Resumen Teórica 10 : Firewall

Tomás F. Melli

September 2025

Índice

1 Introducción

1.1 ¿Qué es un Firewall?

El término *firewall* proviene de la analogía con la "pared cortafuego" utilizada en arquitectura y construcción. En un edificio, una pared cortafuego separa ambientes con el fin de contener un posible incendio, evitando que se propague hacia otras áreas.

En informática y redes, un firewall cumple una función equivalente: es un **sistema de separación y control** que regula el tráfico de información entre diferentes redes, generalmente entre la red interna (privada) y la red externa (Internet).

Su misión principal es **permitir, bloquear o filtrar comunicaciones** en función de un conjunto de reglas definidas previamente. Así, se transforma en una herramienta clave de seguridad:

- Si ocurre un ataque o emergencia, el firewall puede **contener la propagación**.
- Brinda un punto centralizado para **monitorear y controlar el tráfico**.
- Protege recursos internos frente a accesos no autorizados.

En resumen, un firewall es un **mecanismo de defensa perimetral**, que actúa como la primera barrera entre lo "confiable" y lo "no confiable".

1.2 Historia de los Firewalls

La evolución de los firewalls acompañó de cerca el crecimiento de Internet y la necesidad de proteger sistemas conectados en red:

- **Finales de los 80**: los primeros mecanismos eran simples routers que separaban redes. Su función era principalmente de enrutamiento, pero con una primera capa de aislamiento.
- **Principios de los 90**: aparecieron las **Listas de Control de Acceso (ACLs)** en routers, que permitían definir reglas básicas sobre qué tráfico aceptar o rechazar.
- **Bastion Hosts**: servidores especialmente reforzados, colocados en la "frontera" de la red, configurados para resistir ataques y servir como punto de entrada controlado.
- **13 de junio de 1991**: se registra la primera "venta comercial" de un firewall como producto específico.
- **1991-1992 – TCP Wrapper**: herramienta muy popular en entornos UNIX, que permitía controlar accesos a servicios de red a nivel de host.
- **1993 – FWTK (Firewall Toolkit)**: conjunto de herramientas de código abierto para construir firewalls más completos.
- **Stateful Inspection**: evolución clave que permitió a los firewalls no solo filtrar paquetes aislados, sino también **recordar el estado de las conexiones** y tomar decisiones más inteligentes.
- **1994**: se incorporan **interfaces amigables**, que facilitan la configuración y uso del firewall por administradores sin necesidad de bajo nivel.
- **Finales de los 2000 – UTM y NGFW**: surgen los *Unified Threat Management* y *Next Generation Firewalls*. Estos integran múltiples funciones de seguridad (IDS/IPS, antivirus, filtrado de contenidos, control de aplicaciones, VPN, etc.) en una misma plataforma.
- **Actualidad – Cloud y Kubernetes**: en entornos de contenedores y microservicios, el concepto de firewall evoluciona hacia políticas definidas a nivel de red en plataformas como Kubernetes, a través de **NetworkPolicies**, que permiten controlar cómo los pods se comunican entre sí y con el exterior.

2 Tipos de Firewalls

2.1 Filtrado de paquetes estático

El **filtrado de paquetes estático** es una de las técnicas más antiguas y sencillas utilizadas en firewalls. Su funcionamiento consiste en verificar cada paquete que entra o sale de la red y **permitirlo o denegarlo** según un conjunto de reglas predefinidas por el administrador.

Estas reglas se basan en criterios simples como:

- Dirección IP de origen y destino.

- Protocolo utilizado (por ejemplo, TCP o UDP).
- Puertos de origen y destino.

Debido a su simplicidad, este tipo de filtros suele implementarse directamente en los **routers**, actuando como la primera línea de defensa.

Ventajas

- **Eficiencia**: el procesamiento de las reglas es rápido, ya que solo se inspecciona la cabecera del paquete.
- **Fácil implementación**: configurar reglas básicas es relativamente sencillo para un administrador de red.

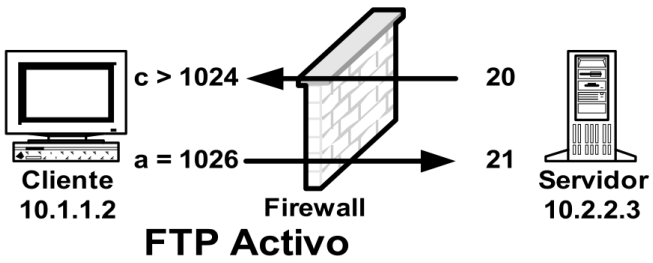
Desventajas

- **Complejidad de reglas**: cuando se requiere controlar múltiples protocolos o servicios, las reglas se vuelven numerosas y difíciles de mantener.
- **Limitaciones funcionales**: no inspecciona el contenido del paquete, ni mantiene el estado de las conexiones (a diferencia del *stateful inspection*).

2.1.1 Ejemplo: FTP y filtrado de paquetes estático

El caso del protocolo **FTP (File Transfer Protocol)** es ilustrativo porque utiliza puertos diferentes según el modo de operación (activo o pasivo). Esto obliga a definir reglas específicas en el firewall para permitir el correcto funcionamiento de la conexión.

2.1.2 FTP Activo



- El cliente inicia la conexión hacia el puerto **21** del servidor (control).
- El servidor abre la conexión de datos desde su puerto **20** hacia un puerto mayor a **1024** del cliente.

Cliente: 10.1.1.2
Servidor: 10.2.2.3

Reglas en el servidor

```
Permit any 1024:65535 to 10.2.2.3 21
Permit 10.2.2.3 21 to any 1024:65535
Permit 10.2.2.3 20 to any 1024:65535
Permit any 1024:65535 to 10.2.2.3 20
```

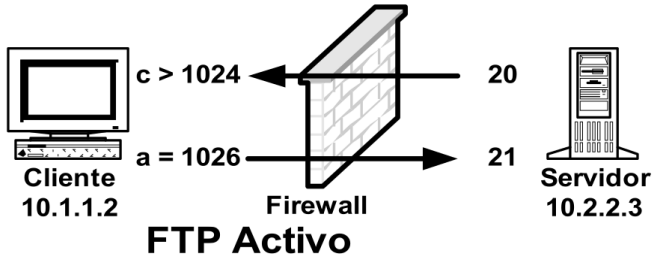
- **Permit any 1024:65535 to 10.2.2.3 21**: Permite que cualquier cliente, desde un puerto mayor a 1024, pueda iniciar una conexión hacia el puerto 21 del servidor FTP (puerto estándar de control).
- **Permit 10.2.2.3 21 to any 1024:65535**: Permite que el servidor, desde su puerto 21, responda hacia cualquier cliente utilizando un puerto mayor a 1024. Esto asegura la comunicación de control en ambas direcciones.
- **Permit 10.2.2.3 20 to any 1024:65535**: Habilita al servidor para enviar datos desde el puerto 20 (puerto de datos en FTP activo) hacia los puertos dinámicos de los clientes.
- **Permit any 1024:65535 to 10.2.2.3 20**: Permite que los clientes inicien conexiones hacia el puerto 20 del servidor para transferencias de datos.

