LinuxFocus article number 225
http://linuxfocus.org

door Mark Nielsen (homepage)

Over de auteur:
Mark is een onafhankelijke consultant die tijd vrijmaakt voor zaken als GNUJobs.com. Hij schrijft artikelen en gratis software en werkt als vrijwilliger bij eastmont.net.

Vertaald naar het Nederlands door:
Hendrik-Jan Heins
<hjh/at/passys.nl>

Alle diensten in Linux onder Chroot draaien

Kort:

Ge-chroote systeemdiensten verbeteren de veiligheid door de schade te beperken die een inbreker kan aanrichten.

Inleiding

Wat is chroot? Chroot is in principe een herdefinitie van de omgeving voor een programma. Om exacter te zijn herdefinieert chroot de ’ROOT’ directory (ook bekend als ’/’ voor een programma of login sessie. Hierdoor bestaat er voor het programma buiten de directory waar je chroot op hebt toegepast niets.

Warom is dit nuttig? Als iemand inbreekt op je computer, dan kan hij niet alle bestanden in jou systeem zien. En doordat ze niet alle bestanden in je systeem kunnen zien, is de keuze van de uit te voeren commando’s beperkt en kunnen ze ook geen andere onbeveiligde bestanden misbruiken. Het enige nadeel is volgens mij dat de aanvaller niet tegengehouden wordt om andere netwerkverbindingen en dergelijke aan te vallen. Dus, als je chroot gebruikt zal je toch ook nog een paar andere zaken moeten beveiligen, maar daar gaan we in dit artikel niet te veel op in:

- Beveilig je netwerkpoorten.
- Zorg ervoor dat al je diensten onder een andere dan de root naam draaien. En zorg er bovendien voor dat alle diensten onder chroot lopen.
- Stuur logbestanden van je systeem door naar een andere computer.
- Analyseer je logbestanden!!!

- Hou die mensen in de gaten die op goed geluk poorten scannen op je computer.
- Beperk het maximum cpu- en geheugengebruik voor een dienst.
- Activeer account quota's.

De reden dat ik chroot (met een non-root dienst) zie als een verdedigingslinie is de volgende: Wanneer iemand inbreekt onder een andere dan de root account, en hij kan de root-account zelf niet kraken, dan kan hij slechts beperkte schade aanrichten in het gebied waarin hij inbreekt. Bovendien is het gebied waarin ze inbreken meestal eigendom van de root, dus hebben ze minder mogelijkheden om een aanval uit te voeren. Er is duidelijk iets mis als iemand echt inbreekt op je account, maar het is goed dat je de schade die ze kunnen aanrichten kan beperken.

Onthoud alstjeblieft dat de manier waarop ik dit doe waarschijnlijk niet 100% accuraat is. Dit is mijn eerste poging om dit te doen en als het slechts gedeeltelijk werkt, dan moet het niet al te moeilijk zijn om de scherpe kantjes eraf te halen. Dit is slechts een vingerwijzing voor een HOWTO die ik wil schrijven over chroot.

Hoe gaan we alles chrooten?

Nou, we maken een directory "/chroot" en daar gaan we alle diensten onder het volgende formaat in stoppen:

- Syslogd zal bij iedere dienst worden ge-chroot.
- Apache zal worden geplaatst op /chroot/httpd.
- Ssh zal zitten op /chroot/sshd.
- PostgreSQL op /chroot/postmaster.
- Sendmail zal worden ge-chroot, maar hij zal helaas niet draaien onder een niet-root account.
- Ntpd wordt ge-chroot op /chroot/ntpd .
- Named op /chroot/named .

Iedere dienst moet compleet geïsoleerd worden.

Mijn Perl script om ge-chroote omgevingen te maken

Config_Chroot.pl.txt deze moet na het downloden worden hernoemd tot Config_Chroot.pl. Dit Perl script laat je toe om de geïnstalleerde diensten als lijst te bekijken, de configuratie bestanden te zien, de dienst te configureren en te starten en te stoppen. Dit is in grote lijnen wat je moet doen.

1. Maak de chroot directory.
   mkdir -p /chroot/Config/Backup
2. Download Config_Chroot.pl.txt naar /chroot/Config_Chroot.pl
3. Verander de $Home variabele in het Perl script als je /chroot niet als home directory gebruikt.
4. Download mijn config files.

Let op, belangrijk om te weten: **Ik heb alleen getest op RedHat 7.2 en RedHat 6.2**.

Pas het Perl script aan voor je eigen distributie.

