Seguridad en Redes

Seguridad Perimetral

Seguridad Perimetral

Problemáticas en la red

- Accesos a servicios restringidos
- Escaneo de puertos
- Escaneo de vulnerabilidades
- Explotación de vulnerabilidades
- Denegación de Servicio
- Navegación irrestricta

Seguridad en Redes

Escaneo de puertos

Escaneo de puertos

nmap

El escaneo de red suele utilizarse para identificar los hosts que se encuentran disponibles dentro de una red.

Nos permitirá ver qué puertos se encuentran disponibles en cada host vivo.

Nos mostrará qué servicio se encuentra corriendo en el puerto descubierto.

En algunos casos nos permitirá descubrir vulnerabilidades sobre los servicios enumerados.

Dentro de un proceso de Pentesting, este proceso se encuentra en la fase de Enumeración.

nmap

nmap es una herramienta muy potente con la cual podremos trazar un mapa de nuestra red, descubriendo hosts vivos, puertos abiertos, servicios disponibles y algunas debilidades de los servicios.

¿Qué obtenemos si ejecutamos: *nmap ip.0wn3d ?*

Probemos...

nmap

Obtendremos los puertos abiertos que dispone el host escaneado. Por defecto escanea los 1000 puertos más utilizados según nmap y solo escanea TCP. Si queremos que escanee UDP tendremos que especificárselo.

Nmap nos permite :

- Trazar un mapa de la red.
- Conocer hosts vivos
- Identificar puertos activos
- Identificar servicios
- Identificar vulnerabilidades
- Entre otras acciones

nmap

Ahora sumemos a Wireshark al escaneo para ver lo que está haciendo nmap

nmap

Ejemplo 1:

nmap -sP 192.168.0.0/24

Con este comando identificaremos todos los hosts vivos en la red 192.168.0.0/24

nmap

Ejemplo 2:

nmap -sS -sV -O -sC -oA escaneo 192.168.0.21

-sS: Realiza escaneo de tipo SYN

-sV: Identifica servicios

-O: Identifica Sistema Operativo

-sC: Ejecuta scripts

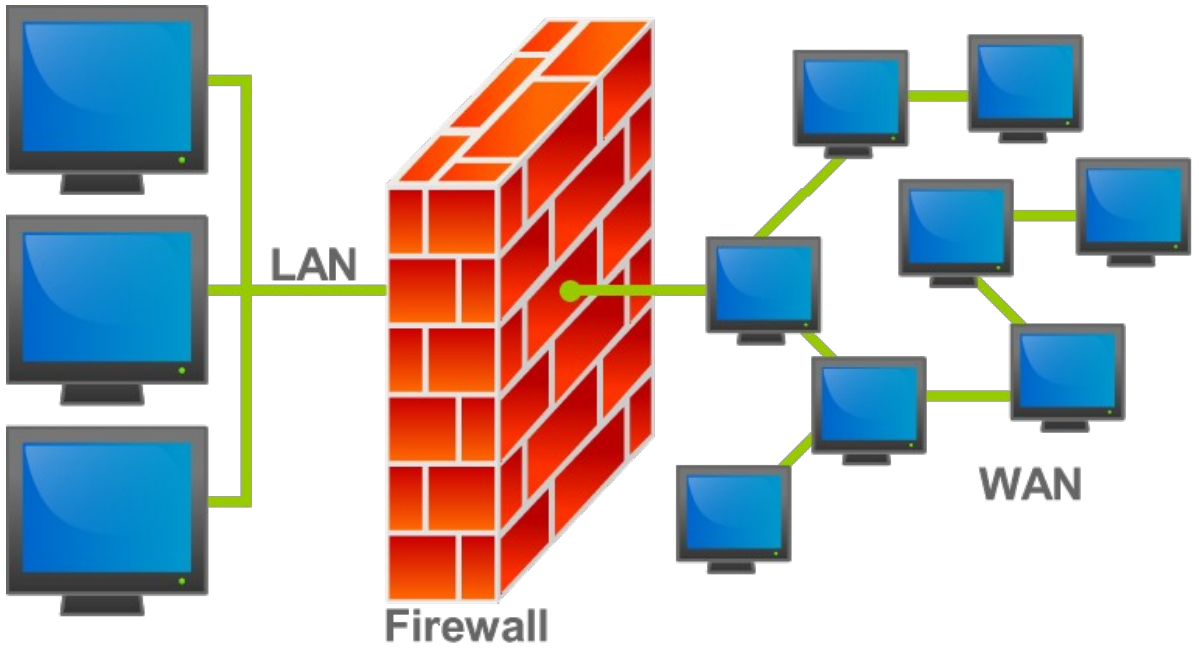
-oA: Guarda la salida en formato nmap, grepeable y XML

Seguridad en Redes

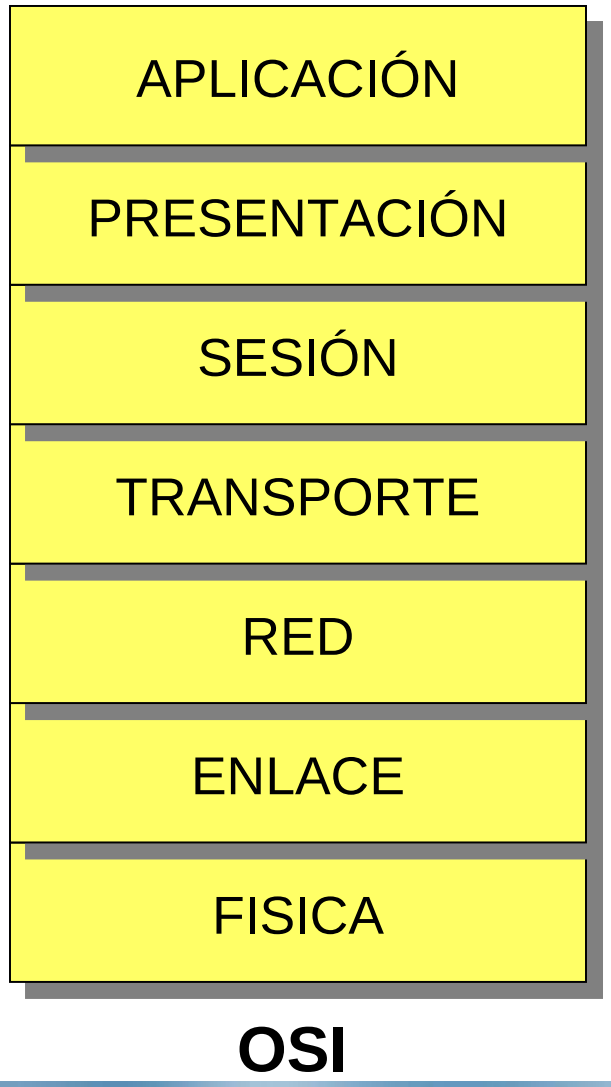
Firewall

Firewall

- Controla las comunicaciones, permitiéndolas o prohibiéndolas según las políticas de seguridad.

