Cómo hacer jaulas chroot y montarlas al momento de iniciar el sistema automáticamente en Debian.

Author : muammar

¿Qué es una Jaula?
Una jaula, o chroot, en un sistema operativo Unix es una operación que cambia el directorio raíz, afectando solamente al proceso actual y a sus procesos hijos. "chroot" se refiere a la llamada de sistema chroot(2) o al programa ejecutable chroot(8).
El sistema chroot fue introducido por Bill Joy el 18 de marzo de 1982 (17 meses antes de que BSD fuera liberado) para probar su sistema de instalación y construcción. No fue pensado como un mecanismo de seguridad.
Un programa que se ejecuta en un entorno de directorio raíz cambiado, no puede acceder a archivos fuera de ese directorio. A menudo es malentendido como un dispositivo de seguridad, usado en un vago intento de crear una zona segura para correr un programa que provoca desconfianza, o que no está probado o que de algún modo es peligroso, como si chroot fuera un mecanismo real de jaula
funcional.
En la práctica, hacer un chroot es complicado ya que los programas esperan encontrar en lugares predeterminados el espacio de almacenamiento, archivos de configuración, sus bibliotecas de enlace dinámico; entonces para habilitar a los programas que puedan correr dentro de un directorio chroot, se debe incluir también un conjunto mínimo de estos archivos, ya que el programa no puede
acceder fuera de la jaula.
Los programas tienen permitido llevarse descriptores de archivos abiertos (sean de archivos físicos, de tuberías, o de conexiones de red) dentro del chroot, lo cual puede simplificar el diseño de la jaula haciendo innecesario dejar archivos funcionales dentro del directorio chroot. Esto también funciona como un sistema de capacidades simple, en el cual, al programa se le otorga acceso explícito a
los recursos externos del chroot basado en los descriptores que puede llevar a su interior.[1]
En el kernel, cada proceso tiene asociados ciertos valores, como son el UID/GID del usuario que ejecuta el proceso, el pid que identifica al proceso, el estado del mismo, punteros a otros procesos, el directorio actual del proceso y entre otras cosas: **EL DIRECTORIO RAIZ DEL PROCESO**.
Chroot es una syscall que permite modificar este último valor mencionado. Es decir, cambiar el directorio raíz del proceso actual y todos sus descendientes. De ahí lo de chroot: change root. Bueno, al hacer esto, el proceso no va a poder abrir descriptores de ficheros que estén fuera del directorio configurado como directorio raíz. Por ejemplo, si rerooteamos un proceso a /var/jaula/. Si este proceso intenta acceder a /etc/passwd, va a acceder a /var/jaula/etc/passwd. Como pueden ver, el proceso queda enjaulado dentro de /var/jaula/, ya que es su directorio raíz.[2]

¿Qué se necesita para montar una jaula en Debian?
Para montar una jaula se necesita instalar:
1. debootstrap
2. coreutils (este paquete es requerido por el sistema, por tanto debería estar instalado por

defecto
en Debian)
3. Conexión a internet o un CD que contenga los paquetes necesarios (el disco 1 de Debian sería
suficiente).
Instalar estos paquetes en Debian es muy sencillo. Lo primero que se debe buscar es el archivo /etc/apt/sources.list y verificar que al menos contamos con esta línea (en caso de que no poseas internet, deberías tener la línea que corresponde al uso de APT con DVD/CDROM):

```
deb http://ftp.us.debian.org/debian/ stable main
```

Luego de verificado esto, procedemos a actualizar la base de datos de los paquetes ejecutando como super usuario cualquiera de los siguientes comandos:

```
# aptitude update
```

o

```
# apt-get update
```

Una vez finalizado este lo descrito arriba procedemos a instalar debootstrap. Para este, paso de nuevo como súper usuario, ejecutamos cualquiera de estos comandos para iniciar la instalación:

```
# aptitude install debootstrap
```

o

```
# apt-get install debootstrap
```

Preparando el entorno
Ahora lo que se requiere es preparar el entorno. Para ello solo se necesita saber un poco de Bash Scripting. Supongamos que se quieren instalar un número *n* de jaulas en la partición /mnt/ en nuestro sistema. Lo que tenemos que hacer es crear un directorio bajo dicha partición:

```
 # cd /mnt/
# mkdir jaula1
```

Luego exportamos la ubicación del directorio que acabamos de crear, y le asignamos un nombre, supongamos J1. De ahora en adelante nos referiremos a la jaula1 como $J1:

```
# export J1=/mnt/jaula1
```

En el siguiente paso procederemos a instalar el sistema base con debootstrap. En caso de que no uses Debian, tendrás que bajar el código fuente y compilarlo.

Descargando el sistema base

Debootstrap es utilizado para crear un sistema base Debian prístino, sin requerir la disponibilidad de dpkg o apt. Este paquete es capaz de hacer esto bajando todos los archivos *.deb desde un mirror y luego cuidadosamente los desempaca dentro de un directorio el cual puede ser chrooteado.[3]

El uso de debootstrap es el siguiente, en líneas generales (Para más información por favor consultar el manual de DEBOOTSTRAP(8)):

```
# debootstrap --arch i386 lenny $J1 http://ftp.us.debian.org/debian
```

Donde debootstrap invoca dicho programa, el argumento –arch escoge el tipo de arquitectura, lenny es la rama de Debian a utilizar, $J1 es la variable que contiene la ubicación de dónde va ser instalada la jaula (en cualquiera de los casos si no funciona se puede poner a mano el nombre de la ubicación), y por último se escoge el mirror de donde se va a descargar todo el sistema base.