Uiteindelijk heb ik een heel groot artikel over chroot geschreven, maar mijn Perl script is veel kleiner

geworden. In principe viel me na het chrooten op dat de diensten configuratie-bestanden hebben die erg veel op elkaar lijken en dat die chroot nodig hebben. De eenvoudigste manier om uit te vinden welke bestanden er voor een bepaalde dienst moeten worden gekopieerd is het lezen van de man pagina's, typ ook 'ldd /usr/bin/file' voor programma's die gebruik maken van bibliotheken. Je kan ook de dienst die je installeert handmatig starten om te zien welke fouten je krijgt en tenslotte kan je ook naar de log bestanden kijken.

Over het algemeen moet je het volgende doen om een dienst te installeren:

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

Ntpd chrooten

Ntpd is slechts een tijdservice die jouw en andere computers synchroon en op tijd laat lopen. Dit is een eenvoudige dienst om te chrooten.

```
cd /chroot
 # Uncomment the next line if you don't use my config file.
#./Config_Chroot.pl config  ntpd
./Config_Chroot.pl install ntpd
./Config_Chroot.pl start   ntpd
```

Het Chrooten van DNS of named

Is al gedaan, kijk maar eens hier
http://www.linuxdoc.org/HOWTO/Chroot-BIND8-HOWTO.html
or
http://www.linuxdoc.org/HOWTO/Chroot-BIND-HOWTO.html

Of, als je mijn script wilt gebruiken,

```
cd /chroot
 # Uncomment the next line if you don't use my config file.
#./Config_Chroot.pl config  named
./Config_Chroot.pl install named
./Config_Chroot.pl start   named
```

Het chrooten van Syslog bij diensten en mijn klachten

Ik wil syslogd chrooten. Mijn probleem hierbij is dat syslogd stndaard gebruik maakt van /dev/log, dat niet kan gezien worden door ge-chroote diensten. Hier zijn enkele mogelijke oplossingen:

- Syslogd chrooten bij iedere dienst. Ik heb dit echt getest en ja, ik kon dingen loggen. Ik vind dit echter geen goede oplossing, aangezien ik nu een dienst heb die root draait.
- Controleren of we kunnen verbinden met een log faciliteit op een andere computer.
- Gewoon log bestanden schrijven naar een bestand en niet via syslogd is waarschijnlijk de veiligste optie, echter als iemand dan toch weet in te breken, kan hij de log bestanden bekijken.
- Syslogd zo configureren dat hij op verschillende plaatsen kijkt om alle diensten te vinden. Je moet de -a optie met syslogd gebruiken om dit te kunnen doen.

Mijn enige echte oplossing was me ervan te verzekeren dat syslogd bij iedere dienst ge-chroot werd. Ik zou graag een oplossing hebben waarbij alles op een niet-root account werd gelogd en er gebruik zou worden gemaakt van de eigen ge-chroote omgeving, zoals bijvoorbeeld een netwerkpoort. Dit is waarschijnlijk wel mogelijk, maar ik laat het nu even zoals het is en ik verzin op een ander tijdstip wel een betere oplossing.

Als je geen losse syslogd voor iedere dienst aan wilt maken, maar wilt werken met de ’standaard’ syslogd op je systeem, dan moet je het volgende commando toevoegen wanneer syslogd start:

```
syslogd -a /chroot/DIENST/dev/log
```

Als ssh en dns bij mij zouden draaien, dan zou het er zo uit kunnen zien

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

Laatste opmerking over syslogd: Ik wou dat ik het kon laten draaien onder een niet-root account. Ik heb een paar eenvoudige dingen geprobeerd maar het werkte niet en ik heb het opgegeven. Als ik syslogd onder een andere dan de root account kon laten draaien met iedere dienst, dan zouden mijn veiligheidsproblemen opgelost zijn. Mogelijk zou hij dan zelfs naar een andere machine kunnen loggen.

Apache chrooten

Dit was zeer eenvoudig om te doen. Zodra ik hem opgezet had, kon ik Perl scripts utvoeren. Nu is mijn configuratiebestand vrij lang doordat ik de Perl en PostgresSQL bibliotheken in het ge-chroote gebied moest plaatsen. Je moet op één ding letten als je een connectie naar een databankt maakt: zorg ervoor dat je database draait op 127.0.0.1 (het loopback apparaat) en dat je de gastheer hebt aangegeven op 127.0.0.1 in je Perl scripts voor de DBI module. Hier is een voorbeeld van hoe ik een database gekoppeld heb door gebruik te maken van continue connecties (persisten connections) 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});}
```

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

Compileer en installeer Apache op je hoofdsysteem op /usr/local/apache. En gebruik daarna het perl script.