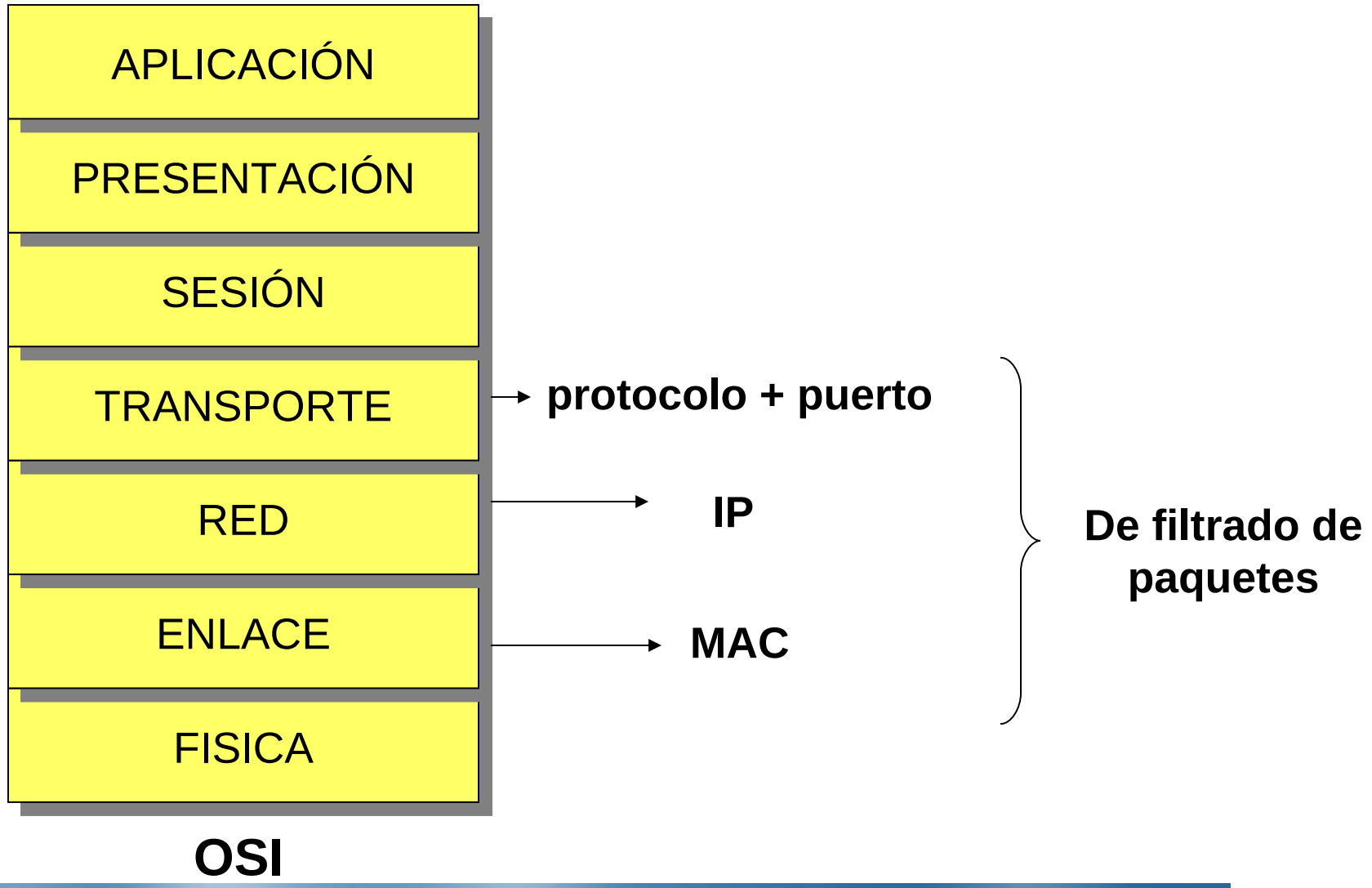


Firewall

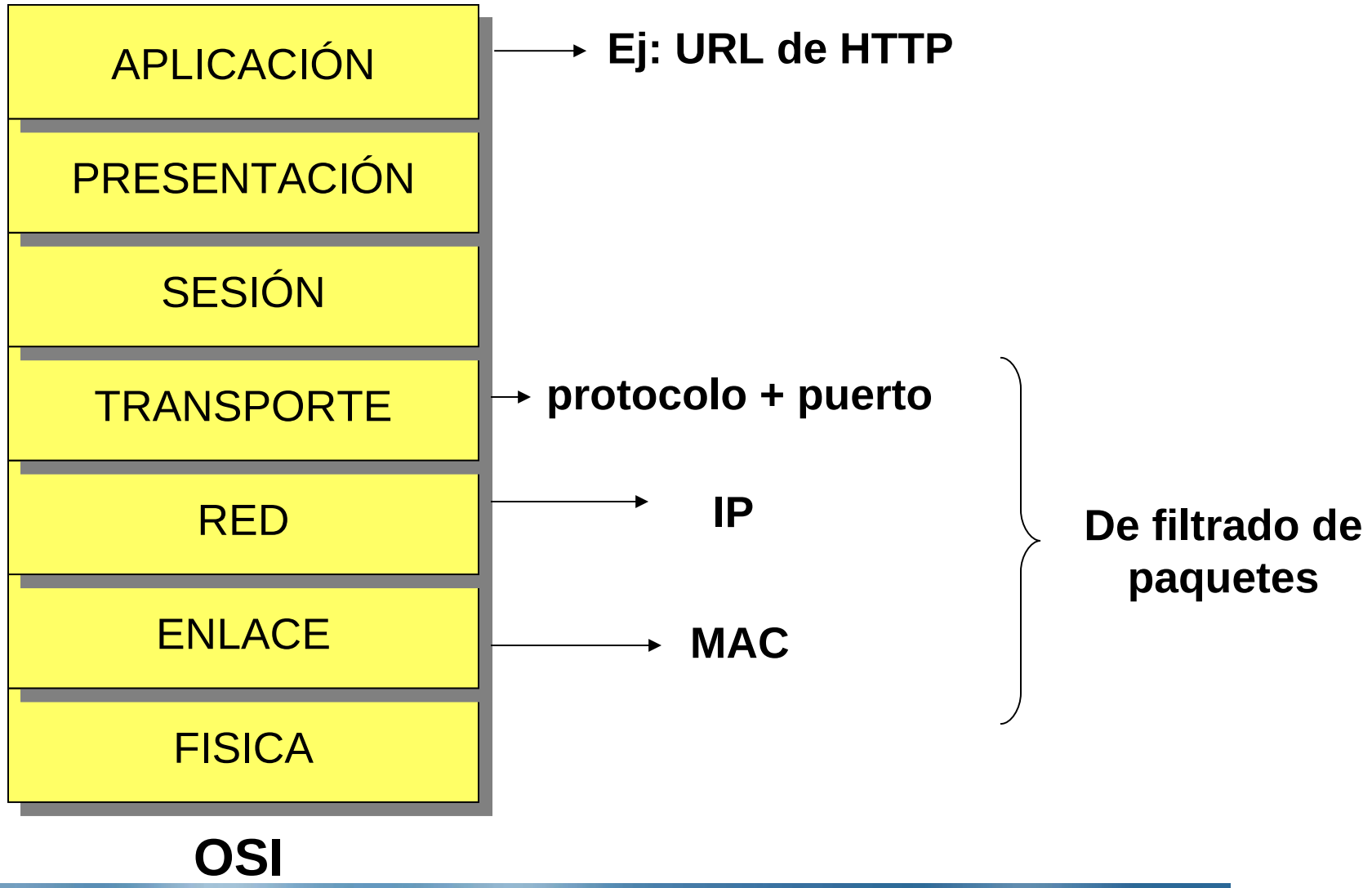


OSI

Firewall


APLICACIÓN
PRESENTACIÓN
SESIÓN
TRANSPORTE
RED
ENLACE
FISICA
OSI
protocolo + puerto
IP
MAC
De filtrado de paquetes

Firewall


APLICACIÓN
PRESENTACIÓN
SESIÓN
TRANSPORTE
RED
ENLACE
FISICA
OSI
Ej: URL de HTTP
protocolo + puerto
IP
MAC
De filtrado de paquetes

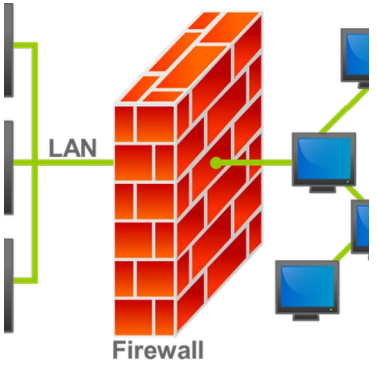
Políticas de Firewall

- Permisiva: se permiten todas las conexiones y se deniegan los rangos de puertos y protocolos

- Restrictiva: se deniega todo y únicamente se permite lo que es necesario que atraviese el FW

Firewall según su ubicación

Firewall de red: Firewall que intercomunica dos redes el cual permite o deniega el paso de una red a otra.



Firewall de host: Firewall que permite que los datos ingresen o egresen a un host determinado. Se encuentra instalado y configurado dentro del host a proteger.

Firewall según su tipo

Firewall por HW

Firewall por SW

```
root@core ~# iptables -L --line-numbers
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere
2    ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
