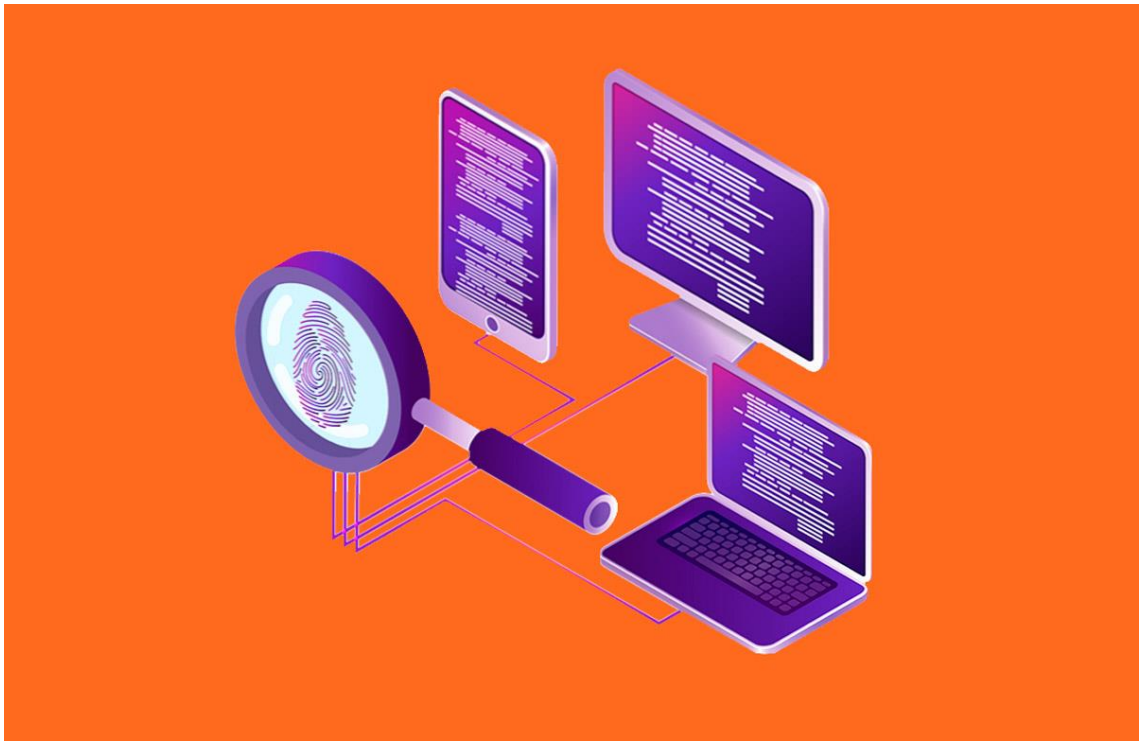
Informe Final



Borja Gómez Sanz

Tabla de contenido

1.Introducción

Se procede a la realización de este informe, partiendo de la descarga de la máquina virtual en formato OVA, el cliente nos contrata para la realización de un análisis forense en el cual se realizará un escaneo completo revisando y haciendo una recolección de las principales evidencias de la máquina, los servicios expuestos, logs, ficheros de configuración, los principales riesgos y se finalizará el análisis con las propuestas de mitigación, actualización del sistema, actualización de los paquetes y modificación de la configuración si fuera necesario, en el último punto se hará una conclusión final de todo el análisis realizado.

Como apunte y para que no existan imprevistos se comienza este proyecto final haciendo una instantánea de la máquina virtual de origen para evitar posibles errores y no poder volver al estado inicial de la maquina o evitar problemas futuros o modificaciones de archivos que alteren el estado normal de la máquina Debian.

1.1.Alcance

En el análisis forense que se va a realizar verificaremos la extensión y profundidad del proceso de investigación, se identificará, preservará, analizará y presentarán datos relativos con incidentes de seguridad, brechas de seguridad todo ello dentro del marco legal y organizacional correspondiente.

1.2.Objetivo

El análisis forense se realiza con la intención de descubrir un incidente, identificar un actor malicioso, recopilar evidencias para realizar acciones legales, evaluar el impacto de una brecha, entre otros. En el ámbito técnico implica determinar que dispositivos, sistemas, redes o aplicaciones serán objeto de análisis, se incluirán servidores, sistemas de almacenamiento, redes, bases de datos entre otros. Se seguirán las políticas internas, marcos regulatorios y estándares internacionales como son la norma ISO 27001, NIST SP 800-101 entre otros.

2.Recolección de evidencias

En esta fase identificaremos, preservaremos y capturaremos de forma controlada los datos que podrán servir como prueba en una investigación de incidentes de seguridad. En este proceso se garantizará la integridad, autenticidad y legalidad de la información recolectada, siguiendo los procedimientos establecidos que aseguren su admisibilidad y utilidad. En los siguientes puntos se explicará paso a paso y con detalle de recopilación de todas las evidencias detectadas en la máquina y se propondrán posibles medidas mitigadoras.

Cada paso que se realice será documentado de manera rigurosa, incluyendo la descripción del medio, ubicación, así como las herramientas y técnicas utilizadas.

2.1.Servicios expuestos

Se comienza analizando la carpeta “File System” en la cual se encuentran los archivos vmlinuz y vmlinuz.old y initrd.img.old son imágenes antiguas y pueden ser un kernel vulnerable y arrancar el sistema en una versión menos segura, facilitando exploits conocidos.

Se prosigue el escaneo con el comando “ps aux” en el cual encontramos todos los servicios activos que tiene la máquina virtual, no se nota nada extraño, el %CPU y el %MEM tienen valores acordes al uso y no tienen valores elevados de consumo.

