

Grado en Ingeniería del Software
Doble Grado en Matemática Computacional e Ingeniería del Software
Doble Grado en Física Computacional e Ingeniería del Software



Redes de Ordenadores

Tema 7

Dr. Constantino Malagón Luque
Dr. Rafael Socas Gutiérrez

Septiembre 2024



7 Seguridad en Redes de Ordenadores

1) Redes de Ordenadores e Internet

2) Nivel de Aplicación

3) Nivel de Transporte

4) Nivel de Red

5) Nivel de Enlace: Redes de Acceso y LAN

6) Redes Inalámbricas y Redes Móviles

7) Seguridad en Redes de Ordenadores

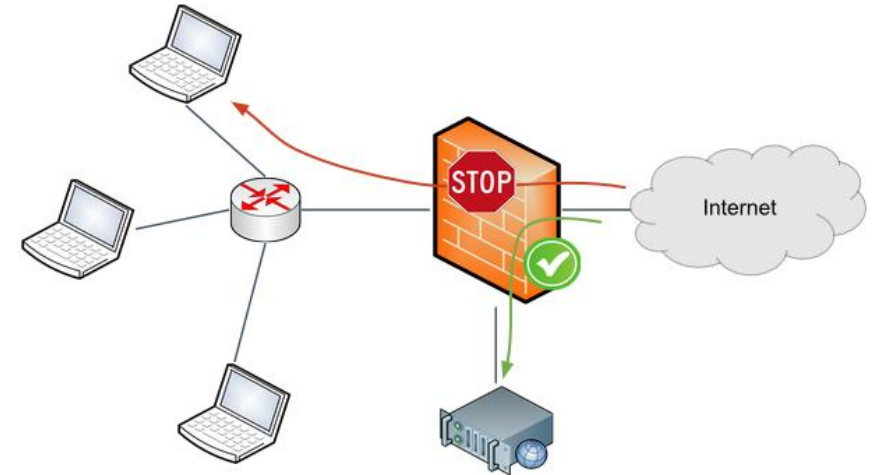
Seguridad: Introducción a los Firewalls

- **Los firewalls son dispositivos de hardware dedicados** que se colocan estratégicamente en una red, o que se ejecutan como **software en cualquier host o router**. El router de tu proveedor de Internet (ISP) lo más probable es que incluya también un firewall. También tu sistema operativo contiene un firewall.
- Un firewall **inspecciona** paquetes IP y los **filtra**. Este elemento contiene **políticas**, que básicamente son **reglas** que definen que sucede con los paquetes que cumplen ciertos criterios. Dependiendo de cómo se configuren, las políticas deciden si se descarta un paquete que proviene de Internet o si éste puede alcanzar una máquina. Un ejemplo podría ser permitir sólo el tráfico que pertenece a las conexiones iniciadas por su equipo.
- ¿Cómo funciona el filtrado de tráfico de conexiones establecidas?. Para identificar una conexión se usa el **protocolo TCP**. Un firewall podría, por ejemplo, permitir a hosts dentro de la red a establecer una **conexión TCP (SYN, SYN+ACK, ACK)** con hosts en Internet. A esos hosts se les permite enviar paquetes de vuelta siempre y cuando pertenezcan al **stream TCP establecido**.



