video di una sessione di X. Dovrebbe essere incluso in ogni distribuzione che supporti X. Controllate la sua pagina man per trovare altre opzioni, ma in genere potete eseguire `xlock` da xterm per bloccare il monitor e obbligare ad inserire una password per sbloccarlo.

`vlock` è un semplice programmino che vi permette di bloccare alcune o tutte le console virtuali del vostro Linux. Potete bloccarle tutte o solo quella che state usando. Se ne bloccate una sola, altri potranno venire ad usare le altre; semplicemente non potranno usare la vostra finché non la sbloccate. `vlock` è distribuito con RedHat Linux, ma non è detto che sia l'unica.

Chiaramente bloccare la vostra console impedirà ad altri di manipolare il vostro lavoro, ma non di riavviare la macchina o distruggere il vostro lavoro in altri modi. Inoltre non evita che qualcuno acceda alla vostra macchina dalla rete e causi problemi

Ancora più importante, non impedisce che qualcuno esca da X e, passando a un normale prompt di login di una console virtuale o alla VC da cui X11 era stato avviato, lo sospenda, ottenendo così i vostri privilegi.

3.5 Scoprire Infrazioni della Sicurezza Fisica

La prima cosa da notare è quando la macchina è stata riavviata. Visto che Linux è un SO stabile e robusto, dovrebbe essere riavviato solo quando *voi* lo riavviate per aggiornamenti del SO, cambiamenti all' hardware, o cose del genere. Se la vostra macchina si è riavviata senza un vostro comando, potrebbe essere il segno della manomissione di un intrusore. Molti dei metodi di manomissione richiedono un riattivo o uno spegnimento del computer.

Cercate segni di manomissione sul case o intorno al computer. Anche se molti intrusori cancellano le tracce della loro presenza dai log, è buona norma controllarli tutti e cercare delle discrepanze.

E' una buona idea anche tenere i log in un posto sicuro, come un log server dedicato all'interno di una rete ben protetta. Una volta che una macchina è stata compromessa, i log perdono utilità perché probabilmente sono anch'essi stati manomessi.

Il demone syslog può essere configurato per mandare automaticamente dati di log a un server syslog centralizzato, ma in genere viene mandato non crittato, permettendo a un intrusore di vedere i dati mentre vengono trasferiti . Questo può rivelare informazioni sulla vostra rete che non ddovrebbero essere pubbliche. Ci sono anche demoni syslog che crittano i dati inviati.

Tenete in conto anche del fatto che falsificare messaggi del syslog è facile – con un exploit che è stato pubblicato. Syslog accetta anche log di rete che dichiarino di venire da local host senza indicare la loro vera origine.

Alcune cose da controllare nei vostri log:

- Log brevi o incompleti.
- Log che contengano date improbabili.
- Log con permessi sbagliati.
- Registrazioni di riavvii o di servizi riavviati.
- Log mancanti.
- Uso fuori logo di `su` o login da posti strani.

Parleremo dei dati di log più9.4 (avanti) nel HOWTO.

4 Sicurezza Locale

La prossima cosa da controllare è la sicurezza contro attacchi da utenti locali. Abbiamo detto solo utenti *locali*? Si!

Ottenere l'accesso all'account di un utente locale è uno dei primi tentativi che gli intrusori fanno per arrivare all'account di root. Con una debole sicurezza locale, possono "promuovere" il loro accesso di utente ad accesso di root usando una serie di bug e servizi locali mal configurati. Se stringerete le maglie della vostra sicurezza locale un intrusore dovrà saltare un ulteriore ostacolo.

Gli utenti locali possono creare un sacco di danni anche se sono veramente chi dicono di essere. è una pessima idea fornire account a persone che non conoscete o di cui non avete informazioni su come contattarli.

4.1 Creare Nuovi Account

Dovreste essere certi di accordare agli utenti solo i privilegi indispensabili per il lavoro che devono svolgere. Se date a vostro figlio (10 anni) un account, potreste volere che abbia accesso solo a un programma di scrittura o di disegno, ma che non possa cancellare dati non suoi.

Diverse regolette da seguire quando si fornisce ad altri un accesso legittimo al vostro sistema Linux:

- Dare loro meno privilegi possibili.
- Sapere da dove e quando fanno un login o dovrebbero farlo.
- Assicurarsi di rimuovere gli account inutilizzati

- E' consigliabile per la manutenzione l'uso della stessa userid su tutti i computer e le reti, e inoltre permette una più facile analisi dei log.
- LA creazione di userid di gruppo dovrebbe essere assolutamente proibita. Gli account degli utenti permettono l'attribuzione delle responsabilità, e questo non è possibile con account di gruppo.

Molti account locali che sono sfruttati in infrazioni di sicurezza non sono stati usati per mesi o anni. Visto che nessuno li usa, sono un ideale mezzo di attacco.

4.2 Sicurezza di Root

L'account più preso di mira sulla vostra macchina è l'account di root, o superuser. Questo account ha autorità su tutta la macchina, che può anche includere l'autorità su altre macchine sulla rete. Ricordate che dovreste usare l'account di root solo per compiti specifici e molto brevi, usando quindi per la maggior parte del tempo il vostro utente normale. Anche piccoli errori fatti da root possono causare problemi. Meno usate i vostri privilegi di root, più sarete al sicuro.

Alcuni trucchi per evitare di fare danni da root sul vostro computer:

- Quando eseguite un comando complesso, provate prima ad eseguirlo in modo non distruttivo... soprattutto programmi che usano le wildcard: e.g., se volete eseguire `rm foo*.bak`, prima usate `ls foo*.bak` e assicuratevi di star per cancellare i file che volete davvero. Anche usare `echo` al posto di comandi distruttivi può andare bene.
- Fornite ai vostri utenti un alias al comando `rm` per chiedere conferma della cancellazione dei file.
- Usate root solo per portare a termine singoli specifici compiti. Se vi rendete conto che state cercando di capire come fare qualcosa, tornate alla shell di utente normale finche non sarete *sicuri* di cosa fare da root.
- Il path dei comandi per l'utente root è molto importante. Il path dei comandi (ciò la variabile d'ambiente `PATH`) specifica le directory in cui la shell cerca i programmi. Provate a limitare il path di root il più possibile e non includete *mai* . (che significa "la directory corrente") nella variabile PATH. Inoltre, non mettete mai directory scrivibili nel path, perché potrebbe permettere a degli intrusori di modificare o inserire nuovi eseguibili nel path, permettendo loro di divenire root la prossima volta che eseguirete quel comando.
- Non usate mai la suite di comandi rlogin/rsh/rexec (dette le r-utilities) da root. Sono soggette a molti tipi di attacco e sono decisamente pericolose se eseguite da root. Non create mai un file `.rhosts` per root.
- Il file `/etc/securetty` contiene una lista di terminali da cui root può fare login. Per default (su RedHat Linux) è impostata sulle sole console virtuali locali (vtys). Siate consapevoli di cosa fate quando aggiungete qualcosa a questo file. Sarebbe meglio loggarsi da remoto come utente normale e poi usare `su` se ne avete bisogno (preferibilmente attraverso `6.4 (ssh)` o un altro canale crittato), in modo che non sia necessario fare un login remoto da root.
- Siate sempre calmi e riflessivi quando siete root. Le vostre azioni possono avere effetti su molte cose. Pensate prima di digitare!

Se dovete assolutamente permettere a qualcuno (possibilmente molto fidato) di accedere da root alla vostra macchina, ci sono un paio di strumenti che possono essere d'aiuto. `sudo` permette agli utenti di usare la loro password per accedere a una gamma limitata di comandi come root. Questi vi permetterebbe, per esempio, di lasciare che un utente espella e monti media removibili sul vostro computer, senza avere altri privilegi di

root. `sudo` inoltre tiene un log di tutti i tentativi, riusciti e non, di usarlo, permettendovi di rintracciare chi ha usato il comando per fare cosa. Per questa ragione `sudo` funziona bene persino in posti dove molte persone hanno accesso di root, perché vi permette di rintracciare i cambiamenti fatti.

Nonostante `sudo` possa essere usato per dare a specifici utenti specifici privilegi per specifici lavori, ha alcune mancanze. Dovrebbe essere usato solo per una limitata serie di compiti, come riavviare un server o aggiungere utenti. Ogni programma che offre un modo per tornare alla shell darà un accesso di root a un utente che lo usi attraverso `sudo`. Questo include la maggior parte degli editor, per esempio. Inoltre, un programma innocuo come `/bin/cat` può essere usato per sovrascrivere file, il che permetterebbe di prendere possesso del superuser. Considerate `sudo` come un mezzo per conoscere le responsabilità di certe azioni, e non sperare che possa sostituire root rimanendo sicuro.

5 Sicurezza dei File e dei Filesystem

Alcuni minuti di preparazione e pianificazione prima di mettere in funzione i vostri sistemi può aiutare a proteggere loro e i loro dati.

- Non ci dovrebbe essere alcuna ragione per cui le home directory degli utenti permettano di usare programmi SUID/SGID. Usate l'opzione `nosuid /etc/fstab` per partizioni scrivibili da chi non è root. Forse vorrete anche usare `nodev` e `noexec` sulle partizioni home degli utenti e su `/var`, proibendo così l'uso di programmi e la creazioni di character o block device, che comunque non dovrebbe mai essere necessaria.

- Se esportate filesystem usando NFS, siate sicuri di configurare `/etc/exports` con l'accesso più restrittivo possibile. Questo significa non usare wildcard, non permettere accesso in scrittura di root, e esportare solo in lettura quando è possibile.

- Configurate l'`umask` di creazione dei file dei vostri utenti in modo che sia più restrittiva possibile. Cfr. 5.1 (settaggi di umask).

- Se montate filesystem usando un filesystem di rete come NFS, siate sicuri di configurare /etc/exports con restrizioni adeguate. Tipicamente, usare nodev, nosuid, e magari noexec, è consigliabile.

- Impostate limiti per il filesystem invece di lasciarlo `illimitato` come di default. Potete controllare i limiti per utente usando il modulo PAM per la limitazione delle risorse e `/etc/pam.d/limits.conf`. Per esempio, i limiti per il gruppo `users` potrebbero essere:

  ```
  @users     hard  core    0
  @users     hard  nproc   50
  @users     hard  rss     5000
  ```

  Così si proibisce la creazione di file core, si limita il numero di processi a 50 e la memoria disponibile ad ogni utente a 5 MB.

- I file `/var/log/wtmp` e `/var/run/utmp` contengono i record di login per tutti gli utenti del sistema. LA loro integrità deve essere conservata perché possono essere usati per determinare quando e da dove un utente (o un potenziale intrusore) è entrato nel sistema. Questi file dovrebbero avere i permessi `644`, senza limitare il normale uso del sistema.

- Il bit immutable può essere usato per prevenire la cancellazione o sovrascrittura accidentale di un file che deve essere protetto. Inoltre evita che qualcuno crei un link simbolico al file (questi link simbolici sono stati l'origine di attacchi che hanno cancellato `/etc/passwd` o `/etc/shadow`). Cfr. la man page `chattr`(1) per informazioni sul bit immutable.

- I file SUID e SGID sono un potenziale rischio, e dovrebbero essere tenuti d'occhio. Visto che questi programmi danno speciali privilegi all'utente che li esegue, è necessario assicurarsi che non vengano istallati programmi insicuri. Un trucco molto diffuso fra i cracker è sfruttare programmi con SUID-root, quindi lasciare un programma SUID a fare da backdoor per entrare la prossima volta, anche se il buco originale viene chiuso. Trovate tutti i programmi SUID/SGID sul vostro sistema, e tenete appunti su cosa sono, così da essere al corrente di qualsiasi cambiamento che potrebbe indicare un eventuale intrusore. Usate questo comando per trovare tutti i programmi con SUID/SGID sul vostro sistema:

  ```
  root#  find / -type f \( -perm -04000 -o -perm -02000 \)
  ```

  La distribuzione Debian esegue un job ogni sera per determinare quali file con SUID esistano. Quindi li compara con quelli della sera precedente. Potete cercare questo log in `/var/log/setuid*`.

  Potete togliere i permessi SUID o SGID da un programma sospetto con `chmod`, quindi rimetterli se pensate che siano assolutamente necessari.

- I file scrivibili da tutti, soprattutto i file di sistema, possono essere un buco nella sicurezza se un cracker accede al vostro sistema e li modifica. Inoltre, le directory scrivibili da tutti sono pericolose: permettono a un cracker di aggiungere o cancellare file come vuole. Per trovare tutti i file di libera scrittura sul vostro sistema, usate il comando:

  ```
  root# find / -perm -2 ! -type l -ls
  ```

  e assicuratevi di sapere perché quei file sono scrivibili. Durante l'uso normale alcuni file saranno liberamente scrivibili, inclusi alcuni da `/dev`, e i link simbolici, da cui il `! -type l` che li esclude dal comando `find`.

- Dei file senza un proprietario possono indicare che qualcuno è entrato nel vostro sistema. Potete trovare i file senza proprietario, o senza gruppo, col comando:

  ```
  root# find / -nouser -o -nogroup -print
  ```

- Cercare file `.rhosts` dovrebbe essere uno dei vostri compiti di amministratore , visto che questi file non dovrebbero essere permessi sul vostro sistema. Ricordate che a un cracker basta un solo account insicuro per poter avere accesso a tutta la rete. Potete trovare tutti i file `.rhosts` sul vostro sistema col seguente comando:

  ```
  root# find /home -name .rhosts -print
  ```

- Per finire, prima di cambiare i permessi su un qualsiasi file di sistema, siate certi di sapere quello che fate. Non cambiate mai i permessi di un file perché sembra la via più facile di far funzionare le cose. Determinate sempre la ragione per cui quel file ha quei permessi prima di cambiarli.