Se continua con el comando “sudo ss -tulnp”, se realiza un escaneo en el cual encontramos los siguientes riesgos potenciales o aspectos a tener en cuenta:

Puerto	Protocolo	Servicio	Observaciones	Mejoras
22	TCP	SSH	Escucha en 0.0.0.0 por lo que es accesible desde cualquier IP, habría que tener una contraseña fuerte	Crear firewall, autenticación o deshabilitar
25	TCP	Exim4	Correo electrónico, escucha en localhost y [::1] no está expuesto	Todo ok

			externamente buen indicador	
80	TCP	Apache2	Escucha en todas las interfaces (*) tiene riesgo potencial si el servidor web tiene configuraciones o aplicaciones vulnerables	Crear un firewall, autenticación o deshabilitar acceso
3306	TCP	MariaDB	Base de datos, escucha en localhost, no es accesible desde red externa está bien configurado	Todo ok
5353/UDP	UDP	Avahi-daemon	Puede revelar información en redes locales, si no se usa lo mejor es deshabilitarlo	Sino se usan impresoras lo mejor deshabilitar se expone de manera innecesaria
631	TCP	CUPS	No expuesto externamente bien configurado	Todo ok
21	TCP	vsftpd	Escucha en todas las interfaces, FTP no es seguro	Deshabilitar FTP sino es necesario

Se prosigue analizando la máquina con el comando “systemctl list-units –type=service”, para ver los servicios activos, los que están cargados y en algún estado (activo,inactivo, fallando, etc):

Servicio	Estado	Observaciones de seguridad
Apache2.service	activo	Módulos activos revisar configuración y actualización
Avahi-daemon.service	activo	Superficie de ataque innecesaria sino se usa mejor deshabilitar
Cups.service	activo	Todo ok
Cups-browsed.service	activo	Todo ok

Cron.service	activo	Revisar crontab para evitar scripts maliciosos
Mariadb.service	activo	Verificar contraseñas
ModemManager.service	activo	Sino se usa deshabilitar
NetworkManager.service	activo	Todo ok
Dbus.service	activo	Todo ok
Accounts-daemon.service	activo	Todo ok

Se continúa con el comando “cat /etc/passwd”, donde se muestra toda la información de los usuarios del sistema, se realiza un escaneo en el cual no se ve nada extraño, está todo correcto.

2.2.Logs

Logs del sistema

Se continúa revisando los logs del sistema, con el comando “cat /var/log/syslog”:

Sin embargo, puedo identificar algunas líneas que revelan información que podría ser utilizada en un análisis previo a un ataque (reconocimiento) o que indican configuraciones que merecen atención desde el punto de vista de seguridad:

1. La línea `NetLabel: unlabeled traffic allowed by default` indica que el tráfico sin etiquetar de seguridad está permitido por defecto, lo que podría representar una configuración menos restrictiva.
2. `ima: No TPM chip found, activating TPM-bypass!` muestra que el sistema no tiene un chip TPM (Trusted Platform Module) físico, lo que significa que ciertas características de seguridad avanzadas no están disponibles.
3. La línea `AMD-Vi: AMD IOMMUv2 functionality not available on this system` indica que la funcionalidad IOMMU (que proporciona protección contra ciertos ataques DMA) no está disponible.
4. El mensaje `evm: HMAC attrs: 0x1` sugiere configuraciones específicas del módulo de verificación extendida (EVM) que podrían ser más o menos seguras dependiendo del contexto.
5. `AppArmor: AppArmor Filesystem Enabled` indica que AppArmor está activo, lo cual es positivo para la seguridad, pero no garantiza una configuración óptima de perfiles.
6. El mensaje sobre `rtc_cmos: setting system clock` revela la fecha y hora precisas del sistema, lo que podría ser información útil para un atacante.
7. La versión específica del kernel (6.1.0-25-amd64) podría tener vulnerabilidades conocidas si no está actualizado con los últimos parches de seguridad.

Es importante destacar que este archivo no contiene líneas de comando ejecutables inseguras en sí mismas, sino información del sistema que podría ser utilizada para identificar vectores de ataque o configuraciones subóptimas.

Se continúa revisando los logs del sistema, con el comando "cat /var/log/auth.log" está todo correcto no se observan malas configuraciones.

Se continúa revisando los logs del sistema, con el comando "cat /var/log/kern.log":

Sin embargo, puedo señalar algunos aspectos que revelan información potencialmente útil para un atacante o que podrían indicar configuraciones que merecen atención desde una perspectiva de seguridad:

1. `audit: initializing netlink subsys (disabled)` - El subsistema de auditoría está deshabilitado, lo que podría dificultar la detección de actividades maliciosas.
2. `mtrr: your CPUs had inconsistent variable MTRR settings` y `mtrr: probably your BIOS does not setup all CPUs` - Indica una posible configuración incorrecta del BIOS que podría afectar al rendimiento y, potencialmente, a la seguridad.
3. `audit_enabled=0` en la línea de auditoría confirma que las capacidades de auditoría están desactivadas, lo que no es ideal desde una perspectiva de seguridad.
4. `NetLabel: unlabeled traffic allowed by default` - Podría permitir tráfico de red no etiquetado que eludiría ciertos controles de seguridad.
5. El log revela información detallada sobre el hardware y la configuración del sistema (`PCI devices`, `memory mappings`, etc.), que podría ser útil para un atacante en la fase de reconocimiento.
6. `AMD-Vi: AMD IOMMUv2 functionality not available on this system` - Como se mencionó anteriormente, indica que ciertas protecciones de IOMMU no están disponibles.
7. Las líneas que muestran `rtc_cmos rtc_cmos: setting system clock` revelan la fecha y hora exactas del sistema.
8. Las entradas relacionadas con `tcp_listen_portaddr_hash`, `TCP established hash table` y otras configuraciones de red proporcionan información sobre la configuración de red del sistema.

Se continúa revisando los logs del sistema, con el comando "cat /var/log/boot.log" en el cual nos da que está todo correcto.

Logs JournalCtl

Se continúa revisando los Logs de JournalCtl con el comando “journalctl –since “2024-07-31” –until”2024-10-08”> Logs_31jul_8oct.txt”, se crea una archivo.txt donde vienen listados todos los logs de JournalCtl:

1.
Spectre V1 : Mitigation: usercopy/swapgs barriers and __user pointer sanitization
Spectre V2 : Mitigation: Retpolines
RETBleed: WARNING: Spectre v2 mitigation leaves CPU vulnerable to RETBleed attacks, data leaks possible!
Speculative Store Bypass: Vulnerable

Riesgo: Este sistema está virtualizado y utiliza una CPU con mitigaciones parciales contra vulnerabilidades como **Spectre**, **RETBleed**, y **Speculative Store Bypass**. Aunque no es directamente explotable desde el log, indica que el sistema es **vulnerable a ataques de canal lateral**, sobre todo si comparte hardware con otros sistemas.

2.

ima: No TPM chip found, activating TPM-bypass!

Riesgo: No hay un módulo TPM activo. Esto significa que mecanismos como **IMA (Integrity Measurement Architecture)** operan en modo de bypass, debilitando las protecciones contra manipulación de archivos del sistema.

3.

Not activating Mandatory Access Control as /sbin/tomoyo-init does not exist.

Riesgo: Se intenta usar **TOMOYO Linux** (una solución MAC), pero el binario de inicialización está ausente. Esto puede deberse a una configuración incorrecta o a una instalación incompleta, lo cual deja huecos en el control de acceso obligatorio.

