

SSO Práctica 1 Sesión 2

Andrés Merlo Trujillo

Índice

Ejercicio 1	2
Apartado A	2
sshd	2
avahi-daemon	2
Apartado B	3
Apartado C	4
Ejercicio 2	6
Apartado A	6
Apartado B	8
Ejercicio 3	12
Apartado A	12
Apartado B	13

Ejercicio 1

Apartado A

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

```
root@andres-kvm:~# lsdf -i
COMMAND  PID    USER    FD  TYPE DEVICE SIZE/OFF NODE NAME
systemd-r 1092  systemd-resolve 13u IPv4 21232 0t0  UDP localhost:domain
systemd-r 1092  systemd-resolve 14u IPv4 21233 0t0  TCP localhost:domain (LISTEN)
avahi-dae 1144  avahi    12u IPv4 22665 0t0  UDP *:mdns
avahi-dae 1144  avahi    13u IPv6 22666 0t0  UDP *:mdns
avahi-dae 1144  avahi    14u IPv4 22667 0t0  UDP *:49676
avahi-dae 1144  avahi    15u IPv6 22668 0t0  UDP *:53167
NetworkMa 1147  root     25u IPv4 24614 0t0  UDP andres-kvm:bootpc->_gateway:bootps
sshd      1319  root     3u  IPv4 22997 0t0  TCP *:ssh (LISTEN)
sshd      1319  root     4u  IPv6 23008 0t0  TCP *:ssh (LISTEN)
cupsd     1368  root     6u  IPv6 23631 0t0  TCP ip6-localhost:ipp (LISTEN)
cupsd     1368  root     7u  IPv4 23632 0t0  TCP localhost:ipp (LISTEN)
cups-brow 1416  root     7u  IPv4 24645 0t0  UDP *:631
root@andres-kvm:~#
```

La orden ofrece 9 columnas con los siguientes significados:

- **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 conexión (en capa de red) (IPv4, IPv6, X.25, etc.).
- **DEVICE:** Número de dispositivo.
- **SIZE/OFF:** Tamaño del archivo.
- **NODE:** Número de nodo/inodo de un fichero o el protocolo en capa de transporte (TCP, UDP, ...).
- **NAME:** Punto de montaje y sistema de archivos que usa el archivo abierto. También puede significar la dirección 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 (FD 3 y 4. La letra “u” indica acceso de lectura y escritura)
- **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 ACK).
- **NAME:** *:ssh (LISTEN) (El asterisco indica que espera de cualquier IP, en el puerto ssh (configurable, por defecto el 22)).

avahi-daemon

- **COMMAND:** avahi-daemon (avahi-daemon)
- **PID:** 1144
- **USER:** avahi
- **FD:** 14u (FD 14. La letra “u” indica acceso de lectura y escritura)
- **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 quedaría así: `lsof -i :ssh`.

```
root@andres-kvm:~# lsof -i :ssh
COMMAND PID USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
sshd     1319 root   3u   IPv4  22997      0t0  TCP *:ssh (LISTEN)
sshd     1319 root   4u   IPv6  23008      0t0  TCP *:ssh (LISTEN)
root@andres-kvm:~#
```

Ahora mismo no hay nadie conectado, solo están los “daemons” a la escucha de peticiones de conexión. Si ahora me conecto desde otra máquina virtual a la de Ubuntu, la salida es la siguiente:

```
root@andres-kvm:~# lsof -i :ssh
COMMAND PID  USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
sshd     1319  root   3u   IPv4  22997      0t0  TCP *:ssh (LISTEN)
sshd     1319  root   4u   IPv6  23008      0t0  TCP *:ssh (LISTEN)
sshd     4868  root   4u   IPv4  54634      0t0  TCP andres-kvm:ssh->archlinux:57686 (ESTABLISHED)
sshd     4906  andres 4u   IPv4  54634      0t0  TCP andres-kvm:ssh->archlinux:57686 (ESTABLISHED)
root@andres-kvm:~#
```

Aparecen dos líneas nuevas y en el apartado NAME se ve que la conexión 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 información (escuchar) a “archlinux”.