```
cd /chroot
```

```
 # Uncomment the next line if you don’t use my config file.
 # ./Config_Chroot.pl config  httpd
./Config_Chroot.pl install httpd
./Config_Chroot.pl start   httpd
```

Ik heb mijn httpd.conf bestand veranderd om dit te krijgen:

```
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>
```

Daarna hoef je de browser alleen maar te laten wijzen naar http://127.0.0.1/server-status of http://127.0.0.1/server-info en zie maar!

Ssh chrooten

Om te beginnen zou je in het ideale geval poort 22 door moeten verbinden naar poort 2222. En dan, als je ssh start, zou deze moeten luisteren op poort 2222 onder een andere dan de root account. Voor de initiële ssh verbinding willen we beveiligde accounts en wachtwoorden hebben die enkel dienen om mensen binnen te laten en verder niets doen. Nadat ze ingelogd zijn hebben ze een tweede ssh programma draaien op poort 127.0.0.1:2322 en deze verbindt ze met het echte systeem -- het tweede ssh programma moet dus alleen luisteren naar het loopback apparaat. Dit is wat je moet doen, maar we doen het deze keer niet. Het enige dat we gaan doen is ssh chrooten voor dit voorbeeld. Oefeningen die de lezer zelf moet uitvoeren zijn onder andere het plaatsen van ssh onder een andere dan de root account en het installeren van een tweede sshd die luistert op het loopback apparaat om mensen het systeem in te laten.

We gaan opnieuw alleen ssh chrooten en we laten je zelf verzinnen wat de consequenties daarvan zijn (je zal niet je complete systeem kunnen zien als je dit doet). Bovendien zou het in het ideale geval aardig zijn om dit op te zetten naar bestandslogs op andere machines. En je zou Openssh moeten gebruiken, ik gebruik het commerciële ssh omdat dat eenvoudiger is (maar dat is geen goed excuus).

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

Installeer ssh op /usr/local/ssh_chroot. En gebruik daarna het Perl script.

```
cd /chroot
 # Uncomment the next line if you don’t use my config file.
 # ./Config_Chroot.pl config  sshd
./Config_Chroot.pl install sshd
./Config_Chroot.pl start   sshd
```

Ik denk dat een van de betere aspecten van het draaien van ssh onder een ge-chroote omgeving is dat als je er gebruikt van wilt maken als vervanging voor een ftp server, mensen slechts gelimiteerde toegang hebben tot jou machine. Rsync en SCP werken zeer goed samen om mensen bestanden te laten uploaden. Ik hou er niet echt van om een ftp dienst te starten waar mensen op kunnen inloggen. Veel ftp servers draaien ook ge-chroot, maar ze verzenden nog steeds ongecodeerde wachtwoorden en daar houd ik niet van.

PostgreSQL chrooten

Dit was bijna zo eenvoudig als Perl, behalve dat het een paar bibliotheken meer nodig had. Over het geheel genomen was het niet moeilijk om te doen. Eén van de dingen die ik moest doen was PostgreSQL open zetten naar het netwerk, maar alleen op het loopback apparaat. Aangezien hij was ge-chroot, konden andere diensten er niet bij, zoals bij de webserver Apache. Ik heb Perl mee gecompileerd in PostgreSQL, dus ik moest een boel Perl spul toevoegen in mijn configuratiebestand.

Bron: ftp://ftp.us.postgresql.org/source/v7.1.3/postgresql-7.1.3.tar.gz

Compileer en installeer Apache op je basissysteem in /usr/local/postgres. Gebruik daarna het Perl script.

```
cd /chroot
 # Uncomment the next line if you don't use my config file.
 # ./Config_Chroot.pl config  postgres
./Config_Chroot.pl install postgres
./Config_Chroot.pl start   postgres
```

Sendmail chrooten

Ga je gang en voer mijn script maar uit.