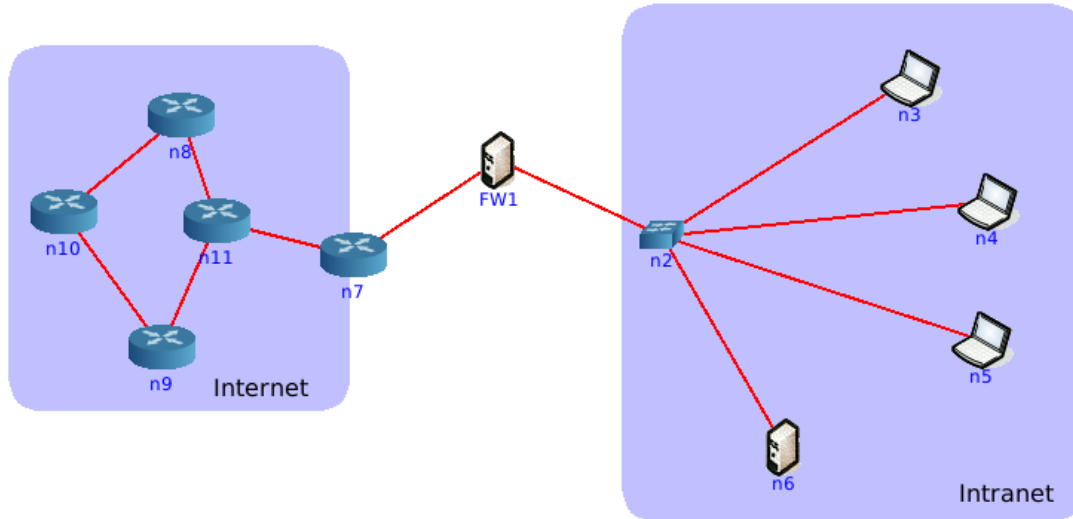
Seguridad: Modo de Funcionamiento

- Un firewall **separa diferentes partes de una red** entre sí.
- Imagine una configuración sencilla en la que se coloca un firewall en el único punto de conexión entre nuestra red e Internet. Esto significa que cada paquete que debe salir o entrar en nuestra red tiene que pasar a través del firewall. El trabajo del firewall es:
 - **Comprobar los paquetes que entran /salen** de la red en un único punto.
 - **Evitar que los atacantes y el tráfico no deseado** entren en la red.
 - **Restringir qué tráfico** puede salir de la red.
- Un firewall normalmente trabaja sobre **múltiples capas de red**. El tráfico de red puede ser gestionado basándose en la **información de Capa 2**, tal como direcciones MAC. Por ejemplo, en las WiFis públicas las direcciones MAC se analizan y el firewall decide si pasan los paquetes o los bloquea según sea esa MAC.
- Direcciones IP y puertos, **capa 3 y 4** respectivamente, pueden usarse también para filtrarse el tráfico. Un ejemplo puede ser filtrar el acceso a servidores web permitiendo sólo los puertos 80 (HTTP) y 443 (HTTPS). Los firewalls operan también en las capas superiores, **capa de aplicación**. Esto implica mirar la parte más interna de una trama Ethernet, el **payload de aplicación**, esto se denomina **Deep Packet Inspection (DPI)**.