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

```
root@andres-kvm: ~  
sshd 5556 andres mem REG 0,28 68552 554906 /usr/lib/x86_64-linux-gnu/libresolv.so.2  
sshd 5556 andres mem REG 0,28 22600 60368 /usr/lib/x86_64-linux-gnu/libkeyutils.so.1.9  
sshd 5556 andres mem REG 0,28 52080 60384 /usr/lib/x86_64-linux-gnu/libkrb5support.so.0.1  
sshd 5556 andres mem REG 0,28 182928 60364 /usr/lib/x86_64-linux-gnu/libk5crypto.so.3.1  
sshd 5556 andres mem REG 0,28 613064 60862 /usr/lib/x86_64-linux-gnu/libpcre2-8.so.0.10.4  
sshd 5556 andres mem REG 0,28 1296312 59778 /usr/lib/x86_64-linux-gnu/libgcrypt.so.20.3.4  
sshd 5556 andres mem REG 0,28 39024 59200 /usr/lib/x86_64-linux-gnu/libcap.so.2.44  
sshd 5556 andres mem REG 0,28 125152 60442 /usr/lib/x86_64-linux-gnu/liblz4.so.1.9.3  
sshd 5556 andres mem REG 0,28 841808 61844 /usr/lib/x86_64-linux-gnu/libzstd.so.1.4.8  
sshd 5556 andres mem REG 0,28 170456 60446 /usr/lib/x86_64-linux-gnu/liblzma.so.5.2.5  
sshd 5556 andres mem REG 0,28 27072 59196 /usr/lib/x86_64-linux-gnu/libcap-ng.so.0.0.0  
sshd 5556 andres mem REG 0,28 93280 60686 /usr/lib/x86_64-linux-gnu/libnsl.so.2.0.1  
sshd 5556 andres mem REG 0,28 2216304 554880 /usr/lib/x86_64-linux-gnu/libc.so.6  
sshd 5556 andres mem REG 0,28 18504 559016 /usr/lib/x86_64-linux-gnu/libcom_err.so.2.1  
sshd 5556 andres mem REG 0,28 828000 60380 /usr/lib/x86_64-linux-gnu/libkrb5.so.3.3  
sshd 5556 andres mem REG 0,28 338712 60004 /usr/lib/x86_64-linux-gnu/libgssapi_krb5.so.2.2  
sshd 5556 andres mem REG 0,28 166280 61166 /usr/lib/x86_64-linux-gnu/libselinux.so.1  
sshd 5556 andres mem REG 0,28 198664 59314 /usr/lib/x86_64-linux-gnu/libcrypt.so.1.1.0  
sshd 5556 andres mem REG 0,28 108936 557818 /usr/lib/x86_64-linux-gnu/libz.so.1.2.11  
sshd 5556 andres mem REG 0,28 4447536 557770 /usr/lib/x86_64-linux-gnu/libcrypto.so.3  
sshd 5556 andres mem REG 0,28 807936 555482 /usr/lib/x86_64-linux-gnu/libsystemd.so.0.32.0  
sshd 5556 andres mem REG 0,28 67736 60780 /usr/lib/x86_64-linux-gnu/libpam.so.0.85.1  
sshd 5556 andres mem REG 0,28 133200 59010 /usr/lib/x86_64-linux-gnu/libaudit.so.1.0.0  
sshd 5556 andres mem REG 0,28 44872 61676 /usr/lib/x86_64-linux-gnu/libwrap.so.0.7.6  
sshd 5556 andres mem REG 0,28 51928 595004 /usr/lib/x86_64-linux-gnu/security/pam_sss.so  
sshd 5556 andres mem REG 0,28 240936 554874 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2  
sshd 5556 andres 0u CHR 1,3 0t0 5 /dev/null  
sshd 5556 andres 1u CHR 1,3 0t0 5 /dev/null  
sshd 5556 andres 2u CHR 1,3 0t0 5 /dev/null  
sshd 5556 andres 3u unix 0xffff8b23c94c4440 0t0 58967 type=DGRAM  
sshd 5556 andres 4u IPv4 59598 0t0 TCP andres-kvm:ssh->archlinux:40436 (ESTABLISHED)  
sshd 5556 andres 5u unix 0xffff8b238b712a80 0t0 59672 type=STREAM  
sshd 5556 andres 6u unix 0xffff8b23c94c5980 0t0 58950 type=STREAM  
sshd 5556 andres 7u CHR 0t0 87 /dev/ptmx  
sshd 5556 andres 8w FIFO 0,26 0t0 2470 /run/systemd/sessions/15.ref  
sshd 5556 andres 10u CHR 5,2 0t0 87 /dev/ptmx  
sshd 5556 andres 11u CHR 5,2 0t0 87 /dev/ptmx  
root@andres-kvm: ~#
```

