

Configuración de Firewall por software bajo Linux

Advertencia

Nota : Esta guía está reservada a gente con un **buen nivel** en la administración de servidores dedicados bajo Linux.

La manipulación de un filtro firewall puede ser **peligrosa** y puede producir pérdida de conectividad en su servidor.

En efecto, las operaciones aquí descritas puede bloquear su servidor lo cual le forzará a reiniciar en HARD. Si se equivoca en el script final y lo pone en auto reinicio, no podrá acceder a su máquina.

Ponga pues **mucha atención** y si no se siente muy confiado con esta guía, no parametrize su firewall.

En caso de que no pueda volver a entrar en su máquina, ni siquiera reiniciando el servidor, reinicie de nuevo en modo rescue a través del Netboot para desactivar el Firewall en el inicio.

Para más información sobre el modo rescue, consulte la guía : [ModoRescue](#)

Cuales son los puertos que usted utiliza

ADVERTENCIA /!\

Antes de nada, hay que prestar atención a lo que va a hacer. En efecto vd corre el riesgo de equivocarse de puertos y cerrar los erroneos.

Imagine si cierra el puerto SSH !
Habría que reiniciar mediante webmin o desde el Manager (HARD).

Por tanto preste atención

Los puertos abiertos por defecto sobre los servidores OVH son:

- 21 – ftp (el servidor FTP, a dejar según utilización).
- 22 – ssh (el acceso al shell criptado, dejarlo abierto !).
- 23 – telnet (el acceso al shell no criptado, dejar en caso de avería).
- 25 – smtp (el servidor de correo entrante, dejar en la mayor parte de los casos).
- 53 – dns (el servidor DNS, dejar en la mayoría de los casos).
- 80 – http (servidor web, dejar).
- 110 – pop3 (el acceso a los e-mails, dejar en la mayoría de los casos).

OVH

- 143 – imap (acceso a los mails, dejar si no utiliza pop3).
- 443 – https (acceso al web criptado, dejar según sus preferencias de utilización).
- 10000 – webmin (panel de configuración de servidor, dejar si lo necesita).

Estos puertos son los abiertos por defecto pero vd tendrá seguramente programas lanzados que abren otros.

Es su tarea saber cuales debe guardar o no.

Una vez ha efectuado su elección, pasamos a aplicarlo.

ADVERTENCIA /!\

El Manager tiene un Firewall para el puerto IRC (6667)

Si desea abrir dicho puerto debe consultar la guía sobre instalación de IRC : InstalarServidorIrc

Iptables

El comando **iptables** es un firewall muy potente instalado en todos los servidores Linux/BSD/Solaris. El funcionamiento será el siguiente: abriremos ciertos puertos y cerraremos el resto.

En este ejemplo, vamos a dejar sólomente el puerto 22(SSH) y 80(HTTP).

Solo es un ejemplo; es su decisión adaptarlo a sus necesidades.

Actualizar la versión

Conéctese con SSH en root.

Lo primero a hacer es comprobar la versión de iptables :

```
/sbin/iptables -V  
iptables v1.2.4
```

Si la versión es muy antigua, vamos a instalar una más actual, por ejemplo la 1.2.9 :

```
# cd /root
```

```
# wget http://www.netfilter.org/files/iptables-1.2.9.tar.bz2
```

```
# tar xvfj iptables-1.2.9.tar.bz2
# cd iptables-1.2.9

# make KERNEL_DIR=/usr/src/linux
...

# make install KERNEL_DIR=/usr/src/linux
...

# cd /sbin
# mv iptables iptables.old
# mv iptables-restore iptables-restore.old
# mv iptables-save iptables-save.old

# ln -s /usr/local/sbin/iptables iptables
# ln -s /usr/local/sbin/iptables-restore iptables-restore
# ln -s /usr/local/sbin/iptables-save iptables-save
```

Ahora iptables está actualizado, podemos empezar.

```
# /sbin/iptables -V
```

```
iptables v1.2.9
```

Configuración

Conectese con SSH en root.

Listamos las instrucciones existentes :

```
# /sbin/iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
```

Vemos tres secciones : Input, Forward et Output.

Vamos a ocuparnos de la sección Input de momento (para el trafico entrante).

