[Cod.: CC481 Curso: Administración de Redes]
[Tema: Firewall – Parte II]
[Prof.: Manuel Castillo]

Laboratorio dirigido 8.2

Firewall en Linux - iptables

Instrucciones:

1. Fecha de entrega será antes del **domingo a las 23:59**.
2. El formato de entrega será *pdf*.
3. El laboratorio tendrá una puntuación sobre 20.
4. Para los laboratorios de servicios se deberá realizar capturas de cada paso que se están llevando a cabo. Además de realizar un **videotutorial** al final que muestre la correcta implementación de nuestro servicio.
5. La primera hoja será para la portada que se especificará, el número de laboratorio, nombre y apellidos de los integrantes, nombre asignatura y el escudo de la UNI.
6. La segunda página será el índice. Donde se deberá tener en cuenta la página de cada Actividad.
7. Las citas y extracciones realizadas de Internet se deberán especificar. No se corregirá el laboratorio en caso de copiar y pegar fragmentos sin especificar.
8. Utilizar letra clara y adecuada a un documento técnico con tamaño 12 y márgenes superior e inferior de 3 cm y laterales de 2,5 cm.
9. Se corregirá la claridad y exactitud de la pregunta, en ningún caso se expondrán fundamentos no preguntados, además de la claridad del documento.
10. En las actividades realice una captura de pantalla mínimo por actividad para verificar su autoría.

0. Antecedentes

En este laboratorio vamos a trabajar la conexiones entre máquinas (por lo que tenemos que tener otra máquina virtual activa). Trabajaremos la conexión entre máquinas virtuales como red interna por lo que la configuración es como hemos venido haciendo hasta ahora.

Realizar un snapshot de nuestra máquina ya que vamos a trabajar dos laboratorios parecidos para que nos queden claros los conceptos.

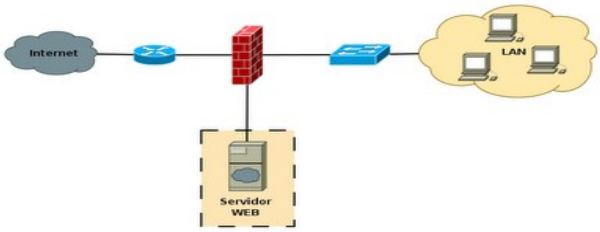
Trabajaremos con una nueva máquina CentOS o Ubuntu Server.

1. Introducción

Un *firewall* es, por lo general, un *software* o *hardware*, a través del cual nos conectamos a una red como Internet, y que sirve como filtro sobre el tráfico que por él pasa, en ambas direcciones, y que en un momento dado puede rechazar cierto tráfico en alguna de las direcciones.

Eso quiere decir que, mediante un *firewall*, podemos detectar el tráfico no deseado hacia nuestros sistemas, y en general, los posibles ataques de que seamos objeto. De esta manera podremos aislar nuestros equipos del exterior, permitiendo nuestro uso de Internet de manera absolutamente normal pero minimizando en lo posible la probabilidad de padecer las consecuencias de un ataque.

Es frecuente también que se necesite exponer algún servidor a Internet (como es el caso de un servidor web, un servidor de correo, etc...), y en esos casos obviamente en principio se debe aceptar cualquier conexión a ellos.



1.1. Iptables

Iptables es el software de filtrado de paquetes y de IP *masquerading* para *kernels 2.3* y superiores, mientras que *ipchains* lo era hasta las versiones 2.2, e *ipfwadm* para las más antiguas. Al arrancar y en /boot podemos ver el kernel que estamos usando. Por tanto podemos resumir que *iptables* es la herramienta por la cual podemos crear reglas para cada filtrado de paquetes y módulos NAT que se encuentra a disposición de los administradores.

El paquete de *iptables* debe estar instalado. Comprueba la versión instalada con:

```
# iptables –version
```

Si no tenemos instalado el paquete lo instalamos:

```
# yum -y install iptables
```

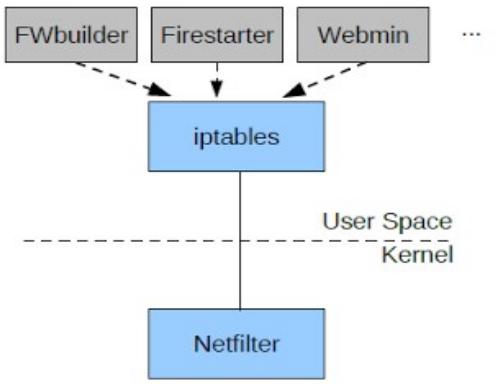
1.2. Netfilter

Por otro lado *netfilter*, nos permite el filtrado de paquetes y debe estar activada en el *kernel*, por lo que si no lo tenemos compilado hay que recompilarlo. Más detenidamente podemos detallar que *netfilter* es un conjunto de ganchos (*Hooks*) que son utilizados para interceptar y manipular paquetes de red. Su componente principal es *firewall* con *iptables* que es el que realiza la filtración de los paquetes.