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: `lsuf -p PID`.

```
root@andres-kvm: ~# lsuf -p 4029  
lsuf: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1000/gvfs  
Output information may be incomplete.  
lsuf: WARNING: can't stat() fuse.portal file system /run/user/1000/doc  
Output information may be incomplete.  
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME  
bash 4029 root cwd DIR 0,38 10 34 /root  
bash 4029 root rtd DIR 0,28 25 34 /  
bash 4029 root txt REG 0,28 1396520 6378 /usr/bin/bash  
bash 4029 root mem REG 0,28 8391520 476654 /usr/lib/locale/locale-archive  
bash 4029 root mem REG 0,28 2216304 554880 /usr/lib/x86_64-linux-gnu/libc.so.6  
bash 4029 root mem REG 0,28 200136 61382 /usr/lib/x86_64-linux-gnu/libtinfo.so.6.3  
bash 4029 root mem REG 0,28 27002 555412 /usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache  
bash 4029 root mem REG 0,28 240936 554874 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2  
bash 4029 root 0u CHR 136,3 0t0 6 /dev/pts/3  
bash 4029 root 1u CHR 136,3 0t0 6 /dev/pts/3  
bash 4029 root 2u CHR 136,3 0t0 6 /dev/pts/3  
bash 4029 root 255u CHR 136,3 0t0 6 /dev/pts/3  
root@andres-kvm: ~#
```

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

```

root@andres-kvm: ~
gjs 4463 andres mem REG 0,28 1306280 58560 /usr/lib/x86_64-linux-gnu/libX11.so.6.4.0
gjs 4463 andres mem REG 0,28 42920 59168 /usr/lib/x86_64-linux-gnu/libcairo-gobject.so.2.11600.0
gjs 4463 andres mem REG 0,28 1203976 59176 /usr/lib/x86_64-linux-gnu/libcairo.so.2.11600.0
gjs 4463 andres mem REG 0,28 335936 61056 /usr/lib/x86_64-linux-gnu/libreadline.so.8.1
gjs 4463 andres mem REG 0,28 12184728 503084 /usr/lib/x86_64-linux-gnu/libmozjs-91.so.91.10.0
gjs 4463 andres mem REG 0,28 47688 59626 /usr/lib/x86_64-linux-gnu/libbfti.so.8.1.0
gjs 4463 andres mem REG 0,28 137288 59846 /usr/lib/x86_64-linux-gnu/libgirepository-1.0.so.1.0.0
gjs 4463 andres mem REG 0,28 2216304 554880 /usr/lib/x86_64-linux-gnu/libc.so.6
gjs 4463 andres mem REG 0,28 125488 503436 /usr/lib/x86_64-linux-gnu/libgcc_s.so.1
gjs 4463 andres mem REG 0,28 2252096 503310 /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.30
gjs 4463 andres mem REG 0,28 1920400 59838 /usr/lib/x86_64-linux-gnu/libgio-2.0.so.0.7200.1
gjs 4463 andres mem REG 0,28 387464 59926 /usr/lib/x86_64-linux-gnu/libgobject-2.0.so.0.7200.1
gjs 4463 andres mem REG 0,28 1277712 59858 /usr/lib/x86_64-linux-gnu/libglib-2.0.so.0.7200.1
gjs 4463 andres mem REG 0,28 1299592 573548 /usr/lib/x86_64-linux-gnu/libgjs.so.0.0.0
gjs 4463 andres mem REG 0,28 1668 64092 /usr/lib/x86_64-linux-gnu/girepository-1.0/GModule-2.0.typelib
gjs 4463 andres mem REG 0,28 14344 64246 /usr/lib/x86_64-linux-gnu/girepository-1.0/cairo-1.0.typelib
gjs 4463 andres mem REG 0,28 5384 573546 /usr/lib/x86_64-linux-gnu/gjs/girepository-1.0/GjsPrivate-1.0.typelib
gjs 4463 andres mem REG 0,28 27002 555412 /usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache
gjs 4463 andres mem REG 0,28 240936 554874 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
gjs 4463 andres mem REG 0,28 836 64258 /usr/lib/x86_64-linux-gnu/girepository-1.0/xlib-2.0.typelib
gjs 4463 andres 0r CHR 1,3 0t0 5 /dev/null
gjs 4463 andres 1w FIFO 0,13 0t0 51802 pipe
gjs 4463 andres 2w FIFO 0,13 0t0 51802 pipe
gjs 4463 andres 3u unix 0xffff8b2387db6200 0t0 51801 type=STREAM
gjs 4463 andres 4u a_inode 0,14 0 12443 [eventfd]
gjs 4463 andres 5u a_inode 0,14 0 12443 [eventfd]
gjs 4463 andres 6u unix 0xffff8b23c3216a80 0t0 52493 type=STREAM
gjs 4463 andres 7u a_inode 0,14 0 12443 [eventfd]
gjs 4463 andres 8u a_inode 0,14 0 12443 [eventfd]
gjs 4463 andres 9u REG 0,1 2356992 18685 /memfd:wayland-cursor (deleted)
gjs 4463 andres 10u unix 0xffff8b23c3217300 0t0 52494 type=STREAM
gjs 4463 andres 11u unix 0xffff8b23c9ba2200 0t0 52495 type=STREAM
gjs 4463 andres 12u a_inode 0,14 0 12443 [eventfd]
gjs 4463 andres 13u unix 0xffff8b23c5a42200 0t0 52497 type=STREAM
gjs 4463 andres 14r a_inode 0,14 0 12443 inotify
gjs 4463 andres 18r REG 0,45 444 4198 /home/andres/.local/share/gvfs-metadata/home
gjs 4463 andres 19r REG 0,45 32768 4200 /home/andres/.local/share/gvfs-metadata/home-5b6c4650.log
root@andres-kvm:~#

