📁 LPIC-2 / 🔒 Tema 212 - La seguridad del sistema - Ejercicios

*Nota: Estos ejercicios cubren una variedad de tareas de seguridad generales. Realízalos **SIEMPRE en una VM de prueba dedicada**. Ten cuidado al modificar archivos de configuración de seguridad críticos. Necesitarás privilegios de superusuario (`sudo`).*

Ejercicio 12.4.1: Verificando y Aplicando Actualizaciones del Sistema

- **Objetivo:** Usar el gestor de paquetes para buscar e instalar actualizaciones de seguridad.
- **Requisitos:** Acceso a la línea de comandos. Privilegios de superusuario (`sudo`). Conexión a internet.
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Actualiza la lista de paquetes y verifica si hay actualizaciones (Diferencias):**
        - Debian/Ubuntu: `sudo apt update && apt list --upgradable`
        - Red Hat/CentOS/Fedora: `sudo dnf check-update`
    3. **Simula una actualización (para ver qué se actualizaría sin instalar):**
        - Debian/Ubuntu: `sudo apt upgrade --simulate`
        - Red Hat/Centos/Fedora: `sudo dnf upgrade --simulate`
    4. **Aplica las actualizaciones (requiere confirmación):**
        - Debian/Ubuntu: `sudo apt upgrade`
        - Red Hat/CentOS/Fedora: `sudo dnf upgrade`

Ejercicio 12.4.2: Verificando Permisos en Archivos Críticos

- **Objetivo:** Asegurarse de que los archivos de configuración de usuarios y sudo tienen permisos restrictivos.
- **Requisitos:** Acceso a la línea de comandos.
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Verifica permisos de `/etc/passwd` (información de usuarios, legible por todos):** Ejecuta `ls -l /etc/passwd`. Debería tener permisos como `-rw-r--r--` (644) o más restrictivos.
    3. **Verifica permisos de `/etc/shadow` (contraseñas hasheadas, solo legible por root):** Ejecuta `ls -l /etc/shadow`. Debería tener permisos como `-rw-------` (600).
    4. **Verifica permisos de `/etc/group` (información de grupos, legible por todos):** Ejecuta `ls -l /etc/group`. Debería tener permisos como `-rw-r--r--` (644) o más restrictivos.
    5. **Verifica permisos de `/etc/sudoers` (configuración de sudo, solo legible por root):** Ejecuta `ls -l /etc/sudoers`. Debería tener permisos como `-r--r-----` (440) o `-r--------` (400).

6. **Verifica permisos del directorio `/etc/sudoers.d/` (requiere sudo):** Ejecuta `sudo ls -ld /etc/sudoers.d/`. Debería tener permisos como `drwxr-xr-x` (755) y propiedad de root.

Ejercicio 12.4.3: Explorando la Configuración de `sudo`

- **Objetivo:** Usar `visudo` y ver el contenido de `/etc/sudoers`.
- **Requisitos:** Acceso a la línea de comandos. Privilegios de superusuario (`sudo`).
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Abre el archivo `/etc/sudoers` usando `visudo` (requiere sudo):** Ejecuta `sudo visudo`. Esto abre el archivo en tu editor por defecto. **¡Nunca edites `/etc/sudoers` directamente con `vi` o `nano`!**
    3. **Observa la sintaxis:** Busca ejemplos de reglas (`root ALL=(ALL) ALL`), alias (User_Alias, Cmnd_Alias, Host_Alias). Busca la línea que incluye archivos de `/etc/sudoers.d/` (ej: `#includedir /etc/sudoers.d`).
    4. **Sin guardar, sal del editor** (ej: en vi, `:q!`).
    5. **Explora el contenido del directorio `sudoers.d` (requiere sudo):** Ejecuta `sudo ls -l /etc/sudoers.d/`. Aquí encontrarás archivos de configuración para sudo añadidas por paquetes (ej: para `wheel` group, para `cloud-init`, etc.). Estos archivos deben tener permisos restrictivos (ej: 440).