Actividad 1

1. Busca exactamente que es un hooks, haga un análisis sobre ellos y su utilización en NAT.

Dirección: www.netfilter.org



Mas adelante veremos su componente práctica, para entender estos conceptos teóricos.

1.3. Ipforwarding

Este concepto viene dado que un equipo Linux puede actuar como *router*. Es decir, un equipo con 2 tarjetas de red puede enrutar entre varias subredes si tiene activado *Ipforwarding* (todas las distribuciones de Linux lo traen compilado como un módulo). Para asegurarnos:

```
# cat /proc/sys/net/ipv4/ip_forward
```

Si el valor leído es 1, entonces está activado. Si es 0, no. Para activarlo, si no lo está ya:

```
# echo "1" > /proc/sys/net/ipv4/ip_forward
```

2. Procedimientos

2.1. Cadenas

Si hablamos del concepto de filtrado (*filter*) toda cadena puede ser INPUT (tráfico entrante), OUTPUT (tráfico saliente) o FORDWARD (tráfico reenviado). Aunque existe NAT y Mangle con diferentes cadenas que veremos como concepto más adelante.

Actividad 2

1. Busque más información sobre cadenas y tenga claros sus conceptos.

2.2. Tablas

Iptables usa por defecto tres tablas. Cada tabla tiene un conjunto de cadenas que gestionan los paquetes. Una cadena es una serie de acciones a realizar sobre los paquetes. Si estamos usando la tabla "filter", cada interfaz en la red tiene tres cadenas diferentes: *input*, *forward* y *output*. Veamos las cadenas que corresponden a las tablas.

Nombre tabla	Cadenas por defecto	Descripción
Filter	INPUT FORWARD OUTPUT	Permite el filtrado de paquetes
Nat	PREROUTING OUTPUT POSTROUTING	Permite el IP masquerading
Mangle	PREROUTING OUTPUT	Permite una manipulación (“mangle”) de los paquetes cambiando su contenido.

Veamos la definición:

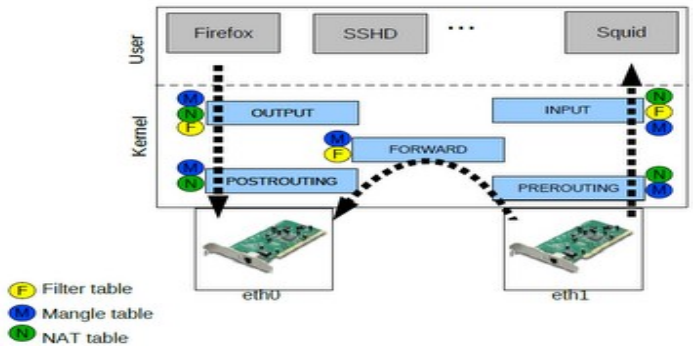
- **FILTER**: Tabla por defecto, para los paquetes que se refieran a nuestra máquina
  - INPUT: Paquetes recibidos para nuestro sistema.
  - FORWARD: Paquetes enrutados a través de nuestro sistema
  - OUTPUT: Paquetes generados en nuestro sistema y que son enviados.
- **NAT**: Tabla referida a los paquetes enrutados en un sistema con masquerading
  - PREROUTING: Para alterar los paquetes según entren.
  - OUTPUT: Para alterar paquetes generados localmente antes de ser enrutados.
  - POSTROUTING: Para alterar los paquetes cuando están a punto para salir.
- **MANGLE**: Alteraciones más especiales de paquetes
  - PREROUTING: Para alterar los paquetes entrantes antes de ser erutados.
  - OUTPUT: Para alterar los paquetes generados localmente antes de enrutar.