3    ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
4    ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:8443
5    ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:12320
6    ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:12321
7    ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:25565
8    ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
9    ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:25517
10   ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:8442
11   ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
```

Firewall según su generación

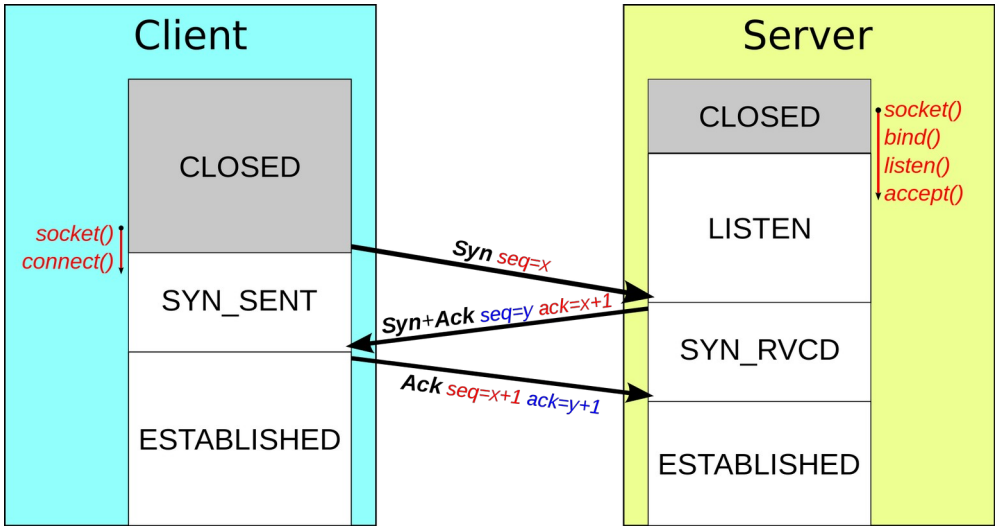
- 1ra. Generación o Stateless
- 2da. Generación o Stateful
- Unified Threat Management (UTM)
- Next Generation Firewall (NGFW)

Firewall Stateless

Los Firewall Stateless son aquellos que únicamente van a estar analizando IP origen, puerto origen, IP destino, puerto destino y protocolo.

Firewall Stateful

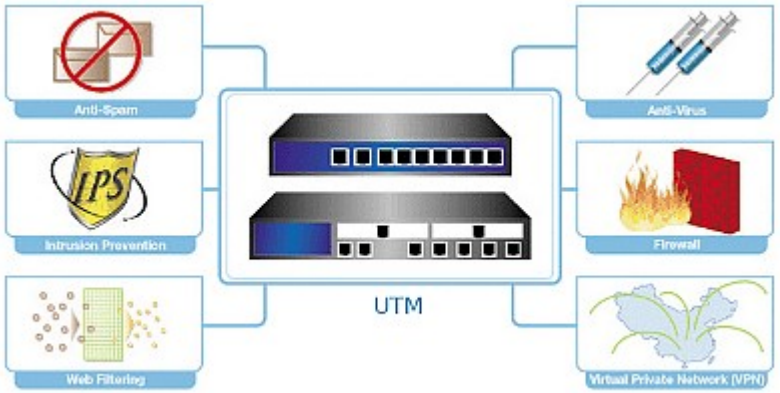
En este caso, el firewall va a poder controlar no solo IP origen, puerto origen, IP destino, puerto destino y protocolo, sino también el estado de la conexión.