Reglas en el cliente

```
Permit 10.1.1.0/24 1024:65535 to any 21
Permit any 21 to 10.1.1.0/24 1024:65535
Permit any 20 to 10.1.1.0/24 1024:65535
Permit 10.1.1.0/24 1024:65535 to any 20
```

- **Permit 10.1.1.0/24 1024:65535 to any 21**: Los clientes de la red interna (subred 10.1.1.0/24) pueden conectarse desde un puerto mayor a 1024 hacia cualquier servidor en el puerto 21 (control FTP).
- **Permit any 21 to 10.1.1.0/24 1024:65535**: Permite que los servidores FTP respondan desde su puerto 21 hacia los clientes internos en puertos mayores a 1024.
- **Permit any 20 to 10.1.1.0/24 1024:65535**: Permite que los servidores FTP envíen datos desde su puerto 20 hacia los clientes internos en puertos dinámicos.
- **Permit 10.1.1.0/24 1024:65535 to any 20**: Permite a los clientes iniciar conexiones desde puertos dinámicos hacia el puerto 20 del servidor, necesario para el modo activo de FTP.

2.1.3 FTP Pasivo



- El cliente se conecta al puerto **21** del servidor (control).
- El servidor abre un puerto dinámico mayor a **1024** y el cliente establece la conexión de datos hacia ese puerto.

Cliente: 10.1.1.2
Servidor: 10.2.2.3

Reglas en el servidor

```
Permit any 1024:65535 to 10.2.2.3 21
Permit 10.2.2.3 21 to any 1024:65535
Permit any 1024:65535 to 10.2.2.3 1024:65535
Permit 10.2.2.3 1024:65535 to any 1024:65535
```

- **Permit any 1024:65535 to 10.2.2.3 21**: Permite que cualquier cliente (cualquier dirección IP), desde un puerto dinámico mayor a 1024, pueda conectarse al servidor FTP (`10.2.2.3`) en el puerto 21 (puerto de control de FTP).
- **Permit 10.2.2.3 21 to any 1024:65535**: Permite que el servidor, desde su puerto 21, envíe respuestas hacia cualquier cliente en puertos mayores a 1024. Esto asegura la comunicación de control en sentido servidor → cliente.
- **Permit any 1024:65535 to 10.2.2.3 1024:65535**: Permite que los clientes abran conexiones hacia el servidor en un rango dinámico de puertos (mayores a 1024), lo cual es característico del modo FTP pasivo.
- **Permit 10.2.2.3 1024:65535 to any 1024:65535**: Permite que el servidor, utilizando puertos dinámicos, envíe datos hacia clientes que también están en puertos dinámicos. Esta regla es necesaria para completar las transferencias de archivos en FTP pasivo.

Reglas en el cliente

```
Permit 10.1.1.0/24 1024:65535 to any 21
Permit any 21 to 10.1.1.0/24 1024:65535
Permit 10.1.1.0/24 1024:65535 to any 1024:65535
Permit any 1024:65535 to 10.1.1.0/24 1024:65535
```

- **Permit 10.1.1.0/24 1024:65535 to any 21**: Permite que los clientes de la subred interna `10.1.1.0/24`, usando puertos mayores a 1024, inicien conexiones hacia cualquier servidor en el puerto 21.
- **Permit any 21 to 10.1.1.0/24 1024:65535**: Permite que los servidores FTP (desde su puerto 21) respondan hacia los clientes internos en puertos mayores a 1024.
- **Permit 10.1.1.0/24 1024:65535 to any 1024:65535**: Permite que los clientes internos establezcan conexiones hacia servidores en puertos dinámicos (mayores a 1024), como ocurre en FTP pasivo.
- **Permit any 1024:65535 to 10.1.1.0/24 1024:65535**: Permite que los servidores, desde puertos dinámicos, envíen tráfico hacia los clientes internos en puertos dinámicos. Es la regla que habilita el flujo de datos en modo pasivo.

2.2 Stateful Packet Inspection (SPI)

El **Stateful Packet Inspection** (SPI), también conocido como filtrado de paquetes con estado, es una evolución del filtrado de paquetes tradicional. Mientras que el filtrado de paquetes estático analiza cada paquete de manera independiente, un firewall **stateful** tiene en cuenta el *estado de la conexión* para tomar decisiones de filtrado.

- **Stateful** significa que el firewall mantiene información sobre las sesiones de red, desde el comienzo hasta el final de la conexión.
- El firewall puede registrar los números de secuencia TCP, el estado de la conexión (SYN, ESTABLISHED, FIN, etc.) y otras métricas necesarias para controlar dinámicamente el tráfico.
- El control se realiza sobre *la sesión completa* y no sobre paquetes individuales.
- Además, puede analizar parcialmente algunos protocolos de nivel superior para mejorar la seguridad.

Ventajas

- Mayor precisión en el filtrado, ya que se puede permitir o denegar tráfico según el estado de la sesión.
- Facilita la escritura de reglas, reduciendo la cantidad de reglas necesarias en comparación con filtrado estático.
- Permite detectar y bloquear intentos de conexión no válidos o inesperados.
- Mejora la seguridad de protocolos complejos que requieren múltiples puertos o conexiones dinámicas, como FTP pasivo.

Desventajas

- Mayor procesamiento en el firewall, debido a la necesidad de mantener tablas de estado y controlar los números de secuencia.
- Puede aumentar el consumo de memoria en sistemas con muchas conexiones simultáneas.

2.2.1 Ejemplos de reglas

Regla para aceptar conexiones preestablecidas

```
Permit any to any established,related
```

Esta regla indica que se permite el tráfico únicamente si forma parte de una conexión ya establecida o está relacionado con otra sesión existente. El firewall verifica el estado de la sesión antes de decidir si se permite el paquete.

Reglas para FTP pasivo

Cliente:

```
Permit 10.1.1.0/24 1024:65535 to any 21
```

Permite que los clientes de la subred interna `10.1.1.0/24` inicien conexiones desde puertos efímeros hacia el puerto 21 de cualquier servidor FTP (control).

Servidor:

```
Permit any 1024:65535 to 10.2.2.3 21
```

Permite que cualquier cliente establezca conexiones hacia el puerto 21 del servidor `10.2.2.3`, y el firewall controlará dinámicamente las sesiones de datos relacionadas con FTP pasivo.

