Confinando procesos con jail

Víctor Balada Díaz

victor@alf.dyndns.ws

Este artículo repasa el confinamiento de procesos en jail, el sistema similar a chroot de FreeBSD.

Víctor Balada Díaz con 16 años es administrador de varios sistemas FreeBSD. Ha colaborado con proyectos como www.userbsd.net y J-Lan Communicator.

Contacto: `<victor@alf.dyndns.ws>`

Tabla de contenidos

1. 1. Introducción.

1.1. 1.1 Un vistazo por encima.

Jail es un sistema para confinar los procesos en un entorno restringido parecido a chroot, pero bastante más avanzado. Su utilidad es diversa (se puede ver un ejemplo práctico de un servidor shell en la parte final del documento), desde montar servidores virtuales para sus clientes con un falso sentido de ser administradores, a aumentar la seguridad del sistema haciendo que los demonios corran en entornos aislados de tal forma que si uno fuese vulnerable no quedase el sistema entero a merced del atacante.

Entre sus ventajas frente a chroot se encuentran:

- Mayor seguridad.
- Puede tener su propio fichero de contraseñas con su correspondiente root y usuarios independientes al resto del sistema.
- Puede tener un hostname diferente al de la máquina principal.
- Puede tener una zona horaria diferente.
- El usuario root de la máquina puede cambiar el hostname (esto se puede cambiar con una sysctl que veremos mas adelante).
- Se pueden tener varios procesos de diferentes usuarios en una misma jaula.

Por supuesto no todo son ventajas, también tiene sus incovenientes:

- Solo se puede tener una IP por cada jaula.
- El sistema no es portable, es espec\x{00ED}fico de FreeBSD.
- Cada jaula necesita su propia IP y no puede ser compartida.

Todas las partes han sido probadas en un equipo con FreeBSD 5.1-CURRENT, para cualquier pregunta, sugerencia o comentario puede mandarme un mail a: `<victor@alf.dyndns.ws>`

1.2. 1.2 Requisitos previos.

Para crear una jaula vamos a necesitar las siguientes cosas:

I. Una direccion IP.
II. El código fuente del sistema con el que estamos funcionando o el cdrom con los paquetes precompilados del sistema instalado.

Ademas de esto, usaremos también las siguientes variables (es recomendable que las añada a su perfil de shell por ejemplo `.cshrc` para no olvidar ponerlas tras un reboot o algo similar):

1. Una variable de entorno $JAIL_PATH para referenciar al directorio donde estara la jaula.
2. Otra variable de entorno para referenciar a la IP de la jaula, a la que llamaremos $JAIL_IP.
3. Y la última variable más para definir el hostname de la jaula, a la que llamaremos $JAIL_HOSTNAME.

Para referenciar a la maquina que contiene las jaulas nos referiremos a ella como "equipo principal".

Una vez comentado esto vamos a ver como conseguir todas estas cosas.

1.3. 1.2.1 Consiguiendo una dirección IP.

Bueno, este es el mas complejo de todos, para IPs públicas tiene que pedirlas a su ISP (con el correspondiente cargo extra) o pedirlas directamente a IANA (http://www.iana.org/)

1.4. 1.2.2 Conseguir el código fuente.

El código fuente del sistema se puede obtener en tres simples pasos:

I. Descargar la aplicación cvsup, para ello tiene 3 opciones:

   A. Usar el sistema de ports:

      ```
      # cd /usr/ports/net/cvsup-without-gui && \
      make install clean
      ```

   B. Usar pkg_add -r (esta es la mas rápida) así:

      ```
      # pkg_add -r ftp://ftp.freebsd.org/pub/FreeBSD/ports/\
      packages/net/cvsup-without-gui-16.1h.tgz
      ```

   C. Descargar el paquete con sysinstall:

      ```
      # /stand/sysinstall
      ```

      Tras arrancar la aplicación saldrá un menú, debe seguir este orden:

      ```
      Configure -> Packages -> FTP -> Primary Site -> Yes \
      -> net -> cvsup-without-gui-16.1g
      ```

      Pulsamos la tecla Tab una vez para ir a ok, pulsamos enter y al volver al menú principal de ports le volvemos a dar a la tecla Tab una vez para llegar hasta install, luego pulsamos enter y a esperar a que lo baje y lo instale.

II. Configurar cvsup. Esto es tan simple como crear un fichero con las siguientes lineas:

   ```
   *default host=cvsup.es.FreeBSD.org
   *default base=/usr
   *default prefix=/usr
   *default release=cvs tag=.
   *default compress
   ports-all
   ```

   Donde pone . en "*default release=cvs tag=." debería poner el nombre de su release (por ejemplo RELENG_4_8)

III. Arrancar cvsup y dejar que baje los sources, para ello usaremos este comando:

   ```
   # cvsup -g -L 2 /path/donde/puso/el/fichero/de/antes
   ```

1.5. 1.2.3 Poner las variables de entorno.

La sintaxis es 'setenv VARIABLE valor':

```
# setenv JAIL_PATH /home/jail1/
# setenv JAIL_IP 10.0.0.1
# setenv JAIL_HOSTNAME jaula
```

2. 2. Creación de una jaula.