2.3. Reglas de destino

Para las reglas de destino podemos diferenciar en:

- ACCEPT: conexiones aceptadas
- DROP: conexiones descartadas
- REJECT: rechazo de conexiones
- POSTROUTING: Encaminamiento posterior. Para alterar los paquetes cuando están a punto para salir.

- PREROUTING: Encaminamiento previo. Para alterar los paquetes entrantes antes de ser enrutados.
- …



Actividad 3

1. Estas no son las únicas reglas que existen, busque más información sobre las puestas aquí y otras reglas más

2.4. Especificación de reglas

Se hace con los siguientes parámetros (especificando aquellos que se necesite):

- **-p [protocolo]**: Protocolo al que pertenece el paquete.
- **-s [origen]**: dirección de origen del paquete, puede ser un nombre de host, una dirección IP normal, o una dirección de red (con máscara, de forma dirección/máscara).
- **-d [destino]**: Al igual que el anterior, puede ser un nombre de host, dirección de red o dirección IP singular.
- **-i [interfaz-entrada]**: Especificación del interfaz por el que se recibe el paquete.
- **-o [interfaz-salida]**: Interfaz por el que se va a enviar el paquete.
- **[!] -f**: Especifica que la regla se refiere al segundo y siguientes fragmentos de un paquete fragmentado. Si se antepone !, se refiere sólo al primer paquete, o a los paquetes no fragmentados.

Y además, uno que nos permitirá elegir qué haremos con el paquete:

- **-j [target]**: Nos permite elegir el target al que se debe enviar ese paquete, esto es, la acción a llevar a cabo con él.

Actividad 4

1. Busque información si no recuerda que es un tarjet y una interfaz.

3. Instalación de un firewall personal

Iptables nos permitirá controlar el tráfico entre dos redes (equipos ***multihomed***: con dos tarjetas de red, generalmente una conectada a la red interna y otra la externa). Sin embargo, podemos usarlo también en los equipos conectados a una red para bloquear o monitorizar puertos, o deshabilitar el *ping*. Estaremos creando un firewall personal.

Nota: podemos deshabilitar el ping sin usar iptables así:

```
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all.
```

Actividad 5

1. Prueba a modificar este fichero en su máquina y realice un ping. Busque información sobre este archivo y su funcionalidad. Finalmente lo volvemos a poner a 0.

3.1. Consulta de configuración

En esta primera parte vamos a ver las opciones básicas que tenemos de para ver el estado de nuestro propio firewall, así como las reglas que poseemos.

Actividad 6

1. Para consultar todas las cadenas y reglas. Estudie dicho comando así como todas salidas que nos muestra, como cada una de las filas y columnas para tener claros los conceptos.

```
# iptables -nL --line-numbers
```

2. Para consultar una cadena específica. Según el comando siguientes muestre las restantes cadenas acorde a lo visto en clase.

```
# iptables -L fordward
```

3. Para consultar una tabla específica. Especifique el comando para consultar las otras tablas.

```
# iptables -t nat -L
```

4. Indique el comando para consultar una tabla y cadena específicas. Muestre todas ellas.

```
# iptables -t nat -L output
```

3. 2. Cambios de configuración

Pasamos a configurar nuestro *firewall*. En esta sección estableceremos algunas reglas y aprenderemos a manipularlas.

Actividad 7

1. Borramos todas las reglas actuales (salvo las reglas por defecto) con la opción *-F*. ¿A qué se le puede llamar una regla por defecto? Reiniciamos la máquina virtual y volvemos a lanzar el comando. ¿Que conclusiones sacas con la información dada?

2. Instalación de regla por defecto para una cadena. ¿Que realiza la sentencia ejecutada anteriormente? Explique el resultado obtenido y busque algún ejemplo más.

3. Guardamos la configuración actual en un fichero. Realice primero un *cat* al archivo original, ¿que nos está mostrando? Ahora hágalo en el copiado en el que podrás observar el contenido. ¿Por qué puedo ver el copiado y el original no? Explique el por qué. Encima no coincide con lo listado de *iptables* lo mostrado en el backup, lo veremos a continuación.

```
# /sbin/iptables-save > iptables.txt
```