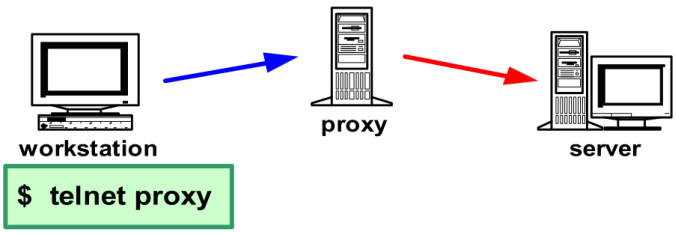
2.3 Gateways de Circuito

Los **gateways de circuito** son un tipo de proxy que actúa como intermediario entre un cliente y un servidor. A diferencia de otros proxies más avanzados, **no son inteligentes**, es decir, no inspeccionan el contenido de los paquetes ni aplican reglas complejas basadas en el protocolo. Su función principal es simplemente **generar una nueva conexión** en nombre del cliente.

2.3.1 Características principales

- **No inteligentes:** Se limitan a establecer un nuevo circuito de comunicación; no analizan los datos que pasan por ellos.
- **Generan nuevas conexiones:** Cuando un cliente desea comunicarse con un servidor a través del gateway, éste abre una conexión nueva hacia el destino, actuando como puente.
- **El cliente debe conocer el gateway:** Para que funcione, el cliente necesita estar configurado para usar el gateway de circuito.
- **Independientes del protocolo:** No dependen del tipo de protocolo de aplicación; pueden transportar TCP, UDP u otros protocolos que requieran establecer una sesión.
- **Usados junto con políticas estrictas de filtrado:** Debido a su simplicidad, suelen combinarse con firewalls y políticas de filtrado más estrictas para garantizar seguridad adicional.

2.3.2 Ejemplo práctico



El ejemplo más conocido de gateway de circuito es **SOCKS** (SOCKet Secure es un protocolo de red que funciona como un proxy de nivel de sesión, permitiendo que un cliente se conecte a un servidor a través de un intermediario (proxy), sin necesidad de que el cliente y el servidor conozcan directamente las direcciones IP del otro).

- El **cliente** (workstation) inicia la conexión hacia el proxy, por ejemplo usando `telnet` o un cliente SOCKS.
- El **proxy** recibe la solicitud y genera una **nueva conexión** hacia el **servidor** final.
- Todos los paquetes que el cliente envía son retransmitidos a través del proxy, pero este no inspecciona ni modifica los datos.

2.4 Gateways de Aplicación (Proxy)

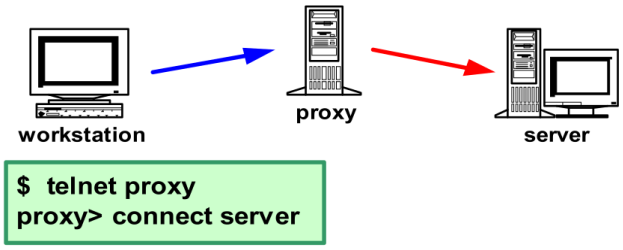
Los **gateways de aplicación**, también conocidos simplemente como **proxies**, son un tipo de intermediario de red que opera a nivel de **protocolo de aplicación**. A diferencia de los *gateways de circuito*, que solo generan una nueva conexión, los proxies **entienden y manejan el protocolo específico** que están filtrando o retransmitiendo.

Esto les permite ofrecer **mayor control, seguridad y funcionalidad** sobre el tráfico que pasa a través de ellos.

2.4.1 Características principales

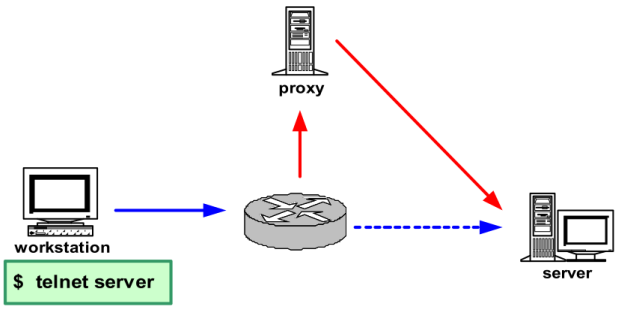
- **Conocidos como "Proxy":** Actúan como intermediarios entre el cliente y el servidor, procesando la solicitud del cliente antes de reenviarla.
- **Interpretación del protocolo:** A diferencia de los gateways de circuito, los proxies analizan el contenido y la estructura de los mensajes de aplicación. Por ejemplo, un proxy HTTP puede inspeccionar URLs, encabezados y métodos HTTP.
- **Configuración del cliente:** Generalmente, el cliente debe conocer el proxy y el protocolo debe soportar su uso (por ejemplo, configurar un navegador web para usar un proxy HTTP).
- **Autenticación y control de uso:** Permiten implementar autenticación de usuarios y políticas de acceso más complejas, controlando quién puede usar qué servicios y cuándo.
- **Auditoría y registro:** Facilitan la generación de logs detallados sobre las solicitudes, útil para auditorías de seguridad, monitoreo y resolución de problemas.
- **Funciones adicionales:** Pueden incluir almacenamiento en **cache**, filtrado de contenido, compresión de datos o inspección de malware a nivel de aplicación.

2.4.2 Proxy tradicional



- La **workstation** inicia la conexión hacia el **proxy**, usando el cliente adecuado (por ejemplo, `telnet`).
- El **proxy** recibe la solicitud y establece la conexión con el **servidor** final.
- Todo el tráfico de la sesión pasa a través del proxy, que puede inspeccionarlo y aplicar reglas específicas del protocolo.

2.4.3 Proxy transparente



- En el caso de un **proxy transparente**, el cliente no necesita estar configurado para usar el proxy; el gateway intercepta automáticamente las conexiones hacia el servidor.
- Esto permite aplicar control de acceso, monitoreo y caching sin requerir configuración adicional en los clientes.

2.4.4 Ventajas de los Gateways de Aplicación

- Mayor control sobre el tráfico y el uso de los servicios.
- Capacidad de aplicar autenticación y políticas de acceso detalladas.
- Generación de registros de auditoría completos y fáciles de analizar.
- Posibilidad de agregar funcionalidades adicionales (cache, filtrado de contenido, inspección de malware).

3 Ingress / Egress Filtering

Las técnicas de **Ingress** y **Egress Filtering** se utilizan para **controlar el flujo de paquetes IP** que entran y salen de una red, principalmente con el objetivo de **prevenir el IP spoofing**, es decir, la suplantación de direcciones IP. Estas técnicas suelen implementarse directamente en **firewalls y routers**, formando parte de la política de seguridad de la red.

