Firewall en Linux

Luis Eduardo Vivero Peña.
Director Centro de Difusión del Software Libre
Ingeniero de Proyectos Corporación Linux

Temario

1) Introducción a Firewall
   - ¿Qué es un Firewall?
   - Objetivos de un Firewall
   - Tipos de Firewall
2) Netfilter/Iptables
   - ¿Qué es Netfilter/Iptables?
   - ¿Qué se puede hacer con Iptables?
   - Políticas por defecto
   - Cadenas

Temario

2) ...Netfilter/Iptables
   - Tablas
   - Acciones Básicas
   - NAT
3) Ejemplos Prácticos
   - Compartir internet
   - Firewall para una red corporativa
   - Firewall como servicio en Debian GNU/Linux

Introducción a Firewall

Introducción a Firewall

- ¿Qué es un Firewall?
  - Un cortafuegos o firewall es un elemento de hardware o software que se utiliza para aumentar la seguridad de redes informáticas.

Introducción a Firewall

- Objetivos de un Firewall
  - Establecer seguridad entre diferentes zonas de confianza.
    - Internet --> confianza nula.
    - DMZ --> Zona desmilitarizada, confianza mayor que internet.
    - LAN --> red local, usuarios, zona de alta confianza.

Introducción a Firewall

- ...Objetivos de un Firewall
  - Proteger una red o host de intrusiones o accesos ilícitos.
  - Redirigir paquetes a una máquina en una red interna.
  - Otorgar acceso (ssh por ejemplo) solo desde sitios conocidos.

Tipos de Firewall

- Por Hardware
  - Cisco, Pix, etc.
- Por software
  - Firewall de Capa de Red (internet)
    - Stateless
    - Statefull
  - Firewall de Capa de Aplicación

Netfilter/Iptables

- ¿Qué es Netfilter/Iptables?
  - Netfilter corresponde a la infraestructura que posee un sistema GNU/Linux para realizar diferentes operaciones en el manejo y control de tráfico de paquetes.
  - Iptables es una estructura genérica de tablas para la definición de reglas.

Netfilter/Iptables

- ...¿Qué es Netfilter/Iptables?
  - Las dos estructuras mencionadas se encuentran dentro del kernel linux de las ramas 2.4.x y 2.6.x.
  - Existe una implementación similar para ramas anteriores como la 2.2.x, la cual posee ipchains, y la 2.0.x que posee ipfwadm.
  - Netfilter, Iptables, connection tracking y el subsistema NAT constituyen el framework completo.

¿Qué se puede hacer con iptables?

- Filtrado de paquetes
- Redirección de paquetes
- Cambiar fuente de los paquetes
- Cambiar destino de los paquetes

Políticas por defecto

- Permitir todo, denegar algunos puertos, protocolos y/o redes...mala idea...siempre quedará algo abierto de más, es inseguro.
- Denegar todo (entradas y lo que pasa a través de las interfaces), y luego permitir lo estrictamente necesario. ¡Sí, esto lo lo mejor!

Políticas por defecto

- Dejar salir todo (algunos maniáticos filtran la salida...).
- Si dentro del firewall, un determinado tráfico de paquetes no hace match con ninguna regla en particular, entonces se aplica la política por defecto que corresponda.

Cadenas

- Input
  - El tráfico de paquetes que entra, independientemente de la interfaz por la cual lo hace.
- Output
  - El tráfico de paquetes que sale, independientemente de la interfaz por la cual lo hace.

Cadenas

- Forward
  - Corresponde al tráfico de paquetes que pasa desde una interfaz a otra.
- Prerouting
  - En esta instancia se modifica el destino de los paquetes (IP).
- Postrouting
  - En esta instancia se modifica la fuente de los paquetes (IP).

Esquema

Tablas

- Filter
  - Tabla por defecto.
  - Cumple funciones de filtrado.
  - Opciones:
    - INPUT cadena
    - OUTPUT cadena
    - FORWARD cadena

Tablas

- NAT
  - Traduce direcciones IP.
  - Se puede traspasar el tráfico entrante hacia otro destino, Prerouting, DNAT.
  - Los paquetes salientes pasan a través de esta cadena antes de salir, se modifica su fuente (IP), SNAT.
  - Opciones
    - PREROUTING cadena
    - POSTROUTING cadena
    - OUTPUT cadena

Tablas

- Mangle
  - Reglas Marcianas o_O
  - Es utilizada para manejar opciones de paquetes, como quality of service.
  - Posee todas las posibles cadenas predefinidas.
    - PREROUTING cadena
    - INPUT cadena
    - FORWARD cadena
    - OUTPUT cadena
    - POSTROUTING cadena

Acciones Básicas

- ACCEPT
  - Acepta el paquete.
  - Tiene sentido en la cadena en donde se está utilizando.
  - Si es accept en la cadena INPUT, entonces se está aceptando a recibir paquetes desde un host.
  - Si es accept en la cadena FORWARD, entonces se está permitiendo el ruteo del paquete a través del host.

