SEGURIDAD EN SISTEMAS OPERATIVOS
4º Grado en Informática – Complementos de Ing. del Software
Curso 2020-21

Práctica [1]. .....

Sesión [2]. .....

Autor[1]: ...José Armando Albarado Mamani...

Ejercicio 1.

a) Utiliza esta herramienta para conocer que procesos/servicios de nuestro sistema están accediendo a la red o tiene archivos abiertos. Indicar algunos de los servicios que tenéis activos, es decir, la actividad de la red, indicando que información da la herramienta.

Introduciendo **lsof** en mi terminal obtengo:

```
spectac:d  5864  5882            jose   24r      REG                8,3     220
88   12191241 /usr/share/icons/hicolor/icon-theme.cache
spectac:d  5864  5882            jose   25r      REG                8,3    1455
80   11274001 /usr/share/mime/mime.cache
spectac:d  5864  5882            jose   26r  a_inode               0,14
 0      11434 inotify
spectac:d  5864  5882            jose   27r      REG                8,3    6893
99    8665573 /home/jose/.cache/ksycoca5_es_plJrzQV0DuuJm3vrIbsk5PjmyXM=
spectac:d  5864  5883            jose  cwd       DIR                8,3      40
96    8650754 /home/jose
spectac:d  5864  5883            jose  rtd       DIR                8,3      40
96          2 /
spectac:d  5864  5883            jose  txt       REG                8,3    3380
64   10888181 /usr/bin/spectacle
spectac:d  5864  5883            jose  mem       REG                8,3    7570
76   11928208 /usr/share/fonts/truetype/dejavu/DejaVuSans.ttf
spectac:d  5864  5883            jose  DEL       REG               0,47
          886 /i915
spectac:d  5864  5883            jose  mem       REG                8,3     105
12    4456683 /usr/lib/x86_64-linux-gnu/qt5/qml/QtQuick/Window.2/libwindowplugin
.so
spectac:d  5864  5883            jose  mem       REG                8,3     105
20    4456687 /usr/lib/x86_64-linux-gnu/qt5/qml/QtQuick.2/libqtquick2plugin.so
spectac:d  5864  5883            jose  DEL       REG               0,47
          564 /i915
spectac:d  5864  5883            jose  DEL       REG               0,47
          360 /i915
```

Un lista demasiado grande para poder colocarlo completamente, es por ello

---

1 Como autor declaro que los contenidos del presente documento son originales y elaborados por mi. De no cumplir con este compromiso, soy consciente de que, de acuerdo con la "Normativa de evaluación y de calificaciones de los estudiantes de la Universidad de Granada" esto "conllevará la calificación numérica de cero … independientemente del resto de calificaciones que el estudiante hubiera obtenido ..."

que solo he puesto una pequeña parte. Investigando como mejorar la salida de **lsof** he decidido consultar el manual, para ello empleo **man lsof.** He encontrado lo siguiente:

En primer lugar la opción **-i:**

```
-i [i]   selects the listing of files any of whose Internet address
         matches the address specified in i.  If no address is
         specified, this option selects the listing of all Internet
         and x.25 (HP-UX) network files.
```

Otra opción bastante interesante es la opción **-n:**

```
-n       inhibits the conversion of network numbers to host names for
         network files.  Inhibiting conversion may make lsof run
         faster.  It is also useful when host name lookup is not
         working properly.
```

Por último he encontrado otra opción que sería bastante útil para lo que se solicita en este apartado, la opción **-P:**

```
-P       inhibits the conversion of port numbers to port names for
         network files.  Inhibiting the conversion may make lsof run
         a little faster.  It is also useful when port name lookup is
         not working properly.
```

En conclusión la opción **-i** nos permite ver todos los archivos y procesos/servicios que están conectados a internet. La opción **-n** nos permite evitar que las direcciones IPs con los nombres de dominio. La opción **-P** nos permite evitar que se resuelvan los nombres de los puertos. Juntando estas tres opciones especiales en mi máquina obtengo los resultados siguientes:

```
jose@jose-CX61-2PC:~$ lsof -i -n -P
COMMAND    PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
kdeconnec 3860 jose   21u  IPv6  46984      0t0  UDP *:1716
kdeconnec 3860 jose   22u  IPv6  46985      0t0  TCP *:1716 (LISTEN)
chrome    4387 jose   47u  IPv4  51032      0t0  UDP 224.0.0.251:5353
jose@jose-CX61-2PC:~$
```

De esta forma se obtiene lo pedido en el apartado. Estos son los procesos/archivos que están accediendo a la red o que tiene archivos abiertos. Como se puede ver en la imagen anterior uno de lo servicios es **chrome,** voy a verlo de forma más detallada para ello:

```
jose@jose-CX61-2PC:~$ lsof -i -n -P | grep chrome
chrome    4345 jose  148u  IPv4 119206      0t0  UDP 224.0.0.251:5353
chrome    4387 jose   24u  IPv4 116518      0t0  TCP 192.168.43.79:49962->74.125
.71.188:5228 (ESTABLISHED)
chrome    4387 jose   32u  IPv4 112475      0t0  UDP 224.0.0.251:5353
jose@jose-CX61-2PC:~$
```

La información que se obtiene siguiendo el orden de derecha a izquierda es el siguiente:

