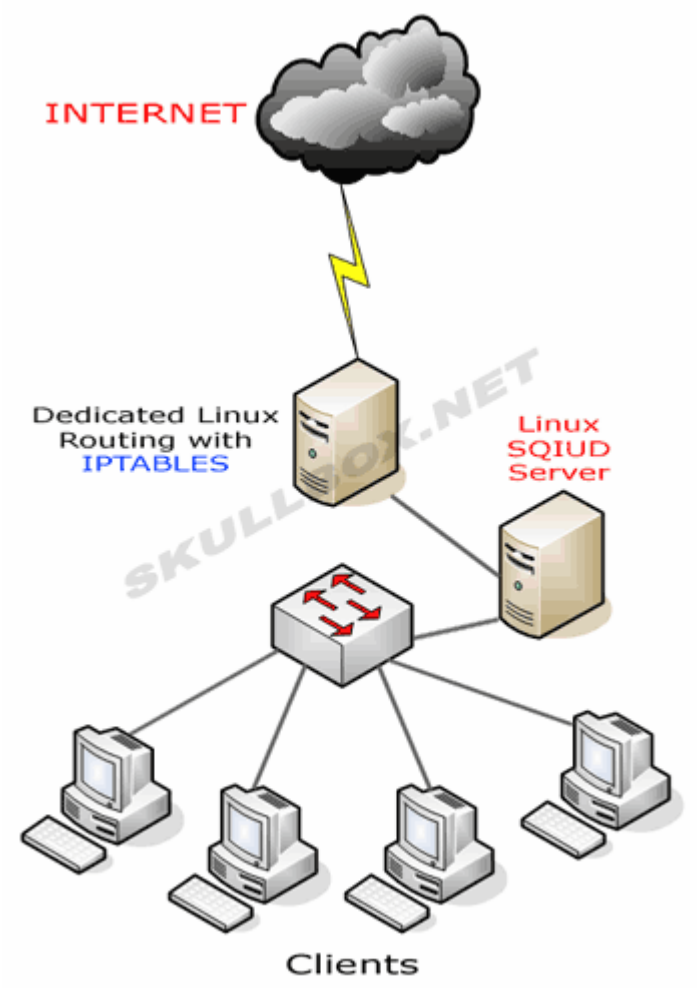
UNIDAD DIDACTICA 12

CONFIGURACIÓN DE IPTABLES EN GNU/LINUX

Eduard Lara

IPTABLES



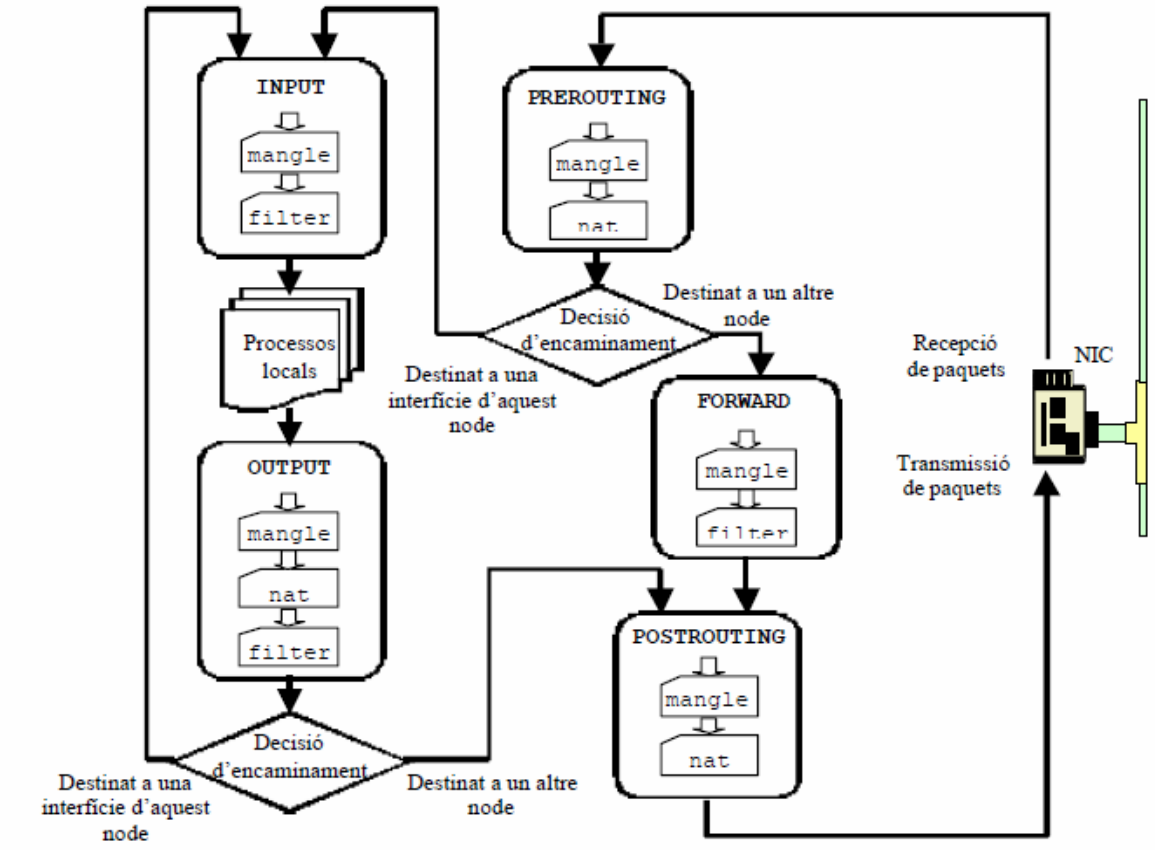
- La comanda IPTABLES se utiliza en linux para la configuración de un *firewall*.
- IPTABLES permite realizar la programación de servicios NAT y Listas de Acceso