Acciones Básicas

- DROP
  - El paquete es eliminado sin realizar ningun tipo de procesamiento.
  - El paquete desaparece sin dar ninguna indicación a la aplicación que lo está enviando que ha sido eliminado.
  - El remitente finalmente se entera por timeout.

Acciones Básicas

- REJECT
  - El efecto es similar al de DROP, salvo que en este caso se da aviso al remitente que el paquete ha sido rechazado.
- LOG
  - Se utiliza para logear los paquetes que hacen match en una determinada cadena.
  - Puede ser usado en cualquier cadena y en cualquier tabla.
  - Se usa generalmente para debugging.

Acciones Básicas

- DNAT
  - Hace que la dirección de destino del paquete (y opcionalmente el puerto) sea reescrito/modificado por NAT.
  - Es válido solamente en las cadenas OUTPUT y PREROUTING de la tabla nat.

Acciones Básicas

- SNAT
  - Causa que la dirección fuente (y opcionalmente el puerto) sea reescrito/modificado por NAT.
  - Es válido solo en la cadena POSTROUTING en la tabla nat.

Acciones Básicas

- MASQUERADE
  - Es una forma especial y restringida de SNAT.
  - Se utiliza generalmente cuando el tráfico saliente de un gateway tiene una IP obtenida en forma dinámica.
  - Generalmente para compartir internet desde un router que está conectado a internet por módem o *DSL.

Ejemplos Prácticos

- Para compartir internet (router):
  - Ingredientes:
    - Un tarro con GNU/Linux, kernel 2.4.x/2.6.x.
    - Que el tarro tenga 2 tarjetas de red.
    - Una de la interfaces de red está conectada a internet.
    - La otra interfaz de red está configurada en una red local. Es este mismo segmento estarán los equipos que saldrán a internet.
    - Un cable cruzado, hub o switch.

Ejemplos Prácticos

- Para compartir internet (router):
  - Script:
  - #touch firewall
  - #vim firewall (sí, soy vimero, ¿y qué?)
  - Script: http://www.rootshell.be/~more/6el/inet

Ejemplos Prácticos

- Considerando una red corporativa, con LAN y DMZ
  - Condiciones:
    - Se tiene los siguientes servicios en la DMZ:
      - SMTP, IMAP
      - HTTP
      - DNS
      - VPN
    - En el gateway se tiene un proxy-caché.
    - Se debe aceptar las conexiones al puerto 22 (ssh).

Ejemplos Prácticos

- Considerando una red corporativa, con LAN y DMZ
  - Condiciones:
    - La LAN debe acceder a los servicios de la DMZ.
    - La LAN debe salir a internet y se posee IP fija.
    - En la oficina del contador hay una caja fuerte, y una cámara web (con IP). Se desea acceder a la cámara desde internet.
    - La cámara en cuestión está en la red LAN (mala idea, pero sucede).
  - Script: http://www.rootshell.be/~more/6el/firewall

Ejemplos Prácticos

- Cómo dejar el script con las reglas para que inicie en forma automática en Debian GNU/Linux:
- Copia tu script a /etc/init.d
  - #cp firewall /etc/init.d
- Puedes dejarlo que se inicie y detenga en los runlevels por defecto:
  - #update-rc.d firewall defaults

Observaciones

- Detalles a considerar:
  - Si se tiene una LAN con un DNS de caché en la DMZ para la LAN, ¡tiene que ser con vistas!
  - Un firewall no asegura que no crackearán tu máquina.
  - Si tienes abiertos algunos puertos para acceso a servicios, ¡estos deben estar bien asegurados!
  - Si tienes un DNS de caché o primario/secundario, ¡que no sea recursivo para internet!
  - Olvídate de POP-3...
  - Asegura tu kernel...

Para estudiar y probar

- Hardened Debian, una distro endurecida para la seguridad.
- http://www.debian-hardened.org/

Preguntas

- Vamos despertando...
- Saludos varios...
- ¡Gracias!
- ¿¿¿Donde está la máquina de café???