Auditorías básicas en Linux

Objetivo

- Revisar la **configuración de usuarios y permisos.**
- Analizar **logs y servicios activos.**
- Verificar **seguridad básica** y cumplimiento de buenas prácticas.

1. Auditoría de usuarios y permisos

Comandos útiles:

```
# Listar todos los usuarios
cat /etc/passwd

# Listar grupos
cat /etc/group

# Usuarios con privilegios sudo
getent group sudo
```

Revisión de permisos de archivos críticos:

```
# Ver permisos de archivos importantes
ls -l /etc/passwd /etc/shadow /etc/sudoers

# Buscar archivos con permisos peligrosos
find / -perm -4000 -type f 2>/dev/null  # archivos SUID
find / -perm -2000 -type f 2>/dev/null  # archivos SGID
```

Estos archivos son peligrosos porque tienen activados los bits **SUID (Set User ID)** o **SGID (Set Group ID).**

1. Ejecución con Privilegios Elevados

- **SUID (Set User ID - Permiso 4000):** Cuando un archivo ejecutable tiene el bit SUID activado, el proceso se ejecuta con los permisos del **dueño del archivo**, en lugar de los permisos del usuario que lo está ejecutando.

    - **Peligro Principal:** Si el dueño del archivo es root (el superusuario), cualquiera que ejecute ese programa lo hará con los permisos de root. Si un atacante encuentra una vulnerabilidad (como un *buffer overflow*) en ese programa SUID, o si el programa permite la ejecución de comandos de *shell*, puede escalar sus privilegios al nivel de root.
    - **Ejemplo Común:** El comando passwd debe leer y escribir en el archivo de contraseñas (/etc/shadow), que solo es accesible por root. Por lo tanto, passwd tiene el bit SUID para que un usuario normal pueda ejecutarlo y cambiar su propia contraseña, a pesar de los permisos restringidos del archivo de destino.

- **SGID (Set Group ID - Permiso 2000):** De manera similar, si un ejecutable tiene el bit SGID, el proceso se ejecuta con los permisos del **grupo propietario del archivo**.
    - **Peligro Principal:** Aunque no es tan crítico como el SUID de root, aún permite que un usuario acceda a archivos y recursos que normalmente están restringidos a un grupo específico. Un uso indebido podría permitir el acceso no autorizado o la manipulación de datos sensibles del grupo.

2. Escalada de Privilegios

La búsqueda de archivos con estos permisos (find / -perm -4000 y find / -perm -2000) es una técnica estándar utilizada por los atacantes (y los probadores de penetración) como parte de un proceso de **escalada de privilegios**.

1. El atacante obtiene acceso como un **usuario de bajo privilegio**.
2. Busca binarios SUID/SGID.
3. Si encuentran un binario SUID propiedad de root que pueden manipular o que tiene fallos de seguridad (como los programas de *shell* que no se han escrito de forma segura), pueden explotarlo para ejecutar comandos como root, tomando control total del sistema.

El permiso SUID se representa con una **s** en lugar de la x para el dueño del archivo. El SGID se representa con una **s** en lugar de la x para el grupo.

- **Ejemplo SUID (propiedad de root):** -rwsr-xr-x (4755 en octal). La s está en la posición de ejecución del dueño.
- **Ejemplo SGID:** -rwxr-sr-x (2755 en octal). La s está en la posición de ejecución del grupo.

En resumen, son peligrosos porque permiten que un programa se ejecute con una **identidad (y, por lo tanto, con privilegios) diferente y usualmente superior** a la del usuario que lo invocó, lo que es un vector primario para la **escalada de privilegios**.

2. Auditoría de servicios y procesos

Listar servicios activos

```
# Con systemd
systemctl list-units --type=service --state=running

# Ver puertos abiertos
ss -tuln
```

Revisión de servicios habilitados en arranque

```
systemctl list-unit-files --type=service | grep enabled
```

Revisar los servicios habilitados (enabled) es fundamental para la seguridad:

- **Reducción de la Superficie de Ataque:** Cada servicio habilitado y en ejecución potencialmente **aumenta la superficie de ataque** del sistema. Deshabilitar los servicios innecesarios reduce el número de posibles puntos de entrada que un atacante podría explotar.
- **Principio del Mínimo Privilegio:** Solo deben ejecutarse los servicios estrictamente necesarios. Esta es la aplicación del **principio del mínimo privilegio** a los servicios.
- **Identificación de Servicios no Autorizados:** Esta verificación ayuda a identificar cualquier servicio que pueda haber sido habilitado por *malware*, una intrusión o una configuración accidental, asegurando que solo los servicios **autorizados** se inicien al arrancar el sistema.

