

BLindoLinux

una guida rapida per rendere sicura la propria linux box

Ruggero Tonelli <ruggero@valtellinux.it>

Commenti e suggerimenti riguardanti questa pagina vanno mandati a Ruggero T. ruggero@valtellinux.it

introBLindo

Connettendo il nostro sistema alla rete, lo esponiamo ad una serie di potenziali attacchi informatici, che vanno dalla semplice installazione di una back-door al furto di dati importanti per la nostra azienda od altri DoS che possono recarci un danno, economico o di immagine che sia. *(vedi gli attacchi al DNS della Micro\$oft)*

Per evitare questi attacchi generalmente si installa un firewall (*software o hardware*) che ha il compito di filtrare i pacchetti che transitano attraverso la nostra rete decidendo, in base a regole stabilite dal SysAdmin, quali debbano passare o meno.

Una volta installato e configurato il firewall siamo pronti a resistere a determinati tipi di attacco, escludendo gli exploits ed altri attacchi più "subdoli" ma non sapremo mai chi e quando ha tentato di attaccare il nostro sistema, va quindi aggiunto un wrapper, il logging per i servizi messi a disposizione degli utenti ed altre utility, tutte cose che vedremo più avanti.

1. Installazione di una linux-box:

Un'installazione solida è un buon punto di partenza per qualsiasi sistema, soprattutto per quelli connessi alla rete. Prenderemo quindi in esame la creazione di una linux-box "blindata" a partire dalla sua installazione. Non entreremo nei particolari delle varie distribuzioni per rendere questa guida utile a tutti e non solo agli utenti Mandrake, Suse, Debian etc.. Quindi non verranno descritti i tool di configurazione particolari che una distribuzione o l'altra fornisce per facilitare la configurazione se non nelle "aggiunte".

1.1 Tipologia di installazione e partizionamento dei dischi.

Prima di tutto dobbiamo scegliere il tipo di installazione che vogliamo a seconda dell'utilizzo che intendiamo fare della nostra linux-box, ormai le distribuzioni principali prevedono l'installazione di gruppi di pacchetti come: workstation grafica, workstation con strumenti per l'ufficio, server altro, quindi se vogliamo una macchina per giocare e navigare in internet sarà meglio scegliere un'installazione di tipo workstation mentre se vogliamo installare un server web od un file-server e non vogliamo perderci nel decidere quali pacchetti installare, sceglieremo quella di tipo server.

Solitamente le varie distribuzioni hanno anche un tipo di installazione custom/expert che permette di variare a piacimento molti parametri come la dimensione e la tipologia delle partizioni, i servizi da caricare allo start-up della macchina ed i vari pacchetti da installare.

In fase di partizionamento (usando DiskDrake, diskdruid, fdisk o altri tools) creeremo partizioni separate per **/var** e **/home**, un'eventuale **/altra** per tenerci i dati più riservati, una **/backup** per i file di configurazione ed una **/chroot** per "ingabbiarci" i programmi chrooted.

Sceghieremo quindi come tipologia per tutte le partizioni **ReiserFS** che ci mette a disposizione il file-journaling che limiterà il down-time in caso di riavvio forzato della nostra linux-box, per **/backup** è meglio continuare ad usare **ext2**.

Si è scelto di tenere separate **/var** e **/home** dalla **root** per evitare di coinvolgerla in situazioni di overfilling e causare quindi il blocco del sistema.

Se il nostro sistema è destinato a servire diversi utenti ricordiamoci di assegnare ad ognuno di loro solo una porzione ben definita di spazio sul disco, questo per impedire che la crescita smisurata di un utente saturi lo spazio-disco necessario alla gestione del sistema stesso. Dovremo quindi ricordarci di abilitare l'opzione **quota** per le partizioni destinate allo spazio utente.

1.2 Scelta dei pacchetti e dei servizi:

Una volta partizionato il nostro sistema provvederemo a scegliere i pacchetti ed i servizi da installare, tenendo presente che: ogni cosa che installiamo e non usiamo è inutile, nonché potenziale fonte di exploit; ogni servizio che attiviamo e non usiamo è una fonte di guai: il mancato utilizzo porta ad una minore attenzione nei confronti delle eventuali anomalie che si potrebbero verificare.