Unified Threat Management (UTM)

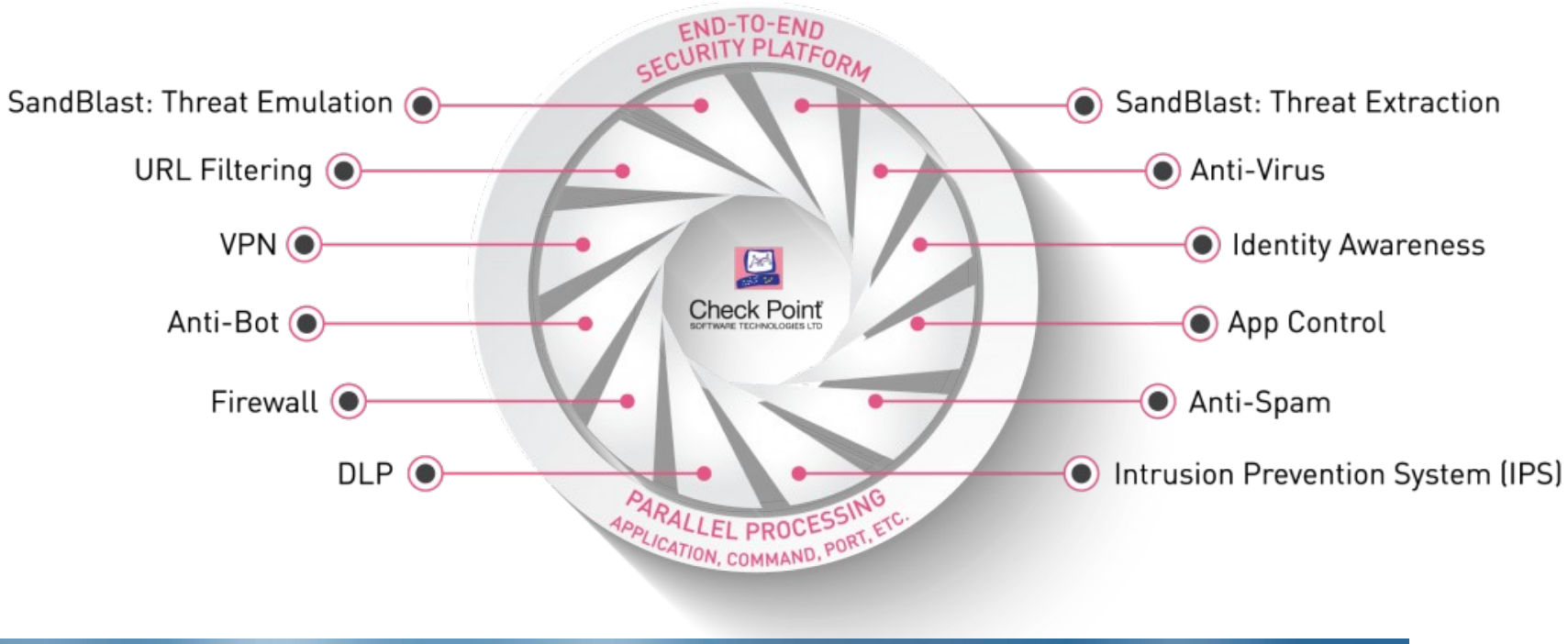
El UTM agrega múltiples funcionalidades al Firewall tradicional con el fin de reducir la cantidad de dispositivos. Están apuntados para las Pequeñas y Medianas Empresas.

- Firewall
- NAT
- VPN
- Antispam
- Filtro de contenidos
- Antivirus
- IDS/IPS



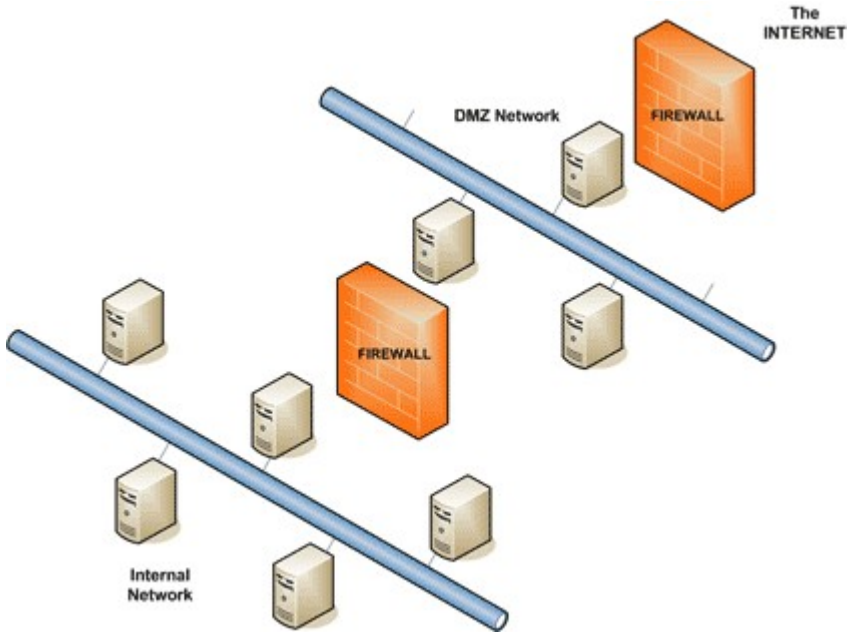
Next Generation Firewall (NGFW)

El NGFW agrega múltiples funcionalidades al Firewall tradicional con el fin de reducir la cantidad de dispositivos. Están apuntados para las Corporaciones debido a que maneja un mayor throughput que un UTM.



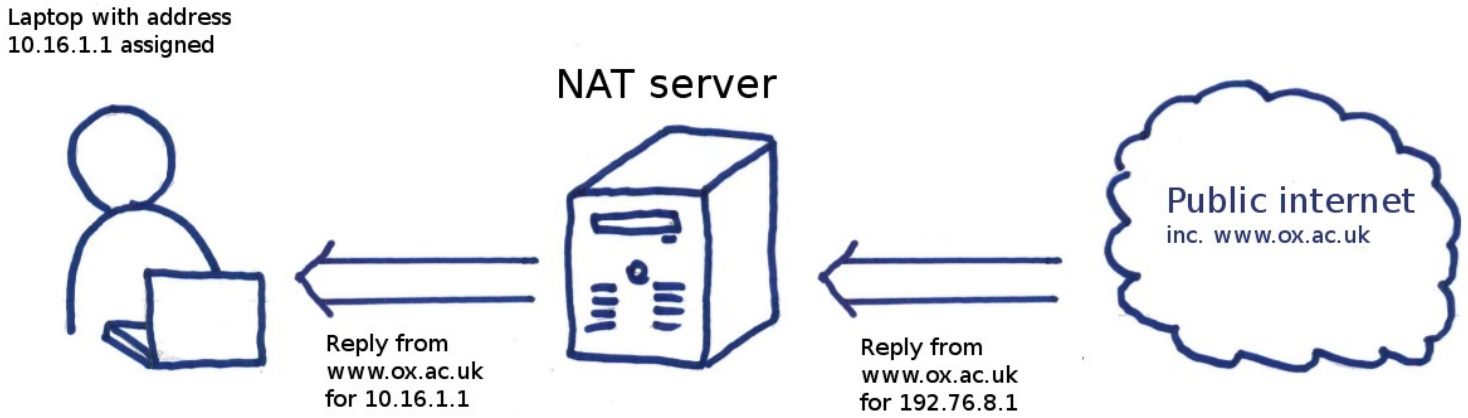
Zona Desmilitarizada (DMZ)

- En la DMZ, se ubican los servidores de la organización que deben permanecer accesibles desde la red exterior.
- De esta forma si un servicio publicado a Internet es comprometido el ciberdelincuente no estará dentro de nuestra LAN interna



