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

Laboratorio dirigido 8.3

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 esta segunda parte vamos a seguir realizando configuraciones de nuestro firewall, por lo que es muy importante tener claros los conceptos vistos anteriormente.

En esta parte vamos a trabajar con una conexión virtual tipo puente para ver tratamiento de tráfico de entrada y salida.

1. Recordamos puntos importantes.

1.1. Políticas por defecto

Recordamos que las políticas por defecto establecen las reglas (o acciones) generales ante cualquier tipo de conexión.

```
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
# iptables -P OUTPUT ACCEPT
```

Ya lo deberíamos saber pero recordamos:

1. *-P*: Cambia una política para una cadena.
2. *DROP - INPUT*: Esta sentencia descarta todas las conexiones entrantes.
3. *FORWARD – DROP*: Descarta todas las conexiones que se reenvíen.

Actividad 1

1. ¿Que quiere decir conexiones que se reenvíen? ¿Y la ultima sentencia puesta? Prueba a realizar pruebas con tus máquinas virtuales a la conexión de IP (debe de estar activado Apache) y denegando-aceptando tráfico para cada cadena. Explique claramente cuando cierra y permite en cada sentencia.

1.2. Inicializando nuestra propia configuración

Llegados a este punto ya deberíamos conocer como borrar las reglas que existentes en nuestra configuración, aunque vamos a recordarlas. Estas reglas principales serán para el tráfico reenviado, entrante/saliente y NAT.

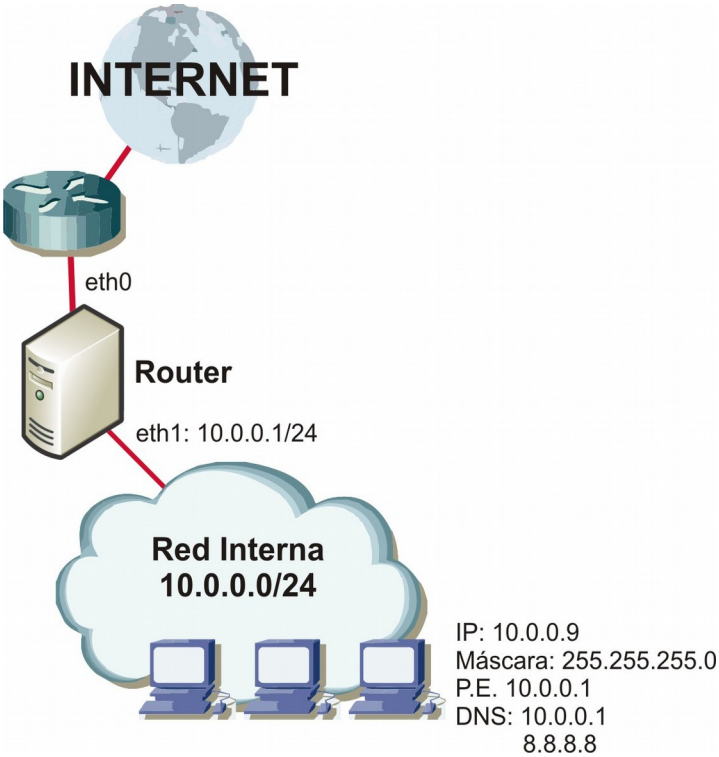
```
# iptables -F INPUT
# iptables -F FORWARD
# iptables -F OUTPUT
# iptables -F -t nat
```

Actividad 2

1. Recuerde que hace cada una de las reglas y visualice la configuración que ha quedado al lanzarlas.

2. Mas Reglas

Ya sabemos como son las reglas principales y su aplicación práctica. Vamos a ver más ejemplo para que nos quede realmente claro la importancia de tener una buena configuración de seguridad en nuestro propio firewall.

