Linux Security HOWTO

Kevin Fenzi

tummy.com, ltd.

kevin-securityhowto@tummy.com

Dave Wreski

linuxsecurity.com

dave@linuxsecurity.com

Questo documento è una visione d'insieme dei problemi di sicurezza che un amministratore di sistemi Linux si trova ad affrontare. Tratta una filosofia di sicurezza generale e una serie di esempi specifici di come mettere meglio al sicuro da intrusi un sistema Linux. Sono anche inclusi riferimenti a materiale e programmi riguardanti la sicurezza. Miglioramenti, critiche costruttive, aggiunte e correzioni saranno ben accette. Per favore inviate i vostri commenti ad entrambi gli autori, con il subject "Security HOWTO". Traduzione a cura di Elisabetta Galli lab at kkk dot it, revisione a cura di Michele Ferritto m dot ferritto at virgilio dot it.

Sommario

1. Introduzione

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

Diversi altri HOWTO si sovrappongono a questo per le questioni di sicurezza e, quando necessario, sono presenti riferimenti a questi documenti.

Questo documento *non* è una lista aggiornata di exploit. Ne vengono scoperti in continuazione di nuovi. Qui si potranno avere indicazioni su dove cercare questo tipo di informazioni aggiornate e alcuni metodi generici per evitare che tali exploit vengano utilizzati sul proprio sistema.

1.1. Nuove versioni di questo documento

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

`http://www.linuxdoc.org/`

L'ultimissima versione di questo documento dovrebbe essere disponibile in vari formati anche da:

- `http://scrye.com/~kevin/lsh/`
- `http://www.linuxsecurity.com/docs/Security-HOWTO`
- `http://www.tummy.com/security-howto`

1.2. Commenti e correzioni

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

`kevin-securityhowto@tummy.com (mailto:kevin-securityhowto@tummy.com)`

e

`dave@linuxsecurity.com (mailto:dave@linuxsecurity.com)`

Nota: per favore si mandino i commenti ad *entrambi* gli autori. Inoltre, ci si assicuri di includere "Linux" "security" o "HOWTO" nel subject per evitare il filtro antispam di Kevin.

1.3. Liberatoria

Non mi prendo la 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 delle descrizioni si riferiscono al setup di sistema della distribuzione RedHat(tm). Il setup del proprio sistema potrebbe variare da quanto descritto qui.

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

2. Panoramica

Questo documento proverà a spiegare alcune procedure e il software usato comunemente per migliorare la sicurezza del proprio sistema Linux. È importante, prima di cominciare, discutere alcuni dei concetti di base e descrivere i fondamenti della sicurezza informatica.

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 intrusi, detti "cracker", possono ottenere l'accesso al sistema, e poi usare le loro approfondite conoscenze per impersonare altri utenti, rubare informazioni o anche negare l'accesso alle proprie stesse risorse. Se ci si sta chiedendo qual'è la differenza fra un "Hacker" e un "Cracker", si veda il documento di Eric Raymond "How to Become A Hacker" ("Come Diventare Un Hacker"), reperibile presso http://www.catb.org/~esr/faqs/hacker-howto.html (http://www.catb.org/~esr/faqs/hacker-howto.html).

2.2. Quanta sicurezza garantisce sicurezza?

Prima di tutto, si tenga a mente che nessun sistema informatico potrà mai essere completamente sicuro. Tutto quello che si può fare è rendere sempre più difficile per gli altri compromettere il 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 sistema è sicuro, più la sicurezza diventa intrusiva. Si deve decidere a quale punto di equilibrio il sistema sarà ancora utilizzabile ma sicuro, relativamente all'uso che se ne fa.

Per esempio, si può esigere che chiunque tenti di fare un login via telefono col vostro sistema usi un modem che lo richiami a casa. Questo dà più sicurezza, ma se questo qualcuno non è a casa sarà difficile che possa connettersi. Si potrebbe anche configurare il sistema Linux senza rete o connessione a Internet, ma questo ne limiterebbe l'utilità.

Se si è un organizzazione di dimensioni medio-grandi, si dovrebbe stabilire una politica di sicurezza che definisca quanto deve essere sicuro il sito e quale verifica effettuare per controllarla. Si può trovare un famoso esempio di politica di sicurezza su http://www.faqs.org/rfcs/rfc2196.html. Il documento è stato recentemente aggiornato e contiene un'ottima struttura per stabilire una politica di sicurezza per la propria organizzazione.

2.3. Cosa si sta cercando di proteggere?

Prima di tentare di rendere sicuro il sistema, si dovrebbe determinare contro che livello di attacco ci si vuole proteggere, quali rischi si dovrebbero correre o meno e quanto vulnerabile ne risulterebbe il sistema. Lo si dovrebbe analizzare per sapere cosa si sta proteggendo, perché lo si sta proteggendo, quanto vale e chi ne ha la responsabilità.

- *Il rischio* è la possibilità che un intruso riesca nel tentativo di accedere al computer. Un intruso può leggere o scrivere file, o eseguire programmi che potrebbero fare danno? Può cancellare dei dati importanti? Può bloccare il proprio lavoro e quello della organizzazione? Da non dimenticare: qualcuno che ha accesso al vostro account, o al sistema, può anche impersonarvi.

  Inoltre, avere un account insicuro nel sistema può compromettere l'intera rete. Se si permette anche ad un solo utente di fare login usando un file `.rhosts` o di usare un servizio insicuro come `tftp`, si rischia di far mettere ad un intruso "il piede nella porta". Una volta che costui ha un account sul sistema, o su quello di qualcun altro, può usarlo per avere accesso ad un altro sistema, o ad un altro account.

- Una *minaccia* viene tipicamente da qualcuno con un movente per cercare di ottenere un accesso non autorizzato alla rete o al computer. Si deve decidere di chi ci si può fidare e che tipo di pericolo potrebbe rappresentare.

  Ci sono diversi tipi di intrusi, ed è utile tenere a mente le loro caratteristiche quando si configurano le difese del sistema.

  - *Il Curioso* - Questo tipo di intruso è principalmente interessato a scoprire che tipo di dati e di sistema si hanno.
  - *Il Malizioso* - Questo tipo di intruso tenta di mettere fuori uso il sistema, o di modificare la pagina web, o altrimenti di far perdere tempo e denaro per riparare i danni che causa.
  - *L'intruso di Alto Profilo* - Questo tipo di intruso tenta di usare il sistema per guadagnare popolarità o fama. Potrebbe usare il sistema di alto profilo per pubblicizzare le sue abilità.
  - *La Concorrenza* - Questo tipo di intruso ha interesse per i dati che si hanno sul sistema. Potrebbe essere qualcuno che pensa che si abbia qualcosa di cui potrebbe beneficiare, finanziariamente o in altro modo.
  - *Lo Sfruttatore* - Questo tipo di intruso è interessato a sfruttare il 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 giocatore di cavallina* - Questo tipo di intruso è interessato al sistema solo per usarlo come trampolino per entrare in altri. Se il sistema ha una buona connessione o è il passaggio verso molti altri host, è probabile che ci si trovi ad affrontare questo tipo di intruso.

- La vulnerabilità esprime quanto bene il computer è protetto dalle altre reti, e la possibilità che qualcuno ottenga accesso non autorizzato.

  Cosa è in gioco se qualcuno entra nel sistema? Ovviamente le preoccupazioni di un utente casalingo con un IP dinamico saranno diverse da quelle di una società che connette le proprie macchine ad Internet o ad un'altra grande rete.

  Quanto tempo richiederebbe recuperare/ricreare dei dati andati persi? Un investimento di tempo ora può far risparmiare dieci volte tanto in seguito, se si dovessero ricreare dati perduti. È stato controllato il vostro sistema di backup recentemente e sono stati verificati i dati?

2.4. Sviluppare una politica di sicurezza.

Si crei una strategia semplice e generica per il sistema che gli utenti possano facilmente capire e seguire. Dovrebbe proteggere, oltre ai dati a cui si tiene, anche la privacy degli utenti. Alcune cose che si potranno considerare sono: chi ha accesso al sistema (Il mio amico può usare il mio account?), chi è autorizzato a installare programmi sul sistema, chi possiede quali dati, recupero di "disastri" e uso appropriato del sistema.

Una strategia di sicurezza generalmente approvata comincia col motto:

"*Quello che non è permesso è proibito*"

Questo significa che a meno che non si sia dato ad un utente l'accesso ad un servizio, quell'utente non dovrebbe usarlo finché non gli viene permesso. Ci si assicuri che la 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 (ftp://www.faqs.org/rfcs/rfc1244.html) è un documento che spiega come creare una propria strategia di sicurezza di rete.

rfc1281 (ftp://www.faqs.org/rfcs/rfc1281.html) è un documento che mostra una strategia di esempio con descrizioni dettagliate di ogni passo.

Infine si potrebbe dare un'occhiata all'archivio politiche COAST presso ftp://coast.cs.purdue.edu/pub/doc/policy per vedere come sono fatte delle strategie applicabili alla pratica.

2.5. Mezzi per proteggere il vostro sito

Questo documento discuterà diversi mezzi con cui si può mettere al sicuro la struttura per la quale si è duramente lavorato: la propria macchina locale, i dati, gli utenti, la rete e persino la propria reputazione. Cosa succederebbe alla propria reputazione se un intruso cancellasse dati degli utenti? O modificasse il sito? O pubblicasse il piano corporativo della società per la prossima stagione? Se si sta pianificando l'installazione di una rete, ci sono diversi fattori da considerare prima di aggiungerle una singola macchina.

Anche se si ha un solo account PPP, o solo un piccolo sito, non significa che gli intrusi non saranno interessati a voi. Grandi sistemi di alto profilo non sono i soli bersagli -- molti intrusi vogliono solo superare la sicurezza di più siti possibile, ignorando le loro dimensioni. Inoltre, possono usare un buco nella sicurezza del sistema per riuscire ad entrare in altri siti a cui si è connessi.

Questi individui hanno tutto il tempo che vogliono, e possono evitare di cercare di indovinare come è stato protetto il sistema semplicemente provando tutte le possibilità. C'è inoltre una serie di ragioni per cui un intruso può interessarsi ai propri sistemi, che discuteremo più avanti.