5.1 Settaggi di Umask

Il comando `umask` può essere usato per stabilire la modalità standard di creazione dei file. E' il complementare ottale della modalità desiderata. Se i file venissero creati senza tenere in conto i settaggi dei loro permessi, un utente potrebbe inavvertitamente dare permessi di lettura o scrittura a qualcuno che non li dovrebbe avere. I tipici settaggi di `umask` includono `022`, `027`, e `077` (che è il più restrittivo). Normalmente la umask viene impostata in `/etc/profile`, così che abbia effetto su tutti gli utenti sul sistema. La maschera di creazione dei file si calcola sottraendo da 777 il valore desiderato. In altre parole, una umask di 777 farebbe in modo che i file creati non contengano permessi di lettura, scrittura o esecuzione per nessuno. Una maschera di 666 creerebbe nuovi file con permessi 111. Per esempio, potreste avere una linea di questo genere:

```
# Imposta la maschera di default degli utenti
umask 033
```

In questo caso, le nuove directory avrebbero permesso 744, ottenuto dalla sottrazione di 033 da 777. I nuovi file avrebbero permesso 644. Siate sicuri che la umask di root sia `077`, che impedirà la lettura, scrittura e esecuzione ad altri utenti, eccetto che per quei file che siano stati esplicitamente cambiati con chmod `chmod`.

Se usate RedHat, e aderite al loro schema di creazione di ID di utenti e gruppi (User Private Groups), sarà sufficiente usare `002` per la vostra `umask`. Questo è dovuto al fatto che la configurazione di default è di utente per gruppo.

5.2 Permessi dei File

è importante assicurarsi che i vostri file di sistema non siano aperti da utenti e gruppi che non dovrebbero fare manutenzione di sistema.

Unix separa il controllo di accesso ai file e alle directory secondo tre caratteristiche: proprietario, gruppo e altri. C'è sempre un solo proprietario, un numero variabile di membri del gruppo, e tutti gli altri.

Ecco un veloce spiegazione dei permessi di Unix:

Proprietà - Quale/i utente/i e gruppo/i ha controllo sui permessi del nodo.

Permessi - Bit che possono essere settati o resettati per concedere certi tipi di accesso. I permessi delle directory possono avere significati diversi dai corrispondenti permessi sui file.

Lettura:

- Poter vedere i contenuti del file
- Poter aprire la directory

Scrittura:

- Poter aggiungere parti o fare modifiche a un file
- Poter cancellare o spostare i file in una directory

Esecuzione:

- Poter eseguire un programma binario o script di shell
- Poter eseguire una ricerca nella directory, se con il permesso di lettura

Attributo Save Text: (Per le directory)

Lo "sticky bit" ha anche un significato diverso quando applicato a directory e quando a file. Se lo sticky bit ìmpostato per una directory, allora un utente può solo cancellare file di cui è proprietario o di cui ha espliciti permessi di scrittura, anche se ha accesso in scrittura alla directory. Ciò è stato progettato per directory come `/tmp`, che sono scrivibili a tutti, ma in cui si preferisce che non tutti possano cancellare file a volontà. Lo sticky bit è segnato come una `t` nel listato di una directory.

Attributo SUID: (Per i file)

Descrive permessi set-user-id sul file. Quando la modalità di accesso set user ID è impostata nei permessi del proprietario, e il file è eseguibile, i processi che lo eseguono hanno accesso alle risorse di sistema basati sul proprietario del file, invece che sull'utente che ha creato il processo. Questa è la causa di molti exploit basati sul buffer overflow.

Attributo SGID: (Per i file)

Se impostato nei permessi del gruppo, questo bit controlla lo stato "set group id" di un file. In pratica si comporta come SUID, escluso che si basa sul gruppo. Il file deve essere eseguibile perché questo abbia un effetto.

Attributo SGID: (Per le directory)

Se impostate il bit SGID su una directory (con `chmod g+s` *`directory`*), I file creati in quella directory avranno il gruppo impostato sul gruppo della directory.

Voi - Il proprietario del file

Gruppo - Il gruppo a cui appartenete

Tutti - Chiunque non sia il proprietario o membro del gruppo

File Esempio:

```
-rw-r--r--  1 kevin  users         114 Aug 28  1997 .zlogin
1° bit - directory?                       (no)
 2° bit - lettura per il proprietario?     (yes, per kevin)
  3° bit - scrittura per il proprietario?   (yes, per kevin)
   4° bit - esecuzione per il proprietario?  (no)
    5° bit - lettura per il gruppo?           (yes, per users)
     6° bit - scrittura per il gruppo?         (no)
      7° bit - esecuzione per il gruppo?        (no)
       8° bit - lettura per tutti?               (yes, per tutti)
        9° bit - scrittura per tutti?             (no)
         10° bit - esecuzione per tutti?           (no)
```

Le seguenti linee sono esempi dei permessi minimi richiesti per l'accesso a lato. Forse vorrete dare più permessi di quanto vedete qui, ma questo mostra solo ciò che questi permessi minimi fanno:

```
-r--------  Permette accesso in lettura al file per il proprietario
--w-------  Permette al proprietario di modificare o cancellare il file
            (Notate che chiunque con permesso di scrittura alla directory
            in cui si trova il file ha lo stesso privilegio)
---x------  Il proprietario può eseguire questo programma, ma non script
            della shell, che hanno bisogno anche del permesso in lettura
---s------  Il file verrà eseguito con User ID = utente
--------s-  Il file verrà eseguito con Group ID = gruppo
-rw------T  Non viene segnata l'ultima modifica. Usato spesso per file di swap
---t------  Nessun effetto.  (Era lo sticky bit)
```

Esempio di directory:

```
drwxr-xr-x  3 kevin  users         512 Sep 19 13:47 .public_html/
1° bit - directory?                       (si, contiene molti file)
 2° bit - lettura per il proprietario?      (si, per kevin)
  3° bit - scrittura per il proprietario?   (si, per kevin)
   4° bit - esecuzione per il proprietario?  (si, per kevin)
    5° bit - lettura per il gruppo?           (si, per users)
     6° bit - scrittura per il gruppo?         (no)
```

```
7° bit - esecuzione per il gruppo?        (si, per users)
 8° bit - lettura per tutti?               (si, per tutti)
  9° bit - scrittura per tutti?             (no)
   10° bit - esecuzione per tutti?            (si, per tutti)
```

Le seguenti linee sono esempi dei permessi minimi richiesti per l'accesso a lato. Forse vorrete dare più permessi di quanto vedete qui, ma questo mostra solo ciò che questi permessi minimi fanno:

```
dr--------  I contenuti possono essere listati, ma gli attributi dei file non
            possono essere letti
d--x------  La directory può essere aperta e usata nei percorsi di
            esecuzione
dr-x------  Gli attributi dei file possono essere letti dal proprietario
d-wx------  I file possono essere creati/cancellati, anche se la directory non
            è quella corrente
d------x-t  Impedisce che i file siano cancellati da chi ha i permessi in
            scrittura. Usato su /tmp
d---s--s--  Nessun effetto
```

I file di configurazione del sistema (di solito in `/etc`) so in genere in modo `640` (`-rw-r---`), e proprietà di root. A seconda delle necessità: di sicurezza del vostro sistema potreste cambiare questa impostazione. Non lasciate mai che dei file di sistema siano scrivibili da un gruppo o da tutti. Alcuni file di configurazione, incluso `/etc/shadow`, dovrebbero essere leggibili solo da root, e le directory in `/etc` non dovrebbero essere accessibili da altri.

Script della Shell con SUID

Gli script della shell con SUID sono un serio rischio per la sicurezza, per cui il kernel non li considererà. A prescindere da quanto pensate che lo script sia sicuro, può essere sfruttato per dare a un cracker una shell di root

5.3 Controllo dell'Integrità

Un altro ottimo modo per rilevare attacchi locali (e anche di rete) è eseguire un programma che faccia un controllo d'integrità come `Tripwire`, `Aide` o `Osiris`. Questi programmi eseguono una serie di controlli su tutti i vostri binari importanti e sui file di configurazione e li compara con un database di valori precedenti che si presumono corretti. In questo modo, ogni cambiamento nei file verrà segnalato.

E' una buona idea installare questo tipo di programmi in un floppy e quindi proteggerlo fisicamente dalla scrittura. Così degli intrusori non potranno sabotare il programma per il controllo o cambiare i database. Una volta che avrete impostato una programma del genere, è una buona idea eseguirlo come parte dei vostri compiti amministrativi di routine per vedere se qualcosa è cambiato.

Potreste persino aggiungere un elemento al `crontab` per eseguire il controllo ogni notte e inviarvi un e-mail con i risultati al mattino. Qualcosa come:

```
# imposta mailto
MAILTO=kevin
# esegui Tripwire
15 05 * * * root /usr/local/adm/tcheck/tripwire
```

vi spedirà un rapporto ogni mattina alle 5:15.

I controlli d'integrità possono essere una manna dal cielo per rilevare in tempi brevi un intrusore. Visto che molti file cambiano su un sistema normale, fate attenzione a cosa è l'attività di un cracker e cosa è la vostra attività.

Potete trovare la versione open source e gratuita di `Tripwire` presso *http://www.tripwire.org*. Manuali e supporto sono invece a pagamento.

`Aide` si trova presso *http://www.cs.tut.fi/~rammer/aide.html*.

`Osiris` si trova presso *http://www.shmoo.com/osiris/*.

5.4 Cavalli di Troia

"Cavalli di Troia" prende il nome dal famoso inganno nell'Iliade di Omero. Il concetto è che un cracker distribuisce un programma o un binario che sembra attraente, e incita altre persone a scaricarlo ed eseguirlo come root. A quel punto il programma può compromettere il loro sistema mentre non se lo aspettano. Mentre pensano che il programma che hanno preso faccia una cosa (e magari la fa davvero), compromette anche la sicurezza del sistema.

Dovreste controllare quali programmi installate sulla vostra macchina. RedHat fornisce checksum MD5 e firme PGP sui suoi pacchetti RPM perché possiate verificare ciò che installate. Altre distribuzioni hanno metodi simili. Non dovreste mai eseguire binari che non conoscete, di cui non avete il sorgente, come root! Ovviamente pochi intrusori rilasciano il codice sorgente al pubblico.

Per quanto può essere complesso, prendete sempre i sorgenti di un programma dal suo vero sito di distribuzione. Se il programma deve essere eseguito da root, controllate i sorgenti o fateli controllare da qualcuno di cui vi fidate.

6 Sicurezza delle Password e Crittografia

Uno dei mezzi di sicurezza più importanti usati oggi sono le password. E' importante per voi e i vostri utenti avere password sicure e difficili da indovinare. Molte delle recenti distribuzioni Linux includono programmi `passwd` che non permettono di inserire password troppo facili. Assicuratevi che il vostro `passwd` sia aggiornato e abbia questa caratteristica.

Un'approfondita discussione sulla crittografia trascende gli scopi di questo documento, ma è necessaria almeno un'introduzione. La crittografia ùtilissima, forse persino necessaria, in questi giorni. Ci sono metodi crittografici di ogni genere, ognuno con la propria gamma di caratteristiche.

La maggior parte degli Unix (e Linux non fa eccezione) usa un algoritmo di crittografia a senso unico, chiamato DES (Standard di Crittografia di Dati), per crittografare le password. Questa password crittata è poi conservata in `/etc/passwd` o (più raramente) `/etc/shadow`. Quando tentate di entrare la password che scrivete è crittata ancora e confrontata con quella nel file che conserva le password. Se coincidono, potete entrare. Anche se il DES è un algoritmo di compressione a due vie (potete codificare e decodificare, un messaggio, con le giuste chiavi), la variante che Unix usa è a senso unico. Questo significa che non dovrebbe essere possibile invertire la crittografia per ottenere la password dal contenuto di `/etc/passwd` (o `/etc/shadow`).

Attacchi di forza bruta, come "Crack" o "John the Ripper" (vedi la Sezione 6.9 ()) possono spesso indovinare la password se la vostra non è abbastanza casuale. I moduli PAM (vedi sotto) vi permettono di usare una diversa routine crittografica per le vostre password (MD5 o simili). Potreste anche usare Crack a vostro vantaggio. Eseguite periodicamente Crack contro il vostro database di password, per trovare quelle insicure. Poi contattate l'utente in questione e fategli cambiare password.

Potete visitare *http://consult.cern.ch/writeup/security/security_3.html* per avere informazioni su come scegliere una buona password.

6.1 PGP e Crittografia a Chiave Pubblica

La crittografia a chiave pubblica, come quella usata per PGP, usa una chiave per crittare e una per decrittare. Al contrario, la crittografia tradizionale usa la stessa chiave per crittare e decrittare; questa chiave deve essere conosciuta da entrambe le parti, e quindi trasferita in qualche modo da una all'altra.

Per evitare la necessità di trasmettere in modo sicuro la chiave, la crittografia a chiave pubblica usa due chiavi separate: una pubblica e una privata. La chiave pubblica di ognuno è disponibile a tutti per crittare , mentre ogni persona tiene la sua chiave privata per decrittare i messaggi crittati con la sua chiave pubblica.

Ci sono vantaggi per entrambi i metodi di crittografia, e potete leggere di queste differenze nella *FAQ della RSA sulla crittografia* `<http://www.rsa.com/rsalabs/newfaq/>`, citata alla fine di questa sezione.