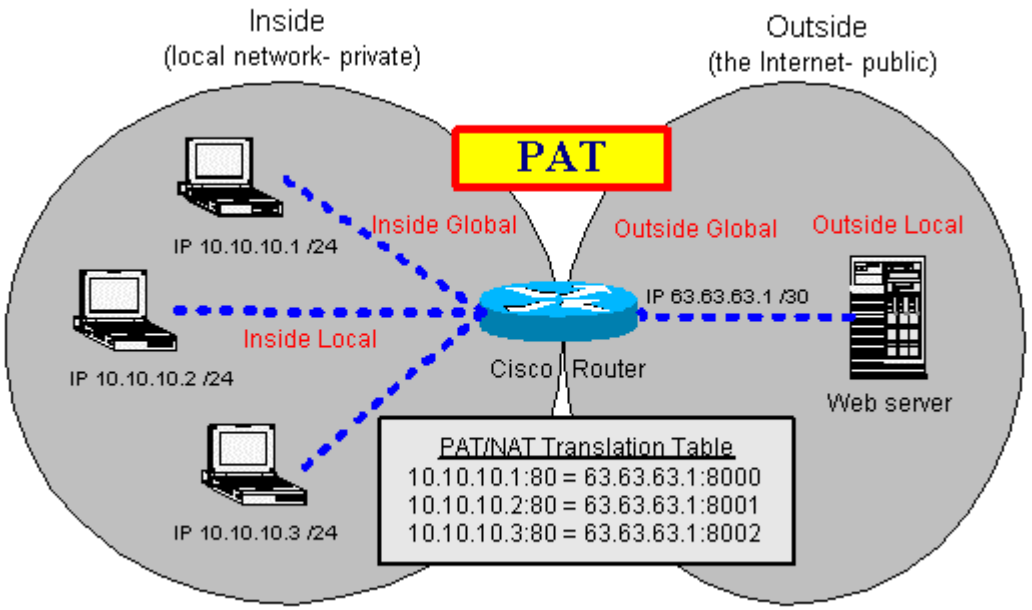
Network Address Translation (NAT)

- Es la conversión de una IP a otra IP.
- Existen 2 tipos de tablas NAT: estáticas y dinámicas.
  - Las tablas dinámicas se utilizan generalmente en la salida a Internet
  - Las tablas estáticas suelen utilizarse dentro de nuestras redes. Ejemplo: entre la DMZ y la LAN de Servidores
- En algunos casos es por necesidad y en otros casos es por seguridad
- Por necesidad: desde nuestra red hacia Internet
- Por seguridad: desde la DMZ hacia la red interna.

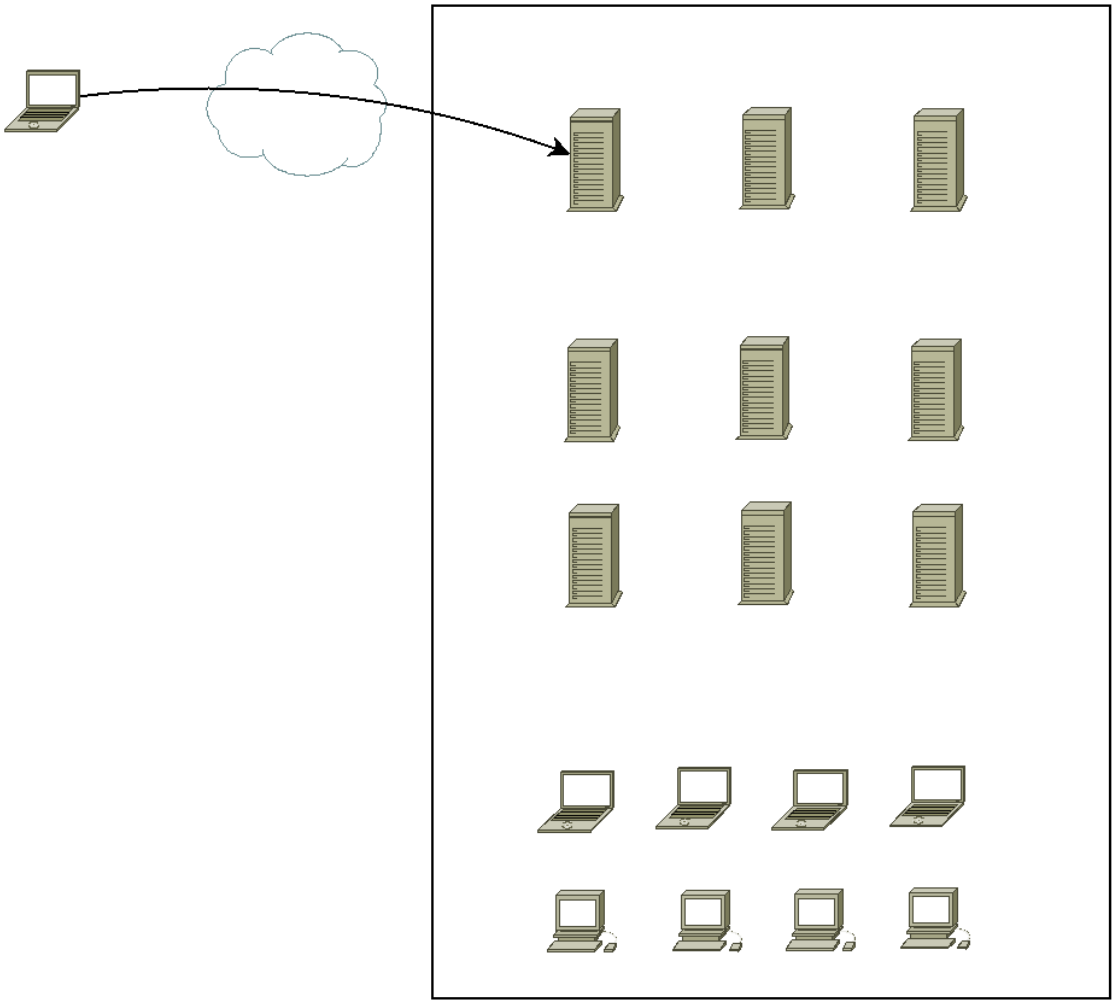


Port Address Translation (PAT)