2.5.1. Sicurezza dell'host

Forse l'area di sicurezza sulla quale gli amministratori si concentrano di più è quella basata sui singoli host. Questo in genere implica l'assicurarsi che il proprio sistema sia sicuro, e sperare che tutti gli altri nella rete facciano lo stesso. Scegliere buone password, rendere sicuri i servizi di rete del proprio host locale, tenere buoni log degli account, e aggiornare i programmi con ben noti 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 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 più computer nella stessa rete, non si può sperare che ognuno di essi sia sicuro. Assicurarsi che solo utenti autorizzati possano accedere al sistema, costruire firewall, usare la crittografia forte e essere certi che non vi siano macchine "vagabonde" (cioè insicure) sono tutti compiti dell'amministratore della sicurezza di rete.

Questo documento discuterà alcune delle tecniche usate per rendere sicuro il sito e mostrerà alcuni modi per impedire che un intruso riesca ad accedere a quello che si sta tentando di proteggere.

2.5.3. Sicurezza tramite oscurità

Un tipo di sicurezza che deve essere discusso è la "sicurezza tramite oscurità". Questo significa, per esempio, spostare un servizio che è notoriamente "fragile" per la sicurezza su di una porta non standard, nella speranza che chi attacca non noti che è lì. Sicuramente verrà trovato e sfruttato. La sicurezza tramite oscurità non è assolutamente sicurezza. Semplicemente perché si ha un piccolo sito, o di profilo relativamente basso, non significa che un intruso non sarà interessato a quello che si ha. 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, la Sezione 3, spiega come proteggere fisicamente la macchina dalle manomissioni. Il secondo, la Sezione 4, descrive come proteggere il sistema dalle manomissioni degli utenti locali. Il terzo, la Sezione 5, mostra come configurare il filesystem e i privilegi sui file. Il successivo, la Sezione 6, discute come usare la crittografia per rendere più sicura la propria macchina e la rete. la Sezione 7 discute quali opzioni del kernel si dovrebbero attivare o conoscere per un sistema più sicuro. la Sezione 8, descrive come rendere il sistema Linux più difficile da attaccare via rete. la Sezione 9, discute come preparare le macchine prima di metterle in rete. Quindi, la Sezione 10, tratta cosa fare quando si scopre una compromissione in atto o una successa recentemente. In la Sezione 11, sono citate alcune importanti risorse per la sicurezza. La sezione D e R la Sezione 13, risponde ad alcune domande frequenti, e infine troverete una conclusione in la Sezione 14

I due punti principali da tenere a mente leggendo questo documento sono:

- Essere consapevoli del proprio sistema. Si controllino i log, come `/var/log/messages` si tenga d'occhio il sistema, e
- lo si tenga aggiornato assicurandosi di avere installato l'ultima versione dei software e di averli aggiornati in seguito ad allarmi di sicurezza. Fare semplicemente questo, sarà di aiuto per rendere il sistema decisamente più sicuro.

3. Sicurezza fisica

Il primo livello di sicurezza di cui bisogna tenere conto è quello fisico dei sistemi. Chi ha fisicamente accesso ai computer? Dovrebbe averlo? Si riesce a proteggere la macchina da eventuali intrusioni? Lo si dovrebbe fare?

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

Se si è un utente casalingo, probabilmente non ne serve molta (anche se si potrebbe voler proteggere la macchina dalla curiosità di bambini o parenti noiosi). Se si tratta di un laboratorio, ne servirà molta di più, ma gli utenti dovranno comunque essere in grado di lavorare sulle macchine. Molte delle seguenti sezioni saranno d'aiuto. Se si è in un ufficio, si potrebbe o meno avere bisogno di tenere al sicuro le macchine fuori dall'orario di lavoro o quando non si è presenti. In certe società, lasciare incustodita la propria postazione è un motivo di licenziamento.

Ovvi metodi di sicurezza fisica come lucchetti, macchine chiuse a chiave e video sorveglianza sono buone idee, ma vanno oltre lo scopo 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 che può essere alternata fra le posizioni "chiuso" e "aperto" con una chiave. Queste serrature possono aiutare ad evitare che qualcuno rubi il 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 componenti modificati.

Queste serrature fanno cose diverse a seconda del supporto nella scheda madre e di come è costruito il case. In molti PC fanno in modo di obbligare a rompere il case per aprirlo. In altri, non permetteranno di inserire nuove tastiere o mouse. Si controllino le istruzioni della 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 eluse con lo scasso.

Alcune macchine (soprattutto SPARC e Mac) hanno un gancio sul retro in cui far passare una catena che un ipotetico intruso dovrebbe tagliare oppure rompere il case per entrarci. Semplicemente metterci un lucchetto a chiave o a combinazione, può essere un buon deterrente contro i furti.

3.2. Sicurezza del BIOS

Il BIOS è il più basso livello software per configurare o manipolare il proprio hardware basato su architettura 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...). Si può usare il BIOS per evitare che un intruso riavvii la macchina e manipoli il sistema Linux.

Molti BIOS di PC 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 si perderà del tempo e si lasceranno delle tracce). Allo stesso modo, su S/Linux (Linux per macchine con processore SPARC(tm)), la EEPROM può essere configurata per richiedere una password di avvio. Questo può rallentare l'intruso.

Un altro rischio nel fidarsi delle password di avvio per rendere più sicuro il vostro sistema è il problema delle password predefinite. Molti produttori di BIOS non si aspettano che la gente apra il proprio computer e stacchi le batterie se si dimentica della propria password e hanno perciò equipaggiato i BIOS con delle password predefinite che funzionano indipendentemente dalla password scelta. Alcune delle più comuni sono:

j262 AWARD_SW AWARD_PW lkwpeter Biostar AMI Award bios BIOS setup cmos AMI!SW1 AMI?SW1 password hewittrand shift + s y x z

Ho testato un Award BIOS con AWARD_PW e ha funzionato. Queste password sono disponibili sui siti web dei produttori e su http://astalavista.box.sk e quindi una password di avvio non può essere considerata una protezione adeguata contro un attaccante preparato.

Molti BIOS permettono anche di specificare altri e migliori settaggi di sicurezza. Si controlli il manuale del BIOS o lo si controlli la prossima volta che viene avviato il computer. Per esempio, alcuni BIOS possono disabilitare l'avvio da floppy e alcuni richiedono una password per accedere ad alcune funzioni.

Nota: se si ha una macchina server e viene inserita una password di avvio, il server non ripartirà da solo. Si tenga a mente che si dovrà andare a mettere la password in caso di mancanza di 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 si specificano opzioni (come `single`) al prompt di `LILO`.

>Dalla pagina man di lilo.conf:

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

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

       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).
```

Si tenga presente quando vengono settate tutte queste password che poi bisognerà ricordarsele. :) Si ricordi anche che tutte queste password si limiteranno a rallentare un intruso ben determinato. Non impediranno di avviare da un floppy e quindi montare la partizione di root. Se si sta usando questo tipo di sicurezza si dovrà anche disabilitare il boot da floppy nel BIOS del computer, e quindi proteggerlo con una password.

Si tenga anche presente che /etc/lilo.conf dovrà avere i permessi "600" (leggibile e scrivibile solo da root) o altri saranno in grado di leggere le password!

>Dalla pagina info di Grub: GRUB fornisce la funzione "password", così che solo l'amministratore possa far partire le operazioni interattive (per es. modificare voci del menù e usare l'interfaccia a linea di comando). Per usare questa funzione, si deve inserire il comando "password" nel file di configurazione (*vedere password::), così:

password --md5 PASSWORD

Se specificato, GRUB può disabilitare tutti i controlli interattivi, finché si preme il tasto <p> e si inserisce la password corretta. L'opzione '--md5'indica a GRUB che 'PASSWORD' è nel formato MD5. Se omesso, GRUB assume che la password sia in chiaro.

Si può crittografare la password con il comando 'md5crypt' (*vedere md5crypt::). Per esempio, si faccia partire la shell di grub (*vedere Richiamare la shell di grub::), e si inserisca la password:

grub>md5crypt Password: ********** Encrypted: 1U$JK7xFegdxWH6VuppCUSIb.

Ora, si può copiare e incollare la password crittografata nel proprio file di configurazione.

Grub ha anche un comando 'lock' che permette di bloccare una partizione se non viene inserita la password corretta. È sufficiente aggiungere 'lock' e la partizione non sarà accessibile finché non si fornisce una password.

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

Nota: se si ha una macchina server e viene inserita una password di avvio, il server *non* ripartirà da solo. Si tenga a mente che bisognerà andare a mettere la password in caso di mancanza di corrente. ;(

3.4. xlock e vlock

Se ci si allontana dal computer ogni tanto, torna utile poter "bloccare" la postazione così che nessuno possa manomettere o vedere il proprio 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. Si controlli la sua pagina man per trovare altre opzioni, ma in genere si può 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 della Linux box. Si può bloccare solo quella in uso o tutte. Se se ne blocca una sola, altri potranno venire ad usare quelle sbloccate; semplicemente non potranno usare la console bloccata finché non verrà sbloccata. `vlock` è distribuito con RedHat Linux, ma non è detto che sia l'unica.

Chiaramente bloccare la console impedirà ad altri di manipolare il proprio lavoro, ma non di riavviare la macchina o distruggere il lavoro in altri modi. Inoltre non evita che qualcuno acceda alla 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. Per questi motivi, si dovranno usare questi programmi solo sotto il controllo di xdm.

3.5. Sicurezza dei dispositivi locali

Se si hanno una webcam o un microfono collegati al sistema, si dovrà considerare se c'è pericolo che un attaccante possa accedere a questi dispositivi. Quando non vengono utilizzati, si potrebbe scollegarli o rimuoverli. Altrimenti si dovrà leggere con attenzione e controllare ogni software che permette di accedere a questi dispositivi.

3.6. Scoprire infrazioni della sicurezza fisica

La prima cosa da tenere sempre d'occhio è quando la macchina è stata riavviata. Visto che Linux è un Sistema Operativo stabile e robusto, dovrebbe essere riavviato solo per aggiornamenti del SO, cambiamenti all' hardware, o cose del genere. Se la macchina si è riavviata senza un comando voluto, potrebbe essere il segno della manomissione di un intruso. Molti dei metodi di manomissione richiedono un riavvio o uno spegnimento del computer.

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

È 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 vengono inviati non cifrati, permettendo a un intruso di vederli mentre vengono trasferiti. Questo può rivelare informazioni sulla rete che non dovrebbero essere pubbliche. Ci sono anche demoni syslog che cifrano i dati inviati.

Si tenga 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 host locali senza indicare la loro vera origine.

Alcune cose da controllare nei log:

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

Parleremo dei dati dei log di sistema più avanti ne la Sezione 9.5 dell'HOWTO.

4. Sicurezza locale

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

Ottenere l'accesso all'account di un utente locale è uno dei primi tentativi che gli intrusi 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 si stringeranno le maglie della sicurezza locale, un intruso 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 si conoscono o che non si sa come contattare.

4.1. Creare nuovi account

Bisognerà essere certi di accordare agli utenti solo i privilegi indispensabili per il lavoro che devono svolgere. Se si da al proprio figlio (10 anni) un account, si potrebbe 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 sistema Linux:

- Dare loro solo i privilegi di cui hanno bisogno.
- Fare attenzione dove e quando fanno un login, o da dove dovrebbero farlo.
- Assicurarsi di rimuovere gli account inutilizzati, che possono essere determinati usando il comando "last" e/o controllando se nei file di log ci sono segni di attività dell'utente.
- È consigliabile, per facilitare la manutenzione degli account e per permettere una più facile analisi dei log, l'uso della stessa userid su tutti i computer e le reti.
- La creazione di gruppi di utenti dovrebbe essere assolutamente proibita. Gli account degli utenti permettono l'attribuzione delle responsabilità, e questo non è possibile con account di gruppo.

Molti account utente locali che vengono usati per compromettere la sicurezza non sono stati utilizzati per mesi o anni. Visto ciò, forniscono un ideale mezzo di attacco.

4.2. Sicurezza di root

L'account più preso di mira sulla vostra macchina è quello di root (superutente) . Questo account ha autorità su tutta la macchina e può anche includere l'autorità su altre macchine della rete. Si ricordi che si dovrebbe usarlo solo per compiti specifici e molto brevi, usando quindi per la maggior parte del tempo l' utente normale. Anche piccoli errori fatti da root possono causare problemi. Meno si useranno i privilegi di root, più si sarà al sicuro.

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

- Quando si esegue un comando complesso, provare prima ad eseguirlo in modo non distruttivo... soprattutto comandi che usano le wildcard: per es., se si vuole eseguire `rm foo*.bak`, prima si usi `ls foo*.bak` e ci si assicuri che si stanno per cancellare i file giusti. Anche usare echo al posto di comandi distruttivi può andare bene.
- Si fornisca agli utenti un alias predefinito per il comando `rm` per chiedere conferma della cancellazione dei file.
- Si usi root solo per portare a termine singoli compiti specifici. Se ci si trova a cercare di immaginare come fare qualcosa, si torni alla shell di utente normale finché non si sarà *sicuri* di cosa fare da root.
- Il percorso dei comandi per l'utente root è molto importante. Questo (cioè la variabile d'ambiente `PATH`) specifica le directory in cui la shell cerca i programmi. Si provi a limitare il percorso di root il più possibile e non si includa *mai* `.` (che significa "la directory corrente") nella variabile PATH. Inoltre, non si abbiano mai directory scrivibili nel percorso di ricerca, perché questo potrebbe permettere a degli intrusi di modificare o inserire nuovi eseguibili nel percorso, permettendo loro di divenire root la prossima volta che verrà eseguito quel comando.
- Non usare 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 creare 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). Si stia attenti quando si aggiunge qualcosa a questo file.

Sarebbe meglio loggarsi da remoto come utente normale e poi usare `su` se se ne ha bisogno (preferibilmente attraverso la Sezione 6.4 o un altro canale cifrato), in modo che non sia necessario fare un login remoto da root.

- Si cerchi di essere sempre calmi e riflessivi quando si è root. Le proprie azioni possono avere effetti su molte cose. Pensare prima di digitare!

Se si deve assolutamente permettere a qualcuno (possibilmente molto fidato) di accedere da root alla 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. Questo permette, per esempio, di lasciare che un utente espella e monti media removibili sul computer, senza avere altri privilegi di root. `sudo` inoltre tiene un log di tutti i tentativi, riusciti e non, di usarlo, permettendo 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 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. Si consideri `sudo` come un mezzo per conoscere le responsabilità di certe azioni, e non si speri 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 sistemi possono essere d'aiuto per 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. Si utilizzi l'opzione `nosuid` nel file `/etc/fstab` per partizioni scrivibili da chi non è root. Forse si vorrà anche usare `nodev` e `noexec` sulle partizioni home degli utenti e su `/var`, proibendo così l'uso di programmi e la creazione 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.
- Si configuri l'`umask` di creazione dei file dei vostri utenti in modo che sia più restrittiva possibile. Si veda la Sezione 5.1.
- Se si montano filesystem usando un filesystem di rete come NFS, ci si assicuri di configurare /etc/exports con restrizioni adeguate. Tipicamente è consigliabile usare nodev, nosuid, e magari noexec.
- Si impostino limiti per il filesystem invece di lasciarlo `illimitato` come di default. Si possono 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.

Si può anche usare il file di configurazione /etc/login.defs per impostare gli stessi limiti.

- 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 intruso) 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. Si veda la pagina man `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 installati 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 volta successiva, anche se il buco originale viene chiuso.

  Si trovino tutti i programmi SUID/SGID sul sistema, e si tenga traccia di cosa sono, così da essere al corrente di qualsiasi cambiamento che potrebbe indicare un eventuale intruso. Si usi questo comando per trovare tutti i programmi con SUID/SGID sul 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 confronta con quelli della sera precedente. Potete cercare questo log in `/var/log/setuid*`.

  Si possono togliere i permessi SUID o SGID da un programma sospetto con `chmod`, quindi rimetterli se si pensa 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 sistema, si usi il comando:

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

  e ci si assicuri 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` li esclude dal comando `find` precedente.

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

  ```
  root# find / \( -nouser -o -nogroup \) -print
  ```

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

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

- Per finire, prima di cambiare i permessi su un qualsiasi file di sistema, si sia certi di sapere quello che viene fatto. Non cambiare mai i permessi di un file perché sembra la via più facile di far funzionare le cose. Determinare 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 nel sistema. È 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. I permessi risultanti vengono calcolati come segue: i permessi predefiniti di proprietario/gruppo/altri (7 per le directory, 6 per i files) sono combinati con la maschera invertita (NOT) eseguendo un'operazione AND bit per bit.

Esempio 1:

file, permesso predefinito 6, binario: 110 maschera, per es. 2: 010, NOT: 101

permessi risultanti, AND: 100 (uguale a 4, r__)

Esempio 2:

file, permesso predefinito 6, binario: 110 maschera, per es. 6: 110, NOT: 001

permessi risultanti, AND: 000 (uguale a 0, ___)

Esempio 3:

directory, permesso predefinito 7, binario: 111 maschera, per es. 2: 010, NOT: 101

permessi risultanti, AND: 101 (uguale a 5, r_x)

Esempio 4:

directory, permesso predefinito 7, binario: 111 maschera, per es. 6: 110, NOT: 001

permessi risultanti, AND: 001 (uguale a 1, __x)

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

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

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 un utente per gruppo.

5.2. Permessi dei file

È importante assicurarsi che i 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 una 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, nel caso abbia il permesso di lettura

Attributo Save Text: (per le directory)

Anche lo "sticky bit" ha un significato diverso quando applicato a directory e quando a file. Se lo sticky bit è impostato 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 lungo 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, solo che ha effetto sul gruppo. Il file deve essere eseguibile perché questo abbia effetto.

Attributo SGID: (per le directory)

Se si imposta 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 si appartiene

Tutti - Chiunque non sia il proprietario o membro del gruppo

File di esempio:

```
-rw-r--r--  1 kevin  users         114 Aug 28  1997 .zlogin
1° bit - directory?             (no)
 2° bit - lettura per il proprietario?         (si, per kevin)
  3° bit - scrittura per il proprietario?        (si, per kevin)
   4° bit - esecuzione per il proprietario?      (no)
    5° bit - lettura per il gruppo?         (si, per users)
     6° bit - scrittura per il gruppo?        (no)
      7° bit - esecuzione per il gruppo?      (no)
       8° bit - lettura per tutti?      (si, 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 descritto. Forse vorrete dare più permessi di quanto vedete qui, ma questo mostra solo ciò che questi permessi minimi sui file 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 descritto. Forse vorrete dare più permessi di quanto vedete qui, ma questo mostra solo ciò che questi permessi minimi sulle directory 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`) sono in genere in modo `640` (`-rw-r-----`), e proprietà di root. A seconda delle necessità di sicurezza del sistema si può cambiare questa impostazione. Non si lasci 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 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.

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

Potreste persino aggiungere un elemento al `crontab` per eseguire il controllo dal floppy 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 intrusioni prima che possano essere notate in altri modi. Dato che molti files cambiano durante il normale uso del sistema, si deve fare attenzione a cosa è dovuto all'attività di un cracker e cosa a quello che si sta facendo.

Si può 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'Eneide di Virgilio. 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.

Si dovrebbe controllare quali programmi vengono installati sulla macchina. RedHat fornisce checksum MD5 e firme PGP sui suoi pacchetti RPM perché si possa verificare ciò che viene installato. Altre distribuzioni hanno metodi simili. Non si dovrebbero mai eseguire binari che non si conoscono, di cui non si ha il sorgente, come root. Ovviamente pochi intrusi rilasciano il codice sorgente al pubblico.

Per quanto possa essere complesso, si prendano sempre i sorgenti di un programma dal suo vero sito di distribuzione. Se il programma deve essere eseguito da root, si controllino i sorgenti o si facciano controllare da qualcuno di cui ci si fida.