Quindi se non useremo mai blender, xmms o gli screen-saver non installiamoli, il sistema sarà più snello e più facile da monitorare, se non utilizziamo telnet o rstat non montiamo questi servizi, stessa cosa vale (soprattutto) per i tool di amministrazione remota come web-admin. Se il sistema che stiamo installando è destinato a diventare un router/firewall a difesa di una lan ricordiamoci che la GUI grafica, i vari compilatori ed i player mp3 non servono, così come non ci servono apache e Zend.

2. Configurazione

Una volta terminata la nostra installazione dobbiamo configurare (*blindare*) il nostro sistema, per fare questo useremo Linuxconf (*DrakConf* , *YAST*, *Setup*) con il quale, tramite i vari menù setteremo servizi da attivare, permessi per utenti/gruppi ed altro. Dove non si può arrivare con i tools provvederemo a mano.

2.1 le password:

L'utente "pippo" con password "pippo" è, in italia, quello più gettonato per i test, nonché il primo digitato dallo sprovveduto che si trova di fronte alla richiesta di nome utente e password. Evitiamo quindi di usare "pippo" come password per l'utente root, così come: root, admin, test, mandrake, redhat, suse, joshua e simili. Per creare una password "sicura" o difficilmente individuabile anche usando algoritmi del tipo "brute-force" si dovrebbe scegliere un insieme di almeno 8 caratteri alfanumerici posti a caso es: 10voLTe10 <@S1Nò29 i99p0sSE 12M0\$FeT <i@0Baby. Per avere un insieme del genere facilmente memorizzabile usate le iniziali di una frase a voi familiare es.: a 12 anni ho iniziato a programmare in c : a12ahiapic
Le password da non usare si trovano negli elenchi usati dai principali password_finder, dategli un'occhiata!!

2.2 permessi ad utenti / gruppi:

All'inizio dovremo cancellare gli utenti inutili che vengono creati di default durante installazione legati ad uno specifico servizio o programma come anonymous e gopher:

```
[root @ bastion]# userdel adm
[root @ bastion]# userdel anonymous
[root @ bastion]# userdel lp
[root @ bastion]# userdel sync
[root @ bastion]# userdel shutdown
[root @ bastion]# userdel halt
[root @ bastion]# userdel news
[root @ bastion]# userdel uucp
[root @ bastion]# userdel operator
[root @ bastion]# userdel games (se non si usa X Window).
[root @ bastion]# userdel gopher
```

Cancelliamo anche i gruppi che non interessano:

```
[root @ bastion]# groupdel adm
[root @ bastion]# groupdel lp
[root @ bastion]# groupdel news
[root @ bastion]# groupdel dip
[root @ bastion]# groupdel slipusers
```

```
[root @ bastion]# groupdel popusers (se non si ha un server pop)
[root @ bastion]# groupdel pppusers (se non si hanno utenti che usano ppp)
[root @ bastion]# groupdel uucp
[root @ bastion]# groupdel games (se non si usa X Window).
```

Ricordiamoci anche di togliere i vari utenti/password "di prova" come guest/guest, pippo/pippo, admin/admin e test/test.

2.3 i servizi:

Sceghieremo ora, tra i servizi da caricare all'avvio del sistema, solo quelli che realmente servono al nostro sistema: un buon modo di operare è disattivare tutti i servizi indistintamente per poi andare ad attivarli in base alle nostre esigenze.

Qui sotto sono elencati buona parte dei servizi che non servono se non nelle condizioni descritte a fianco (indicati con il nome degli script per farli partire al boot) questi si trovano in /etc/rc.d/rc3.d o /etc/rc.d/rc5.d se si fa il boot in modo grafico.

Volendo fare le variazioni a mano basta rinominare il file interessato con una **s** (start) minuscola al posto della **S** maiuscola e questo non verrà più eseguito, per terminare il servizio si possono usare gli script con la **K** (kill) al posto della **S**.