1. INTRODUCCIÓN

- El comando iptables se utiliza en linux para la configuración de un firewall.
- La comanda iptables permite filtrar i/o modificar algunos campos de los paquetes a medida que atraviesan diferentes etapas (o chains) del nivel IP de la máquina linux. Estas etapas son PREROUTING, FORWARD, POSTROUTING, INPUT y OUTPUT.
- Cuando el paquete atraviesa una de estas etapas, el nivel IP consulta unas tablas donde la comanda iptables permite añadir las reglas de procesado. Estas tablas son mangle, filter y nat.

1. ETAPAS QUE SIGUE UN PAQUETE DENTRO EN LAS RUTINAS TCP/IP

❖ Esquema de las etapas que sigue un paquete desde que se recibe de la red o se genera en un proceso local, hasta que se transmite o se recibe por un proceso local.

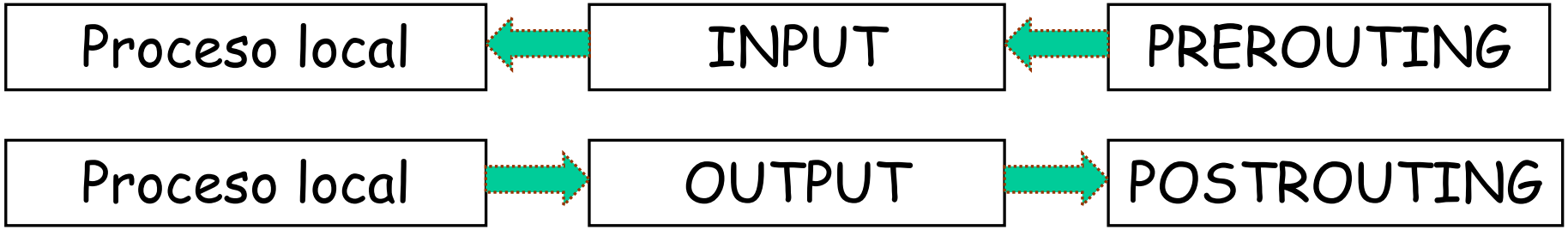


En la figura se muestra las tablas que hay en cada etapa y el orden en el que se consultan.

