

## Práctica 08: Firewalls

Rafael García Damián  
13103591

Carmona Mendoza Martín  
313075977

Barajas Figueroa José de Jesús  
314341015

Vazquez Aguilar Lisandro  
314272117

08/Febrero/2019

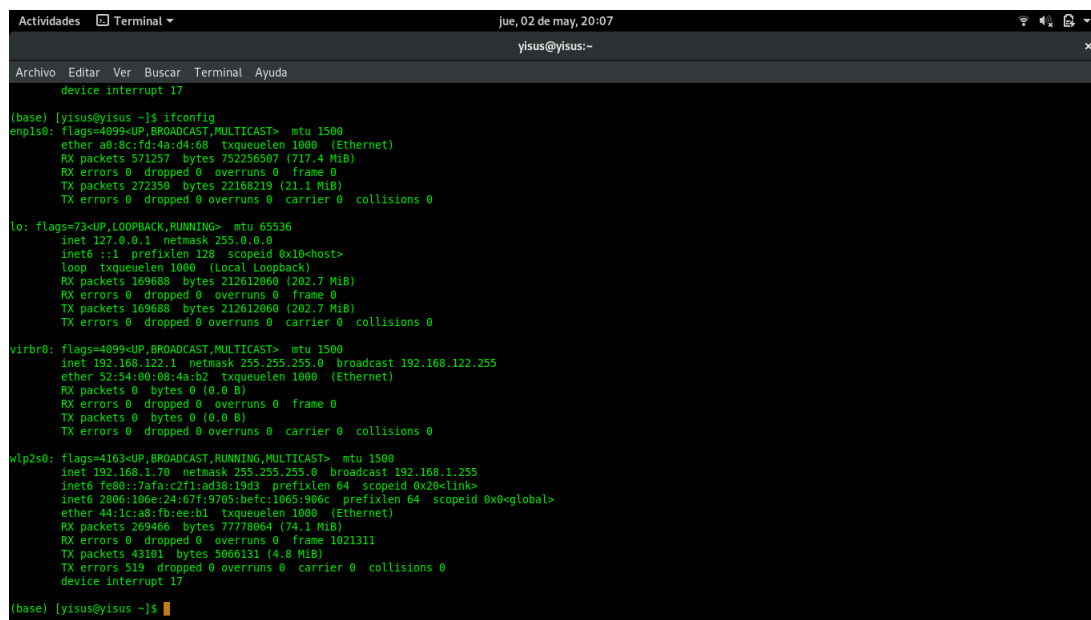
# 1 Practica Firewalls

Los firewalls son una herramienta con amplio uso en el presente, sobre todo en el campo laboral para la protección de recursos, y la limitación a la navegación tanto interna como el tráfico del exterior. De manera literal lo podemos ver como un punto de acceso para todo el tráfico en red interna, tanto el que sale como el que fluctúa dentro de la red. Para esta práctica trabajaremos con 3 firewalls, Shorewall, ufw y firewalld.

## 1.1 Shorewall

*Shorewall es un software que te facilita generar las reglas de configuración del netfilter. Es un conjunto de ficheros que se utiliza para configurar y controlar los paquetes del núcleo Linux.*

- Verificamos el estado de las redes con el comando `ifconfig`



```
Actividades Terminal
jue, 02 de may, 20:07
yisus@yisus:~

(base) [yisus@yisus ~]$ ifconfig
enp1s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        ether ab:8c:fd:4a:d4:d0 txqueuelen 1000 (Ethernet)
        RX packets 571257 bytes 752256507 (717.4 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 272350 bytes 22168219 (21.1 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 169688 bytes 212612060 (202.7 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 169688 bytes 212612060 (202.7 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
        ether 52:54:00:08:4a:b2 txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.70 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::7afa:c2f1:ad38:19d3 prefixlen 64 scopeid 0x20<link>
        inet6 2006:1006:24:67f:9705:befc:1065:906c prefixlen 64 scopeid 0x0<global>
        ether 44:1c:a8:fbee:b1 txqueuelen 1000 (Ethernet)
        RX packets 269466 bytes 77778064 (74.1 MiB)
        RX errors 0 dropped 0 overruns 0 frame 1021311
        TX packets 43101 bytes 5066131 (4.8 MiB)
        TX errors 519 dropped 0 overruns 0 carrier 0 collisions 0
        device interrupt 17

(base) [yisus@yisus ~]$
```