6. Sicurezza delle password e crittografia

Uno dei mezzi di sicurezza più importanti usati oggi sono le password. È importante per se e per i propri utenti avere password sicure e difficili da indovinare. Molte delle recenti distribuzioni Linux includono programmi `passwd` che non permettono di inserire password troppo facili. Ci si assicuri che il proprio `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 è utilissima, 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 crittografata è poi conservata in `/etc/passwd` o (più raramente) `/etc/shadow`. Quando si tenta di entrare la password che viene scritta è crittografata 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 (si può codificare e decodificare un messaggio, con le chiavi giuste), 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" (si veda la Sezione 6.9) possono spesso indovinare la password se non è abbastanza casuale. I moduli PAM (vedi sotto) permettono di usare una diversa routine crittografica per le proprie password (MD5 o simili). Si potrebbe anche usare Crack a proprio vantaggio. Si esegua periodicamente Crack sul database delle password, per trovare quelle insicure. Poi si contatti l'utente in questione e gli si faccia cambiare password.

Si può 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 crittografare e una per decrittografare. Al contrario, la crittografia tradizionale usa la stessa chiave per crittografare e decrittografare; 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 crittografare, mentre ogni persona tiene la sua chiave privata per decrittografare i messaggi codificati con la sua chiave pubblica.

Ci sono vantaggi in entrambi i metodi di crittografia, e si possono scoprire queste differenze nelle FAQ della RSA sulla crittografia (http://www.rsa.com/rsalabs/faq/), 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, si dia un'occhiata alla FAQ di PGP: http://www.pgp.com/service/export/faq/55faq.cgi

Ci si assicuri di usare la versione valida per il proprio 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. È scritta per la versione internazionale di PGP, ma è facilmente adattabile alla versione USA. Si potrebbe 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 (http://www.faqs.org/rfcs/rfc2440.html). Si veda il sito di GNU Privacy Guard per avere più informazioni: http://www.gnupg.org/ (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 si troveranno informazioni su termini come "Diffie-Hellman", "crittografia a chiave pubblica", "certificati digitali", ecc.

6.2. SSL, S-HTTP 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 sullo strato di trasporto, crea un canale di dati sicuro e crittografato e può quindi crittografare dati 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 esso, oltre che con altro software Netscape. Ulteriori informazioni si trovano presso http://www.consensus.com/security/ssl-talk-faq.html. Informazioni su altre forme di sicurezza di Netscape e un buon punto di partenza su questi protocolli sono disponibili presso http://home.netscape.com/info/security-doc.html. È importante notare che il protocollo SSL può essere usato per passare molti altri protocolli comuni, "incapsulandoli" per sicurezza. Si veda: http://www.quiltaholic.com/rickk/sslwrap/
- *S-HTTP:* - S-HTTP è un altro protocollo che fornisce servizi di sicurezza su Internet. È 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 crittografa 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 la posta elettronica e altri tipi di messaggi su Internet. È 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. Si troveranno 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 di rete detti x-kernel, e si trova presso http://www.cs.arizona.edu/xkernel/hpcc-blue/linux.html. Più semplicente, l'x-kernel è un metodo per passare messaggi al livello del kernel, il che rende l'implementazione più facile.

Un'altra implementazione liberamente disponibile di IPSEC è 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 crittografata dalla macchina gateway IPSEC e decrittografata dal gateway all'altro capo. Il risultato è una Virtual Private Network (Rete Privata Virtuale) o VPN. È una rete che rimane privata nonostante includa macchine connesse attraverso l'insicura Internet."

È scaricabile 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 permettono di fare login a sistemi remoti su di una connessione crittografata.

`openssh` è una suite di programmi usata come sostituto sicuro di `rlogin`, `rsh` e `rcp`. Usa una crittografia a chiave pubblica per crittografare 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. Può effettuare la 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 ssh di Data Fellows ed è stata completamente ricostruita per non includere alcuna parte proprietaria o brevettata. È 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 si veda: http://www.net.lut.ac.uk/psst/

Si può 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. Si può 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 si visiti http://www-cs-students.stanford.edu/~tjw/srp/

6.5. PAM - Pluggable Authentication Modules (moduli aggiuntivi di autenticazione)

Le ultime versioni delle distribuzioni Red Hat Linux e Debian Linux sono distribuite 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 ci si assicuri di dare un'occhiata al sito web di PAM per altre informazioni. http://www.kernel.org/pub/linux/libs/pam/index.html.

Solo alcune delle cose che si possono 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, si possono prevenire molti attacchi. Per esempio, si può usare 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 crittografati. Il motore crittografico è affiancato al driver che manda e riceve pacchetti.

Il comportamento è diverso in SSH, il quale esegue la crittografia dei dati per connessione, a livello socket. Viene crittografata una connessione logica tra programmi che girano su host diversi.

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 in maniera trasparente fra le due reti connesse con la VPN, senza modifiche al software applicativo.

Riassumendo la documentazione CIPE:

"Gli standard IPSEC definiscono un set di protocolli che possono essere usati (tra le altre cose) per costruire VPN crittografate. 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.

Si possono trovare ulteriori informazioni su Kerberos leggendo le FAQ di Kerberos (http://www.cis.ohio-state.edu/hypertext/faq/usenet/kerberos-faq/general/faq.html), inoltre si può reperire 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 proprio host. Porta a molte conseguenze e non è usato quanto, per esempio, SSH.

6.8. Shadow Password.

Le shadow password sono un mezzo per tenere segrete agli utenti normali le password crittografate. Le ultime versioni di RedHat e Debian Linux usano di default le shadow password, ma in altri sistemi le password crittografate 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 le shadow password ci si deve assicurare che tutte le applicazioni che devono leggere le password siano ricompilate per supportarle. PAM (vedi sopra) invece permette di installare semplicemente un modulo shadow; non richiede la ricompilazione degli eseguibili. Si può fare riferimento allo Shadow-Password HOWTO per ulteriori informazioni se necessario. È disponibile presso http://metalab.unc.edu/LDP/HOWTO/Shadow-Password-HOWTO.html È abbastanza datato al momento, e non è necessario per le distribuzioni che supportano 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, crittografandole tutte e comparandole con le password crittografate. 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.openwall.com/john/). Usano moltissimo la cpu, ma, eseguendoli, si può capire se possono trovare le password e si potranno quindi avvertire gli utenti con password deboli. Si noti che un intruso dovrebbe prima riuscire a trovare qualche altro buco per leggere il vostro `/etc/passwd` ma questi buchi sono più comuni di quanto pensiate.

Poichè la sicurezza è forte soltanto quanto il più insicuro degli host, è bene ricordare che se si hanno macchine Windows sulla rete dovreste controllare L0phtCrack, una versione Windows di Crack. È disponibile presso http://www.l0pht.com

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

CFS è un modo per crittografare interi alberi di directory e dare modo 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://www.tcfs.it/.

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

È importante rendere sicura l'interfaccia grafica per evitare che un intruso ottenga la password mentre la si scrive, legga i documenti o le informazioni che si hanno sullo schermo o persino che usi 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 se 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 macchina, può aggiungere la sua macchina all'xhost ed ottenere facile accesso. Inoltre, se si deve dare l'accesso ad una macchina inaffidabile, chiunque sia lì può compromettere la GUI.

Quando si usa `xdm` per entrare, si ha un metodo di accesso molto migliore: MIT-MAGIC-COOKIE-1. Un "cookie" a 128 bit viene creato nel file `.Xauthority`. Se si deve permettere che una macchina remota acceda al display, si può usare il comando `xauth` e le informazioni nel file `.Xauthority` per dare accesso solo a quella connessione. Si legga il Remote-X-Apps mini-howto, disponibile presso http://metalab.unc.edu/LDP/HOWTO/mini/Remote-X-Apps.html.

Si può anche usare `ssh` (si veda la Sezione 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 crittografato passi per la rete.

Si possono anche impedire connessioni remote al server X usando l'opzione '-nolisten tcp' del server X. Questo previene connessioni di rete al server tramite socket tcp.

Si dia uno sguardo alla pagina man di `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 macchina Linux. Questo li rende molto pericolosi. Se si bloccano, in genere si avrà bisogno di riavviare la macchina per riavere una console usabile. Ci si assicuri che tutti i programmi SVGA che vengono usati siano autentici e siano almeno abbastanza affidabili. Ancor meglio, non se ne eseguano 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 console in ogni momento. Inoltre permetterà di usare una chiave sicura, per poter essere certi che non c'è alcun cavallo di Troia che manometta il programma `login` della console. http://synergy.caltech.edu/~ggi/

7. Sicurezza del kernel

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

Visto che il kernel controlla il networking del computer, è importante che sia molto sicuro e non venga compromesso. Per evitare alcuni dei recenti attacchi via rete si dovrebbe tentare di mantenere aggiornato il kernel. Si possono trovare nuovi kernel presso <65533> (ftp://ftp.kernel.org) o presso il distributore del proprio 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 si visiti la loro pagina web presso: http://www.kerneli.org

7.1. Opzioni di compilazione del kernel 2.0

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

- Network Firewalls (CONFIG_FIREWALL)

  Questa opzione dovrebbe essere attivata se si vuole usare qualche tipo di firewall o masquerading sulla macchina. Se sarà solo un semplice client, si può rispondere no.

- IP: forwarding/gatewaying (CONFIG_IP_FORWARD)

  Se si abilita L'IP forwarding, la macchina potrà assumere le funzioni di un router. Se la macchina è su una rete, si potranno inoltrare dati da una rete ad un'altra e forse si potrebbe pregiudicare un firewall che era stato messo lì proprio per evitarlo. Chi si connette con un modem può farne a meno, e gli altri dovrebbero riflettere sulle implicazioni di sicurezza. I computer che fanno da firewall lo abiliteranno e lo useranno insieme con un firewall.