PGP (Pretty Good Privacy) è ben supportata da Linux. Le versioni 2.6.2 e 5.0 funzionano bene. Per una buona introduzione a PGP e a come usarlo, date un'occhiata alla FAQ di PGP: *http://www.pgp.com/service/export/faq/55faq.cgi*

Siate sicuri di usare la versione per il vostro stato. A causa dei limiti di esportazione del governo USA, è proibito portare fuori dagli USA ogni forma di crittografia elettronica.

I controlli sull'esportazione dagli USA sono ora gestiti dall'EAR (Regole di Amministrazione dell'Esportazione). Non sono più gestite dall'ITAR.

C'è anche una guida passo-passo per configurare PGP su Linux presso *http://mercury.chem.pitt.edu/~angel/LinuxFocus/English/November1997/article7.html*. E' scritta per la versione internazionale di PGP, ma è facilmente adattabile alla versione USA. Potreste aver bisogno di una patch anche per alcune recenti versioni di Linux; la patch è disponibile presso *ftp://metalab.unc.edu/pub/Linux/apps/crypto*.

C'è un progetto che lavora ad una versione libera e open source di PGP. GnuPG è un sostituto completo e libero per PGP. Visto che non usa IDEA o RSA può essere usato senza restrizioni. GnuPG aderisce a *OpenPGP*. Visitate il sito di GNU Privacy Guard per avere più informazioni: *http://www.gnupg.org/*.

Ulteriori informazioni sulla crittografia si trovano nella FAQ della RSA sulla crittografia, disponibile presso *http://www.rsa.com/rsalabs/newfaq/*. Qui troverete informazioni su termini come "Diffie-Hellman", "crittografia a chiave pubblica", "certificati digitali", ecc.

6.2 SSL, S-HTTP, HTTPS e S/MIME

Spesso gli utenti chiedono informazioni sulle differenze fra i vari protocolli di sicurezza e crittografia, e su come usarli. Anche se questo non è un documento sulla crittografia, è una buona idea spiegare brevemente cosa sono i protocolli e dove trovare altre informazioni.

- **SSL:** - SSL, o Secure Sockets Layer (Layer di Socket Sicuri), è un metodo crittografico sviluppato da Netscape per dare sicurezza su Internet. Supporta diversi protocolli crittografici, e fornisce autenticazione del client e del server. SSL opera sul layer di trasporto, crea un canale di dati sicuro e crittato e può quindi crittare date di molti tipi. Ciò si nota soprattutto quando si visita un sito sicuro con Communicator ed è la base di tutte le comunicazioni sicure con Communicator, oltre che con altro software Netscape. Ulteriori informazioni si trovano presso *http://www.consensus.com/security/ssl-talk-faq.html*. Sono disponibili informazioni su altre forme di sicurezza di Netscape e un buon punto di partenza su questi protocolli presso *http://home.netscape.com/info/security-doc.html*.

- **S-HTTP:** - S-HTTP è un altro protocollo che fornisce servizi di sicurezza su Internet. E' stato progettato per dare privacy, autenticazione, integrità e per impedire scambi di persona, supportando contemporaneamente meccanismi di controllo con chiavi multiple e algoritmi crittografici attraverso la negoziazione di opzioni fra le parti coinvolte in ogni transazione. S-HTTP è limitato al software specifico che lo implementa, e critta ogni messaggio individualmente. (Dalla FAQ della RSA sulla crittografia, pagina 138)
- **S/MIME:** - S/MIME, o Secure Multipurpose Internet Mail Extension (Estensione Multiuso Sicura per la Posta via Internet), è uno standard crittografico usato per crittare posta elettronica e altri tipi di messaggi su Internet. E' uno standard aperto sviluppato da RSA, quindi probabilmente lo vedremo presto su Linux. Altre informazioni su S/MIME si trovano presso *http://home.netscape.com/assist/security/smime/overview.html*.

6.3 Implementazioni IPSEC per Linux

Oltre a CIPE, e altre forme di crittografia, ci sono anche molte altre implementazioni di IPSEC per Linux. IPSEC è un tentativo della IETF di creare comunicazioni crittograficamente sicure al livello della rete IP e di fornire autenticazione, integrità controllo di accesso e privacy. Informazioni su IPSEC si trovano presso *http://www.ietf.org/html.charters/ipsec-charter.html*. Troverete anche link ad altri protocolli, una mailing list di IPSEC e degli archivi.

L'implementazione Linux di x-kernel, che viene sviluppata all'Università dell'Arizona, usa una struttura basata su oggetti per implementare protocolli i rete detti x-kernel, e si trova presso *http://www.cs.arizona.edu/xkernel/hpcc-blue/linux.html*. Semplificando, l'x-kernel è un metodo di passare messaggi al livello del kernel, che rende più facile un'implementazione.

Un'altra implementazione libera di IPSEC è la IPSEC FreeS/WAN. Sul loro sito si legge:

> "Questi servizi vi permettono di creare tunnel sicuri attraverso reti inaffidabili. Ogni cosa che passa per la rete inaffidabile è crittata dalla macchina gateway IPSEC e decrittata dal gateway all'altro capo. Il risultato è una Virtual Private Network (Rete Privata Virtuale) o VPN. E' una rete che rimane privata nonostante includa macchine connesse attraverso l'insicura Internet."

E' disponibile per lo scaricamento presso *http://www.xs4all.nl/~freeswan/*, ed ha appena raggiunto la versione 1.0 al momento della scrittura.

Come per altre forme di crittografia, non è distribuita col kernel di default a causa di restrizioni di esportazione.

6.4 `ssh` (Shell Sicura) e `stelnet`

`ssh` e `stelnet` sono suite di programmi che vi permettono di fare login a sistemi remoti su una connessione crittata.

`openssh` è una suite di programmi usata come sostituto sicuro di `rlogin`, `rsh` e `rcp`. Usa una crittografia a chiave pubblica per crittare le comunicazioni fra due host, oltre che per autenticare gli utenti. Può essere usata per fare login in sicurezza su host remoti o copiare dati fra host, evitando intercettazioni (dirottamenti della sessione) e il DNS spoofing. Eseguirà compressione dei dati sulle connessioni e comunicazioni X11 sicure fra host.

Ci sono diverse implementazioni ssh al momento. Quella commerciale originale di Data Fellows si trova nella homepage di `ssh` presso *http://www.datafellows.com*.

L'eccellente implementazione Openssh è basata su una delle prime versioni del ssh di Data Fellows ed è stata completamente ricostruita per non includere alcuna parte proprietaria o brevettata. E' libera e sotto licenza BSD. Si trova presso: *http://www.openssh.com*.

Esiste anche un progetto open source per reimplementare ssh da zero chiamato "psst...". Per ulteriori informazioni visitate: *http://www.net.lut.ac.uk/psst/*

Potete anche usare `ssh` dalla vostra workstation Windows verso un server `ssh` Linux. Ci sono diversi client liberi per Windows, incluso quello presso *http://guardian.htu.tuwien.ac.at/therapy/ssh/* oltre ad una versione commerciale di DataFellows, presso *http://www.datafellows.com*.

SSLeay è una versione libera del protocollo di Netscape Secure Sockets Layer, Sviluppato da Eric Young. Include diverse applicazioni, come Secure telnet, un modulo per Apache, diversi database, oltre a molti algoritmi inclusi DES, IDEA e Blowfish.

Usando questa libreria è stato creato un sostituto sicuro per telnet che usa la crittografia per la connessione. A differenza di SSH, stelnet usa SSL, il protocollo di Netscape. potete trovare Secure telnet e Secure FTP iniziando con la FAQ di SSLeay, disponibile presso *http://www.psy.uq.oz.au/~ftp/Crypto/*.

SRP è un'altra implementazione sicura di telnet/ftp. Dalla loro pagina web:

> "Il progetto SRP sta sviluppando software sicuro per Internet per l'uso libero in tutto il mondo. Iniziando con una distribuzione del tutto sicura di telnet e FTP, speriamo di soppiantare i deboli sistemi di autenticazione con forti sostituti che non sacrifichino la facilità d'uso per la sicurezza. La sicurezza dovrebbe essere lo standard, non un'opzione!"

Per ulteriori informazioni visitate *http://srp.stanford.edu/srp*.

6.5 PAM - Pluggable Authentication Modules (Moduli Aggiuntivi di Autenticazione)

Le ultime versioni della distribuzione Red Hat Linux sono distribuiti con uno schema di autenticazione unificato detto "PAM". PAM permette di cambiare i vostri metodi e requisiti di autenticazione al volo, e di aggiornare di conseguenza tutti i metodi di autenticazione senza ricompilare alcun binario. La configurazione di PAM trascende gli scopi di questo documento, ma siate certi di dare un'occhiata al sito web dei PAM per altre informazioni. *http://www.kernel.org/pub/linux/libs/pam/index.html*.

Solo alcune delle cose che potete fare con PAM:

- Usare crittografia non-DES per le password. (Rendendole più difficili da decodificare con la forza bruta.)
- Impostare limiti alle risorse degli utenti perché non possano attuare attacchi di denial-of-service (numero di processi, quantità di memoria, ecc.)
- Abilitare le shadow passwords (vedi sotto) al volo
- Permettere a singoli utenti di accedere solo ad ore precise da posti precisi

Nel giro di poche ore dall'installazione e configurazione del vostro sistema, potete prevenire molti attacchi. Per esempio, usando PAM per impedire l'uso nel sistema di file `.rhosts` nella home directory degli utenti aggiungendo queste linee a `/etc/pam.d/rlogin`:

```
#
# disabilita rsh/rlogin/rexec per gli utenti
#
login auth required pam_rhosts_auth.so no_rhosts
```

6.6 Incapsulamento Crittografico IP (CIPE)

Lo scopo primario di questo software è di fornire un mezzo per avere interconnessione sicura (contro l'eavesdropping, inclusi l'analisi del traffico e l'inserimento di falsi messaggi)fra le sottoreti attraverso una rete a pacchetti inaffidabile come Internet.

CIPE critta i dati al livello di rete. I pacchetti che viaggiano fra gli host di rete sono crittati. Il motore crittografico è affiancato al driver che manda e riceve pacchetti.

In questo si differenzia da SSH, che critta i dati per connessione, a livello dei socket. Viene crittata una connessione logica fra programmi eseguiti su host differenti.

CIPE può essere usato per il tunnelling, al fine di creare una Virtual Private Network. La crittografia a basso livello ha il vantaggio di poter essere usata fra le due reti connesse con la VPN, senza cambiamento per il software applicativo.

Riassumendo la documentazione CIPE:

> Gli standard IPSEC definiscono un set di protocolli che possono essere usati (oltre alle altre cose) per costruire VPN crittate. Comunque, IPSEC è un protocollo piuttosto pesante e complicato da impostare con molte opzioni, le implementazioni del protocollo completo sono tuttora usate raramente e alcune caratteristiche (come la gestione delle chiavi) non sono ancora del tutto risolte. CIPE ha un approccio più semplice in cui molte cose che possono essere parametrizzate (come la scelta del algoritmo crittografico effettivamente usato) sono una scelta da fare all'installazione. Questo limita la flessibilità, ma permette una implementazione più semplice e quindi efficiente e semplice in quanto a debug.

Ulteriori informazioni possono essere trovate presso *http://www.inka.de/~bigred/devel/cipe.html*

Come per altre forme di crittografia, non è distribuita di default con il kernel a causa di restrizioni di esportazione.

6.7 Kerberos

Kerberos è un sistema di autenticazione sviluppato dal progetto Athena al MIT. Quando un utente fa un login, Kerberos lo autentica (usando una password), e dà all'utente un modo per provare la sua identità: ad altri server ed host sparsi per la rete.

Questa autenticazione è poi usata da programmi come `rlogin` per lasciar entrare l'utente in altri host senza una password (al posto del file `.rhosts`). Questo metodo di autenticazione può anche essere usato dal sistema di posta per garantire che la posta sia consegnata alla persona giusta, oltre che per garantire che il mittente sia chi dice di essere.

Kerberos e i programmi con esso distribuiti evitano che gli utenti possano ingannare il sistema facendogli credere di essere qualcun altro. Purtroppo installare Kerberos è molto intrusivo, visto che richiede la modifica o la sostituzione di molti programmi standard.

Troverete ulteriori informazioni su Kerberos leggendo *le FAQ di Kerberos*; inoltre troverete il codice presso *http://nii.isi.edu/info/kerberos/*.

[Da: Stein, Jennifer G., Clifford Neuman, e Jeffrey L. Schiller. "Kerberos: An Authentication Service for Open Network Systems." USENIX Conference Proceedings, Dallas, Texas, Winter 1998.]

Kerberos non dovrebbe essere il primo passo nel miglioramento della sicurezza del vostro host. Porta a molte conseguenze e non è usato quanto, per esempio, SSH.

6.8 Shadow Password.

Le shadow password sono un mezzo di tenere segrete agli utenti normali le password crittate. Le ultime versioni di RedHat e Debian Linux usano di default le shadow password, ma in altri sistemi le password crittate sono tenute in `/etc/passwd` dove tutti possono leggerle. Quindi chiunque potrebbe eseguire su di loro programmi che tentino di indovinare quali sono. Al contrario, le shadow password sono salvate in `/etc/shadow` che solo gli utenti privilegiati possono leggere. Per usare delle shadow password dovete assicurarvi che tutte le applicazioni che devono leggere le password siano ricompilate per supportarle. PAM (vedi sopra) invece permette di semplicemente installare un modulo shadow; non richiede la ricompilazione degli eseguibili. Potete fare riferimento allo Shadow-Password HOWTO per ulteriori informazioni se necessario. E' *http://metalab.unc.edu/LDP/HOWTO/Shadow-Password-HOWTO.html* E' abbastanza datato al momento, e non serve per le distribuzioni che supportino PAM.