Servizio	descrizione
S05apmd	gestione alimentazione / batterie solo per portatili
S10xntpd	distribuisce l'orario del nostro sistema in rete
S11portmap	richiesto se si fanno girare servizi rpc come NIS o NFS
S15sound	gestisce il sonoro della nostra linux-box
S15netfs	client nfs, usato per montare filesystem da un server nfs
S20rstatd	i servizi r forniscono informazioni sul sistema ad utenti remoti, si consiglia quindi di disattivarli tutti
S20usersd	"
S20rwhod	"
S20rwalld	"
S20bootparamd	client per sistemi diskless
S25squid	proxy server
S34yppasswdd	da attivare se la macchina è un server NIS, è uno tra i servizi più vulnerabili con quello che segue
S35ypserv	
S35dhcpcd	server dhcp
S40atd	servizio at simile a cron ma non richiesto dal sistema
S45pcmcia	demone per la gestione dei servizi pcmcia, solo per portatili
S50snmpd	demone SNMP, utile ma può distribuire informazioni preziose sul tuo sistema.
S55named	server DNS, se lo usi vai su http://www.isc.org/bind.html per ottenere l'ultima versione, è un servizio molto bersagliato, va tenuto costantemente aggiornato
S55routed	RIP, se non sai cos'è non ti serve
S60lpd	servizi di stampa, poco utile per un firewall-box
S60mars-nwe	server Netware, servizi file e stampa

S60nfs	server NFS
S72amd	demone AutoMount, usato per "montare" filesystem remoti
S75gated	da usare se si ha intenzione di usare altri protocolli di routing come OSPF
S80sendmail	se si disabilita questo servizio si potranno mandare e-mail ma non riceverle o farne il relay
S85httpd	webserver Apache, da tenere costantemente aggiornato: dai un'occhiata su http://www.apache.org
S87ypbind	se il sistema è un client NIS.
S90xfs	server xfont (se non si usa X non serve)
S95innd	server per news
S99linuxconf	usato per configurare la linux-box da remoto

2.4 il file /etc/aliases

Va editato commentando tutti gli alias tranne:

```
MAILER-DAEMON: postmaster
postmaster: root
bin: root
daemon: root
nobody: root
```

2.5 il file /etc/host.conf

Va editato e settato così:

```
#Aggiungere ,host alla fine per far tornare la ricerca degli host dal DNS in /etc/hosts:
order bind,hosts
#Se abbiamo macchine con indirizzi lpmultipli:
multi on
# Per controllare l'IP address spoofing:
nospoof on
```

2.6 il file /etc/exports

Se proprio dobbiamo condividere qualche directory tramite NFS, il file /etc/exports va settato nel modo piu' restrittivo possibile, non usando wildcard e specificando quali host dovranno aver accesso alle directory esportate:

```
/directory_da_condividere host.dominio.it(ro,root_squash)
```

dove ro sta per read-only e root_squash non permette nemmeno a root di scrivervi, lancia `/usr/sbin/exportfs -a` per attuare le modifiche.

2.7 il file /etc/login.defs

Cercate la riga `PASS_MIN_LEN` e mettete un valore superiore a 5 che solitamente è il default per obbligare gli eventuali utenti a scegliere una password lunga.

2.8 il file /etc/lilo.conf

Se vogliamo aumentare la sicurezza della nostra macchina aggiungeremo in `lilo.conf`:

```
restricted
```

```
password=
```

siccome la password è in chiaro dovrà essere vista solo da root quindi:

```
[root @ bastion]# chmod 600 /etc/lilo.conf
```

per attuare le modifiche e vedere i messaggi di ritorno in maniera esaustiva:

```
[root @ bastion]# lilo -v
```

e per maggior paranoia renderemo anche `lilo.conf` immutabile:

```
[root @ bastion]# chattr +i /etc/lilo.conf
```

2.9 il file /etc/services

Si occupa della conversione "nome servizio / numero porta". Solo root deve avere il permesso di leggerlo:

```
[root @ bastion]# chmod 600 /etc/services
```

e siccome capita di rado di doverci mettere mano lo renderemo immutabile:

```
[root @ bastion]# chattr +i /etc/services
```

2.10 le risorse utente

Se vogliamo evitare che un utente saturi le risorse di sistema dobbiamo aggiungere in `/etc/security/limits.conf`:

```
* hard core 0
```

```
# per non fargli creare file core di dimensione uguale a 0
```

```
* hard rss 5000
```

```
# per limitare la memoria a disposizione a 5Mb
```

```
* hard nproc 10
```

```
# per limitare il numero di processi lanciabili a 10
```

2.11 directory /etc/rc.d/init.d/

Anche qui bisogna assicurarsi che sono root abbia accesso agli script quindi digiteremo:

```
[root @ bastion]# chmod -R 700 /etc/rc.d/init.d/*
```

3 ipchains

Il compito di questa guida è quello di fornire gli strumenti basilari per poter configurare ipchains o iptables senza spiegarne alla perfezione le sue opzioni, per quello vi rimandiamo a : "IPCHAINS: Firewall a livello rete in Linux" dei Linux Knights e "Appunti Linux: Firewall secondo la gestione del kernel Linux 2.2.*" di Daniele Giacomini. Prenderemo quindi in esame due script: uno per isolare dall'esterno un sistema stand-alone ed uno per condividere la connessione ad internet da una lan. Le modifiche vanno fatte allo script caricato all'avvio del sistema per il firewalling (es: etc/rc.d/rc.firewall nelle distro Linux-Mandrake e Red Hat).