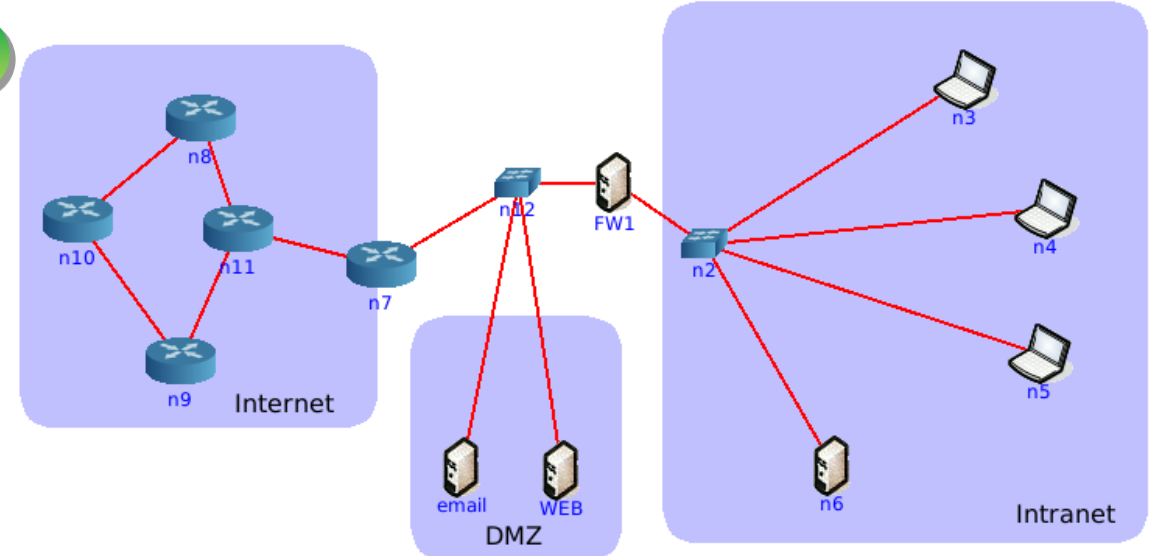


Seguridad: Escenarios de Tráfico

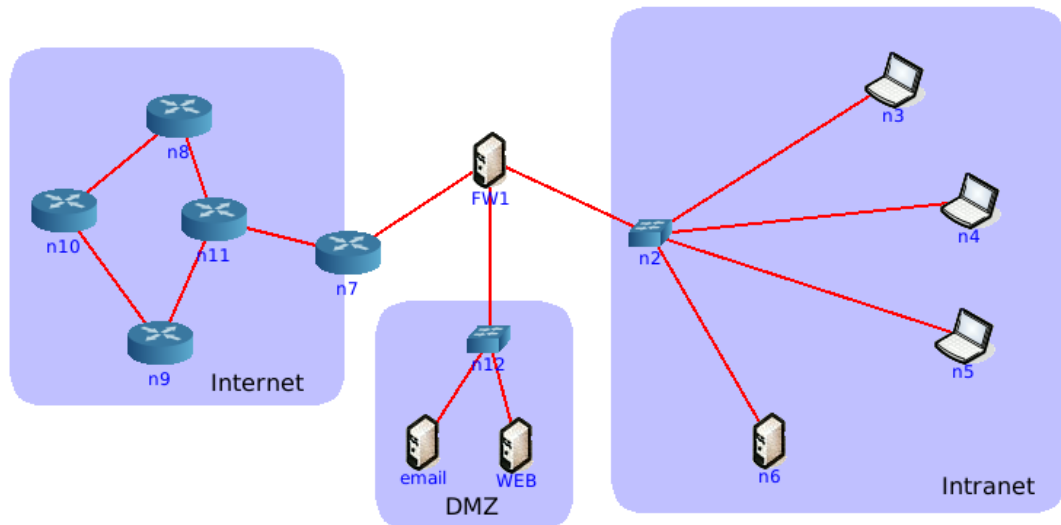
1



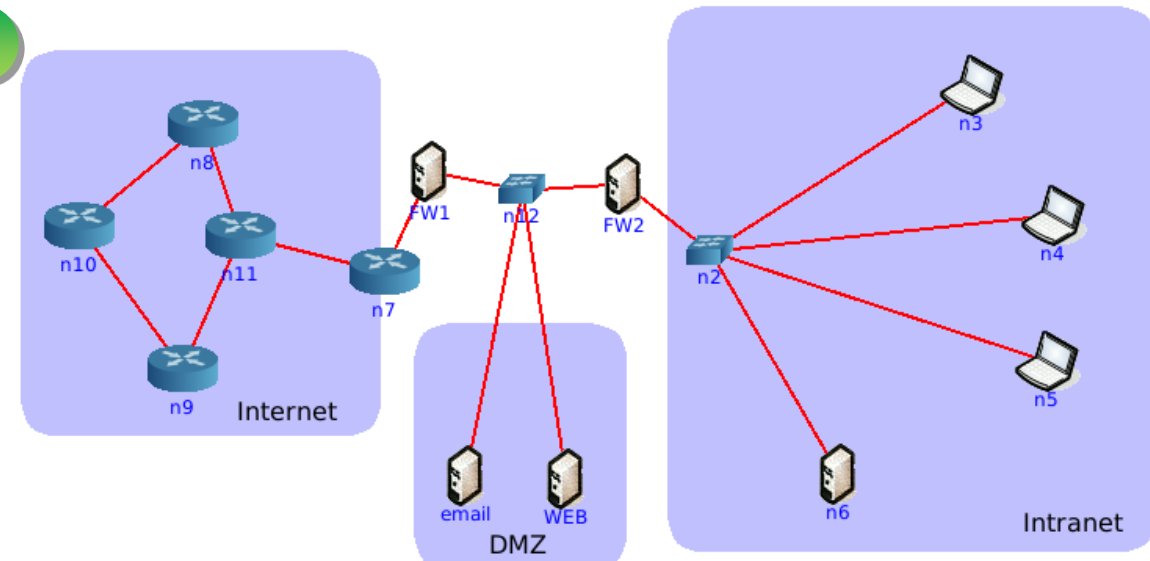
2



3



4



Seguridad: Políticas de Firewall

- Por defecto, un firewall no realiza ninguna acción sobre los paquetes. Necesita ser configurado con ciertas **reglas** y una **política por defecto**. Las políticas por defecto deciden qué sucede con los paquetes que no coinciden con ninguna de las reglas configuradas explícitamente. Las dos políticas más típicas son:
 - **Blacklisting**: dejar pasar todo lo que no está explícitamente bloqueado por una regla.
 - **Whitelisting**: bloquear todo lo que no está explícitamente permitido por una regla.
- La política por defecto se define por el administrador de la red.