- Primero instalamos shorewall, para distribuciones basadas en RedHat, como Fedora o CentOS, basta con este comando `sudo yum -y install shorewal`
- Posteriormente realizaremos la instalación de unas políticas para SELinux, debido a que su funcionamiento podría interferir con nuestro cometido mediante Shorewall, necesitaremos el siguiente comando. `sudo yum -y install policycoreutils-python`
- Vamos a crear un directorio llamado `shorewall2`, lo crearemos con el comando `/usr/share/selinux/packages/shorewall2`, nos cambiamos a ese directorio con el comando `cd`.
- Vamos a descargar el archivo Shorewall2, utilizaremos el siguiente comando `wget http://www.alcancellibre.org/linux/secre` vemos el contenido del archivo y se tiene que ver algo así.
- Ahora debemos crear un módulo `shorewall2.mod`. después una política `shorerwall2.pp` e incluir la política, esto lo conseguimos ejecutando estos comandos secuencialmente en orden.

```

Actividades Terminal
jue, 02 de may, 20:28
yisus@yisus:/etc/rc.d

Archivo Editar Ver Buscar Terminal Ayuda
(8/8): python2-ipaddress-1.0.18-5.fc29.noarch.rpm 12 KB/s | 39 kB 00:03
-----
Total 106 KB/s | 1.2 MB 00:12
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
Preparando : 1/1
Instalando : python2-libselinux-2.8-6.fc29.x86_64 1/8
Instalando : python2-ipaddress-1.0.18-5.fc29.noarch 2/8
Instalando : python2-enum34-1.1.6-7.fc29.noarch 3/8
Instalando : python2-setools-4.1.1-13.fc29.x86_64 4/8
Instalando : python2-IPy-0.81-23.fc29.noarch 5/8
Instalando : python2-libsemanage-2.8-8.fc29.x86_64 6/8
Instalando : python2-audit-3.0-0.7.20190326git03e7489.fc29.x86_64 7/8
Instalando : python2-policycoreutils-2.8-17.fc29.noarch 8/8
Ejecutando scriptlet: python2-policycoreutils-2.8-17.fc29.noarch 8/8
Verificando : python2-audit-3.0-0.7.20190326git03e7489.fc29.x86_64 1/8
Verificando : python2-libselinux-2.8-6.fc29.x86_64 2/8
Verificando : python2-libsemanage-2.8-8.fc29.x86_64 3/8
Verificando : python2-policycoreutils-2.8-17.fc29.noarch 4/8
Verificando : python2-IPy-0.81-23.fc29.noarch 5/8
Verificando : python2-enum34-1.1.6-7.fc29.noarch 6/8
Verificando : python2-ipaddress-1.0.18-5.fc29.noarch 7/8
Verificando : python2-setools-4.1.1-13.fc29.x86_64 8/8

Instalado:
python2-policycoreutils-2.8-17.fc29.noarch
python2-audit-3.0-0.7.20190326git03e7489.fc29.x86_64
python2-libselinux-2.8-6.fc29.x86_64
python2-libsemanage-2.8-8.fc29.x86_64
python2-IPy-0.81-23.fc29.noarch
python2-enum34-1.1.6-7.fc29.noarch
python2-ipaddress-1.0.18-5.fc29.noarch
python2-setools-4.1.1-13.fc29.x86_64

[base] [yisus@yisus rc.d]$ sudo yum -y install policycoreutils-python

```

- *checkmodule -M -m -o shorewall2.mod shorewall2.te*
- *semodule-package -o shorewall2.pp -m shorewall2.mod*
- *semodule -i /usr/share/selinux/packages/shorewall2/shorewall2.pp*

