🔐 Examen 110 - Seguridad

110.2 Configurar la seguridad del host

Teoría

Asegurar un host implica configurar el sistema operativo y los servicios para minimizar las vulnerabilidades y protegerlo de ataques. Los puntos clave incluyen el control de acceso, la gestión de servicios en ejecución y la configuración de firewalls locales.

1. **Reducir la Superficie de Ataque:**
    - Desinstala o deshabilita cualquier servicio que no sea estrictamente necesario (servidores web, FTP, bases de datos si no se usan). Cada servicio en ejecución es un posible punto de entrada para un atacante.
    - Cierra los puertos de red que no estén en uso (relacionado con los firewalls).
2. **Seguridad del Acceso Remoto (SSH):**
    - SSH (Secure Shell) es el método estándar y seguro para acceder a sistemas Linux de forma remota a través de una red no confiable. Proporciona una conexión cifrada.
    - El servidor SSH se llama `sshd` (SSH daemon).
    - **Archivo de Configuración de `sshd`:** `/etc/ssh/sshd_config`. Este archivo controla el comportamiento del servidor SSH (puerto de escucha, métodos de autenticación permitidos, restricciones de usuario/grupo, etc.).
        - **Ubicación del archivo:** `/etc/ssh/sshd_config` en **ambas ramas (Debian y Red Hat)**.
    - **Directivas de Seguridad Clave en `sshd_config`:**
        - `Port <número>`: Especifica el puerto en el que `sshd` escucha (por defecto 22). Cambiarlo puede dificultar escaneos automatizados, pero no es una medida de seguridad robusta por sí sola ("security through obscurity").
        - `PermitRootLogin yes|no|prohibit-password`: Controla si se permite el inicio de sesión directo como usuario root vía SSH. **Se recomienda `no` o `prohibit-password`** y usar `sudo` después de iniciar sesión como un usuario normal.
        - `PasswordAuthentication yes|no`: Controla si se permite la autenticación basada en contraseña. **Se recomienda `no`** si se utiliza autenticación basada en clave SSH.
        - `AllowUsers usuario1 usuario2`: Permite solo a los usuarios listados iniciar sesión vía SSH.
        - `AllowGroups grupo1 grupo2`: Permite solo a los miembros de los grupos listados iniciar sesión vía SSH.
        - `PubkeyAuthentication yes`: Permite la autenticación basada en clave pública (es la opción por defecto y recomendada).

- **Autenticación Basada en Clave SSH:** Un método más seguro que las contraseñas. Implica un par de claves: una clave privada (se guarda en el cliente y se protege con una frase de paso) y una clave pública (se copia al servidor en el archivo `~/.ssh/authorized_keys` del usuario). El servidor utiliza la clave pública para verificar que el intento de conexión proviene del poseedor de la clave privada correspondiente.
    - `ssh-keygen`: Genera un par de claves SSH.
    - `ssh-copy-id usuario@servidor`: Copia la clave pública al servidor en el archivo `~/.ssh/authorized_keys` del usuario.

3. **Firewall Basado en el Host:**

- Un firewall local filtra los paquetes de red que entran o salen de tu host, basándose en reglas predefinidas (puerto de origen/destino, dirección IP, protocolo). Es una capa de defensa esencial.
- **`iptables`:** La herramienta de línea de comandos tradicional en Linux para configurar el framework de filtrado de paquetes del kernel (netfilter). Trabaja con tablas (filter, nat, mangle), cadenas (INPUT, OUTPUT, FORWARD), y reglas. Es muy potente pero sintácticamente complejo. Las reglas configuradas con `iptables` son temporales a menos que se guarden y restauren al arrancar (por scripts específicos o servicios como `iptables-persistent`).
- **`nftables`:** El sucesor de `iptables`, con una sintaxis más flexible y unificada. Está reemplazando gradualmente a `iptables` en las distribuciones modernas, aunque la interfaz `iptables` aún puede usarse como front-end para `nftables` (capa de compatibilidad).
- **Firewalls de Alto Nivel:** Herramientas que simplifican la gestión de `iptables` o `nftables` mediante una interfaz más amigable.
    - **`firewalld`:** Un demonio de firewall dinámico, por defecto en RHEL/CentOS/Fedora. Permite gestionar reglas de firewall utilizando "zonas" y "servicios", y aplicar cambios sin romper las conexiones existentes. Herramienta de línea de comandos: `firewall-cmd`.
    - **`ufw` (Uncomplicated Firewall):** Una interfaz más simple para `iptables`/`nftables`, común en Ubuntu (parte de Debian, pero más prominentemente usado en Ubuntu por defecto). Es más fácil para configurar reglas básicas. Herramienta de línea de comandos: `ufw`.
- **Diferencias Debian vs. Red Hat (Firewall por Defecto):**
    - **Rama Debian/Ubuntu:** A menudo se usa `ufw` en entornos de escritorio o servidores simples. Los servidores pueden usar `iptables` o `nftables` configurados directamente o con scripts.
    - **Rama Red Hat/CentOS/Fedora:** `firewalld` es el servicio de firewall por defecto.

- **Reglas Básicas de Firewall:** Se basan en permitir (ACCEPT) o denegar (DROP/REJECT) el tráfico que coincide con ciertos criterios (protocolo TCP/UDP, puerto de destino, dirección IP de origen). La política por defecto para las cadenas INPUT/FORWARD a menudo es DENY/DROP.

4. **Deshabilitar Servicios Innecesarios:**
    - Usa `systemctl status <nombre_servicio>` para verificar el estado de un servicio.
    - Usa `sudo systemctl disable <nombre_servicio>` para que no inicie automáticamente al arrancar.
    - Usa `sudo systemctl stop <nombre_servicio>` para detenerlo inmediatamente.
    - Identifica los servicios en ejecución con `systemctl list-units --type=service --state=running`.