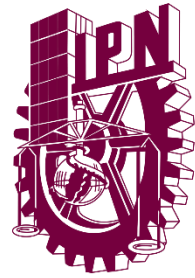




**Instituto Politécnico  
Nacional**



Unidad Profesional Interdisciplinaria en Ingeniería y  
Tecnologías Avanzadas

Seguridad de Redes

Profesor. Polanco Montelongo Francisco Antonio

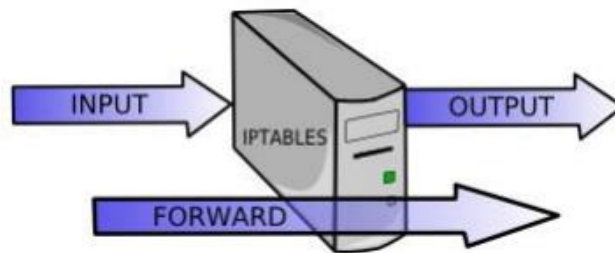
Alumno: Hernández Guzmán Bryan Alexis

Grupo: 3TV4

Práctica: IPtables

## Introducción:

Iptables es un módulo del núcleo de Linux que se encarga de filtrar los paquetes de red, es decir, es la parte que se encarga de determinar qué paquetes de datos queremos que lleguen hasta el servidor y cuáles no. Al igual que ocurre con otros sistemas de cortafuegos, iptables funciona a través de reglas. Es decir, el usuario mediante sencillas instrucciones indica al firewall el tipo de paquetes que debe permitir entrar, los puertos por donde se pueden recibir esos paquetes, el protocolo utilizado para el envío de datos y cualquier otra información relacionada con el intercambio de datos entre redes. Cuando en el sistema se recibe o se envía un paquete, se recorren en orden las distintas reglas hasta dar con una que cumpla las condiciones. Una vez localizada, esa regla se activa realizando sobre el paquete la acción indicada. Gracias a su robustez, iptables se ha convertido hoy por hoy en una de las herramientas más utilizadas para el filtrado de tráfico en sistemas Linux.



## Desarrollo:

Después de realizar algunos cambios dentro de la configuración de la máquina física, como lo es crear una nueva regla que permite el tráfico de datos, además de permitir la conexión de impresora dominio al igual que la privada.

Verificamos nuestra dirección ip de nuestras 3 máquinas, la física y 2 virtuales.

Tenemos que meter nuestras reglas dentro de la máquina virtual 1:

```
Archivo Acciones Editar Vista Ayuda
(bryan@kali)-[~]
$ sudo iptables -F
(bryan@kali)-[~]
$ sudo iptables -A INPUT -i lo -j ACCEPT
(bryan@kali)-[~]
$ sudo iptables -A FORWARD -j ACCEPT
Chain PREROUTING (policy ACCEPT)
target prot opt source destination
DNAT
(bryan@kali)-[~]
$ sudo bash -c 'echo "1" > /proc/sys/net/ipv4/ip_forward'
Chain INPUT (policy ACCEPT)
target prot opt source destination
(bryan@kali)-[~]
$ sudo iptables -t nat -A PREROUTING -p icmp --icmp-type echo-request -d 192.168.1.65 -j DNAT --to-destination 192.168.1.67
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
(bryan@kali)-[~]
$ sudo iptables -t nat -A POSTROUTING -p icmp --icmp-type echo-request -s 192.168.1.68 -j SNAT --to-source 192.168.1.67
Chain POSTROUTING (policy ACCEPT)
target prot opt source destination
SNAT icmp -- 192.168.1.68 anywhere icmp echo-request to:192.168.1.65
(bryan@kali)-[~]
$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
(bryan@kali)-[~]
$ sudo iptables -t nat -L
```

```
Archivo Acciones Editar Vista Ayuda
$ sudo bash -c 'echo "1" > /proc/sys/net/ipv4/ip_forward'
(bryan@kali)-[~]
$ sudo iptables -t nat -A PREROUTING -p icmp --icmp-type echo-request -d 192.168.1.65 -j DNAT --to-destination 192.168.1.67
(bryan@kali)-[~]
$ sudo iptables -t nat -A POSTROUTING -p icmp --icmp-type echo-request -s 192.168.1.68 -j SNAT --to-source 192.168.1.67
(bryan@kali)-[~]
$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
(bryan@kali)-[~]
$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target prot opt source destination
DNAT icmp -- anywhere 192.168.1.65 icmp echo-request to:192.168.1.67
Chain INPUT (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
Chain POSTROUTING (policy ACCEPT)
target prot opt source destination
SNAT icmp -- 192.168.1.68 anywhere icmp echo-request to:192.168.1.65
```