3.1 Ingress Filtering

- La técnica de **Ingress Filtering** se aplica al **tráfico entrante**.
- Su objetivo es **evitar que paquetes con IPs de origen falsificadas lleguen a la red interna**.
- Por ejemplo, si mi red interna utiliza el rango `192.168.1.0/24`, un paquete que ingresa desde Internet con dirección de origen `192.168.1.5` será bloqueado, ya que no tiene sentido que un host externo envíe tráfico desde una IP interna.
- Esta técnica ayuda a **proteger la red contra ataques de suplantación de identidad y ciertas formas de DDoS**.

3.2 Egress Filtering

- La técnica de **Egress Filtering** se aplica al **tráfico saliente**.
- Su objetivo es **asegurar que los paquetes que salen de mi red no tengan IPs falsificadas**.
- Por ejemplo, un paquete que intenta salir hacia Internet con una IP de origen que no pertenece a mi red será bloqueado.
- Esto evita que los atacantes dentro de la red utilicen direcciones IP falsas para ocultar su identidad o lanzar ataques hacia el exterior.

4 NAT (Network Address Translation)

Network Address Translation (NAT) es una técnica que permite **modificar las direcciones IP** en los paquetes que atraviesan un router o firewall, normalmente para **ocultar la topología interna de una red**, permitir el uso de direcciones privadas y facilitar la seguridad de la información.

4.1 Historia y estandarización

- NAT fue definido formalmente en el RFC 1631 (1994).
- Surge ante la **falta de direcciones IP públicas** y la necesidad de preservar la privacidad de la red interna.
- Permite que dispositivos con IP privadas definidas por RFC 1918 puedan comunicarse con redes externas mediante traducción de direcciones.
- Rangos privados definidos por RFC 1918:
  - **10.0.0.0 – 10.255.255.255** → 16.777.216 direcciones disponibles. Muy grande, ideal para empresas o datacenters con muchos dispositivos.
  - **172.16.0.0 – 172.31.255.255** → 1.048.576 direcciones disponibles. Tamaño intermedio, adecuado para redes medianas.
  - **192.168.0.0 – 192.168.255.255** → 65.536 direcciones disponibles. Más pequeño, suficiente para hogares o pequeñas oficinas.

  La notación CIDR indica **cuántos bits de la dirección IP se usan para identificar la red**. Una dirección IPv4 tiene 32 bits en total.
  - **/24:** 24 bits para la red, 8 bits para los hosts. Ejemplo: 192.168.1.0/24 → 256 direcciones posibles, 254 utilizables para hosts.
  - **/16:** 16 bits para la red, 16 bits para los hosts. Ejemplo: 172.16.0.0/16 → 65.536 direcciones posibles, 65.534 utilizables.
  - **/8:** 8 bits para la red, 24 bits para los hosts. Ejemplo: 10.0.0.0/8 → 16.777.216 direcciones posibles, 16.777.214 utilizables.

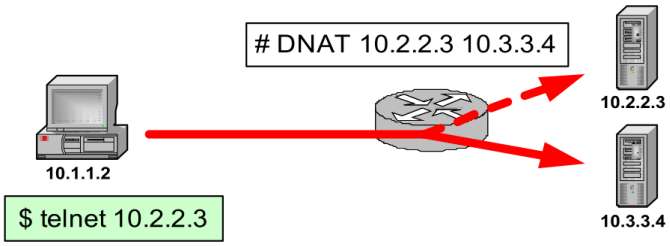
  En general: mientras menor sea el número después de la barra, más grande es la red y más hosts puede contener. /24 es común en redes domésticas, /16 en redes medianas y /8 en redes muy grandes.

4.2 Tipos de NAT

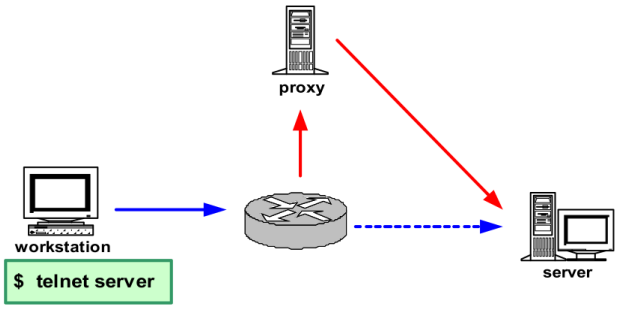
4.2.1 DNAT (Destination NAT)

Modifica la **dirección de destino** de un paquete que entra a la red. Se utiliza, por ejemplo, para redirigir tráfico hacia un servidor interno.

El cliente se conecta a `10.2.2.3`, pero el paquete es redirigido internamente a `10.3.3.4`.

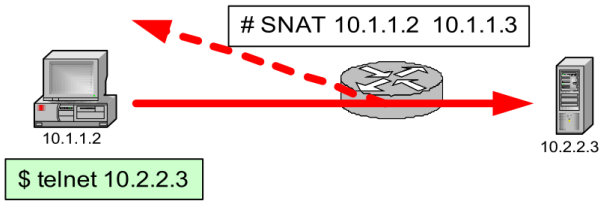


Puede combinarse con un **proxy transparente** para inspección de tráfico.



4.2.2 SNAT (Source NAT)

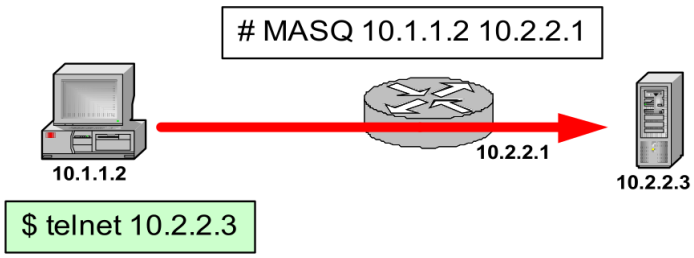
Modifica la **dirección de origen** de los paquetes que salen de la red interna.



- Un host interno con IP `10.1.1.2` envía un paquete hacia Internet; el firewall cambia la IP de origen a `10.1.1.3` antes de salir.
- Esto permite ocultar la IP interna y gestionar la conectividad hacia redes externas.

4.2.3 Masquerading

Es una **forma particular de SNAT** y la más utilizada en entornos domésticos o con direcciones dinámicas.



- La IP interna `10.1.1.2` se traduce a la IP pública o de salida `10.2.2.1`.
- Todos los paquetes de la red interna parecen provenir de esta dirección pública.
- Útil para **ocultar la topología de la red** y permitir la comunicación con Internet.

Otros tipos relacionados

- **PAT (Port Address Translation):** Traduce no solo la IP, sino también los puertos de origen, permitiendo múltiples hosts internos compartiendo una misma IP pública.
- DNAT, SNAT, PAT y Masquerading suelen considerarse variantes de la misma familia de técnicas de NAT.