Como resumen:

- **Blacklisting**:
 - Estrategia de permitir por defecto. Todo lo que no está explícitamente prohibido está permitido.
 - Menos segura.
 - Más fácil de configurar desde el punto de vista de la comodidad del usuario.
- **Whitelisting**:
 - Estrategia de denegar por defecto. Todo lo que no se permite explícitamente se deniega.
 - Mayor seguridad.
 - Más complejo de configurar para que todos los servicios necesarios funcionen.

Best practice: Usar **Whitelisting**, ya que por defecto es la más segura.

Seguridad: Estados del Firewall

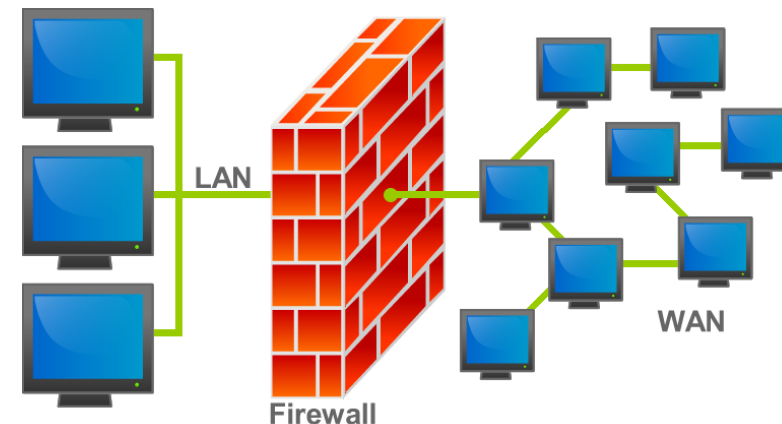
Un firewall puede operar en dos modos diferentes: **stateless (sin estado)** o **stateful (con estado)**. Se trata de un concepto similar a los utilizados en el **transporte TCP/ UDP** y en los **webservers**.

Stateless firewalls

Un firewall **stateless** (sin estado) decide si permitir o bloquear tráfico **independientemente de conexiones previamente establecidas** tales como **conexiones TCP**. El firewall no almacena información de tráfico previamente procesado. La única información que se utiliza para tomar la decisión es la **información de las cabeceras** e información específica de los paquetes entrantes como puede ser **interfaces, direcciones IP, protocolos de nivel 4 o puertos**.

Stateful firewalls

Por otro lado, un firewall **stateful** (con estado) rastrea el **estado de las conexiones establecidas**. Éste comprueba si **un paquete pertenece a una conexión existente o no**.