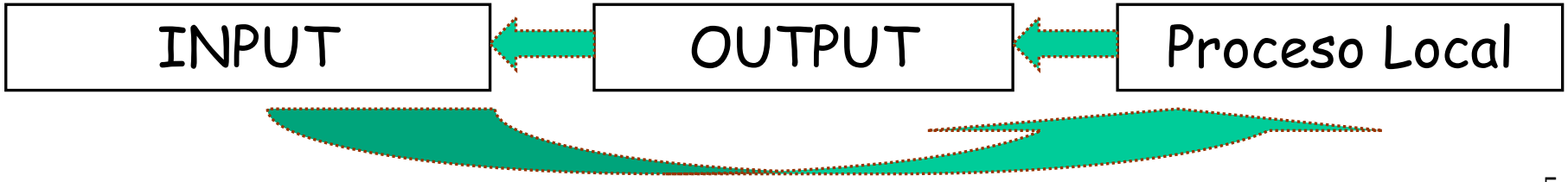
1. ETAPAS QUE SIGUE UN PAQUETE DENTRO EN LAS RUTINAS TCP/IP

Camino del routing

Llamada al firewall



Llamada localhost desde el firewall



1. TABLAS DE LAS IPTABLES

- La tabla mangle permite añadir reglas que cambian algunos campos de los paquetes, como TTL o TOS.
- La tabla nat permite añadir reglas que cambian las direcciones IP de los paquetes. Los tipos de reglas son: SNAT para cambiar la dirección fuente y DNAT para cambiar la dirección destino.
- La tabla filter permite añadir reglas de filtrado. Los tipos de reglas son: DROP para rechazar un paquete y ACCEPT para aceptarlo.

2. CREACIÓN DE REGLAS CON EL COMANDA IPTABLES

El formato genérico del comando iptables es:

```
iptables [-t <taula>] <comando> <expresión> -j <tipo de regla>
```

- Si no se indica la tabla, se considera que la regla se aplica a la tabla filter.
- La expresión (o match) identifica cuales son los paquetes a los que hay que aplicar la regla.
- El tipo de regla (o target) identifica que es lo que se debe de hacer con los paquetes (DROP, ACCEPT, SNAT o DNAT).

2. COMANDOS DE LAS IPTABLES

- -A <chain>: añade una regla a una tabla de la etapa <chain>. Por ejemplo: iptables –A INPUT ...
- -D <chain> <n>: borra la regla <n> de una taula de la etapa <chain>. Las reglas se enumeran a partir de 1.

Por ejemplo: iptables –t <taula> –D <chain> <n>

- -I <chain> <n> ...: inserta una regla en la línea indicada.

Por ejemplo: iptables –I INPUT 1 --dport 80 –j ACCEPT

- -L <chain>: lista las reglas de una tabla de <chain>. Admite la opción –n para que no intente traducir las direcciones numéricas a nombres, -v porque sea más "verbose" y --line para mostrar el número de regla.

Por ejemplo, iptables –t <tabla> --line –nvL <chain>.

2. COMANDOS DE LAS IPTABLES

❖ -P <chain>: especifica el tipo de regla por defecto de una taula de <chain> (análogo a la comanda que rechaza todos los paquetes que hay al final de una lista de acceso en un router Cisco). Si esta comanda no se ejecuta, la regla por defecto es ACCEPT.

Por ejemplo, para que la regla por defecto en la tabla filter de la etapa INPUT sea descartarlo todo ejecutaríamos: iptables –P INPUT DROP.

3. EXPRESIONES GENERICAS

❖ - p <protocol>: identifica el protocolo

Para identificar los protocolos se puede utilizar el número de protocolo o el nombre (fichero /etc/protocols), asi como: icmp, udp, tcp, ip, etc

Se puede utilizar el nombre ALL para identificar todos los protocolos (valor por defecto si no se utiliza -p)

Se puede invertir la expresión con el operador !

Ejemplos:

-p ! tcp → Indica cualquier protocolo excepto TCP.

iptables –A INPUT –p tcp ... → Añadirá a la tabla filter de la etapa INPUT una regla que se aplicará a todos los paquetes TCP.

3. EXPRESIONES GENERICAS

❖ -s <@IP fuente>: Identifica todos los paquetes con <@IP fuente>. Se puede añadir una máscara para identificar un rango de direcciones y se puede utilizar el operador ! para negar la expresión.

Ejemplo:

iptables –A INPUT –s ! 192.168.0.0/24 → añadirá a la tabla filter de la etapa INPUT una regla que se aplicará a todos los paquetes que tengan una @IP fuente que no pertenezca al rango 192.168.0.0/24.

❖ -d <@IP destino>: Idem que antes para la dirección destino.

3. EXPRESIONES GENERICAS

❖ -i <interfície-entrada>: Identifica una interfície de llegada de paquetes. Sólo se puede aplicar en las etapas INPUT, FORWARD y PREROUTING.

