Chrooting todos os serviços do Linux

by Mark Nielsen (homepage)

About the author:
O Mark trabalha como um consultor independente doando o seu tempo a causas como a GNUJobs.com, escrevendo artigos, escrevendo software livre e, trabalhando como voluntário em eastmont.net.

Abstract:

Os sistemas com serviços chrooted melhoram a segurança limitando o estrago que alguém que entra dentro do sistema pode fazer.

Introdução

O que é que é o chroot? O Chroot basicamente redefine o universo de um programa. Mais correctamente, redefine o directório "ROOT" ou "/" de uma programa ou de uma sessão de login. Basicamente, tudo fora do directório onde aplica o chroot não existe respeitante à shell ou a um programa.

Porque é que isto é útil? Se alguém entrar dentro do seu computador, eles não serão capazes de ver todos os ficheiros no seu sistema. Não sendo capazes de ver os seus ficheiros são limitados os comandos que podem introduzir e não lhes dá a possibilidade de explorar outros ficheiros que estão inseguros. O único senão é que, creio eu, não os impede de olhar para as ligações de rede e outras coisas do género. Assim deve querer fazer mais algumas coisas que não aprofundizaremos muito neste artigo:

- Securizar os seus portos de rede.
- Ter todos os serviços a correr sob uma conta não root. Adicionalmente, ter todos os serviços chrooted.
- Rencaminhar os syslogs para outra máquina.
- Analisar os ficheiros de log
- Analisar as pessoas a tentar detectar portas aleatórias no seu computador
- Limitar os recursos de cpu e memória de um serviço.
- Activar as cotas das contas.

A razão por que considero o chroot (com um serviço não−root) uma linha defensiva é que se alguém "fura" numa conta não root e não existem ficheiros os quais podem utilizar para entrar como root, assim os estragos estão limitados à área em que conseguiram entrar. Também se a área que eles furam pertence praticamente à conta root, eles têm menos opções para o ataque consequentemente. Obviamente, que existe algo errado se alguém consegue furar a sua conta, mas é bom limitar os estragos que eles podem fazer.

Por favor lembre−se que o meu modo de fazer as coisas provavelmente não é 100% correcto. Esta é a minha primeira tentativa de fazer as coisas e se só funcionar parcialmente bem, deve ser fácil terminar a configuração. Isto é somente uma mapa para um HOWTO que eu desejo escrever acerca do chroot.

Como é que vamos aplicar o chroot a tudo ?

Bem, criamos um directório, "/chroot" e pomos lá todos os nossos serviços no seguinte formato:

- O Syslogd será chrooted em cada serviço.
- O Apache estará em /chroot/httpd.
- O Ssh estará em /chroot/sshd.
- O PostgreSQL estará em /chroot/postmaster.
- O Sendmail será chrooted, mas não correrá sob uma conta não root, infelizmente.
- O ntpd será chrooted para /chroot/ntpd
- O named será chrooted para /chroot/named

Cada serviço devia estar completamente isolado.

A minha script Perl para criar ambientes chrooted.

O ficheiro Config_Chroot.pl.txt deve ser renomeado para Config_Chroot.pl após ter feito o seu download. Esta script perl permite−lhe ver os serviços que tem instalados, ver os ficheiros de configuração, configurar um serviço e iniciar e parar os serviços. De um modo geral isto é o que deve fazer.

1. Crie o directório chroot.
   mkdir −p /chroot/Config/Backup
2. Faça download de Config_Chroot.pl.txt para /chroot/Config_Chroot.pl
3. Altere a variável $Home na script de perl se não está a utilizar o directório /chroot como directório de trabalho.
4. Faça download dos meus ficheiros de configuração.

Agora, existe uma coisa importante: **Só testei no RedHat 7.2 e no RedHat 6.2**.

Modifique a script em perl para a sua distribuição.

Quase que me encontrei a escrever um artigo gigantesco acerca do Chroot mas a script em Perl reduziu−o bastante. Basicamente, notei que após ter feito um chroot a muitos serviços que eles tinham ficheiros muito semelhantes e configurações que precisavam de ser chrooted. A maneira mais fácil de nos apercebermos quais os ficheiros que precisam de ser copiados para um determinado serviço é olhar para as páginas manual e também digitando "ldd /usr/bin/file" para os programas que utilizam ficheiros de bibliotecas. Também pode fazer um chroot ao serviço que está a instalar e iniciá−lo manualmente vendo os erros que obtém nos seus ficheiros de log.

De um modo geral, para instalar um serviço faça isto:

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

Aplicando o Chroot ao Ntpd

O Ntpd é somente um serviço de tempo que permite manter o seu computador sincronizado com outros computadores em tempo real. É uma coisa simples para fazer chroot.

```
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
```

Aplicando o Chroot ao DNS ou named

Já feito, verifique:
http://www.linuxdoc.org/HOWTO/Chroot−BIND8−HOWTO.html
ou
http://www.linuxdoc.org/HOWTO/Chroot−BIND−HOWTO.html

Ou, se quiser utilizar a minha script,

```
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
```

Aplicando o chrooting aos serviços do Syslog e as minhas queixas.