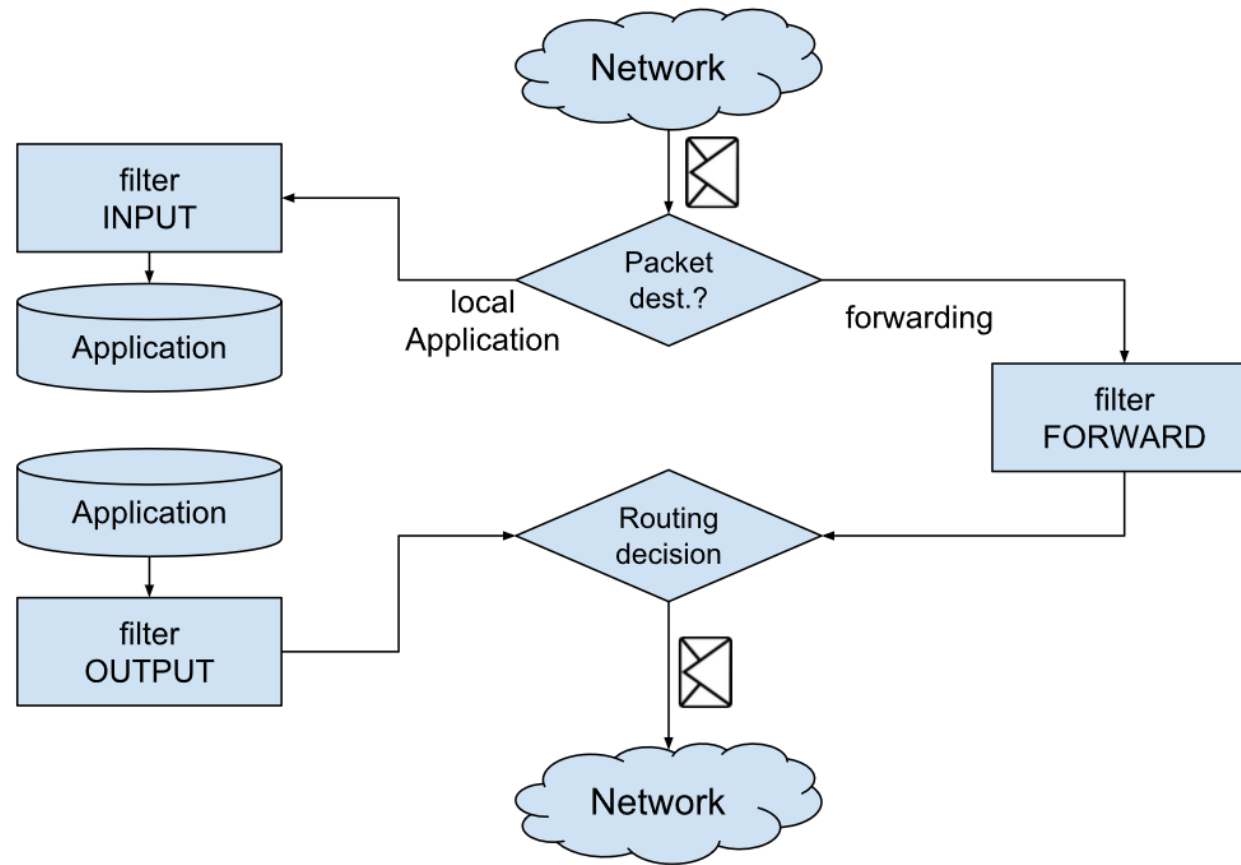


Seguridad: Introducción a las IPTABLES

- Para definir firewalls, nos basaremos en **iptables**, un programa que implementa firewalls en Linux. Existen varias opciones: **ip6tables** es sólo para tráfico IPv6; Los firewalls IPv4 se implementan con **iptables**, ambas opciones tienen la misma sintaxis. Esta aplicación también puede realizar filtrados p.e. a nivel 1,2 o ciertas propiedades de los paquetes a nivel 4.
- Iptables **analiza todos los paquetes entrantes y salientes** por un determinado interfaz. Dependiendo de cómo se configure iptables, los paquetes pueden ser **modificados** (modified), **reenviados** (forwarded) o **descartados** (dropped).
- Como se ha visto previamente, la forma en que un firewall trata un paquete se especifica mediante un conjunto de instrucciones llamadas **reglas**. Una regla especifica un **criterio** (p.e. IP origen/destino o puerto) y aplica ciertas **acciones** a todos los paquetes que las cumplan. En iptables, estas instrucciones se organizan de forma jerárquica en tres capas: **rules** (reglas), **chains** (cadenas), y **tables** (tablas). Las rules se agrupan en chains. Las chains se utilizan para procesar ciertos flujos de paquetes. Las chains son parte de tables que cubren ciertos flujos funcionales.



Cuando un paquete llega al firewall, éste **pasa a través de las diferentes tables y chains, dependiendo del destino del paquete**. Pasar a través significa que todas las reglas de esa chain se han comprobado y el paquete las cumple.

