Firewall

Parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas.

Se trata de un dispositivo o conjunto de dispositivos configurados para permitir, limitar, cifrar, descifrar, el tráfico entre los diferentes ámbitos sobre la base de un conjunto de normas y otros criterios.

Los cortafuegos pueden ser implementados en hardware o software, o en una combinación de ambos.

Métodos de ataque

- Acceso no autorizado
    - Se puede evitar el acceso a la red a todo el mundo excepto a los usuarios deseados.
- Aprovechamiento de las debilidades conocidas de un programa
    - Deshabilitar los servicios vulnerables o encontrar alternativas.
- Denegación de servicio
    - Impedir que el tráfico de red sospechoso alcance sus máquinas y que lleguen órdenes y peticiones de programas sospechosos.
- Suplantación de identidad (spoofing)
    - Verificar la autenticidad de los datagramas y órdenes.
    - Evitar el encaminamiento de datagramas con direcciones de origen no válidas.
    - Introducir mecanismos no predecibles de control de la conexión, como los números de secuencia de TCP y la asignación dinámica de puertos.
- Sniffing
    - Evitar el uso de tecnologías de red con difusiones (bus).

DMZ

También es frecuente conectar el firewall a una tercera red, llamada «zona desmilitarizada» o DMZ, en la que se ubican los servidores de la organización que deben permanecer accesibles desde la red exterior.

Un firewall correctamente configurado añade una protección necesaria a la red, pero que en ningún caso debe considerarse suficiente.

La seguridad informática abarca más ámbitos y más niveles de trabajo y protección.

Filtrado IP

El filtrado de IP es simplemente un mecanismo que decide qué tipos de datagramas de IP serán procesados normalmente y cuáles serán descartados.

Por descartados se entiende que el datagrama se elimina y se ignora completamente, como si nunca se hubiera recibido.

Formas de Filtrado

Se recibe el datagrama de IP. (1)

Se examina el datagrama de IP entrante para determinar si está destinado a un proceso de esta Máquina. Si el datagrama es para esta máquina,se procesa localmente.(2)

Si no está destinado a esta máquina, se realiza una búsqueda en la tabla de encaminamiento de una ruta adecuada y el datagrama se reenvía por la interfaz adecuada o se elimina si no se puede encontrar una ruta. (3)

Los datagramas procedentes de procesos locales se envían hacia el 'software' de encaminamiento para ser reenviados hacia la interfaz apropiada. (4)

Se examina el datagrama de IP saliente para determinar si existe una ruta válida que escoger, si no es así, se elimina.

Se transmite el datagrama de IP. (5)

El flujo **432** representaría un flujo de datos vía una conexión 'loopback'.

Los flujos **12** y **45** representan los flujos de entrada y de salida de datos de un programa de red ejecutándose en nuestro 'host' local.

El flujo **135** representa nuestra máquina encaminando datos entre un 'host' sobre nuestra red Ethernet y un 'host' alcanzable vía nuestro enlace PPP.

Linux aceptables

El 'software' de firewall de Linux que se verá, iptables, proporciona dos características muy útiles:

- auditoría de IP
- enmascaramiento de IP

La forma de acceder al firewall es a través de comandos en una terminal de caracteres.

Pueden existir otros firewalls, e incluso interfaces gráficas que facilitan la utilización, pero dado que en la práctica se utilizan máquinas virtuales limitadas, utilizaremos solo la interface de caracteres usando una terminal.

El comando iptables se utiliza para configurar tanto el filtrado de IP como la traducción de direcciones de red.

Existen varias tablas de reglas, pero solo veremos las denominadas **filter** y **nat**. Por defecto, se asume la tabla 'filter' salvo que se especifique la opción −t nat.