Ejemplo:

iptables –A INPUT –i eth0... → añadirá una regla que se aplicará a todos los paquetes que lleguen por la interfície eth0.

❖ -o <interfície-salida>: Idem que antes para una interfície de salida. Sólo se puede aplicar en las etapas OUTPUT, FORWARD y POSTROUTING.

4. EXPRESIONES PARA TCP Y UDP

Sólo se pueden aplicar a los protocolos TCP y UDP, por tanto, la expresión ha de comenzar con –p tcp ó –p udp.

- --sport <puerto-fuente>: Identifica los paquetes TCP que tienen el <puerto-fuente>. Se puede utilizar un número o uno de los nombres de /etc/services.

También se puede especificar un rango: inicial:final. Si no se indica el valor inicial es 0 y si no se indica el final es 65535. Por ejemplo:

iptables –A INPUT –p tcp –-sport 1024: → aplicará la regla a todos los paquetes TCP que tengan un número de puerto 1024.

- --dport <port-destino>: Idem que antes por el puerto destino.

4. EXPRESIONES SÓLO PARA TCP

- --tcp-flags <lista de flags> <lista de flags a 1>: Mira los paquetes de la <lista de flags> que sólo tienen a 1 los flags de la <lista de flags a 1>. Los flags pueden ser SYN, FIN, ACK, RST, URG, PSH, ALL, NONE.

Ejemplo:

iptables –p tcp -- tcp-flags SYN,RST,ACK SYN ...→ aplicará la regla a todos los paquetes que tienen el flag de SYN activado y los de RST y ACK a 0.

4. EXPRESIONES PARA ICMP

Sólo se pueden aplicar al protocolo ICMP, por tanto, la expresión ha de comenzar con –p icmp.

- --icmp-type <tipo>: identifica los paquetes icmp del tipo <tipo>. Ejecutando iptables –p icmp –-help nos da un listado de los posibles tipos.

4. EXPRESION DE ESTADO

- Este tipo de expresión se considera "no implícita", y se debe de añadir –m state para poder aplicarla.
- Los estados pueden ser:
  - NEW. Cuando se recibe un paquete que se identifica como el de una conexión nueva, la conexión se registra como NEW. Afecta a los paquetes TCP, UDP y ICMP (aunque UDP y ICMP sean no orientados a la conexión)
  - ESTABLISHED. Si se recibe un paquete en sentido contrario asociado a una conexión registrada como NEW, entonces la conexión se registra como ESTABLISHED.

4. EXPRESION DE ESTADO

- RELATED. Si se detecta una nueva conexión que se interpreta que es consecuencia de una ya existente (por ejemplo, porque se deduce que un paquete ICMP de error se ha generado como consecuencia de una conexión ya establecida), entonces la nueva conexión se registra como RELATED.

- --state <lista de estados>: identifica los paquetes de una conexión que está en uno de los estados de <lista de estados>.

Por ejemplo: iptables –m state –-state RELATED, ESTABLISHED ... aplicará la regla a los paquetes de las conexiones que estén en uno de estos estados.

5. TIPOS DE REGLAS

Los tipos de reglas pueden ser: ACCEPT, DROP, SNAT y DNAT.

ACCEPT → Acepta el paquete

DROP → Descarta el paquete.

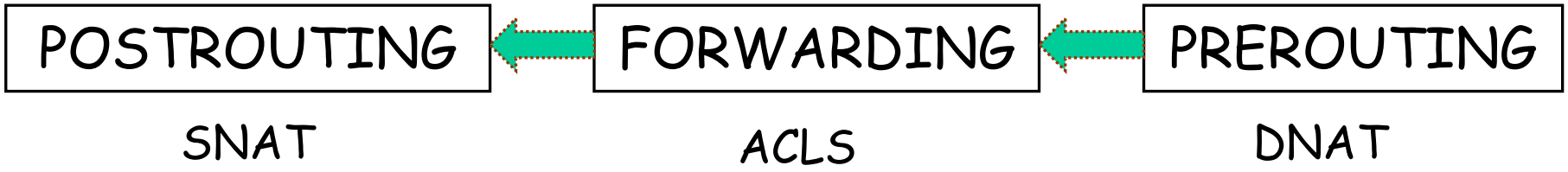
En caso de ser aceptado/descartado, se dejan de mirar el resto de reglas de la tabla y se continua con las reglas del resto de tablas y etapas que queden por atravesar.

