Chroot di Tutti i Servizi in Linux

by Mark Nielsen (homepage)

About the author:
Mark lavora come consulente indipendente dedicandosi a cause come GNUJobs.com, scrivendo articoli, scrivendo software libero e lavorando come volontario a eastmont.net.

Abstract:

I servizi di sistema sotto chroot incrementano la sicurezza limitando il danno causato da qualcuno che riesca a entrare nel sistema.

Introduzione

Cos'è chroot? Chroot semplicemente ridefinisce l'universo di un programma. Più precisamente, ridefinisce la root directory o "/" di un programma o di una sessione di login. Praticamente tutto quello che sta al di fuori della directory su cui usate chroot non esiste agli occhi del programma o della shell.

A cosa serve? Se qualcuno riesce a entrare nel vostro computer, non sarà in grado di vedere tutti i files del vostro sistema. La limitazione sulla visibilità dei files limita anche i comandi che può eseguire e non dà la possibilità di sfruttare altre insicurezze in altri file. L'unico svantaggio è che credo non possa fermarli dal vedere le connessioni di rete o altre cose. Quindi dovreste fare alcune altre cose di cui non parleremo molto in questo articolo:

- Rendere sicure le porte di rete.
- Avere tutti i servizi che girano con un account non di root. Inoltre, avere tutti i servizi sotto chroot
- Mandare i log di syslog ad un altro computer.
- Analizzare i file di log
- Analizzare tentativi di controllo di porte casuali nel vostro computer
- Limitare le risorse di CPU e memoria per ogni servizio.
- Attivare gli account di quota.

Il motivo per cui considero chroot (su un servizio che non giri come root) una linea di difesa è che se qualcuno si impossessa di un account non di root e non trova nessun file da utilizzare per diventare root, i danni provocati vengono limitati solo all'area in cui riescono a penetrare. Inoltre, se l'area in cui riescono ad entrare appartiene per la maggior parte a root, hanno ancora minori possibilità di fare danni. Ovviamente c'è qualcosa che non va se qualcuno riesce a impossessarsi di un vostro account, ma è comunque utile sapere di poter limitare i danni.

RICORDATE che il mio modo di farlo potrebbe non essere perfetto al 100%. Questo è il mio primo tentativo in questo campo, e se riesce a funzionare almeno in parte, dovrebbe essere facile rifinire il lavoro. Questa è solo un'indicazione per un HOWTO che vorrei creare su chroot.

Come possiamo applicare chroot a tutto?

Creeremo una directory, "/chroot", e metteremo qui tutti i nostri servizi seguendo questo formato:

- Syslogd sarà messo in chroot con ogni servizio.
- Apache sarà sotto /chroot/httpd.
- Ssh sarà sotto /chroot/sshd.
- PostgreSQL sarà sotto /chroot/postmaster.
- Sendmail sarà messo in chroot, ma non girerà con un account non−root, sfortunatamente.
- ntpd sarà sotto /chroot/ntpd
- named sarà sotto /chroot/named

Ogni servizio dovrebbe essere completamente isolato

Il mio script in Perl per creare ambienti chroot.

Config_Chroot.pl.txt dovrebbe essere rinominato in Config_Chroot.pl dopo il download. Questo script in Perl vi permette di avere una lista dei servizi installati, vederne i file di configurazione, configurare un servizio, e farlo partire e fermare. In generale, questo è quello che dovreste fare:

1. Creare la directory chroot.
   mkdir −p /chroot/Config/Backup
2. Scaricare Config_Chroot.pl.txt come /chroot/Config_Chroot.pl
3. Cambiare la variabile $Home nello script nel caso non steste usando /chroot come home.
4. Scaricare i miei file di configurazione.

La cosa importante è che **Ho provato questa procedura solo su RedHat 7.2 e RedHat 6.2**.

Modificate lo script Perl in base alla vostra distribuzione.

Ho finito col fare un articolo immenso su Chroot, ma con questo script Perl è diventato molto più piccolo. Fondamentalmente ho notato, dopo aver messo in chroot molti servizi, che hanno tutti file e configurazioni molto simili da mettere sotto chroot. Il modo più semplice per capire quali files vadano copiati per un particolare servizio è di controllare la manpage e poi scrivere "ldd /usr/bin/file" per i programmi che usano librerie. Inoltre potete mettere in chroot il servizio che state installando e farlo partire manualmente per vedere gli errori che vi riporta o guardare il file di log.

