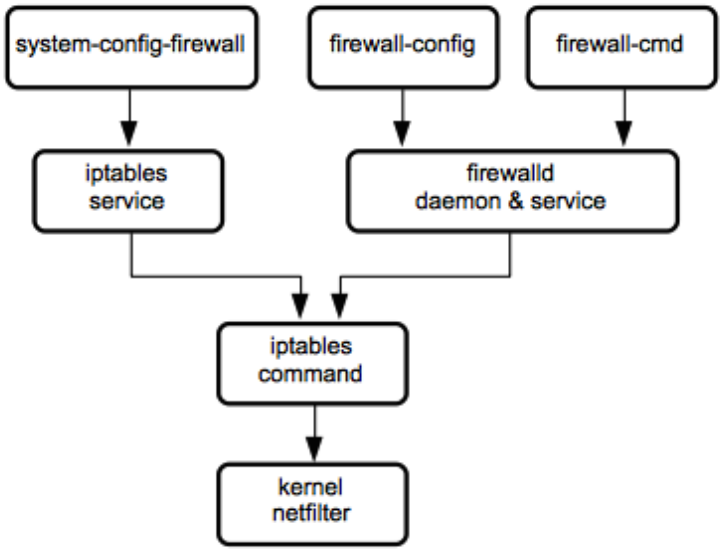
13. FIREWALL

13.1.- Introducción

El kernel de Linux incluye un subsistemas de filtrado de red llamado *netfilter*. Este subsistema permite al kernel examinar cada paquete de red que pasa por el sistema, ya sea para salir, entrar o ser reenviado. El propio kernel dedide si el paquete tiene que ser descartado, rechazado, reenviado con o sin *nat* o aceptado.

El comando *iptables* es una herramienta de bajo nivel que permite añadir reglas de cómo el kernel tiene que tratar esos paquetes que inspecciona. Sólo para IPv4, para IPv6 se utiliza *ip6tables* y para bridges *ebtables*.

FirewallD fue desarrollado (en Python) en 2011 para sustituir a las herramientas *iptables, ipt6tables y ebtables*, tan complicadas de gestionar. Viene instalado por defecto en CentOS7, RHEL7 y Fedora 18 (y posteriores) y está disponible en los repositorios públicos para el resto de distribuciones.



Interacción de los servicios FirewallD e Iptables con netfilter

FirewallD es un servicio que gestiona las reglas de iptables/netfilter en *runtime*. Puede ser gestionado con el comando *firewalld-cmd* o con la interfaz gráfica *firewall-config.*

También se puede configurar FirewallD modificando sus archivos de configuración de */etc/firewalld* pero está totalmente desaconsejado.

Si se va a usar Firewalld en un sistema, se utilizará el servicio *firewalld*, arrancándolo y activándolo, y entonces el servicio *iptables* tendrá que estar parado y desactivado. Incluso se enmascara para que no pueda ser iniciado.

13.2.- Funcionamiento de FirewallD

FirewallD clasifica el tráfico por zonas aplicando los siguientes criterios en orden:

- Se mira la dirección de origen del paquete, si está asignada a una determinada zona, se aplican las reglas de esa zona.
- Se mira por qué interfaz entró el paquete, si la interfaz está asociado a una zona, se aplican las reglas de la zona. Si la interfaz no está asociada a ninguna zona, se toma la zona por defecto.

13.2.1- Zonas por defecto de FirewallD

FirewallD utiliza zonas para segregar el tráfico de red que entra, sale o se reenvía del sistema. Viene por defecto configurado con una serie de zonas:

- *block:* rechaza todo el tráfico entrante.
- *dmz*: rechaza el trafico entrante salvo el de ssh.
- *drop*: descarta todo el tráfico entrante.
- *external*: rechaza el tráfico entrante salvo el del servicio ssh. El tráfico que se reenvía se enmascara para que parezca que fue originado desde la dirección IP de la interfaz que tiene esta zona configurada.
- *home*: rechaza el tráfico entrante salvo el de los servicios de *ssh, mdns, ipp-client, samba-client* o *dhcpv6-client*.
- *internal*: rechaza el tráfico entrante salvo el de los servicios de *ssh, mdns, ipp-client, samba-client* o *dhcpv6-client*.
- *public*: rechaza el tráfico entrante salvo el de los servicios de *ssh* o *dhcpv6*.
- *trusted*: permite todo el tráfico entrante.
- *work*: rechaza el tráfico entrante salvo el de los servicios de ssh, ipp-client o dhcpv6-client.