6.9 "Crack" e "John the Ripper"

Se per qualche ragione il vostro programma `passwd` non forza l'uso di password difficili da indovinare potreste volere eseguire un programma per crackare le password per essere sicuri che le password dei vostri utenti siano sicure.

I programmi per crackare le password funzionano su un semplice principio: provano ogni parola nel dizionario, e quindi variazioni di quelle parole, crittandole tutte e comparandole alle vostre password crittate. Se trovano una corrispondenza sanno di aver trovato quella giusta.

Esistono moltissimi programmi di questo tipo... i due più degni di nota sono "Crack" e John the Ripper" (*http://www.false.com/security/john/index.html*). Usano moltissimo la cpu, ma, eseguendoli, capirete se possono trovare le vostre password e potrete quindi avvertire gli utenti con password deboli. Notate che un intrusore dovrebbero prima riuscire a trovare qualche altro buco per leggere il vostro `/etc/passwd`, ma questi buchi sono più comuni di quanto pensiate

Visto che la sicurezza è resistente quanto il più insicuro degli host, è bene ricordare che se avete macchine Windows sulla vostra rete dovreste controllare L0phtCrack, una versione Windows di Crack. E' disponibile presso *http://www.l0pht.com*

6.10 CFS - File System Crittografico e TCFS - File System Crittografico Trasparente

CFS è un modo per crittare interi alberi di directory e per agli utenti di salvare al loro interno file crittografati. Usa un server NFS eseguito sulla macchina locale. Gli RPM sono disponibili presso *http://www.zedz.net/redhat/*, e troverete altre informazioni su come funziona presso *ftp://ftp.research.att.com/dist/mab/*.

TCFS migliora CFS aggiungendo più integrazione con il file system. Ulteriori informazioni sono presso: *http://edu-gw.dia.unisa.it/tcfs/*.

Inoltre non serve che sia usato su interi file system. Funziona anche su alberi di directory.

6.11 X11, SVGA e Sicurezza della GUI

6.11.1 X11

E' importante rendere sicuro la vostra interfaccia grafica per evitare che un intrusore ottengano la vostra password mentre la scrivete, leggano i documenti o le informazioni che avete sullo schermo o persino che

usino un buco per avere l'accesso di root. Anche eseguire applicazioni X remote su una rete deve essere considerato come un pericolo, perché permette a chi intercetta i pacchetti di vedere tutte le interazioni fra voi e il sistema remoto.

X ha una serie di meccanismi di controllo degli accessi. Il più semplice di questi è basato sugli host: usate `xhost` per specificare gli host a cui è permesso l'accesso al vostro display. Questo metodo non è affatto sicuro, perché se qualcuno ha accesso alla vostra macchina, può aggiungere la *sua macchina*al vostro `xhost` ed ottenere facile accesso. Inoltre, se dovete dare l'accesso ad una macchina inaffidabile, chiunque sia lì può compromettere la vostra GUI.

Quando usate `xdm` (X Display Manager) per entrare, avete un metodo di accesso molto migliore: MIT-MAGIC-COOKIE-1. Un "cookie" a 128 bit viene creato nel vostro file `.Xauthority`. Se dovete lasciare che una macchina remota acceda al vostro display, potete usare il comando `xauth` e le informazioni nel vostro file `.Xauthority` per dare accesso solo a quella connessione. Leggete il Remote-X-Apps mini-howto, disponibile presso *http://metalab.unc.edu/LDP/HOWTO/mini/Remote-X-Apps.html*.

Potete anche usare `ssh` (cfr. 6.4 (), sopra) per permettere connessioni a X sicure. Ciò ha il vantaggio anche di essere trasparente per l'utente finale, e comporta che nessun dato non crittato passa per la rete.

Controllate la man page `Xsecurity` per avere più informazioni sulla sicurezza di X. La via sicura è usare `xdm` per entrare nella console e quindi usare `ssh` per eseguire programmi X su macchine remote.

6.11.2 SVGA

I programmi che usano SVGAlib sono in genere SUID-root allo scopo di accedere a tutto l'hardware della vostra macchina Linux. Questo li rende molto pericolosi. Se si bloccano, in genere avete bisogno di riavviare la macchina per riavere una console usabile. Assicuratevi che tutti i programmi SVGA che usate siano autentici e siano almeno abbastanza affidabili. Ancor meglio, non eseguitene affatto.

6.11.3 GGI (Generic Graphics Interface project) (Progetto di Interfaccia Grafica Generica)

Il progetto Linux GGI sta tentando di risolvere i diversi problemi delle interfacce grafiche di Linux. GGI sposterà una piccola parte del codice video nel kernel di Linux e quindi controllerà direttamente l'accesso al sistema video. Questo significa che GGI potrà ripristinare la vostra console in ogni momento. Inoltre permetteranno di usare una chiave sicura, per assicurarvi che non c'è alcun cavallo di Troia che manometta il `login` della vostra console. *http://synergy.caltech.edu/~ggi/*

7 Sicurezza nel Kernel

Questa è una descrizione delle opzioni del kernel che riguardano la sicurezza, una spiegazione di ciò che fanno e di come si usano.

Visto che il kernel controlla il networking del vostro computer, è importante che sia molto sicuro e non venga compromesso. Per evitare alcuni dei recenti attacchi via rete dovreste tentare di mantenere aggiornato il vostro kernel. Potete trovare nuovi kernel presso `<ftp://ftp.kernel.org>` o dal distributore del vostro sistema.

Esiste anche un gruppo internazionale che fornisce una singola patch crittografica per il kernel principale di Linux. Questa patch fornisce il supporto per una serie di sottosistemi crittografici e caratteristiche che non possono essere incluse nel kernel principale per limiti di esportazione. Per ulteriori informazioni visitate la loro pagina web presso: *http://www.kerneli.org*

7.1 2.0 Kernel Compile Options

Per i kernel 2.0.x, si applicano le seguenti opzioni. Dovreste trovare le seguenti opzioni durante il processo di configurazione del kernel. Molti dei commenti qui vengono da `./linux/Documentation/Configure.help`, che è lo stesso documento che leggete usando l'Help durante la fase `make config` della compilazione del kernel.

- Network Firewalls (CONFIG_FIREWALL) Questa opzione dovrebbe essere attivata se volete usare qualche tipo di firewall o masquerading sulla vostra macchina. Se sarà solo un semplice client, potete rispondere no.

- IP: forwarding/gatewaying (CONFIG_IP_FORWARD) Se abilitate L'IP forwarding, potrà assumere le funzioni di un router. Se la vostra macchina è su una rete, potreste inoltrare dati da una rete ad un'altra e forse potreste evitare un firewall che era stato messo lì proprio per evitarlo. Chi si connette con un modem lo disattiverà, e gli altri dovrebbero riflettere sulle implicazioni di sicurezza. I computer che fanno da firewall lo abiliteranno e lo useranno insieme con un firewall.

  Potete abilitare dinamicamente l'IP forwarding usando il seguente comando:

  ```
  root#  echo 1 > /proc/sys/net/ipv4/ip_forward
  ```

  e disabilitarlo con il comando:

  ```
  root#  echo 0 > /proc/sys/net/ipv4/ip_forward
  ```

  Ricordate che i file in /proc sono "virtuali" e le dimensioni mostrate dal file potrebbero non riflettere i dati che contiene.

- IP: syn cookies (CONFIG_SYN_COOKIES) Un "Attacco SYN" è un attacco di denial of service (DoS), o negazione di servizio, che consuma tutte le risorse della vostra macchina, obbligandola al riavvio. Non c'è una ragione per cui non dovreste abilitare questa opzione. Nei kernel 2.2.x questa opzione si limita a permettere i syn cookies, ma non li abilita. Per abilitarli, dovete digitare:

  ```
  root# echo 1 > /proc/sys/net/ipv4/tcp_syncookies <P>
  ```

- IP: Firewalling (CONFIG_IP_FIREWALL) Questa opzione è necessaria se volete usare la vostra macchina come firewall, usare il masquerading o se volete proteggere il vostro sistema da qualcuno che entri attraverso l'interfaccia di composizione PPP.

- IP: firewall packet logging (CONFIG_IP_FIREWALL_VERBOSE) Questa opzione vi dà informazioni (come origine, destinazione, porta ecc.) sui pacchetti ricevuti dal firewall.

- IP: Drop source routed frames (CONFIG_IP_NOSR) Questa opzione dovrebbe essere abilitata. Questi pacchetti contengono l'intero percorso verso o la loro destinazione all'interno. Questo significa che i router da cui passa il pacchetto non lo controllano, lo inoltrano solamente. Questo potrebbe portare nel vostro sistema dati che potrebbero essere un exploit.

- IP: masquerading (CONFIG_IP_MASQUERADE) Se uno dei computer sulla rete locale per cui il vostro server Linux fa da firewall volesse mandare fuori dei pacchetti, il vostro server potrebbe "recitare la parte" di questa macchina, cioè inoltrare il traffico verso la destinazione richiesta, ma facendolo sembrare proveniente dal firewall stesso. Controllate *http://www.indyramp.com/masq* se volete altre informazioni.

- IP: ICMP masquerading (CONFIG_IP_MASQUERADE_ICMP) Questa opzione aggiunge il masquerading ICMP alla precedente opzione che maschera solo il traffico TCP o UDP.

- IP: transparent proxy support (CONFIG_IP_TRANSPARENT_PROXY) Questa scelta abilita il vostro firewall Linux a ridirigere in modo trasparente tutto il traffico dalla vostra rete destinato ad un host remoto verso un server locale, detto "server proxy trasparente". Questo fa credere ai computer locali di parlare con l'host remoto, mentre sono connessi al proxy locale. Leggete l'IP-Masquerading HOWTO e *http://www.indyramp.com/masq* per ulteriori informazioni.
- IP: always defragment (CONFIG_IP_ALWAYS_DEFRAG) In genere questa opzione è disabilitata ma se state costruendo un firewall o un host di masquerading, la dovreste abilitare. Quando vengono mandati dei dati fra un host ed un altro, non sempre viene mandato come un singolo pacchetto di dati, ma piuttosto viene scomposto in vari pezzi. Il problema di questo processo è che i numeri delle porte sono scritti solo nel primo frammento. Questo significa che qualcuno potrebbe inserire nei rimanenti pacchetti informazioni che non dovrebbero esserci. Questa opzione protegge anche attacchi del genere contro una macchina interna che non abbia ancora questa protezione.
- Packet Signatures (CONFIG_NCPFS_PACKET_SIGNING) Questa opzione, disponibile nei kernel 2.2.x, firma i pacchetti NCP per avere più sicurezza. Normalmente potete lasciarla fuori, ma rimane nel caso servisse.
- IP: Firewall packet netlink device (CONFIG_IP_FIREWALL_NETLINK) Questa utile opzione vi permette di analizzare i primi 128 byte dei pacchetti con un programma, per determinare se accettare o respingere il pacchetto a seconda della sua validità.

7.2 2.2 Kernel Compile Options

Per i kernel 2.2.x molte opzioni sono rimaste uguali, ma ne sono state aggiunte di nuove. Molti dei commenti vengono da `./linux/Documentation/Configure.help`, Che è lo stesso documento che leggete quando usate l'Help nella fase `make config` della compilazione del kernel. Sono segnate qui solo le opzioni nuove. Consultate la descrizione del 2.0 per avere una lista di altre opzioni necessarie. Il cambiamento più significativo nei kernel 2.2.x è il codice del firewall IP. Ora viene usato il programma `ipchains` per istallare il firewall, invece del `ipfwadm` usato nei kernel 2.0.

- Socket Filtering (CONFIG_FILTER) Per la maggior parte delle persone, dire no è una scelta sicura. Questa opzione vi permette di connettere un filtro a qualsiasi socket e determinare se i pacchetti debbano essere accettati o respinti. A meno che non abbiate una necessità particolare e siate capaci di programmare un tale filtro, dovreste dire di no. Notate anche che, quando è stato scritto questo documento, erano supportati tutti i protocolli tranne il TCP.
- Port Forwarding Il Port Forwarding (Inoltro di una porta) è un'aggiunta all'IP Masquerading che permette di inoltrare pacchetti dall'esterno all'interno di un firewall su certe porte. Torna utile se, per esempio, volete eseguire un web server dietro un firewall e lasciare che quel server sia accessibile al mondo esterno. Un client esterno manda una richiesta alla porta 80 sul firewall, questo inoltra la richiesta al server web che risponde alla richiesta attraverso il firewall. Il client penserà che sia l'host del firewall ad eseguire il web server. Potete usare questa caratteristica anche per bilanciare il traffico se avete una serie di web server identici dietro al firewall.

  Informazioni su questa opzione si trovano presso: http://www.monmouth.demon.co.uk/ipsubs/portforwarding.html (per accedere al WWW dovete avere accesso ad una macchina su Internet che abbia un programma come lynx o netscape). Per informazioni generali leggete: ftp://ftp.compsoc.net/users/steve/ipportfw/linux21/
- Socket Filtering (CONFIG_FILTER) Usando questa opzione, certi programmi possono porre un filtro ad un socket qualsiasi e quindi dire al kernel di bloccare o far passare certi dati attraverso quel socket. Il filtraggio dei socket per ora funziona su tutti i tipi di socket, TCP escluso. Leggete il file `./linux/Documentation/networking/filter.txt` per ulteriori informazioni.

- IP: Masquerading Il masquerading del kernel 2.2 è stato migliorato. Fornisce supporto addizionale per special protocolli di masquerading ecc. Assicuratevi di leggere l'IP Chains HOWTO per ulteriori informazioni.