Questo script isola il nostro pc dall'esterno e ci permette di navigare in internet:

```
#-----
# (C) Lance Spitzner 2000
#-----
#!/bin/sh
#
:input DENY
:forward ACCEPT
:output ACCEPT
-A input -s 0.0.0.0/255.255.255.255 -d 0.0.0.0/0.0.0.0 -j DENY
-A input -s 0.0.0.0/0.0.0.0 -d 255.255.255.255/255.255.255.255 -j DENY
-A input -s 0.0.0.0/0.0.0.0 -d 224.0.0.0/255.0.0.0 -j DENY
-A input -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 -p 6 -j ACCEPT ! -y
-A input -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 -p 17 -j ACCEPT -l
-A input -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 -p 1 -j ACCEPT -l
-A input -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 -j DENY -l
```

Questo invece è un po' più complesso, permette un maggiore controllo sui servizi e le porte più delicate, l'accesso a servizi come icq e ftp, permette anche l'accesso alla nostra macchina dalla rete locale alla quale siamo collegati. Lo script è commentato in modo da permettere una facile configurazione on-the-road.

```
#!/bin/sh
# -----valtellinux.it--2-0-0-1-----
#
# CopyLeft valtellinux.it
#----- D-i-s-c-l-a-i-m-e-r -----
#
# Questo script è fornito come esempio di base per lo sviluppo di un
# firewall basato su ipchains. E' distribuito senza garanzia alcuna,
# implicita od esplicita, riguardo qualsiasi situazione derivante dal
# suo utilizzo.
#
#-----e-n-d---o-f---d-i-s-c-l-a-i-m-e-r-----
#
# RICORDA questo script logga in /etc/log/messages - /etc/log/syslog
#
```

```

# Le variabili in MAIUSCOLO vanno adeguate alla tua rete
#
DAINTERNO="eth0"
RETEINTERNA="192.168.1.0/24"
IPQUESTOBOX="192.168.1.1"
LOOPBACK="lo"
BCAST_S="0.0.0.0"
BCAST_D="255.255.255.255"
DAESTERNO="ppp0"
RETEESTERNA="0.0.0.0/0"
IPESTERNO="0.0.0.0"
#
# Localizza ipchains
IPCHAINS="/sbin/ipchains"
#
# -----S-T-A-R-T-----
## Annulla tutte le regole precedenti.
#
$IPOCHAINS -F
## Blocca tutti i transiti
$IPOCHAINS -P input DENY
$IPOCHAINS -P output REJECT
$IPOCHAINS -P forward REJECT
#-----altre-opzioni-e-fix-----v-t-x-
# Abilita IP Forwarding, se non è già stato fatto
echo 1 > /proc/sys/net/ipv4/ip_forward
# Abilita la protezione da TCP SYN Cookie
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
# Abilita la protezione per il defragging
echo 1 > /proc/sys/net/ipv4/ip_always_defrag
# Abilita la protezione per broadcast echo
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
## Così il sistema non risponde al ping
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
# Abilita la protezione per bad error message
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
# Abilita la protezione per IP spoofing
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
echo 1 > $f
done
# Disabilita accettazione ICMP Redirect
for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
echo 0 > $f
done
for f in /proc/sys/net/ipv4/conf/*/send_redirects; do
echo 0 > $f
done
# Disabilita i pacchetti Source Routed
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
echo 0 > $f
done
# Logga o pacchetti Spoofed, Source Routed e Redirect
for f in /proc/sys/net/ipv4/conf/*/log_martians; do
echo 1 > $f