Todas las zonas permiten el tráfico entrante que proviene de una comunicación previamente establecida por el propio sistema y todo el tráfico saliente.

Para ver las zonas disponibles se utiliza el comando *firewall-cmd --get-zones* y para ver las zonas activas (las que tienen alguna interfaz asociada) con *firewall-cmd --get-active-zones*.

La zona *default*, no es una zona propiamente dicha, es una de las anteriores que se toma por defecto en el caso de crear una nueva interfaz de red. Por defecto está configurada a *public*. Para ver la zona por defecto se utiliza *firewall-cmd --get-default-zone* y para cambiarla *firewall-cmd --set-default-zone=<zona>*.

13.2.2.- Servicios predefinidos en FirewallD

Existen una serie de servicios predefinidos en *firewalld* que usan los puertos por defecto de los servicios, por ejemplo, para SSH existe un servicio en FirewallD llamado *ssh* que abriría el puerto 22/tcp.

La definición de estos servicios se puede encontrar en el directorio */usr/lib/firewalld* en los archivos xml. El formato de estos archivos se puede ver en la sección 5 del man *firewalld.zones.*

Se pueden reescribir estos xml, cambiando por ejemplo el número de puerto del SSH al 44, y situándolos en */etc/firewalld/services* . Los archivos del *etc* toman preferencia frente a los del *usr*.

Los servicios predefinidos se pueden ver con el comando *firewall-cmd --get-services.*

En algunas zonas de FirewallD, hay unos servicios configurados de forma predefinida:

- *ssh*: servidor local SSH; abre el tráfico del puerto *22/tcp*.
- *dhcpv6-client*: cliente local DHCPv6; abre el tráfico del puerto *546/udp* en la red *fe80::/64*.
- *ipp-client*: servicio local de impresión por Internet; tráfico del puerto *631/udp*.
- *samba-client*:cliente local SAMBA; tráfico de los puertos *137/udp* y *138/udp*.
- *mdns*: resolución de nombres local por multicast; tráfico del puerto *5353/udp* para las redes *224.0.0.251* y *ff02::fb*.

13.2.3.- Comando firewall-cmd para añadir y borrar reglas

Sintaxis del comando *firewall-cmd*:

```
firewall-cmd <opciones> <regla>
```

donde las opciones son:

--*zone*=<*zona*>: añade/elimina una regla sólo en la zona dada. Si se omite esta opción, la regla se añadirá en la zona *default*.

--*permanent*: la regla se añade/elimina en los archivos de configuración para que persista. No se aplica al *runtime*, si queremos que, además de persistir la regla, la añada/elimine en este momento, hay que ejecutar a continuación *firewall-cmd --reload*.

--*timeout*=<*segundos*>: para añadir/eliminar la regla sólo durante ese número de segundos.

Y donde las reglas se definen con:

--*add-source*=<*dirección*>: añade esa dirección IP o dirección de red.

--*remove-source*=<*dirección*>: borra esa dirección IP o dirección de red.

--*add-interface*=<*interfaz*>: añade la interfaz dada.

--change-interface=<interfaz>: modifica la zona de la interfaz dada.

--add-service=<service>: añade el servicio de FirewallD dado. Para poner varios servicios, el valor será *{<servicio1>,<servicio2>,...}*.

--remove-service=<service>: borra el servicio de FirewallD dado. Para poner varios servicios, el valor será *{<servicio1>,<servicio2>,...}*.

--add-port=<puerto/protocolo>: añade el puerto y protocolo dado.

--remove-port=<puerto/protocolo>: borra el puerto y protocolo dado.

13.2.4.- Otras sintaxis del comando firewall-cmd

Para cargar la configuración en *runtime*, llevar el *runtime* a la configuración y consultar cómo está configurado FirewallD:

- *firewall-cmd --reload*: carga las reglas persistidas en los archivos de configuración en el *runtime*, perdíendose aquellas que estuviesen en el *runtime* y no estuviesen persistidas.
- *firewall-cmd --runtime-to-permament*: guarda las reglas del *runtime* en los archivos de configuración de FirewallD para persistirlas.
- *firewall-cmd --list-all*: lista todo lo configurado en FirewallD: interfaces, orígenes, servicios y puertos. Si no se añade *--zone=<zona>*, se lista sólo lo de la zona *default*.
- *firewall-cmd --status*: ver si está funcionando FirewallD.

