📁 LPIC-2 / 🔒 Examen 212 - Seguridad del Sistema

212.2 Gestionar un firewall

Teoría

Un firewall es un sistema de seguridad que monitoriza y controla el tráfico de red entrante y saliente basándose en un conjunto predefinido de reglas de seguridad. Su propósito principal es establecer una barrera entre una red interna de confianza y redes externas no confiables (como Internet) para protegerse contra accesos no autorizados.

Netfilter: Es el framework de packet filtering del kernel de Linux. Proporciona "hooks" (puntos de enganche) en la pila de red donde el kernel puede interceptar paquetes y aplicar reglas definidas en el espacio de usuario.

Herramientas de Espacio de Usuario para Configurar Netfilter:

Hay varias herramientas para interactuar con Netfilter, con diferencias significativas entre distribuciones:

1. **`iptables`:**
    - La utilidad de línea de comandos "tradicional" para configurar Netfilter. Interactúa directamente con las estructuras de Netfilter (tablas y cadenas). Aún es muy relevante, ya que otras herramientas pueden usarlo como backend o exportar reglas en su formato.
    - **Conceptos Clave:**
        - **Tablas:** Grupos de cadenas por función (ej: `filter` - filtrado de paquetes, `nat` - Network Address Translation, `mangle` - modificación de encabezados, `raw` - procesamiento temprano). La tabla `filter` es la más común para filtrar tráfico.
        - **Cadenas (Chains):** Puntos predefinidos en el flujo de paquetes donde se aplican reglas (ej: `INPUT` - para paquetes dirigidos al propio sistema, `OUTPUT` - para paquetes originados por el sistema, `FORWARD` - para paquetes que pasan a través del sistema - como un router, `PREROUTING`, `POSTROUTING` - para NAT).
        - **Reglas (Rules):** Condiciones (`matches`) y una acción (`target`) a tomar si las condiciones se cumplen.
    - **Sintaxis de Regla Básica:** `iptables -A <cadena> [condiciones] -j <acción>`
        - `-A <cadena>`: Añadir una regla al final de una cadena (ej: `INPUT`).
        - `-I <cadena> [<número>]`: Insertar una regla al principio o en una posición específica.
        - `-D <cadena> [<número> | regla]`: Eliminar una regla.

    - `-L [<cadena>]`: Listar reglas (opción `-v` para verboso, `-n` para IPs/puertos numéricos).
    - `-F [<cadena>]`: Eliminar todas las reglas de una cadena.
    - `-P <cadena> <política>`: Establecer la política por defecto de una cadena (`ACCEPT`, `DROP`, `REJECT`).
    - **Condiciones:** `-i <interfaz_in>`, `-o <interfaz_out>`, `-p <protocolo>` (tcp, udp, icmp, etc.), `-s <ip/red_origen>`, `-d <ip/red_destino>`, `--sport <puerto_origen>`, `--dport <puerto_destino>`, `-m state --state <estados>` (NEW, ESTABLISHED, RELATED, INVALID).
    - **Acciones (`-j <target>`):** `ACCEPT` (permite), `DROP` (descarta silenciosamente), `REJECT` (descarta y envía un mensaje de error al origen), `LOG` (registra el paquete antes de procesarlo), saltar a otra cadena definida por el usuario, NAT targets (`SNAT`, `DNAT`, `MASQUERADE` en la tabla `nat`).
  - **Persistencia:** Las reglas añadidas con `iptables` se pierden al reiniciar por defecto. Se necesita guardar/restaurar usando herramientas específicas de distribución (ej: `iptables-save > /etc/sysconfig/iptables`, `iptables-restore < /etc/sysconfig/iptables` o servicios `netfilter-persistent`/`iptables-persistent` en Debian/Ubuntu).
2. **`firewalld`:**
    - Un demonio de firewall dinámico que gestiona reglas Netfilter. Es el **sistema de firewall por defecto en Red Hat/CentOS/Fedora** y otras distribuciones. Utiliza un modelo de "zonas" y "servicios" para simplificar la gestión.
    - **Conceptos Clave:**
        - **Zonas:** Definen diferentes niveles de confianza para diferentes interfaces de red o fuentes. Puedes asignar interfaces a zonas (ej: `public`, `external`, `internal`, `trusted`). Cada zona tiene su propio conjunto de reglas.
        - **Servicios:** Conjuntos predefinidos de puertos y protocolos para aplicaciones comunes (ej: `ssh`, `http`, `https`, `samba`, `nfs`). Puedes permitir un servicio en una zona.
        - **Comando:** `firewall-cmd` es la utilidad de línea de comandos para interactuar con el demonio `firewalld`.
    - **Configuración:** `firewall-cmd --list-all --zone=<zona>` (ver configuración de una zona), `--add-service=<>`, `--remove-service=<>`, `--add-port=<>`, `--remove-port=<>`, `--add-masquerade`, `--remove-masquerade`. La opción `--permanent` hace que los cambios persistan después del reinicio; requiere `firewall-cmd --reload` para aplicar los cambios permanentes en tiempo de ejecución.
    - **Reglas Directas/Ricas:** Permite añadir reglas más detalladas o de bajo nivel cuando las zonas/servicios no son suficientes.

3. **`ufw` (Uncomplicated Firewall):**
    - Una interfaz de firewall más sencilla, diseñada para simplificar tareas comunes de firewalling. Es el **sistema de firewall por defecto en Debian/Ubuntu**. También gestiona Netfilter.
    - **Conceptos Clave:** Centrado en reglas de "permitir" o "denegar" para puertos, protocolos o direcciones IP.
    - **Comando:** `ufw`.
    - **Configuración:** `ufw enable` (habilita el firewall), `ufw disable` (deshabilita), `ufw status` (ver estado y reglas), `ufw allow <puerto>/<protocolo>`, `ufw deny <puerto>/<protocolo>`, `ufw allow from <ip/red>`, `ufw allow from <ip> to any port <puerto>`. `ufw reload` recarga la configuración. Las reglas son persistentes por defecto.
4. **`nftables`:** El nuevo sistema de línea de comandos para Netfilter. Utiliza su propia sintaxis (`nft`). Aunque reemplaza a `iptables` a bajo nivel, muchas herramientas y documentación aún se refieren a `iptables`. `firewalld` y `ufw` en sistemas modernos a menudo usan `nftables` como backend. LPIC-2 puede evaluar la comprensión de `iptables` o `firewalld`/`ufw`.

Conceptos de Seguridad de Firewall:

- **Política por Defecto:** La política por defecto de una cadena es crucial. Una política de `DROP` o `REJECT` en `INPUT` y `FORWARD` (si actúas como router) es más segura que `ACCEPT` ("deny all by default").
- **Stateful Inspection:** La mayoría de los firewalls modernos son stateful. Utilizan el módulo `state` (en iptables) para permitir automáticamente el tráfico de retorno para conexiones ya establecidas (`--state ESTABLISHED,RELATED`). Esto simplifica las reglas (solo necesitas permitir el tráfico entrante inicial).
- **Orden de las Reglas:** En `iptables`, el orden de las reglas importa, ya que se procesan secuencialmente. La primera regla que coincide determina la acción. En `firewalld`/`ufw`, el orden se gestiona internamente, aunque las reglas directas/ricas pueden ser más complejas.
- **Troubleshooting:** Usar las herramientas para listar reglas (`iptables -L`, `firewall-cmd --list-all`, `ufw status`), verificar contadores de paquetes (`iptables -vL`), y revisar logs del sistema (si se usa el target `LOG` en iptables o si el firewall registra eventos) para ver qué paquetes se bloquean y por qué.