```

```

done
# Moduli necessari per mascherare i relativi servizi
/sbin/modprobe ip_masq_ftp
# /sbin/modprobe ip_masq_vdolive.o
# /sbin/modprobe ip_masq_quake.o
# -----fine-altre-opzioni-----
#
## Permette le connessioni ad interfaccia LOOPBACK
#
$IPOCHAINS -A input -i $LOOPBACK -j ACCEPT
#
$IPOCHAINS -A output -i $LOOPBACK -j ACCEPT
#
## Abilita connessioni rete interna - firewall
#
ipchains -A input -i $DAINTERNO -s $RETEINTERNA -j ACCEPT
ipchains -A output -i $DAINTERNO -d $RETEINTERNA -j ACCEPT
#
## IP MASQUERADING #####
#
## Maschera tutti gli IP interni che vanno all'esterno
$IPOCHAINS -A forward -i $DAESTERNO -s $RETEINTERNA -j MASQ
#
##
# Rifiuta pacchetti 'spoofed' che pretendono di essere esterni
ipchains -A input -i $DAESTERNO -s $IPQUESTOBOX -j DENY -I
#
## Qui si specificano le porte da bloccare
## gli eventuali pacchetti bloccati sono loggati
## NetBEUI
$IPOCHAINS -A input -i $DAESTERNO -p tcp -s 0/0 -d 0/0 139 -I -j DENY
$IPOCHAINS -A input -i $DAESTERNO -p udp -s 0/0 -d 0/0 139 -I -j DENY
## MS-SQL
$IPOCHAINS -A input -i $DAESTERNO -p tcp -s 0/0 -d 0/0 1433 -I -j DENY
$IPOCHAINS -A input -i $DAESTERNO -p udp -s 0/0 -d 0/0 1433 -I -j DENY
## NFS
$IPOCHAINS -A input -i $DAESTERNO -p tcp -s 0/0 -d 0/0 2049 -I -j DENY
$IPOCHAINS -A input -i $DAESTERNO -p udp -s 0/0 -d 0/0 2049 -I -j DENY
## postgresSQL
$IPOCHAINS -A input -i $DAESTERNO -p tcp -s 0/0 -d 0/0 5432 -I -j DENY
$IPOCHAINS -A input -i $DAESTERNO -p udp -s 0/0 -d 0/0 5432 -I -j DENY
## Blocca le connessioni X11disp:0:-2-
$IPOCHAINS -A input -i $DAESTERNO -p tcp -s 0/0 -d 0/0 5999:6003 -I -j DENY
$IPOCHAINS -A input -i $DAESTERNO -p udp -s 0/0 -d 0/0 5999:6003 -I -j DENY
## SOCKS
$IPOCHAINS -A input -i $DAESTERNO -p tcp -s 0/0 -d 0/0 1080 -I -j DENY
$IPOCHAINS -A input -i $DAESTERNO -p udp -s 0/0 -d 0/0 1080 -I -j DENY
## Blocca Traceroute in entrata
$IPOCHAINS -A input -i $DAESTERNO -p udp \
-s 32769:65535 -d 33434:33523 -I -j DENY
## Apre Porte non privilegiate
#
$IPOCHAINS -A input -p tcp -s 0/0 -d 0/0 1023:65535 -j ACCEPT
$IPOCHAINS -A input -p udp -s 0/0 -d 0/0 1023:65535 -j ACCEPT