```

Figura 1: Salida de “lsuf -u andres”, se puede ver que en la tercera columna solo aparece ese usuario.

Por último, 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: `lsuf -u usuario -p PID -a`

```

root@andres-kvm: ~
root@andres-kvm:~# lsuf -p 4029 -u root -a
lsuf: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1000/gvfs
Output information may be incomplete.
lsuf: WARNING: can't stat() fuse.portal file system /run/user/1000/doc
Output information may be incomplete.
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
bash 4029 root cwd DIR 0,38 10 34 /root
bash 4029 root rtd DIR 0,28 25 34 /
bash 4029 root txt REG 0,28 1396520 6378 /usr/bin/bash
bash 4029 root mem REG 0,28 8391520 476654 /usr/lib/locale/locale-archive
bash 4029 root mem REG 0,28 2216304 554880 /usr/lib/x86_64-linux-gnu/libc.so.6
bash 4029 root mem REG 0,28 200136 61382 /usr/lib/x86_64-linux-gnu/libtinfo.so.6.3
bash 4029 root mem REG 0,28 27002 555412 /usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache
bash 4029 root mem REG 0,28 240936 554874 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
bash 4029 root 0u CHR 136,3 0t0 6 /dev/pts/3
bash 4029 root 1u CHR 136,3 0t0 6 /dev/pts/3
bash 4029 root 2u CHR 136,3 0t0 6 /dev/pts/3
bash 4029 root 255u CHR 136,3 0t0 6 /dev/pts/3
root@andres-kvm:~#

```

Figura 2: Archivos abiertos por el PID 4029 **Y** el usuario root

Ejercicio 2

Apartado A

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

```
root@andres-kvm: ~  
=====
```

Lynis security scan details:

Hardening index : 59 [#####]
Tests performed : 255
Plugins enabled : 1

Components:

- Firewall [V]
- Malware scanner [X]

Scan mode:

Normal [V] Forensics [] Integration [] Pentest []

Lynis modules:

- Compliance status [?]
- Security audit [V]
- Vulnerability scan [V]

Files:

- Test and debug information : /var/log/lynis.log
- Report data : /var/log/lynis-report.dat

```
=====
```

Lynis 3.0.7

Auditing, system hardening, and compliance for UNIX-based systems
(Linux, macOS, BSD, and others)

2023-08-26 15:56:56 - https://cisofy.com/lynis/

Y las posibles vulnerabilidades son las siguientes:

```
root@andres-kvm: ~  
-[ Lynis 3.0.7 Results ]-
```

Warnings (2):

```
-----
```

- ! Found one or more vulnerable packages. [PKGS-7392]
<https://cisofy.com/lynis/controls/PKGS-7392/>
- ! iptables module(s) loaded, but no rules active [FIRE-4512]
<https://cisofy.com/lynis/controls/FIRE-4512/>

Suggestions (52):

```
-----
```

- * This release is more than 4 months old. Check the website or GitHub to see if there is an update available. [LYNIS]
<https://cisofy.com/lynis/controls/LYNIS/>
- * Install libpam-tmpdir to set \$TMP and \$TMPDIR for PAM sessions [DEB-0280]
<https://cisofy.com/lynis/controls/DEB-0280/>
- * Install apt-listbugs to display a list of critical bugs prior to each APT installation. [DEB-0810]
<https://cisofy.com/lynis/controls/DEB-0810/>
- * Install apt-listchanges to display any significant changes prior to any upgrade via APT. [DEB-0811]
<https://cisofy.com/lynis/controls/DEB-0811/>
- * Install needrestart, alternatively to debian-goodies, so that you can run needrestart after upgrades to determine which daemons are using old versions of libraries and need restarting. [DEB-0831]
<https://cisofy.com/lynis/controls/DEB-0831/>
- * Install fail2ban to automatically ban hosts that commit multiple authentication errors. [DEB-0880]
<https://cisofy.com/lynis/controls/DEB-0880/>
- * Set a password on GRUB boot loader to prevent altering boot configuration (e.g. boot in single user mode without password). [DEB-0907]
<https://cisofy.com/lynis/controls/DEB-0907/>

Como se puede ver, solo hay dos avisos. Suponiendo que es una máquina 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.