Eu queria o demónio syslogd chrooted. O meu problema é que o syslogd utiliza, por omissão dev/log, o qual não pode ser visto pelos serviços chrooted. Assim, eu não posso obter logs facilmente. Eis aqui as soluções possíveis:

- Aplicar o chroot ao syslogd por cada serviço. Eu testei este e fui capaz de obter os logs. Mas não gosto deste visto que tenho uma conta root a correr o serviço.
- Ver se podemos ligar a um site externo capaz de nos fornecer logs.
- Fazer somente log dos ficheiros para um ficheiro sem ser através do syslogd. Esta é provavelmente a melhor opção de segurança, mas se alguém entra podia brincar com os logs.
- Configurar o syslogd principal para procurar em várias localizações por todos os serviços. Adiciona a opção −a ao syslogd para tal.

A minha única solução foi ter a certeza que o syslogd está chrooted por cada serviço. Eu gostaria mais de uma solução do tipo que não utilizasse uma conta root utilizando um ambiente chrooted, como por exemplo através de uma porta de rede. Provavelmente, pode ser feito, mas vou parar por aqui e tentar descobrir uma solução melhor.

Se não quiser separar o syslogd por cada serviço, então ao principal syslogd a correr no seu sistema adicione o seguinte comando quando o syslogd começa:

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

Se eu tivesse o ssh e o dns a correr, podia parecer−se com,

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

Última nota acerca do Syslogd, eu gostaria de poder corrê−lo sem ser sob uma conta root. Tentei várias coisas simples, mas não trabalhou e acabei por desistir. Se conseguisse correr o syslogd sob uma conta não root então as minhas necessidades de segurança estariam preenchidas. Possivelmente ter log externos é uma solução.

Aplicando o chrooting ao Apache

Este foi extremamente fácil de fazer. Depois de o ter configurado fui capaz de executar as scripts em Perl. Agora o meu ficheiro de configuração é um pouco grande pois tive de incluir as bibliotecas de Perl e PostgreSQL na área de chroot. Note−se uma coisa, se está a ligar a uma base de dados certifique−se que o serviço de base de dados está a correr no dispositivo de loopback 127.0.01 e que especificou a máquina 127.0.0.1 na sua script de Perl para o módulo DBI. Eis aqui um exemplo de como me ligo à base de dados utilizando ligações persistentes no 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});}
```

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

Compile e instale o apache no seu sistema principal em /usr/local/apache. Depois utilize a script em perl.

```
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
```

Eu apliquei o chroot ao meu ficheiro httpd.conf para ter isto:

```
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>
```

De seguida, aponte o seu browser para http://127.0.0.1/server−status ou http://127.0.0.1/server−info e verifique−o!

Aplicando o chrooting ao Ssh

Antes de tudo, idealmente, devia aplicar o rencaminhamento do ssh do porto 22 para o porto 2222. Depois inicia o ssh, tendo−o à escuta no porto 2222 sob uma conta não−root. Para a ligação ssh, queremos ter ligações seguras com passwords para permitir a entrada de pessoas, mas para nada mais. Depois de fazer login ter um segundo programa ssh a rodar no porto 127.0.0.1:2322 que os deixará ligar ao sistema real −− O segundo programa ssh SÓ deve escutar no dispositivo de loopback. Agora é isto que deveria fazer. Não o vamos fazer. A única coisa que vamos fazer para este exemplo é aplicar o chroot. Os exercícios deixados ao leitor incluem por o sshd a correr sob uma conta não−root e instalar um segundo programa o qual escuta no dispositivo de loopback permitindo as pessoas entrar no sistema real.

Novamente, só vamos aplicar o chroot ao ssh e deixar que você se preocupe com as consequências de tal (não conseguirá ver todo o sistema se fizer, simplesmente isto). Idealmente, seria agradável configurar isto para gravar os logs externamente. Deveríamos, também utilizar o OpenSSH, mas estou a utilizar o SSH comercial por simplicidade (o que não é uma desculpa).

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

Instale o ssh em /usr/local/ssh_chroot. Depois utilize a script em Perl.

```
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
```

Suponho que uma coisa realmente boa é pôr o ssh sob um ambiente chrooted, é que se quiser utilizá−lo para substituir um servidor ftp, as pessoas terão acesso limitado à sua área. O Rsync e o SCP combinam−se para permitir que as pessoas façam transferência de ficheiros. Eu, realmente, não gosto de pôr um servidor ftp a correr só para as pessoas fazerem login. Imensos servidores ftp estão também chrooted, mas continuam a transmitir as palavras passe em texto, o que eu não gosto.

Aplicando o chroot ao PostgreSQL

Este foi praticamente simples como o perl, com a excepção de requerer mais algumas bibliotecas. Mas no final, não foi difícil de fazer. Uma coisa que tive de fazer foi pôr o PostgreSQL aberto para a rede, mas só no dispositivo de loopback. Como estava chrooted outros serviços não conseguiam chegar até ele, como o servidor apache de web. Eu compilei o Perl dentro do PostgreSQL assim tive de adicionar muitas coisas do Perl ao meu ficheiro de configuração.

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

Compile e instale o seu postgreSQL no seu sistema principal em /usr/local/postgres. Depois utilize a script em Perl.