La salida esperada al ejecutar debootstrap debería ser algo como esto:

: Retrieving Release
I: Retrieving Packages
I: Validating Packages
I: Resolving dependencies of required packages...
I: Resolving dependencies of base packages...
I: Checking component main on http://ftp.us.debian.org/debian...
I: Retrieving adduser
I: Validating adduser
I: Retrieving apt
I: Validating apt
.
.
.
I: Base system installed successfully.

Este proceso dura un tiempo y dependerá tanto de la conexión a Internet como de la configuración de hardware de la PC.

Actualización (14/12/2009): Si tienes un disco de instalación de Debian, este proceso se puede acelerar utilizando el contenido de dicho disco. Para ello, ejecutas:

```
#debootstrap --arch i386 lenny carpetadeinstalación/ file:///media/cdrom/debian
```

Configurando la jaula (chroot)

Antes de poder acceder a la jaula, es necesario montar el sistema de archivo del procesador y el peuso terminal esclavo. Para ello ejecutamos:

```
# mount -t proc proc $J1/proc
```

```
# mount -t devpts devpts $J1/dev/pts
```

Nota: Si este paso falla, no te preocupes, pero no por eso deja de ser necesario. Si no montas dicho pseudo terminal, e instalas un servidor SSH no podrás acceder a él en la jaula. Se colgará el shell. Para más información leer el manual de pts(4).
Cuando no vayas a usar las jaulas es importante que desmontes con umount tanto el procesador como el pseudo terminal.
Ahora ya podemos entrar a nuestra jaula virtual como súper usuario ejecutando lo siguiente:

```
# chroot $J1
```

Comandos para configurar la jaula (chroot)
Cuando accedes a tu jaula debes al menos correr esta serie de comandos una vez:

```
# cd /dev; /sbin/MAKEDEV generic; cd -
```

Debes también acceder al archivo /etc/apt/sources.list y agregar al menos estas líneas:

```
deb http://ftp.us.debian.org/debian lenny main contrib
deb http://security.debian.org/ lenny/updates main contrib
```

Ahora debes actualizar la base de datos de los paquetes con aptitude update o apt-get update. Además de lo mencionado anteriormente el paquete locales debe ser instalado y configurado, para ello ejecutamos:

```
# aptitude install locales && dpkg-reconfigure locales
```

o

```
# apt-get install locales && dpkg-reconfigure locales
```

Es importante señalar que el usuario root no tiene contraseñas cuando se instala una jaula. Para configurar un password se debe ejecutar passwd.
Configurando e instalando el servicio deseado.

Ya configurada e instalada la jaula, es el momento de instalar los servicios deseados. Digamos que se quiere instalar Apache2. Para ello ejecutamos:

```
# aptitude install apache2
```

o

```
# apt-get install apache2
```

Una vez instalado, iniciamos el servicio de la siguiente manera:

```
#/etc/init.d/apache2 start
```

Al acceder en el navegador en http://localhost debería aparecer el mensaje: It Works!. De la manera descrita anteriormente podemos proceder a instalar los servicios deseados así como también el número de jaulas para cada servicio deseado.

Iniciando las jaulas *at boot time*.

La pregunta natural que se formula en este punto es: ¿cómo hago que mis jaulas se monten al iniciar el sistema de manera automatizada? La respuesta: un Script hecho en Bash. En caso de no ser muy diestros con Bash, por favor revisar: http://tldp.org/LDP/Bash-Beginners-Guide/html/ la cual es una excelente guía para principiantes.

El script pertinente en nuestro caso debe crearse en /etc/init.d/, en este caso el script se llama prueba.sh (nótese la terminación sh) y contiene las siguientes líneas:

```
export CHROOTDIR=/mnt/nombre
mount -t devpts devpts $CHROOTDIR/dev/pts
mount -t proc proc $CHROOTDIR/proc
chroot $CHROOTDIR /etc/init.d/apache2 start
```

Nota: Para el caso de apache2. Si fuera SSH, debes cambiar apache2 por ssh. Lo importante es saber que no se inician por defecto los servicios dentro de las jaulas y hay que iniciarlos en el script que se está creando. Este script lo podemos probar de la siguiente manera (ejecutándolo fuera de la jaula):

```
# /etc/init.d/prueba.sh
```

El último paso es hacer un enlace simbólico en /etc/rc2.d/ para de esta manera hacer permanente las acciones ejecutadas por el script creado anteriormente:

```
# ln -s /etc/init.d/prueba.sh /etc/rc2.d/S99prueba
```

Bibliografía consultada

1. http://es.wikipedia.org/wiki/Chroot
2. http://sherekan.com.ar/2008/11/19/que-es-y-como-crear-chroot/
3. http://packages.debian.org/lenny/debootstrap