SSO Práctica 1 Sesión 2

Andrés Merlo Trujillo

Índice

Ejercicio 1

Apartado A

Mediante la orden `lsof -i` ejecutada como root, podemos obtener la informacion de los servicios y procesos que tienen alguna conexion abierta o archivo abierto.

La orden ofrece 9 columnas con los siguientes signifiacdos:

- **COMMAND:** Nombre del comando asociado al proceso/archivo.
- **PID:** Process IDentificator (identificador de proceso).
- **USER:** UID del usuario al que pertenece el proceso/archivo.
- **FD:** Descriptor de fichero.
- **TYPE:** Tipo de archivo asociado al mismo (GDIR, GREG, ...) o indica el tipo de conexion (en cpaa de red) (IPv4, IPv6, X.25, etc.).
- **DEVICE:** Numero de dispotivio.
- **SIZE/OFF:** Tamaño del archivo.
- **NODE:** Numero de nodo/inodo de un fichero o el procotolo en capa de transporte (TCP, UDP, ...).
- **NAME:** Punto de montaje y sistema de archivos que usa el archivo abierto. Tambien puede significar la direccion local o remota de internet o de un socket.

A continuación explicaré dos procesos de la salida del comando anterior:

sshd

- **COMMAND:** sshd
- **PID:** 1319
- **USER:** root
- **FD:** 3u/4u (FDs 3 y 4. La letra “u” indica acceso de lectura y escritua)
- **TYPE:** IPv4/IPv6 (está a la espera de recibir algo en las dos versiones del protocolo IP.)
- **DEVICE:** 22997/23008
- **SIZE/OFF:** 0t0 (Offset, el segundo “0” indica que no hay offset)
- **NODE:** TCP (usan este protocolo de transporte porque asegura que se reciben los paquetes mediante ACKs).
- **NAME:** *:ssh (LISTEN) (El asterisco indica que espera de cualquier IP, en el puerto ssh (configurable, por defecto el 22)).

avahi-daemon

- **COMMAND:** avahi-dae (avahi-daemon)
- **PID:** 1144
- **USER:** avahi
- **FD:** 14u (FD 14. La letra “u” indica acceso de lectura y escritua)
- **TYPE:** IPv6
- **DEVICE:** 22668
- **SIZE/OFF:** 0t0 (Offset, el segundo “0” indica que no hay offset)
- **NODE:** UDP
- **NAME:** *:53167 (Cualquier IP en el puerto 53167).

Apartado B

Leyendo el manual, hace falta usar el switch "-i", como en el apartado anterior, y añadiendo que busque las conexiones con el servicio "ssh". Por tanto, el comando quedaria asi: `lsof -i :ssh`.

Ahora mismo no hay nadie conectado, solo estan los "daemons" a la escucha de peticiones de conexion. Si ahora me conecto desde el otra maquina virtual a la de Ubuntu, la salida es la siguiente:

Aparecen dos lineas nuevas y en el apartado `NAME` se ve que la conexion es entre el usuario "andres-kvm" (Ubuntu) usando el servicio "ssh" (en mi caso es el puerto 22) y el usuario "archlinux" en el puerto 57686, que es un puerto que se asigna aleatoriamente para enviar informacion (escuchar) a "archlinux".

Con la orden `lsof -c sshd` se puede ver los archivos que tiene abiertos SSH:

Como se puede ver, aparece el usuario conectado y con el mismo PID aparecen todos los archivos abiertos por `sshd`

Apartado C

Para mostrar los archivos que usa un proceso concreto, es necesario referenciarlo con su PID. Para ello es necesario usar el siguiente comando: `lsof -p PID`.

Y ahora para ver los archivos que esta usando un usuario concreto, se debe usar el switch "-u": `lsof -u usuario`

Por ultimo, para obtener los archivos que tiene abiertos un proceso **Y** un usuario, es necesario usar el switch adicional "-a". Esto es debido a que por defecto solo busca, en caso de haber varios switches, utilizando un criterio **OR**. Comando: `lsof -u usuario -p PID -a`

Ejercicio 2

Apartado A

Para ver que vulnerabilidades hay en el sistema es neceasrio instalar el paquete `lynis` junto al comando `lynis audit system`.

Y las posibles vulnerabilidades son las siguientes:

Como se puede ver, solo hay dos avisos. Suponiendo que es una maquina para desarrollar aplicaciones, voy a listar los grados de severidad:

- **Found one or more vulnerable packages. [PKGS-7392]** → Severidad: **Alta**. Puede llegar a ser muy peligroso, ya que pueden ser vulnerabilidades que potencialmente le otorguen acceso root al sistema.

  Solucion: Para solucionarlo, es necesario actualizar todos los paquetes del sistema con la orden (en Ubuntu y en distros basadas en Debian) `sudo apt upgrade`.

- **iptables module(s) loaded, but no rules active [FIRE-4512]** → Severidad: **Alta**. `iptables` es un paquete que se utiliza principalmente junto a un firewall para permitir/-bloquear cierto trafico. Si fuera una compañia importante sin firewall, podria darse el caso de que alguien entrase en el sistema y obtuviese datos sin permiso, produciendo asi un "leak" o incluso chantaje.