7.3 Device nel Kernel

Nel kernel ci sono alcuni device a blocchi e a caratteri che vi aiuteranno con la sicurezza.

I due device `/dev/random` e `/dev/urandom` sono forniti dal kernel per avere sempre a disposizione dati casuali.

Sia `/dev/random` che `/dev/urandom` dovrebbero essere abbastanza sicuri da essere usati per generare chiavi PGP, autenticazioni `ssh` ed altre applicazioni in cui servono numeri casuali sicuri. Un attaccante non dovrebbe essere capace di predire il numero seguente, data una qualsiasi sequenza di numeri da queste sorgenti. E' stato fatto un grande sforzo perché i numeri ottenuti da questi device siano casuali in ogni senso della parola.

L'unica differenza fra i due device è che `/dev/random` finisce i byte casuali e vi fa aspettare finché non ce ne sono di nuovi. Notate che su alcuni sistemi potrebbe bloccarsi per molto tempo in attesa che nuova entropia sia generata dagli utenti. Quindi riflettete prima di usare `/dev/random`. (Forse la miglior cosa da fare è usarlo quando generate dovete generare codice importante: dite agli utenti di scrivere molto sulle tastiere fino a quando dite "Ok, basta".)

`/dev/random` è entropia di qualità, generata dalla misura dei tempi fra gli interrupt ecc. Si blocca finché non ha abbastanza bit casuali di dati.

`/dev/urandom` è simile, ma quando la riserva di entropia sta finendo fornisce una replica crittograficamente forte di ciò che rimane. Non è altrettanto sicuro, ma basta per la maggior parte delle applicazioni.

Potete leggere da questi device con qualcosa del genere:

```
root#  head -c 6 /dev/urandom | mimencode
```

Questo comando visualizza otto caratteri casuali sulla console, adatti per la generazione di password. Potete trovare `mimencode` nel pacchetto `metamail`.

Leggete `/usr/src/linux/drivers/char/random.c` per la descrizione dell'algoritmo.

Grazie a Theodore Y. Ts'o, Jon Lewis, e altri sviluppatori del kernel per avere aiutato me (Dave) in questo.

8 Sicurezza di Rete

La sicurezza di rete diventa sempre più importante perché il tempo di connessione ad Internet è sempre maggiore. Compromettere la sicurezza di rete è spesso molto più facile che compromettere la sicurezza fisica o locale ed è anche molto più comune.

Esiste una serie di buoni strumenti per aiutare nella sicurezza di rete e molti nuovi vengono inclusi nelle distribuzioni Linux.

8.1 Sniffer di Pacchetti

Uno dei modi più comuni con cui gli intrusori entrano in molti sistemi della vostra rete è l'uso di uno sniffer di pacchetti su un host già compromesso. Questo "sniffer" aspetta sulla porta Ethernet dati come `passwd`, `login` e `su` nel flusso dei dati e quindi registra il traffico successivo. In questo modo chi lo usa ottiene le

password di sistemi a cui non hanno neppure tentato un attacco. Le password in chiaro sono molto vulnerabili a questo tipo di attacco.

Esempio: L'host A è stato compromesso. L'attaccante installa uno sniffer. Lo sniffer intercetta l'amministratore mentre fa un login dall'host B al C. Ottiene la password personale dell'amministratore. Poi, l'amministratore usa `su` per risolvere un problema. Ora si conosce anche la password di root per l'host B. Più tardi l'amministratore lascia che dal suo acconto qualcuno usi telnet per connettersi all'host Z. Ora l'attaccante ha il login e la password per l'host Z.

Oggi l'attaccante non ha neanche più bisogno di compromettere un sistema per usare questa tecnica: potrebbe semplicemente portare un portatile o un PC nell'edificio e connettersi alla vostra rete.

Usare `ssh` o altri metodi di crittografia delle password blocca questo attacco. Anche cose come APOP per gli account POP lo impediscono (I normali login POP sono molto vulnerabili a questo attacco, come ogni cosa che mandi password in chiaro su una rete).

8.2 Servizi di sistema e tcp_wrappers

Prima di mettere un sistema Linux su `QUALSIASI` rete la prima cosa da sapere è quali servizi volete offrire. I servizi che non volete devono essere disabilitati, così da avere una cosa in meno di cui preoccuparsi e un possibile buco in meno per un intrusore.

Ci sono molti modi per disabilitare i servizi sotto Linux. Potete leggere il vostro `/etc/inetd.conf` e vedere quali servizi vengono offerti dal vostro `inetd`. Potete disabilitare tutti quelli che non vi servono commentandoli (`#` all'inizio della linea), e quindi mandando al processo inetd un SIGHUP.

Potreste rimuovere (o commentare) servizi nel vostro `/etc/services`. Questo impedirà l'uso del servizio anche da client locali (cioè se rimuovete `ftp` e provate a connettervi con ftp ad un host remoto da quella macchina otterrete solo un messaggio di "servizio sconosciuto"). In genere il gioco non vale la candela, visto che non dà ulteriori garanzie. Se qualcuno volesse usare `ftp` anche se l'avete commentato, potrebbe farsi un proprio client che usi la porta FTP e funzionerebbe benone.

Alcuni servizi che dovreste lasciare disponibili sono:

- `ftp`
- `telnet` (o `ssh`)
- mail, come `pop-3` o `imap`
- `identd`

Se sapete che non vi serve un pacchetto potete anche cancellarlo del tutto. `rpm -e` *`nome_pacchetto`* nella distribuzione RedHat cancella un pacchetto intero. Nella Debian `dpkg -remove` fa lo stesso.

Inoltre, dovreste davvero evitare che le utility rsh/rlogin/rcp, inclusi login (usato da `rlogin`), shell (usato da `rcp`) ed exec (usato da `rsh`) siano eseguite da `/etc/inetd.conf`. Questi protocolli sono molto insicuri e sono stati l'origine di exploit in passato.

Dovreste controllare `/etc/rc.d/rc[0-9].d` (sulla Red Hat; `/etc/rc[0-9].d` sulla Debian), e vedere se vengono avviati server che non sono necessari. I file in quelle directory sono link simbolici ai file nella directory `/etc/rc.d/init.d` (sulla Red Hat; `/etc/init.d` sulla Debian). Rinominare i file nella directory `init.d` blocca tutti i link simbolici che puntano a quel file. Se volete solo disabilitare un servizio per un particolare runlevel, rinominate il giusto link simbolico sostituendo la `S` maiuscola con una minuscola, così:

```
root#  cd /etc/rc6.d
root#  mv S45dhcpd s45dhcpd
```

Se avete file rc `rc` in stile BSD, controllate in `/etc/rc*` se ci sono programmi che non servono.

Con molte distribuzioni Linux viene distribuito un tcp_wrapper che "incapsula" tutti i vostri servizi TCP. Un tcp_wrapper(`tcpd`) viene invocato da `inetd` al posto del vero server. Quindi `tcpd` controlla l'host che sta richiedendo il servizio ed esegue il server o nega l'accesso all'host.`tcpd` permette di bloccare l'accesso ai vostri servizi TCP. dovreste creare un file `/etc/hosts.allow` e aggiungervi solo gli host che devono avere accesso ai servizi della macchina.

Se avete una connessione telefonica casalinga dovreste negarli TUTTI. Inoltre `tcpd` segna nei log i tentativi falliti di accedere ai servizi, per avvertirvi se siete sotto attacco. Se aggiungete nuovi servizi dovreste configurarli per usare i tcp_wrapper se usano TCP. Per esempio, un normale utente telefonico può impedire ad altri di connettersi alla propria macchina, conservando l'accesso alla posta e ad Internet. Per farlo, potreste aggiungere le linee seguenti al vostro `/etc/hosts.allow`:

ALL: 127.

E ovviamente /etc/hosts.deny dovrebbe contenere:

ALL: ALL

che bloccherà connessioni esterne alla macchina, ma permetterà la connessione a server su Internet.

Ricordatevi che i tcp_wrapper proteggono solo i servizi eseguiti da `inetd`, e pochi altri. Ci potrebbero benissimo essere altri servizi in esecuzione sulla vostra macchina. Potete usare `netstat -ta` per avere una lista di tutti i servizi offerti dalla vostra macchina.

8.3 Verificare i vostri DNS

Tenere aggiornate le informazioni dei DNS su tutti gli host della vostra rete aiuta ad aumentare la sicurezza. Se un host non autorizzato si connette alla rete, potete riconoscerlo dalla sua mancata presenza sul DNS. Molti servizi possono essere configurati per negare l'accesso ad host che non sono elencati dal DNS.

8.4 `identd`

`identd` è un piccolo programma che in genere è eseguito dal vostro `inetd`. Tiene nota di quale utente sta usando quale servizio e quindi fa rapporto a chi lo richiede.

Molte persone non capiscono l'utilità di `identd` e quindi lo disabilitano o bloccano tutte le richieste provenienti dall'esterno. `identd` non è al servizio di siti remoti. Non c'è modo di saper se i dati che ricevete da un `identd` remoto siano corretti o no. Non c'è autenticazione nelle richieste ad `identd`.

Perché eseguirlo allora? Perché aiuta *voi*, ed è un'informazione in più quando indagate. Se il vostro `identd` è integro, sapete che rivela ad altri l'user-name o l'uid di chi usa servizi TCP. Se l'amministratore di una rete remota viene a dire che un certo utente ha provato ad intromettersi nella sua rete, potrete facilmente prendere provvedimenti contro l'utente. Se non aveste eseguito `identd`, avreste dovuto leggere log su log, capire chi era nel sistema alla tal ora e in generale sprecare molto tempo a trovare l'utente.

L'`identd` distribuito con molte distribuzioni si può configurare più di quanto molti pensino. Potete disabilitarlo per utenti specifici, (con un file `.noident`), potete avere i log di tutte le richieste a `identd` (E' raccomandato), potete persino far sì che identd risponda con l'uid di un utente o persino con NO-USER.

8.5 SATAN, ISS, e altri scanner di rete

C'è una serie di differenti pacchetti software che fanno scansioni di porte e servizi su macchine o reti intere. SATAN, ISS, SAINT e Nessus sono alcuni dei più conosciuti. Questo software si connette con la macchina

bersaglio (o tutte le macchine bersaglio su una rete) su tutte le porte possibili e tentano di capire che servizio è attivo. Basandosi su queste informazioni si capisce se la macchina è vulnerabile ad un particolare exploit.

SATAN (Security Administrator's Tool for Analyzing Networks) è un port scanner con un'interfaccia web. Può essere configurato per eseguire controlli leggeri, medi o pesanti su una macchina o una rete. E' una buona idea usare SATAN sulla vostra rete e sistemare i problemi che trova. Assicuratevi di avere la copia di SATAN da *metalab* `<http://metalab.unc.edu/pub/packages/security/Satan-for-Linux/>` o da un sito FTP o web affidabile. E' stata distribuita in rete una copia trojan di SATAN. *http://www.trouble.org/~zen/satan/satan.html*. Notate che SATAN non è stato aggiornato da molto tempo, ed altri strumenti potrebbero avere risultati migliori.

ISS (Internet Security Scanner) è un altro port scanner. E' più veloce di Satan, quindi potrebbe essere migliore per grandi reti. Comunque Satan in genere fornisce più informazioni.

Abacus è una suite di strumenti che fornisce sicurezza agli host e rilevamento delle intrusioni. Leggete la home page sul web per ulteriori informazioni. *http://www.psionic.com/abacus/*

SAINT è una versione aggiornata di SATAN. E' basata sul web ed molti più test aggiornati di SATAN. Potete saperne di più presso: *http://www.wwdsi.com/~saint*

Nessus è uno scanner di sicurezza libero. Ha una interfaccia grafica GTK per semplicità d'uso. Inoltre è basato su una struttura a plug-in per aggiungere nuovi test. Per ulteriori informazioni, date un'occhiata a: *http://www.nessus.org*

8.5.1 Rilevare Scansioni delle Porte

Esistono strumenti progettati per avvisarvi di scansioni da parte di SATAN, ISS ed altri software di scansione. Comunque se usate molto i tcp_wrapper e leggete spesso i vostri log dovreste notare certe scansioni. Anche al settaggio inferiore SATAN lascia tracce nei log di un sistema RedHat "di serie".

Esistono anche port scanner "silenziosi". Un pacchetto con il bit TCP ACK attivo (come si fa per le connessioni stabilite) probabilmente passerebbe un firewall che filtri i pacchetti. Il pacchetto RST in risposta da una porta che _*non aveva comunicazioni in corso*_ viene preso come una prova di vita su quella porta. Non penso che i tcp_wrapper lo rilevino.

8.6 `sendmail`, `qmail` e MTA (Agenti di Trasporto di Posta)

Uno dei servizi più importanti che potete fornire è il server di posta. Sfortunatamente, è anche uno dei più vulnerabili agli attacchi, a causa del numero di compiti che esegue e dei privilegi che di solito richiede.

Se state usando `sendmail` è molto importante tenerlo aggiornato.`sendmail` ha una lunga, lunga storia di exploit di sicurezza. Assicuratevi di eseguire sempre l'ultima versione da: *http://www.sendmail.org*.

Ricordatevi che sendmail non deve per forza essere in esecuzione per spedire posta. Se siete un utente casalingo potete disabilitare del tutto sendmail e usare semplicemente il vostro client di posta per spedire. Potreste anche togliere l'opzione "-bd" da file di avvio di sendmail, disabilitando l'arrivo di richieste di posta. In altre parole, potete eseguire sendmail dal vostro script di avvio usando il seguente comando:

```
# /usr/lib/sendmail -q15m
```

In questo modo sendmail riprocesserà ogni 15 minuti i messaggi nella coda di stampa che non erano stati precedentemente consegnati.