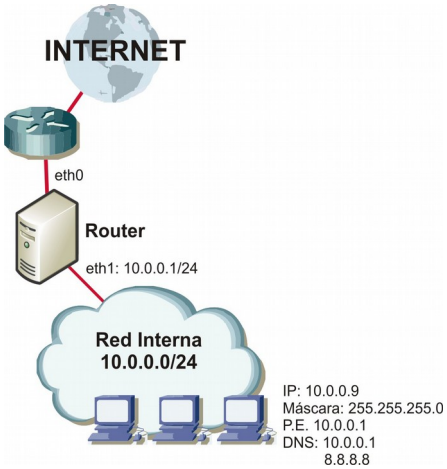
4. Finalmente se puede restaurar nuestro backup de *iptables* desde un fichero. Reinicie el sistema y haga una copia de seguridad del archivo de *iptables*, borra todas las reglas y vuelva a restaurar toda la información, realizando consultas de nuestro firewall.

```
# /sbin/iptables-restore < iptables.txt
```

NOTA: Antes de proseguir realice “Practica 1”

3.3. Targets en las reglas del firewall

Volvemos a la configuración en máquinas virtuales como muestra la siguiente figura. En esta sección vamos a aprender a activar el *log* de la misma.



Actividad 8

1. En *iptables* se necesitan dos reglas, la primera de las cuales tiene *DROP* como acción. Por ejemplo, para descartar todo el tráfico TCP y hacer el log. En este contexto genere las siguientes reglas:
   1. Registrar todos los paquetes entrantes (LOG) que tenga cualquier origen, con protocolo *TCP* y entrando por la interfaz de red correspondiente.
   2. Desechar (DROP) posteriormente los paquetes que cumpla con la regla mencionada anteriormente.

Con ***-j*** se especifica el **target o destino del paquete** que cumpla la regla. *ACCEPT*, *DROP* y *REJECT* (como *DROP* pero enviando un mensaje de error) son targets terminantes, mientras que *LOG* no lo es (después de hacer el *log*, se sigue con la siguiente regla). Los mensajes se van grabando en ficheros de ***/var/log/messages*** junto con otros mensajes del kernel. Puedes consultarlos con las herramientas ***dmesg.***

Actividad 9

1. Busque como podríamos obtener estos mensajes específicos para lo que estamos trabajando. Ya sabemos que los dos descritos anteriormente muestra mensajes de todo nuestro sistema.

4. Agregar o eliminar relgas del firewall

Usaremos ***-A*** para agregar reglas al final de una cadena y ***-I*** para agregar al principio de la cadena. El resto de comandos puedes consultarlo

usando "*man IPTables*". Lo primero que vamos a realizar es limpiar la configuración del *firewall* y establece ACCEPT como regla por defecto:

```
# iptables –F
# iptables –P INPUT ACCEPT
# iptables –L
```

4. 1. Comprobamos que los servicios funcionan.

Antes de realizar la configuración vamos a intentar trabajar esta parte con nuestro compañeros, para ello utilizamos la configuración en red adecuada en nuestra máquina y vemos si no nos cierra el *proxy* este tipo de conexiones.

Actividad 10

1. Lo primero es realizar un *ping* a las máquinas para ver su correcto funcionamiento en cuanto a conexión.

2. Instalamos apache en la que va a actuar como servidor e iniciamos el servicio.

   ```
   # /etc/rc.d/init.d/apache2 start
   ```

3. Verificamos que se encuentra activo el servidor. Recordamos que o bien abriendo nuestro navegador y poniendo *http://localhost*. Si hemos tenido éxito en la conexión, ponemos la *IP* de cada una de las máquinas en la barra de direcciones y vemos si nos acepta la conexión. La otra forma mediante comandos era:

   ```
   # netstat -vat
   ```

   En mis dos máquinas virtuales obtengo:

   ```
   tcp        0      0 *:44602                 *:*                         LISTEN
   tcp        0      0 192.168.1.10:34496      192.168.1.15:http           TIME_WAIT
   tcp        0      0 192.168.1.10:36482      gru03s14-in-f7.1e100.n:http TIME_WAIT
   tcp        0      0 192.168.1.10:37339      gru03s07-in-f1.1e100.n:http ESTABLISHED
   ```

4. Arrancamos el servidor *SSH* de tu máquina (procede a instalarlo en caso necesario).

   ```
   # /etc/rc.d/init.d/ssh start
   ```

5. Creamos un usuario para realizar las pruebas de *SSH* y le asignamos una contraseña. Realizamos lo mismo pero a la máquina virtual.