Si può 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
```

Si ricordi 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 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, si deve digitare:

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

- IP: Firewalling (CONFIG_IP_FIREWALL)

  Questa opzione è necessaria se si vuole usare la macchina come firewall, usare il masquerading o se si vuole proteggere il sistema da qualcuno che entri attraverso l’interfaccia di composizione PPP.

- IP: firewall packet logging (CONFIG_IP_FIREWALL_VERBOSE)

  Questa opzione dà informazioni sui pacchetti ricevuti dal firewall, come origine, destinazione, porta ecc.

- IP: Drop source routed frames (CONFIG_IP_NOSR)

  Questa opzione dovrebbe essere abilitata. Questi pacchetti contengono l’intero percorso verso 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 server Linux fa da firewall volesse mandare fuori dei pacchetti, il server potrebbe "recitare la parte" di questa macchina, cioè inoltrare il traffico verso la destinazione richiesta, ma facendolo sembrare proveniente dal firewall stesso. Si controlli http://www.indyramp.com/masq se si vogliono 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 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 in realtà sono connessi al proxy locale. Si legga 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 si sta costruendo un firewall o un host di masquerading, la si dovrebbe abilitare. Quando vengono mandati dei dati fra un host ed un altro, non sempre vengono spediti come un singolo pacchetto di dati, ma piuttosto vengono scomposti 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 da attacchi del genere anche se rivolti 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 la si può lasciare fuori, ma rimane nel caso servisse.

- IP: Firewall packet netlink device (CONFIG_IP_FIREWALL_NETLINK)

  Questa utile opzione 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. Opzioni di compilazione del kernel 2.2

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 si legge quando viene usato l'Help nella fase `make config` della compilazione del kernel. Qui sono riportate solo le opzioni nuove. Si consulti 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 installare 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 permette di connettere un filtro a qualsiasi socket e determinare se i pacchetti debbano essere accettati o respinti. A meno che non si abbia una necessità particolare e si sia capaci di programmare un tale filtro, dovreste dire di no. Si noti 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. Si può usare questa caratteristica anche per bilanciare il traffico se si hanno 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 si deve avere accesso ad una macchina su Internet che abbia un programma come lynx o netscape). Per informazioni generali si legga: 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. Si legga il file `./linux/Documentation/networking/filter.txt` per maggiori informazioni.

- IP: Masquerading

  Il masquerading del kernel 2.2 è stato migliorato. Fornisce supporto addizionale per speciali protocolli di masquerading ecc. Ci si assicuri di leggere l'IP Chains HOWTO per ulteriori informazioni.

7.3. Device del kernel

Nel kernel ci sono alcuni device a blocchi e a caratteri che 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. È 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. Si noti che su alcuni sistemi potrebbe bloccarsi per molto tempo in attesa che nuova entropia sia generata dagli utenti. Quindi riflettere prima di usare /dev/random. (Forse la miglior cosa da fare è usarlo quando si deve generare codice importante: si dica 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 sei caratteri casuali sulla console, adatti per la generazione di password. Si può trovare `mimencode` nel pacchetto `metamail`.

Si legga `/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.

C'è una serie di buoni strumenti che aiutano con la sicurezza di rete, e un numero sempre maggiore viene incluso nelle distribuzioni Linux.

8.1. Sniffer di pacchetti

Uno dei modi più comuni con cui gli intrusi ottengono l'accesso a più sistemi della propria rete, è l'uso di uno sniffer di pacchetti su un host già compromesso. Questo "sniffer" semplicemente ascolta sulla porta Ethernet cose come `passwd`, `login` e `su` nel flusso dei pacchetti e quindi registra il traffico successivo. In questo modo chi lo usa ottiene le password di sistemi in cui non ha neppure tentato effrazioni. 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 account 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 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 di 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 si vogliono offrire. I servizi non desiderati devono essere disabilitati, così da avere una cosa in meno di cui preoccuparsi e un possibile buco in meno per un intruso.

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

Si potrebbero rimuovere (o commentare) servizi nel proprio `/etc/services`. Questo ne impedirà l'uso anche da client locali (cioè se si rimuove `ftp` e si prova a connettersi con ftp ad un host remoto da quella macchina, si otterrà 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 dovrebbero essere disponibili sono:

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

Se si sa che non serve un particolare pacchetto si può anche cancellarlo del tutto. `rpm -e nome_pacchetto` nella distribuzione RedHat cancella un pacchetto intero. Nella Debian `dpkg --remove` fa lo stesso.

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

Si dovrebbe 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 contenuti in `/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 si vuole solo disabilitare un servizio per un particolare run level, si rinomini il giusto link simbolico sostituendo la `S` maiuscola con una `s` minuscola, così:

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

Se si ha file `rc` in stile BSD, si controlli in `/etc/rc*` se ci sono programmi che non servono.

Con molte distribuzioni Linux viene distribuito un tcp_wrappers che "incapsula" tutti i 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 servizi TCP. Si dovrebbe creare un file `/etc/hosts.allow` e aggiungervi solo gli host che devono avere accesso ai servizi della macchina.

Se si ha una connessione telefonica casalinga suggeriamo di negarli TUTTI. Inoltre `tcpd` segna nei log i tentativi di accesso falliti, per avvertire se si è sotto attacco. Se si aggiungono nuovi servizi ci si deve assicurare di configurarli per usare tcp_wrappers se sono 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, si potrebbero aggiungere le linee seguenti al proprio `/etc/hosts.allow`:

ALL: 127.

E ovviamente /etc/hosts.deny dovrà contenere:

ALL: ALL

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

Si ricordi che tcp_wrapper protegge solo i servizi eseguiti da `inetd` e pochi altri. Ci potrebbero benissimo essere altri servizi in esecuzione sulla macchina. Si può usare `netstat -ta` per avere una lista di tutti i servizi offerti.

8.3. Verificare le proprie informazioni DNS

Tenere aggiornate le informazioni dei DNS su tutti gli host della rete aiuta ad aumentare la sicurezza. Se un host non autorizzato si connette alla rete, lo si può riconoscere dalla sua assenza dal DNS. Molti servizi possono essere configurati per negare l'accesso ad host che non hanno valide voci DNS.

8.4. identd

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

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

Perché eseguirlo allora? Perché aiuta ed è un'informazione in più quando si indaga. Se il proprio `identd` è integro, allora si saprà che fornisce ai siti esterni, il nome utente o l'uid di chi sta usando i servizi TCP. Se l'amministratore di una rete remota riporta che un certo utente ha provato ad intromettersi nella sua rete, si potranno facilmente prendere provvedimenti. Se non si sta eseguendo `identd`, si dovranno leggere righe e righe di log, capire chi era nel sistema alla tal ora e in generale sprecare molto tempo a trovarlo.

L' `identd` che viene fornito con molte distribuzioni è più configurabile di quanto molti pensino. Lo si può disabilitare per utenti specifici, (possono creare un file `.noident`) si possono avere i log di tutte le richieste `identd` (è raccomandato) e si può persino far sì che identd risponda con l'uid di un utente o persino con NO-USER.

8.5. Configurare e rendere sicuro lo MTA Postfix

Il server di posta Postfix è stato scritto da Wietse Venema, autore di Postfix e di alcuni altri prodotti relativi alla sicurezza per Internet, come un "tentativo di fornire un'alternativa all'ampiamente usato Sendmail. Postfix si propone di essere veloce, facile da amministrare e, si spera, sicuro, cercando di essere allo stesso tempo sufficientemente compatibile con Sendmail da non sconvolgere gli utenti."

Si possono trovare ulteriori informazioni su Postfix presso il sito: Postfix (http://www.postfix.org) e in Configurare e rendere sicuro Postfix (http://www.linuxsecurity.com/feature_stories/feature_story-91.html).

8.6. SATAN, ISS, e altri scanner di rete

C'è una serie di differenti pacchetti software che fanno scansioni di porte e servizi di macchine o reti intere. SATAN, ISS, SAINT e Nessus sono alcuni dei più conosciuti. Questi software si connettono con la macchina bersaglio (o tutte le macchine bersaglio su di una rete) su tutte le porte possibili e tentano di capire quale servizio è attivo. Basandosi su queste informazioni, si può dire 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 di una macchina o una rete. È una buona idea usare SATAN sulla rete e sistemare i problemi che trova. Ci si assicuri 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. È stata distribuita in rete una copia trojan di SATAN http://www.trouble.org/~zen/satan/satan.html. Si noti che SATAN non è aggiornato da molto tempo ed alcuni degli altri strumenti indicati sotto potrebbero dare risultati migliori.

ISS (Internet Security Scanner) è un altro port scanner. È 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. Si guardi alla sua home page sul web per ulteriori informazioni. http://www.psionic.com/abacus/ (http://www.psionic.com/abacus)

SAINT è una versione aggiornata di SATAN. È basata sul web ed ha molti più test aggiornati di SATAN. Se ne può sapere di più presso: http://www.wwdsi.com/~saint (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, si dia un'occhiata a: http://www.nessus.org (http://www.nessus.org/)

8.6.1. Rilevare scansioni delle porte

Esistono strumenti progettati per avvisare di scansioni da parte di SATAN, ISS ed altri software. Comunque se si usa molto tcp_wrapper, e si leggono spesso i log, si dovrebbero notare certe scansioni. Anche con la configurazione minima, SATAN lascia tracce nei log di un sistema RedHat "di serie".

Esistono anche port scanner "invisibili". Un pacchetto con il bit TCP ACK attivo (come si fa per le connessioni stabilite) probabilmente attraverserebbe un firewall che filtri i pacchetti. Il pacchetto RST in risposta da una porta che *_non ha sessioni attive_* viene preso come una prova di vita su quella porta. Non penso che i wrapper TCP lo rilevino.

Si può anche provare SNORT che è un IDS (Sistema di Rilevazione delle Intrusioni) libero che può rilevare molte intrusioni. http://www.snort.org

8.7. sendmail, qmail e MTA (agenti di trasporto di posta)

Uno dei servizi più importanti che si può offrire 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 si sta usando `sendmail` è molto importante tenerlo aggiornato. `sendmail` ha una lunga, lunga storia di exploit di sicurezza. Ci si assicuri di eseguire sempre l'ultima versione da: http://www.sendmail.org (http://www.sendmail.org/).

Si tenga a mente che per spedire la posta non è necessario che sendmail sia in esecuzione. Se si è un utente casalingo, si può disabilitare completamente sendmail e usare il proprio client di posta per spedire. Si potrebbe anche rimuovere l'opzione "-bd" dal file di avvio di sendmail, disabilitando le richieste di posta in arrivo. In altre parole, potete eseguire sendmail dal vostro script di avvio, usando il seguente comando:

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

In questo modo sendmail svuoterà la coda della posta ogni 15 minuti per ogni messaggio che non è stato consegnato con successo al primo tentativo.

Molti amministratori preferiscono non usare sendmail e scelgono invece uno degli altri MTA. Si potrebbe considerare di passare a `qmail`. `qmail` è stato progettato da zero tenendo a mente la sicurezza. È veloce, stabile e sicuro. Si trova presso: http://www.qmail.org

In competizione con qmail si pone "postfix", scritto da Wietse Venema, l'autore di tcp_wrappers e altri strumenti di sicurezza. Precedentemente chiamato vmailer e sponsorizzato da IBM, anche questo è un MTA fatto da zero per la sicurezza. Si possono trovare altre informazioni su postfix presso http://www.postfix.org (http:/www.postfix.org)

8.8. Attacchi Denial of Service

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. Si noti però che ne nascono in continuazione, quindi questi sono solo esempi. Si leggano le liste di sicurezza di Linux e le liste di bugtraq per informazioni aggiornate.

- *SYN Flooding* - Il SYN flooding 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 macchine. Si veda la Sezione 7 a proposito delle opzioni adeguate.
- *Il Bug "F00F" nei Pentium* - Si è scoperto recentemente che una serie di codici assembly mandati ad un Pentium Intel originale potrebbero riavviare 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 invia una enormità di pacchetti ICMP alla macchina. Se lo fa da un host con un'ampiezza di banda maggiore rispetto alla rete aggredita, la macchina non potrà mandare niente sulla rete. Un variazione di questo attacco, chiamato "smurfing", manda ad un terzo host pacchetti ICMP con l'indirizzo IP della macchina da attaccare permettendo una quasi anonimità. Si possono trovare altre informazioni circa lo "smurf" presso http://www.quadrunner.com/~chuegen/smurf.txt (http://www.quadrunner.com/~chuegen/smurf.txt)

  Se si viene attaccati in questo modo, si usi uno strumento come `tcpdump` per determinare da dove provengono i pacchetti (o da dove sembrano venire), quindi si contatti il proprio provider con queste informazioni. I ping flood possono essere fermati facilmente a livello 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. È 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".

Si può 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.

8.9. Sicurezza di NFS (Network File System)

NFS è un protocollo di condivisione di file molto diffuso. Permette a server che eseguano `nfsd` e `mountd` di esportare interi filesystem verso altre macchine che usano il file system NFS nel loro kernel (o qualche altro client di supporto 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.

È possibile avere un po' di sicurezza quando si esportano filesystem. Si può far mappare a `nfsd` l'utente root remoto (uid=0) sull'utente `nobody` negando l'accesso totale ai file 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 si deve usare NFS, ci si assicuri di esportare solo verso quelle macchine che lo richiedono. Non si esporti mai la propria intera directory root; si esporti solo il necessario.

Si legga lo NFS HOWTO per ulteriori informazioni su NFS, disponibile presso
http://metalab.unc.edu/mdw/HOWTO/NFS-HOWTO.html

8.10. NIS (Network Information Service) (ex YP).

Network Information service (ex YP) 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 file `/etc/passwd`). Questo permette agli utenti di cambiare la loro password una volta per tutte le macchine nel dominio NIS.

NIS non è affatto sicuro. Non ha mai voluto esserlo. Doveva solo essere comodo ed utile. Chiunque possa indovinare il nome del dominio NIS (ovunque sia nella rete) può ottenere una copia del file delle password ed usare Crack o John the Ripper contro le password degli utenti. Se si deve usare NIS si faccia attenzione ai pericoli cui si va incontro.

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

8.11. Firewall

I Firewall sono un mezzo per controllare quali informazioni vengono lasciate entrare ed uscire dalla 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, permettono di cambiare al volo i tipi di traffico di rete permessi. Si possono anche registrare specifici tipi di traffico di rete.

I firewall sono un'utilissima ed importante tecnica per la sicurezza di rete. Comunque, non pensare mai che solo perché si ha un firewall non sia necessaria la sicurezza delle macchine che copre. Sarebbe un errore fatale. Si veda 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 possono trovare altre informazioni anche nell'IP-Masquerade mini-howto:
http://metalab.unc.edu/mdw/HOWTO/mini/IP-Masquerade.html

Altre informazioni su `ipfwadm` (lo strumento che permette di cambiare le impostazioni del firewall) si trovano presso: http://www.xos.nl/linux/ipfwadm/

Se non si ha esperienza con i firewall e si vuole costruirne per un piano di sicurezza che non sia superficiale, il libro Firewalls di O'Reilly and Associates o qualche altro documento online sui firewall sono letture obbligatorie. Si troveranno 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. Lo si può trovare presso: http://csrc.nist.gov/nistpubs/800-10/main.html. Interessante anche:

- 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. È disponibile presso http://www.sunworld.com/swol-01-1996/swol-01-firewall.html
- Mason - il costruttore automatico di firewall per Linux. È uno script di firewall che impara cosa vi serve mentre lo fate. Altre informazioni presso: http://www.pobox.com/~wstearns/mason/

8.12. IP Chains - Firewall per Linux kernel 2.2.x

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

- Manipolazioni dei pacchetti più flessibile.
- Accounting più complesso.
- Semplici e istantanei 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 si sta usando `ipfwadm` con il kernel 2.0, sono disponibili degli script per convertire il formato di `ipfwadm` nel formato che usa `ipchains`.

Ci si assicuri di leggere l'IP Chains HOWTO per altre informazioni. È disponibile presso http://www.adelaide.net.au/~rustcorp/ipfwchains/ipfwchains.html

8.13. Netfilter - Firewall per Linux kernel 2.4.x

Sempre nell'ottica di migliorare il codice di filtraggio dei pacchetti IP, netfilter permette di costruire, mantenere e controllare le regole di firewalling nel nuovo kernel 2.4.

Il sottosistema netfilter è una riscrittura completa dell'implementazione precedente di ipchains e ipfwadm. Mette a disposizione un gran numero di miglioramenti, ed è diventato la soluzione più robusta per la protezione di reti aziendali.