```
cd /chroot
 # Uncomment the next line if you don't use my config file.
 # ./Config_Chroot.pl config  sendmail
./Config_Chroot.pl install sendmail
./Config_Chroot.pl start   sendmail
```

Ziten er nu nog haken en ogen aan? Ja. Hij draait nog steeds als root. Balen! Bovendien worden enkele bestanden aangemaakt door /etc/rc.d/init.d/sendmail wanneer hij gestart wordt. Mijn script doet hier niets aan. Kopieer de veranderingen die je aanbrengt in /etc/mail, dan ook naar /chroot/sendmail/etc. Je moet bovendien /var/spool/mail verwijzen naar /chroot/sendmail/var/spool/mail zodat het sendmail programma en de gebruikers (als ze inloggen) dezelfde bestande kunnen zien.

Het positieve is dat je altijd post kan verzenden, het is het ontvangen dat problemen geeft. Uiteindelijk kon ik sendmail installeren met Apache zonder enige problemen. Enkele van mijn Perl scripts verzenden mail, en ik had dus een kopie van de de sendmail bestanden nodig in het chroot gebied van Apache

Andere programma's onder chroot.

Dit is mijn filosofie:

1. Alles zou onder chroot moeten draaien, inclusief sendmail, ssh, Apache, PostgreSQL, syslogd en iedere andere dienst die draait op de computer.
2. Alles zou onder een andere dan de root account moeten worden geplaatst (je moet misschien een beveiligde poort doorsturen naar een niet beveiligde poort). Dit is inclusief sendmail en syslogd trouwens.
3. Logs moeten naar een andere machine worden gestuurd.
4. Een partitie zou moeten worden opgezet voor iedere dienst zodat er een beperking is aan de hoeveelheid schijfruimte die een hacker kan gebruiken als hij besluit bestanden weg te schrijven. Als je te weinig partities hebt zou je een loopback apparaat kunnen gebruiken om bestanden als een bestandssysteem te mounten voor sommige diensten.
5. De root zou eigenaar moeten zijn van alle bestanden die niet veranderen.

Nu, als het gaat om sendmail en syslogd, denk ik nog steeds dat ze zouden moeten draaien onder een andere dan de root acount. Voor sendmail zou dit mogelijk moeten zijn, maar ik heb ondervonden dat het zeer lastig is. Tot nu toe ben ik niet succesvol geweest in mijn pogingen met sendmail en ik meen dat het een echte fout zou zijn als het niet mogelijk zou zijn. Ik weet dat er problemen zijn om dat voor elkaar te krijgen, maar ik denk dat ze allemaal op te lossen zijn. Zolang de bestandstoegang correct wordt ingesteld zie ik niet in waarom sendmail zou moeten draaien onder een root account. Er kan iets zijn wat ik over het hoofd zie, maar ik betwijfel of er echte onoverkomelijke obstakels zijn.

Voor syslog heb ik het niet eens geprobeerd, maar ik zou zeggen dat logs gelogd dienen te worden onder een andere dan de root account en ik zie niet in waarom dat niet mogelijk zou zijn. Ik kon in ieder geval syslogd onder chroot draaien voor iedere dienst.

Alle diensten zouden moeten worden gedaaid onder een andere dan de root account. Zélfs NFS. Alles!

Suggesties

- Gebruik een dubbele login voor ssh en draai twee verschillende ssh daemons.
- Zoek uit hoe je sendmail of een ander e-mail programma onder een andere dan de root account draaiende krijgt.
- Haal alle ongebruikte bibliotheken weg uit /lib. Ik heb gewoon alles gekopieerd om het mezelf makkelijker te maken. De meesten heb je toch niet nodig.
- Laat syslogd de logs op een andere machine zetten en zoek uit of we syslogd aan een netwerkpoort kunnen hangen en op die manier alle diensten uit een netwerk kunnen aansluiten op het loopback apparaat. Kijk of we syslogd draaiende kunnen krijgen onder een andere dan de root account.

Conclusie

Ik denk dat chroot een goede oplossing is voor alle diensten. Ik meen dat het een grote fout is om niet alle diensten onder chroot en een andere dan de root account te draaien. Ik zou willen dat een grote distributie dit gaat doen, of een kleine; welke distributie dan ook! Mandrake is ooit begonnen door dingen van RedHat te gebruiken en te verbeteren, dus misschien kan iemand nu beginnen met Mandrake en dat uitbouwen met chroot. Niets weerhoudt mensen ervan om werk dat anderen al gedaan hebben opnieuw te doen onder GNU/Linux, dus ik denk dat het mogelijk is. Als er een bedrijf zou zijn dat alles

wil draaien onder chroot en een systematische omgeving zou creëren om mensen op een eenvoudige manier hun diensten te laten beheren, dan zouden ze een fantastische distributie hebben! Onthoud nu dat Linux mainstream wordt, mensen willen geen commandoregel meer zien, dus alles moet worden gedaan op grafisch niveau, ze hoeven de onderbuik niet te zien en ze hoeven dan ook niet te weten wat daar alemaal gebeurt -- ze moeten het alleen maar kunnen configureren en te weten dat het werkt!

Ik ben er 100% van overtuigd dat alle diensten zouden moeten draaien onder chroot met een andere dan de root account en iedere distributie die dat niet doet is wat mij betreft niet zuiver op de graat en dus niet bruikbaar als productie-omgeving. Ik ga alles onder chroot draaien, zoveel mogelijk -- uiteindelijk zal het me lukken.

Ik heb het idee om een HOWTO over chrooten te schrijven. Ik heb een verzoek ingediend om iemand te vinden die me kan helpen bij het omzetten van dit artikel naar LyX formaat zodat het in de Linux HOwTOs gezet kan worden.

Refereties

1. Als dit artikel verandert dan kan het hier gevonden worden
   http://www.gnujobs.com/Articles/23/chroot.html

Site onderhouden door het LinuxFocus editors team © Mark Nielsen "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org	Vertaling info: en --> -- : Mark Nielsen (homepage) en --> nl: Hendrik-Jan Heins <hjh/at/passys.nl>

2005-01-14, generated by lfparser_pdf version 2.51