Autorizamos los puertos 22 y 80 :

```
# /sbin/iptables -A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT
# /sbin/iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT
```

- `-A INPUT` : situamos nuestra instrucción a la entrada del firewall.
- `-i eth0` : aquí la interfaz ethernet que nos interesa.
- `-p tcp` : el protocolo tratado es el TCP(solo tratamos este de momento).
- `--dport 22` : la instrucción va a ser aplicada sobre el puerto SSH (nº 22).
- `-j ACCEPT` : aceptamos este tráfico.

Volvemos a listar todo :

```
# /sbin/iptables -L

Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp anywhere anywhere tcp dpt:ssh
ACCEPT tcp anywhere anywhere tcp dpt:www

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
```

La sección Input se ha llenado, es buena señal ;)

Vemos que la política por defecto es de aceptarlo todo => Chain INPUT (policy ACCEPT). Queremos bloquear todo el tráfico que no tenga una autorización previa. Por tanto vamos a añadir una instrucción para bloquear los demás puertos.

Entonces surge un problema : en el momento en que una conexión se va a realizar desde nuestro servidor, por ejemplo hacia el servidor kernel.org para descargar el nuevo núcleo (solo es un ejemplo), se va a establecer una conexión con el sitio web y va a esperar su respuesta. La petición de conexión saldrá bien pero... ¿va a regresar dado que lo hemos bloqueado todo ?

Afortunadamente, iptables es potente y puede seleccionar los paquetes según su estado.

Por tanto, antes de bloquear vamos a introducir una instrucción :

```
# /sbin/iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j
```

ACCEPT

Solo falta bloquear el resto.

Atención, es aquí donde el firewall va a hacer verdadero efecto, compruebe que tiene bien introducidas sus instrucciones ACCEPT, sino corre el riesgo de **bloquear su servidor**)

```
# /sbin/iptables -A INPUT -i eth0 -j REJECT
```

Tenemos dos opciones al nivel de esta instrucción.

En la primera hacemos un DROP de los paquetes, es decir que si un paquete llega y no está aceptado se borra. El cliente esperará de su parte una respuesta en vano, hasta el timeout.

La segunda solución es relanzar los paquetes (REJECT en lugar de DROP). Si un paquete no solicitado llega, se reenvía al cliente un error y él no espera más porque hay una respuesta negativa.

Relanzar los paquetes es más propio pero eliminarlos es más seguro. En efecto, imaginemos que alguien le envía paquetes repetidos sobre un puerto, su servidor no los tratará mientras que con la regla reject tomará su tiempo en responder.

Es su elección ;)

En nuestro ejemplo utilizamos REJECT.

Prueba

El firewall está en acción. Pruebe de escanear su servidor, únicamente verá los puertos 22 y 80 abiertos. No se sorprenda si el scan es muy lento si utiliza la instrucción DROP.

Para poner a cero su firewall escriba :

```
# /sbin/iptables -F INPUT
```

Este comando suprimirá todas las instrucciones de la sección INPUT.

Si desea añadir una instrucción entre la primera y la segunda, introduzca :

```
# /sbin/iptables -I INPUT 2 < su instrucción >
```

Para suprimir la instrucción 3ª introduzca esto :

```
# /sbin/iptables -D INPUT 3
```

Para bloquear totalmente una IP (AAA.BBB.CCC.DDD) :

```
# /sbin/iptables -I INPUT 1 -s AAA.BBB.CCC.DDD -j DROP
```

IP a excluir y a autorizar

Entramos en un ejemplo real de configuración de Firewall que autoriza las IPs de los servidores de monitorización de OVH.

En el caso de un servidor dedicado hay que autorizar las IPs que gestionan la monitorización, las intervenciones, el RTM y las MRTG del servidor.

En el caso de un RPS hay que autorizar además el acceso a la SAN de almacenamiento.

Para un servidor dedicado

Si desea bloquear el protocolo ICMP (las peticiones ping), debe dejar al menos el paso a los ordenadores de monitorización.

ADVERTENCIA /!