Comandos Útiles para la Revisión

- **Listar todos los servicios *en ejecución* (running) (suelen ser la mayor preocupación de seguridad):**

```
systemctl list-units --type=service | grep running
```

- **Listar todos los servicios que *no están* habilitados (enabled) (para comparación):**

```
systemctl list-unit-files --type=service | grep -v enabled
```

Una vez que tengas la lista de **servicios habilitados**, debes revisar cada uno para confirmar que es necesario para la funcionalidad de tu sistema. Si un servicio no es esencial, generalmente puedes deshabilitarlo usando:

```
sudo systemctl disable <nombre-del-servicio.service>
```

3. Auditoría de logs

Revisar intentos de acceso

```
# Últimos logins
last

# Intentos fallidos de SSH
sudo grep "Failed password" /var/log/auth.log
```

Analizar eventos críticos

```
# Ver errores del sistema
sudo journalctl -p err

# Supervisar logs en tiempo real
sudo tail -f /var/log/syslog
```

4. Auditoría de seguridad básica

Firewall

```
# Ver reglas activas (UFW)
sudo ufw status verbose

# Ver reglas con iptables
sudo iptables -L -v -n
```

La principal diferencia entre **UFW** e **iptables** es que **UFW es una interfaz o "frontend" simplificada y amigable para el usuario, mientras que iptables es la herramienta subyacente de bajo nivel que realmente gestiona las reglas del firewall en el kernel de Linux.**

Piensa en la relación así: **UFW es el control remoto, e iptables es el televisor.** UFW te permite cambiar de canal (configurar reglas) con botones simples, pero es iptables quien hace el trabajo pesado en el interior.

iptables: El Núcleo del Firewall

Característica	**Descripción**
Nivel	**Bajo Nivel** (Mecanismo del *kernel* de Linux).
Función	Configura el **Netfilter**, el *framework* de filtrado de paquetes del núcleo de Linux.
Sintaxis	**Compleja y detallada.** Requiere comandos largos y precisos para cada regla, especificando tablas, cadenas y objetivos (-A INPUT -p tcp --dport 22 -j ACCEPT).
Control	**Máximo control y flexibilidad.** Ideal para administradores de sistemas avanzados que necesitan configuraciones de firewall muy específicas, NAT o reglas de enmascaramiento complejas.
Uso	Típicamente usado en servidores, *routers* o entornos donde se requiere una manipulación granular de los paquetes.

UFW (Uncomplicated Firewall): La Interfaz Sencilla

Característica	Descripción
Nivel	**Alto Nivel** (Utilidad de espacio de usuario).
Función	Es un programa diseñado para ser **fácil de usar** (de ahí su nombre: **U**ncomplicated **F**irewall) que traduce comandos simples en reglas complejas de iptables.
Sintaxis	**Simple e intuitiva.** Utiliza comandos cortos y legibles (sudo ufw allow 22/tcp).
Control	**Facilidad de uso y rapidez.** Ideal para usuarios que necesitan un firewall funcional, rápido y seguro sin la curva de aprendizaje de iptables.
Uso	Es el *frontend* de firewall recomendado y a menudo preinstalado en distribuciones basadas en Debian/Ubuntu.

Resumen de la Diferencia

	UFW	IPTABLES
Propósito	Simplificar la gestión del firewall.	Implementar el filtrado de paquetes en el *kernel*.
Relación	Es un **frontend** o capa superior.	Es el **backend** o herramienta central.

	UFW	**IPTABLES**
Sintaxis	Simple (ufw allow 80).	Compleja (iptables -A INPUT -p tcp --dport 80 -j ACCEPT).
Uso Ideal	Seguridad rápida, administración diaria, usuarios principiantes/intermedios.	Configuraciones avanzadas, *scripts* personalizados, administración a nivel de *kernel*.

