Menu Principal
- Home
- Servicios
- Empresa
- Actualidad
- Web 2.0
- Soporte
- Portfolio Web
- Monitorización
- Contactar
- Links

Tags in Keywords

IPTABLES HOWTO: Ejemplos de iptables para Sysadmins

Friday, 10 January 2014 15:46 in **Documentacion**, **firewall**, **iptables**, **Linux**

VERSION ANALIZADA: iptables 1.4.7-1.4.12 (iptables -V)

Si no sabes **que es Iptables** o quieres una introducción sobre que es y como utilizarlo, puedes consultar la siguiente guía
IPTABLES HOWTO: Guia iptables para Sysadmins

Para poder trabajar con iptables, necesitamos privilegios de root.
En muchos sistemas, por defecto, se encuentra en /sbin/iptables o /usr/sbin/iptables

Ejemplos de reglas para IPTABLES:

cache CDN Cloud

community_management drupal

drupal6 hosting Linux

Rendimiento sistemas

virtualizacion webs

more tags

Reset iptables firewall rules

```
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -F
iptables -t nat -F
```

Ver el estado de tu firewall y listar reglas

Type the following command as root:

```
iptables -L -n -v
iptables -n -L -v --line-numbers

iptables -L INPUT -n -v
iptables -L OUTPUT -n -v --line-numbers
```

Eliminar reglas de Firewall

```
iptables -L INPUT -n --line-numbers
iptables -L OUTPUT -n --line-numbers
iptables -L OUTPUT -n --line-numbers | less
iptables -L OUTPUT -n --line-numbers | grep 202.54.1.1
Se puede eliminar por número de regla
iptables -D INPUT 4
O especificar una regla que coincida
iptables -D INPUT -s 202.54.1.1 -j DROP
```

Establecer las políticas de firewall por defecto

```
Para eliminar todo el tráfico:
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
iptables -L -v -n
```

Sólo a bloquear el tráfico entrante

```
Para eliminar todos los paquetes entrantes / enviado, pero permitir el tráfico saliente, escriba:
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -L -v -n
```

Eliminar direcciones de red privada en la interfaz pública (Suponiendo eth1 como interfaz pública)

```
iptables -A INPUT -i eth1 -s 192.168.0.0/24 -j DROP
iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j DROP
```

Bloquear IP:

```
iptables -A INPUT -s 192.168.1.5 -j DROP
iptables -A INPUT -s 192.114.71.13 -j DROP
```

Guardar reglas:

```
iptables-save > /root/iptables.fw
```

To save firewall rules under CentOS / RHEL / Fedora Linux, enter:

```
service iptables save
```

Restaurar reglas:

```
iptables-restore < /root/iptables.fw
```

To restore firewall rules under CentOS / RHEL / Fedora Linux, enter:

```
service iptables restart
```

Una forma de recargar reglas automáticamente al reiniciar el servidor:

vi /etc/rc.local

```
...
# Reload IPTABLES Rules automatically on restart
/sbin/iptables-restore < /root/iptables.fw
```

En CentOS pueden almacenarse las reglas en el siguiente fichero para recargarlas automáticamente:

/etc/sysconfig/iptables

Permitir todo el tráfico de loopback, y eliminar todo el tráfico a 127/8 que no usa lo0

```
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT ! -i lo -d 127.0.0.0/8 -j REJECT
```

Eliminar cualquier paquete TCP que no se ha iniciado con el Flag SYN activo

```
iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
```

Aceptar todas las conexiones entrantes establecidas

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Eliminar cualquier paquete inválido que no pueda ser identificado

```
iptables -A INPUT -m state --state INVALID -j DROP
```

Drop all inbound telnet traffic

```
iptables -I INPUT -p tcp --dport 23 -j DROP
```

Drop all outbound web traffic

```
iptables -I OUTPUT -p tcp --dport 80 -j DROP
```

Drop all outbound traffic to 192.168.0.1

```
iptables -I OUTPUT -p tcp --dest 192.168.0.1 -j DROP
```

Allow all inbound web traffic

```
iptables -I INPUT -p tcp --dport 80 -j ACCEPT
```

Allow inbound port traffic for localhost

```
iptables -I INPUT -s 12.0.0.1 -p tcp --dport 2003 -j ACCEPT
```

Allow inbound HTTPS traffic from 10.2.2.4

```
iptables -I INPUT -s 10.2.2.4 -p tcp -m tcp --dport 443 -j DROP
```

Deny outbound traffic to 192.2.4.0-192.2.4.255

```
iptables -I OUTPUT -d 192.2.4.6.0/24 -j DROP
```

Bloquear tráfico al dominio facebook.com

Primero miramos la IP de facebook.com y luego su RANGO DE IPs

```
host -t a www.facebook.com
whois 69.171.228.40 | grep CIDR
```