Solución: 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 o bloquear cierto tráfico. Si fuera una compañía importante sin firewall, podría darse el caso de que alguien entrase en el sistema y obtuviese datos sin permiso, produciendo así un “leak” o incluso pidiendo un rescate para no publicar la información (que en este caso podrían ser aplicaciones que no se desean que se publiquen aún).

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

```
root@andres-kvm:~# ufw status verbose
Estado: activo
Acceso: on (low)
Predeterminado: deny (entrantes), allow (salientes), disabled (enrutados)
Perfiles nuevos: skip
root@andres-kvm:~#
```

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

```
=====
Lynis security scan details:
Hardening index : 65 [##### ]
Tests performed : 255
Plugins enabled : 1

Components:
- Firewall           [V]
- Malware scanner    [X]

Scan mode:
Normal [V] Forensics [ ] Integration [ ] Pentest [ ]

Lynis modules:
- Compliance status [?]
- Security audit    [V]
- Vulnerability scan [V]

Files:
- Test and debug information : /var/log/lynis.log
- Report data                : /var/log/lynis-report.dat

=====

Lynis 3.0.7

Auditing, system hardening, and compliance for UNIX-based systems
(Linux, macOS, BSD, and others)

2027-2021-01-01 01:00:00 https://linuxfromp.org/lynis/
```

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

```
root@andres-kvm: ~  
=====
```

```
-[ Lynis 3.0.7 Results ]-  
  
Great, no warnings  
  
Suggestions (52):  
-----  
* This release is more than 4 months old. Check the website or GitHub to see if there is an update available. [LYNIS]  
  https://cisofy.com/lynis/controls/LYNIS/  
  
* Install libpam-tmpdir to set $TMP and $TMPDIR for PAM sessions [DEB-0280]  
  https://cisofy.com/lynis/controls/DEB-0280/  
  
* Install apt-listbugs to display a list of critical bugs prior to each APT installation. [DEB-0810]  
  https://cisofy.com/lynis/controls/DEB-0810/  
  
* Install apt-listchanges to display any significant changes prior to any upgrade via APT. [DEB-0811]  
  https://cisofy.com/lynis/controls/DEB-0811/  
  
* Install needrestart, alternatively to debian-goodies, so that you can run needrestart after upgrades to determine which  
  daemons are using old versions of libraries and need restarting. [DEB-0831]  
  https://cisofy.com/lynis/controls/DEB-0831/  
  
* Install fail2ban to automatically ban hosts that commit multiple authentication errors. [DEB-0880]  
  https://cisofy.com/lynis/controls/DEB-0880/  
  
* Set a password on GRUB boot loader to prevent altering boot configuration (e.g. boot in single user mode without password) [BOOT-5122]  
  https://cisofy.com/lynis/controls/BOOT-5122/  
  
* Consider hardening system services [BOOT-5264]
```

Por tanto, en cuanto a advertencias el sistema ya está “seguro” (nunca se puede decir con total 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

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`.