En esencia, si usas **UFW**, sigues usando la funcionalidad de **iptables** (o su sucesor, **nftables**) sin tener que aprender su sintaxis compleja.

Usuarios con shell no estándar

```
awk -F: '($7 !~ "/bin/bash|/sbin/nologin|/bin/false/") {print $1,$7}' /etc/passwd
```

Se utiliza habitualmente en auditorías de seguridad y administración de sistemas para detectar cuentas que podrían representar un riesgo o que tienen una configuración inesperada.

- **awk -F:**: Le dice a awk que use el carácter de dos puntos (:) como delimitador de campo. El archivo /etc/passwd usa este delimitador para separar la información de cada cuenta.
- **$7**: Representa el séptimo campo de cada línea en /etc/passwd, que es la **ruta del *shell*** asignado al usuario.
- **!~ "/bin/bash|/sbin/nologin|/bin/false/"**: Esta es la condición de filtro. Significa: "Selecciona las líneas donde el campo $7 (**el *shell***) **NO coincida** (!~) con los patrones /bin/bash **o** (|) /sbin/nologin **o** (|) /bin/false".
    - **/bin/bash**: Es el *shell* interactivo más común y seguro para usuarios normales.
    - **/sbin/nologin o /bin/false**: Son *shells* que **impiden el inicio de sesión** interactivo. Son los *shells* estándar y seguros para cuentas de sistema (como sync, daemon,

etc.) que no necesitan acceder al sistema directamente.

- **{print $1,$7}**: Si la condición se cumple, imprime el primer campo (**el nombre de usuario**) y el séptimo campo (**el *shell***).
- **/etc/passwd**: El archivo que contiene la información de las cuentas de usuario.

Salida

Varias cuentas, todas con un *shell* de /bin/false:

sync /bin/sync
dhcpcd /bin/false
... (otras cuentas de sistema)

Interpretación: La salida te está diciendo: "Estas son las cuentas que *no* usan un *shell* interactivo común (/bin/bash), pero tampoco usan los *shells* de 'no login' más típicos (/sbin/nologin o /bin/false)".

- **Nota:** En este caso, la salida mostró que todas las cuentas listadas usan /bin/false (o similar, como /bin/sync para el usuario sync). Esto puede parecer contradictorio, pero la razón es que **/bin/false es un *shell* que termina la sesión inmediatamente después de empezar, por lo que se utiliza para deshabilitar inicios de sesión.**
- **El caso de sync /bin/sync** es similar: sync es un binario que fuerza la escritura de datos en el disco; al usarlo como *shell*, la cuenta intenta ejecutarlo y se cierra, impidiendo una sesión interactiva.

Importancia para la Seguridad

Esta búsqueda es fundamental por las siguientes razones:

1. **Detección de Shells Interactivos Inesperados:** El objetivo principal es encontrar cuentas de sistema (como daemon, sys, o mail) que, por error o malicia, hayan sido configuradas con un *shell* interactivo como /bin/sh o /bin/bash. Estas cuentas no deberían permitir el inicio de sesión y, si lo permiten, representan un gran riesgo de seguridad si se ven comprometidas.
2. **Identificación de Usuarios Maliciosos:** Si un atacante compromete el sistema y añade una cuenta oculta, a menudo le asignará un *shell* interactivo. Este comando puede ayudar a detectar cuentas nuevas o modificadas que tienen

un *shell* válido pero que no son las cuentas de usuario principales esperadas.

3. **Higiene del Sistema:** Ayuda a un administrador a verificar que todas las **cuentas de servicio** (dhcpcd, whoopsie, gdm, etc.) estén configuradas correctamente para **NO permitir un inicio de sesión interactivo**, utilizando mecanismos seguros como /sbin/nologin o /bin/false. La presencia de estas cuentas con /bin/false en tu salida confirma la correcta **segregación de privilegios**.

Revisar configuraciones SSH

```
sudo grep -E "PermitRootLogin|PasswordAuthentication" /etc/ssh/sshd_config
```

Si el comando anterior falla, busca el archivo sshd_config en todo el sistema:
```
sudo find / -name sshd_config 2>/dev/null
```

Esto te devolverá la ruta correcta. Una vez que tengas la ruta, podrás ejecutar tu comando grep.

