Tema 6. Seguridad Perimetral

Seguridad en Sistemas Informáticos

Diciembre-2019

Contenido

Conceptos básicos (I)

Cortafuegos/Firewalls: Mecanismos de control de acceso a la red y a los recursos informáticos de una organización

- Formado por componentes hardware y software
- Separa red interna (*equipos de confianza*) de los equipos externos (*potencialmente hostiles*) mediante control del tráfico
- Deniega intentos de conexión no autorizados (en ambos sentidos)
- **Finalidad:** prevención de ataques desde el exterior hacia equipos internos
  - Opcionalmente: control del uso de la red por parte de los equipos internos
  - Protección del propio equipo: *"firewalls personales"*

Principios básicos de funcionamiento

1. Todo el tráfico desde interior a exterior y viceversa debe pasar por el Cortafuegos/Firewall.
   - bloqueo de todos los accesos físicos a red propia excepto el del Cortafuegos
   - diferentes topologías $\Rightarrow$ diferentes niveles de aislamiento
2. Cortafuegos permite sólo tráfico autorizado
   - definido por las políticas de seguridad de la organización
   - cada tipo de Cortafuegos permite distintos tipos de control
3. Cortafuegos debe ser inmune a intrusiones
   - Sistema Operativo y software fiable

Conceptos básicos (II)

Limitaciones

- No protegen contra ataques que no pasen a través del Cortafuegos
  - Conexiones adicionales no previstas pueden ofrecer un punto de entrada alternativo fuera de su control
- **No** protegen **contra amenazas internas**
- Pueden proporcionar una sensación de falsa seguridad
  - Cortafuegos no basta por si sólo
  - Seguridad en redes afecta a muchos aspectos
  - En la práctica: defensa en profundidad
    - implementar diversas capas de mecanismos de defensa complementarios y coordinados
    - nunca confiar la seguridad de la red a un único mecanismo (cortafuegos)

Clasificación general (I)

Filtros de paquetes

- Inspeccionan los paquetes recibidos/enviados y comprueban si encajan en las reglas
- Filtrado basado en la información contenida en cada paquete
  - cada paquete se inspecciona de forma aislada y la decisión se toma de forma independiente
  - filtro ”sin estado” (no tiene en cuenta si paquetes son parte de una conexión)
- Uso de puertos estándar para bloquear servicios concretos

Filtros ”con estado”

- Llevan registro de las conexiones que pasan a través del Cortafuegos
- Estudian y reconocen paquetes { de inicio/fin de conexión; parte de conexiones abiertas

Filtros a nivel de aplicación (Proxies)

- Cortafuegos basados en el uso de Proxies del nivel de aplicación
  - **retransmiten** mensajes del nivel de aplicación
  - bloqueo de aplicaciones no permitidas (las que no cuenten con Proxy)
  - control del tráfico de las aplicaciones permitidas
- Proxy ”comprende” el protocolo de una aplicación concreta
  - previene abusos
  - permite limitar porciones concretas del protocolo
  - pueden detectar uso de protocolos no permitidos en puertos estándar

Filtrado de paquetes

Dispositivos encaminan tráfico entre red externa e interna

- Trabajan en las capas de red (IP) y/o transporte (TCP,UDP)
- Pueden implementarse como un elemento añadido a un router o como un equipo dedicado

Analizan cada paquete (antes del enrutado) y aplican reglas para decidir si se retransmite o descarta

- inspecciona cabeceras del paquete y comprueba si encajan en la lista de reglas aceptación/rechazo
- filtrado basado en la información de cada paquete concreto
  - cada paquete se analiza de forma aislada
  - no tiene en cuenta si son parte de una conexión (métodos *"sin estado"*)

Reglas de filtrado emplean la información contenida en cada paquete de red analizado

```
NIVEL DE RED          NIVEL DE TRANSPORTE   OTROS
- dir. IP origen      - puerto origen       - interfaces entrada/salida
- dir. IP destino     - puerto destino      - tamaño
- tipo protocolo      - flags TCP           - info. MAC
                                            - patrones simples sobre carga útil
```

$\Rightarrow$ Control de servicios basado en filtrado de puertos estándar

Funcionamiento general

Filtrado de paquetes se configura como una lista de reglas estáticas

- **condiciones:** basadas en los campos de la cabecera IP y/o TCP
- **acciones:** descartar, rechazar, retransmitir, logear