Creamos la regla para no permitir acceso

```
iptables -A OUTPUT -p tcp -d 69.171.224.0/19 -j DROP
```

También es posible hacerlo por nombre de dominio, pero la ayuda ya dice que es una mala idea ;)

```
You can also use domain name, enter:
iptables -A OUTPUT -p tcp -d www.facebook.com -j DROP
iptables -A OUTPUT -p tcp -d facebook.com -j DROP
```

IPTABLES man page:

```
"... specifying any name to be resolved with a remote query such as DNS (e.g., facebook.com is a really
bad idea), a network IP address (with /mask), or a plain IP address ..."
```

Allow incoming connections to port 21 from one IP address 11.22.33.44

```
iptables -A INPUT -p tcp -m state --state NEW --dport 21 --source 11.22.33.44
```

Permitir todo el tráfico entrante SSH

```
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
```

```
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

Permitir SSH saliente

```
iptables -A OUTPUT -o eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

Permitir tráfico HTTP

```
iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

Permitir HTTPS saliente.

```
iptables -A OUTPUT -o eth0 -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
```

Combinar reglas con multipuerto

```
iptables -A INPUT -i eth0 -p tcp -m multiport --dports 22,80,443 -m state --state NEW,ESTABLISHED -j
ACCEPT
iptables -A OUTPUT -o eth0 -p tcp -m multiport --sports 22,80,443 -m state --state ESTABLISHED -j ACCEPT
```

Deny all other incoming connections to port 21.

```
iptables -A INPUT -p tcp -m state --state NEW --dport 21 -j DROP
```

Eliminar la primera regla de entrada

```
iptables -D INPUT 1
```

Bloquear o Permitir solicitud de ICMP Ping

```
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
iptables -A INPUT -i eth1 -p icmp --icmp-type echo-request -j DROP
iptables -A INPUT -s 192.168.1.0/24 -p icmp --icmp-type echo-request -j ACCEPT
```

A continuación sólo acepta tipo limitado de peticiones ICMP:

```
### ** assumed that default INPUT policy set to DROP ** #############
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
iptables -A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
iptables -A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
## ** all our server to respond to pings ** ##
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

Permitir ping de fuera hacia adentro

```
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

Permitir PING de dentro a fuera.

```
iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

Permitir tráfico DNS saliente.

```
iptables -A OUTPUT -p udp -o eth0 --dport 53 -j ACCEPT
iptables -A INPUT -p udp -i eth0 --sport 53 -j ACCEPT
```

Abrir un rango de puertos de entrada

```
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 7000:7010 -j ACCEPT
```

Abrir un rango de IPs

```
## only accept connection to tcp port 80 (Apache) if ip is between 192.168.1.100 and 192.168.1.200 ##
iptables -A INPUT -p tcp --destination-port 80 -m iprange --src-range 192.168.1.100-192.168.1.200 -j
ACCEPT
```

```
## nat example ##
iptables -t nat -A POSTROUTING -j SNAT --to-source 192.168.1.20-192.168.1.25
```

Restringir el número de conexiones concurrentes a un servidor por IP

Puedes utilizar el módulo connlimit para crear estas restricciones. Permitir 3 conexiones SSH por cliente:

```
iptables -A INPUT -p tcp --syn --dport 22 -m connlimit --connlimit-above 3 -j REJECT
```

Limitar el número de conexiones HTTP a 20:

```
iptables -p tcp --syn --dport 80 -m connlimit --connlimit-above 20 --connlimit-mask 24 -j DROP
```

Donde:

--connlimit-above 3 : Match if the number of existing connections is above 3.

--connlimit-mask 24 : Group hosts using the prefix length. For IPv4, this must be a number between (including) 0 and 32.

Eliminar o aceptar paquetes desde una MAC Address

```
iptables -A INPUT -m mac --mac-source 00:0F:EA:91:04:08 -j DROP
## *only accept traffic for TCP port # 22 from mac 00:0F:EA:91:04:07 * ##
iptables -A INPUT -p tcp --destination-port 22 -m mac --mac-source 00:0F:EA:91:04:07 -j ACCEPT
```

Prevenir ataques DoS

-Limit 25/minute : Limita a sólo 25 conexiones por minuto.

-Limit-burst 100: Indica que el valor de limit/minute será forzado sólo después del número de conexiones en este nivel

```
iptables -A INPUT -p tcp --dport 80 -m limit --limit 25/minute --limit-burst 100 -j ACCEPT
```

Permitir redirección de puertos.