4.2.4 Integración con seguridad y firewalls

- NAT se combina frecuentemente con **ACLs (Access Control Lists)** para definir qué tráfico traducido está permitido o denegado:

```
DNAT 10.1.1.1 >1024 any 80 to proxy 8080
```

  - **DNAT:** Destination NAT → se cambia la dirección de destino de los paquetes entrantes.
  - **10.1.1.1 >1024:**
    - `10.1.1.1` es la IP de destino original que el cliente intenta alcanzar.
    - `>1024` indica que se consideran puertos mayores a 1024 (por ejemplo, puertos efímeros de los clientes).
  - **any 80:**
    - `any` indica cualquier IP de origen (cualquier cliente externo).
    - `80` indica que el puerto de destino original es el 80, es decir, tráfico HTTP.
  - **to proxy 8080:**
    - Redirige el tráfico hacia el proxy interno en el puerto 8080.
    - El firewall o router cambia la dirección de destino y el puerto del paquete, enviándolo al proxy en vez del servidor original.
  - Todo el tráfico HTTP (puerto 80) que llegue a la IP `10.1.1.1` desde cualquier cliente externo será redirigido a un proxy interno en el puerto 8080.
  - El NAT se encarga de la traducción de dirección y puerto.
  - La ACL define qué tráfico está permitido, evitando que otros paquetes no deseados pasen.

- También puede combinarse con **Stateful Packet Inspection**, permitiendo NAT específico para protocolos complejos como FTP.
- De esta manera, NAT no solo facilita la conectividad, sino que **refuerza la seguridad de la red**, protegiendo direcciones internas y controlando el tráfico hacia el exterior.

5 Esquemas de redes con firewalls

Dependiendo de la arquitectura de la red, los firewalls pueden integrarse de distintas maneras. Uno de los esquemas clásicos es el **Screening Router**.

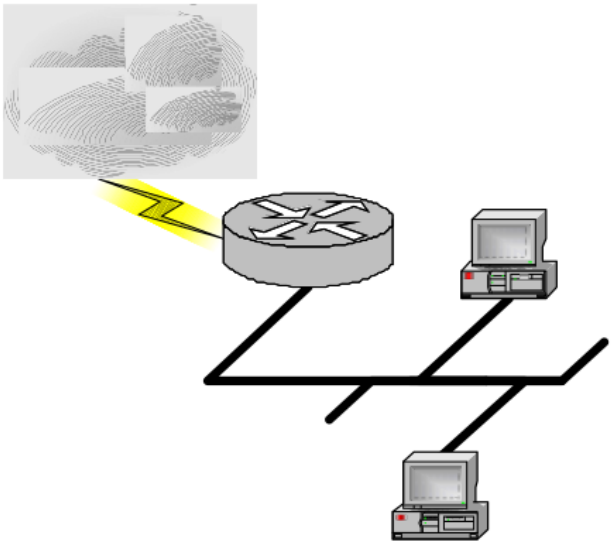
5.1 Screening Router

Un **Screening Router** es un router configurado para realizar **filtrado de paquetes**, normalmente con reglas predefinidas.

Características principales

- **Filtra paquetes:** decide si un paquete se permite o se descarta según la dirección IP, puerto o protocolo.
- **Reglas complejas:** aunque puede manejar reglas detalladas, la complejidad no se traduce en inteligencia avanzada sobre el tráfico.
- **Poca inteligencia:** no analiza el estado de la conexión ni el contenido de la aplicación; solo inspecciona encabezados de paquetes.
- **Generalmente asociado a filtrado estático:** las decisiones se toman de forma fija, sin tener en cuenta sesiones ni historial de conexiones.

Funcionamiento típico

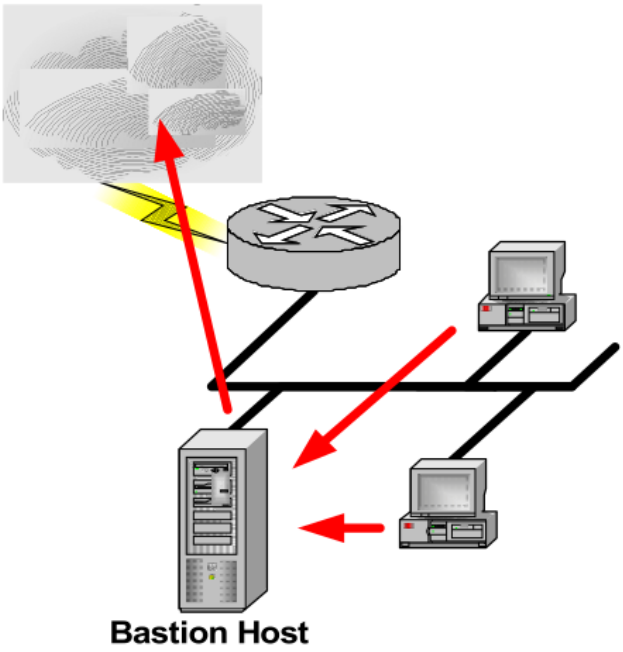


1. Se define un conjunto de **reglas ACL** que determinan qué paquetes pueden entrar o salir de la red.
2. El router analiza cada paquete de forma individual (sin estado).
3. Los paquetes que cumplen las reglas son reenviados; los que no, se descartan.

5.2 Screened Host (Bastion Host)

El esquema de **Screened Host**, también conocido como **Bastion Host (BH)**, es un enfoque de seguridad que combina **filtrado de paquetes** y un **host protegido** para ofrecer control y supervisión del tráfico que entra a la red.

Concepto principal



El **Bastion Host** es un **servidor especialmente protegido** que se coloca entre la red externa (Internet) y la red interna. Solo **los paquetes destinados al BH** son permitidos por el firewall o screening router; el resto del tráfico hacia la red interna se bloquea. Este esquema proporciona una **primera línea de defensa concentrada** en un único punto seguro.

Características principales

- **Solo permite paquetes al Bastion Host:** el tráfico que no va destinado al BH se descarta, reduciendo la superficie de ataque sobre la red interna.
- **Reglas más simples:** al concentrar el tráfico en un solo host, las reglas de filtrado son menos complejas que en un screening router tradicional. Normalmente se manejan ACL básicas para direccionar el tráfico hacia el BH.
- **El Bastion Host debe ser seguro:** es un objetivo crítico para la seguridad, por lo que se refuerza con configuraciones estrictas, parches al día y monitoreo constante. Suele contar con logging y auditoría para detectar intentos de intrusión.
- **Suele tener proxies:** se pueden instalar **proxies de aplicación** en el BH para analizar y controlar protocolos específicos (HTTP, FTP, etc.), permitiendo inspección de tráfico a nivel de aplicación y añadiendo una capa adicional de seguridad.