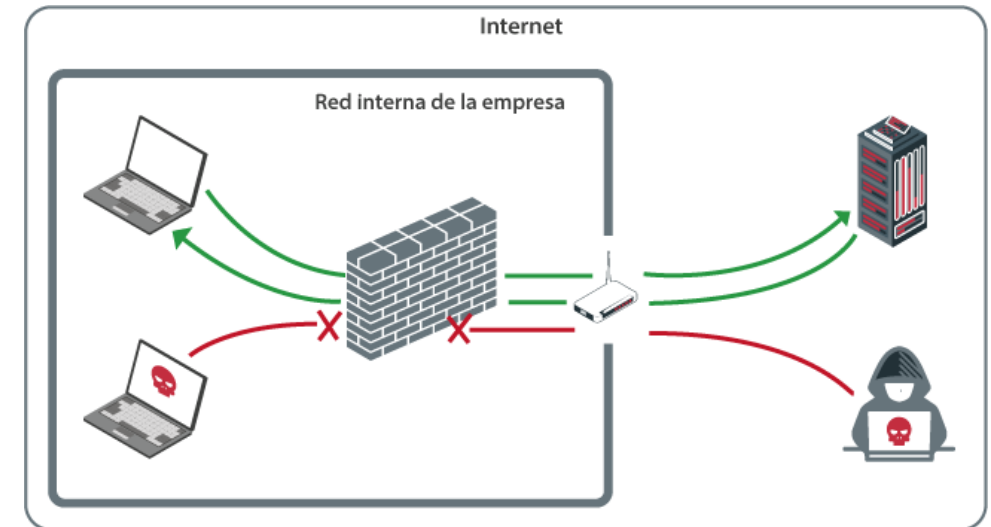


Las **Rules** son simples **directivas if-then** que especifican el patrón de coincidencia (condition) y que hacer (target) con un paquete que cumpla dicho criterio. Una regla consiste en una o más **condiciones** y un **target** (objetivo). Si las características del paquete monitorizado **coinciden con las condiciones**, la **acción objetivo (target)** se ejecuta. Por ejemplo:

“if this packet has destination **port 80**, then discard it”

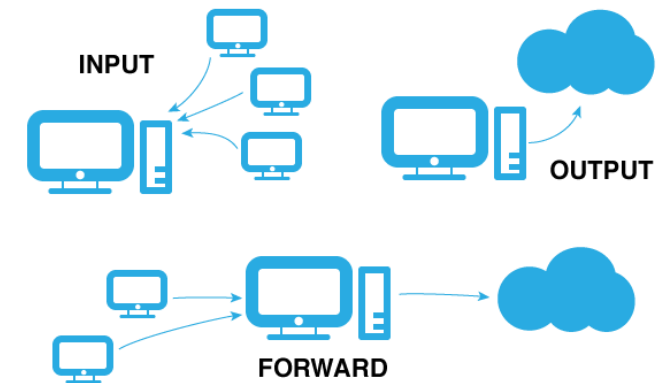
O

“if this packet arrives on **interface eth2** and has the source **IP address 192.168.1.10** and should be forwarded to the destination **IP address 192.168.1.1**, then accept it”.



Seguridad: IPTABLES, Chains

- En iptables las reglas se organizan en **chains**. Los paquetes son “matched” por las reglas **en el orden que se enumeran** en la chain. Esto significa que un paquete se compara primero con la primera regla de una cadena. Si la primera regla no coincide, entonces se compara con la segunda, y así sucesivamente. En la mayoría de los casos, el target de la regla realiza una acción que hace que la cadena no continúe analizándose, p.e. porque el paquete es aceptado por el firewall.
- Algunas cadenas predefinidas son:
 - **INPUT**: las reglas en esta cadena son aplicadas a todos los paquetes que van destino hacia el host donde se ejecuta el firewall.
 - **OUTPUT**: aplica a todos los paquetes originado por el firewall.
 - **FORWARD**: aplica a todos los paquetes que deben ser reenrutados por el host que contiene el firewall (p.e. paquetes que llegan desde Internet destino a la intranet)
 - **PREROUTING**: aplica a los paquetes antes de que se aplique la decisión de routing.
 - **POSTROUTING**: aplica a los paquetes después de que se ha hecho la decisión de routing.
- Para nosotros las cadenas INPUT, OUTPUT and FORWARD son las más relevantes.



Seguridad: IPTABLES, Tables

- Finalmente, las **tables** se usan para organizar la funcionalidad del firewall en ciertos grupos. Igual que las rules son asignadas a las chains, las chains se asignan a las tables. Iptables contiene 5 tablas diferentes.
- El siguiente cuadro resume las tables existentes y su propósito. La tercera columna indica la chains predefinidas para estas tablas. La más importante para nosotros es la table **filter** . Si tienes curiosidad investiga el resto de las tablas, para este curso las tables **nat**, **mangle**, **raw** and **security** son **opcionales**.