Molti amministratori non usano sendmail, e al suo posto scelgono uno degli altri MTA. Potreste passare a `qmail`.`qmail` è stato progettato da zero tenendo a mente la sicurezza. E' veloce, stabile e sicuro. Qmail si trova presso *http://www.qmail.org*

In diretta competizione con qmail si pone "postfix", scritto da Wietse Venema, l'autore di tcp_wrapper ed altri strumenti di sicurezza. Precedentemente intitolato vmailer, e sponsorizzato da IBM, anche questo è un MTA fatto da zero per la sicurezza. Troverete altre informazioni su postfix presso *http://www.postfix.org*

8.7 Attacchi di Denial of Service (Negazione di un Servizio)

Un "Denial of Service" (DoS) è un attacco con cui un aggressore tenta di rendere una risorsa troppo occupata per rispondere a richieste legittime o di negare a utenti legittimi l'accesso ad una macchina.

Questi attacchi sono molto aumentati negli ultimi anni. Sotto sono elencati alcuni dei più comuni o recenti. Notate però che ne nascono in continuazione, quindi questi sono solo esempi. Leggete le liste di sicurezza di Linux e le liste di bugtraq per informazioni aggiornate.

- **SYN Flooding** - Il SYN flooding ("Inondazione di SYN") è un attacco DoS di rete. Sfrutta un buco nel modo in cui sono create le connessioni TCP. Gli ultimi kernel di Linux (2.0.30 e seguenti) hanno diverse opzioni configurabili per evitare che attacchi del genere neghino l'accesso alle vostre macchine. Vedi 7 (Sicurezza del Kernel) a proposito delle opzioni adeguate.
- **Il Bug "F00F" nei Pentium** - Si è scoperto recentemente che una serie di codici assembly mandati ad un Pentium originale Intel riavvierebbero la macchina. Questo bug affligge tutte le macchine Pentium (non i cloni, i Pentium Pro o i PII), a prescindere dal sistema operativo. I kernel Linux 2.0.32 e successivi contengono un rimedio che impedisce alla macchina di bloccarsi. Il kernel 2.0.33 ha una versione migliore rispetto al 2.0.32. Se usate un Pentium, aggiornate subito il kernel!
- **Ping Flooding** - Il ping flooding ("Inondazione di ping") è un attacco DoS basato sulla forza bruta. L'aggressore manda un'inondazione di pacchetti ICMP alla vostra macchina. Se lo fa da un host con un'ampiezza di banda maggiore della vostra, la vostra macchina non potrà mandare niente sulla rete. Un variazione di questo attacco, chiamato "smurfing", manda ad un host pacchetti ICMP con il *vostro* IP, permettendo loro di attaccarvi in modo quasi anonimo. Potete trovare altre informazioni circa lo "smurf" presso *http://www.quadrunner.com/~chuegen/smurf.txt* Se venite attaccati in questo modo, usate uno strumento come `tcpdump` per determinare da dove provengono i pacchetti (o da dove sembrano venire), quindi contattate il vostro provider con queste informazioni. I ping flood possono essere fermati facilmente all'altezza del router o usando un firewall.
- **Ping della Morte** - Il Ping della Morte manda pacchetti ICMP ECHO REQUEST che sono troppo grandi per entrare nelle strutture dati del kernel che li dovrebbero contenere. Poiché mandare un solo grande ping (65,510 byte) causa il blocco o il crash di molti sistemi, questo problema fu subito soprannominato "Ping della Morte". Il fatto è stato risolto da tempo, e non è più preoccupante.
- **Teardrop / New Tear** - Uno dei più recenti exploit coinvolge un bug presente nel codice di frammentazione IP sulle piattaforme Linux e Windows. E' stato risolto nel kernel 2.0.33 e non richiede la selezione di alcuna opzione di compilazione. Linux non sembra essere vulnerabile al nuovo exploit "newtear".

Potete trovare il codice della maggior parte degli exploit e una più approfondita descrizione del loro funzionamento presso *http://www.rootshell.com* usando il loro motore di ricerca.

8.8 Sicurezza del NFS (Network File System).

NFS è un protocollo di condivisione di file molto diffuso. Permette a server che eseguano `nfsd` e `mountd` per "esportare" interi filesystem in altre macchine usando il supporto per il filesystem NFS compilato nel

kernel (o qualche altro client se non sono macchine Linux). `mountd` tiene traccia dei filesystem montati in `/etc/mtab`, e li mostra con `showmount`.

Molti siti usano NFS per fornire le home directory agli utenti di modo che abbiano i loro file da qualunque macchina si colleghino.

E' possibile avere un po' di sicurezza quando si esportano filesystem. Potete far mappare a `nfsd` l'utente root remoto (uid=0) sull'utente `nobody`, negando l'accesso totale ai filesystem esportati. Comunque, visto che i singoli utenti hanno accesso ai propri file (o almeno a quelli con la stessa uid), l'utente root remoto può fare un login o su con il loro account ed avere accesso totale ai loro file. Questo è solo un piccolo ostacolo per un aggressore che ha i privilegi per montare i filesystem remoti.

Se dovete usare NFS, assicuratevi di esportare solo verso quelle macchine che lo richiedono. Non esportate mai la vostra intera directory root; esportate solo il necessario.

Leggete il NFS HOWTO per ulteriori informazioni su NFS, disponibile presso *http://metalab.unc.edu/mdw/HOWTO/NFS-HOWTO.html*

8.9 NIS (Network Information Service) (già YP).

Network Information service (servizio di informazione di rete) è un modo di distribuire informazioni ad un gruppo di macchine. Il master NIS tiene le tabelle di informazioni e le converte in file mappa di NIS. Queste mappe vengono distribuite nella rete, permettendo ai client di ottenere informazioni di login, password, home directory e shell (tutte le informazioni che in genere stanno in un normale `/etc/passwd`). Questo permette agli utenti di cambiare la loro password una volta per tutte le macchine sul dominio NIS.

NIS non è affatto sicuro. Non ha mai voluto esserlo. Doveva solo essere comodo ed utile. Chiunque possa indovinare il nome del vostro dominio NIS (ovunque sia nella rete) può ottenere una copia del file delle password ed usare Crack o John the Ripper contro le password. Se dovete usare NIS siate consapevoli dei pericoli.

Esiste un sostituto molto sicuro di NIS, chiamato NIS+. Controllate il NIS HOWTO: *http://metalab.unc.edu/mdw/HOWTO/NIS-HOWTO.html*

8.10 Firewall

I Firewall sono un mezzo per controllare quali informazioni vengono lasciate entrare ed uscire dalla vostra rete locale. Tipicamente l'host firewall è connesso ad Internet e alla LAN locale, ed è l'unico accesso ad Internet dalla LAN; in questo modo controlla cosa entra ed esce.

Ci sono molti tipi di firewall e metodi di impostarli. Le macchine Linux sono ottimi firewall. Il codice del firewall può essere compilato all'interno dei kernel 2.0 e superiori. Gli strumenti utente `ipfwadm` per i kernel 2.0 e `ipchains` per i kernel 2.2, vi permettono di cambiare al volo i tipi di traffico di rete permessi.

I firewall sono un'utilissima ed importante tecnica per la sicurezza di rete. Comunque, non pensate mai che solo perché avete un firewall non sia necessaria la sicurezza delle macchine che copre. Sarebbe un errore fatale. Controllate l'ottimo `Firewall-HOWTO` presso l'archivio metalab per avere più informazioni sui firewall e Linux. *http://metalab.unc.edu/mdw/HOWTO/Firewall-HOWTO.html*

Si trovano altre informazioni anche nell'IP-Masquerade mini-howto: *http://metalab.unc.edu/mdw/HOWTO/mini/IP-Masquerade.html*

Altre informazioni su `ipfwadm` (lo strumento per cambiare impostazioni al firewall) si trovano presso: *http://www.xos.nl/linux/ipfwadm/*

Se non avete esperienza con i firewall e volete metterne su uno per un piano di sicurezza che non sia superficiale, leggere "Firewalls" di O'Reilly and Associates o qualche altro documento online sui firewall è obbligatorio. Troverete altre informazioni su *http://www.ora.com*. Il National Institute of Standards and Technology (Istituto Nazionale degli Standard e della Tecnologia) ha scritto un eccellente documento sui firewall. Anche se datato 1995, è sempre abbastanza buono. Potete trovarlo anche *http://csrc.nist.gov/nistpubs/800-10/main.html*.

- Il progetto Freefire – una lista di strumenti firewall liberi, disponibili presso *http://sites.inka.de/sites/lina/freefire-l/index_en.html*
- SunWorld Firewall Design – scritto dagli autori del libro della O'Reilly, dà una introduzione di base ai differenti tipi di firewall. E' disponibile presso *http://www.sunworld.com/swol-01-1996/swol-01-firewall.html*
- Mason - il costruttore automatico di firewall per Linux. E' uno script di firewall che impara cosa vi serve mentre lo fate. Altre informazioni presso: *http://www.pobox.com/~wstearns/mason/*

8.11 IP Chains - Firewall per Linux Kernel 2.2.x

IP Firewalling Chains è un aggiornamento dal codice di firewalling del kernel 2.0 al 2.2. Ha molte più caratteristiche delle versioni precedenti, inclusi:

- Manipolazioni dei pacchetti più flessibili.
- Accounting più complesso.
- Sono istantanei i semplici cambiamenti di impostazioni.
- I frammenti possono essere esplicitamente bloccati, rifiutati ecc.
- I pacchetti sospetti vengono segnati in un log.
- Può gestire protocolli non ICMP/TCP/UDP.

Se state usando `ipfwadm` con il vostro kernel 2.0, sono disponibili degli script per convertire il formato di `ipfwadm` nel formato che usa `ipchains`.

Assicuratevi di leggere l'IP Chains HOWTO per altre informazioni. E' disponibile presso *http://www.rustcorp.com/linux/ipchains/HOWTO.html*

8.12 VPN - Virtual Private Networks (Reti Private Virtuali)

Le VPN sono un modo per stabilire una rete "virtuale" su una rete esistente. Questa rete virtuale è spesso crittata e passa il traffico solo a e da entità conosciute. Le VPN sono spesso usate per connettere attraverso Internet qualcuno che lavora a casa a un rete interna di una compagnia.

Se eseguite un firewall/masquerading con Linux dovete passare pacchetti di MS PPTP (un prodotto VPN punto-a-punto della Microsoft), c'è una patch per il kernel di Linux fatta apposta. Leggete: *ip-masq-vpn*.

Esistono diverse soluzioni VPN per Linux:

- vpnd. Leggete *http://sunsite.auc.dk/vpnd/*.
- Free S/Wan, disponibile presso *http://www.xs4all.nl/~freeswan/*
- ssh può essere usato per costruire una VPN. Leggete il VPN mini-howto per altre informazioni.

- vps (virtual private server) presso *http://www.strongcrypto.com*.

Controllate anche la sezione su IPSEC per altre bibliografie e informazioni.

9 Preparazione della Sicurezza (prima di entrare in rete)

Bene: avete controllato il sistema, è sicuro per quanto possibile e siete pronti a metterlo in rete. Ci sono alcune cose che dovreste fare ora per prepararvi ad un'intrusione, in modo da poter mettere fuori gioco velocemente l'aggressore e tornare alla piena funzionalità.

9.1 Fate un backup completo della macchina.

La discussione dei metodi di backup va oltre gli scopi di questo documento, ma vanno spese alcune parole su backup e sicurezza:

Se avete meno di 650mb di dati da salvare su una partizione, un CD-R è un'ottima strada (perché difficile da manomettere e molto durevole). Nastri e altri media riscrivibili dovrebbero essere protetti dalla scrittura non appena il backup è completo e quindi verificati per evitare la manomissione. Assicuratevi di lasciare i backup in un'area sicura e off-line. Un buon backup vi darà un buon punto di riferimento da cui ripristinare il sistema.

9.2 Scegliere una Buona Tabella di Backup

Un ciclo di sei nastri è facile da mantenere. Prevede quattro nastri per la settimana, uno per i Venerdì pari e uno per quelli dispari. Eseguite un backup incrementale ogni giorno, e un backup completo sul nastro del Venerdì. Potreste fare un backup completo anche per dei particolari cambiamenti importanti.

9.3 Fate un Backup dei Vostri Database di RPM o Debian

Nel caso di un'intrusione, potete usare il vostro database di pacchetti come usereste `tripwire`, ma solo se siete sicuri che non è stato modificato. Dovreste copiare il database RPM in un floppy e tenerlo sempre off-line. Probabilmente la Debian ha qualcosa di simile.

I file `/var/lib/rpm/fileindex.rpm` e `/var/lib/rpm/packages.rpm` probabilmente non entreranno in un solo floppy, ma se compressi dovrebbero entrare in un floppy ciascuno.

Ora, se il vostro sistema viene compromesso potete usare il comando:

```
root#  rpm -Va
```

per verificare ogni file sul sistema. Leggete la pagina man di `rpm`, perché esistono alcune opzioni che possono essere usate per avere un output più: conciso. Ricordate che dovete essere sicuri che anche l'eseguibile di RPM non sia stato compromesso.

Questo significa che ogni volta che viene aggiunto al sistema un RPM, il database dovrà essere ri-archiviato. voi decidere i vantaggi contro gli svantaggi.

9.4 Tenete Nota dei Dati degli Account

E' molto importante che le informazioni che vengono da `syslog` non siano modificate. Rendendo leggibili e scrivibili file in `var/log` solo da poche persone è un buon inizio.