In generale, per installare un servizio, fate così:

```
cd /chroot
./Config_Chroot.pl config  SERVICE
./Config_Chroot.pl install SERVICE
./Config_Chroot.pl start   SERVICE
```

Chroot di Ntpd

Ntpd è semplicemente un time server che mantiene il vostro computer e gli altri computer sincronizzati con l'ora reale. È stato semplice da mettere sotto chroot.

```
cd /chroot
 # Decommentate la prossima linea se non usate il mio file di configurazione
 #./Config_Chroot.pl config  ntpd
./Config_Chroot.pl install ntpd
./Config_Chroot.pl start   ntpd
```

Chroot di DNS o named

Già fatto, date un occhiata a
http://www.linuxdoc.org/HOWTO/Chroot−BIND8−HOWTO.html
o
http://www.linuxdoc.org/HOWTO/Chroot−BIND−HOWTO.html

O, se volete usare il mio script,

```
cd /chroot
 # Decommentate la prossima linea se non usate il mio file di configurazione
#./Config_Chroot.pl config  named
./Config_Chroot.pl install named
./Config_Chroot.pl start   named
```

Chroot di Syslog con i serizi e le mie lamentele.

Voglio usare chroot con syslogd. Il problema è che di default usa /dev/log, che non può essere visto dagli altri servizi in chroot. Perciò non posso usarlo con facilità. Queste sono le soluzioni possibili:

- Chroot di syslogd con ogni servizio. Ci ho provato e sono riuscito ad avere i log. Non mi piace perchè ho un servizio che gira come root.
- Vedere se ci si può collegare con un servizio di logging esterno.
- Fare logging semplicemente attraverso dei file e non attraverso syslogd. Probabilmente è la scelta più sicura, anche se, nel caso qualcuno riuscisse a penetrare, potrebbere mettere mano ai log.
- Configurare il syslogd principale per controllare diverse fonti per avere tutti i servizi. Si usa l'opzione −a di syslogd per ottenerlo.

La mia unica soluzione è stata di assicurarmi che syslogd fosse sotto chroot con ogni servizio. Mi piacerebbe trovare una soluzione che permetta di di fare log con un account non di root e che giri all'interno di un suo ambiente chroot, come per esempio sulla porta di rete. Probabilmente può essere realizzato, ma mi fermo dove sono e cercherò una soluzione migliore più avanti.

Se non volete usare in syslogd per ogni servizio potete aggiungere il seguente comando allo script di inizializzazione di syslogd che viene installato normalmente:

```
syslogd −a /chroot/SERVICE/dev/log
```

Avendo ssh e dns installati, sarebbe una cosa tipo

```
syslogd -a /chroot/ssh/dev/log -a /chroot/named/dev/log -a /dev/log
```

Un ultimo appunto su syslogd. mi poacerebbe farlo girare con un account non di root. Ho provato un paio di cose semplici, ma non ha funzionato e ci ho rinunciato. Se potesse girare come non−root con ogni servizio, questo soddisferebbe i miei dubbi di sicurezza. Possibilmente anche mandando i log all'esterno.

Chroot di Apache

Questo è stato estremamente facile. Una volta configurato, sono stato in grado di eseguire script in Perl. Il mio file di configurazione è abbastanza lungo perchè ho dovuto includere Perl e le librerie di PostgreSQL nell'area di chroot. Una cosa da tenere presente, se effettuate connessioni al database, è di assicurarsi che il database giri sul device di loopback 127.0.0.1 e di specificare l'host come 127.0.0.1 nei vostri script per il modulo DBI. Questo è un esempio di come connettersi a un database con connessioni persistenti in apache:

```
$dbh ||= DBI->connect('dbi:Pg:dbname=DATABASE',"","", {PrintError=>0});

if ($dbh ) {$dbh->{PrintError} = 1;}
else
  {$dbh ||= DBI->connect('dbi:Pg:dbname=DATABASE;host=127.0.0.1',"","",
      {PrintError=>1});}
```

Fonte: http://httpd.apache.org/dist/httpd/

Compilate e installate apache nel vostro sistema normalmente, sotto /usr/local/apache. Poi usate lo script Perl.

```
cd /chroot
 # Decommentate la prossima linea se non usate il mio file di configurazione
 # ./Config_Chroot.pl config  httpd
./Config_Chroot.pl install httpd
./Config_Chroot.pl start   httpd
```

Ho modificato il mio httpd.conf per includere quello che segue:

```
ExtendedStatus On

<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
</Location>

<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
</Location>
```

Quindi puntate il vostro browser a http://127.0.0.1/server−status o http://127.0.0.1/server−info e controllate!