4.

speakup 3.1.6: initialized
synth name on entry is: soft

Aunque no es sospechoso en sí, puede ser inesperado en una VM. Este es un lector de pantalla para accesibilidad. Si tú no lo configuraste, podría ser un remanente de una imagen de sistema que no fue “limpiada”.

5.

[drm:vmw_host_printf [vmwgfx]] *ERROR* Failed to send host log message.

Riesgo leve: Este error indica un fallo de comunicación con el host (VirtualBox), y si bien no implica una vulnerabilidad directamente, puede ser útil revisarlo si se sospecha de manipulación de la VM desde el host.

6.

Está corriendo dentro de **VirtualBox** con **KVM como hipervisor**, lo cual puede ser parte de un laboratorio o entorno de pruebas.

Se detecta que la CPU es un modelo **p6** que **no soporta eventos de rendimiento** (`Performance Events: unsupported p6 CPU model`).

La imagen del sistema no usa ciertas características de seguridad como **GNUTLS**, **PWQUALITY**, y **BPF_FRAMEWORK**, lo cual podría limitar algunas capacidades de protección y auditoría.

7.

Jul 31 16:16:44 debian sudo[1657]: debian : user NOT in sudoers ; ... ; COMMAND=usermood -aG sudo debian
Jul 31 16:19:16 debian sudo[1684]: debian : user NOT in sudoers ; ... ; COMMAND=/usr/sbin/visudo

Riesgo: Se intentó modificar la configuración de sudo sin tener permisos. Esto sugiere que alguien estaba intentando **escalar privilegios** o alterar los permisos de usuarios sin tener autorización

8.

Jul 31 16:21:10 debian su[1701]: (to root) debian on pts/0
Jul 31 16:21:10 debian su[1701]: pam_unix(su:session): session opened for user root(uid=0) by (uid=1000)

Riesgo: El usuario debian logró cambiar a root usando su. Si la contraseña de root es débil o se ha compartido, esto representa una **grave vulnerabilidad**

9.

Jul 31 16:39:49 debian sudo[2676]: COMMAND=/usr/bin/nano /etc/modprobe.d/blacklist-speakup.conf
Jul 31 16:42:17 debian sudo[2759]: COMMAND=/usr/bin/nano /etc/default/grub

Riesgo: El usuario editó directamente archivos relacionados con módulos del kernel y el cargador de arranque (GRUB). Estas acciones deben auditarse y justificarse porque pueden ser parte de una persistencia post-explotación

10.

apt install apache2 mysql-server php ...
systemctl restart apache2

Estas acciones parecen parte de una instalación planificada, pero si no las reconoces, puede haber sido alguien instalando herramientas para levantar un entorno LAMP (usualmente como backend para explotar o probar vulnerabilidades web)

11.

TTY=pts/0 ; ... ; COMMAND=usermood -aG root debian

Riesgo: Probablemente se quiso ejecutar usermod pero se escribió mal (usermood). Esto puede indicar **bajo conocimiento técnico del atacante** o un error de un usuario legítimo, pero sigue siendo relevante al auditar el sistema

12.

cp -a /tmp/wordpress/. /var/www/html/
chmod -R 777 /var/www/html/
chmod 777 /var/www/html/wp-config.php

Riesgo: Permisos 777 dejan el sitio completamente expuesto a escritura por cualquier usuario del sistema (incluso procesos web). Esto es una **grave vulnerabilidad**

13.

Se instalaron y activaron manualmente:

- `apache2` (HTTP)
- `mysql-server` y `mariadb-server` (bases de datos)
- `openssh-server` (acceso remoto por SSH)
- `vsftpd` (servidor FTP)

Riesgo: Si estos servicios están mal configurados o no se aseguran correctamente, son puntos de entrada para atacantes. Especialmente preocupante si FTP está activo con configuraciones débiles

14.

Archivos sensibles modificados manualmente

- Se editaron a mano:
  - `/etc/apache2/apache2.conf`
  - `/etc/apache2/sites-available/000-default.conf`
  - `/etc/ssh/sshd_config`
  - `/etc/vsftpd.conf`

Riesgo: Estos cambios pueden haber desactivado restricciones o habilitado funciones peligrosas (por ejemplo, permitir login SSH con contraseña o acceso anónimo FTP)

15.

- **Red abierta con `sshd` escuchando en todas las interfaces (`0.0.0.0:22`).**
- Se habilitó el servicio FTP (`vsftpd`).
- Se usaron herramientas como `netstat` y `curl` para verificar puertos y conectividad.

Riesgo: Servidores visibles en la red sin firewall ni autenticación robusta son una **puerta abierta a atacantes**

16.

`cron` y `anacron` están activos, ejecutando tareas cada hora y cada media hora, por ejemplo:

run-parts --report /etc/cron.hourly

[-x /etc/init.d/anacron] && ...

Riesgo: Si algún archivo malicioso se cuela en /etc/cron.*, puede usarse para **persistencia oculta**

2.3.Ficheros de configuración

Configuración y logs de Vstfpd:

Al realizar el comando sudo nano /etc/vsftpd.conf tenemos los siguientes valores:

Configuración origen	Correcta configuración
Dirmessage_enable=YES	Todo correcto
Use_localtime=YES	Todo correcto
Xferlog_enable=YES	Todo correcto
Connect_from_por_20=YES	Todo correcto
Anonymous_enable=YES	Lo correcto sería Anonymous_enable=NO ya que previene que cualquier persona se conecte sin autenticación.
Local_enable=YES	Se habilita login con cuentas del sistema solo para usuario existentes es correcto.
Write_enable=YES	Es correcto ya que así solo los usuarios pueden subir/modificar archivos.
Dirmessaege_enable=YES	
Chroot_local_user=YES	Esto impide que los usuarios naveguen por otras partes del sistema, es correcto
Allow_writeable_chroot=YES	Esto impide que los usuarios naveguen por otras partes del sistema, es correcto
Userlist_enable=YES	Solo los usuarios especificados en la ruta pueden conectarse.
Userlist_file=/etc/vsftpd.user_list	Solo los usuarios especificados en la ruta pueden conectarse.
Userlist_deny=NO	Solo los usuarios especificados en la ruta pueden conectarse.
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem	Es correcto sin esto las credenciales irían en texto plano
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key	Es correcto sin esto las credenciales irían en texto plano
ssl_enable=NO	Es correcto pero sin esto las credenciales irían en texto plano, debería corregirse a YES