```
iptables
```

è l'interfaccia da linea di comando usata per manipolare le tabelle di firewall.

Netfilter fornisce una struttura grezza per manipolare i pacchetti mentre attraversano le varie parti del kernel. Parte di questa include il supporto per il masquerading, il filtraggio standard dei pacchetti e un più completo sistema di traduzione degli indirizzi di rete. Include perfino un supporto migliorato per le richieste di bilanciamento del carico di rete, per uno specifico servizio, all'interno di un gruppo di server dietro il firewall.

Le caratteristiche di controllo dello stato sono particolarmente potenti. Il controllo dello stato permette di rintracciare e controllare il flusso di dati che passa il filtro. La capacità di tenere traccia delle informazioni sullo stato di una sessione, rende le regole più semplici e prova ad interpretare i protocolli di più alto livello.

In più, si possono sviluppare piccoli moduli allo scopo di effettuare funzioni specifiche supplementari, come passare i pacchetti da elaborare ai programmi in spazio utente e poi re-inserirli nel normale flusso di pacchetti. La possibilità di sviluppare questi programmi a livello utente riduce la complessità che precedentemente era associata al dover fare cambiamenti direttamente a livello del kernel.

Altri riferimenti a IP Tables:

- *Il tutorial di Oskar Andreasson su IP Tables (http://www.linuxsecurity.com/feature_stories/feature_story-94.html)* -- Oskar Andreasson parla con LinuxSecurity.com del suo esauriente tutorial su IP Tables e di come può essere usato per costruire un robusto firewall per la propria organizzazione.
- *Hal Burgiss presenta le guide rapide sulla sicurezza di Linux (http://www.linuxsecurity.com/feature_stories/feature_story-93.html)* -- Hal Burgiss ha scritto due autorevoli guide su come rendere sicuro Linux, compresa la gestione dei firewall.
- *Il sito di Netfilter (http://netfilter.samba.org)* -- Il sito di netfilter/iptables.
- *La maturazione dei firewall nel kernel Linux 2.4: netfilter (http://www.linuxsecurity.com/feature_stories/kernel-netfilter.html)* -- Questo articolo di LinuxSecurity.com presenta le basi del filtraggio dei pacchetti, come iniziare con iptables, e una lista di nuove funzioni disponibili nell'ultima generazione di firewall per Linux.

8.14. VPN - Virtual Private Network (reti private virtuali)

Le VPN sono un modo per stabilire una rete "virtuale" su una rete esistente. Questa rete virtuale è spesso crittografata e accetta il traffico solo verso e da entità conosciute. Le VPN sono spesso usate per connettere attraverso Internet qualcuno che lavora da casa ad un rete interna di una organizzazione.

Se si esegue un firewall/masquerading con Linux e si devono passare pacchetti di MS PPTP (un prodotto VPN punto-a-punto della Microsoft), c'è una patch per il kernel di Linux fatta apposta. Si veda: ip-masq-vpn (ftp://ftp.rubyriver.com/pub/jhardin/masquerade/ip_masq_vpn.html).

Esistono diverse soluzioni VPN per Linux:

- vpnd. Si veda http://sunsite.dk/vpnd/.
- Free S/Wan, disponibile presso http://www.xs4all.nl/~freeswan/
- ssh può essere usato per costruire una VPN. Si veda il VPN mini-howto per altre informazioni.
- vps (virtual private server) presso http://www.strongcrypto.com.
- yawipin presso http://yavipin.sourceforge.net (mailto:http://yavipin.sourceforge.net)

Si veda anche la sezione su IPSEC per altre bibliografie e informazioni.

9. Preparazione della sicurezza (prima di entrare in rete)

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

9.1. Fare 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 si hanno meno di 650mb di dati da salvare su una partizione, un CD-R è un'ottima strada (perché difficile da manomettere e molto durevole), ovviamente si ha bisogno di 650mb di spazio sul disco per creare l'immagine. Nastri e altri media riscrivibili dovrebbero essere protetti dalla scrittura non appena il backup è completo e quindi verificati per evitare la manomissione. Ci si assicuri di lasciare i backup in un'area sicura e off-line. Un buon backup 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. Si esegua un backup incrementale ogni giorno, e un backup completo sul nastro del Venerdì. Si dovrebbe fare un backup completo anche quando si apportano cambiamenti o si aggiungono dati particolarmente importanti al sistema.

9.3. Verificare i propri backup

Si dovrebbero controllare periodicamente i backup per verificare che funzionino come dovrebbero. Si dovrebbero recuperare regolarmente i files e confrontarli con i dati reali, controllare la dimensione e l'indice dei backup e rileggere quelli più vecchi.

9.4. Si faccia un backup dei propri database di RPM o Debian

Nel caso di un'intrusione, si può usare il proprio database di pacchetti come si userebbe tripwire, ma solo se si è sicuri che non è stato modificato. Si dovrebbe 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 sistema viene compromesso si può usare il comando:

```
root#  rpm -Va
```

per verificare ogni file sul sistema. Si legga la pagina man di `rpm`, perché esistono alcune opzioni che possono essere usate per avere un output più conciso. Si ricordi che si deve 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. Sta a voi valutare i vantaggi contro gli svantaggi.

9.5. Tenere nota dei dati degli account

È molto importante che le informazioni che vengono da `syslog` non siano modificate. Rendere leggibili e scrivibili i files in `/var/log` solo da poche persone è un buon inizio.

Ci si assicuri 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 distribuzione. In un sistema Linux conforme al "Linux Filesystem Standard", come RedHat, si dovrà cercare in `/var/log` e controllare `messages`, `mail.log`, ed altri.

Si può 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.

Si potrebbe anche voler configurare lo script o il demone che ruota i log per farglieli tenere più a lungo, così da avere tutto il tempo per esaminarli. Si dia un'occhiata al pacchetto `logrotate` su distribuzioni RedHat recenti. Altre distribuzioni hanno probabilmente un processo simile.

Se i log sono stati manomessi, bisogna cercare 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.

I log, anche se di solito gli intrusi li modificano per coprire le loro tracce, dovrebbero comunque essere controllati. Si potrebbe notare l'intruso che cerca un accesso o sfrutta un programma per ottenere l'account di root. Si potrebbero vedere voci nei log che l'intruso non ha avuto il tempo di cambiare.

Ci si deve anche assicurare di separare `auth` dagli altri log, inclusi i tentativi di cambiare utente con `su`, i tentativi di login e altre informazioni sugli account.

Se possibile, si configuri `syslog` per mandare una copia dei log ad un sistema sicuro. Questo eviterà che un intruso copra le sue tracce cancellando i tentativi di login/su/ftp/ecc. Si legga la pagina man di `syslog.conf` e si cerchi l'opzione `@`.

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

Un altro `syslogd` con altre caratteristiche è syslog-ng (http://www.balabit.hu/en/downloads/syslog-ng/). Permette molta flessibilità e crittografa i flussi remoti di syslog per evitare la manomissione.

Infine, i log sono inutili se nessuno li controlla. Ci si prenda del tempo ogni tanto per leggere i propri log e farsi un'idea di come devono essere in normali condizioni. Saperlo fa risaltare molte cose strane.

9.6. Applicate tutti i nuovi aggiornamenti di sistema.

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

10. Cosa fare durante e dopo un'intrusione

Sono stati seguiti alcuni di questi consigli ed è stata rilevata 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 di tensione. Il modo in cui si reagisce può avere grandi conseguenze.

Se la compromissione è fisica probabilmente è stato colto qualcuno che si è intrufolato in casa, in ufficio o nel laboratorio. Si dovrebbero avvisare le autorità competenti. In un laboratorio si potrebbe aver visto qualcuno che provava ad aprire un case o a riavviare una macchina. A seconda dell'autorità di cui si dispone e delle procedure, si potrebbe farli desistere o chiamare direttamente la sicurezza.

Se si è preso un utente locale che tentava di compromettere la sicurezza, la prima cosa da fare è confermare che sia davvero chi si pensa. Si controlli il posto da cui si connettono. È da dove si connettono normalmente? No? Allora si utilizzi un mezzo di comunicazione non elettronico. Per esempio, li si chiami al telefono o si vada nel loro ufficio/appartamento per parlarci. Se ammettono il fatto, chiedere spiegazioni su cosa stavano facendo, o gli si dica di smetterla. Se non c'entrano e non sanno di cosa si sta parlando, probabilmente si dovrà investigare ulteriormente. Lo si faccia e ci si procuri più informazioni possibili prima di accusare qualcuno.

Se è stato rilevato un attacco di rete la prima cosa da fare (se possibile) è disconnetterla. Se sono connessi via modem, si stacchi il doppino; se via Ethernet, il cavo Ethernet. Questo impedirà loro di fare altri danni e penseranno ad un problema di rete piuttosto che ad un rilevamento.

Se non si può staccare la rete (se si ha un sito trafficato o non si ha il controllo fisico sulle macchine), il prossimo miglior passo è usare qualcosa come i `tcp_wrappers` o `ipfwadm` per negare l'accesso dal sito dell'intruso.

Se non si possono rifiutare tutte le connessioni dal sito dell'intruso, ci si deve accontentare di disattivare l'account dell'utente. Farlo però non è semplice. Si devono tenere a mente i file `.rhosts`, l'accesso FTP e una montagna di possibili backdoor.

Una volta preso uno dei provvedimenti sopra (disconnessa la rete, negato l'accesso dal suo sito e/o disabilitato il suo account), si dovranno uccidere tutti i suoi processi ed espellerlo.

Si dovrà sorvegliare bene il 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.

È stata trovata una manomissione già avvenuta o è stato rilevato e chiuso fuori (si spera) l'intruso. E ora?

10.2.1. Chiudere il buco

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

Si controllino tutti i log, e si vedano le liste di sicurezza per controllare se ci sono exploit che si possono riparare. Si possono trovare 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. Si veda: http://www.debian.org/security/ per altre informazioni.

È 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@fer ret.lmh.ox.ac.uk Per iscriversi, mandate una e-mail a: security-audit-subscribe@ferret.lmh.ox.ac.uk"

Se non si chiudono fuori gli aggressori, probabilmente torneranno. Non solo sulla stessa macchina ma forse da qualche parte nella rete. Se avessero eseguito uno sniffer, ci sono buone probabilità che abbiano accesso ad altre macchine locali.

10.2.2. Stimare il danno

La prima cosa è stimare il danno. Cosa è stato compromesso? Se si stava eseguendo un test di integrità come `Tripwire`, lo si può usare per eseguire un controllo; dovrebbe aiutare a capire cosa è stato compromesso. Altrimenti si dovranno controllare tutti i dati importanti.

Visto che i sistemi Linux diventano sempre più semplici da installare si potrebbe considerare di salvare i file di configurazione, vuotare i dischi, reinstallare e quindi ripristinare i file degli utenti e di configurazione dai backup. Questo assicura un sistema nuovo e pulito. Se si dovessero ripristinare dei file dal sistema compromesso si sia molto cauti con ogni binario che viene ripristinato, perché potrebbe essere un cavallo di Troia messo lì dall'intruso.

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

Ora ci si deve preoccupare di quanto tempo è passato dalla compromissione e della possibilità che i backup possano 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, si potrebbero 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 si avrà ancora.

Si dovrebbero controllare backup anche molto vecchi, prima di ripristinare un file che è stato manomesso. L'intruso potrebbe aver compromesso i file molto tempo fa e potrebbero essere stati fatti molti backup del file già compromesso!

Ovviamente, c'è una serie di problemi di sicurezza con i backup. Ci si assicuri di tenerli in un posto sicuro. Si sappia chi vi ha accesso. (Se l'aggressore ottiene i backup, ha accesso a tutti i dati senza che neanche lo si venga a sapere.)

10.2.4. Rintracciare l'intruso.

Bene, è stato chiuso fuori l'intruso e ripristinato il sistema, ma non si è ancora finito. Anche se è improbabile che l'intruso venga preso, si dovrebbe notificare l'attacco.

Si dovrebbe notificarlo all'amministratore del sito da cui proveniva l'attacco. Si può trovare questo contatto con `whois` o sul database dell'Internic. Si potrebbe mandare una e-mail con tutte le voci, le date e le ore dei log inerenti. Se si è notato qualcosa di particolare nell'intruso lo si dovrebbe menzionare. Dopo aver mandato l'e-mail si dovrebbe, se si vuole, far seguire una telefonata. Se l'amministratore a sua volta vedesse l'aggressore potrebbe parlarne 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 si parla aiuta molto ad avere la loro collaborazione.

Si dovrebbe anche darne notizia ad ogni organizzazione di sicurezza di cui si fa parte (il CERT (http://www.cert.org/) o simili), oltre al distributore del sistema Linux.

11. Documenti sulla sicurezza

Esistono TANTISSIMI buoni siti per la sicurezza Unix in generale e quella Linux nello specifico. È 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. Riferimenti su LinuxSecurity.com

Il sito web di LinuxSecurity.com ha molti riferimenti alla sicurezza su Linux e l'open source, scritti dal loro staff e da altre persone da tutto il mondo.

- *Linux Advisory Watch (http://www.linuxsecurity.com/vuln-newsletter.html)* -- Una completa newsletter che riassume le vulnerabilità di sicurezza annunciate nel corso della settimana. Include riferimenti ai pacchetti aggiornati e descrizioni delle vulnerabilità.
- *Linux Security Week (http://www.linuxsecurity.com/newsletter.html)* -- Questo documento si propone di far avere ai lettori un sommario veloce dei più importanti problemi di sicurezza di ogni settimana.
- *Linux Security Discussion List (http://www.linuxsecurity.com/general/mailinglists.html)* -- Questa mailing list contiene domande e commenti generali sulla sicurezza.
- *Linux Security Newsletters (http://www.linuxsecurity.com/general/mailinglists.html)* -- Informazioni di sottoscrizione per tutte le newsletters.
- *FAQ di comp.os.linux.security (http://www.linuxsecurity.com/docs/colsfaq.html)* -- Domande ricorrenti (FAQ) con le risposte del newsgroup comp.os.linux.security.
- *Linux Security Documentation (http://www.linuxsecurity.com/docs/)* -- Un ottimo punto di partenza per avere informazioni relative alla sicurezza di Linux e dell'Open Source.

11.2. Siti FTP

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

ZEDZ (ex 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 `tue.nl` è un grande sito FTP Olandese sulla sicurezza. ftp.win.tue.nl (ftp://ftp.win.tue.nl/pub/security/)

11.3. Siti Web

- The Hacker FAQ è una FAQ sugli hacker: The Hacker FAQ (http://www.solon.com/~seebs/faqs/hacker.html)
- L'archivio COAST ha un gran numero di programmi e informazioni per la sicurezza di Unix: COAST (http://www.cs.purdue.edu/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 (http://www.netspace.org/lsv-archive/bugtraq.html)
- Il CERT, Computer Emergency Response Team, diffonde avvisi su comuni attacchi a piattaforme Unix: homepage del CERT (http://www.cert.org/)
- 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 (http://www.aoy.com/Linux/Security/)
- 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. È presso http://www.debian.com/security/.
- La WWW Security FAQ, scritta da Lincoln Stein, è un'ottima guida alla sicurezza del web. La si trova presso http://www.w3.org/Security/Faq/www-security-faq.html

11.4. Mailing lists

Bugtraq: Per iscriversi a bugtraq, si mandi una mail a listserv@netspace.org contenente nel corpo del messaggio subscribe bugtraq. (Si vedano gli archivi sopra).

CIAC: si mandi una e-mail a majordomo@tholia.llnl.gov. Mettere nel CORPO (non nel subject) del messaggio: subscribe ciac-bulletin

La Red Hat ha una serie di mailing list, la più importante delle quali è la lista redhat-announce. Si potrà leggere di patch di sicurezza (e altro) non appena sono disponibili. Si mandi una e-mail a redhat-announce-list-request@redhat.com con il subject Subscribe. Si veda https://listman.redhat.com/mailman/listinfo/ per altri informazioni e archivi.

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

11.5. 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 By D. Brent Chapman & Elizabeth D. Zwicky, 1ma Edizione Settembre 1995, ISBN: 1-56592-124-0
- Practical UNIX & Internet Security, 2nd Edition By Simson Garfinkel & Gene Spafford, 2nda Edizione Aprile 1996, ISBN: 1-56592-148-8
- Computer Security Basics By 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

Qui sotto si trovano alcuni dei termini usati più frequentemente a proposito della sicurezza dei computer. Un dizionario completo è disponibile presso LinuxSecurity.com (http://www.linuxsecurity.com/dictionary/)

- *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 sullo stack di una funzione 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:* Un computer per uso generico che ha almeno due interfacce di rete.
- *firewall:* Un componente o set di componenti che restringe l'accesso 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. È 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, numero 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. È più sicuro compilare il supporto dei driver direttamente nel kernel, invece che farne un modulo?

   Risposta: Alcune persone ritengono che sia meglio disabilitare il supporto dei moduli, perché un intruso potrebbe caricare un Cavallo di Troia o un altro modulo pericoloso per la sicurezza.

   Comunque, per caricare i moduli, si dovrebbe essere root. Anche i file dei moduli sono scrivibili solo da root. Questo significa che l'intruso avrebbe bisogno dell'accesso di root per inserire un modulo. Se l'intruso 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 dispositivo 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: Vedere la Sezione 4.2. Questo fatto è intenzionale per evitare che utenti remoti tentino di connettersi via `telnet` come `root` sulla macchina, il che è un serio rischio, perché la password di root sarebbe trasmessa in chiaro sulla rete. Non si dimentichi: i potenziali intrusi hanno il tempo dalla loro e possono eseguire programmi che trovino la password. In più, viene fatto per tenere traccia di chi è entrato nel sistema, non solo root.

3. Come abilito le shadow password sulla mia macchina Linux?

   Risposta:

   Per abilitare le shadow password, si esegua `pwconv` da root, quindi sarà creato `/etc/shadow` che verrà usato dalle applicazioni. Se si sta 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 si possono abilitare o disabilitare gli account senza che un utente sappia qual'è lo stato degli account degli altri utenti.

   Quindi il file `/etc/passwd` viene usato per registrare i nomi degli utenti e dei gruppi, usati da programmi come `/bin/ls` per risalire dall' 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 si vogliono rendere più sicure le password, forse si potrebbe essere anche interessati a generarne di sicure. Per questo si può usare il modulo `pam_cracklib` che fa parte dei PAM. Questo modulo prova le password contro le librerie di Crack per aiutare a decidere se sono troppo facili da trovare con programmi del genere.

4. Come posso abilitare le estensioni SSL di Apache?

   Risposta:

   a. Si prenda SSLeay 0.8.0 o successivo da <65533> (ftp://ftp.psy.uq.oz.au/pub/Crypto/SSL)
   b. Lo si compili e lo si provi,quindi lo si installi!
   c. Si prendano i sorgenti di Apache
   d. Si prendano le estensioni SSLeay di Apache da qui (ftp://ftp.ox.ac.uk/pub/crypto/SSL/)
   e. Le si scompattino nella directory dei sorgenti di apache e si aggiorni Apache seguendo il README
   f. Lo si configuri e compili.

   Si potrebbe anche provare su ZEDZ net (http://www.zedz.net) che ha molti pacchetti precompilati ed è collocata fuori dagli USA.

5. Come posso manipolare gli account, mantenendo la sicurezza?

   Risposta: quasi tutte le distribuzioni contengono un gran numero di strumenti per cambiare le proprietà degli account utente.

   - `Pwconv` e `unpwconv` possono essere usati per convertire da password normali a shadow e viceversa.
   - `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.

   Si leggano 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 (http://www.apacheweek.com), sbaglio?

   Si possono 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

Iscrivendosi alle mailing list di sicurezza e tenendosi aggiornati, si può fare molto per la sicurezza della propria macchina. Se si fa attenzione ai log e si esegue qualcosa tipo `tripwire` regolarmente, si può 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 (mailto:rob@DevilsThumb.com)
```