Ejemplo puerto 422 redirigimos al 22, con lo que podemos tener conexiones al puerto 22 y al puerto 422 por ssh.

```
iptables -t nat -A PREROUTING -p tcp -d 192.168.102.37 --dport 422 -j DNAT --to 192.168.102.37:22
iptables -A INPUT -i eth0 -p tcp --dport 422 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 422 -m state --state ESTABLISHED -j ACCEPT
```

Logar y eliminar paquetes:

```
iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j LOG --log-prefix "IP_SPOOF A: "
iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j DROP
```

Logar y eliminar paquetes limitando el número de entradas repetidas en el LOG

```
iptables -A INPUT -i eth1 -s 10.0.0.0/8 -m limit --limit 5/m --limit-burst 7 -j LOG --log-prefix "IP_SPOOF
A: "
iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j DROP
```

Sacar en el log los paquetes caídos.

Primero creamos una cadena llamada LOGGING:

```
iptables -N LOGGING
```

Luego todas las conexiones entrantes vayan por la cadena LOGGING

```
iptables -A INPUT -j LOGGING
```

Logueamos paquetes con un log-prefix

```
iptables -A LOGGING -m limit --limit 2/min -j LOG --log-prefix "IPTables Packet Dropped: " --log-level 7
```

Y los bloqueamos:

```
iptables -A LOGGING -j DROP
```

Listado de reglas ejemplo para puertos TCP/UDP comunes:

```
Replace ACCEPT with DROP to block port:
## open port ssh tcp port 22 ##
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 22 -j ACCEPT

## open cups (printing service) udp/tcp port 631 for LAN users ##
iptables -A INPUT -s 192.168.1.0/24 -p udp -m udp --dport 631 -j ACCEPT
iptables -A INPUT -s 192.168.1.0/24 -p tcp -m tcp --dport 631 -j ACCEPT

## allow time sync via NTP for lan users (open udp port 123) ##
iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 123 -j ACCEPT

## open tcp port 25 (smtp) for all ##
iptables -A INPUT -m state --state NEW -p tcp --dport 25 -j ACCEPT

# open dns server ports for all ##
iptables -A INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT
iptables -A INPUT -m state --state NEW -p tcp --dport 53 -j ACCEPT

## open http/https (Apache) server port to all ##
iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT

## open tcp port 110 (pop3) for all ##
iptables -A INPUT -m state --state NEW -p tcp --dport 110 -j ACCEPT

## open tcp port 143 (imap) for all ##
iptables -A INPUT -m state --state NEW -p tcp --dport 143 -j ACCEPT

## open access to Samba file server for lan users only ##
```

```
iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 137 -j ACCEPT
iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 138 -j ACCEPT
iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 139 -j ACCEPT
iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 445 -j ACCEPT

## open access to proxy server for lan users only ##
iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 3128 -j ACCEPT

## open access to mysql server for lan users only ##
iptables -I INPUT -p tcp --dport 3306 -j ACCEPT
```

Conclusión:

Esta entrada es una lista básica de reglas para iptables. Puedes crear y construir reglas más complejas a partir de estas, e incluso crear sistemas para automatizar la adición o eliminación de reglas.

Para crear las reglas, es importante comprender y conocer bien TCP/IP.

Puede ser importante o necesario aplicar ciertas modificaciones al kernel mediante sysctl.conf

Documentos Relacionados

IPTABLES HOWTO: Guia y documentación iptables para Sysadmins

Como crear ScreenCast en Ubuntu y conversion de video

SERVICIOS Y NOVEDADES

- SEAVTEC: IT Cloud - Cloud Dedicado Dinámico
- Arquitectura de Sistemas - Servicios IT
- Sistemas y Soporte
- Auditoria de rendimiento web
- Monitorizacion

NOTICIAS

- SEAVTEC se convierte en OVH Web Community Partner
- Nueva Infraestructura en el Cloud de SEAVTEC

DOCUMENTACIÓN RECIENTE

- XEN: Excluir disco de máquinas virtuales en backups con Xen (XenServer y XCP)
- IPTABLES HOWTO: Ejemplos de iptables para Sysadmins
- IPTABLES HOWTO: Guia y documentación iptables para Sysadmins
- Apache2 - Eliminar modulos no utilizados

mdadm: Comandos de Ayuda - mdadm CheatSheet

1 2 3 4 5 next › last »

more

Monthly archive

November 2014 (1)

August 2014 (1)

February 2014 (1)

January 2014 (3)

August 2013 (4)

July 2013 (4)

more

Portfolio

Respira mejor - Fisioterapia Respiratoria

Pilar Puyol es una **Fisioterpeuta Diplomada** especializada en el tratamiento infantil de dolencias relacionadas con el aparato respiratorio, una técnica que disminuye la medicación de los niños y que no solo se limita a la aplicación del tratamiento, sino también la educación sanitaria tanto a padres, escuela,...

more