Si seguimos sin encontrarlo, El mejor paso es forzar al sistema a que te diga dónde está su archivo de configuración:

```
sudo systemctl cat sshd | grep "sshd_config"
```

Si el comando systemctl cat sshd devuelve un error, el servicio **sshd probablemente no está instalado**

```
sudo systemctl status ssh
```

5. Auditoría de integridad de archivos

- Instalar AIDE (Advanced Intrusion Detection Environment):

```
sudo apt install aide -y
sudo aideinit
sudo aide --check
```

- Permite detectar cambios inesperados en archivos del sistema.

El comando sudo aide --check es el paso final para realizar una **auditoría de integridad**, pero la efectividad de AIDE depende de cómo hayas configurado y guardado la **base de datos de referencia (baseline)**.

Guía de Uso de AIDE (Advanced Intrusion Detection Environment)

AIDE funciona creando una **instantánea criptográfica** (un *hash* o *checksum*) de todos los archivos y directorios importantes del sistema. El comando check compara el estado actual con esa instantánea inicial.

Asegúrate de que no haya habido cambios no autorizados en el sistema **antes** de comenzar, ya que AIDE usará el estado actual como la "base de oro" (referencia segura).

1. Instalación de AIDE

Utilizaremos el gestor de paquetes apt de Ubuntu:

1. Actualiza la lista de paquetes disponibles:

sudo apt update

2. Instala el paquete AIDE:

sudo apt install aide

4. Responde 'S' o 'Y' (Sí) cuando se te pregunte para confirmar la instalación.

2. Inicialización de la Base de Datos (Creación de la "Base de Oro")

Una vez instalado, debes generar la primera base de datos. Este archivo es la **firma digital** de todos tus archivos críticos.

Ejecuta el comando de inicialización. Esto puede tardar unos minutos ya que está escaneando todo el sistema.

sudo aide --init

El error **"ERROR: missing configuration"** (error: falta configuración) indica que, aunque AIDE está instalado, no está encontrando automáticamente el archivo de configuración que necesita para saber qué directorios escanear.

Aunque la ruta por defecto es /etc/aide/aide.conf, la forma en que está configurado el entorno de la terminal no permite que AIDE la encuentre por sí mismo.

Solución: Especificar la Configuración

Para solucionar esto, debes indicar explícitamente a AIDE dónde está el archivo de configuración usando el parámetro --config:

```
sudo aide --config /etc/aide/aide.conf --init
```

Duración de la Inicialización de AIDE

Es completamente normal que el comando sudo aide --config /etc/aide/aide.conf --init tarde **mucho rato** (varios minutos, e incluso más dependiendo del tamaño del disco y la velocidad de la máquina virtual).

¿Por qué tarda?

AIDE está haciendo un trabajo intensivo:

1. **Escaneo Profundo:** Está leyendo tu archivo de configuración (/etc/aide/aide.conf) y, según las reglas que contiene, está **escaneando** miles, o incluso millones, de archivos y directorios en todo tu sistema.
2. **Cálculo de Hashes:** Para cada archivo importante, AIDE no solo registra su nombre y permisos, sino que también calcula uno o varios **hashes criptográficos** (como SHA256, por ejemplo). Este proceso de calcular una "huella digital" única para cada archivo es intensivo en recursos de la CPU.
3. **Construcción de la Base de Datos:** Finalmente, está compilando toda esta información en el archivo aide.db.new (la "Base de Oro").

Mientras ves el cursor parpadeando o el sistema parece inactivo, **AIDE está trabajando en segundo plano**, recorriendo el disco y haciendo cálculos.

Mensajes de "Permiso denegado" (WARNING)

Los mensajes de **WARNING: get_file_status: lstat() failed for /run/user/1000/gvfs: Permission denied** (y el de /run/user/1000/doc) **también son normales** en este contexto y no detienen el proceso:

- **Razón:** Esas rutas (/run/user/1000/gvfs, /run/user/1000/doc) son directorios temporales o de montaje relacionados con la sesión gráfica o el entorno de usuario.
- **AIDE como root:** Aunque estás ejecutando AIDE con sudo (como root), estas carpetas a menudo tienen permisos tan restrictivos que incluso root podría encontrar problemas o, simplemente, AIDE puede estar diseñado para ignorar esos puntos de montaje específicos en /run/user/.