Funcionamiento

1. Reglas comprobadas secuencialmente una a una (el orden es relevante)
2. Cuando hay una correspondencia, se invoca la regla (aceptar o denegar el paquete)
3. Si ninguna regla encaja, se aplica la acción predeterminada
   - **denegar por defecto:** lo que no está expresamente permitido, está prohibido
     - política más conservadora, todo está bloqueado
     - servicios permitidos deben añadirse explícitamente
     - más robusta (mayor nivel de protección)
   - **aceptar por defecto:** lo que no está expresamente prohibido, está permitido
     - política más permisible, todo está permitido
     - servicios vulnerables/peligrosos deben bloquearse explícitamente
     - nivel de protección más bajo (incrementa el riesgo)

Filtros ”con estado”

- Llevan registro de las conexiones que pasan a través del Cortafuegos
- Estudian y reconocen los paquetes
  - que inician/finalizan las conexiones
  - que forman parte de conexiones establecidas
  - que están relacionados con conexiones previas
- Permiten un control más fino que los filtros sin estado
- Ejemplo en GNU/Linux: NETFILTER/iptables con módulos de seguimiento de conexiones *(connection tracking)*

Ventajas y limitaciones del Filtrado de Paquetes

Ventajas.

- Simplicidad: maneja una información mínima (cabeceras de los paquetes) y la especificación de reglas es simple
  $\Rightarrow$ permite establecer un filtrado en casi cualquier red
- Rapidez/eficiencia: mínimo proceso a realizar sobre los paquetes para la toma de decisiones (coste y retardos reducidos)
- Son transparentes al usuario (no requieren participación por su parte)

Limitaciones.

- Usan info. de ”bajo nivel”, no examinan datos de niveles superiores (capa aplicación)
  - no puede evitar ataques que aprovechen vulnerabilidades o funcionalidades específicas de las aplicaciones/protocolos de capas superiores
  - no pueden bloquear comandos específicos del protocolo de aplicación
- Posibilidades de registro (log) reducidas
  - manejan info. limitada (capas IP y/o TCP/UDP)
- No admiten esquemas de autenticación/control de usuarios (requiere info. de niveles más altos)
- Reglas de filtrado muy complejas pueden ser difíciles de definir/gestionar
  - pueden ocultar agujeros de seguridad consecuencia de una configuración inadecuada

Pasarelas nivel aplicación (Proxies)

Dispositivos repetidores de tráfico a nivel de aplicación.

- Actúan como servidor intermediario, ofreciendo un núm. limitado de servicios a nivel de aplicación
- Posible control a más alto nivel
  - permite analizar las conexiones a nivel de cada aplicación concreta
  - permite la autenticación de usuarios
- Para cada protocolo de nivel aplicación permitido se debe ejecutar el correspondiente Proxy en el equipo que actúe como cortafuegos.
  - Cortafuegos sólo permite tráfico de aplicaciones que cuenten con Proxy

$\Rightarrow$ Control de servicios basado en intermediarios para los servicios aceptados

Funcionamiento

- Para conectar con servidor externo, cliente interno establece conex. con Proxy
- Proxy establece conexión con servidor externo en nombre del cliente
- 2 conexiones $\begin{cases} \textbf{cliente\_interno} \leftrightarrow \textbf{Proxy :} \text{ Proxy hace papel de servidor} \\ \textbf{Proxy} \leftrightarrow \textbf{servidor\_externo :} \text{ Proxy hace papel de cliente} \end{cases}$
- Proxy recibe, examina y retransmite el tráfico bidireccionalmente entre cliente(interno) y servidor(externo) tomando todas las decisiones de envío de mensajes
  - puede realizar otras tareas: cache de datos/recursos recibidos,...

Ventajas y limitaciones de los Proxies

Ventajas

- Mayor control que con filtros de paquetes
  - Proxy está especializado en analizar/controlar una aplicación concreta
  - Mayor flexibilidad que filtros de paquetes $\Rightarrow$ control "fino" de cada aplicación concreta
- Centralización de la información de cada protocolo del nivel de aplicación
- Evitan comunicación directa con servidor destino
- Posibilidad de funcionalidades adicionales
  - Autenticación de usuario a alto nivel
  - Posibilita registro de eventos a nivel de aplicación
  - Servicios añadidos: caché, gestión/compartición conexiones, ...