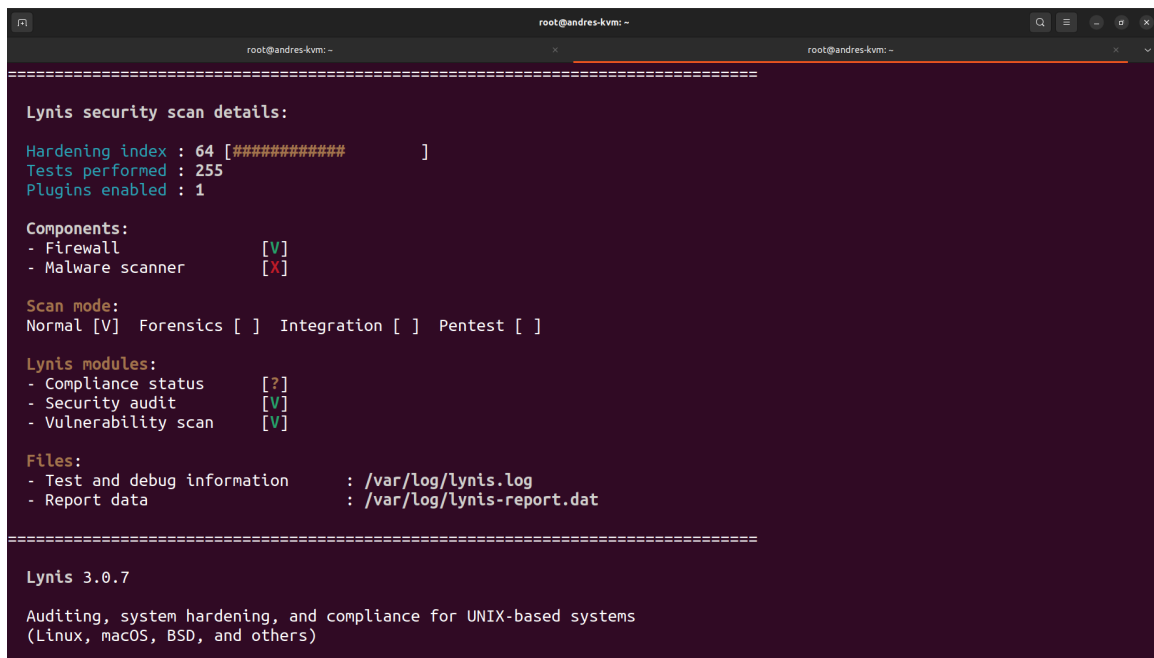
En este caso, para poder ver los antivirus que detecta actualmente es necesario inspeccionar el archivo `/usr/share/lynis/include/tests_malware`:

```
root@andres-kvm: ~  
22  #  
23  #####  
24  #  
25  InsertSection "${SECTION_MALWARE}"  
26  #  
27  #####  
28  #  
29  AVAST_DAEMON_RUNNING=0  
30  AVIRA_DAEMON_RUNNING=0  
31  BITDEFENDER_DAEMON_RUNNING=0  
32  CLAMD_RUNNING=0  
33  CLAMSCAN_INSTALLED=0  
34  CROWDSTRIKE_FALCON_SENSOR_RUNNING=0  
35  ESET_DAEMON_RUNNING=0  
36  FRESHCLAM_DAEMON_RUNNING=0  
37  KASPERSKY_SCANNER_RUNNING=0  
38  MCAFEE_SCANNER_RUNNING=0  
39  MALWARE_SCANNER_INSTALLED=0  
40  MALWARE_DAEMON_RUNNING=0  
41  ROOTKIT_SCANNER_FOUND=0  
42  SOPHOS_SCANNER_RUNNING=0  
43  SYMANTEC_SCANNER_RUNNING=0  
44  SYNOLOGY_DAEMON_RUNNING=0  
45  TRENDMICRO_DSA_DAEMON_RUNNING=0  
46  #  
47  #####  
:
```


Como se puede ver, detecta los siguientes antivirus:

- Avast
- Avira
- Bitdefender
- ClamAV (clamd, clamscan y freshclam)
- CrowdStrike
- ESET
- Kaspersky
- McAfee
- 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:



```
root@andres-kvm: ~  
=====
```

```
Lynis security scan details:  
  
Hardening index : 64 [##### ]  
Tests performed : 255  
Plugins enabled : 1  
  
Components:  
- Firewall [V]  
- Malware scanner [X]  
  
Scan mode:  
Normal [V] Forensics [ ] Integration [ ] Pentest [ ]  
  
Lynis modules:  
- Compliance status [?]  
- Security audit [V]  
- Vulnerability scan [V]  
  
Files:  
- Test and debug information : /var/log/lynis.log  
- Report data : /var/log/lynis-report.dat  
  
=====
```