S. Coffin scoffin@netcom.com (mailto:scoffin@netcom.com)

Viktor Przebinda viktor@CRYSTAL.MATH.ou.edu (mailto:viktor@CRYSTAL.MATH.ou.edu)

Roelof Osinga roelof@eboa.com (mailto:roelof@eboa.com)

Kyle Hasselbacher kyle@carefree.quux.soltc.net (mailto:kyle@carefree.quux.soltec.net)

David S. Jackson dsj@dsj.net (mailto:dsj@dsj.net)

Todd G. Ruskell ruskell@boulder.nist.gov (mailto:ruskell@boulder.nist.gov)

Rogier Wolff R.E.Wolff@BitWizard.nl (mailto:R.E.Wolff@BitWizard.nl)

Antonomasia ant@notatla.demon.co.uk (mailto:ant@notatla.demon.co.uk)

Nic Bellamy sky@wibble.net (mailto:sky@wibble.net)

Eric Hanchrow offby1@blarg.net (mailto:offby1@blarg.net)

Robert J. Berger rberger@ibd.com (mailto:rberger@ibd.com)

Ulrich Alpers lurchi@cdrom.uni-stuttgart.de (mailto:lurchi@cdrom.uni-stuttgart.de)

David Noha dave@c-c-s.com (mailto:dave@c-c-s.com)

Pavel Epifanov epv@ibm.net (mailto:epv@ibm.net)

Joe Germuska joe@germuska.com (mailto:joe@germuska.com)

Franklin S. Werren fswerren@bagpipes.net (mailto:fswerren@bagpipes.net)

Paul Rusty Russell Paul.Russell@rustcorp.com.au (mailto:Paul.Russell@rustcorp.com.au)

Christine Gaunt cgaunt@umich.edu (mailto:cgaunt@umich.edu)

lin bhewitt@refmntutl01.afsc.noaa.gov (mailto:bhewitt@refmntutl01.afsc.noaa.gov)

A. Steinmetz astmail@yahoo.com (mailto:astmail@yahoo.com)

Jun Morimoto morimoto@xantia.citroen.org (mailto:morimoto@xantia.citroen.org)

Xiaotian Sun sunx@newton.me.berkeley.edu (mailto:sunx@newton.me.berkeley.edu)

Eric Hanchrow offby1@blarg.net (mailto:offby1@blarg.net)

Camille Begnis camille@mandrakesoft.com (mailto:camille@mandrakesoft.com)

Neil D neild@sympatico.ca (mailto:neild@sympatico.ca)

Michael Tandy Michael.Tandy@BTInternet.com (mailto:Michael.Tandy@BTInternet.com)

Tony Foiani tkil@scrye.com (mailto:tkil@scrye.com)

Matt Johnston mattj@flashmail.com (mailto:mattj@flashmail.com)

Geoff Billin gbillin@turbonet.com (mailto:gbillin@turbonet.com)

Hal Burgiss hburgiss@bellsouth.net (mailto:hburgiss@bellsouth.net)

Ian Macdonald ian@linuxcare.com (mailto:ian@linuxcare.com)

M. Kiesel m.kiesel@iname.com (mailto:m.kiesel@iname.com)

Mario Kratzer kratzer@mathematik.uni-marburg.de (mailto:kratzer@mathematik.uni-marburg.de)

Othmar Pasteka pasteka@kabsi.at (mailto:pasteka@kabsi.at)

Robert M rom@romab.com (mailto:rom@romab.com)

Cinnamon Lowe clowe@cinci.rr.com (mailto:clowe@cinci.rr.com)

Rob McMeekin blind_mordecai@yahoo.com (mailto:blind_mordecai@yahoo.com)

Gunnar Ritter g-r@bigfoot.de (mailto:g-r@bigfoot.de)

Frank Lichtenheldfrank@lichtenheld.de (mailto:frank@lichtenheld.de)

BjA¶rn Lotzblotz@suse.de (mailto:blotz@suse.de)

Othon Marcelo Nunes Batistaothonb@sup erig.com.br (mailto:othonb@superig.com.br)

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 (mailto:ziembor@FAQ-bot.ZiemBor.Waw.PL)

Giapponese: FUJIWARA Teruyoshi fjwr@mtj.biglobe.ne.jp (mailto:fjwr@mtj.biglobe.ne.jp)

Indonesiano: Tedi Heriyanto 22941219@students.ukdw.ac.id (mailto:22941219@students.ukdw.ac.id)

Coreano: Bume Chang Boxcar0001@aol.com (mailto:Boxcar0001@aol.com)

Spagnolo: Juan Carlos Fernandez piwiman@visionnetware.com (mailto:piwiman@visionnetware.com)

Olandese: "Nine Matthijssen" nine@matthijssen.nl (mailto:nine@matthijssen.nl)

Norvegese: ketil@vestby.com ketil@vestby.com (mailto:ketil@vestby.com)

Turco: tufan karadere tufank@metu.edu.tr (mailto:tufank@metu.edu.tr)

[NdT: Italiano: Elisabetta Galli lab@kkk.it (mailto:lab@kkk.it)]