```
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
```

Aplicando o chroot ao Sendmail

Avançe e execute a minha script.

```
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
```

Agora é que são elas? Sim. Ainda continua a correr como root. Maldição. Também alguns ficheiros são recriados pelo /etc/rc.d/init.d/sendmail quando é iniciado e a minha script não está preparada para tal. Qualquer vez que faça alterações ao sendmail em /etc/mail, copie, por favor, ao alterações para /chroot/sendmail/etc também. Terá, também de apontar o /var/spool/mail para /chroot/sendmail/var/spool/mail para que o programa sendmail e os utilizadores (quando fazem login) possam ver os mesmos ficheiros.

O lado bom é que pode enviar para fora o mail, é ao receber que o problema está. Assim, fui capaz de instalar o sendmail com o apache sem qualquer problema. Algumas das minhas scripts enviam mail para fora, e assim preciso dos ficheiros do sendmail copiados para a área chroot devido ao apache.

Outras coisas a aplicar o chroot.

Eis aqui a minha filosofia:

1. Tudo devia estar chrooted, incluindo o sendmail, o ssh, o apache, o postgresql, o syslog, e qualquer serviço a correr no computador.
2. Tudo devia ser posto sob uma conta não−root (pode ter necessidade de rencaminhar os portos protegidos para portos não protegidos). Isto inclui o sendmail e o syslog.
3. Os Logs deviam ser enviados para um site externo.
4. Uma partição devia ser configurada para cada serviço para limitar o espaço de disco que um pirata pode utilizar se ele decidir escrever ficheiros. Pode utilizar um dispositivo de loopback para montar os ficheiros como sistemas de ficheiros para alguns destes serviços, se esgotar as partições.
5. O Root deve possuir todos os ficheiros que não se alteram.

O que diz respeito ao sendmail e ao syslogd, ainda continuo a pensar que devia correr sob uma conta não−root. Para o sendmail, isto deve ser possível, mas achei−o extremamente difícil. Ainda não tive sucesso em pôr o sendmail a correr sob uma conta não−root e penso que é um erro grave em não correr sob uma conta não−root. Sei que existem problemas devido a isto, mas penso que todos podem ser resolvidos. Logo que se trate da permissão dos ficheiros, não vejo por que razão o sendmail tem de correr como root. Provavelmente, existe alguma razão que não percebo mas não tenho dúvidas que não consigamos sobrepôr os obstáculos.

Para o syslog, nem sequer tentei, mas eu diria que os logs deviam ser escritos sob uma conta não−root e não vejo por que razão tal não é possível. Pelo menos consegui ter o syslog chrooted para cada serviço.

Todos os serviços deviam ser configurados para contas não−root. Mesmo o NFS. Tudo.

Sugestões

- Utilizo dois logins para o ssh e tenho dois demónios sshd a correr.
- Descobrir como ter o sendmail ou outro programa de mail a correr numa conta não−root.
- Retirar as bibliotecas desnecessárias na /lib. Eu copiei tudo para me facilitar. Não precisa da maior parte delas.

- Faça logging remoto do syslogd e descubra como é que podemos associar o syslogd a uma porta de rede e ter todos os serviços a ligarem−se a esse porto através do dispositivo de loopback. Ver se sonseguimos por o syslogd a correr com uma conta não−root.

Conclusão

Penso que o chroot é adequado para todos os serviços. Acredito que é um grande erro não aplicar o chroot a todos os serviços que correm sob contas não−root. Gostaria que as grandes distribuições o fizessem, ou uma pequena: QUALQUER distribuição. A Mandrake começou através de material da RedHat expandindo−o, assim, provavelmente alguém pegue no Mandrake e expanda o seu chroot. Nada previne que outras pessoas refaçam o trabalho de outras na GNU/Linux, por isso creio ser possível. Se alguma companhia quisesse aplicar o chroot a tudo e criar um ambiente sistemático para as pessoas administrarem os serviços chrooted, teriam uma distribuição fantástica! Lembre−se que agora que o Linux se está a expandir, as pessoas que não gostam da linha de comandos, assim tudo será feito ao nível da gui, elas não precisam de ver as entranhas e nem sequer precisam de saber o que se passa −− só precisam de ser capazes de o configurar e saber como trabalha!

Eu suporto a 100% a ideia de todos os serviços serem chrooted com contas não−root e que qualquer distribuição que não faça isto é menos própria para mim para utilizar num ambiente de produção. Eu vou aplicar o chroot a tudo, tanto quanto possível −− eventualmente chegarei lá.

Planeio criar os HOWTO acerca do chrooting. Estou a submeter um pedido para alguém me ajudar a converter este artigo no formato LyX para que possa ser posto nos HOWTOs do Linux.

Referências

1. Se este artigo for modificado, estará disponível aqui http://www.gnujobs.com/Articles/23/chroot.html

2005−01−10, generated by lfparser_pdf version 2.51