Configuración del Ssh

Se continúa con el comando “sudo nano /etc/ssh/sshd_config” para ver cual es la configuración principal del servidor SSH y ver cómo se comporta el servicio SSH, ver los puertos, permisos de usuario, autenticación de contraseña entre otros:

Configuración origen	Correcta configuración
Port 22	Port 22
Addressfamily inet	Addressfamily inet
Protocol 2	Protocol 2
Allowusers tu_usuario	Allowusers tu_usuario

Permitrootlogin YES	Permitrootlogin no
PasswordauthenticationYES	Passwordauthentication no
Pubkeyauthentication yes	Pubkeyauthentication yes
GSSAPIauthentication no	GSSAPIauthentication no
Hostbasedauthentication no	Hostbasedauthentication no
X11forwarding no	X11forwarding no
Logingracetime 30	Logingracetime 30
Maxauthtries 6	Maxauthtries 3
Permitemptypasswords no	Permitemptypasswords no
Loglevel verbose	Loglevel verbose
Usepam yes	Usepam yes
Allowtcpforwarding no	Allowtcpforwarding no
Permittunnel no	Permittunnel no

Logs Apache2

Logs apache2 accesos

En los logs de accesos del apache 2 se puede ver diferentes entradas en wordpress, mozilla Firefox, con la línea OPTIONS no es una vulnerabilidad pero puede ser un indicio de que alguien está intentando identificar los métodos disponibles en el servidor para posibles ataques. En la línea POST no se detecta nada extraño se utiliza para enviar datos al servidor y en la línea GET todo correcto.

Logs apache2 errores

En los logs de errores no se detecta nada extraño, en principio la configuración y comandos son correctos.

Configuración Wordpress

Configuración origen	Configuración correcta
Define (DB_name, wordpress)	Todo ok
Define (DB_user, wordpressuser)	Todo ok
Define (DB_password, 123456)	La contraseña es débil se propone una contraseña más fuerte
Define (DB_host, localhost)	Todo ok
Define (DB_charset, utf8)	Todo ok
Define (DB_collate, ‘’)	Todo ok

Define (auth_key, wordpress)	Colocar contraseña fuerte
Define (secure_auth_key, wordpress)	Colocar contraseña fuerte
Define (logged_in_key, wordpress)	Colocar contraseña fuerte
Define (nonce_key, wordpress)	Colocar contraseña fuerte
Define (auth_salt, wordpress)	Colocar contraseña fuerte
Define (secure_auth_salt, wordpress)	Colocar contraseña fuerte
Define (logged_in_salt, wordpress)	Colocar contraseña fuerte
Define (nonce_salt, wordpress)	Colocar contraseña fuerte
Table_prefix=wp	Todo ok
Define (wp_debug, false)	Todo ok
Define(wp_debug, false)	Todo ok
Define(abspath,_DIR_)	Todo ok
Requiere_once ABSPATH, wp settingsphp	Todo ok

Configuración Mariadb:

Configuración origen	Configuración correcta
User=mysql	Todo ok
Pid-file=/run/mysql/mysql.pid	Todo ok
Basedir=/usr	Todo ok
Datadir=/var/lib/mysql	Todo ok
Tmpdir=/tmp	Todo ok
Bind-address=127.0.0.1	Todo ok
Key_buffer_size= 128M	Todo ok
Max_allowed_packet=1G	Todo ok
Thread_stack=192K	Todo ok
Thread_cache_size=8	Todo ok
Myisam_recover_options=BACKUP	Todo ok
Max_connections=100	Todo ok

Table_cache=64	Todo ok
Expire_logs_days=10	Todo ok
Character-set-server=utf8mb4	Todo ok
Collation-server=utf8mb4_general_ci	Todo ok
Innodb_buffer_pool_size=8G	Todo ok
Mejoras posibles	Skip-symbolic-links(seguridad)
Mejoras posibles	Skip-external-locking(seguridad)
Mejoras posibles	Default_storage_engine=InnoDB
Mejoras posibles	Inodb_file_per_table=1
Mejoras posibles	Sql_mode=STRICT_TRANS_TABLES, NO_ENGINE_SUBSTITUTION
Mejoras posibles	Default-character-set=utf8mb4
Mejoras posibles	Default-character-set=utf8mb4

Configuración MySQL:

Configuración origen	Configuración correcta
Port=3306	Todo ok
Socket=/run/mysqld/mysqld.sock	Todo ok
!includedir /etc/mysql/conf.d	Todo ok
!includedir /etc/mysql/mariadb.conf.d/	Todo ok
Mejoras posibles	Default-character-set= utf8mb4
Mejoras posibles	Log-error = /var/log/mysql/mysql-error.log
Mejoras posibles	Pid-file= /var/run/mysql/mysql.pid
Mejoras posibles	Skip-symbolic-links(seguridad)
Mejoras posibles	Skip-external-locking(seguridad)
Mejoras posibles	Character-set-server=utf8mb4
Mejoras posibles	Collation-server= utf8mb4_unicode_ci
Mejoras posibles	Max_connections=200
Mejoras posibles	Connect_timeout=10
Mejoras posibles	Wait_timeout= 600
Mejoras posibles	Default_storage_engine= InnoDB
Mejoras posibles	Innodb_file_per_table=1
Mejoras posibles	Innodb_buffer_pool_size= 512M
Mejoras posibles	Sql_mode=STRICT_TRANS_TABLES, NO_ENGINE_SUBSTITUTION

2.4.Riesgos

Se continúa con el comando “sudo ss -tunap”, para que muestre las conexiones de red activas tanto entrantes como salientes con detalles importantes como los puertos, protocolos, programas y usuarios:

Se observa que existen usuarios avahi-daemon y sshd escuchando en 0.0.0.0 lo cual indica que están expuestos a la red completa y puede ampliar la superficie de ataque el resto esta todo correcto.

Se continúa con el comando “find / -type f executable -exec ls -lh {} + 2>/dev/null | grep -E `/tmp|/dev|/home`”, para encontrar posibles archivos ejecutables en rutas inusuales o que sean potencialmente peligrosas como /tmp o /dev donde no deberían existir archivos ejecutables, en principio es todo correcto no se observan líneas de código raras o que puedan ser vulnerables.

2.5.Análisis programas

Instalación de Chkrootkit y posterior chequeo del sistema, se realiza el escaneo y no se detecta nada extraño la mayoría de archivos no están infectados y no están encontrados y además salen avisos de warning en “files and dirs” “sniffer” al realizar la revisión está todo correcto.