Puedes ignorar estas advertencias sin problema para el propósito de una auditoría de seguridad, ya que los archivos del sistema crítico generalmente no residen allí.

```
Number of entries:      333222

---------------------------------------------------
The attributes of the (uncompressed) database(s):
---------------------------------------------------

/var/lib/aide/aide.db.new
 MD5         : hBtPep4Uk5rHXz7iHgaLXQ==
 SHA1        : n0GpZe4uQKb3GVsV/UtkIrZ2EzY=
 SHA256      : dD3uCj90Pst9G+CQzmh44M/3hUXGZRTN
               iHqFcvqc3tM=
 SHA512      : DLSUbs9NPajsbUU+S3mbgToQpvWmR9aj
               rkdkeTKOXiexBDuLVQeud3ZeaEcik8iO
               EkoZJ7RHNxReiR/UNC0N7A==
 RMD160      : 0sgr5f03WASederxNKxVY1tnZX4=
 TIGER       : goPhlenCQSy+rv7quRw7T6dHMN98zhEq
 CRC32       : qIbSYw==
 CRC32B      : Z2qS0g==
 HAVAL       : SA+hYe/D1e9esCHHiKL70QkVVJFrfMos
               Vh57jREOC1M=
 WHIRLPOOL   : v4beo27LYZKgLn10L4YVN9Q5X8gy9GrG
               MQmvCbwkCN/RQN/TgpscSU7MhXRmyk3a
               1rOG/vjyFKAka4UDBjBguQ==
 GOST        : MLVGhD603X875YMTDkrH3o519dwwXC9e
               8olUTKoTouU=


End timestamp: 2025-10-29 10:30:51 +0100 (run time: 21m 7s)
```

Resumen y Siguiente Paso

1. **Espera:** Dale tiempo suficiente al proceso. Cuando termine, volverá a la línea de comandos (feval@feval-VirtualBox:~$).
2. **Activa la Base de Datos:** Una vez que el comando haya terminado, no olvides el paso final para que AIDE pueda hacer futuras verificaciones:

```
sudo mv /var/lib/aide/aide.db.new /var/lib/aide/aide.db
```

Resultado: Este comando crea un archivo llamado aide.db.new (generalmente en la ruta /var/lib/aide/).

3. Activación de la Base de Datos

Para que AIDE sepa qué base de datos usar en las futuras revisiones, debes renombrar el archivo aide.db.new al nombre que espera la herramienta: aide.db.

Mueve/Renombra el archivo recién creado a su ubicación final como lo hemos hecho en el paso anterior.

Tu base de datos de referencia está ahora lista. Podemos confirmarlo

sudo ls -l /var/lib/aide/aide.db

-rw------- 1 root root 78101866 /var/lib/aide/aide.db

4. Ejecutar la Primera Verificación de Integridad

Ahora que AIDE está instalado y tiene una base de datos de referencia, puedes ejecutar una comprobación en cualquier momento para ver si algún archivo ha sido modificado, agregado o eliminado desde la inicialización.

Ejecuta la comprobación (este es el comando que querías usar al principio):

sudo aide --config /etc/aide/aide.conf --check

Resultado esperado: Si no has cambiado nada en tu sistema desde el paso 2, el resultado de aide --check debería ser:

AIDE did not find any differences between the AIDE database and the filesystem. (O el mensaje equivalente en español).

Si ves diferencias, significa que algo cambió **después de que corriste la inicialización** (paso 2), y deberás investigarlo.