```

```

#
#
## SERVIZI DI BASE
# Se hai specificato i servizi in /etc/services puoi usare quelli
# senza specificare il numero della porta.
# ftp
$IPOCHAINS -A output -p tcp -s 0/0 -d 0/0 21 -j ACCEPT
# ssh
$IPOCHAINS -A input -p tcp -s 0/0 -d 0/0 22 -j DENY
# telnet
#$IPOCHAINS -A input -p tcp -s 0/0 -d 0/0 23 -j ACCEPT
# smtp
$IPOCHAINS -A output -p tcp -s 0/0 -d 0/0 25 -j ACCEPT
# DNS
$IPOCHAINS -A input -p tcp -s 0/0 -d 0/0 53 -j ACCEPT
$IPOCHAINS -A input -p udp -s 0/0 -d 0/0 53 -j ACCEPT
# http
$IPOCHAINS -A output -p tcp -s 0/0 -d 0/0 80 -j ACCEPT
# POP-3
$IPOCHAINS -A output -p tcp -s 0/0 -d 0/0 110 -j ACCEPT
# indented
#$IPOCHAINS -A input -p tcp -s 0/0 -d 0/0 113 -j ACCEPT
# https
#$IPOCHAINS -A input -p tcp -s 0/0 -d 0/0 443 -j ACCEPT
#
# IRC
#
$IPOCHAINS -A input -i $DAESTERNO -p tcp ! -y -s 0/0 6667 \
-d $IPQUESTOBOX 1024:65535 -j ACCEPT
$IPOCHAINS -A output -i $DAESTERNO -p tcp -s $IPQUESTOBOX 1024:65535 \
-d $ANYWHERE 6667 -j ACCEPT
## ICMP-----
#
#
# Abilita ICMP in entrata
$IPOCHAINS -A input -i $DAESTERNO -p icmp -s 0/0 -d 0/0 -j ACCEPT
$IPOCHAINS -A input -i $DAINTERNO -p icmp -s 0/0 -d 0/0 -j ACCEPT
# Abilita ICMP in uscita
$IPOCHAINS -A output -i $DAESTERNO -p icmp -s 0/0 -d 0/0 -j ACCEPT
$IPOCHAINS -A output -i $DAINTERNO -p icmp -s 0/0 -d 0/0 -j ACCEPT
#
##### REGOLE DI DEFAULT
# Se un pacchetto è sfuggito alle regole di sopra,
# viene filtrato dalle seguenti
##
$IPOCHAINS -A input -j DENY
$IPOCHAINS -A output -j ACCEPT
$IPOCHAINS -A forward -j DENYscarica lo script

```

Questo script, rielaborato partendo da altri proposti sul web, dovrebbe rappresentare un buon modello di partenza per "mettere in sicurezza" la propria rete. Se avete bisogno di un ulteriore aiuto per configurare ipchains, su <http://www.linux-firewall-tools.com/linux/> potrete creare on-line lo script che meglio si adatta alla vostra rete. Mr.Shark propone

nel suo QuickConfigHowTo uno script per ipchains molto interessante e commentato in italiano, dategli un'occhiata su <http://mrshark.sourceforge.net/qechowto/firewall.html>.

Qui potete trovare uno script creato da Kurd che permette di configurare iptables su si un sistema linux standard: <http://my.netfilter.se/>

Se avete suggerimenti per migliorare gli script e/o aggiungere nuove features scrivete, i commenti e le critiche sono ben accetti.

P.S. Ricordatevi di dare i permessi di esecuzione allo script `/etc/rc.d/rc.firewall...`)

4 inetd e TCP Wrappers

4.1 inetd

inetd è il demone supervisore dei servizi di rete, si mette in ascolto sulle porte stabilite ed avvia il demone del servizio corrispondente quando richiesto, permette inoltre di controllarne le richieste filtrandole attraverso wrapper come tcpd. I passi per "sistemare" questo demone sono sostanzialmente questi:

- cambiarne i permessi a 600: `[root @ bastion]# chmod 600 /etc/inet.conf`
- controllare che solo root vi abbia accesso: `[root @ bastion]# stat /etc/inet.conf`
- editare `/etc/inet.conf`, commentare tutte le linee (`#`), riavviare il demone: `[root @ bastion]# killall -HUP inetd`, ri-abilitare i servizi che ci servono, aggiungendo o meno un wrapper e riavviarlo di nuovo.
- rendere il file immutabile: `[root @ bastion]# chattr +i /etc/inet.conf`.

4.2 TPC che?

Il wrapper è un programma che si occupa di filtrare, in modo del tutto trasparente, le richieste fatte ad un qualsiasi servizio di rete attivo sul nostro server, loggandole (se specificato) attraverso il demone syslogd.

4.3 A cosa serve?

Il compito principale del wrapper consiste nel filtrare e monitorare gli accessi ad un particolare servizio. All'instaurazione della connessione il demone tcpd intercetta la chiamata al servizio e decide, qualora la richiesta non vada respinta, se instradarla verso il servizio richiesto, verso uno script o verso una trappola.

L'accesso può essere filtrato a seconda del servizio, dell'host che lo richiede o in base ad una combinazione delle due cose. Una configurazione accurata del demone previene anche attacchi del tipo "host address spoofing" e "host name spoofing".