Assicuratevi di tenere d'occhio cosa viene scritto lì soprattutto in `auth`. La presenza di molti login falliti, per esempio, indica una tentata intrusione.

Dove cercare i log dipende dalla vostra distribuzione. In un sistema Linux conforme al "Linux Filesystem Standard", come RedHat, cercherete in `/var/log` e controllerete `messages`, `mail.log`, ed altri.

Potete capire dove il sistema tiene i log leggendo il file `/etc/syslog.conf`. Questo è il file che dice a `syslogd` (il demone dei log di sistema) dove tenere i messaggi dei log.

Potreste anche voler configurare lo script o il demone che alterna i log per fargli tenere a lungo i log per avere tutto il tempo di esaminarli. Date un'occhiata al pacchetto `logrotate` su distribuzioni RedHat recenti. Altre distribuzioni hanno probabilmente un processo simile.

Se i vostri log sono stati manomessi, cercate di capire quando è iniziata la manomissione e cosa sembra cambiato. Ci sono lunghi periodi di tempo che non hanno log? Controllare i nastri di backup in cerca di log intatti è un buon inizio.

Gli intrusori in genere modificano i log per coprire le loro tracce, ma dovrebbero comunque essere controllati. Potreste notare l'intrusore che cerca un accesso o sfrutta un programma per ottenere l'account di root. Potreste vedere voci nei log che l'intrusore non ha avuto il tempo di cambiare.

Dovreste anche assicurarvi di separare `auth` da gli altri log, inclusi i tentativi di cambiare utente con `su`, i tentativi di login, e altre informazioni sugli account.

Se possibile, configurate `syslog` per mandare una copia dei log ad un sistema sicuro. Questo eviterà che un intrusore cancelli le sue tracce insieme ai tentativi di login/su/ftp/ecc. Leggete la pagina man di `syslog.conf` e cercate l'opzione `@`.

Ci sono diversi altri programmi `syslogd` più avanzati. Leggete *http://www.core-sdi.com/ssyslog/* a proposito di Secure Syslog. Secure Syslog permette di crittare le voci del syslog per essere sicuri che nessuno le manometta.

Un altro `syslogd` con altre caratteristiche è *syslog-ng*. Vi permette molta flessibilità e critta i flussi remoti di syslog per evitare la manomissione.

In fine, i log sono inutili se nessuno li controlla. Prendetevi del tempo ogni tanto per leggere i vostri log e farvi un'idea di come devono essere in normali condizioni. Saperlo fa risaltare molte cose strane.

9.5 Applicate Tutti i Nuovi Aggiornamenti di Sistema.

Molti utenti Linux istallano da un CD-ROM. A causa della natura repentina degli aggiornamenti di sicurezza, vengono rilasciati continuamente nuovi programmi. Prima di connettere la vostra macchina alla rete è bene controllare sul sito ftp della vostra distribuzione e prendere tutti i pacchetti aggiornati da quando avete installato il CD-ROM. Spesso questi pacchetti contengono molte soluzioni di sicurezza, quindi è una buona idea installarli.

10 Cosa Fare Durante e Dopo un'Intrusione

Avete seguito alcuni di questi consigli e avete rilevato un'intrusione? La prima cosa da fare è restare calmi. Azioni affrettate possono causare più danni dell'aggressore stesso.

10.1 Compromissione della Sicurezza in Corso.

Rilevare una compromissione in corso è una situazione molto tesa. Come reagite avrà grandi conseguenze.

Se la compromissione è fisica probabilmente avete colto qualcuno che si è intrufolato in casa vostra, in ufficio o nel laboratorio. Dovreste avvisare le autorità competenti. In un laboratorio potreste aver visto qualcuno che provava ad aprire un case o riavviare una macchina. A seconda della vostra autorità e delle procedure, potreste farli desistere o chiamare direttamente la sicurezza.

Se avete preso un utente locale che tentava di compromettere la sicurezza la prima cosa da fare è confermare che sia davvero chi credete. Controllate il posto da cui si connettono. E' da dove si connettono normalmente? No? Allora usate un mezzo di comunicazione non elettronico. Per esempio, chiamatelo al telefono o andate nel suo ufficio/appartamento e parlateci. Se ammette il fatto, chiedetegli spiegazioni su quel che sta succedendo. Se non c'entra e non sa di cosa stiate parlando, probabilmente dovrete investigare ulteriormente. Investigate e procuratevi molte informazioni prima di accusare qualcuno.

Se avete rilevato un attacco di rete la prima cosa da fare (se possibile) è disconnettere la rete. Se sono connessi via modem, staccate il doppino; se via Ethernet, staccate il cavo Ethernt. Questo impedirà loro di fare altri danni e peneranno ad un problema di rete piuttosto che ad un rilevamento.

Se non potete staccare la rete (se avete un sito trafficato o non avete il controllo fisico sulle vostre macchine), il prossimo miglior passo è usare qualcosa come i `tcp_wrapper` o `ipfwadm` per negare l'accesso dal sito dell'intrusore.

Se non potete rifiutare tutte le connessioni dal sito dell'intrusore, vi dovete accontentare di disattivare l'account dell'utente. Farlo però non è semplice. Dovete tenere a mente i file `.rhosts`, l'accesso FTP, e una montagna di possibili backdoor.

Una volta preso uno dei provvedimenti sopra (disconnesso la rete, negato l'accesso dal suo sito e o disabilitato il suo account), dovete uccidere tutti i suoi processi ed espellerli.

Dovete controllare bene il vostro sito nei minuti seguenti, perché l'aggressore tenterà di rientrare. Forse usando un account differente, o da un altro indirizzo di rete.

10.2 La sicurezza è già stata compromessa.

Avete trovato una manomissione già avvenuta o avete rilevato e chiuso fuori (si spera) l'intrusore. E ora?

10.2.1 Chiudere il Buco

Se siete capaci di determinare in che modo l'aggressore è entrato, dovreste tentare di chiudere quel buco. Per esempio, forse troverete diverse voci FTP subito prima del login dell'utente. Disabilitate il servizio FTP e controllate se esiste una versione aggiornata o se qualche lista conosce un rimedio.

Controllate tutti i log, e visitate le liste di sicurezza e controllate se ci sono exploit che potete riparare. Trovate gli aggiornamenti di sicurezza di Caldera presso *http://www.caldera.com/tech-ref/security/*. Red Hat non ha ancora separato i propri aggiornamenti di sicurezza da quelli dei bug, ma l'errata corrige della loro distribuzione si trova presso *http://www.redhat.com/errata*

Debian ha ora una mailing list sulla sicurezza e una pagina web. Visitate *http://www.debian.org/security/* per altre informazioni.

E' molto probabile che se un distributore Linux ha rilasciato un aggiornamento lo abbiano fatto anche gli altri.

Esiste un progetto di sondaggio della sicurezza di Linux. Passano metodicamente tutti programmi utente e cercano exploit e overflow. Dal loro annuncio:

> "Stiamo tentando un sondaggio sistematico dei sorgenti di Linux con l'obiettivo di renderlo sicuro come OpenBSD. Abbiamo già scoperto (e riparato) alcuni problemi, ma altro aiuto è

> benvenuto. La lista non è moderata e rappresenta anche una buona fonte di discussioni generiche sulla sicurezza. L'indirizzo della lista è security-audit@ferret.lmh.ox.ac.uk Per iscriversi, mandate una e-mail a: security-audit-subscribe@ferret.lmh.ox.ac.uk"

Se non chiudete fuori l'aggressore, probabilmente tornerà. Non solo sulla vostra macchina, ma forse da qualche parte nella vostra rete. Se avesse eseguito uno sniffer, ci sono buone probabilità che abbia accesso ad altre macchine locali.

10.2.2 Stimare il Danno

La prima cosa è stimare il danno. Cosa è stato compromesso? Se stavate eseguendo un test di integrità come `Tripwire`, potete usarlo per eseguire un controllo; dovrebbe aiutare a capire cosa è stato compromesso. Altrimenti dovrete controllare tutti i dati importanti.

Visto che i sistemi Linux diventano sempre più semplici da istallare potreste considerare di salvare i file di configurazione, vuotare i dischi, reinstallare e quindi ripristinare i file degli utenti e di configurazione dai backup. Questo vi assicura un sistema nuovo e pulito. Se doveste ripristinare dei file dal sistema compromesso siate molto cauti con ogni binario che ripristinate, perché potrebbe essere un cavallo di Troia messo lì dall'intrusore.

La reinstallazione dovrebbe essere considerata obbligatoria se un intrusore ottenesse l'accesso di root. Inoltre, potreste voler tenere le prove lasciate, quindi avere un disco vuoto a disposizione può essere sensato.

Ora dovete preoccuparvi del tempo passato dalla compromissione e se i backup possono contenere lavoro danneggiato. Segue un approfondimento sui backup.

10.2.3 Backup, Backup, Backup!

Avere backup regolari è una manna per i problemi di sicurezza. Se il sistema venisse compromesso, potreste ripristinare i dati dai backup. Ovvio che alcuni dati hanno valore anche per l'aggressore e non solo li distruggerà ma li ruberà e si farà le sue copie; ma per lo meno li avrete anche voi.

Dovreste controllare backup anche molto vecchi, prima di ripristinare un file che è stato manomesso. L'intrusore potrebbe aver compromesso i file molto tempo fa e potreste aver fatto molti backup del file già compromesso!

Ovviamente, c'è una serie di problemi di sicurezza con i backup. Assicuratevi di tenerli in un posto sicuro. Sappiate chi vi ha accesso. (Se l'aggressore ottiene i vostri backup, ha accesso a tutti i vostri dati senza che neanche lo sappiate.)

10.2.4 Rintracciare l'Intrusore.

Bene, avete chiuso fuori l'intrusore e ripristinato il sistema, ma non avete ancora finito. Anche se è improbabile che l'intrusore venga preso, dovreste notificare l'attacco.

Dovreste notificarlo all'amministratore del sito da cui proveniva l'attacco. Potete trovare questo contatto con `whois` o sul database dell'Internic. Potreste mandare una e-mail con tutte le voci, le date e le ore dei log inerenti. Se aveste notato qualcosa di particolare nell'intrusore dovreste menzionarlo. Dopo aver mandato l'e-mail dovreste, se volete, far seguire una telefonata. Se l'amministratore a sua volta vedesse l'aggressore potrebbe parlare con l'amministratore del sito da cui proviene e via così.

I bravi cracker usano spesso diversi sistemi intermedi, alcuni (o molti) dei quali non sanno neanche di essere stati compromessi. Provare a inseguire un cracker fino al suo sistema base sarà difficile. Essere gentili con gli amministratori con cui parlate aiuta molto ad avere la loro collaborazione.

Dovreste anche dare notizia anche ad ogni organizzazione di sicurezza di cui fate parte (il *CERT* <http://www.cert.org/> o simili), oltre al distributore del vostro sistema Linux.

11 Documenti sulla Sicurezza

Esistono TANTISSIMI buoni siti per la sicurezza Unix in generale e quella Linux specificamente. E' molto importante iscriversi ad una (o più) delle mailing list sulla sicurezza per conoscere gli aggiornamenti. Molte di queste hanno pochissimo traffico e sono molto edificanti.

11.1 Siti FTP

Il CERT è il Computer Emergency Response Team (Squadra di Reazione alle Emergenze Informatiche). Spesso diffondono allarmi di nuovi attacchi e rimedi. Visitate *ftp://ftp.cert.org* per altre informazioni.

ZEDZ (già Replay) (*http://www.zedz.net*) ha archivi di molti programmi di sicurezza. Visto che sono fuori dagli USA non devono sottostare alle restrizioni crittografiche.

Matt Blaze è l'autore del CFS ed un grande pioniere della sicurezza. L'archivio di Matt è disponibile presso *ftp://ftp.research.att.com/pub/mab* <ftp://ftp.research.att.com/pub/mab>

`tue.nl` è un grande sito FTP Olandese sulla sicurezza. *ftp.win.tue.nl*

11.2 Siti Web

- The Hacker FAQ è una FAQ sugli hacker: *The Hacker FAQ*
- L'archivio COAST ha un gran numero di programmi e informazioni per la sicurezza di Unix: *COAST*
- Pagina della Sicurezza di SuSE: *http://www.suse.de/security/*
- Rootshell.com è un grande sito per capire quali exploit sono usati al momento dai cracker: *http://www.rootshell.com/*
- BUGTRAQ diffonde avvisi su questioni di sicurezza: *archivi di BUGTRAQ*
- Il CERT, Computer Emergency Response Team, diffonde avvisi su comuni attacchi a piattaforme Unix: *homepage del CERT*
- Dan Farmer è l'autore di SATAN e di molti altri strumenti di sicurezza. Il suo sito ha alcune interessanti informazioni oltre che strumenti di sicurezza: *http://www.trouble.org*
- Il Linux security WWW è un buon sito sulla sicurezza di Linux: *Linux Security WWW*
- Infilsec ha un motore di ricerca di vulnerabilità che sa dirvi quali punti deboli ha una specifica piattaforma: *http://www.infilsec.com/vulnerabilities/*
- Il CIAC diffonde bollettini di sicurezza periodici su exploit comuni: *http://ciac.llnl.gov/cgi-bin/index/bulletins*
- Un buon punto d'inizio per i Linux Pluggable Authentication Modules si trova presso *http://www.kernel.org/pub/linux/libs/pam/*.
- Il progetto Debian ha una pagina web per i propri fix di sicurezza ed altre informazioni. E' presso *http://www.debian.com/security/*.
- La WWW Security FAQ, scritta da Lincoln Stein, è un'ottima guida alla sicurezza del web. La trovate presso *http://www.w3.org/Security/Faq/www-security-faq.html*

11.3 Mailing List

Bugtraq: Per iscrivervi a bugtraq, mandate una mail a listserv@netspace.org contenente nel corpo del messaggio subscribe bugtraq. (Vedete gli archivi sopra).