¡AIDE ya está instalado y configurado correctamente para comenzar a monitorear la integridad de tu sistema!

```
---------------------------------------------------
The attributes of the (uncompressed) database(s):
---------------------------------------------------

/var/lib/aide/aide.db
 MD5         : hBtPep4Uk5rHXz7iHgaLXQ==
 SHA1        : n0GpZe4uQKb3GVsV/UtkIrZ2EzY=
 SHA256      : dD3uCj90Pst9G+CQzmh44M/3hUXGZRTN
               iHqFcvqc3tM=
 SHA512      : DLSUbs9NPajsbUU+S3mbgToQpvWmR9aj
               rkdkeTKOXiexBDuLVQeud3ZeaEcik8iO
               EkoZJ7RHNxReiR/UNC0N7A==
 RMD160      : 0sgr5f03WASederxNKxVY1tnZX4=
 TIGER       : goPhlenCQSy+rv7quRw7T6dHMN98zhEq
 CRC32       : qIbSYw==
 CRC32B      : Z2qS0g==
 HAVAL       : SA+hYe/D1e9esCHHiKL70QkVVJFrfMos
               Vh57jREOC1M=
 WHIRLPOOL   : v4beo27LYZKgLn10L4YVN9Q5X8gy9GrG
               MQmvCbwkCN/RQN/TgpscSU7MhXRmyk3a
               1rOG/vjyFKAka4UDBjBguQ==
 GOST        : MLVGhD603X875YMTDkrH3o519dwwXC9e
               8olUTKoTouU=


End timestamp: 2025-10-29 11:08:33 +0100 (run time: 29m 4s)
```

Análisis del Reporte de Integridad de AIDE

El informe indica que **AIDE encontró diferencias entre la base de datos y el sistema de archivos** (AIDE found differences between database and filesystem!!). Sin embargo, en un sistema operativo en ejecución, la mayoría de estos cambios son **normales y esperados**, y no representan un ataque.

1. Resumen General

Categoría	Cantidad	Significado
Total de Entradas	333223	El número total de archivos monitoreados.
Added entries	3	Archivos **NUEVOS** que no estaban cuando se hizo la inicialización (10:30).
Removed entries	3	Archivos que **desaparecieron** desde la inicialización.

Categoría	Cantidad	Significado
Changed entries	15	Archivos que **fueron MODIFICADOS** (contenido, permisos o metadatos).

2. Cambios Normales y de Sistema (No Preocupantes)

La gran mayoría de los cambios que ves están relacionados con **archivos temporales, cachés de usuario y archivos de *log***. Esto es completamente normal en un sistema con un entorno gráfico (como VirtualBox con Ubuntu) que está en uso.

Sección del Reporte	Elementos Encontrados	Causa Típica (Riesgo bajo)
Added/Removed	/run/systemd/inhibit/*, /run/user/*	Archivos de **sesión de usuario y sistema (systemd).** Se crean y eliminan constantemente cuando la pantalla se bloquea, una aplicación se abre, o la sesión de usuario cambia.
Removed	/var/lib/aide/aide.db.new	**Excelente noticia:** Este es el archivo temporal que eliminaste al ejecutar sudo mv .... AIDE lo detecta como "removido", lo cual es **lo esperado** y confirma tu acción.
Changed	/home/feval/.cache/*, /home/feval/.config/dconf/*, /run/user/1000/dconf/user	**Cachés de Navegación/Gráficos y Configuraciones de GNOME/dconf.** Estos archivos de base de datos registran el estado de las ventanas, iconos y configuraciones de tu escritorio y cambian constantemente mientras usas la interfaz gráfica.

Sección del Reporte	Elementos Encontrados	Causa Típica (Riesgo bajo)
Changed	/var/log/sysstat/sa29	**Archivos de *log* o estadísticas de sistema.** El tamaño (Size) y el *timestamp* (Mtime/Ctime) cambiaron porque el sistema sigue registrando datos de actividad. Esto es normal para archivos de registro.

3. El Cambio Clave: ¡El Hash Criptográfico!

Observa el cambio detallado en /home/feval/.cache/mesa_shader_cache_db/...:

Atributo	Valor Anterior (Base de Datos)	Valor Nuevo (Actual)
MD5	DOzaL3KZ9nrW6IHSOmVimQ==	I3gPdvSb2xnUA8SAozX4Ag==
SHA256	7L5kAjc...	5qZBP4c...

- **Diferencia:** El **Hash (MD5, SHA1, SHA256, etc.)** es totalmente diferente.
- **Significado:** Esto demuestra la efectividad de AIDE y del *hashing*. Un cambio en la caché de *shaders* (un archivo binario) de solo un byte (o un bit) hace que el hash sea irreconocible, confirmando que el **contenido del archivo fue alterado** (de forma normal, debido al uso de la GPU/gráficos en la VirtualBox).