Chroot di Ssh

Prima di tutto dovreste fare un forward della porta di ssh da 22 a 2222. A questo punto, quando fate partire ssh, mettetelo in ascolto sulla porta 2222, facendolo girare con un account non−root. Per la connessione iniziale in ssh vogliamo avere account sicuri con password per consentire alle persone di entrare, ma di non

fare altro. Una volta loggati, avranno a disposizione un secondo server ssh che ascolta su 127.0.0.1:2322 che gli consentirà di connettersi al sistema vero e proprio −− la seconda istanza di ssh dovrebbe stare in ascolto SOLO sul device di loopback. Questo è quello che dovreste fare. Ma non lo faremo adesso. L'unica cosa che faremo sarà di mettere sshd sotto chroot per questo esempio. Esercizi lasciati al lettore includono mettere ssh sotto un account non−root e installare un secondo sshd in ascolto sul device di loopback per consentire alle persone di entrare nel sistema vero e proprio.

Anche qui ci limiteremo a mettere sotto chroot ssh e vi lasceremo meditare sulle conseguenze di questa azione (non sarete in grado di vedere tutto il sistema se lo farete). Inoltre sarebbe interessante configurarlo per mandare i log all'esterno. Dovremmo usare OpenSSH, ma io sto usando SSH commerciale per semplicità (anche se non è una buona scusa).

Fonte: http://www.ssh.com/products/ssh/download.cfm

Installate ssh sotto /usr/local/ssh_chroot. Quindi usate lo script in Perl.

```
cd /chroot
 # Decommentate la prossima linea se non usate il mio file di configurazione
  # ./Config_Chroot.pl config  sshd
./Config_Chroot.pl install sshd
./Config_Chroot.pl start   sshd
```

Suppongo che una delle cose migliori che si ottengono mettendo ssh in un ambiente chroot è che se lo usate per sostituire un server ftp, le persone avranno un accesso limitato al vostro spazio. Rsync e SCP funzionano molto bene assieme per permettere alla gente di fare upload. Non mi piace proprio mettere un server ftp per permettere alla gente di entrarci. Molti server ftp possono essere messi in chroot, ma passano le password in chiaro, e a me non piace.

Chroot di PostgreSQL

Questo è stato semplice come apache, eccetto per il fatto che ha richiesto più librerie. Non è stato molto difficile. Una cosa che ho dovuto fare è stato aprire PostgreSQL alla rete, ma solo sul device di loopback. Poiché era sotto chroot, gli altri servizzi in chroot non potevano accedervi, come il web server apache. Ho compilato Perl all'interno di PostgreSQL, quindi ho dovuto aggiungere molte cose riguardanti perl al mio file di configurazione.

Fonte: ftp://ftp.us.postgresql.org/source/v7.1.3/postgresql−7.1.3.tar.gz

Compilate e installate postgres sul vostro sistema sotto /usr/local/postgres. Quindi usate lo script in Perl.

```
cd /chroot
 # Decommentate la prossima linea se non usate il mio file di configurazione
 # ./Config_Chroot.pl config  postgres
./Config_Chroot.pl install postgres
./Config_Chroot.pl start   postgres
```

Chroot di Sendmail

Lanciat semplicemente il mio script.

```
cd /chroot
 # Decommentate la prossima linea se non usate il mio file di configurazione
```

```
 # ./Config_Chroot.pl config  sendmail
./Config_Chroot.pl install sendmail
./Config_Chroot.pl start   sendmail
```

Ora, ci sono problemi? Sí. Sta ancora girando come root. Maledizione. Inoltre alcuni file vengono ricreati da /etc/rc.d/init.d/sendmail quando viene lanciato. Il mio script non li supporta. Quando fate delle modifiche a /etc/mail copiate il file anche in /chroot/sendmail/etc Inoltre dovrete creare un link da /var/spool/mail a /chroot/sendmail/var/spool/mail in modo che gli utenti che si loggano possano vedere gli stessi file.

La cosa buona è che potete ancora mandare posta verso l'esterno, il problema è riceverla. Inoltre, sono stato in grado di installare sendmail con apache senza problemi. Alcuni miei script in perl mandano posta verso l'esterno, quindi ho dovuto copiare i file di sendmail nell'area chroot di apache.