SNAT → NAT en la fuente. Para conexiones iniciadas dentro de la red. Es equivalente a PAT (NAT por puertos)

DNAT → NAT en el destino. Para conexiones iniciadas fuera de la red. Por tanto lo primero que se hace es una traducción del destino. Es equivalente a NAT estático.

5. LA REGLAS ACCEPT

Etapas en el camino de routing



Definición de reglas ACL →
iptables –A FORWARD –i eth0 –o eth1 –j ACCEPT
iptables –A FORWARD –i eth0 –o eth1 –m state –state ESTABLISHED,RELATED –j ACCEPT

Se permite el paso de todo paquete que entra por eth0 y sale por eth1.

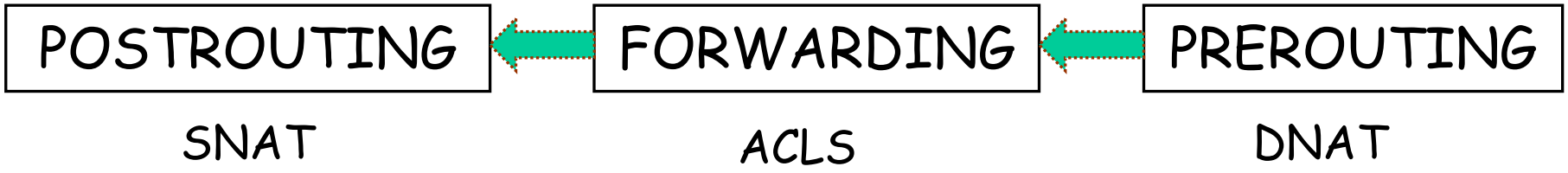
Se permite el paso de todo paquete que entra por eth1 y sale por eth0 siempre que sea una conexión ya iniciada o establecida.

5. LA REGLA DNAT

- Este tipo de regla se utiliza con la opción --to-destination para indicar la dirección donde debe hacerse la traducción.
- Sólo se puede poner en la tabla nat de las etapas PREROUTING y OUTPUT.
- Se puede especificar una dirección o un rango de direcciones (y se distribuirán las conexiones aleatoriamente entre las direcciones indicadas, es decir, se hará un load-balancing).
- En el caso de una expresión TCP o UDP también se puede indicar un puerto o rango de puertos (para distribuir las conexiones entre el rango de puertos).

5. LA REGLA DNAT

Etapas en el camino de routing



Definición de un DNAT → iptables –t nat –A PREROUTING –p tcp –i eth0 –d 200.10.10.5 --dport ssh –j DNAT --to-destination 192.168.1.2

Todo paquete TCP con servicio ssh que entre por la interficie eth0 con destino 200.10.10.5, se traducirá esa dirección destino por la dirección 192.168.1.2, que es la dirección interna privada del servidor ssh.

200.10.10.5 → Dirección pública externa del servidor ssh

192.168.1.2 → Dirección privada interna del servidor ssh

5. LA REGLA SNAT

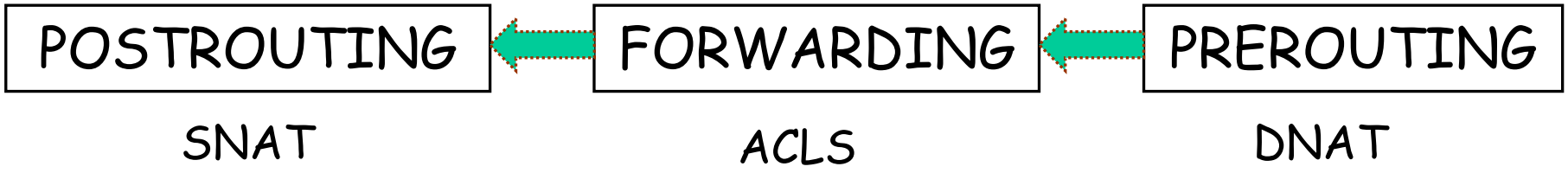
- Tiene una sintaxis análoga a la de DNAT, pero con la opción --to-source.
- Sólo se puede poner en la tabla nat de la etapa POSTROUTING.

iptables –t nat –A POSTROUTING –p tcp –o ppp0 –j SNAT --to-source 200.10.10.10-200.10.10.20:1024-32000