- usamos el comando *ls / sys/class/net* para ver como estas configurados los nombres de los dispositivos de red, una vez terminado esto, empezamos con la configuracion de Shorewall, seguiremos la guia en la referencia
- En esta parte de la configuracion podemos configuras las ip o segmento de ip betadas de nuestro fire-wall, en este caso se toman las IP del ejemplo en la bibliografia. Podemos modificar el siguiente archivo */etc/shorewall/blrules*, despues iniciamos el servicio con la instrucción, *sudo systemctl start shorewall*
- El servicio tiene varias instrucciones, puede ser restart o stop.

## 1.2 ufw

UFW viene ya pre-instalado en Ubuntu y en todas sus ediciones y sabores (de hecho, incluso Debian o Linux Mint cuentan con él). Sin embargo, en Ubuntu viene deshabilitado por defecto (supongo que lo justifican por el hecho de que, por defecto, Ubuntu no viene con ningún puerto abierto al exterior). Es por eso que una de las primeras cosas que suelo hacer en cualquier nueva instalación de Ubuntu es habilitar UFW.

- Primero debemos comprobar si UFW está activo:
- Como no lo está debemos activarlo con el siguiente comando:
- En general, la mejor política de un firewall es la de denegar todas las conexiones entrantes por defecto, y a partir de ahí, y en caso de que sea estrictamente necesario, establecer las excepciones que te convengan dependiendo de cada caso. Por defecto UFW ya debería venir con la política de denegar todas las conexiones entrantes. Pero para asegurarnos de que cierto, ponemos en la consola:

Hecho todos los pasos anteriores queda activado Netfilter/Iptables en el kernel de Linux, y configurado para denegar todo el tráfico entrante por defecto, lo que significa que desde el exterior, nuestro sistema operativo mantendrá todos los puertos invisibles e inaccesibles.

```

Actividades Terminal
jue, 02 de may, 20:30
yisus@yisus:/usr/share/selinux/packages/shorewall2

Archivo Editar Ver Buscar Terminal Ayuda

Instalando : python2-libselinux-2.8-6.fc29.x86_64 1/8
Instalando : python2-ipaddress-1.0.18-5.fc29.noarch 2/8
Instalando : python2-enum34-1.1.6-7.fc29.noarch 3/8
Instalando : python2-setools-4.1.1-13.fc29.x86_64 4/8
Instalando : python2-IPy-0.81-23.fc29.noarch 5/8
Instalando : python2-libsemanage-2.8-8.fc29.x86_64 6/8
Instalando : python2-audit-3.0-8.7.20190326git03e7489.fc29.x86_64 7/8
Instalando : python2-policycoreutils-2.8-17.fc29.noarch 8/8
Ejecutando scriptlet: python2-policycoreutils-2.8-17.fc29.noarch 8/8
Verificando : python2-audit-3.0-8.7.20190326git03e7489.fc29.x86_64 1/8
Verificando : python2-libselinux-2.8-6.fc29.x86_64 2/8
Verificando : python2-libsemanage-2.8-8.fc29.x86_64 3/8
Verificando : python2-policycoreutils-2.8-17.fc29.noarch 4/8
Verificando : python2-IPy-0.81-23.fc29.noarch 5/8
Verificando : python2-enum34-1.1.6-7.fc29.noarch 6/8
Verificando : python2-ipaddress-1.0.18-5.fc29.noarch 7/8
Verificando : python2-setools-4.1.1-13.fc29.x86_64 8/8

Instalado:
python2-policycoreutils-2.8-17.fc29.noarch
python2-audit-3.0-8.7.20190326git03e7489.fc29.x86_64
python2-libselinux-2.8-6.fc29.x86_64
python2-libsemanage-2.8-8.fc29.x86_64
python2-IPy-0.81-23.fc29.noarch
python2-enum34-1.1.6-7.fc29.noarch
python2-ipaddress-1.0.18-5.fc29.noarch
python2-setools-4.1.1-13.fc29.x86_64

¡Listo!
(base) [yisus@yisus rc.d]$ mkdir /usr/share/se
seabios/ seabios/ services/ setroubleshoot/
seahorse/ selinux/ servicetypes/ setupptool/
(base) [yisus@yisus rc.d]$ mkdir /usr/share/selinux/packages/shorewall2
mkdir: no se puede crear el directorio «/usr/share/selinux/packages/shorewall2»: Permission denied
(base) [yisus@yisus rc.d]$ sudo mkdir /usr/share/selinux/packages/shorewall2
[sudo] password for yisus:
(base) [yisus@yisus rc.d]$ sudo cd /usr/share/selinux/packages/shorewall2
(base) [yisus@yisus rc.d]$ cd /usr/share/selinux/packages/shorewall2
(base) [yisus@yisus shorewall2]$