Altre cose da mettere in chroot.

Questa è la mia filosofia:

1. Tutto dovrebbe essere in chroot, inclusi sendmail, ssh, apache, postgresql, syslog e qualsiasi servizio giri sul computer.
2. Tutto dovrebbe girare con un account non−root (potreste dover fare un forward delle porte protette verso porte non protette, quindi oltre la 1024). Questo include sendmail e syslog, tra l'altro.
3. I log dovrebbero essere mandati verso l'esterno.
4. Si dovrebbe usare una partizione per ogni servizio per limitare lo spazio che un hacker potrebbe usare se decidesse di scrivere file. Potreste usare un device di loopback per montare dei file come filesystem nel caso non aveste partizioni disponibili.
5. Root dovrebbe essere proprietario di tutti i file che non devono essere modificati.

Ora è il turno di sendmail e syslogd. Penso ancora che dovrebbero girare sono un account non−root. per sendmail sarebbe possibile, ma l'hoi trovato estremamente difficile. Non sono riuscito a farlo girare con un account non−root, e penso sia un grosso errore non poterlo fare. So che ci sono problemi ad implementarlo, ma penso che TUTTI possano essere risolti. Per quanto riguarda i permessi dei file, non capisco perché sendmail debba girare come root. Potrebbero esserci motivi che sto sottovalutando, ma dubito che tutti gli ostacoli possano essere superati.

Per quanto riguarda syslog non ho nemmeno provato, ma direi che i log andrebbero scritti come utente non−root e non vedo perché non debba essere possibile. Almeno sono riuscito ad avere syslog sotto chroot per ogni servizio.

Tutti i servizi dovrebbero essere impostati per usare un account non−root. Anche NFS. Tutto.

Suggerimenti

- Usate il doppio login per ssh e tenere due daemon sshd.
- Trovate il sistema di usare sendmail o qualche altro server di posta come utente non−root.
- Togliete le librerie non necessarie sotto /lib. Io ho semplicemente copiato tutto per rendermi facile la vita. La maggior parte non vi serivranno.
- Fate logging remoto di syslogd e vedete se si riesce a mettere in ascolto syslogd su una porta di rete, convincendo i servizi a connettersi a tale porta sul device di loopback. Vedete se riuscite a far funzionare syslog come utente non−root.

Conclusioni

Penso che chroot sia ottimo per tutti i servizi. Credo sia un grave errore non usarlo per tutti i servizi che girino come non−root. Spero che una delle distribuzioni maggiori lo implementi, o anche una minore: QUALSIASI distribuzione. Mandrake è partita dalle basi di RedHat ed espandendosi sopra, quindi qualcuno potrebbe prendere Mandrake ed applicarci chroot sopra. Niente vieta a qualcuno di rifare il lavoro di altri sotto GNU/Linux, quindi credo sia possibile. Se qualche compagnia volesse applicare chroot a tutto e creare un ambiente sistematicamente facile per amministrare i servizi chrooted, avrebbe una distribuzione fantastica! Ricordate che, ora che Linux sta diventando un sistema di massa, la gente non vorrà più vedere la linea di comando, quindi se tutto venisse gestito dalla linea di comando, non dovrebbero sapere cosa sta succedendo sotto e a dire il vero non devono nemmeno saperlo −− devono solo essere in grado di configurare tutto e sapere che funziona!

Supporto al 100% l'idea che tutti i servizi dovrebbero essere sotto chroot e girare come non−root e che ogni distribuzione che non permetta questo non è adatta a me per essere usata in un ambiente di produzione. Io tendo ad applicare chroot a tutto, a più cose possibili −− prima o poi ci arriverò.

Ho intenzione di creare un HOWTO su chroot. Sto cercando aiuto da parte di qualcuno per convertire questo articolo nel formato di LyX in modo da poterlo inserire nell'HOWTO.

Riferimenti

1. Se questo articolo dovesse essere modificato, sarà disponibile qui http://www.gnujobs.com/Articles/23/chroot.html

---

Webpages maintained by the LinuxFocus Editor team © Mark Nielsen "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org	Translation information: en −−> −− : Mark Nielsen (homepage) en −−> it: Alessandro Pellizzari <alex/at/neko.it>

2005−01−10, generated by lfparser_pdf version 2.51