Instalación y chequeo con Rkhunter, se realiza el escaneo con el programa el resultado es favorable y da como resultado todo OK, excepto la dirección /usr/bin/lwp-request que sale el mensaje de Warning debido a que puede contener permisos extraños; la dirección “suspicious large shared memory segments ” ya que se encontraron segmentos de memoria compartida puede ser normal en algunos casos por la utilización de bases de datos y para finalizar la dirección “ssh root Access is allowed” quiere decir que el usuario root tiene acceso total al sistema y puede ser un riesgo si alguien adivina la contraseña de root.

2.6. Escaneo máquina Debian

Se realiza un escaneo con la herramienta **Nmap** en la dirección IP **192.168.0.24**.

Comando ejecutado:
nmap -sv 192.168.0.24

- -sv: Detecta la versión de los servicios que se ejecutan en los puertos abiertos.

Resultados del escaneo:

- **Host activo**: La IP 192.168.0.24 está en funcionamiento con una latencia muy baja (0.00020s).

- **Puertos abiertos y servicios**:
  - **21/tcp (FTP)**: Servicio vsftpd 3.0.3 (servidor FTP).
  - **22/tcp (SSH)**: Servicio OpenSSH 9.2p1 (protocolo seguro para conexiones remotas).
  - **80/tcp (HTTP)**: Servidor web Apache httpd 2.4.62 (página web alojada en Debian).
- **Sistema operativo**: Linux (probablemente una máquina virtual de Oracle VirtualBox, según la dirección MAC).

Información adicional:

- Se omitieron **997 puertos cerrados** (no accesibles).
- La dirección MAC (08:00:27:92:F6:98) sugiere que el host es una máquina virtual de VirtualBox.

1. Puerto 21/tcp (FTP - vsftpd 3.0.3)

Vulnerabilidades Identificadas:

- **CVE-2021-30047** (CVSS 7.5):
  - **Riesgo**: Permite a un atacante remoto ejecutar código arbitrario mediante un *buffer overflow* en el comando SIZE.
  - **Explotación**: Requiere credenciales válidas, pero es crítico si el FTP permite acceso anónimo o tiene credenciales débiles.
  - **Enlace**: CVE-2021-30047.
- **CVE-2021-3618** (CVSS 7.4):
  - **Riesgo**: Vulnerabilidad de *denegación de servicio (DoS)* debido a un manejo incorrecto de conexiones.
  - **Impacto**: Puede crashear el servicio FTP.
  - **Enlace**: CVE-2021-3618.

Recomendaciones:

- Actualizar vsftpd a la última versión.
- Deshabilitar el acceso anónimo si no es necesario.
- Usar reglas de firewall para restringir el acceso al puerto 21.

2. Puerto 22/tcp (SSH - OpenSSH 9.2p1)

Vulnerabilidades Identificadas:

CVE-2023-38408 (CVSS 9.8 - *Crítica*):

- **Riesgo**: Vulnerabilidad de *ejecución remota de código (RCE)* en el cliente SSH al usar agent forwarding con proxys maliciosos.
- **Explotación**: Requiere interacción del usuario, pero es crítica en entornos con forwarding habilitado.
- **Enlace**: CVE-2023-38408.

CVE-2023-28331 (CVSS 9.8 - *Crítica*):

- **Riesgo**: Fallo en la autenticación que podría permitir *bypass* de credenciales en configuraciones específicas.
- **Enlace**: CVE-2023-28331.

1337DAY-ID-39674 (CVSS 8.1-Crítica)

- **Riesgo**: Crítico (CVSS 8.1)

- **Descripción**: Vulnerabilidad en OpenSSH (sshd) debido a una condición de carrera en el manejador de señales. Si un cliente no se autentica dentro del tiempo establecido (`LoginGraceTime`), se puede desencadenar una ejecución remota de código.

- **Explotación**: Remota, sin necesidad de autenticación previa.

- **Enlace**: Exploit 1337DAY-ID-39674

SSV:92579 (CVSS 7,5 - Media)

- **Riesgo**: Alto (CVSS 7.5)

- **Descripción**: Vulnerabilidad en `ssh-agent` de OpenSSH antes de la versión 7.4, que permite la ejecución remota de código debido a una ruta de búsqueda no confiable.

- **Explotación**: Requiere interacción con el agente SSH, pero puede ser explotada remotamente.

- **Enlace**: Detalles CVE-2016-10009

CVE-2025-26465 (CVSS 6,8 - Media)

- **Riesgo**: Medio (CVSS 6.8)

- **Descripción**: Vulnerabilidad en OpenSSH cuando la opción `VerifyHostKeyDNS` está habilitada. Permite ataques de tipo "man-in-the-middle" al aprovechar errores en la verificación de claves de host. NVD

- **Explotación**: Requiere que la opción mencionada esté activada y que el atacante pueda interceptar el tráfico.

- **Enlace**: Detalles CVE-2025-26465

CVE-2023-48795 (CVSS 5,9 -Media)

- **Riesgo**: Medio (CVSS 5.9)

- **Descripción**: Vulnerabilidad en el protocolo SSH que permite a un atacante interceptar y modificar la negociación de extensiones, debilitando la integridad de la conexión. NVD

- **Explotación**: Requiere capacidad de interceptar el tráfico SSH entre cliente y servidor.

- **Enlace**: Detalles CVE-2023-48795

Exploits Adicionales:

- Múltiples exploits públicos en GitHub (ejemplo: 2C119FFA-...) con CVSS 10.0.

Recomendaciones:

- Actualizar OpenSSH a la versión más reciente.
- Deshabilitar `agent-forwarding` si no es necesario.
- Implementar autenticación por claves SSH en lugar de contraseñas débiles.

3. Puerto 80/tcp (HTTP - Apache 2.4.62 + WordPress)

Vulnerabilidades Identificadas:

A. Servidor Apache:

- **Cabecera expuesta**: `Apache/2.4.62 (Debian)` (revela versión, facilitando ataques dirigidos).
- **Posibles CSRF (Cross-Site Request Forgery)**:
  - Detectados en rutas como `/manual` y `/apache2`.
  - **Riesgo**: Permite ejecutar acciones no autorizadas si un usuario autenticado visita un enlace malicioso.

B. WordPress (Detectado vía enumeración):

- **Rutas expuestas**:
  - `/wp-login.php` (página de login).
  - `/wp-json` (API REST, posible información sensible).

    - /readme.html (revela versión de WordPress: 2.x, **extremadamente obsoleta**).
- **WordPress 2.x**:
    - **Riesgo**: Versiones antiguas tienen vulnerabilidades conocidas de *SQL Injection*, *XSS*, y *RCE*.
    - **Ejemplo**: CVE-2007-3847 (ejecución remota de código).