```

```

Actividades Emacs
jue, 02 de may, 20:36
yisus@yisus:/usr/share/selinux/packages/shorewall2
emacs@yisus

Archivo Editar Ver Buscar Terminal Ayuda

¡Listo!
(base) [yisus@yisus rc.d]$ mkdir /usr/share/se
seabios/ seabios/ services/ setroubleshoot/
seahorse/ selinux/ servicetypes/ setupptool/
(base) [yisus@yisus rc.d]$ mkdir /usr/share/selinux/packages/shorewall2
mkdir: no se puede crear el directorio «/usr/share/selinux/packages/shorewall2»: Pe
rmission denied
(base) [yisus@yisus rc.d]$ sudo mkdir /usr/share/selinux/packages/shorewall2
[sudo] password for yisus:
(base) [yisus@yisus rc.d]$ sudo cd /usr/share/selinux/packages/shorewall2
(base) [yisus@yisus rc.d]$ cd /usr/share/selinux/packages/shorewall2
(base) [yisus@yisus shorewall2]$ wget http://www.alcancellibre.org/linux/secrets/sho
rewall2.te
--2019-05-02 20:35:23-- http://www.alcancellibre.org/linux/secrets/shorewall2.te
Resolviendo www.alcancellibre.org (www.alcancellibre.org)... 189.211.120.203
Conectando con www.alcancellibre.org (www.alcancellibre.org)[189.211.120.203]:80... c
onectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 406 [text/plain]
shorewall2.te: Permission denied

No se puede escribir a "shorewall2.te" (Permission denied).
(base) [yisus@yisus shorewall2]$ sudo wget http://www.alcancellibre.org/linux/secre
t/s/shorewall2.te
[sudo] password for yisus:
--2019-05-02 20:35:34-- http://www.alcancellibre.org/linux/secrets/shorewall2.te
Resolviendo www.alcancellibre.org (www.alcancellibre.org)... 189.211.120.203
Conectando con www.alcancellibre.org (www.alcancellibre.org)[189.211.120.203]:80... c
onectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 406 [text/plain]
Grabando a: "shorewall2.te"

shorewall2.te 100%[=====] 406 --.-KB/s en 0s

2019-05-02 20:35:34 (13.0 MB/s) - "shorewall2.te" guardado [406/406]

(base) [yisus@yisus shorewall2]$ emacs shorewall2.te

```

```

emacs@yisus
File Edit Options Buffers Tools Help

module shorewall2 1.0;

require {
    type shorewall_t;
    type usr_t;
    type sysfs_t;
    class file { execute execute_no_trans };
    class dir search;
    class dir getattr;
    class process signal;
}

#===== shorewall_t =====
allow shorewall_t usr_t:file { execute execute_no_trans };
allow shorewall_t sysfs_t:dir search;
allow shorewall_t sysfs_t:dir getattr;
allow shorewall_t self:process signal;

-:%%- shorewall2.te All (1,0) (Fundamental)
Note: file is write protected

```

Figure 1:

- Para abrir puertos se utiliza la instrucción allow y hay dos maneras de hacerlo: La primera de ellas es mediante el nombre de servicio (como se muestra en el primer comando) y por numero de puerto (como en le segundo comando).

```
Actividades Terminal ▼ jue, 02 de may, 21:35
yisus@yisus:/usr/share/selinux/packages/shorewall2

Archivo Editar Ver Buscar Terminal Ayuda
File Edit Options Buffers Tools Help
#
# Shorewall -- /etc/shorewall/blrules
#
# For information about entries in this file, type "man shorewall-blrules"
#
# Please see http://shorewall.net/blacklisting_support.htm for additional
# information.
#
#####
#ACTION      SOURCE      DEST      PROTO  DPORT  SPORT  ORIGDEST  RATE  USER  MARK  CONNLIMIT  TIME  HEADERS  SWITCH  HELP
DROP         net:197.0.0.0/8    all
DROP         net:92.241.160.0/19 all
DROP         net:91.144.176.0/22 all
DROP         net:212.191.0.0/17 all
DROP         net:79.171.80.0/21 all

-UU-:***-F1 blrules All L15 (Fundamental) -----
End of buffer
```

Figure 2:

```
Tilix: Por defecto
1: ▼
mcfly@mcfly-HP-Pavilion-g4-Notebook-PC:~$ sudo ufw status
Estado: inactivo
mcfly@mcfly-HP-Pavilion-g4-Notebook-PC:~$
```

Figure 3:

```
Tilix: Por defecto
1: ▼
mcfly@mcfly-HP-Pavilion-g4-Notebook-PC:~$ sudo ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
mcfly@mcfly-HP-Pavilion-g4-Notebook-PC:~$
```

Figure 4:

Estos comandos permite a todas las conexiones entrantes de HTTP.

```
Tilix: Por defecto
1: ▾
mcfly@mcfly-HP-Pavilion-g4-Notebook-PC:~$ sudo ufw default deny
La política incoming predeterminada cambió a «deny»
(asegúrese de actualizar sus reglas consecuentemente)
mcfly@mcfly-HP-Pavilion-g4-Notebook-PC:~$
```

Figure 5:

```
Tilix: Por defecto
1: ▾
mcfly@mcfly-HP-Pavilion-g4-Notebook-PC:~$ sudo ufw allow http
[sudo] contraseña para mcfly:
Regla añadida
Regla añadida (v6)
mcfly@mcfly-HP-Pavilion-g4-Notebook-PC:~$ sudo ufw allow 80
Regla añadida
Regla añadida (v6)
mcfly@mcfly-HP-Pavilion-g4-Notebook-PC:~$
```

Figure 6:

- Para cerrar los puertos se utiliza la instrucción deny y funciona igual que allow.

```
Tilix: Por defecto
1: ▾
mcfly@mcfly-HP-Pavilion-g4-Notebook-PC:~$ sudo ufw deny http
Regla actualizada
Regla actualizada (v6)
mcfly@mcfly-HP-Pavilion-g4-Notebook-PC:~$ sudo ufw deny 80
Regla actualizada
Regla actualizada (v6)
mcfly@mcfly-HP-Pavilion-g4-Notebook-PC:~$
```

Figure 7:

### 1.3 Firewalld

Un firewall es un dispositivo de seguridad de la red que monitorea el tráfico de red entrante y saliente y decide si permite o bloquea tráfico específico en función de un conjunto definido de reglas de seguridad.

Para configurar directamente el firewall, linux tiene cuenta con iptables, pero a veces es un poco complicado usar esta herramienta y no tiene una gran versatilidad de operaciones administrativas, para esto nos sirve firewalld.

#### Firewalld

se podría decir que es un controlador de iptables que hace mas fácil la definición de de reglas para el tráfico

de la red, las principales diferencias entre definir las reglas directamente en iptables y definirlas en firewalld son:

- Firewalld usa zonas y servicios para definir y administrar las reglas a diferencia de cadenas y reglas que es lo que usa iptables y es más complicado.
- Firewalld maneja los conjuntos de reglas dinámicamente, es decir se pueden hacer modificaciones sin cerrar las sesiones y conexiones activas.

Ahora vamos a ver cómo usar Firewalld:

### Configuración:

Primero necesitamos activar el servicio con 'sudo systemctl start firewalld'.