Deberá permitir el acceso a los ordenadores de monitorización :

- **ping.ovh.net – 213.186.33.13**
- **cache.ovh.net – 213.186.50.100**
- **proxy.ovh.net – 213.186.50.98**
- **proxy.p19.ovh.net – 213.186.45.4**
- **proxy.rbx.ovh.net – 213.251.184.9**
- **proxy.rbx2.ovh.net – 91.121.150.4**
- **proxy.rbx3.ovh.net – 91.121.180.2**
- **proxy.rbx4.ovh.net – 46.105.96.3**
- **a2.ovh.net – 213.186.33.62 (Monitoring firewalls ASA)**

Ello permite a los técnicos de OVH verificar el buen estado de su servidor.

Si bloquea todas las peticiones ping, incluso las de OVH, no podremos comprobar el buen estado de su servidor y si este se cae no seremos advertidos.

Para autorizar el acceso al servidor SLA y tener RTM, debe autorizar la IP de su servidor terminada en **.250** y **.251**, por ejemplo, si su IP es de la forma **aaa.bbb.ccc.ddd**, debe autorizar el acceso a la dirección

aaa.bbb.ccc.250 y aaa.bbb.ccc.251.

Deberá permitir el acceso a las direcciones de SLA y MRTG:

- **sla-X.ovh.net – aaa.bbb.ccc.250**
- **mrtg-X.ovh.net – aaa.bbb.ccc.251**
- **puertos 6100 a 6200 en TCP y UDP (Puertos de comunicación RTM)**

Siendo la dirección *aaa.bbb.ccc.ddd*, la dirección IP de su servidor.

Para autorizar el ping desde nuestros servidores, introduzca las siguientes instrucciones:

```
# /sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.ovh.net -j
ACCEPT
# /sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.p19.ovh.net -j
ACCEPT
# /sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.rbx.ovh.net -j
ACCEPT
# /sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.rbx2.ovh.net -j
ACCEPT

# /sbin/iptables -A INPUT -i eth0 -p icmp --source ping.ovh.net -j ACCEPT

# /sbin/iptables -A INPUT -i eth0 -p icmp --source AAA.BBB.CCC.250 -j
ACCEPT # SLA, IP servidor = aaa.bbb.ccc.ddd
# /sbin/iptables -A INPUT -i eth0 -p icmp --source AAA.BBB.CCC.251 -j
ACCEPT # IP para el sistema de monitoring
```

En cuanto al SSH, si desea restringir el acceso únicamente desde su IP, se le aconseja también dejar **cache.ovh.net**.

Así en caso de problema en su máquina podremos intervenir en ella.

Si cierra el puerto 22 para los técnicos OVH no podremos ayudarle si su máquina está bloqueada.

Para autorizar el SSH desde nuestros servidores introduzca:

```
# /sbin/iptables -A INPUT -i eth0 -p tcp --dport 22 --source
cache.ovh.net -j ACCEPT
```

Si desea acceder a un servidor de ficheros (filer) externo de OVH, no olvide autorizar las conexiones NFS.

OVH

Para ello, le aconsejamos que autorice las direcciones de la red interna 192.168.0.0/16 :

```
# /sbin/iptables -A INPUT -i eth0 -p tcp --source 192.168.0.0/16 -j ACCEPT
# /sbin/iptables -A INPUT -i eth0 -p udp --source 192.168.0.0/16 -j ACCEPT
```

Para un servidor dedicado HG

Si el servidor es un servidor HG con dos tarjetas de red, debe autorizar la monitorización de la segunda interfaz de red.

Autorice la entrada de la dirección del switch :

- **sw-X.ovh.net – aaa.bbb.ccc.249**

Siendo la dirección *aaa.bbb.ccc.ddd*, la dirección IP de su servidor.

La instrucción a añadir es por tanto :

```
# /sbin/iptables -A INPUT -i eth0 -p icmp --source AAA.BBB.CCC.249 -j ACCEPT # sólo para los servidores HG
```

Para un servidor dedicado que forma parte de un clúster con IP Load-balancing

Si el servidor forma parte de un clúster de OVH, debe autorizar el puerto 79 para que OCO pueda comunicarse con el repartidor de carga :

```
# /sbin/iptables -A INPUT -i eth0 -p tcp --dport 79 -j ACCEPT
```

Para un servidor RPS

La interfaz monitorizada en los RPS es la eth0, luego las reglas de Firewall se aplicarán sobre esta última.