Conclusión de Seguridad y Siguiente Paso

Interpretación: Los cambios detectados son **ruido del sistema operativo en funcionamiento**, no indicios de una intrusión.

Acción Requerida: Para que AIDE sea útil, debes **actualizar su base de datos** para aceptar los cambios **legítimos** que acabas de revisar.

Actualizar la Base de Datos

Si estás seguro de que todos estos cambios son benignos y provienen del uso normal del sistema, debes actualizar tu "Base de Oro" con el siguiente proceso:

1. **Generar la Nueva Base de Datos (--update):**

```
sudo aide --config /etc/aide/aide.conf --update
```

2. Activar la Nueva Base de Datos:

El comando anterior genera aide.db.new. Debes moverlo para reemplazar el archivo antiguo:

```
sudo mv /var/lib/aide/aide.db.new /var/lib/aide/aide.db
```

Después de esto, si ejecutas sudo aide --check inmediatamente, ¡debería reportar que **no hay diferencias**!

Resumen

1. Instalación de AIDE

Instala el paquete AIDE:

```
sudo apt install aide -y
```

2. Creación de la Base de Datos de Referencia (Baseline)

Este es el paso **más importante**. La primera vez que ejecutas el sistema, generas el archivo que AIDE utilizarás para detectar cualquier cambio futuro. **Debes ejecutar este paso inmediatamente después de instalar el sistema y configurarlo a tu gusto**, antes de cualquier intrusión.

El comando sudo aideinit se encarga de crear el archivo de base de datos inicial:

```
sudo aideinit
```

Lo que hace aideinit:

1. Ejecuta aide --init, que escanea el sistema según las reglas definidas en /etc/aide/aide.conf.
2. Crea el archivo de base de datos inicial, normalmente llamado /var/lib/aide/aide.db.new.
3. Muestra un aviso pidiendo que se mueva ese archivo a la ubicación de referencia.

Mover la Base de Datos de Referencia:

Para que AIDE sepa qué archivo debe usar como "estado original", debe renombrar el archivo *.new al nombre de base de datos activo.

```
sudo mv /var/lib/aide/aide.db.new /var/lib/aide/aide.db
```

Nota de Seguridad: En un entorno de producción, es crucial **mover esta base de datos (aide.db) a un medio de solo lectura** (como un CD/DVD, o un disco USB desconectado después de la copia) o a un servidor seguro. Si un atacante compromete el sistema, podría modificar la base de datos de AIDE para ocultar sus cambios antes de que usted ejecute la comprobación.

3. Ejecución de la Auditoría de Integridad (Check)

Una vez que la base de datos de referencia está en su lugar, ejecuta la auditoría de integridad:

```
sudo aide --check
```

Detalles de la Salida de --check:

El resultado de aide --check categoriza los archivos en tres secciones:

Sección	Descripción	Causa Típica
Added entries:	Archivos y directorios **nuevos** que no estaban en la base de datos original.	Instalación de nuevos paquetes o creación manual de archivos.

Sección	Descripción	Causa Típica
Removed entries:	Archivos y directorios que **existían** en la base de datos original y han sido eliminados.	Desinstalación de paquetes o eliminación manual de archivos de configuración.
Changed entries:	Archivos que **existen**, pero han sido **modificados**.	Una actualización del sistema o la alteración de un binario o archivo de configuración por parte de un atacante.

Atributos de los cambios (Ejemplo):

Si un archivo ha cambiado, AIDE muestra qué atributos se modificaron. Por ejemplo:

/etc/hosts : p+i+n+u+g+s+acl+selinux+xattrs

Esto indica que el archivo /etc/hosts ha cambiado su **p**ermiso, **i**nodo, **n**úmero de enlaces, **u**suario, **g**rupo y **s**ize (tamaño).

4. Actualización de la Base de Datos (Si los Cambios son Legítimos)

Si ejecuta aide --check y los cambios detectados son **legítimos** (por ejemplo, después de una actualización del sistema o una instalación normal), debe actualizar la base de datos para incluir los nuevos estados.

1. **Generar la Nueva Base de Datos:**

   sudo aide --update

   Esto crea un nuevo archivo de base de datos (aide.db.new) que incluye los cambios detectados.