Cambiará las direcciones fuente de los paquetes que salgan por la interficie ppp0 por una de las direcciones del rango 200.10.10.10-200.10.10.20 y puerto del rango 1024-32000. Si no se indica el rango de puertos y es necesario cambiarlo (para que se haga PAT), entonces los puertos inferiores a 512 se mapean a otro puerto inferior a 512, e igualmente para los rangos entre 512-1023 y 1024 (para conservar la semántica del puerto). A los paquetes de una misma conexión se les aplica la misma traducción de dirección/puerto

5. LA REGLA SNAT

Etapas en el camino de routing

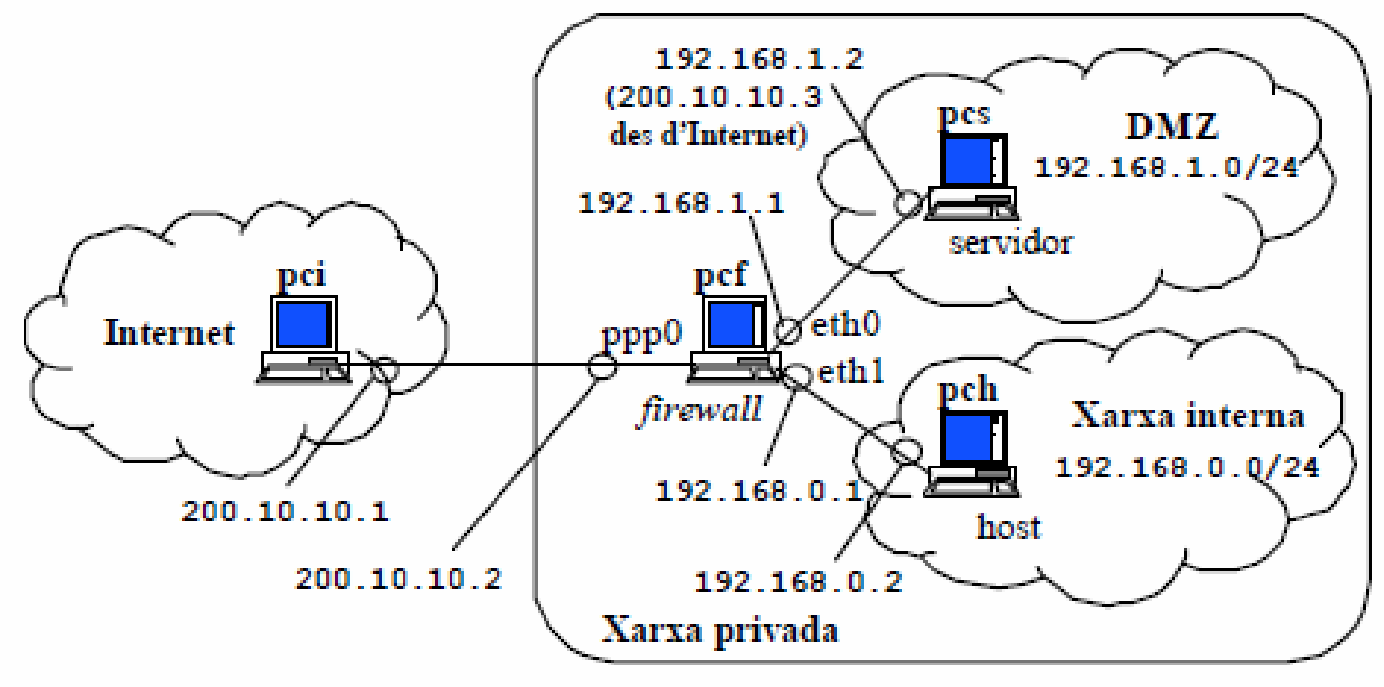


Definición de un SNAT → iptables –t nat –A POSTROUTING –o eth0 –j SNAT --to-source 200.10.10.5

Todo paquete que salga por la interficie eth0, se le traducirá su dirección privada origen por la dirección pública 200.10.10.5 para que pueda circular por Internet

PRACTICA 12. CONFIGURACIÓN DE IPTABLES EN LINUX

Sea la siguiente red:



PRACTICA 12. CONFIGURACIÓN DE IPTABLES EN LINUX

Características de la red

- El PC 200.10.10.1 (pci) representa Internet y está conectado por un enlace punto a punto al firewall (pcf) de la red privada.
- El PC 200.10.10.1 (pci) representa Internet y está conectado por un enlace punto a punto al firewall (pcf) de la red privada.
- En la red privada hay una red DMZ donde hay el servidor (pcs). Es el único lugar donde se puede acceder desde Internet. Además hay una red interna con pch que ha de tener acceso al servidor.
- Hemos contratado dos direcciones públicas en el ISP 200.10.10.2-200.10.10.3: una identifica el firewall (200.10.10.2) y la otra la utilizaremos para poder salir de la red privada.
- pci no ve pcs ni pch, debido a que pci pertenece a Internet y no puede encaminar paquetes hacia las direcciones privadas.