También se proporciona cinco cadenas predefinidas:

- **INPUT** y **FORWARD** están disponibles para la tabla filter
- **PREROUTING** y **POSTROUTING** están disponibles para la tabla nat (veremos NAT más adelante)
- **OUTPUT** está disponible para ambas tablas

La sintaxis general de la mayoría de las órdenes de iptables es:

iptables orden especificación_de_regla extensiones

Cadenas iptables

Las cadenas de filtrado de la tabla filter son:

- **INPUT**: paquetes destinados al host
- **OUTPUT**: paquetes que salen del host
- **FORWARD**: paquetes que deben ser reenviados por host

El software iptables está en todos los linux, y como ya es conocido, un host puede funcionar como router.

En la cadena **INPUT** se ubican reglas que regulan el tráfico que entra al host, pero que esta dirigido al host. Esto significa que los paquetes traen como dirección IP destino algunas de las IP de host/router.

En la cadena **OUTPUT** se ubican reglas que regulan el tráfico que sale del host, de las capas superiores y con destino a un host externo. Significa que los paquetes salen con dirección IP origen del host y con IP destino la del host externo.

En la cadena **FORWARD** se ubican las reglas de tráfico que pasan por el host. En este caso el host oficia como router y por lo tanto el filtrado se hace entre una interface de entrada y otra de salida. En esta cadena se ubican las reglas que regulan el tráfico en la red.

Ejemplos:

- Si queremos que el router de la red local no sea accedido con ssh vamos a poner una regla en la cadena INPUT bloqueando esto.
- Si queremos que el router no envíe mensajes de error a los hosts pondremos una regla de bloqueo en la cadena OUTPUT.
- Si queremos evitar el trafico torrent de entrada o salida a nuestra red, debemos poner reglas en la cadena FORWARD.

Reglas

Cada cadena tiene una series de reglas. Estas reglas especifican alguna condición y una acción en caso de cumplirse.

Las acciones pueden ser dos:

- ACCEPT sigue camino
- DROP se descarta

Cada paquete es checkeado contra estas reglas, comenzando por el tope de la lista, y si machea con alguna regla, se toma la acción definida en la regla.

Una vez que el paquete macheó con una regla y se llevó adelante alguna acción, el paquete es procesado de acuerdo con ese resultado y no se lo checkea con otras reglas en la tabla.

Si un paquete pasa todas las reglas en la tabla y llega al final sin machear con ninguna de ellas, se lleva adelante la acción por defecto, que puede ser aceptar o descartar el paquete.

Reglas por defecto:

- Podemos setear como regla por defecto DROP, y luego agregar reglas ACCEPT para los paquetes que vienen con IP origen confiable, o para ciertos puertos en los que tenemos servicios ejecutándose, por ejemplo: FTP server, Web Server, Samba file server etc.
- Podemos setear como regla por defecto ACCEPT y luego agregar reglas específicas para bloquear (DROP) paquetes que pueden venir de direcciones IP falsas, o para ciertos puertos en los que tenemos servicios privados o no tenemos servicios ejecutándose.

Generalmente la primer opción es usada para la tabla INPUT, donde queremos controlar los paquetes entrantes y la segunda para la tabla OUTPUT, ya que podemos confiar más en el tráfico que sale desde nuestra máquina.

No se aconseja intercalar acciones en las reglas. Es más simple todas las reglas ACCEPT y por defecto DROP, o a la inversa, todas DROP y por defecto ACCEPT.

Iptables desde la consola

Para trabajar con IPTABLES debemos ser root.

Para saber si tenemos IPTABLES instalado podemos ejecutar:

```
redes~ $ sudo iptables --help
iptables v1.6.0

Usage: iptables -[ACD] chain rule-specification [options]
       iptables -I chain [rulenum] rule-specification [options]
       iptables -R chain rulenum rule-specification [options]
       iptables -D chain rulenum [options]
       iptables -[LS] [chain [rulenum]] [options]
       iptables -[FZ] [chain] [options]
       iptables -[NX] chain
       iptables -E old-chain-name new-chain-name
       iptables -P chain target [options]
       iptables -h (print this help information)

Commands:
```

Iptables - Comandos útiles

Y para ver si IPTABLES se esta ejecutando podemos ejecutar:

- sudo iptables -L

Para saber si los módulos de IPTABLES están cargados podemos ejecutar:

- lsmod | grep ip

Iptables - más Comandos

```
iptables -F
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED, RELATED -j ACCEPT
iptables -L -v
```

- **iptables -F**: -F borra todas las reglas presentes
- i**ptables -P INPUT DROP**: -P setea la política por defecto para la cadena especificada, en este caso INPUT. Hemos seteado la política por defecto DROP en la tabla INPUT. Esto significa que si un paquete entrante no machea alguna de las reglas que siguen será descartado.
- **iptables -P FORWARD DROP**: ídem anterior, hemos seteamos la política por defecto en la cadena FORWARD a DROP, ya que no estamos usando nuestro host como router, así que no debería haber paquetes pasando por él.
- **iptables -P OUTPUT ACCEPT**: hemos seteado la política de la cadena OUTPUT a ACCEPT, ya que queremos permitir el tráfico saliente
- **iptables -A INPUT -i lo -j ACCEPT**:
    - -A agregar una regla a una cadena, INPUT en este caso
    - -i (interface) para especificar paquetes que son destinados a la interfaz local (127.0.0.1)

    - -j (jump) para especificar la acción para los paquetes que machean la regla, en este caso ACCEPT.
    - Así que esta regla acepta todo el tráfico entrante destinado a lo. Esto suele ser requerido por aplicaciones que necesitan poder comunicarse con la placa de red.

- **iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT**:
    - -A estamos agregando la regla a la cadena INPUT.
    - -m es usada para cargar un modulo state. El modulo state examina el estado de un paquete y determinar si es NEW, ESTABLISHED or RELATED NEW hace referencia a paquetes entrantes que son conexiones entrantes nuevas que no fueron iniciadas por el host.
    - ESTABLISHED y RELATED hacen referencia a paquetes que son parte de una conexión ya establecida, o relacionados a una conexión ya establecida
- **iptables -L -v**:
    - -L lista las reglas que hemos agregado para ver que se hayan cargado correctamente

Por último, debemos salvar las reglas que cargamos:

- **sudo iptables-save**

Iptables - Parámetros Principales

Veamos ahora como abrir algunos huecos de manera segura para permitir alguna conexiones externas.

Interface:

- En el ejemplo anterior vimos como aceptar todos los paquetes que ingresan por una interface en particular, en este caso la interface localhost
    - **iptables -A INPUT -i lo -j ACCEPT**
- Suponiendo que tenemos 2 interfaces
    - eth0, para la conexión LAN
    - ppp0, para el módem de Internet

- Podríamos querer permitir todo el tráfico entrante desde la LAN y continuar filtrando paquetes desde Internet. Escribimos:
    - **iptables -A INPUT -i lo -j ACCEPT**
    - **iptables -A INPUT -i eth0 -j ACCEPT**
- Pero, cuidado, si aceptáramos todo el tráfico entrante desde Internet, habríamos deshabilitado el firewall
    - **iptables -A INPUT -i lo -j ACCEPT**
    - **iptables -A INPUT -i ppp0 -j ACCEPT**

Direcciones IP:

- Abrir una interface a todos los paquetes entrantes no es muy restrictivo, queremos tener más control sobre lo que vamos a aceptar y rechazar.
- Supongamos que tenemos una pequeña red privada 192.168.0.x. Podemos abrir nuestro firewall a los paquetes entrantes que vienen desde una IP de confianza, por ejemplo 192.168.0.4
    - **iptables -A INPUT -s 192.168.0.4 -j ACCEPT**
    - Primero agregamos la regla ACCEPT a la cadena INPUT para todos los paquetes que provienen de la IP origen (-s) 192.168.0.4
- Si queremos permitir el ingreso de paquetes desde un rango de direcciones IP, usamos una máscara de red o la notación / para especificar el rango.
- Por ejemplo, si queremos abrir nuestro firewall a todos los paquetes entrantes del rango 192.168.0.x (x desde 1 a 254), podríamos alguno de estos métodos:
    - **iptables -A INPUT -s 192.168.0.0/24 -j ACCEPT** (usando notación barra)
    - **iptables -A INPUT -s 192.168.0.0/255.255.255.0 -j ACCEPT** (usando máscara de red)
- También podemos filtrar por direcciones MAC.
- Para hacer esto debemos cargar el modulo mac que permite el filtrado por dirección MAC.