Inconvenientes

- Exige contar con un Proxy para cada aplicación que se pretenda controlar
- No totalmente transparentes al usuario (requieren cierta intervención)
- Mayor coste de procesamiento (mantener/controlar 2 conexiones)
  - menor rendimiento que filtros de paquetes
  - cantidad de servicios con su propio Proxy está limitada por recursos del equipo cortafuegos.

Pasarelas de nivel de circuitos

Mecanismo de control híbrido entre filtros de paquetes y Proxies de aplicación

- Puede ser un sistema autónomo o una función complementaria realizada por un Proxy para ciertas aplicaciones concretas
- Funciona como intermediario (Proxy) de conexiones TCP
  - No permiten conexión TCP directa de extremo a extremo
  - La pasarela de circuitos establece 2 conexiones TCP
    `pasarela` ↔ `equipo_interno`, `pasarela` ↔ `equipo_externo`
  - Pasarela de circuitos SOLO retransmite paquetes TCP desde una conexión a la otra sin analizar sus contenidos
  - La seguridad que aporta consiste en determinar qué conexiones se permiten
- Diferencia con Proxies de aplicación
  - no analiza el tráfico (no maneja info. del protocolo de aplicación)
  - menor necesidad de procesamiento (sólo retransmite paquetes TCP entre 2 conexiones ya abiertas)
- Ejemplo: SOCKS (servidor + librería cliente)

Topologías de cortafuegos (I)

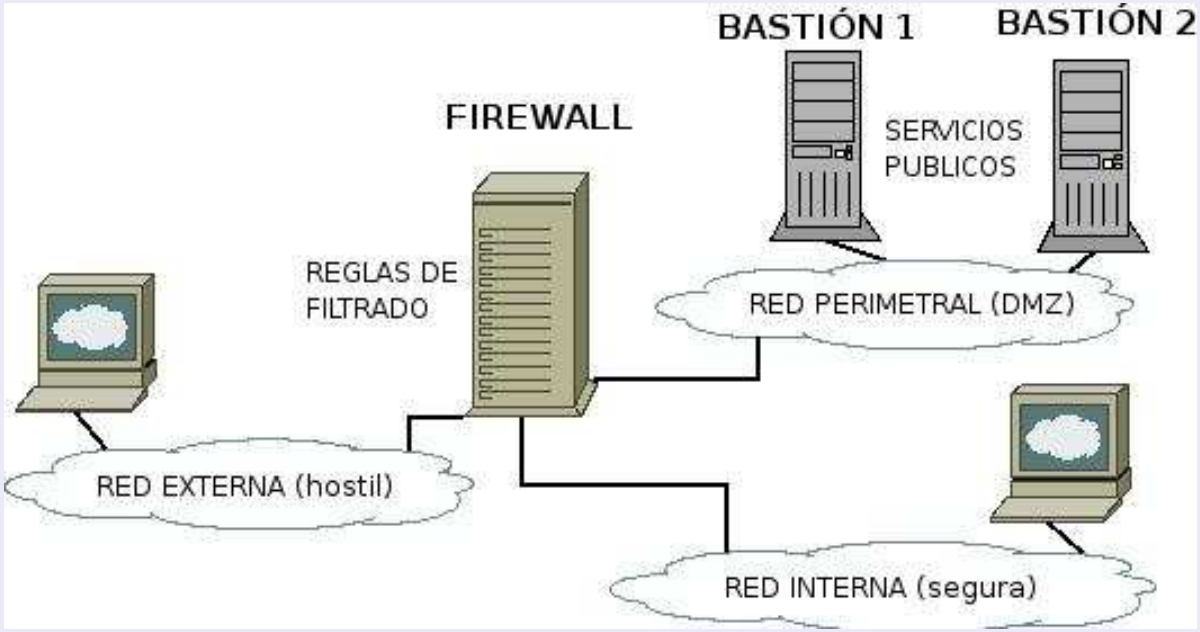
Decisiones clave: ubicación de { reglas de filtrado / servicios públicos