2. **Mover el Nuevo Archivo:**

   sudo mv /var/lib/aide/aide.db.new /var/lib/aide/aide.db

Este paso sobrescribe el archivo antiguo con la nueva instantánea, estableciendo un nuevo punto de referencia para futuras comprobaciones.

6. Auditoría de recursos y rendimiento

```
# Uso de CPU, memoria, disco
top
htop
df -h
free -m

# Procesos que consumen más recursos
ps aux --sort=-%cpu | head -10
ps aux --sort=-%mem | head -10
```

7. Buenas prácticas

1. Mantener **software actualizado**:

```
sudo apt update && sudo apt upgrade -y
```

2. Revisar **cuentas inactivas o huérfanas.**
3. Activar **Fail2Ban** o mecanismos de protección frente a fuerza bruta.
4. Revisar **permisos de archivos críticos.**
5. Mantener copias de seguridad periódicas.

Checklist Práctica de Auditoría Básica en Linux

Área	Comando	Objetivo	Posibles alertas / Qué buscar
Usuarios y grupos	cat /etc/passwd	Listar todos los usuarios del sistema	Usuarios desconocidos, UID 0 sospechosos
	cat /etc/group	Listar grupos	Grupos sospechosos o permisos excesivos
	getent group sudo	Ver usuarios con privilegios sudo	Usuarios no autorizados en sudo
	`awk -F: '($7 !~ "/bin/bash	/sbin/nologin	/bin/false/") {print $1,$7}' /etc/passwd`

Permisos críticos

ls -l

/etc/passwd

/etc/shadow

/etc/sudoers

Verificar permisos de archivos sensibles

Permisos que permitan escritura a usuarios no root
find / -perm -4000 -type f 2>/dev/null

Archivos SUID

Archivos con SUID peligrosos
find / -perm -2000 -type f 2>/dev/null

Servicios y procesos

systemctl list-units --type=service --state=running

Listar servicios activos

Servicios innecesarios o desconocidos
systemctl list-unit-files --type=service | grep enabled

Ver servicios habilitados en arranque

Servicios inseguros habilitados por defecto
ss -tuln

Ver puertos abiertos

Puertos no esperados abiertos

SSH

sudo grep -E "PermitRootLogin|PasswordAuthentication"
/etc/ssh/sshd_config

Revisar seguridad SSH

PermitRootLogin yes, PasswordAuthentication yes pueden ser
riesgos |
sudo grep "Failed password" /var/log/auth.log

Ver intentos fallidos de login

Intentos de fuerza bruta o ataques externos

Firewall y seguridad de red

sudo ufw status verbose

Revisar reglas activas de UFW

Falta de reglas o puertos abiertos innecesarios
sudo iptables -L -v -n

Revisar reglas iptables

Políticas laxas, INPUT ACCEPT general

Logs del sistema

last

Revisar últimos logins

Logins sospechosos o inusuales
sudo journalctl -p err

Revisar errores críticos

Mensajes de error persistentes o repetidos
sudo tail -f /var/log/syslog

Monitorización en tiempo real

Eventos inusuales, fallos de servicios

Integridad de archivos

sudo apt install aide -y y sudo aideinit && sudo aide –check

Detectar cambios no autorizados en archivos del sistema

Archivos modificados sin justificación

Rendimiento y recursos

top / htop

Monitorizar CPU y memoria

Procesos que consumen recursos excesivos
df -h

Revisar uso de disco

Discos casi llenos
free -m

Ver memoria disponible. Memoria insuficiente o swapping constante
ps aux --sort=-%cpu | head -10

Procesos más intensivos en CPU Procesos sospechosos o no autorizados
ps aux --sort=-%mem | head -10

Procesos más intensivos en RAM

Procesos sospechosos o no autorizados

Actualizaciones y parches

sudo apt update && sudo apt upgrade -y

Mantener el sistema actualizado

Paquetes obsoletos o vulnerables

Consejos

1. Ejecutar **checklist completa** como auditoría.
2. Registrar cualquier **alerta o anomalía** en un archivo de reporte.
3. Eliminar usuarios innecesarios, deshabilitar servicios, ajustar permisos y firewall.