  Solucion: La solucion es habilitar el firewall y aplicarle las reglas que sean necesarias. En Ubuntu viene instalado por defecto `ufw`, pero viene deshabilitado por defecto. Para habilitarlo hay que poner: `sudo ufw enable` y con la orden `sudo ufw status verbose` se pueden ver las reglas (por defecto prohibe trafico entrante y permite trafico saliente, prohibiendo asi conexiones del tipo SSH).

Ahora, ejecutando de nuevo `lynis audit system` aparece la siguiente puntuacion:

Y al ver los warnings se ve que no aparece ninguno:

Por tanto, a nivel de advertencias el sistema ya está "seguro" (nunca se puede decir con seguridad). En cuanto a las sugerencias, las principales son para reforzar SSH y el uso de bloqueadores de IP como "fail2ban". No son fallos demasiado críticos.

Apartado B MAL

Con el comando `lynis show tests` podemos obtener todos los tests que realiza:
Y se puede ver que el codigo de uno de ellos es "MALW-*"
Ahroa con el comando `lynis show details MALW-3280`
Por tanto, la lista de antivirus que escanea son: Avast, Avira, epagd, CrowdStrike, CylanceSvc, esets_daemon, Kapersky, McAffee, savscand, SophosScanD, rtvscand, Symantec management client service, Symantec Endpoint Protection configuration service, synoavd, Trend Micro Deep Anti Malware component y TmccMac to test for Trend Micro anti-virus (macOS)
Además, con la orden `lynis show tests` también aparecen en la seccion "MALW" algunos antivirus extras, estos son: ClamAV, Rootkit Hunter, LMD y chkrootkit
Ahora instalaré "unhide" con la orden `sudo apt install unhide`. Una vez hecho esto, seguirá saliendo en el report que no tengo antivirus.

Apartado B

Lynis permite añadir nuevos tests o modificar existentes para añadirles más funcionalidad. Todo esto se realiza mediante los archivos que se encuentran en el directorio `/usr/share/lynis/include`.
En este caso, para poder ver los antivirus que detecta actualmente es necesario inspeccionar el archivo `/usr/share/lynis/include/tests_malware`:
Como se puede ver, detecta los siguientes antivirus:

- Avast
- Avira
- Bitdefender
- ClamAV (clamd, clamscan y freshclam)
- CrowdStrike
- ESET
- Kapersky
- McAffee
- chkrootkit
- rkhunter
- LMD
- CylanceSvc
- SophosScanD
- Symantec
- Synology Antivirus Essential
- Trend Micro Anti Malware for Linux

Ahora voy a instalar el programa "unhide", el cual no es detectado por Lynis:
AHora, modifico el archivo anterior y añado la macro:
Y añadimos un nuvo "if" en la cadena de "if" del test "MALW-3280":
Y ahora al pasar el test ya aparece como que existe un antivirus:
Y si desinstalo `unhide` aparece como que no hay ningún antivirus instalado:

Ejercicio 3

Para instalar la herramienta en Ubuntu se ejecuta la orden: `sudo apt install rkhunter`.

Apartado A

Para realizar el analisis es necesario ejecutar el comando: `sudo rkhunter --check`

Como se puede ver en los resultados, aparece que hay alguna advertencia. Ahora, revisando el archivo `/var/log/rkhunter.log` y buscando la palabra "Warning".

La primera advertencia tiene que ver con la posible modificacion del binario `lwp-request`, ne caso de ser una modificacion malintencionada, un atacante podria modificar el binario para recibir todos los datos que se envian o reciben a traves de el (por ejemplo, si se usa SSH, poner un keylogger para obtener las contraseñas).

El segundo error tiene que ver con la seguridad de la configuracion SSH, ya que el parametro "PermitRootLogin" no esta puesto a ninugn valro y por defecto puede ser "Yes", dando la posibilidad de que un atacante pueda a entrar al sistema por SSH como usuario root.

Apartado B

El primer error se puedde solucionar cambiando en `/etc/rkhunter.log` el macro `PKGMGR` (por defecto esta a "NONE"), en el caso de Ubuntu y Debian se debe cambiar a "DPKG". Este cambio hace que coteje con los hashes de cada paquete para ver si ha sido modificado malintencionadamente:

Una vez realizado este cambio, se debe ejecutar el comando `sudo rkhunter --propupd`.

Todo esto se puede solucionar cambiando en el archivo de configuracion `/etc/ssh/sshd_config` y poniendo el parametro a "no":

Ademas, es necesario reinicar el servicio con el comando `systemctl restart ssh`.

Por ultimo, para comprobar que el sistema ya no tiene mas advertencias, ejcutamos de nuevo la orden `sudo rkhunter --check`:

Y como se puede ver, le sistema ya es seguro, eran solo falsos positivos en el primer caso, y en el segundo una mala configuracion de un servicio importante.