Esquema habitual: Red perimetral/ Zona desmilitarizada (DMZ)

- Ofrecer servicios al exterior en máquinas ubicadas en la DMZ
  - Alojará los servidores accesibles desde la red externa
  - Opcionalmente, alojará a los Proxies de aplicación usados por la red interna
- Únicas máquinas accesibles desde exterior (*hosts bastion*)
  - Elementos potencialmente vulnerables
  - Administración delicada
    - mínimos servicios software instalados (sólo los imprescindibles)
    - actualizaciones de seguridad del S.O. + servidores
    - monitorización de ficheros de log
- **Finalidad:** aislar servicios al exterior para evitar/controlar su acceso a la red protegida en caso de verse comprometidos por un atacante
  - Cortafuegos tiene control sobre el tráfico entre DMZ y red interna
  - Equipos internos siguen sin poder confiar en hosts de DMZ
  - Mayor seguridad y robustez

Topologías de cortafuegos (II)

(a) DMZ con cortafuegos con 3 interfaces



Topologías de cortafuegos (III)

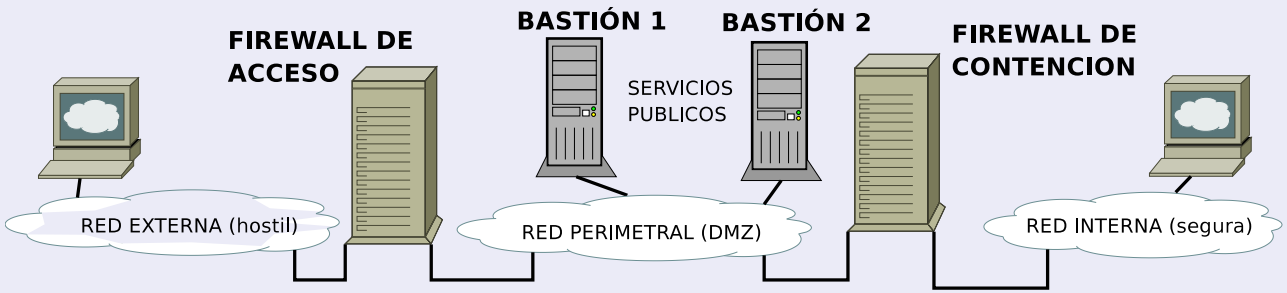
(b) DMZ con doble firewall (*screened subnet*)

Mejora del esquema anterior: añade un segundo cortafuegos

- cortafuegos externo (de acceso): bloquea tráfico no deseado externo → DMZ
- cortafuegos interno (de contención): bloquea tráfico no deseado DMZ → interno

Idea: aumentar la separación entre la red de servicios externos(DMZ) y la red interna

- DMZ se sitúa entre cortafuegos externo e interno
- Se crean 2 niveles de seguridad (red DMZ + red interna)
- Tráfico de exterior a red interna debe atravesar 2 cortafuegos



Topologías de cortafuegos (IV)

(b) DMZ con doble firewall (cont.)

Mayor tolerancia a fallos: evita puntos únicos de fallo

- Superando cortafuegos externo (acceso), sólo quedaría desprotegida DMZ
- Aún comprometiendo un equipo de la DMZ, se contaría con el cortafuegos de contención (no hay acceso directo desde DMZ a red interna)

Ventajas.

- mayor robustez y tolerancia a fallos
- mayor flexibilidad: pueden definirse tantas DMZ como sea preciso (con distintos requisitos/niveles de seguridad)

Limitaciones.

- dificultad de administración (gestionar 2 conjuntos de reglas de filtrado funcionando en conjunto)
- sensación de falsa seguridad

Redes Privadas Virtuales

VPN (*Virtual Private Network*): Conjunto de tecnologías que permiten extender el alcance de una red local (privada) sobre la infraestructura de una red pública no controlada, manteniendo la confidencialidad del tráfico.