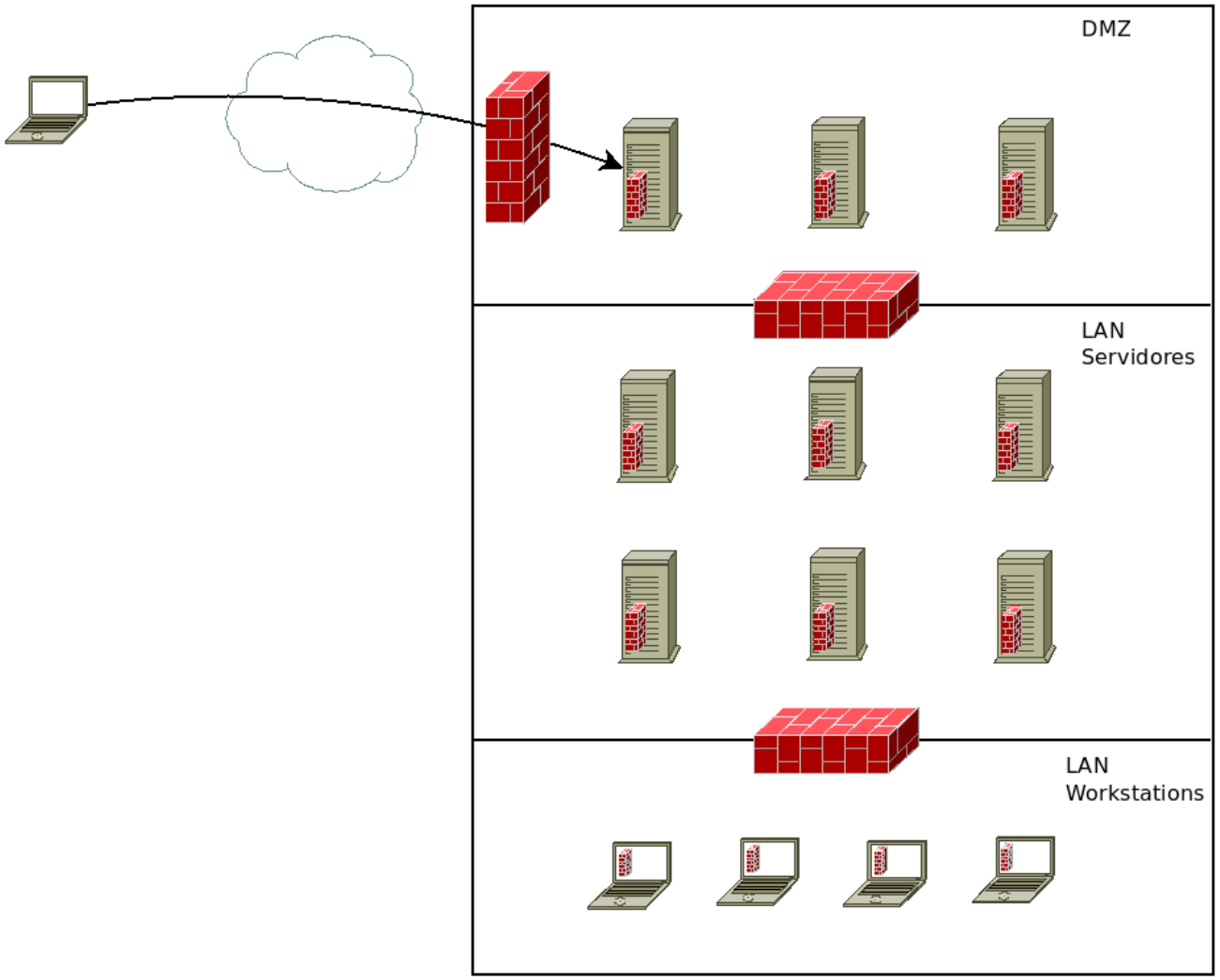
- Es la conversión de un puerto hacia otro puerto.