   ```
   # useradd usuario1
   ```

```
# passwd usuario1
```

6. Ahora ya estamos en condiciones de conectarnos. Realizamos lo mismo pero a la maquina de un compañero.

```
# ssh –l user@IPDETUMAQUINA
# whoami
# exit
```

5. Configuraciones de comunicación

5.1. Establecer política por defecto

Establece *DROP* como política por defecto para la regla *INPUT* de la tabla *Filter*. Esto evita cualquier conexión remota, pero también impide que lleguen las respuestas de las conexiones que establezca este host. Toda salida está autorizada.

```
# iptables –P INPUT DROP
```

Actividad 11

1. Vea el listado de *iptables* que nos da.
2. Realicemos una prueba con las demás máquinas y veamos lo que ocurre. Pruebe intentando acceder a la web y con ssh. Como observarás no hay comunicación (obviamente por DROP), por lo que procedemos a dar comunicación a los diferentes servicios.

5.2. Reglas para permitir que nuestro navegador web pueda conectarse al exterior.

Actividad 12

1. Autorizo para que mi navegador web pueda salir al exterior (recibir respuestas, pues la salida está autorizada). Primeramente busca información sobre todo lo expuesto aquí y anótela. Seguidamente intente conectarse a Internet y comprobarás que todavía no tiene acceso, esta cerrado.

2. Lo anterior no es suficiente: debo autorizar la resolución de nombres DNS para que el navegador pueda encontrar el sitio web. ¿Por qué hay que autorizar al DNS de mi conexión.

3. Además autorizo que desde mi cliente pueda acceder a por ssh al servidor.

5.3. Reglas para permitir que nuestro servidor web sea accesible desde el exterior

Actividad 13

1. Autorizo para que se pueda acceder desde el exterior a mi servidor web, pero activo el *log* para esta regla. No se olvide de extraer toda la información de lo puesto anteriormente. ¿Que significa cada línea?
2. Comprueba que desde otro equipo (que se encuentre en otra subred) pueden conectarse a tu servidor web y por ssh.
3. Comprueba con *dmesg* o en */var/log/messages* que en el log del *firewall* ha quedado registrada la actividad TCP al puerto 80, pero no la actividad ICMP. Seguro que en todo el reporte no entiendes nada. Busca información sobre este tipo de log y extraiga información.

```
# dmesg | grep firewall
```

6. Algunas preguntas frecuentes

1. **¿Porqué en el log no veo la URL invocada?**

   Porque la URL forma parte del paquete HTTP, que es de nivel de aplicación (7 en el modelo OSI), mientras que el firewall trabaja en los niveles 3 y 4, por lo que nunca llega a analizar el contenido del paquete de nivel 7. Para eso existe el Proxy.

2. **¿Se mantienen las reglas cuando reinicio?**

   No, deben de cargarse en el proceso de arranque. En Red Hat, el fichero */etc/sysconfig/iptables* contiene las reglas cargadas al arrancar.

PRACTICA 1

1. Consulte la configuración inicial del firewall
2. Realice a tu propia máquina y a otra maquina virtual un *ping*.
3. Guarda la configuración en el fichero */tmp/iptables.txt*
4. Establece *DROP* como regla por defecto para la cadena *INPUT*. Así estamos deshabilitando todo el tráfico entrante. Si tiene alguna duda en los primeros apartados explica que realiza *DROP* y qué es la cadena *INPUT*.
5. Cambia a *NAT* en tu máquina virtual, comprueba que tienes acceso al exterior y finalmente si otros pueden establecer conexiones contigo. ¿Que es lo que ha sucedido? Explíquelo de forma clara tus conclusiones.

   Recuerda: que para este estado necesitamos reiniciar la máquina teniendo nuestra configuración de red en *adaptador puente*.
6. ¿Como puedo volver al estado anterior para que pueda volver a realizar ping? No utilice *iptables -F*. La solución no es destruir todas nuestras tablas, busque la solución aconsejable para este caso.

PRACTICA 2

1. ¿Cómo puedo saber si el paquete fue aceptado o rechazado?
2. ¿Cómo sacar estadísticas de esta información?
3. ¿En qué fichero físico se graban los datos?
4. ¿Cómo puedo personalizar esta información?
5. Existen muchas herramientas de interfaz gráfica para configuración de firewall. Estudie alguna de ellas tendiendo los conocimientos adquiridos
6. Al haber negado toda conexión también se ha negado como hemos comprobado para ICMP para ping y SSH. Abra estas conexiones.
7. En su trabajo de máquinas virtuales hay un servidor FTP por ejemplo, también debería darle conexión.