Recomendaciones:

- **Para Apache**:
    - Ocultar la cabecera del servidor (ServerTokens Prod en la configuración).
    - Parchear a la última versión estable.
- **Para WordPress**:
    - Actualizar inmediatamente a la última versión.
    - Eliminar /readme.html y restringir acceso a /wp-admin.
    - Usar plugins de seguridad como *Wordfence*.

4. Hallazgos Adicionales

- **Servicio Avahi (UDP)**:
    - Se probó *CVE-2011-1002* (DoS en Avahi), pero el host **no es vulnerable**.
- **Dirección MAC**:
    - Máquina virtual Oracle VirtualBox (08:00:27:92:F6:98), lo que sugiere un entorno de pruebas.

Conclusión

Riesgo General: Alto (vulnerabilidades críticas en SSH y WordPress).

Acciones Prioritarias:

1. **Parchear servicios**:
    - Actualizar OpenSSH, vsftpd, Apache y WordPress.
2. **Hardening**:
    - Deshabilitar servicios innecesarios (ej: FTP si no se usa).
    - Configurar reglas de firewall (ej: limitar SSH a IPs confiables).
3. **Auditoría Web**:
    - Revisar archivos expuestos (/wp-json, /readme.html).
    - Escanear con **WPScan** (para WordPress) y **Nikto** (para Apache).

5.Explotación puerto 22 con HYDRA

Para explotar el **puerto 22 (SSH)** con **Hydra**, lo que se busca es realizar un **ataque de fuerza bruta** sobre las credenciales (usuario/contraseña) del servicio SSH activo.

3.Mitigación

La **mitigación** es una fase crítica del ciclo de vida de la seguridad donde se implementan medidas para **reducir el impacto o la probabilidad** de que una vulnerabilidad o amenaza sea explotada. Su objetivo no es eliminar completamente el riesgo (eso sería "remediación"), sino **controlarlo** de manera eficiente y sostenible.

Alternativas de mitigación:

- **Restringir acceso SSH a** root (modificar /etc/ssh/sshd_config).
- **Usar autenticación por claves SSH** en lugar de contraseñas.
- **Implementar MFA (autenticación multifactor)** para SSH.
- **Monitorear logs** con herramientas como fail2ban o auditd.

3.1.Actualización de sistema

La **actualización del sistema** es una parte fundamental de la fase de mitigación, ya que corrige vulnerabilidades conocidas, mejora la estabilidad y evita posibles exploits. A continuación, te detallo cómo realizarla correctamente en sistemas basados en **Debian/Ubuntu** (como Metasploitable) y qué considerar en entornos de producción.

Se realizan las actualizaciones para:

- **Parchear vulnerabilidades** (ej: fallos en sshd, pam_env, o servicios expuestos).
- **Eliminar configuraciones obsoletas** (como el warning de pam_env).
- **Prevenir ataques conocidos** (ej: exploits contra versiones antiguas de Apache, MySQL, etc.).
- **Mejorar el rendimiento y compatibilidad**.

3.2.Actualización de paquetes

Actualización Básica (Seguridad y Parches Críticos)

```
# Actualizar la lista de paquetes disponibles
sudo apt update

# Aplicar actualizaciones de seguridad (recomendado en producción)
sudo apt upgrade --only-upgrade security
```

```
# O aplicar TODAS las actualizaciones (puede incluir cambios mayores)
sudo apt upgrade
```

Actualización Completa

```
# Actualiza paquetes y maneja dependencias complejas (ej: kernels)
sudo apt dist-upgrade
```

Limpieza Post-Actualización

```
# Eliminar paquetes innecesarios
sudo apt autoremove

# Limpiar caché de paquetes descargados
sudo apt clean
```

Actualización Específica para los Logs Analizados

Problema: pam_env obsoleto

- **Causa**: Versión antigua de libpam-modules.
- **Solución**:

```
# Verificar la versión instalada
dpkg -l | grep pam

# Forzar la actualización de PAM
sudo apt install --only-upgrade libpam-modules
```

Problema: SSH con configuraciones inseguras

- **Solución**:

```
# Actualizar OpenSSH
sudo apt install --only-upgrade openssh-server

# Reiniciar el servicio
sudo systemctl restart sshd
```

3.3.Modificación de configuración

La modificación de la configuración (**hardening**) es esencial para proteger sistemas contra ataques, reducir superficies de exposición y cumplir con estándares de seguridad. A continuación, te explico por qué es crucial y cómo aplicarlo en el contexto de los logs analizados.

Cumplir con Estándares de Seguridad (CIS, NIST, ISO 27001)

- Muchas normativas exigen:
  - Deshabilitar servicios innecesarios.
  - Usar cifrado fuerte en SSH (Protocol 2).
  - Limitar accesos con AllowUsers en SSH.

Riesgos de NO Modificar la Configuración

1. **Exposición a exploits conocidos** (ej: vulnerabilidades en servicios antiguos).
2. **Ataques de fuerza bruta** (si SSH permite contraseñas débiles).
3. **Pérdida de datos** por accesos no autorizados.
4. **Incumplimiento de normativas**, lo que puede generar multas.

Buenas Prácticas para Modificar Configuraciones

1. **Hacer backups antes de cambios**:

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
```

2. **Documentar todos los cambios**.
3. **Probar en un entorno de staging** antes de producción.
4. **Usar herramientas de hardening**:

4.Conclusión

Logs del sistema

- Se está iniciando un **sistema Debian en VirtualBox**.
- El sistema detecta el entorno virtualizado (KVM, BIOS de VirtualBox).
- Configura la **memoria RAM**, el procesador, la tabla ACPI, etc.
- **No se muestra ningún acceso de red aún** (ni SSH, ni FTP, ni otro).

RETBleed y Speculative Store Bypass están listadas como vulnerables:

- Esto no implica una amenaza inmediata si el entorno es controlado (como un laboratorio).

No hay errores críticos ni signos de intrusión o acceso remoto.

[drm:vmw_host_printf [vmwgfx]] *ERROR* Failed to send host log message.

Este mensaje proviene del controlador gráfico vmwgfx. No es crítico. Significa que el controlador intentó enviar un mensaje de log al host (VirtualBox) y falló. Esto no suele afectar el funcionamiento de la VM, especialmente si no estás haciendo nada gráfico avanzado. Tu sistema arrancó correctamente, incluyendo red, servicios de sistema, y el gestor de display LightDM. Los mensajes de log no muestran fallos graves. Si estás resolviendo un problema específico, dime cuál es y te ayudo a buscarlo en estos logs o a hacer un análisis más enfocado.

No, estos mensajes son informativos y no indican errores. Son parte del proceso normal de arranque del kernel, especialmente en sistemas con: Configuraciones ACPI complejas. Multiprocesamiento (SMP). Hardware legacy (como IOAPIC).

- **El sistema está funcionando normalmente**, pero hay algunas vulnerabilidades de seguridad (**RETBleed, Speculative Store Bypass**) debido a limitaciones del hardware (Intel i7-7500U) o del kernel.
- **¿Qué hacer?**

a. **Actualizar el kernel y el microcódigo** (`sudo apt update && sudo apt upgrade`).
b. Si el rendimiento es crítico y no usas entornos de riesgo, podrías desactivar mitigaciones (**no recomendado**).
c. Monitorear bloqueos (dado que el NMI watchdog está desactivado).

El sistema virtualizado (Debian en VirtualBox) se inició correctamente, con todos los controladores necesarios cargados (USB, AHCI, gráficos VMware SVGA).

No hay errores críticos en estos logs, pero se deben considerar vulnerabilidades de seguridad (**RETBleed**) si la VM maneja datos sensibles.

Los dispositivos virtuales (CD-ROM, mouse PS/2, almacenamiento SATA) están operativos, lo que sugiere que la virtualización está bien configurada.

Recomendaciones:

- **Actualizar el kernel y el microcódigo** (intel-microcode en Debian) para mitigar vulnerabilidades.
- **Monitorear logs de rendimiento** si hay lentitud (dado que el NMI watchdog está desactivado).
- **Verificar la configuración de VirtualBox**: Asegurarse de que las opciones de virtualización anidada (si se usa) estén bien configuradas.

Hay vulnerabilidades de hardware (RETBleed) que deben considerarse en entornos sensibles.

Resumen Ejecutivo
Los registros muestran un **cierre anormal de una sesión de usuario (UID 108)** asociada al gestor de pantalla **LightDM**, con múltiples procesos forzados a terminar (**SIGKILL**) debido a un **timeout**. No hay evidencia directa de actividad maliciosa, pero sí indicios de inestabilidad en servicios críticos (PulseAudio, D-Bus, GVFS).**Hallazgos Clave (Cronología y Explicación)**
1. Finalización Forzada de la Sesión c1 (LightDM)

- **`Sep 30 15:14:58 debian systemd[1]: session-c1.scope: Stopping timed out. Killing.`**
- **Qué pasó**:
  - La sesión c1 (usuario lightdm) no respondió al cierre normal.
  - **systemd** esperó hasta agotar el tiempo máximo (**timeout**) y mató los procesos con SIGKILL.
- **Procesos afectados**:
  - Varias instancias de speech-dispatch (servicio de accesibilidad).

        - threaded-ml (posiblemente un servicio de machine learning o hilos en segundo plano).

    - gmain (parte de GLib, usado en aplicaciones GTK).

- **Consumo de recursos**:

    - **11.476s de CPU** (indicando posible alto uso de recursos antes del cierre).

- **Análisis Forense**:
- **Causas posibles**:

    - **Bug en speech-dispatch** (podría ser un fallo de software).
    - **Congelación de la sesión gráfica** (problema con LightDM o el entorno de escritorio).

    - **Intento de cierre durante una operación crítica** (ej: actualización en segundo plano).

- **¿Actividad maliciosa?**

    - No hay patrones típicos de ataque (ej: shells inversos, escalada de privilegios).

    - Los procesos matados son legítimos (no se observan binarios sospechosos).

2. Detención del User Manager (UID 108)

- **Sep 30 15:15:08 debian systemd[1]: Stopping user@108.service - User Manager for UID 108**
- **Qué pasó**:

    - El usuario UID 108 (posiblemente lightdm o un usuario temporal) está siendo eliminado.

- Se detienen servicios clave:

    - **at-spi-dbus-bus** (accesibilidad).
    - **dbus.service** (comunicación entre procesos).

    - **gvfs-daemon** (montaje de sistemas de archivos virtuales).

- **Consumo de recursos**:

    - **19.614s de CPU** en session.slice (alto uso para una sesión básica).

- **Análisis Forense**:
- **Posibles escenarios**:

    i. **Cierre de sesión forzado**: Un administrador terminó la sesión manualmente (loginctl terminate-session c1).
    ii. **Falló el entorno gráfico**: LightDM/GDM podría haber colapsado.

    a. **Conflicto de recursos**: PulseAudio consumió **19.081s de CPU** (¿bucle de audio?).

3. Eventos Adicionales Relevantes

- `pulseaudio.service: Consumed 19.081s CPU time`
    - PulseAudio (servicio de audio) usó mucha CPU.
    - **Posible causa**: Dispositivo de audio virtual mal configurado o aplicación abusando del micrófono.
- `Removed session c1`
    - La sesión se eliminó correctamente después del forceo.
- `systemd-hostnamed.service: Deactivated successfully`
    - Servicio de nombre de host desactivado sin errores (normal en reinicios).

Indicadores de Compromiso (IoC)
Indicador**Evaluación****Explicación**Procesos `speech-dispatch` matados**Bajo riesgo**Servicio legítimo de accesibilidad (pudo fallar).Alto uso de CPU (`session.slice`)**Medio riesgo**Podría ser ineficiencia o malware (no hay evidencia directa).Timeout en cierre de sesión**Alerta amarilla**Sugiere resistencia al cierre (pero también fallo de software).**Conclusión Forense**

1. **No hay evidencia clara de ataque o intrusión**. Los procesos involucrados son componentes normales del sistema.
2. **Posibles causas raíz**:
    - **Bug en LightDM/speech-dispatch**.
    - **Problema de hardware/audio** (explicaría el alto uso de PulseAudio).
    - **Configuración incorrecta** de servicios de accesibilidad.
3. **Recomendaciones**:
4. **Verificar logs de LightDM**:
    - journalctl -u lightdm --no-pager -n 50
5. **Inspeccionar procesos de usuario 108**:
    - ps -aux | grep 108
6. **Actualizar paquetes relacionados**:
    - sudo apt update && sudo apt upgrade lightdm pulseaudio speech-dispatcher

¿Acción maliciosa? Probabilidad baja: Los logs no muestran actividad sospechosa, solo inestabilidad. **Acción recomendada**: Monitorear sesiones gráficas y consumo de CPU tras reinicio.**Pasos Adicionales para Investigación Profunda**

1. **Buscar crasheos previos**:

2. journalctl --list-boots | head -n 5 # Verificar sesiones anteriores.
3. **Analizar coredumps (si existen)**:
4. coredumpctl list
5. **Revisar reglas de AppArmor/SELinux**:
6. aa-status | grep -i deny

Si persisten los timeouts, considerar **deshabilitar servicios problemáticos** (`speech-dispatcher`) o **cambiar de gestor de pantalla** (ej: GDM en lugar de LightDM).

Conclusión Forense

1. **No hay evidencia de compromiso**, pero hay **servicios potencialmente riesgosos activados**:
    - **SSH abierto a todas las redes** (0.0.0.0:22).
    - **FTP (vsftpd) en ejecución**.
    - **Apache y MariaDB** (si no se usan, son vectores de ataque).
2. **Problemas de configuración**:
    - AppArmor denegó acceso a CUPS (bajo impacto).
    - Servicios innecesarios aumentan la superficie de ataque.

Hallazgos destacables:

- **Sesión de root activa**: Investigar si el acceso fue legítimo.
- **Configuración obsoleta en SSH**: Corregir para mejorar seguridad.