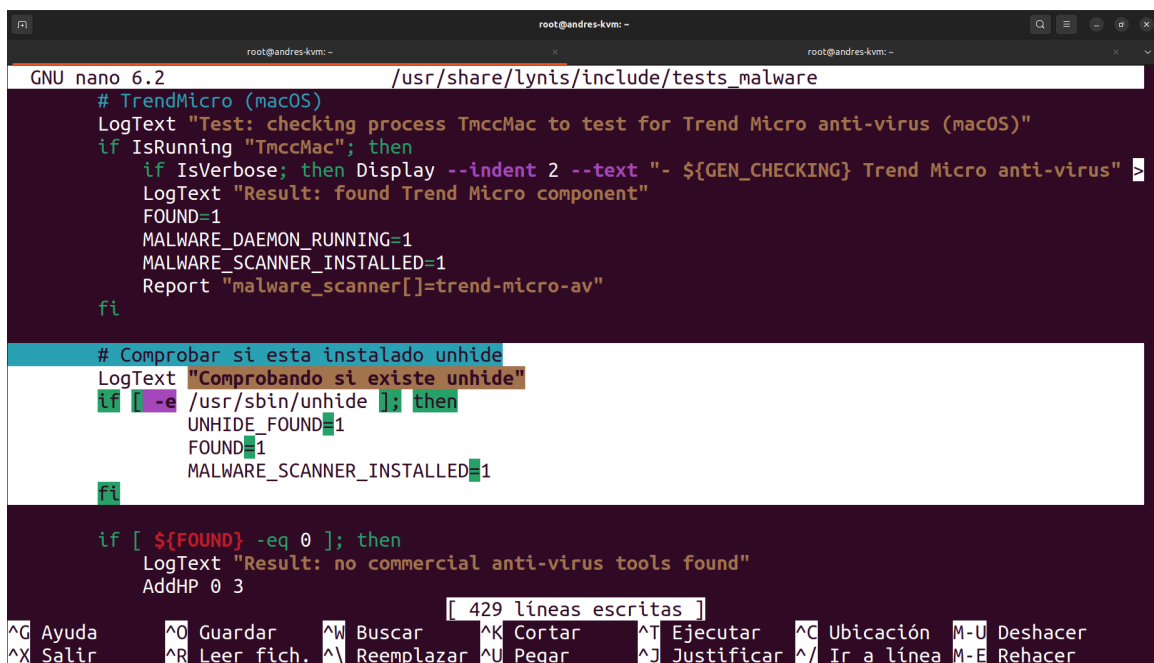
```
Lynis 3.0.7  
  
Auditing, system hardening, and compliance for UNIX-based systems  
(Linux, macOS, BSD, and others)
```

Figura 3: Indica que no hay un antivirus, cuando realmente hay un tipo de protección (unhide).

Ahora, modifico el archivo anterior y añado la macro:

```
AVAST_DAEMON_RUNNING=0
AVIRA_DAEMON_RUNNING=0
BITDEFENDER_DAEMON_RUNNING=0
CLAMD_RUNNING=0
CLAMSCAN_INSTALLED=0
CROWDSTRIKE_FALCON_SENSOR_RUNNING=0
ESET_DAEMON_RUNNING=0
FRESHCLAM_DAEMON_RUNNING=0
KASPERSKY_SCANNER_RUNNING=0
MCAFFEE_SCANNER_RUNNING=0
MALWARE_SCANNER_INSTALLED=0
MALWARE_DAEMON_RUNNING=0
ROOTKIT_SCANNER_FOUND=0
SOPHOS_SCANNER_RUNNING=0
SYMANTEC_SCANNER_RUNNING=0
SYNOLOGY_DAEMON_RUNNING=0
TRENDMICRO_DSA_DAEMON_RUNNING=0
UNHIDE_FOUND=0
```

Y añadimos un nuevo “if” en la cadena del test “MALW-3280”:



```
GNU nano 6.2 /usr/share/lynis/include/tests_malware
# TrendMicro (macOS)
LogText "Test: checking process TmccMac to test for Trend Micro anti-virus (macOS)"
if IsRunning "TmccMac"; then
    if IsVerbose; then Display --indent 2 --text "- ${GEN_CHECKING} Trend Micro anti-virus"
    LogText "Result: found Trend Micro component"
    FOUND=1
    MALWARE_DAEMON_RUNNING=1
    MALWARE_SCANNER_INSTALLED=1
    Report "malware_scanner[]=trend-micro-av"
fi

# Comprobar si esta instalado unhide
LogText "Comprobando si existe unhide"
if [ -e /usr/sbin/unhide ]; then
    UNHIDE_FOUND=1
    FOUND=1
    MALWARE_SCANNER_INSTALLED=1
fi

if [ ${FOUND} -eq 0 ]; then
    LogText "Result: no commercial anti-virus tools found"
    AddHP 0 3
[ 429 líneas escritas ]
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación  M-U Deshacer
^X Salir      ^R Leer fich. ^_ Reemplazar  ^U Pegar      ^J Justificar ^/ Ir a línea  M-E Rehacer
```

Ahora al pasar el test ya aparece como que existe un antivirus:

```
root@andres-kvm: ~  
Components:  
- Firewall [V]  
- Malware scanner [V]  
  
Scan mode:  
Normal [V] Forensics [ ] Integration [ ] Pentest [ ]  
  
Lynis modules:  
- Compliance status [?]  
- Security audit [V]  
- Vulnerability scan [V]  
  
Files:  
- Test and debug information : /var/log/lynis.log  
- Report data : /var/log/lynis-report.dat  
  
=====
```

Lynis 3.0.7

Auditing, system hardening, and compliance for UNIX-based systems
(Linux, macOS, BSD, and others)

2007-2021, CISofy - <https://cisofy.com/lynis/>
Enterprise support available (compliance, plugins, interface and tools)

=====

[TIP]: Enhance Lynis audits by adding your settings to custom.prfl (see /etc/lynis/default.prfl for all settings)

```
root@andres-kvm:~#
```

Figura 4: Se puede observar el resultado en la tercera línea.