Igualmente, será necesario autorizar al servidor de ficheros de la SAN. Para ello utilice el comando :

```
r12xxx ~ # netstat -tanpu | grep iscsi
tcp 0 0 91.121.xx.xx:38632 91.121.191.16:3260 ESTABLISHED 3097/iscsid
```

En este caso la IP de su servidor de ficheros es por tanto : 91.121.191.16

La regla a añadir será :

```
# /sbin/iptables -A INPUT -i eth0 -p tcp --source 91.121.191.16 -j ACCEPT
```

Ejemplo de configuración completa

Aquí tiene un ejemplo de script completo para proteger su servidor mediante iptables. Esta operación está ilustrada con la Release2 sobre un servidor Superplan.

Es bastante permisivo en el sentido en que la mayoría de servicios presentes sobre su máquina son accesibles pero puede servir de base para su propia configuración :

```
/sbin/iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 25 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 53 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p udp --dport 53 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 110 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 443 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 10000 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 21 --source xx.xx.xx.xx -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 22 --source cache.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 22 --source xx.xx.xx.xx -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.p19.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.rbx.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.rbx2.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source ping.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source AAA.BBB.CCC.250 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source AAA.BBB.CCC.251 -j ACCEPT
```

OVH

```
/sbin/iptables -A INPUT -i eth0 -p tcp --source 192.168.0.0/16 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p udp --source 192.168.0.0/16 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 79 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -j REJECT
```

En todas estas reglas, hay que reemplazar :

- **AAA.BBB.CCC.DDD** por la dirección IP de su servidor.
- **xxx.xxx.xxx.xxx** por la dirección de su conexión fija (si existe)

Automatizar el firewall

Una vez su firewall está perfectamente configurado, únicamente tiene que crear un script que se lanzará cada reinicio de su servidor.

Esta operación está ilustrada con la Release2 sobre un servidor Superplan.

Aquí un ejemplo que se tiene que colocar en un fichero nombrado por ejemplo "firewall" en la carpeta /etc/init.d/ :

```
#!/bin/sh
# chkconfig: 3 21 91
# description: Firewall
```

```
IPT=/sbin/iptables
```

```
case "$1" in
start)
$IPT -F INPUT
$IPT -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 25 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 53 -j ACCEPT
$IPT -A INPUT -i eth0 -p udp --dport 53 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 110 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 443 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 10000 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 21 --source xx.xx.xx.xx -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 22 --source cache.ovh.net -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 22 --source xx.xx.xx.xx -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp --source proxy.ovh.net -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp --source proxy.p19.ovh.net -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp --source proxy.rbx.ovh.net -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp --source proxy.rbx2.ovh.net -j ACCEPT
```

OVH

```
$IPT -A INPUT -i eth0 -p icmp --source AAA.BBB.CCC.250 -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp --source AAA.BBB.CCC.251 -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp --source ping.ovh.net -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --source 192.168.0.0/16 -j ACCEPT
$IPT -A INPUT -i eth0 -p udp --source 192.168.0.0/16 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 79 -j ACCEPT
$IPT -A INPUT -i eth0 -j REJECT
exit 0
;;

stop)
$IPT -F INPUT
exit 0
;;
*)
echo "Usage: /etc/init.d/firewall {start|stop}"
exit 1
;;
esac
```

Dele los derechos 700 y escriba `"/etc/init.d/firewall start"` para el reinicio y `"/etc/init.d/firewall stop"` para pararlo.

Para reiniciarlo automáticamente en el reinicio :

```
# /sbin/chkconfig --level 3 firewall on
# /sbin/chkconfig --level 06 firewall off
```

Verifique que el Firewall sea bueno antes de meter el script en cada reinicio del servidor sino su servidor quedará definitivamente bloqueado

La comunicación entre el servicio RTM y su servidor necesita que se autoricen las conexiones entrantes salientes sobre los puertos UDP 6100 a 6200.

En caso de problemas

En caso de que, a pesar de tomar todas las precauciones sufra algún problema de acceso, deberá desactivar el Firewall a través del modo Rescate Rescue-pro o V kvm.

Más información

: ModoRescue :: Reiniciar su servidor en modo de rescate

: ModoKvm :: Reiniciar su servidor en modo V kvm

: NetBoot :: ¿Como seleccionar un kernel mediante Netboot?

: InstalarServidorIrc :: Instalación de IRCd y apertura de puertos en el Manager

: RebootDeLaMaquina :: ¿Cómo puedo reinicializar mi servidor?