Ejercicio 12.4.4: (Conceptual) Concediendo Acceso Limitado con `sudo`

- **Objetivo:** Entender cómo añadir una regla simple en `/etc/sudoers.d/`.
- *Requisitos:* Privilegios de superusuario (`sudo`). Usuario de prueba no-root. **VM de prueba.**
- **Desarrollo Paso a Paso (Conceptual):**
    1. Abre una terminal.
    2. **Crea un archivo nuevo en `sudoers.d` (requiere sudo):** Ejecuta `sudo visudo /etc/sudoers.d/myuser_rules`.
    3. **Añade una regla para permitir a un usuario ejecutar un comando específico como root (adapta el usuario y el comando):**

       ```
       myuser ALL=(root) /sbin/shutdown -h now
       ```

        - Esto permite al usuario `myuser` ejecutar `/sbin/shutdown -h now` como root desde cualquier host.
    4. **Guarda y sal de `visudo`.**
    5. **Cambia al usuario de prueba (`su - myuser`)** o abre una sesión como ese usuario.
    6. **Intenta ejecutar el comando permitido con `sudo`:** Ejecuta `sudo /sbin/shutdown -h now`. Debería pedir la contraseña del usuario `myuser` (no la de root) y ejecutar el comando.

7. **Intenta ejecutar un comando NO permitido:** Ejecuta `sudo /bin/ls`. Debería dar un error de permiso denegado por sudo.
8. **(Limpieza en VM):** Elimina el archivo `/etc/sudoers.d/myuser_rules` o comenta la línea si no quieres que la regla persista.

Ejercicio 12.4.5: Revisando Logs de Autenticación (Intentos Fallidos de Login)

- **Objetivo:** Buscar evidencia de intentos de acceso no autorizados en los logs.
- **Requisitos:** Acceso a la línea de comandos. Privilegios de superusuario (`sudo`) para acceder a logs.
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Busca intentos fallidos de login (Diferencias en logs):**
        - Debian/Ubuntu: Ejecuta `sudo grep "Failed password" /var/log/auth.log`.
        - Red Hat/CentOS/Fedora: Ejecuta `sudo grep "Failed password" /var/log/secure`.
        - Usando journalctl: `journalctl -u sshd.service | grep "Failed password"` o `journalctl -f | grep "Failed password"`.
    3. **Observa la salida:** Verás registros de intentos de login SSH fallidos, incluyendo el usuario intentado y la IP de origen. Un gran número de intentos fallidos desde una IP puede indicar un ataque de fuerza bruta.
    4. **(Contexto):** Herramientas como Fail2ban (a menudo cubiertas en seguridad de servicios o detección de intrusiones, pero relevantes aquí) automatizan el bloqueo de IPs que muestran patrones de ataque (ej: múltiples fallos de login) configurando reglas temporales de firewall.

Ejercicio 12.4.6: (Conceptual) Programando una Tarea de Seguridad (Cron)

- **Objetivo:** Entender cómo automatizar una tarea de seguridad usando cron.
- *Requisitos:* Acceso a la línea de comandos. Un comando de seguridad a programar (ej: `sudo aide --check`). **VM de prueba.**
- **Desarrollo Paso a Paso (Conceptual):**
    1. Abre una terminal.
    2. **Edita la crontab de root (requiere sudo):** Ejecuta `sudo crontab -e`. Esto abre el archivo de configuración de cron para el usuario root.
    3. **Añade una línea para programar la tarea (adapta el comando y la frecuencia):** Fragmento de código

```
# Ejecutar verificacion de AIDE todos los dias a las 3:30 AM
30 3 * * * /usr/bin/aide --check > /var/log/aide_check_$(date +\%Y\
%m\%d).log 2>&1
# O enviar por correo a root (si el sistema envia correos)
# 30 3 * * * /usr/bin/aide --check | mail -s "Daily AIDE Check" root
```

- Adapta la ruta a `aide` si es necesario (`which aide`).
- La redirección de salida `> ... 2>&1` guarda la salida y errores en un archivo de log con fecha.
- `%` debe escaparse con `\` en archivos crontab.

4. **Guarda y sal del editor de `crontab`.**
5. **Verifica que la línea fue añadida:** `sudo crontab -l`.
6. **(Contexto):** El resultado de la tarea programada debe ser revisado regularmente (mirando el archivo de log o la bandeja de entrada de correo de root).