Para configurarlo, los archivos de configuración por default se encuentran en '/usr/lib/Firewalld' pero para sobrescribir la configuración es preferible hacerlo en '/etc/firewalld'.

Firewalld usa dos tipos de configuraciones Runtime y Permanent, la diferencia es que el primero es temporal, cuando se reinicia el servicio se pierden estos cambios. Al usar el comando 'firewall-cmd' si le agregamos la bandera -permanent los cambios serán permanentes. Por ejemplo el siguiente comando:

```
'sudo firewall-cmd --zone=public --add-service=http --permanent'
```

Agregaré la regla de manera permanente, pero si queremos ver los cambios en runtime solo se le quita la bandera -permanent.

### Firewalld Zones:

Las zonas son conjuntos de reglas predefinidos para cierta locación o escenario, por ejemplo si estamos en una red de wi-fi abierto seria conveniente usar la zona public, pero si estamos en una red segura, en la que confiamos en las computadoras que están en la red, como por ejemplo en casa, se usaría la zona home. Y como estas hay más zonas y pueden ser definidas nuevas zonas.

Para ver la configuración de todas las zonas definidas usamos el comando:

```
'sudo firewall-cmd --list-all-zones'
```

Lo que nos daría un resultado como el siguiente:

```
external
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: yes
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

home
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh mdns samba-client dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

### Firewalld Services:

Firewalld se puede configurar para que permita el tráfico de red de determinados servicios de red, una manera de ver todos los servicios predefinidos es:

`'sudo firewall-cmd --get-services'`

```
n-droid@thnk ~  
└─> sudo firewall-cmd --get-services  
RH-Satellite-6 amanda-client amanda-k5-client bacula bacula-client bgp bitcoin bitcoin-rpc bitcoin-testnet  
bitcoin-testnet-rpc ceph ceph-mon cfengine condor-collector ctdb dhcp dhcpv6 dhcpv6-client dns docker-regis  
try docker-swarm dropbox-lansync elasticsearch freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust  
ftp ganglia-client ganglia-master git gre high-availability http https imap imaps ipp ipp-client ipsec irc  
ircs iscsi-target jenkins kadmin kerberos kibana klogin kpasswd kprop kshell ldap ldaps libvirt libvirt-tl  
s managesieve mdns minidlna mongodb mosh mountd ms-wbt mssql murmur mysql nfs nfs3 nmea-0183 nrpe ntp openv  
pn ovirt-imageio ovirt-storageconsole ovirt-vmconsole pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql  
privoxy proxy-dhcp ptp pulseaudio puppetmaster quassel radius redis rpc-bind rsh rsyncd samba samba-client  
sane sip sips smtp smtp-submission smtps snmp snmptrap spideroak-lansync squid ssh syncthing syncthing-gui  
synergy syslog syslog-tls telnet tftp tftp-client tinc tor-socks transmission-client upnp-client vdsms vnc-  
server wbem-https xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server
```

### Firewalld Ejemplos:

Aceptar o denegar un puerto con cierto protocolo:

`sudo firewall-cmd --zone=home --remove-port=22/tcp`

eso bloqueará la conexión con tcp en el puerto 22, lo que deshabilitará la conexión por ssh porque hace uso de ese puerto. Para habilitarlo usamos:

`sudo firewall-cmd --zone=home --add-port=22/tcp`

Firewalld tiene otro tipo de reglas que son mucho más específicas, se llaman rich rules. Por ejemplo, el siguiente comando:

`sudo firewall-cmd --zone=public --add-rich-rule 'rule family="ipv4" source address="192.168.1.10" port port=22 protocol=tcp reject'`

niega el acceso Ipv4 usando TCP del host 192.168.1.10 al puerto 22

## 1.4 Referencias

<http://configurarlinuxserver.com/instalarfirewallenlinuxserver.pdf>

<http://www.alcancellibre.org/staticpages/index.php/configuracion-basica-shorewall> - Guia Shorewall