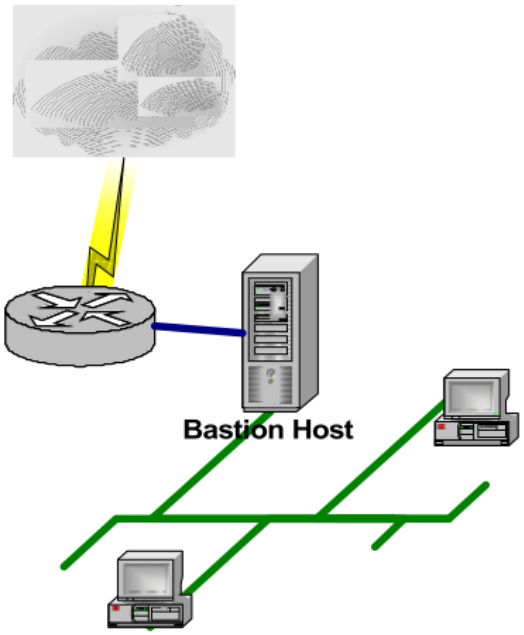
Funcionamiento típico

1. Un **screening router** filtra el tráfico externo, dejando pasar únicamente los paquetes hacia el Bastion Host.
2. El **Bastion Host** recibe los paquetes permitidos y, si es necesario, los reenvía internamente mediante proxies o NAT hacia los servicios seguros.
3. Todo el tráfico hacia la red interna pasa por el BH, asegurando control y monitoreo centralizado.

5.3 Screened Host (Dual-Homed Bastion Host)

El esquema de **Dual-Homed Bastion Host (BH de doble interfaz)** es una variante avanzada del **Screened Host**, diseñado para ofrecer **mayor seguridad y control del tráfico** entre una red interna y una externa.

Concepto principal



El **Dual-Homed Bastion Host** tiene **dos interfaces de red**: una conectada a la red externa (Internet) y otra a la red interna. No realiza **ruteo directo** de paquetes entre la red externa e interna; todo el tráfico debe pasar por **servicios específicos** como proxies. Esta separación física de redes hace al BH mucho más **robusto y seguro**, ya que los paquetes no pueden atravesar el BH directamente.

Características principales

- **Sin ruteo a través del BH:** los paquetes de la red externa no se enrutan automáticamente hacia la red interna. Solo los servicios autorizados (como proxies) pueden procesar y reenviar tráfico.
- **BH con Proxies:** los servicios como HTTP, FTP o SMTP se ejecutan en el BH mediante **proxies de aplicación**, que controlan y analizan el tráfico a nivel de protocolo. Esto permite inspección profunda y control de acceso más granular.
- **Muy robusto:** al no permitir ruteo directo y concentrar servicios en un BH protegido, se reduce la superficie de ataque. Ideal para redes donde la seguridad es crítica, como empresas financieras o gubernamentales.
- **El BH debe ser seguro:** este host es un **punto crítico de defensa**, por lo que se refuerza con configuraciones estrictas, parches constantes y monitoreo activo.
- **No se necesita NAT:** debido a que las interfaces del BH están separadas y todo el tráfico se gestiona mediante proxies, la traducción de direcciones no es necesaria para controlar la comunicación interna y externa.

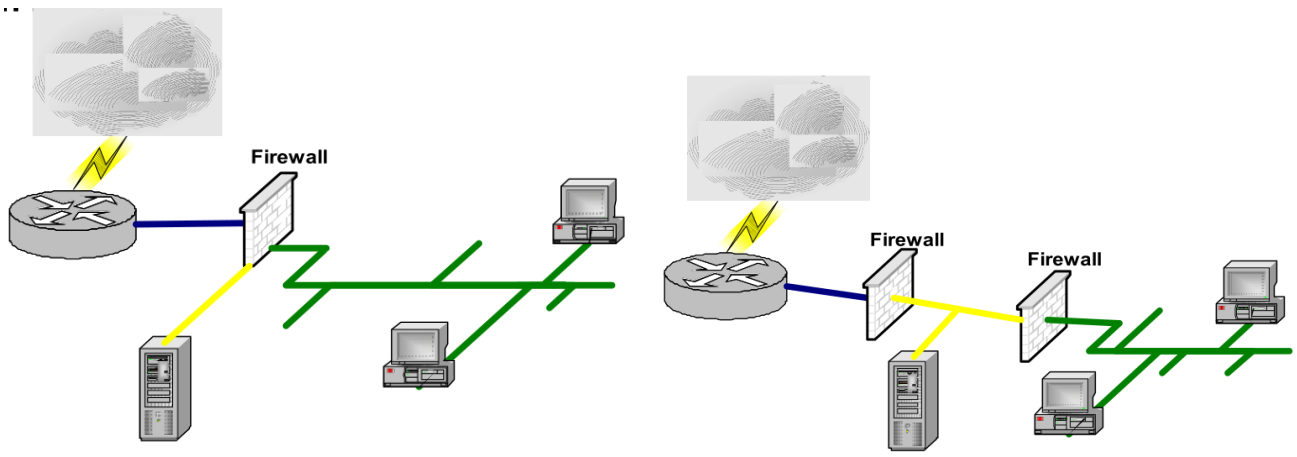
Funcionamiento típico

1. El tráfico externo llega al **Dual-Homed Bastion Host**.
2. El BH, mediante **proxies**, procesa las solicitudes y decide si permitirlas hacia la red interna.
3. Todo el tráfico hacia la red interna está estrictamente controlado, evitando exposición directa de la red interna a Internet.

5.4 Firewall con DMZ (Demilitarized Zone)

Un **Firewall con DMZ** es una arquitectura de seguridad en redes que agrega una zona intermedia entre la red interna confiable y la red externa (Internet). La DMZ permite ofrecer servicios públicos de manera segura, sin exponer la red interna directamente.

Concepto principal



La **DMZ** es una subred separada donde se colocan los servidores que deben ser accesibles desde Internet, como:

- Servidores web
- Servidores de correo
- Servidores FTP

La red interna sigue protegida detrás del firewall y no tiene acceso directo desde la DMZ ni desde Internet. Esto permite aislar los servicios públicos, reduciendo riesgos de que un ataque comprometa la red interna.

Características principales

- **Separación de redes:** se crean al menos dos zonas protegidas, la DMZ y la red interna. El tráfico entre Internet y la DMZ está controlado por el firewall, así como el tráfico entre DMZ e interna.
- **Control granular del tráfico:** se pueden definir reglas específicas para cada tipo de servicio. Por ejemplo, permitir HTTP y HTTPS hacia el servidor web, pero bloquear cualquier otro protocolo.
- **Protección de la red interna:** si un servidor en la DMZ es comprometido, la red interna sigue protegida. La DMZ funciona como capa de contención ante ataques externos.
- **Uso combinado con NAT y proxies:** la DMZ puede usar NAT para traducir direcciones públicas a privadas. También puede implementar proxies de aplicación para inspección profunda de protocolos.