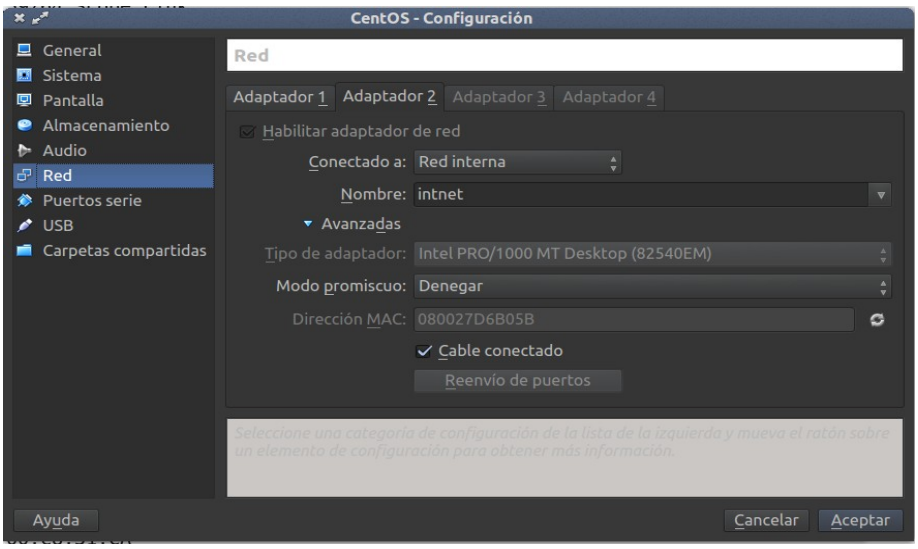


2. 1. Configuración con dos interfaces de red

Lo primero a realizar es crear una interfaz de red nueva *eth1* en nuestra maquina virtual simulando que tenemos una computadora de dos interfaces de red. Este punto no se puede ampliar ya que no poseemos una dirección IP pública, aunque en el anexo se han expuesto algunas configuraciones que el alumno puede utilizar para que vea como sería la configuración con dos interfaces de red, donde una es interna y la otra de salida. Para ello sigamos los paso de configuración.

Actividad 3

1. Añadimos una interfaz de red en VM. Recordad que la VM debe de estar apagada para poder añadir dicha interfaz.



2. Una vez creada inicializamos nuestra VM y vemos que esta correctamente creada. En ella podremos observar las dos interfaces que tenemos *eth0* (con configuración de red *NAT*) y *eth1* (con configuración de red interna).

```
# ifconfig
```

```
[manuel@localhost ~]$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:74:71:59
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe74:7159/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:19 errors:0 dropped:0 overruns:0 frame:0
          TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10122 (9.8 KiB)  TX bytes:1923 (1.8 KiB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:D6:B0:5B
          inet6 addr: fe80::a00:27ff:fed6:b05b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:2862 (2.7 KiB)
```

3. Procedemos a configurar las dos interfaces de red:
   1. *eth0* será la tarjeta que nos permita conectarnos a internet. Modificamos el fichero.

      # gedit /etc/sysconfig/network-script/ifcfg-eth0

      ifcfg-eth0

      ```
      DEVICE=eth0
      ONBOOT=yes
      BOOTPROTO=dhcp
      TYPE=Ethernet
      ```

   2. eth1 será la tarjeta que nos permita conectarnos a la red interna. Modificamos el fichero. Describa lo que significa cada opción.

      # gedit /etc/sysconfig/network-script/ifcfg-eth1

      ifcfg-eth1

      ```
      DEVICE=eth1
      BROOTPROTO=no
      HWADDR=08:00:27:D6:B0:5B
      NM_CONTROLLED=yes
      ONBOOT=yes
      IPV6INIT=no
      IPADDR=10.0.0.1
      NETMASK=255.255.255.0
      TYPE=Ethernet
      ```

      # netstat -vatn

      Podremos observar las conexiones activas que tenemos. Bien vamos ahora al navegador y escribimos la siguiente dirección

      # http://www.linux.org

Para saber la dirección IP pública vamos a la pagina que nos da dicha información.

```
# http://www.dnswatch.info
```

Domain	Type	TTL	Answer
linux.org.	NS	86400	ns1.iqnection.com.
linux.org.	NS	86400	ns2.iqnection.com.
linux.org.	A	14400	209.92.24.80

Finalmente vemos las conexiones activas otra vez:

```
# netstat -vatn
```

Y podremos observar que la ip de eth0 es la que ha realizado la conexión. En *netstat* podemos filtrar el tráfico por interfaz de red al exterior. Realice un *man* y descubra que opción es la que podemos utilizar.

```
tcp        0      1 10.0.2.15:33574         74.125.234.64:80          FIN_WAIT1
tcp        0      1 10.0.2.15:60939         209.92.24.80:80           FIN_WAIT1
tcp        1      0 10.0.2.15:53642         200.60.136.32:80          CLOSE_WAIT
```

Otra forma de poder ver qué interfaz de red es la que tiene acceso al esterior es:

```
# ping -I eth0 www.google.es
```

```
[root@localhost manuel]# ping -I eth0 www.google.es
PING www.google.es (190.98.171.55) from 10.0.2.15 eth0: 56(84) bytes of data.
64 bytes from 190.98.171.55: icmp_seq=1 ttl=63 time=15.4 ms
64 bytes from 190.98.171.55: icmp_seq=2 ttl=63 time=9.56 ms
```