4.4 Funzionamento e configurazione:

Il wrapper può essere usato in due modi: sostituendolo al demone del servizio che si vuole monitorare, evitando così di modificare le varie configurazioni; modificando il file di configurazione del superdemone inetd (`/etc/inetd.conf`) in base alla sintassi:

```
nome_servizio socket protocollo flag utente_demone /usr/bin/tcpd
demone_servizio_monitorato eventuale_altro_flag
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

Al momento della compilazione dei sorgenti ricordiamoci di specificare le seguenti opzioni:

- DPARANOID

per verificare se l'accoppiata ip - host della richiesta formulata risponde al vero, per prevenire attacchi del tipo host name spoofing.

- DKILL_IP_OPTIONS

per verificare che la macchina alla quale stiamo concedendo la connessione sia effettivamente quella desiderata. Con questa opzione attivata il demone rifiuterà le connessioni con il flag di source routing attivato, difendendoci così dagli attacchi "host address spoofing" (quest'opzione specificabile anche livello kernel).

- DPROCESS_OPTIONS

per far processare al demone i files di configurazione; /etc/host/allow e /etc/host/deny.

La sintassi dei files di configurazione del demone tcpd è molto semplice:

```
SERVIZIO: CLIENT [:EVENTUALE SCRIPT DA ESEGUIRE]
```

Dove per SERVIZIO si intende il nome del demone che sovrintende a quel determinato servizio, per CLIENT l'indirizzo ip del client (o il nome risolto) e lo SCRIPT di sistema ci permetterà di espandere le nostre possibilità di logging o di realizzare una trappola. Per indicare i servizi è possibile usare i caratteri jolly ALL e LOCAL, mentre per indicare una lista di client abbiamo ALL, LOCAL, KNOW, UNKNOW e PARANOID (-DPARANOID nelle opzioni di compilazione), EXCEPT. Per avere maggiori informazioni sui caratteri jolly bisogna leggere la man pages hosts_acces(5).

ESEMPI:

```
/etc/hosts.deny
```

```
ALL:ALL@ALL,PARANOID
```

Nega l'accesso a tutti i client e controlla che ip - nome.host corrispondano.

```
/etc/hosts.allow
```

```
ALL: LOCAL 192.168.1.0/255.255.255.0
```

Permette l'accesso a tutte le macchine appartenenti alla rete 192.168.1 mentre:

```
sshd: 207.24.156.1 secure.valtellinux.it
```

Permette l'accesso SSH all'host secure.valtellinux.it corrispondente all'IP 207.24.156.1.

`in.telnetd : ALL@ALL : spawn (/bin/mail -s "Connessione telnet da: %a %u" admin_mail) &`
Manda una mail all'indirizzo specificato `admin_mail` ogni qualvolta qualcuno si connette attraverso il servizio telnet, indicando l'indirizzo del client (`%a`) e l'utente (`%u`), la lista di questi parametri è contenuta nella man page `hosts_access(5)`.

`in.telnetd : ALL@ALL : /bin/script/logga_meglio.sh`

Si limita a chiamare lo script `logga_meglio` che provvederà nel nostro caso ad avviare programmi del tipo `who-is`, `nslookup` o simili per sapere chi ha usato la porta telnet del nostro sistema.

Per configurare al meglio il funzionamento del wrapper possiamo usare `tcpdchk` che ci permette di verificare la correttezza della sintassi nei files di configurazione o `tcpdmatch` che controlla il corretto funzionamento delle regole in essi contenute.

5 Paranoie varie:

5.1 L'aggiornamento... non è una paranoia!

Quando descritto nei paragrafi precedenti mette il nostro sistema al riparo dagli attacchi classici ma per evitare quelli più "cattivi", eseguiti tramite i vari exploit bisogna aggiornare continuamente i programmi a "contatto con la rete" come bind, apache, ipchains, ssh ed altri. Appena completata l'installazione e la configurazione di ipchains/iptables possiamo collegarci al sito della nostra distribuzione preferita per scaricare gli ultimi updates dei vari pacchetti, dando la priorità a quelli contrassegnati "Sicurezza / Urgente". Questi i link alle sezioni Security & Updates delle maggiori distro:

mandrake: <http://www.linux-mandrake.com/en/updates/>
redhat: <http://www.redhat.com/support/errata/>
caldera: <http://www.caldera.com/support/security/>
turbolinux: <http://www.turbolinux.com/security/>
debian: <http://www.debian.org/security/>
suse: <http://www.suse.de/en/support/security/>

Una volta completato anche questo passo possiamo finalmente considerare "sicura" la nostra linux-box e la rete che difende. Ricorda però: Quello che oggi è sicuro potrebbe non esserlo fra una settimana...

5.2 Via i compilatori!

Non installare i compilatori per evitare un'eventuale intruso possa compilare dall'esterno un altro kernel o un applicativo contenente backdoors o codice "maligno".

5.3 Iscrivere alle mailing-list!

Le mailing-list dei siti security-related come packet storm o insecurity.org sono un'ottima fonte di notizie sugli exploit e i possibili rimedi, tenersi aggiornati è importante.

5.4 logging "fisico"

Stampare i file di log è un ottimo sistema per controllare le intrusioni. Un attaccante che non voglia lasciare tracce modificherà sicuramente i nostri file di log, stampandoli non avremo più questo problema. Editiamo quindi /etc/syslog.conf ed aggiungiamo alla fine:
authpriv.*;mail.*;local7.*;auth.*;daemon.info /dev/lp0

per rendere effettive le modifiche:

```
[root @ bastion]# /etc/rc.d/init.d/syslog restart
```

5.5 Crakkati le password!

Un buon metodo per trovare le password deboli nel nostro sistema è quello di usare un programma come crack sul nostro file delle password. Ricordiamoci che, data la "delicatezza" del programma, deve essere di proprietà esclusiva dell'utente root!! Qui potrai trovare un mini how to sulle password ed il programma crack.

5.6 la password del BIOS

Se non abbiamo problemi di macchina soggetta a reboot, possiamo mettere una password d'avvio a livello bios, ricordandoci di settarla anche per l'opzione setup e che molti bios hanno password "universali", non usiamo questa password come unico sistema di difesa della nostra macchina.

6. riferimenti ed altro

6.1 riferimenti:

Paul Russell, Linux IPCHAINS-HOWTO · Terry Dawson, Linux NET-3-HOWTO, Linux Networking · Mark Grennan, Firewalling and Proxy Server HOWTO · Daniele Giacomini, Appunti di informatica libera · Linux Knights, Firewalls · Lance Spitzner, Armoring Linux.

6.2 Ringraziamenti

Per la stesura di questo documento si ringraziano: Lance Spitzner : l'idea di questo documento mi è venuta leggendo il suo "Armoring Linux", i Linux Knights per il loro scritto su Firewalls & TCP Wrappers, Ian Hall-Beyer e Robert L. Ziegler per gli script di configurazione di ipchains.

6.3 Disclaimer, Copyright & Left...

Questo documento è pubblicato sotto licenza GNU Free Documentation License, siete liberi di riprodurlo in ogni sua forma senza distorcerne il significato e menzionando l'autore che non va comunque ritenuto responsabile per eventuali danni recati dal contenuto di queste pagine. Gli script e le procedure descritte sono presentate come base per lo sviluppo di un personal/firewall e fornite "così-come-sono" senza alcuna garanzia implicita od esplicita circa il loro funzionamento o il risultato della loro applicazione. Commenti, consigli e critiche costruttive sono ben accetti. Tutti i marchi citati appartengono alle rispettive compagnie.

INDICE:

1. Installazione di una linux-box:

- 1.1 Tipologia di installazione e partizionamento dei dischi.
- 1.2 Scelta dei pacchetti e dei servizi:

2. Configurazione (tuning):

- 2.1 le password:
- 2.2 permessi ad utenti / gruppi:
- 2.3 i servizi:
- 2.4 il file /etc/aliases
- 2.5 il file /etc/host.conf
- 2.6 il file /etc/exports
- 2.7 il file /etc/login.defs
- 2.8 il file /etc/lilo.conf
- 2.9 il file /etc/services
- 2.10 le risorse utente
- 2.11 directory /etc/rc.d/init.d/

3 ipchains

4 inetd e TCP Wrappers

- 4.1 inetd
- 4.2 TPC che?
- 4.3 A cosa serve?
- 4.4 Funzionamento e configurazione:

5 Paranoie varie:

- 5.1 L'aggiornamento... non è una paranoia!
- 5.2 Via i compilatori!
- 5.3 Iscrivere alle mailing-list!
- 5.4 Logging "fisico"
- 5.5 Crakkati le password
- 5.6 La password del Bios

6 riferimenti ed altro

- 6.1 riferimenti
- 6.2 altro
- 6.3 disclaimer