- Suelen basarse en el concepto de *tunneling*
  - Se crea/mantiene una conexión lógica entre dos extremos
  - Encapsulado de tráfico de un protocolo dentro de paquetes de otro protocolo distinto
- Hacen uso de enlaces cifrados para definir conexiones protegidas entre porciones ”separadas” de la propia red
- Ejemplos típicos:
  - Interconexión ”segura” entre 2 delegaciones de una misma organización usando una red pública no segura (Internet) [VPN punto a punto]
  - Conexión segura de un usuario a la red interna desde equipos fuera de la red de la organización [VPN de acceso remoto]
- Evitan el uso de lineas dedicadas
  - menor coste (red pública vs. enlace de pago)
  - mayor flexibilidad

Tecnologías de VPN (I)

IPsec: *Internet Protocol Security*

Familia de protocolos que protegen el tráfico a nivel IP (capa 3 [red])

- Ofrece autenticación e integridad de los paquetes y, opcionalmente, confidencialidad (cifrado)

Protocolos IPsec

- Protocolo AH (*Authentication Headers*)
- Protocolo ESP (*Encapsulating Security Payload*)
- Protocolos ISAKMP (*Internet Security Association and Key Management Protocol*) y IKE (*Internet Key Exchange*)

Modos de funcionamiento

- Modo **transporte**: IPsec protege ”carga útil” (usado en esquemas *host-a-host*)
- modo **tunel**: IPsec encapsula un paquete IP orginal completo (usado en esquemas *red-a-red*)

Tecnologías de VPN (II)

Encapsulado de protocolos PPP

- **PPTP**: *Point to Point Tunneling Protocol*
  - Encapsula paquetes PPP (*Point-to-Point Protocol*) dentro de datagramas IP que circulan sobre una red de paquetes pública no punto-a-punto.
- **L2TP**: *Layer 2 Tunneling Protocol*
  - Protocolo genérico de *tunneling*, evolución/mejora de PPTP
  - Paquetes L2TP encasulan el tráfico sobre paquetes UDP
  - Suele combinarse con IPsec (L2TP/IPsec)

VPNs de nivel de transporte/aplicación

- **OpenVPN**: encapsula tráfico IP ó Ethernet sobre una conexión SSL/TLS establecida entre lo extremos del enlace
- **Túneles SSH**: SSH permite la redirección de puertos (locales o remotos) sobre la conexión cifrada establecida entre el cliente y el servidor

Conceptos básicos

Intrusión: Conjunto de acciones que pretenden comprometer la confidencialidad, integridad o disponibilidad de un recurso (red o equipo)

- Origen: atacantes externos, usuarios internos, software malicioso (*malware*)

Sistemas de detección de intrusiones

IDS (*Intrusion Detection Systems*): Monitorizan redes o sistemas para detectar e informar actividades o accesos no autorizados.

- Generan alertas y registran los eventos detectados
- Opcionalmente, correlacionan eventos detectados con info. adicional (alertas de cortafuegos, BD de vulnerabilidades, etc)

Pasivos $\rightarrow$ detectan + generan alertas

Sistemas de prevención de intrusiones

IPS (*Intrusion Prevention Systems*): Monitorizan redes o sistemas para detectar e intentar impedir actividades o accesos no autorizados.

- Funcionamiento "*en-linea*" (analizan y actúan "sobre la marcha")
- Bloquean/descartan los paquetes o las acciones sospechosas o no permitidas

Activos $\rightarrow$ detectan + bloquean la intrusión

IDS e IPS complementan a otros mecanismos de seguridad (cortafuegos, cifrado)

Tipos de IDS/IPS (I)

NIDS: detectores de intrusiones en red

Capturan el tráfico de la red (ubicados en ”zonas estratégicas”: DMZ, routers de acceso, etc) y lo evalúan para determinar si se corresponde con una intrusión

- Análisis de intrusiones a nivel de paquetes de red
- Monitoriza todo el tráfico de una porción de la red
  - *on-line* [captura y análisis simultáneto] *vs. off-line* [captura y análisis posterior]
  - Suelen hacer uso de sniffers conectados a hubs, puertos de administración de switches (*span ports*), bridges (dispositivos TAP)
- Sensores accesibles a través de la red de la organización o mediante una ”red de gestión” separada
- Suelen centrarse en detectar ataques DOS (*Denial Of Service*), escaneo de puertos, paquetes malformados, explotación de vulnerabilidades (en servicios o aplicaciones), etc
- Ejemplos:
  - SNORT (`http://www.snort.org`)
  - Suricata (`https://suricata-ids.org`)