Table	Purpose	Predefined Chains
filter	Filter is the default table for performing packet filtering with iptables as it has no other purpose. It therefore is the default table for packets where no other table can be applied.	INPUT FORWARD OUTPUT
nat	The nat table is used for Network Address Translation (not to confuse with NAT64), which allows us to change the source or destination IP address of a packet before or after routing.	PREROUTING OUTPUT , INPUT POSTROUTING
mangle	Special handling of packets	PREROUTING OUTPUT FORWARD INPUT POSTROUTING
raw	Exemptions from connection tracking in firewall	PREROUTING OUTPUT
security	Enabling Mandatory access control for applications (SELinux)	INPUT OUTPUT

- Después de que un paquete cumple una rule, el **target** se ejecuta por iptables. El target de una regla **especifica la acción que se ejecuta cuando un paquete cumple la condición de la regla**. Se pueden especificar diferentes tipos de targets, los más comunes son:
 - **DROP**: Este target simplemente descarta el paquete y para el procesamiento dentro de la chain.
 - **REJECT**: Este target hace lo mismo que el DROP, pero envía cierta información al origen, normalmente mediante ICMP.
 - **ACCEPT**: Si el paquete es aceptado, éste puede pasar a través del filtro.

- En general, la sintaxis de los comandos de iptables tienen la siguiente estructura:

iptables [-t table] -chain -rule-specification

- Los corchetes en **[-t table]** significa que esta parte es **opcional** del comando y se puede omitir. Si no se especifica una table, la **table filter se usa por defecto**.
- La parte **-chain** especifica a que chain de la tabla dada se refiere el comando.
- Finalmente, **-rule-specification** define la rule concreta que se debe comprobar y que target aplicar si la condición se cumple.
- Si quieres tener información precisa del comando, es buena idea consultar el **manpage** de iptables. Recuerda que se puede acceder al man-page usando el siguiente comando en la consola:

man iptables

short	long	meaning
-A chain	--append chain	Append the following rule to the end of the selected chain.
-I chain [rulenum]	--insert chain [rulenum]	Insert the following rule at a specific position in the chain and not necessarily at the end as with append. Without rulenum the rule is added as first entry of the chain.
-D chain	--delete chain	Delete a rule from the selected chain. Followed either by the number of the rule to delete or its specification.
-L [chain]	--list [chain]	Lists all rules in selected chain. Outputs all chains (of specified table) if no specific chain is given.
-S [chain]	--list-rules [chain]	Similar to -L but a bit more compact.
-F [chain]	--flush [chain]	Flush the selected chain -> delete all rules in it. Flush all if no chain is given
...

Seguridad: IPTABLES, Sintaxis (rules)

- Una rule consiste en **cumplir unos criterios** sobre las propiedades de un paquete y un **target** que se ejecuta si todos los criterios se cumplen. El siguiente cuadro muestra algunos de los criterios más importantes que pueden usarse con iptables. Se puede usar tanto la versión corta (short) como la larga (long).

short	long	example	Description
-i	--in-interface	-i eth0	Packet arrived at specified interface .
-o	--out-interface	-o eth0	Packet leaving at specified interface .
-s	--source	-s 2008:db8:11::1	Packet is from specified IP address or subnet .
-d	--destination	-d 2008:db8:11::/64	Packet is for a specified IP address or subnet .
-p	--protocol	-p tcp	Packet has specified L4 protocol .
--sport	--source-port	-p tcp --sport 1024	Packet has specified source port .
--dport	--destination-port	-p udp --dport 80	Packet has specified destination port .
-m conntrack	--match conntrack	-m conntrack --cstate NEW	Packet belongs to a new connection-

- Importante:** Los puertos (ports) pueden usarse sólo en combinación con el flag **-p protocol**, de lo contrario, el firewall no sabría para qué protocolos se deben filtrar los puertos. Además, para -s o -d, puede ser una **dirección IP concreta o una subnet**, esto facilita la definición de reglas para subnets completas.