La creación de un entorno con jail es bastante simple, aunque algunos demonios pueden traernos dolores de cabeza, como por ejemplo sendmail o rpcbind, para los cuales es mejor poner una jaula dedicada para cada uno o común para todos ellos, pero fuera del equipo principal.

Hay dos formas de hacerlo, una por código, y otra con los paquetes del cdrom.

2.1. 2.1 Creación de una jaula por código.

```
# cd /usr/src
# mkdir -p $JAIL_PATH
# make world DESTDIR=$JAIL_PATH
# cd etc
# make distribution DESTDIR=$JAIL_PATH
```

2.2. 2.2 Creación de una jaula desde el cdrom.

```
# setenv DESTDIR $JAIL_PATH
# cd /cdrom
# cd bin
# sh install.sh
# cd ../manpages
# sh install.sh
```

Y seguiríamos así sucesivamente hasta instalar los paquetes deseados.

2.3. 2.3 Creación de una jaula (parte común)

Hay una parte común que corresponde con:

```
# mount_devfs devfs $JAIL_PATH/dev
# cd $JAIL_PATH
```

```
# ln -sf dev/null kernel
```

Instalándolo así, el usuario root de la jaula tendrá acceso a los dispositivos y a la memoria, pudiendo modificar partes externas e incluso salir la jaula si no se está en un secure level apropiado y las reglas de devfs(8) no están ajustadas de forma correcta.

> **Nota:** Dentro de la jaula esta restringido el acceso a la syscall mknod(2) (entre otras) por tanto ningún proceso de la jaula puede crear nuevos dispositivos.

2.4. 2.4 Configuración inicial de la jaula

Seguidamente vamos a configurar las cosas básicas para que una jaula "completa" pueda funcionar, por ejemplo el fichero de contraseñas, y la timezone:

I. Configurar todos los demonios que corran sobre el equipo principal para que solo escuchen peticiones en la IP de la máquina principal, para que no existan problemas con las jaulas, por ejemplo para inetd añadiremos la siguiente linea en `/etc/rc.conf`:

   ```
   inetd_flags="-a 10.0.1.1"
   ```

   Donde 10.0.1.1 seria la ip de la máquina principal.

II. Copiar sysinstall para configurar las cosas de una forma más sencilla:

   ```
   # mkdir $JAIL_PATH/stand
   # cp /stand/sysinstall $JAIL_PATH/stand
   ```

III. Arrancamos todas las jaulas necesarias sin configurar su interfaz de red y configurar así lo básico:

   ```
   # jail $JAIL_PATH $JAIL_HOSTNAME $JAIL_IP /bin/csh
   ```

   Si todo ha salido bien, después de esto deberíamos terminar con una shell, desde la cual ejecutar:

   ```
   # touch /etc/fstab
   # echo rcpbind_enable='"'NO'"' >> /etc/rc.conf
   # newaliases
   # echo network_interfaces='""' >> /etc/rc.conf
   # passwd
   ```

   Escribimos el passwd de root de la jaula 2 veces.

   ```
   # /stand/sysinstall
   Configure -> Time Zone -> No -> Europe -> Spain -> mainland -> yes
   # echo "nameserver 111.111.111.111" >> /etc/resolv.conf
   ```

   Donde 111.111.111.111 es la IP del servidor DNS que tengais.

De forma opcional ahora se pueden añadir cuentas de usuario, instalar paquetes, y cualquier otra configuración que se necesite.

Ahora puede poner exit y eso cerrará la jaula, y con esto damos por concluida la configuración inical.

2.5. Sysctls especificas.

La configuración de las sysctls es imprescindible hacerlas antes de arrancar la jaula, se pueden cambiar con sysctl(8) como root del equipo principal.

security.jail.set_hostname_allowed

Indica si el usuario root de la jaula puede cambiar el hostname de dicha jaula o no.

security.jail.socket_unixiproute_only

Como todas las familias de protocolos no tienen implementada la funcionalidad de jail, esta sysctl se encargara de restringir que dentro de la jaula se puedan crear sockets de esas familias, actualmente las familias soportadas son: PF_LOCAL, PF_INET y PF_ROUTE.

security.jail.sysvipc_allowed

Esta sysctl se encarga de evitar que un proceso dentro de la jaula tenga acceso a comunicar con otras jaulas o el equipo principal mediante los mecanismos de comunicación entre procesos de System V.

3. 3. Jail en la práctica

Ya estamos listos para arrancar una jaula completa, ahora debemos levantar la interfaz de red, e iniciar la jaula:

```
# echo sshd_enable='"'YES'"' >> $JAIL_PATH/etc/rc.conf
# ifconfig fxp0 alias $JAIL_IP
# jail $JAIL_PATH $JAIL_HOSTNAME $JAIL_IP /bin/sh /etc/rc
```

Es tan simple como esto el tener un servidor sshd funcionando dentro de jail.

Espero que siguiendo los pasos descritos no tengais ningún problema para poner en marcha vuestras propias jaulas con jail :-)