6 Implementación del firewall

Cuando configuramos un firewall, debemos definir **políticas por defecto** sobre qué tráfico se permite o se deniega. Las dos opciones más comunes son:

Default Permit (Permitir por defecto)

- Todo el tráfico está permitido, salvo los protocolos o servicios que explícitamente bloqueemos.
- Ejemplo: permitir todo excepto Telnet, rlogin o FTP antiguo.
- **Riesgo:** muy inseguro. Si un servicio vulnerable queda abierto por accidente, cualquier atacante podría explotarlo. Es difícil mantener la seguridad porque siempre hay que cerrar manualmente todo lo que no queremos permitir.

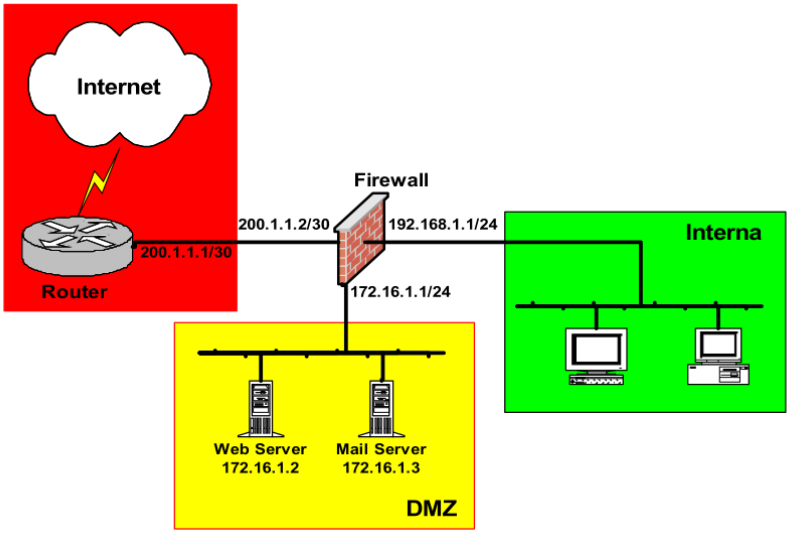
Default Deny (Denegar por defecto)

- Todo el tráfico está bloqueado, salvo los protocolos o servicios que explícitamente permitamos.
- Ejemplo: permitir solo HTTP y HTTPS hacia un servidor web, y todo lo demás queda denegado.
- **Ventaja:** mucho más seguro. Evita accesos no autorizados a servicios que no hemos configurado y protege ante vulnerabilidades en servicios no autorizados.

La implementación de un firewall efectivo requiere un enfoque estructurado que incluya el **diseño de la red**, la **definición de políticas**, la **implementación de reglas** y el **mantenimiento** continuo.

6.1 Diseño de la red

Antes de configurar un firewall, es fundamental definir la arquitectura de la red. Un ejemplo típico incluye:



- Conexión a Internet mediante una interfaz pública del firewall.
- Subred DMZ donde se alojan servidores públicos (Web, Mail, DNS).
- Red interna protegida detrás del firewall para usuarios y recursos internos.

6.2 Definición de políticas

Las políticas definen qué servicios y protocolos estarán permitidos entre las diferentes zonas:

	Internet	DMZ	Interna
Internet (eth0) 200.1.1.2/30		ws:http ms:smtp	NO
DMZ (eth1) 172.16.1.1/24	ms:dns ms:smtp		NO
Interna (eth2) 192.168.1.1/24	-:http	ms:smtp ms:dns ms:pop3 ws:http	

Interpretación de la tabla de políticas

- **Internet (eth0):** interfaz que conecta el firewall a la red externa (Internet) con IP `200.1.1.2/30`.
  - Servicios permitidos hacia la DMZ: `ms:dns` y `ms:smtp` (DNS y SMTP del Mail Server).
  - Servicios hacia la red interna: `-:http` indica que HTTP no está permitido desde Internet hacia la interna directamente.
- **DMZ (eth1):** interfaz que conecta la DMZ con IP `172.16.1.1/24`.
  - Servidores en la DMZ:

        * `ws:http` → Web Server permite tráfico HTTP desde Internet.
        * `ms:smtp` → Mail Server permite tráfico SMTP desde Internet.
    - Tráfico hacia la red interna: se definen reglas específicas; algunos servicios se permiten, otros no.

- **Interna (eth2):** interfaz de la red interna `192.168.1.1/24`.
    - Tráfico hacia la DMZ o Internet:
        * Algunos servicios están permitidos: `ms:smtp`, `ms:dns`, `ms:pop3`, `ws:http`.
        * Otros no están permitidos (`NO`), evitando acceso no autorizado.

Esto permite un **control granular del tráfico** y protege la red interna frente a ataques.

6.3 Herramientas de administración

Configurar un firewall puede ser complejo, por lo que existen herramientas que facilitan la tarea mediante interfaces gráficas o abstracciones de reglas:

- **ufw** (Uncomplicated Firewall)
- **shorewall**
- **pfsense** (https://www.pfsense.org)

Estas herramientas permiten escribir, visualizar y administrar las reglas de manera más sencilla y segura.

6.4 Mantenimiento - Logs

El mantenimiento de un firewall incluye revisar los logs para detectar intentos de acceso no autorizado o patrones sospechosos. Los logs permiten identificar qué paquetes fueron bloqueados, de dónde provienen y hacia dónde se dirigían. Esto es fundamental para ajustar reglas y mejorar la seguridad de la red.

6.4.1 Ejemplo de logs de Shorewall

```
1 Sep 26 00:11:10 rodito Shorewall:net2all:DROP: IN=ppp0 OUT= MAC= SRC=200.59.11.154 DST=200.59.77.76 LEN=48
      ...
2 Sep 26 13:40:48 rodito Shorewall:net2all:DROP: IN=ppp0 OUT= MAC= SRC=200.59.11.245 DST=200.59.77.76 LEN=48
      ...
3 Sep 26 15:46:43 rodito Shorewall:net2all:DROP: IN=ppp0 OUT= MAC= SRC=200.59.47.80 DST=200.59.77.76 LEN=48
     ...
4 Sep 26 17:38:25 rodito Shorewall:net2all:DROP: IN=ppp0 OUT= MAC= SRC=200.59.11.171 DST=200.59.77.76 LEN=48
      ...
5 Sep 26 19:33:34 rodito Shorewall:net2all:DROP: IN=ppp0 OUT= MAC= SRC=200.59.11.225 DST=200.59.77.76 LEN=48
      ...
```

Interpretación del log

Cada línea contiene información detallada sobre un paquete que fue bloqueado:

- **Fecha y hora:** momento en que ocurrió el evento (ej. “Sep 26 00:11:10”).
- **Host:** el nombre del firewall que registró el evento (ej. “rodito”).
- **Fuente y destino:** IP de origen (SRC) y destino (DST) del paquete.
- **Interfaces:** IN y OUT muestran por dónde entró y salió el paquete.
- **Protocolo y puertos:** tipo de protocolo (TCP/UDP), puerto de origen y puerto de destino.
- **Acción:** DROP indica que el paquete fue bloqueado según las reglas del firewall.