CIAC: mandate una e-mail a majordomo@tholia.llnl.gov. Nel CORPO (non nel subject) del messaggio mettete:

```
subscribe ciac-bulletin
```

La Red Hat ha una serie di mailing list, la più importante delle quali è la lista redhat-announce. Potete leggervi di patch di sicurezza (e altro) non appena sono disponibili. Mandate una e-mail a redhat-announce-list-request@redhat.com con il subject

```
Subscribe
```

Visitate *http://www.redhat.com/mailing-lists/redhat-announce-list/* per altri informazioni e archivi.

Il progetto Debian ha una mailing list di sicurezza che tratta dei loro fix. Visitate *http://www.debian.com/security/* per altre informazioni.

11.4 Libri - Materiale Stampato

Esiste una serie di buoni libri di sicurezza. Questa sezione ne elenca alcuni. Oltre che nei libri sulla sicurezza, si parla di quest'argomento in molti altri libri di amministrazione di sistema.

Building Internet Firewalls di D. Brent Chapman & Elizabeth D. Zwicky

1ma Edizione Settembre 1995

ISBN: 1-56592-124-0

Practical UNIX & Internet Security, 2nd Edition di Simson Garfinkel & Gene Spafford

2nda Edizione Aprile 1996

ISBN: 1-56592-148-8

Computer Security Basics di Deborah Russell & G.T. Gangemi, Sr.

1ma Edizione Luglio 1991

ISBN: 0-937175-71-4

Linux Network Administrator's Guide di Olaf Kirch

1ma Edizione Gennaio 1995

ISBN: 1-56592-087-2

PGP: Pretty Good Privacy di Simson Garfinkel

1ma Edizione Dicembre 1994

ISBN: 1-56592-098-8

Computer Crime A Crimefighter's Handbook di David Icove, Karl Seger & William VonStorch (Consulting Editor Eugene H. Spafford)

1ma Edizione Agosto 1995

ISBN: 1-56592-086-4

Linux Security di John S. Flowers

New Riders;

ISBN: 0735700354

Marzo 1999

Maximum Linux Security : A Hacker's Guide to Protecting Your Linux Server and Network

Anonimo

Edizione economica - 829 pagine

Sams;

ISBN: 0672313413

Luglio 1999

Intrusion Detection di Terry Escamilla

Ed. Economica

416 pagine (Settembre 1998)

John Wiley and Sons;

ISBN: 0471290009

Fighting Computer Crime

Donn Parker

Ed. Economica

526 pagine (Settembre 1998)

John Wiley and Sons;

ISBN: 0471163783

12 Glossario

- **autenticazione:** Il processo che verifica che i dati ricevuti siano uguali a quelli spediti e che il mittente dichiarato sia quello effettivo.
- **Host bastione:** Un sistema che deve essere estremamente sicuro perché vulnerabile ad attacchi, in genere a causa del collegamento ad Internet e del continuo contatto con gli utenti della rete interna. Prende il nome dalle strutture fortificate sulle mura dei castelli medioevali. I bastioni sorvegliavano aree difensive strategiche ed in genere avevano grandi mura, spazio per truppe supplementari e il solito, caro, vecchio pentolone di olio bollente per "scoraggiare" gli aggressori.
- **buffer overflow:** spesso i programmatori non allocano mai buffer abbastanza grandi e si dimenticano di controllare se ci sono overflow. Quando un buffer va in overflow, al programma eseguito (sia un demone o un programma set-uid) si potrebbe far fare quel che non deve. In genere viene sovrascritto l'indirizzo di ritorno di una funzione sullo stack in modo che punti ad un'altra locazione . <
- **denial of service (negazione di un servizio):** un attacco che consuma le risorse del computer in cose inutili, bloccando così il loro uso per scopi legittimi.
- **Host dual-homed (a doppia base):** Un computer per uso generico che ha almeno due interfacce di rete.
- **firewall:** Un componente o set di componenti che restringe il passaggio di dati fra la rete protetta ed Internet, o fra due reti qualsiasi.

- **host:** un sistema che faccia parte di una rete.
- **IP spoofing:** l'IP Spoofing è un complesso attacco tecnico fatto di diversi componenti. E' un exploit di sicurezza che agisce facendo credere ai computer in una relazione di fiducia di essere qualcuno che in verità non è. Esiste un esaustivo documento scritto da daemon9, route e infinity nel Volume Settimo, edizione 48 di Phrack Magazine.
- **non-ripudiabilità:** la capacità di un destinatario di provare che il mittente di alcuni dati li ha veramente spediti, anche se in seguito ha negato di averlo fatto.
- **pacchetto:** l'unità fondamentale di comunicazione su Internet.
- **filtraggio dei pacchetti:** l'azione intrapresa da un device per controllare selettivamente il flusso di dati per e da una rete. I filtri lasciano passare o bloccano i pacchetti, spesso mentre li ridirigono da una rete ad un'altra (ancora più spesso da Internet ad una intranet e viceversa). I pacchetti vengono filtrati secondo regole che specificano che tipo di pacchetti sono permessi e quali sono bloccati.
- **rete di perimetro:** una rete inserita fra una rete protetta e una esterna, per aggiungere un ulteriore strato di sicurezza. Una rete di perimetro viene detta alle volte DMZ.
- **server proxy:** un programma che dialoga con server esterni al posto dei client interni. I client proxy parlano ai server proxy, che passano le richieste dei client approvate ai veri server, e consegnano le risposte ai client..
- **superuser:** nome informale per `root`.

13 Domande Frequenti

1. E' più sicuro compilare il supporto dei driver direttamente nel kernel, invece che farne un modulo? Risposta: Alcune persone ritengono meglio disabilitare il supporto dei moduli, perché un intrusore potrebbe caricare un Cavallo di Troia o un altro modulo pericoloso per la sicurezza.

   Comunque, per caricare i moduli, dovreste essere root. Anche i file dei moduli sono scrivibili solo da root. Questo significa che l'intrusore avrebbe bisogno dell'accesso di root per inserire un modulo. Se l'intrusore ha accesso di root, ci sono problemi ben più gravi della possibilità che carichi un modulo.

   I moduli servono per caricare dinamicamente il supporto per un device che viene usato raramente. Su macchine server, o firewall per esempio, è molto difficile che succeda. Per questa ragione, avrebbe più senso compilare il supporto direttamente nel kernel, nelle macchine che facciano da server. I moduli inoltre sono meno veloci del supporto compilato nel kernel.

2. Perché fare un login remoto come root ha sempre esito negativo? Risposta: Vedi 4.2 (Sicurezza di Root). Questo fatto è intenzionale per evitare che utenti remoti tentino di connettersi via `telnet` come `root` sulla vostra macchina, che è un serio rischio, perché la password di root sarebbe trasmessa in chiaro sulla rete. Non dimenticate: i potenziali intrusori hanno il tempo dalla loro e possono eseguire programmi che trovino la vostra password.

3. Come abilito le shadow password sulla mia Red Hat 4.2 or 5.x ? Risposta:

   Per abilitare le shadow password, eseguite `pwconv` da root, quindi dovrebbe essere creato ed usato dalle applicazioni `/etc/shadow`. Se state usando RH 4.2 o superiore, i moduli PAM si adatteranno automaticamente al cambiamento dal normale `/etc/passwd` alle shadow password senza altri cambiamenti.

   Un po' di informazioni di base: le shadow password sono un meccanismo per tenere le password in un file diverso dal solito `/etc/passwd`. Questo ha diversi vantaggi. Il primo è che il file ombra, `/etc/shadow`, è leggibile solo da root, a differenza di `/etc/passwd`, che deve essere leggibile a tutti.

L'altro vantaggio è che come amministratore potete abilitare o disabilitare gli account senza che uno sappia dell'altro.

Quindi il file `/etc/passwd` viene usato per registrare i nomi degli utenti e dei gruppi, usati da programmi come `/bin/ls` per risalire dal ID utente al nome utente nei listati delle directory.

Il file `/etc/shadow` contiene solo il nome utente e la password, magari informazioni sull'account come la scadenza ecc.

Visto che volete rendere più sicure le vostre password, forse sareste anche interessati a generare buone password. Per questo potete usare il modulo `pam_cracklib`, che fa parte dei PAM. Prova le vostre password contro le librerie di Crack per aiutarvi a decidere se sono troppo facili da trovare con programmi del genere.

4. Come posso abilitare le estensioni SSL di Apache? Risposta:

   (a) Prendete SSLeay 0.8.0 o successivo da `<ftp://ftp.psy.uq.oz.au/pub/Crypto/SSL>`

   (b) Compilatelo, provatelo e installatelo!

   (c) Prendete i sorgenti di Apache 1.2.5

   (d) Prendete le estensioni SSLeay di Apache da *qui* `<ftp://ftp.ox.ac.uk/pub/crypto/SSL/apache_1.2.5+ssl_`

   (e) Scompattatele nella directory dei sorgenti di apache-1.2.5 e aggiornate Apache seguendo il README

   (f) Configuratelo e compilatelo.

   Potreste anche provare su *ZEDZ net* che ha molti pacchetti precompilati ed è collocata fuori dagli USA.

5. Come posso manipolare gli account, mantenendo la sicurezza? Risposta: la distribuzione Red Hat, soprattutto RH5.0, contiene un gran numero di strumenti per cambiare le proprietà degli account utente.

   - `pwconv` e `unpwconv` possono essere usati per convertire fra password normali e shadow.
   - `pwck` e `grpck` possono essere usati per controllare la giusta organizzazione dei file `passwd` e `group`.
   - `useradd`, `usermod` e `userdel` si possono usare per aggiungere, togliere o modificare gli account. `groupadd`, `groupmod`, e `groupdel` fanno lo stesso per i gruppi.
   - Le password di gruppo possono essere create usando `gpasswd`.

   Tutti questi programmi riconoscono le shadow password – quindi, se le attivate useranno `/etc/shadow` per informazioni sulle password, altrimenti non lo faranno.

   Leggete le rispettive pagine man per ulteriori informazioni.

6. Come posso proteggere con una password specifici documenti HTML usando Apache? Scommetto che non sapevate di *http://www.apacheweek.org*, sbaglio?

   Potete trovare informazioni sull'autenticazione degli utenti presso *http://www.apacheweek.com/features/userauth* oltre ad altri trucchi a proposito di sicurezza dei server web su *http://www.apache.org/docs/misc/security_tips.html*

14 Conclusioni

Inscrivendovi alle mailing list di sicurezza e tenendovi aggiornati, potete far molto per la sicurezza della vostra macchina. Se fate attenzione ai vostri log ed eseguite qualcosa tipo `tripwire` regolarmente, potete fare anche di meglio.

Non è difficile mantenere un livello ragionevole di sicurezza su una macchina casalinga. Serve un maggiore impegno per le macchine di lavoro, ma Linux può essere una piattaforma senz'altro sicura. Grazie alla natura dello sviluppo di Linux, i fix alla sicurezza vengono spesso diffusi molto prima che per i sistemi operativi commerciali, rendendo Linux una piattaforma ideale quando la sicurezza è una necessità.

15 Ringraziamenti

Queste informazioni sono state raccolte da molte fonti. Grazie alle seguenti persone, che hanno contribuito direttamente o indirettamente:

Rob Riggs *rob@DevilsThumb.com* S. Coffin *scoffin@netcom.com*

Viktor Przebinda *viktor@CRYSTAL.MATH.ou.edu*

Roelof Osinga *roelof@eboa.com*

Kyle Hasselbacher *kyle@carefree.quux.soltc.net*

David S. Jackson *dsj@dsj.net*

Todd G. Ruskell *ruskell@boulder.nist.gov*

Rogier Wolff *R.E.Wolff@BitWizard.nl*

Antonomasia *ant@notatla.demon.co.uk*

Nic Bellamy *sky@wibble.net*

Eric Hanchrow *offby1@blarg.net*

Robert J. Bergerr*berger@ibd.com*

Ulrich Alpers *lurchi@cdrom.uni-stuttgart.de*

David Noha *dave@c-c-s.com*

Pavel Epifanov. *epv@ibm.net*

Joe Germuska. *joe@germuska.com*

Franklin S. Werren *fswerren@bagpipes.net*

Paul Rusty Russell *<Paul.Russell@rustcorp.com.au>*

Christine Gaunt *<cgaunt@umich.edu>*

lin *bhewitt@refmntutl01.afsc.noaa.gov*

A. Steinmetz *astmail@yahoo.com*

Jun Morimoto *morimoto@xantia.citroen.org*

Xiaotian Sun *sunx@newton.me.berkeley.edu*

Eric Hanchrow *offby1@blarg.net*

Queste persone hanno tradotto questo HOWTO in molte altre lingue!

Un grazie speciale a tutti loro per averci aiutato a diffondere la buona novella di Linux...

Polacco: Ziemek Borowski *ziembor@FAQ-bot.ZiemBor.Waw.PL*

Giapponese: FUJIWARA Teruyoshi *fjwr@mtj.biglobe.ne.jp*

Indonesiano: Tedi Heriyanto *22941219@students.ukdw.ac.id*

Coreano: Bume Chang *Boxcar0001@aol.com*

Spagnolo: Juan Carlos Fernandez *piwiman@visionnetware.com*

Olandese: R. Ekkebus *reggy@zeelandnet.nl*

Italiano: Matteo Scordino del SienaLUG *slug-list@siena.linux.it*