PRACTICA 12. CONFIGURACIÓN DE IPTABLES EN LINUX

Paso 1. Si ejecutamos la siguiente comanda en pcf:

pcf# iptables –t nat –A POSTROUTING –o ppp0 –j SNAT --to-source 200.10.10.3

¿Qué hace? Comprobar que ahora pcs y pch ven pci pero no viceversa, ¿por qué?

Paso 2. Ejecutar la comanda:

pcf# iptables –t nat –A PREROUTING –p tcp –i ppp0 –d 200.10.10.3 -- dport ssh –j DNAT --to-destination 192.168.1.2

¿Qué se ha programado? ¿Qué trafico se habilita?

¿Qué pasa si pci hace un ping a pcs?.

Paso 3. ¿Qué regla se tendría que ejecutar para que se pueda hacer ping desde pci a pcs? ¿y para que se pueda conectar al servidor de ftp de pcs?

PRACTICA 12. CONFIGURACIÓN DE IPTABLES EN LINUX

Paso 4. Ejecuta los comandos:

pcf# iptables –P INPUT DROP

pcf# iptables –P OUTPUT DROP

pcf# iptables –P FORWARD DROP

Comprueba que ninguno de los PCs se ve. ¿Por qué?

Paso 5. Ejecuta los comandos:

pcf# iptables –A FORWARD –i eth0 –o ppp0 –j ACCEPT

pcf# iptables –A FORWARD –i ppp0 –o eth0 –m state –state ESTABLISHED,RELATED –j ACCEPT

Comprueba que pcs puede hacer ping a pci pero no viceversa. Comprueba que pcs puede acceder a cualquier servicio de pci (por ejemplo, ssh y ftp) pero no viceversa. ¿Por qué?

PRACTICA 12. CONFIGURACIÓN DE IPTABLES EN LINUX

Paso 6. Ejecuta la comanda:

pcf# iptables –A FORWARD –p tcp –i ppp0 –o eth0 –d 192.168.1.2 –j ACCEPT

Comprueba que ahora sí que pci puede hacer ssh a pcs pero pch no. ¿Por qué?

Comprueba que pci sólo puede acceder al servidor de ssh (comprueba por ejemplo que pci no puede hacer ftp a pcs), ¿Por qué?

Paso 7. Configura el *firewall* para que pci pueda hacer ping a pcs.

Paso 8. Ejecuta las comandas:

pcf# iptables –A FORWARD –i eth1 –o eth0 –j ACCEPT

pcf# iptables –A FORWARD –i eth0 –o eth1 –j ACCEPT

Ahora pcs y pch se ven, pero no ven pcf. ¿Por qué?

PRACTICA 12. CONFIGURACIÓN DE IPTABLES EN LINUX

Paso 9. Ejecuta los comandos:

pcf# iptables –A INPUT –p icmp –i eth1 –d 192.168.0.1 –j ACCEPT

pcf# iptables –A OUTPUT –p icmp –o eth1 –s 192.168.0.1 –m state --state ESTABLISHED,RELATED –j ACCEPT -j ACCEPT

Ahora pch puede hacer ping a pcf, pero no puede acceder a ninguno de los servicios de pcf. ¿Por qué? Ejecuta los comandos necesarios para que pcs tenga las mismas restricciones de acceso a pcf que pch.

Paso 10. Configura el *firewall* para que pch sea el único host que pueda conectarse al servidor de ssh de pcf. Pch no ha de poderse conectar a ningún otro servicio de pcf.

Paso 11. Configura el *firewall* para que pch pueda conectarse al servidor de ssh de pci, pero a ninguno de los servicios de pci ni cap altra host de Internet. pci no ha de poder iniciar ninguna conexión con pch, ni siquiera hacer ping.