Procedemos a hacer lo mismo dentro de nuestra segunda máquina virtual:

```
(bryan1@kali2)-[~]  
$ sudo iptables -F  
  
(bryan1@kali2)-[~]  
$ sudo sysctl -w net.ipv4.ip_forward=1  
net.ipv4.ip_forward = 1  
  
(bryan1@kali2)-[~]  
$ sudo iptables -A INPUT -i lo -j ACCEPT  
  
(bryan1@kali2)-[~]  
$ sudo iptables -A FORWARD -j ACCEPT  
  
(bryan1@kali2)-[~]  
$ sudo bash -c 'echo "1" > /proc/sys/net/ipv4/ip_forward'  
  
(bryan1@kali2)-[~]  
$ sudo iptables -t nat -A PREROUTING -p icmp --icmp-type echo-request -d 192.168.1.67 -j DNAT --to-destination 192.168.1.68  
  
(bryan1@kali2)-[~]  
$ sudo iptables -t nat -A POSTROUTING -p icmp --icmp-type echo-request -s 192.168.1.65 -j SNAT --to-source 192.168.1.67
```

```
(bryan1@kali2)-[~]  
$ sudo iptables -t nat -L  
Chain PREROUTING (policy ACCEPT)  
target      prot opt source                destination          icmp echo-request to:192.168.1.68  
DNAT        icmp -- anywhere            192.168.1.67  
  
Chain INPUT (policy ACCEPT)  
target      prot opt source                destination  
  
Chain OUTPUT (policy ACCEPT)  
target      prot opt source                destination  
  
Chain POSTROUTING (policy ACCEPT)  
target      prot opt source                destination          icmp echo-request to:192.168.1.67  
SNAT        icmp -- 192.168.1.65      anywhere
```

Procederemos a hacer un ping de nuestra maquina física a virtual y visualizaremos en wireshark, aplicaremos un filtro para icmp:}

```
C:\Users\rockr>ping 192.168.1.65 -n 10
```

```
Haciendo ping a 192.168.1.65 con 32 bytes de datos:
```

```
Respuesta desde 192.168.1.65: bytes=32 tiempo=1ms TTL=126
Respuesta desde 192.168.1.65: bytes=32 tiempo=2ms TTL=126
Respuesta desde 192.168.1.65: bytes=32 tiempo=2ms TTL=126
Respuesta desde 192.168.1.65: bytes=32 tiempo=2ms TTL=126
Respuesta desde 192.168.1.65: bytes=32 tiempo=1ms TTL=126
Respuesta desde 192.168.1.65: bytes=32 tiempo=2ms TTL=126
Respuesta desde 192.168.1.65: bytes=32 tiempo=1ms TTL=126
Respuesta desde 192.168.1.65: bytes=32 tiempo=2ms TTL=126
Respuesta desde 192.168.1.65: bytes=32 tiempo=1ms TTL=126
Respuesta desde 192.168.1.65: bytes=32 tiempo=1ms TTL=126
```

```
Estadísticas de ping para 192.168.1.65:
```

```
Paquetes: enviados = 10, recibidos = 10, perdidos = 0
(0% perdidos),
```

```
Tiempos aproximados de ida y vuelta en milisegundos:
```

```
Mínimo = 1ms, Máximo = 2ms, Media = 1ms
```

The image shows a Wireshark packet capture on the eth0 interface. The filter is set to 'icmp'. The packet list shows 10 packets, alternating between Echo (ping) requests and replies. The first request is from 192.168.1.67 to 192.168.1.65, and the first reply is from 192.168.1.65 to 192.168.1.67. The packet details pane shows the structure of a ping request: Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol.

No.	Time	Source	Destination	Protocol	Length	Info
74	23.387598232	192.168.1.67	192.168.1.65	ICMP	74	Echo (ping) reply id=0x0001, seq=12
77	24.395585224	192.168.1.65	192.168.1.67	ICMP	74	Echo (ping) request id=0x0001, seq=12
78	24.395634873	192.168.1.67	192.168.1.68	ICMP	74	Echo (ping) request id=0x0001, seq=12
79	24.396183597	192.168.1.68	192.168.1.67	ICMP	74	Echo (ping) reply id=0x0001, seq=12
80	24.396233691	192.168.1.67	192.168.1.65	ICMP	74	Echo (ping) reply id=0x0001, seq=12
82	25.408108691	192.168.1.65	192.168.1.67	ICMP	74	Echo (ping) request id=0x0001, seq=12
83	25.408169929	192.168.1.67	192.168.1.68	ICMP	74	Echo (ping) request id=0x0001, seq=12
84	25.408726328	192.168.1.68	192.168.1.67	ICMP	74	Echo (ping) reply id=0x0001, seq=12
85	25.408766568	192.168.1.67	192.168.1.65	ICMP	74	Echo (ping) reply id=0x0001, seq=12

Frame 22: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0  
Ethernet II, Src: PcsCompu\_60:a4:d8 (08:00:27:60:a4:d8), Dst: 08:00:27:60:a4:d8  
Internet Protocol Version 4, Src: 192.168.1.65, Dst: 192.168.1.67  
Internet Control Message Protocol

#### Conclusión:

Dentro de esta práctica pudimos observar cómo hacer una conexión entre máquinas físicas y virtuales, la importancia que conlleva eso, en un plano más real que nos motiva a crear cosas nuevas, además de la utilización de diversas herramientas para llevar a cabo el cometido final que es entender lo que estamos haciendo, en general en la práctica tuve algunos problemas con las direcciones ip y a la hora de puentear ambas máquinas virtuales, por otro lado investigar los

comandos y para qué sirven fue un plus para dejar atrás errores que fueron cometidos dentro de la práctica.

Referencias:

<https://www.acens.com/wp-content/images/2014/07/wp-acens-iptables.pdf>