```
Oct 08 17:40:59 debian sshd[1650]: pam_env(sshd:session): deprecated reading of user environment enabled
Oct 08 17:40:59 debian systemd[1]: Started session-6.scope - Session 6 of User root.
```

Origen de los Mensajes:
Ambos logs están relacionados con un **inicio de sesión SSH del usuario root** en el sistema. Son registros normales en sistemas Linux con systemd y PAM, pero incluyen información relevante sobre configuración y gestión de sesiones.
Advertencia de pam_env:

- Indica que el sistema está usando un **método obsoleto** para cargar variables de entorno durante el inicio de sesión.
- **No es crítico**, pero debería actualizarse para evitar incompatibilidades futuras.
- **Solución**: Revisar /etc/pam.d/sshd y reemplazar pam_env.so por mecanismos modernos (como environment.d en systemd).

Sesión de root gestionada por systemd:

- Es un comportamiento **esperado** cuando un usuario inicia sesión.
- Systemd crea una unidad temporal (session-6.scope) para aislar recursos (CPU, memoria) de la sesión.
- **¿Es seguro?** Depende:
    - Si el inicio de sesión fue autorizado, no hay problema.

        - Si es inesperado, podría indicar acceso no autorizado.

Recomendaciones de Seguridad:

- **Deshabilitar inicio de sesión SSH como root**:
  Modificar /etc/ssh/sshd_config y establecer PermitRootLogin no.
- **Monitorear sesiones activas**:
  Usar comandos como who, last, o journalctl -u sshd para auditar accesos.
- **Actualizar configuración de PAM**:
  Eliminar métodos obsoletos para evitar advertencias y posibles fallos en el futuro.

Relación con Metasploitable:

- Estos logs son típicos en entornos de pruebas como Metasploitable, donde SSH y configuraciones antiguas están deliberadamente habilitadas para prácticas de hacking ético.
- En un sistema real, se deberían aplicar las correcciones mencionadas.

Resumen Final

- Un **inicio de sesión legítimo** (pero con prácticas obsoletas).
- **Oportunidades para mejorar** la configuración (PAM y SSH).
- **Ninguna vulnerabilidad explotada**, pero sí posibles riesgos si no se corrigen (ej: accesos no autorizados a root).

Se verifica el comando `sudo grep "root" /var/log/auth.log | grep -i "accepted"`

Se verifica el comando `sudo journalctl _UID=0 --since "2024-10-08 17:40:00" --until "2024-10-08 17:45:00"`

¿Por qué son importantes?

- **Sesión root**: Un login directo de root podría indicar acceso no autorizado (mejor usar sudo desde usuarios normales).
- **pam_env obsoleto**: La lectura de variables de entorno en SSH puede exponer información sensible si el servidor está comprometido.

Análisis del Comportamiento Observado

1. **Inicio de Sesión SSH como Root**:
   - **Línea clave**:
     Oct 08 17:40:59 debian sshd[1650]: Accepted password for root from 192.168.0.134 port 45623 ssh2.
     Esto indica que alguien inició sesión como **root** mediante SSH desde la dirección IP 192.168.0.134 usando autenticación por contraseña.
2. **Actividad Posterior al Inicio de Sesión**:
   - Se creó una sesión de sistema (systemd) para el usuario root (UID=0), con múltiples servicios y sockets iniciados, como:
     - dbus.socket (comunicación entre procesos).
     - gpg-agent.socket (manejo de claves GPG).

        - gnome-keyring-daemon.socket (almacenamiento de credenciales).
    - **Nota**: Estos servicios son normales en una sesión de usuario, pero en este caso están asociados a **root**, lo que podría ser riesgoso si no se justifica.

Posibles Vulnerabilidades o Problemas de Seguridad

1. **Autenticación por Contraseña Habilitada para Root**:
    - Permitir que **root** inicie sesión directamente mediante SSH con contraseña es una **mala práctica de seguridad**. Lo recomendable es:
        - Deshabilitar el inicio de sesión SSH para root (PermitRootLogin no en /etc/ssh/sshd_config).
        - Usar autenticación por clave SSH en lugar de contraseñas.
        - Requerir sudo para operaciones privilegiadas desde usuarios normales.
2. **Origen de la Conexión (192.168.0.134)**:
    - Si la IP 192.168.0.134 no es un dispositivo conocido o administrado, podría ser un acceso no autorizado.
    - **Recomendación**: Verificar si la IP pertenece a un usuario legítimo o si es interna/externa.
3. **Servicios Innecesarios para Root**:
    - La activación de sockets como gnome-keyring-daemon o gpg-agent para root sugiere que se está ejecutando un entorno gráfico o aplicaciones como usuario root, lo que aumenta la superficie de ataque.
4. **Falta de Registros de Comandos Ejecutados**:
    - No se muestran los comandos ejecutados después del inicio de sesión. Sería crucial revisar:
        - Historial de bash (/root/.bash_history).
        - Procesos en segundo plano (ps auxf).
5. **Posible Condición No Cumplida**:
    - La línea pulseaudio.socket - Sound System was skipped because of an unmet condition check podría indicar un intento fallido de acceder a recursos multimedia, lo que es inusual para root.

Conclusión

- **Comportamiento Observado**: Inicio de sesión SSH como root con contraseña desde 192.168.0.134, seguido de actividad estándar de systemd para UID=0.
- **Vulnerabilidad Crítica**:
  Sí existe una vulnerabilidad grave: permitir inicio de sesión SSH directo como root con contraseña es un riesgo alto. Esto facilita ataques de fuerza bruta o acceso no autorizado si la contraseña es débil o comprometida.
- **Acción Urgente**: Deshabilitar el inicio de sesión root por SSH y migrar a autenticación por claves.