Y si desinstalo unhide aparece como que no hay ningún antivirus instalado:

```
root@andres-kvm: ~  
Components:  
- Firewall [V]  
- Malware scanner [X]  
  
Scan mode:  
Normal [V] Forensics [ ] Integration [ ] Pentest [ ]  
  
Lynis modules:  
- Compliance status [?]  
- Security audit [V]  
- Vulnerability scan [V]  
  
Files:  
- Test and debug information : /var/log/lynis.log  
- Report data : /var/log/lynis-report.dat  
  
=====
```

Lynis 3.0.7

Auditing, system hardening, and compliance for UNIX-based systems
(Linux, macOS, BSD, and others)

2007-2021, CISofy - <https://cisofy.com/lynis/>
Enterprise support available (compliance, plugins, interface and tools)

=====

[TIP]: Enhance Lynis audits by adding your settings to custom.prfl (see /etc/lynis/default.prfl for all settings)

```
root@andres-kvm:~#
```

Figura 5: Se puede observar el resultado en la tercera línea.

Ejercicio 3

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

Apartado A

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

```
root@andres-kvm: ~  
[Press <ENTER> to continue]  
  
System checks summary  
=====
```

File properties checks...

```
Files checked: 143  
Suspect files: 1
```

Rootkit checks...

```
Rootkits checked : 498  
Possible rootkits: 0
```

Applications checks...

```
All checks skipped
```

The system checks took: 1 minute and 13 seconds

All results have been written to the log file: /var/log/rkhunter.log

One or more warnings have been found while checking the system.
Please check the log file (/var/log/rkhunter.log)

```
root@andres-kvm:~#
```

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”.

```
339 [20:45:14] /usr/bin/lwp-request [ Warning ]  
340 [20:45:14] Warning: The command '/usr/bin/lwp-request' has been replaced by a script: /usr/b  
in/lwp-request: Perl script text executable
```

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

```
1846 [20:46:12] Checking if SSH root access is allowed [ Warning ]  
1847 [20:46:12] Warning: The SSH configuration option 'PermitRootLogin' has not been set.  
1848 The default value may be 'yes', to allow root access.
```

La segunda advertencia tiene que ver con la seguridad de la configuración SSH, ya que el parámetro “PermitRootLogin” no está puesto a ningún valor y por defecto puede ser “Yes”, dando la posibilidad de que un atacante pueda entrar al sistema por SSH como usuario root.

Apartado B

El primer error se puede 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 han sido modificados malintencionadamente:

```
root@andres-kvm: ~  
GNU nano 6.2 /etc/rkhunter.conf  
#  
# The default value is 'NONE'.  
#  
# Also see the PKGMGR_NO_VRFY and USE_SUNSUM options.  
#  
# NONE is the default for Debian as well, as running --propupd takes  
# about 4 times longer when it's set to DPKG  
#  
PKGMGR=DPKG
```

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

Todo esto se puede solucionar cambiando en el archivo de configuración `/etc/ssh/sshd_config` y poniendo el parámetro a “no”:

```
root@andres-kvm: ~  
GNU nano 6.2 /etc/ssh/sshd_config  
# Ciphers and keying  
#RekeyLimit default none  
  
# Logging  
#SyslogFacility AUTH  
#LogLevel INFO  
  
# Authentication:  
  
#LoginGraceTime 2m  
PermitRootLogin no
```

Además, es necesario reiniciar el servicio con el comando `systemctl restart ssh`.

Por último, para comprobar que el sistema ya no tiene más advertencias, ejecutamos de nuevo la orden `sudo rkhunter --check`:

```
root@andres-kvm: ~  
Checking for hidden files and directories [ None found ]  
[Press <ENTER> to continue]  
  
System checks summary  
=====  
File properties checks...  
  Files checked: 143  
  Suspect files: 0  
  
Rootkit checks...  
  Rootkits checked : 498  
  Possible rootkits: 0  
  
Applications checks...  
  All checks skipped  
  
The system checks took: 4 minutes and 7 seconds  
  
All results have been written to the log file: /var/log/rkhunter.log  
  
No warnings were found while checking the system.  
  
root@andres-kvm:~#
```

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