Llegados aquí vamos a ver que nuestros paquetes ICMP están todos ok. Veamos que pasa con *eth1*. Intentar realizar un ping a *www.duckduckgo.com*. ¿Que sucede? ¿Por que se pierden todos paquetes? Este punto es una anécdota muy interesante descubra el por qué.

```
# ping -I eth1 www.google.es
```

```
[root@localhost manuel]# ping -I eth1 www.google.es
PING www.google.es (190.98.171.24) from 10.0.2.15 eth1: 56(84) bytes of data.
From 10.0.2.15 icmp_seq=2 Destination Host Unreachable
From 10.0.2.15 icmp_seq=3 Destination Host Unreachable
From 10.0.2.15 icmp_seq=4 Destination Host Unreachable
```

4. Lo siguiente que vamos a realizar es indicar al servidor que actúe como router.

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

5. Ahora indicamos que la red interna tenga salida al exterior por NAT

```
# iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -d 0/0 -j MASQUERADE
```

6. Permitimos todo el tráfico proveniente de la red interna denegando todo lo demás.

```
# iptables -A FORWARD -s 10.0.0.0/24 -j ACCEPT
# iptables -A FORWARD -m state --state RELATED, ESTABLISHED -j ACCEPT
# iptables -A FORWARD -j DROP
```

   Vea como permitir el tráfico para la web, es decir para el puerto 80.

   Se dejará al estudiante que habilite todas las maquinas virtuales de manera que establezca un servidor central como router firewall por el que filtre conexiones. Anexo de este mismo manual

7. Finalmente le indicamos al fichero */etc/sysctl.conf* que active la variable *ip_forward*

```
net.ipv4.ip_forward=1
```

3. Algunas reglas

3.1. Creación de reglas

Actividad 4

1. Descartar el tráfico que tenga como puerto de destino el 22 y 23.

```
# iptables -A INPUT -p tcp --destination-port 22 -j DROP
# iptables -A INPUT -p tcp --destination-port 23 -j DROP
```

   Ya vimos como desactivar *openssh* intente ahora desactivar la opción de *telnet* tanto de entrada como de salida.

2. Descartar conexiones entrantes

```
# iptables -A INPUT -s x.w.y.z -j DROP
```

3. Rechazar conexiones hacia determinadas direcciones IP

```
# iptables -A OUTPUT -d a.b.c.d -s 192.168.0.0/24 -j REJECT
```

3.2. Eliminación de reglas

Ya sabemos que *-A* es para crear reglas, para eliminarlas simplemente con la opción *-D (DROP)*.

```
# iptables -D INPUT -s x.y.z.w -j DROP
```

4. El servicio *iptables*

Actividad 5

1. Para guardar las reglas *iptables* generadas. Para Centos *etc/sysconfig/iptables,* para Ubuntu guardar la configuración en el fichero */sbin/iptables-save*
2. Intalamos el servicio *iptables-persistent* y ejecutar el servicio por primera vez *netfilter-persistent*.
3. Borrar todas las reglas.
4. Creamos un fichero donde guardar las reglas y le damos permiso de ejecución.

6. Más Configuraciones

En el punto dos expusimos como crear una dos interfaces de redes. Para este apartado se explicarán los comandos y es cuestión del alumno verificar que su utilización y como afecta a la configuración. Por lo que se deberán buscar información de como probar las configuraciones expuestas y ejecutar las que no requieren IP público sino con nuestra máquina podamos realizar. Además, por supuesto, de entender claramente cada una de las opciones.

Actividad 6

Para esta actividad y con tu configuración de servidor-firewall y cliente comprueba el funcionamiento de cada una de las reglas utilizadas.

1. La primera opción que vamos a realizar es reenviar paquetes de nuestra interfaz interna (*eth1*) hacia la interfaz pública (*eth0*).
2. Acto seguido aceptamos a las conexiones existentes (*ESTABLISHED, RELATED*) reenviar los paquetes son parte de ellas o que están

relacionadas con el tráfico entrante desde nuestra interfaz *eth1* para el saliente *eth0*.

3. Vamos ahora a aceptar para el tráfico saliente a través de *eth0* que son parte de conexiones existentes o relacionadas a los paquetes. Guíese de la regla anterior (3.).
4. También podremos permitir todo el tráfico entrante desde cualquier dirección (0/0) a *eth1* y desde el *loopback* hacia cualquier otro destino (0/0).
5. Hacer un *SNAT* a través de *eth0* proveniente desde 192.168.0.0/24 (red local) utilizando (--to-source) para una dirección IP pública
6. Descartamos a continuación el tráfico entrante desde *eth0* que trate de utilizar la IP pública una dirección IP local o la dirección *loopback*
7. Aceptar todos los paquetes *SYN* para los puertos especificados del protocolo TCP (22, 25, 80 y 443).
8. También podríamos aceptar el tráfico entrante de la interfaz *eth1* desde el puerto de destino 67 por protocolos *TCP* y *UDP* sean establecidas con puerto origen 68.
9. Por último aceptamos tráfico entrante por el protocolo *UDP* cuando se establezcan desde el servidor *DNS* desde el puerto 53 a cualquier destino.