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

*Nota: Estos ejercicios implican explorar configuraciones sensibles y el estado de seguridad del sistema. Realízalos **SIEMPRE en una VM de prueba dedicada**. Necesitarás privilegios de superusuario (`sudo`).*

Ejercicio 12.3.1: Identificando Servicios de Red Activos y sus Puertos

- **Objetivo:** Ver qué servicios están escuchando en puertos de red.
- **Requisitos:** Acceso a la línea de comandos. Privilegios de superusuario (`sudo`).
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Usa `ss` (recomendado en sistemas modernos):** Ejecuta `sudo ss -tulnp`.
        - `-t`: TCP sockets.
        - `-u`: UDP sockets.
        - `-l`: Listening sockets.
        - `-n`: Numérico (no resolver nombres).
        - `-p`: Muestra el programa/PID asociado.
    3. **Usa `netstat` (alternativa más antigua):** Ejecuta `sudo netstat -tulnp`.
    4. **Observa la salida:** Identifica los servicios listados (programa y PID), los puertos (`Local Address`) y el protocolo (`State`). Anota los servicios que no reconoces o que no esperas que estén escuchando en la red.

Ejercicio 12.3.2: (Conceptual) Deshabilitando Servicios No Necesarios

- **Objetivo:** Entender cómo evitar que un servicio se inicie al arrancar.
- *Requisitos:* Identificar un servicio no necesario (Ej. 12.3.1). Privilegios de superusuario (`sudo`). **VM de prueba.**
- **Desarrollo Paso a Paso (Conceptual):**
    1. Abre una terminal.
    2. **Identifica el nombre del servicio systemd (generalmente el nombre del programa):** Ejecuta `systemctl status <nombre_programa_del_servicio>`. Anota el nombre del servicio (ej: `telnetd.service`).
    3. **Deshabilita el servicio para que no inicie al arrancar:** Ejecuta `sudo systemctl disable <nombre_servicio>`.
    4. **Detén el servicio si está corriendo:** Ejecuta `sudo systemctl stop <nombre_servicio>`.
    5. **Verifica el estado:** `systemctl status <nombre_servicio>`.

Ejercicio 12.3.3: Verificando el Usuario Bajo el que Corre un Servicio

- **Objetivo:** Identificar con qué permisos se ejecuta un demonio.

- **Requisitos:** Un servicio corriendo (Ej. 12.3.1). Acceso a la línea de comandos. Privilegios de superusuario (`sudo`).
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Identifica el PID del servicio:** Usa `ss -tulnp` o `netstat -tulnp` para encontrar el PID del servicio que te interesa.
    3. **Usa `ps` para ver los detalles del proceso, incluyendo el usuario:** Ejecuta `ps aux | grep <PID_del_servicio>`. La primera columna muestra el usuario.
    4. **Alternativamente, busca el servicio por nombre:** Ejecuta `ps aux | grep <nombre_programa_del_servicio>`.
    5. **(Contexto):** Idealmente, el usuario no es `root`.

Ejercicio 12.3.4: (Conceptual) Explorando Configuraciones de Seguridad de Servicios Comunes (sshd)

- **Objetivo:** Identificar directivas de seguridad en un archivo de configuración de servicio común.
- *Requisitos:* Servidor SSH instalado (`sshd`). Acceso a la línea de comandos. Privilegios de superusuario (`sudo`).
- **Desarrollo Paso a Paso (Conceptual):**
    1. Abre una terminal.
    2. **Visualiza el archivo de configuración de SSH:** Ejecuta `sudo less /etc/ssh/sshd_config`.
    3. **Busca directivas de seguridad clave:**
        - `Port`: ¿Está en el puerto 22?
        - `PermitRootLogin`: ¿Está en `no`?
        - `PasswordAuthentication`: ¿Está en `no` (si usas claves)?
        - `AllowUsers`, `AllowGroups`: ¿Hay restricciones de quién puede loguearse?
        - `PubkeyAuthentication`: ¿Está en `yes`?
        - `Protocol`: ¿Está en `2`?
    4. **(Contexto):** Modificar estas directivas y recargar el servicio SSH (`sudo systemctl reload sshd`) es crucial para asegurar el acceso remoto.

Ejercicio 12.3.5: Verificando el Estado de SELinux o AppArmor

- **Objetivo:** Determinar si un framework MAC está habilitado y en qué modo.
- **Requisitos:** Acceso a la línea de comandos. Privilegios de superusuario (`sudo`).
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Si usas SELinux (Red Hat/CentOS/Fedora):**
        - Ejecuta `sestatus`. Muestra si SELinux está habilitado, en qué modo (`enforcing`, `permissive`, `disabled`) y qué política se usa.

    - Ejecuta `getenforce`. Muestra solo el modo (`Enforcing`, `Permissive`, `Disabled`).
    - **(Contexto):** `Enforcing` significa que SELinux aplica la política y bloquea acciones. `Permissive` registra las violaciones pero no las bloquea. `Disabled` está apagado.
3. **Si usas AppArmor (Debian/Ubuntu):**
    - Ejecuta `sudo aa-status`. Muestra si AppArmor está corriendo, cuántos perfiles están en modo `enforce` (aplicando reglas) y cuántos en modo `complain` (solo registrando).
4. **(Contexto):** Un sistema con un framework MAC habilitado y en modo de aplicación es más seguro, pero requiere que las políticas/perfiles estén correctamente configurados para que los servicios funcionen.

Ejercicio 12.3.6: (Conceptual) La Importancia de las Actualizaciones de Servicios

- **Objetivo:** Entender por qué mantener los servicios actualizados es una medida de seguridad fundamental.
- **Requisitos:** Comprensión de vulnerabilidades de software.
- **Desarrollo Paso a Paso (Conceptual):**
    1. Abre una terminal.
    2. **Imagina que un investigador de seguridad encuentra una vulnerabilidad** en una versión antigua de OpenSSH que permite a un atacante ejecutar código remotamente.
    3. **Los desarrolladores de OpenSSH publican una nueva versión** con un parche para corregir la vulnerabilidad.
    4. **Los mantenedores de tu distribución incluyen la nueva versión** en sus repositorios de software.
    5. **Si no actualizas tu sistema,** tu servidor SSH sigue siendo vulnerable. Un atacante que conozca la vulnerabilidad podría explotarla para obtener acceso a tu sistema, incluso si tienes un firewall que permite el tráfico SSH.
    6. **Al ejecutar `sudo apt update && sudo apt upgrade` o `sudo dnf upgrade`,** descargas e instalas la versión parcheada de OpenSSH, cerrando esa puerta trasera de seguridad.
    7. **(Conclusión):** Las actualizaciones son la forma más efectiva de protegerse contra vulnerabilidades de software conocidas en los servicios que corren en tu sistema.