Tipos de IDS/IPS (II)

HIDS: detectores de intrusiones en host

Analizan los eventos que se producen en un equipo (host) determinado para determinar si está sufriendo un ataque

- Sensores (agentes) monitorizan un equipo concreto
- Aspectos monitorizados:
  - Logs del sistema y de las aplicaciones
  - Llamadas al sistema
  - Modificaciones sobre el sistema de ficheros (BD con hashes de ficheros/directorios sensibles)
- Suelen centrarse en detectar el ”abuso” de privilegios (escalada de privilegios)
- Ejemplos:
  - OSSEC (`http://www.ossec.net`)
  - SAGAN (`http://sagan.quadrantsec.com`)
  - TRIPWIRE(`http://www.tripwire.com`)

Funcionamiento (I)

Basados en firmas

Cuentan con una BD con firmas de ataques conocidos (aproximación similar a los antivirus)

- Firma = patrón que se corresponde con una amenaza/ataque/vulnerabilidad conocida
- Ejemplo en NIDS: paquete cuya carga útil incluya fragmentos de un *shellcode* conocido
- Ejemplo en HIDS: entrada (o secuencia de entradas) de intentos de acceso no autorizados en log de una aplicación

Ventajas

- Simplicidad y eficiencia
- Efectivo detectando amenazas conocidas
- En general, menores tasas de falsos positivos

Inconvenientes

- Escasa o nula utilidad ante nuevas amenazas nunca vistas
- Vulnerable a técnicas de evasión (*IDS evasion*): recodificación de caracteres, cambios de formato, etc
- En NIDS no pueden analizar tráfico de protocolos cifrados

Funcionamiento (II)

Basados en anomalías

Construyen un ”modelo” o perfil del uso normal de la red o del equipo.

- Perfiles capturan el comportamiento ”normal” (legítimo) de los usuarios, aplicaciones, servicios, etc
- Parámetros: consumo de CPU o memoria, intentos de login, nº de conexiones, tamaño de paquetes, uso de ancho de banda, etc
- Requieren un proceso de previo de monitorización para ”aprender” ese modelo
  - Entrenamiento estático [1 vez] o dinámico [ajuste constante]
  - Técnicas: modelos estadísticos, redes neuronales, etc

Actividad que no encaja con perfil habitual se considera sospechosa

Ventajas

- Efectivo frente a nuevas amenazas
- Puede detectar variantes de ataques/amenazas conocidas

Inconvenientes

- Costoso: requiere entrenamiento + análisis y actualización del comportamiento actual
- Mayor tendencia a producir falsos positivos
- Difícil construir y mantener un perfil adecuado

Funcionamiento (III)

Basados en análisis de protocolos

IDS/IPS comprende y modeliza el comportamiento y estado de los protocolos y aplicaciones empleados/permitidos.

- Dentro de cada estado del protocolo/aplicación se conoce cómo es la actividad legítima esperada
- Ejemplos: tamaños de mensajes usuales, orden usual en los comandos/argumentos del protocolo

Ventaja: detecta secuencias inesperadas de comandos/mensajes
Inconveniente: costoso en recursos y difícil definir el "comportamiento" legítimo de un protocolo/aplicación dado

Falsos positivos vs. Falsos negativos

Todos los tipos de IDS/IPS susceptibles de cometer errores de clasificación

	Situación real	
Predicción IDS	Intrusión	Legítimo
Intrusión	Positivo real	Falso positivo
Legítimo	Falso negativo	Negativo real

- Difícil configurar IDS/IPS para conseguir a la vez tasas bajas de falsos positivos y falsos negativos

- En general, se opta por tolerar cierta cantidad de falsos positivos y reducir al máximo falsos negativos

Uso de honeypots

Sistemas (reales o simulados) cuya intención es atraer a los atacantes

- Aparentan ser vulnerables y presentan debilidades ”evidentes” ante el atacante
- Separados de la red real y constantemente monitorizados

Utilidad

- Como ”entretenimiento” para los atacantes (dedican recursos a atacar equipos no reales)
- Mecanismo para detectar intentos de intrusión, analizarlos y recabar información sobre el atacante

Generalización: *honeynets* (redes de sistemas aparentemente vulnerables)