Ejemplo



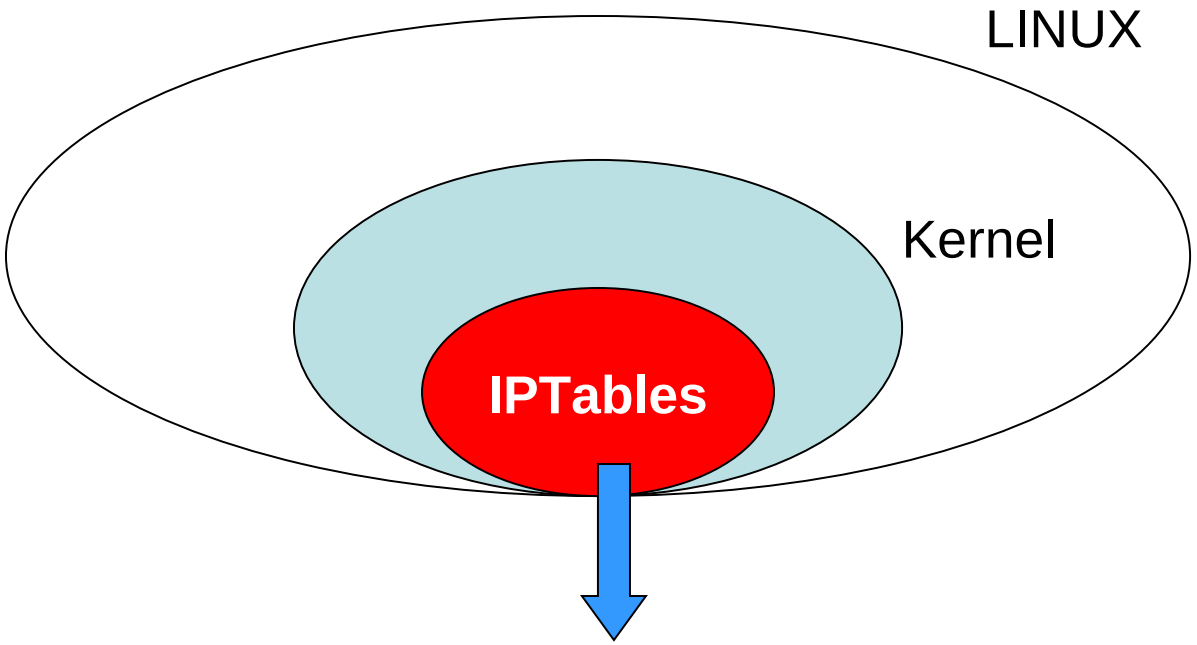
Ejemplo



LINUX IPTables

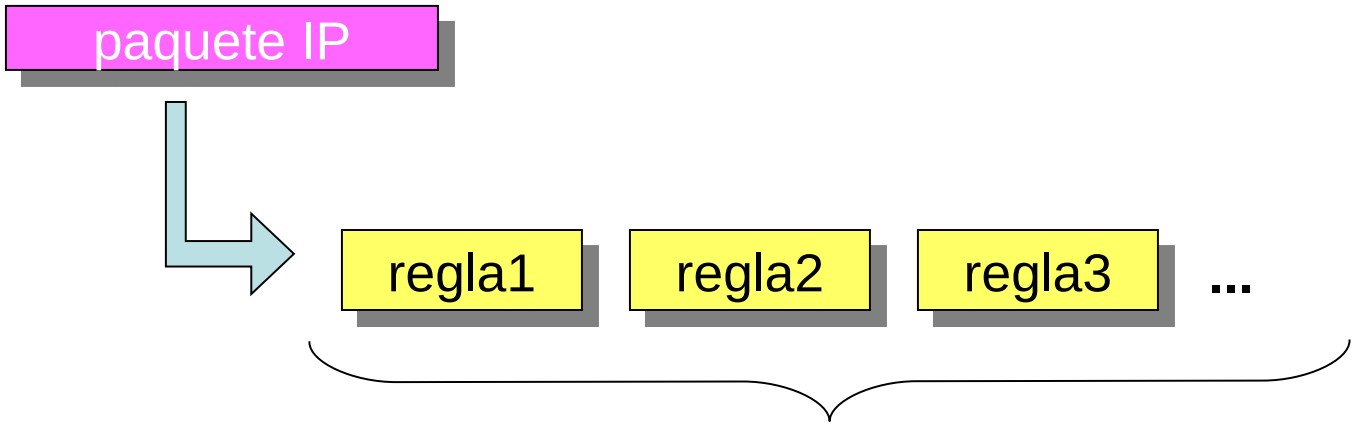
iptables

= NETFILTER



- **Filtrado de paquetes**
- **“Connection tracking”**
- **NAT**

iptables



cadenas básicas

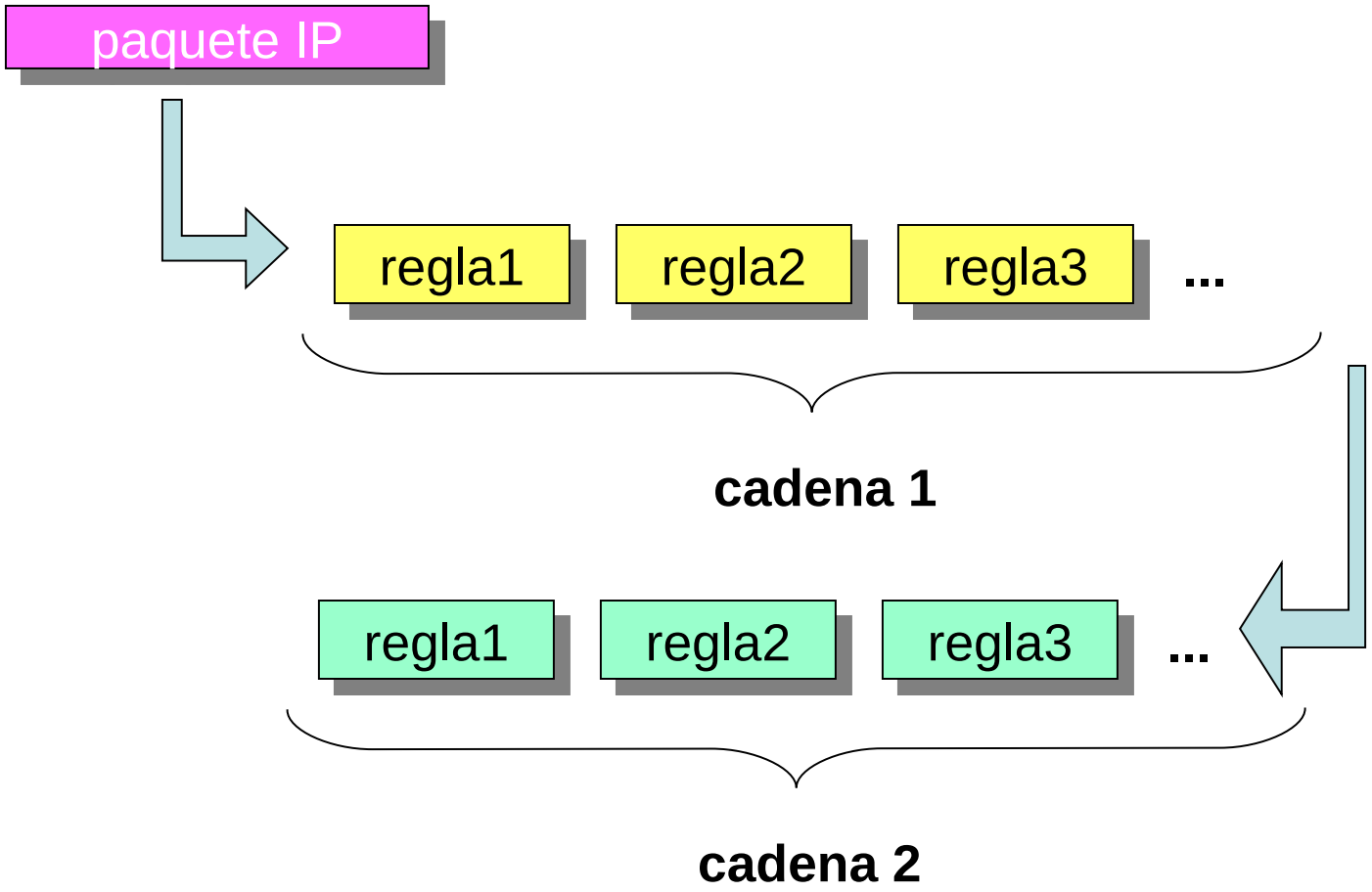
INPUT

OUTPUT

FORWARD

... el usuario puede crear tantas como desee.

iptables



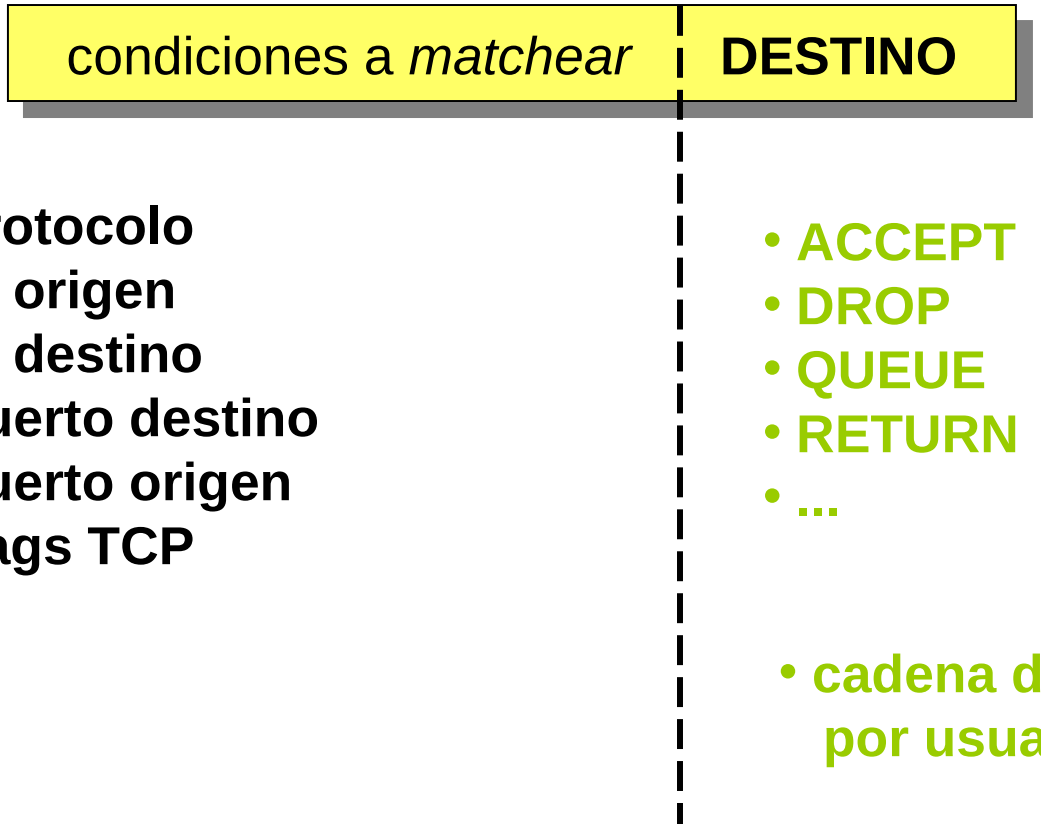
... enlace a otra cadena

iptables

- **ACCEPT**
- **DROP**
- **QUEUE**
- **RETURN**
- **...**

- **cadena definida por usuario**

iptables

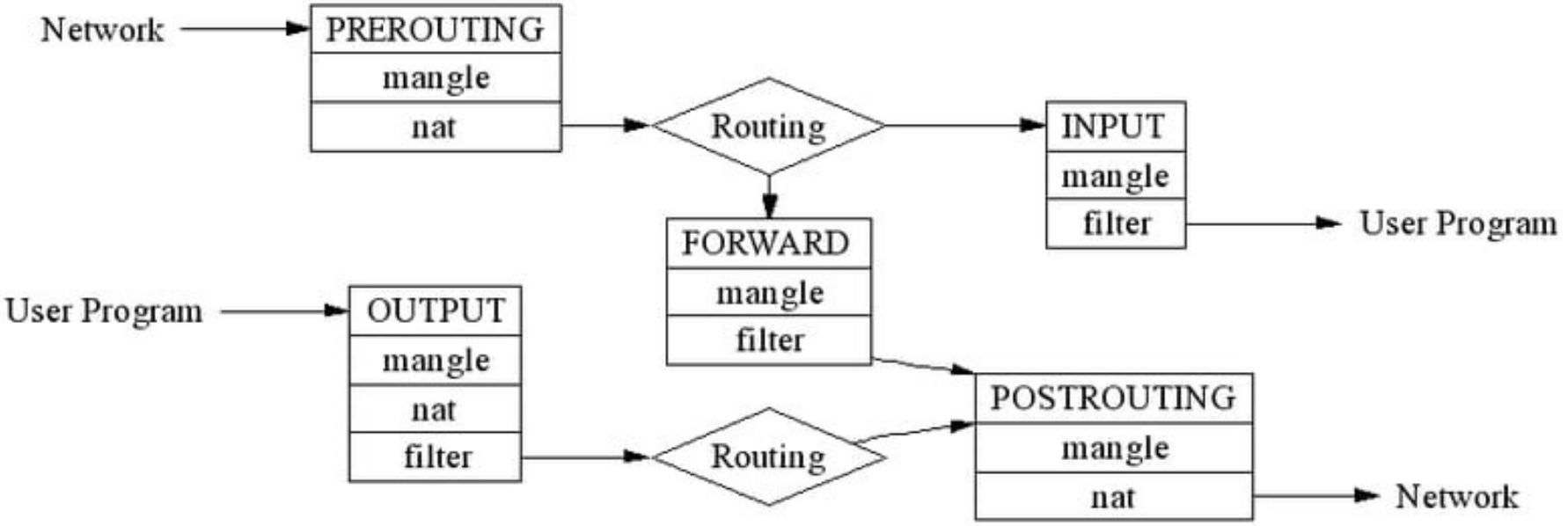

condiciones a *matchear*
DESTINO
• protocolo
• IP origen
• IP destino
• puerto destino
• puerto origen
• flags TCP
• ...
• ACCEPT
• DROP
• QUEUE
• RETURN
• ...
• cadena definida por usuario

iptables

Hay tres tablas ya incorporadas, cada una de ellas contiene ciertas cadenas predefinidas :

- filter
  - responsable del filtrado
  - cadenas predefinidas
    - INPUT
    - OUTPUT
    - FORWARD
- nat → responsable de configurar las reglas de reescritura de direcciones o de puertos de los paquetes
  - PREROUTING (DNAT)
  - POSTROUTING (SNAT)
  - OUTPUT (DNAT local)
- mangle → responsable de ajustar las opciones de los paquetes, como por ejemplo la calidad de servicio; contiene todas las cadenas predefinidas posibles.

Funcionamiento de iptables



iptables – Parámetros comunes

-t : Selecciona la tabla, por defecto trabaja con *filter*

-L : Muestra las reglas activas

-v : Muestra en modo verboso

-n : Muestra en formato numérico

-F : Borra todas las reglas de la tabla

-Z : Pone los contadores en cero

-P : Cambia la Política de la CADENA

-A : Agrega una regla al final de la CADENA

-I : Inserta al principio una regla en la CADENA

-p : Indica el protocolo con el cual se va a trabajar

-m : Incorpora un módulo

-s : Determina el origen del paquete (IP)

-d : Determina el destino del paquete (IP)

-j : Indica la acción a realizar

Acciones terminantes: ACCEPT, DROP, REJECT

iptables – Extensiones

Las extensiones se caracterizan por utilizar “--”. Por ejemplo cuando utilizo el “-p tcp” podría utilizar las siguientes extensiones:

--dport : Puerto destino
--sport : Puerto origen

iptables – Reglas básicas

```
# Abrir el puerto TCP/80 para todos
iptables -A INPUT -p tcp --dport 80 -j ACCEPT

# Descartar el tráfico que venga desde 3.3.3.3
iptables -A INPUT -s 3.3.3.3 -j DROP

# Aceptar paquetes de conexiones ya establecidas
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Escribir logs
iptables -A INPUT -p tcp -j LOG --log-prefix "INPUT packets"

# Listar reglas configuradas
iptables -L -n -v

# Cambiar Política
iptables -P INPUT DROP
```

iptables – Ejemplo básico Workstation

```
iptables -F
iptables -X
iptables -Z

iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j
    ACCEPT
```

iptables – Ejemplo básico Web Server

```
iptables -F
iptables -X
iptables -Z

iptables -P INPUT DROP
iptables -P OUTPUT DROP

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A INPUT -p tcp --dport 80 -j ACCEPT

iptables -A INPUT -p tcp --dport 22 -s 192.168.5.5 -j ACCEPT

iptables -A INPUT -p icmp -j ACCEPT

iptables -A OUTPUT -p udp --dport 53 -d 8.8.8.8 -j ACCEPT
```