Este análisis permite:

- Identificar intentos de intrusión o escaneo de puertos.
- Verificar que las reglas se están aplicando correctamente.
- Ajustar políticas de seguridad de manera proactiva.

7 Firewall Personal

Un **firewall personal** es un software instalado en un sistema individual, generalmente en la estación de trabajo de un usuario, cuyo objetivo es **controlar la comunicación de entrada y salida** de ese equipo. A diferencia de los firewalls de red, que protegen segmentos completos de red, un firewall personal se enfoca en **proteger un único dispositivo**.

Funcionalidades principales

- **Control de aplicaciones:**
  - Permite definir qué aplicaciones pueden iniciar conexiones hacia Internet o recibir datos desde la red.
  - Por ejemplo, se puede permitir que un navegador web acceda a Internet mientras se bloquea un programa desconocido o potencialmente peligroso.
- **Filtrado de tráfico:**
  - Inspecciona los paquetes de datos entrantes y salientes.
  - Aplica reglas basadas en la aplicación, el puerto, el protocolo o la dirección IP.
- **Protección frente a amenazas locales y externas:**
  - Evita que programas maliciosos en el equipo se comuniquen con servidores externos.
  - Permite controlar el acceso de usuarios o procesos internos a recursos de red no autorizados.
- **Notificaciones y registros:**
  - Informa al usuario cuando una aplicación intenta conectarse a la red.
  - Genera logs para auditoría y análisis de seguridad.

Ejemplo en sistemas Windows

Los firewalls personales en Windows permiten crear reglas específicas por aplicación, indicando si **cada programa puede acceder a Internet** o si debe ser bloqueado. Esto proporciona un nivel de control granular sobre el tráfico de cada software, mejorando la seguridad individual sin necesidad de modificar la configuración de red global.

8 Unified Threat Management (UTM)

El concepto de **Unified Threat Management (UTM)** se refiere a soluciones integradas de seguridad de red que combinan múltiples funciones de protección en un único dispositivo o plataforma. Esto permite centralizar la gestión de la seguridad, simplificar la administración y mejorar la eficiencia operativa.

Componentes principales

Aunque no todas las soluciones UTM incluyen todos los módulos, las funcionalidades más comunes son:

- **Firewall de red con Stateful Inspection:**
  - Controla el tráfico entrante y saliente en función del estado de las conexiones.
  - Permite un filtrado más preciso que un firewall tradicional de paquetes, ya que mantiene información sobre sesiones y puede inspeccionar protocolos de nivel superior.
- **Antivirus de red:**
  - Analiza el tráfico en busca de virus y malware antes de que lleguen a los equipos finales.
  - Proporciona protección centralizada, reduciendo la necesidad de múltiples instalaciones locales.
- **Anti-spam:**
  - Filtra correos electrónicos no deseados y potencialmente peligrosos.
  - Mejora la productividad al reducir el volumen de mensajes maliciosos o irrelevantes.
- **Filtrado de contenidos:**

  - Permite controlar el acceso a sitios web y aplicaciones según políticas corporativas.
  - Puede bloquear contenido inapropiado, peligroso o no autorizado.
- **IDS/IPS (Intrusion Detection/Prevention Systems):**
  - Detecta y, en algunos casos, previene intentos de intrusión o ataques sobre la red.
  - Proporciona un análisis en tiempo real y alertas sobre comportamientos sospechosos.
- **Data Leak Prevention (DLP):**
  - Evita la fuga de información confidencial desde la red hacia el exterior.
  - Puede filtrar archivos, correos o datos sensibles según políticas predefinidas.
- **VPN (Virtual Private Network):**
  - Permite conexiones seguras desde ubicaciones remotas hacia la red corporativa.
  - Asegura la confidencialidad e integridad de los datos transmitidos a través de redes públicas.

9 Next-Generation Firewall (NGFW)

Un **Next-Generation Firewall (NGFW)** es un firewall que incorpora funcionalidades avanzadas y ofrece una inspección del tráfico más profunda que los firewalls tradicionales. Además de filtrar por puertos y protocolos, los NGFW pueden analizar el contenido y la lógica de las aplicaciones, proporcionando un control más granular sobre la seguridad de la red.

Funcionalidades principales

- **Detección de protocolo de aplicación:**
  - Identifica aplicaciones independientemente del puerto de comunicación.
  - Permite bloquear o permitir tráfico según la aplicación real, evitando que se eluda la política de seguridad mediante puertos no estándar.
- **User Role Firewalling:**
  - Permite aplicar reglas basadas en la identidad del usuario o del grupo.
  - Facilita políticas de seguridad más flexibles, adaptadas a distintos roles dentro de la organización.
- **IPS (Intrusion Prevention System):**
  - Detecta y previene ataques en tiempo real sobre la red.
  - Integra análisis de firmas y comportamientos sospechosos.
- **SSL Proxy:**
  - Inspecciona tráfico cifrado SSL/TLS para detectar amenazas ocultas.
  - Permite aplicar políticas de seguridad sobre comunicaciones cifradas sin comprometer la privacidad de los datos.
- **Manejo de redundancia y alta disponibilidad:**
  - Los NGFW están diseñados para funcionar en clústeres o modos activos/pasivos, asegurando continuidad del servicio en caso de fallas.
- **Orientación enterprise:**
  - Generalmente, estos firewalls están más enfocados a redes corporativas de gran tamaño o entornos críticos que requieren control avanzado del tráfico.
- **Relación con UTM:**
  - Aunque comercialmente se los presenta como NGFW, a nivel técnico muchas veces las funciones de un NGFW se superponen con las de un UTM.
  - En algunos casos, se podría considerar que UTM = NGFW, dependiendo de los módulos activados y el alcance de inspección.

Conclusión

Los NGFW representan una evolución de los firewalls tradicionales, ofreciendo **mayor visibilidad y control del tráfico**, incluyendo inspección de aplicaciones, control por usuario y prevención de intrusiones, siendo especialmente útiles en entornos empresariales complejos y de alta demanda de seguridad.