Seguridad: IPTABLES, Sintaxis (Rules)

Reglas **Stateful** (con estado):

- Para crear rules stateful iptables ofrece el módulo **conntrack**. Añadiendo “**-m conntrack**” a la rule habilita el seguimiento de conexiones para esa regla, con “**--cstate STATE**” podemos especificar con qué estado debe coincidir la regla.
- Posibles valores para STATE son:
 - **NEW**: una regla con “**--cstate NEW**” este criterio sólo hace “matching” de los paquetes que establecen una nueva conexión (p.e. cuando se inicia una sesión SSH).
 - **ESTABLISHED**: una regla con “**--cstate ESTABLISHED**” este criterio hace “matching” sobre todos los paquetes que pertenecen a una conexión ya establecida (p.e. cuando el primer paquete fue aceptado por una regla NEW).
- El firewall automáticamente hace el seguimiento de la conexión y puede interpretar si un paquete pertenece a una nueva conexión o a una conexión existente (p.e. revisando los flags SYN/ACK en los paquetes TCP).

Y finalmente, como hemos visto anteriormente, posibles **targets** para las rules que filtran tráfico son:

- **DROP**: descartan el paquete y paran el procesamiento del paquete a través de la chain.
- **REJECT**: igual que el DROP, pero también envía feedback al origen, normalmente usando ICMP.
- **ACCEPT**: si el paquete es aceptado, puede pasar a través del filtro.

Specifying a default policy:

Instead of specifying a certain rule that implements all packets not matched by any other rule in a chain, we can also define default policies for chains in the following way:

- **iptables [-t table] -P chain action**
 - E.g.: **iptables -t filter -P FORWARD DROP**

Whenever no rule in a chain matches a packet the default policy of that chain is used, in the example above DROP. Note that we could have omitted the -t filter part, as filter is the default table.

Adding new rules:

- **iptables [-t table] -A chain -rule-specification**
 - This command adds a rule to the specified chain.
 - e.g.: **iptables -A INPUT -s 192.168.1.15 -j DROP**
 - Drops all packets from 192.168.1.15 that are processed in the INPUT chain (of the filter table, which is the default).
 - e.g.: **iptables -A FORWARD -d 193.167.3.10 -m conntrack --cstate NEW -j ACCEPT**
 - Accepts all new packets that should be forwarded to 193.167.3.10. Follow up packets or answer packets can be handled by an ESTABLISHED rule (see next) .
- **iptables [-t table] -I chain [rulenum] -rule-specification**
 - Inserts the rule at the specified number in the chain. If no number is given, the default value is 1, which is at the beginning of the chain.
 - e.g.: **iptables -I INPUT 2 -p tcp --dport 80 -j DROP**
 - Inserts a new rule as second in the INPUT chain that drops all traffic to TCP port 80 (HTTP).
 - e.g.: **iptables -I INPUT -m conntrack --cstate ESTABLISHED -j ACCEPT**
 - Adds a new rule on top of the INPUT chain that accepts all established traffic.

Listing the existing rules:


There exist multiple ways to list the existing rules of a table:

- **iptables [-t table] -L**
 - This command gives a detailed overview of the policies that are in place for all chains of the specified table.
 - E.g.: **iptables -L**
 - Prints an overview of all chains of the filter table (default).
 - E.g.: **iptables -t nat -L**
 - Prints an overview of all chains of the nat table.
- **iptables [-t table] -S**
 - This command does the same as -L but in a more concise fashion.
 - e.g. : **iptables -S**


Deleting rules:

- **iptables [-t table] -D chain rule/rulenum**
 - This command deletes a specific rule in a chain.
 - e.g.: **iptables -D INPUT 3**
 - Delete the third rule in the INPUT chain of the filter table (default).
 - e.g.: **iptables -D INPUT -s 192.168.3.5 -j DROP**
 - Deletes the rule that drops all INPUT traffic from 192.168.3.5.
- **iptables [-t table] -F chain**
 - If one wants to delete all rules in a chain, this command does the same as deleting all rules one by one from a chain.
 - e.g.: **iptables -F OUTPUT**
 - Deletes all rules in the OUTPUT chain of the filter table.



 Calle Playa de Liencres, 2 bis
(entrada por calle Rozabella)
Parque Europa Empresarial
Edificio Madrid
28290 Las Rozas, Madrid

 900 373 379  info@u-tad.com

 [SOLICITA MÁS INFORMACIÓN](#)



CENTRO ADSCRITO A:



PROYECTO COFINANCIADO POR:

