

Linux también como Firewall

Boris Quiroz Q.

<http://boris.penguin.cl>

cc by-nc-nd/2.0/cl



Generalidades

- La seguridad NO ES un problema técnico.
- Defectos de diseño y/o programación.
- Ningún sistema es seguro por sí mismo.
- Una red será tan segura como su eslabón más débil.
- Preguntas clave:
 - ¿Qué asegurar?
 - ¿Porqué asegurarlo?

Firewall? WTF?

- Controla las conexiones que entran y salen de una red.
- RFC 2979.
- Proporciona:
 - Protección de ataques o acceso no autorizado.
 - Confidencialidad.
- En algunos casos, pueden hacer NAT...

Funcionamiento de un Firewall

- Análisis de todas las conexiones que entran y salen de una red.
- Examina la procedencia/destino de cada paquete, y el tipo de servicio.
- Métodos de control de tráfico:
 - Filtro de paquetes
 - Proxy
 - Inspección de estado.
- Filtro de direcciones basado en: IP, protocolo, puertos, etc...

Tipos de Firewall

- Capa 3: filtro en base a la dirección de origen/destino. Ejemplo: Netfilter
- Capa 7: filtro en base a protocolos propios de esta capa: http, ftp, etc.
- Capa 8...
- Cisco PIX:
 - Finess
 - ASA
- Cisco ASA

Políticas

- Permitir todo:
 - Filtra lo explícitamente especificado
 - Fácil de administrar
 - Bajo control de puertos abiertos
- Denegar todo:
 - Deja pasar solo lo especificado
 - Administración un poco más compleja
 - Mayor control de qué está abierto.

Netfilter

- Módulo del kernel 2.4 en adelante.
- Analiza los encabezados antes de tomar una decisión.
- Trabaja con tres cadenas:
 - INPUT
 - OUTPUT
 - FORWARD

¿Qué es iptables?

- Es una herramienta de filtrado y revisión TCP/IP.
- Sucesor de ipchains
- Desarrollado por Rusty Rusell.
- Es parte del módulo Netfilter de los kernel 2.4 en adelante.

Características de iptables

- Filtrado “stateless” de paquetes (ipv4 e ipv6)
- Filtrado “statefull” de paquetes (ipv4)
- Traducción de direcciones y puertos
- Flexible y extensible
- Múltiples capas API para extensiones de terceros
- Graaaaaan número de módulos disponibles en patch-o-matic.

¿Qué puedo hacer con iptables?

- Construir firewalls basados con filtro de paquetes “stateless” y “stateful”
- Usar NAT y enmascaramiento para compartir Internet
- Usar NAT como proxy transparente
- Ayudar a tc y iproute2 para construir routers con QoS y políticas de seguridad
- Manipulación de paquetes
- Muuuuchas cosas más...

Funcionamiento de iptables

- Analiza cada paquete a medida que llegan y realiza alguna acción
- Un “match” indica qué paquetes cumplen dicha regla, y un “jump” que indica que hacer con el paquete
- Todo paquete recorrerá al menos una cadena, y el paquete se comprobará contra las reglas
- Si una regla especifica que hacer con un paquete, la búsqueda termina
- Si un paquete llega al final sin coincidir con ninguna regla, la política por default dirá que hacer con él: aceptarlo o rechazarlo

Compsición de iptables

- Está compuesto por 3 tablas predefinidas:
 - Nat
 - Filter
 - Mangle
- Se pueden agregar más tablas mediante módulos

Tabla filter

- Se encarga del filtro de paquetes
- Todos los paquetes pasan por esta tabla
- Contiene 3 cadenas predefinidas:
 - INPUT
 - OUTPUT
 - FORWARD

Tabla NAT

- Se encarga de la traducción de direcciones y puertos
- Analiza a todos los “primeros paquetes” de una conexión
- Tiene 3 cadenas predefinidas:
 - PREROUTING
 - POSTROUTING
 - OUTPUT

Tabla mangle

- Hace todo lo demás...
- Se encarga de ajustar las opciones de los paquetes
- Contiene todas las cadenas predefinidas posibles: PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING

En resumen...

Iptables contiene una serie de cadenas (definidas por el usuario o por default), cada una de las cuales contiene una lista de reglas. Cuando un paquete entra en una cadena se compara (en orden) con cada regla de la cadena.

Cuando una regla especifica que hacer con el paquete, se realiza la acción y se deja de recorrer la cadena.

Manos a la obra...

Además de iptables tenemos...

- M0n0wall: completo firewall sin nada que envidiar a soluciones propietarias. Está basado en FreeBSD
- Shorewall: “iptables made easy”
- Packet Filter: el “iptables” de OpenBSD
- Exec Shield: desarrollado por RedHat, agrega el bit NX a procesadores x86
- SELinux: desarrollado por la NSA, el 22 de diciembre de 2002 pasó el proyecto a la comunidad OpenSource

¿Preguntas?

<http://boris.penguin.cl>
borisq22@gmail.com