- **iptables -A INPUT -s 192.168.0.4 -m mac --mac-source 00:50:8D:FD:E6:32 -j ACCEPT**

- Primero usamos -m mac para cargar el modulo y luego usamos –mac-source para especificar la dirección mac de la IP origen.

- Esto puede ser útil para prevenir spoofing de la IP origen y sólo permitiremos el ingreso de paquetes genuinamente generados por 192.168.0.4 y bloqueara los paquetes spoofed que llegan con esa dirección IP.

- No es seguro que esta técnica funcione con paquetes que llegan desde internet, pero si dentro de una LAN.

Puertos y Protocolos:

- Lo primero que debemos averiguar es que puertos utiliza cada servicio.

- Y para poder usar macheo de puerto destino y origen (--dport o --sport), primero debemos especificar el protocolo (tcp, udp, icmp, all).

- Por ejemplo bittorrent usa el puerto 6881, así que queremos permitir todos los paquetes TCP con puerto destino 6881:
    - **iptables -A INPUT -p tcp --dport 6881 -j ACCEPT**
- También podemos usar rangos de puertos
    - **iptables -A INPUT -p tcp --dport 6881:6890 -j ACCEPT**
- Seguimos con el ejemplo de bittorrent:
    - No necesitamos que firewall examine cada paquete, sólo debe examinar el primer paquete, ya que la regla de inspección de estado de paquete permitirá que los paquetes de nuestra conexión establecida y relacionada pasen.

    - Solo necesitamos una regla NEW que permita conexiones a un puerto dado, así que si usamos el modulo spi, sólo necesitaremos filtrar contra conexiones tcp nuevas en el puerto 6881

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

        - **iptables -A INPUT -p tcp --dport 6881 -m state --state NEW -j ACCEPT**

- veamos un ejemplo con SSH:
    - SSH usa el puerto 22 y protocolo TCP.
    - Así que si queremos permitir login remoto, necesitamos abrir el puerto 22 de TCP
        - **iptables -A INPUT -p tcp --dport 22 -j ACCEPT**
    - Esta linea abre el puerto 22 a todas las conexiones lo que deja una brecha de seguridad que puede ser fácilmente vulnerada.
    - Si tenemos algunas direcciones IP confiables, podemos limitar el acceso para esas direcciones IP. Por ejemplo, si queremos permitir la conexión desde la red privada 192.168.0.x, podemos especificar ese rango de IP
        - **iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 22 -j ACCEPT**
    - Usar filtrado IP nos permite abrir el acceso a SSH de modo seguro en el puerto 22 sólo para las direcciones IP confiables.
    - Por ejemplo, podemos usar este método para permitir login remoto entre la red del trabajo y la hogareña. Mientras que para todas las otras direcciones IP el puerto parecerá cerrado como si el servicio estuviera deshabilitado.
    - Para poder hackear el sistema, un hacker necesitará conocer la dirección IP confiable y spoof paquetes como si vinieran desde esa dirección.
    - No necesitamos filtrar todos los paquetes, sólo las conexiones nuevas:
        - **iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 22 -m state --state NEW -j ACCEPT**

Preguntas de Repaso

- ¿Qué es un firewall y dónde se implementa?
- ¿Dónde se ubican los firewalls?
- ¿La única función de un firewall es impedir ataques?
- ¿En qué casos se pondrían reglas en la cadena FORWARD de un host?

Referencias

- Notas de esta presentación
- Página de la materia en pedco.uncoma.edu.ar
- Kurose-Ross Computer Networking A Top-down Approach Featuring the Internet Third Edition
- Redes de Computadoras, 5ta Edición - James F. Kurose & Keith W. Ross
- Prefijos Binarios (En la página de la materia)
- IPTables en la Wiki de Centos