13.2.5.- Cambiar una interfaz de zona

Tenemos varias posibilidades, bien utilizando el comando *firewall-cmd*, con Network Manager o en la propia configuración de la interfaz:

1ª forma: utilizando FirewallD:

```
firewall-cmd --permament --zone=<zona> --change-interface=<interfaz>
firewall-cmd --reload
```

2ª forma: utilizando Network Manager:

```
nmcli con show | grep <interfaz>     → obtenemos el nombre de la conexión asociado
nmcli con mod <conexión> connection.zone <zona>
nmcli con reload && nmcli con up <conexión>
```

3ª forma: utilizando los archivos de configuración de la interfaz:

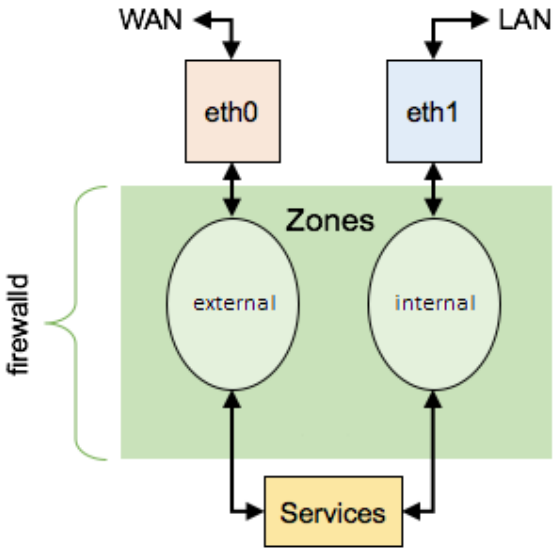
Editar el archivo /etc/sysconfig/network-scripts/ifcfg-<interfaz>

parámetro ZONE=<zona>

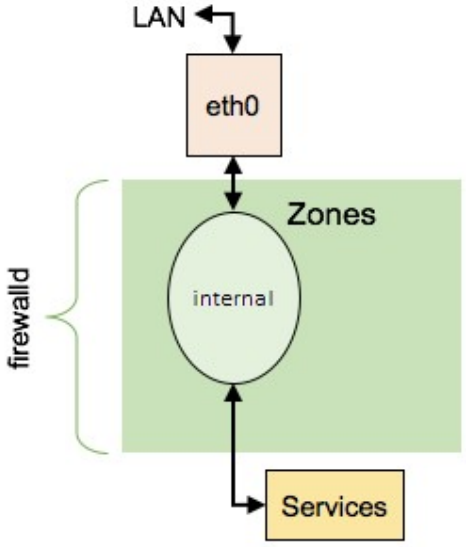
nmcli con reload

13.3.- Caso práctico

El servidor **central** tiene la siguiente configuración de las zonas de FirewallD, dada en su archivo de kickstat:



En **server1** se ha de configurar la zona *internal* como zona por defecto para obtener el esquema siguiente:



Se hará que el tráfico que proviene de **central**, vaya a la zona *internal*. Además se abrirán los puertos para *dns* y el puerto 3260/tcp para la zona *internal*.

RESOLUCIÓN

- Vemos el estado de las zonas en **server1**:

```
[root@server1 ~]# firewall-cmd --get-default-zone
public
```

vemos que la zona por defecto es la zona pública, lo modificamos:

```
[root@server1 ~]# firewall-cmd –set-default-zone=<internal>
success
```

- Hacemos que el tráfico que proviene de **central**, vaya a la zona *internal*:

```
[root@server1 ~]# firewall-cmd --persistent --add-source=10.11.1.254 --zone=internal
success
[root@server1 ~]# firewall-cmd --reload
```

- Añadimos el servicio *dns* y el puerto *3260/tcp*:

```
[root@server1 ~]# firewall-cmd --permanent --add-service=dns --zone=internal
success
[root@server1 ~]# firewall-cmd --permanent –add-port=3260/tcp --zone=internal
[root@server1 ~]# firewall-cmd --reload
```

- Verificamos:

```
[root@server1 ~]# firewall-cmd --list-all
internal (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0
  sources: 10.11.1.254
  services: dhcpv6-client mdns ssh
  ports: 3260/tcp
  protocols:
  masquerade: no
  forward-ports:
  sourceports:
  icmp-blocks:
  rich rules:
```