- **Primera columna**: Se trata del comando del servicio o servicio que se está listando.
- **Segunda columna**: Es el PID (Identificador del proceso) del servicio o proceso que se está listando.
- **Tercera columna:** Indica el usuario que ha ejecutado el servicio o proceso en cuestión.
- **Cuarta columna:** Indica el descriptor del fichero.
- **Quinta columna:** Indica el protocolo que se está ejecutando, como vemos es el **IPv4** protocolo para llevar a cabo una conexión a internet.
- **Sexta columna:** Indica el número de dispositivo.
- **Séptima columna:** Indica el tamaño del fichero.
- **Octava columna:** Indica el tipo de conexión que se está realizando, puede ser TCP o UDP.
- **Novena columna:** Indica las direcciones IP y el puerto del emisor y del receptor, en este caso el emisor sería mi ordenador y receptor aquel al que estamos conectados.

b) Qué órdenes y opciones darías para conocer que cuenta podría estar generando tráfico saliente malicioso de ssh y dónde se encuentra el archivo.

Como se ha mencionado anteriormente con el comando **lsof -i** se pueden ver todos los procesos o servicios que estén conectados a la red, si detectamos que efectivamente se está produciendo algún tipo de conexión a **ssh** cuando no debería haber ninguna, entonces podemos detallar más con la orden **lsof -i -n -P | grep ssh**, de esta forma podemos saber que usuario o usuarios están usando el servicio **ssh.** Para saber que fichero están abiertos actualmente por este servicio podemos usar la orden **-c** que según el manual:

```
-c c     selects the listing of files for processes executing the
         command that begins with the characters of c.  Multiple
         commands may be specified, using multiple -c options.  They
         are joined in a single ORed set before participating in AND
         option selection.
```

Por tanto si introducimos en consola **lsof -c ssh** podremos ver todos los archivos abiertos por el servicio **ssh.** De esta forma sabremos que ficheros está usando y con ello podremos gestionar dichos archivos ya sea para borrar, modificar, bloquear, etc.

c) Muestra los archivos a los que esta accediendo un proceso concreto y lo que están en uso por un usuario.

Como he mencionado anteriormente con la opción **-c** podremos saber a que archivos o ficheros está accediendo un determinado servicio, para comprobarlos voy a probar con el servicio **chrome:**

```
chrome  5656 jose   21u     sock                0,9       0t0    64763 protocol:
 UNIX
chrome  5656 jose   22r     FIFO               0,13       0t0    64764 pipe
chrome  5656 jose   23w     FIFO               0,13       0t0    64764 pipe
chrome  5656 jose   24u     unix 0x0000000000000000       0t0    65685 type=STRE
AM
chrome  5656 jose   25r      REG                8,3    276608  9315402 /home/jos
e/.config/google-chrome/Subresource Filter/Indexed Rules/27/9.17.0/Ruleset Data
chrome  5656 jose   26r      REG                8,3    407700 11928286 /usr/shar
e/fonts/truetype/liberation2/LiberationSans-Regular.ttf
chrome  5656 jose   27u     unix 0x0000000000000000       0t0    66794 type=STRE
AM
chrome  5656 jose   28r      REG                8,3    276608  9315402 /home/jos
e/.config/google-chrome/Subresource Filter/Indexed Rules/27/9.17.0/Ruleset Data
chrome  5656 jose   29u     unix 0x0000000000000000       0t0    64830 type=STRE
AM
chrome  5656 jose   30r      REG                8,3    785066  8716920 /home/jos
e/.config/google-chrome/Dictionaries/es-ES-3-0.bdic
chrome  5656 jose   31r      REG                0,5         0    57832 /proc/565
6/statm
chrome  5656 jose   32r      REG                0,5         0    57831 /proc/565
6/status
chrome  5656 jose   33u      REG                8,3     29152 13238427 /tmp/.com
.google.Chrome.KBUTvt (deleted)
chrome  5656 jose   34u      REG               0,26       144      368 /dev/shm/
.com.google.Chrome.UeT0am (deleted)
chrome  5656 jose   36r      REG               0,26   1048576      451 /dev/shm/
.com.google.Chrome.jiVeFt (deleted)
chrome  5656 jose   37u      REG               0,26       144      426 /dev/shm/
.com.google.Chrome.GrKTKF (deleted)
chrome  5656 jose   41r      REG                8,3    411180 11928283 /usr/shar
e/fonts/truetype/liberation2/LiberationSans-Bold.ttf
chrome  5656 jose   42r      REG                8,3    331992 11928209 /usr/shar
e/fonts/truetype/dejavu/DejaVuSansMono-Bold.ttf
chrome  5656 jose   43r      REG                8,3    340712 11928210 /usr/shar
e/fonts/truetype/dejavu/DejaVuSansMono.ttf
chrome  5656 jose   44r      REG                8,3    251932 11936831 /usr/shar
e/fonts/truetype/dejavu/DejaVuSansMono-Oblique.ttf
chrome  5656 jose   45r      REG                8,3    412876 11928285 /usr/shar
e/fonts/truetype/liberation2/LiberationSans-Italic.ttf
jose@jose-CX61-2PC:~$
```

El listado es bastante grande es por ello que solo he copiado el final, y como se puede ver perfectamente se listan aquellos archivos que están en uso por el servicio **chrome.**

Para conocer que archivos están siendo usados por un determinado usuario hago uso de la opción **-u,** que según los manuales:

```
-u s      selects the listing of files for the user whose login names
          or user ID numbers are in the comma-separated set s - e.g.,
          ``abe'', or ``548,root''.  (There should be no spaces in the
          set.)
```

Como se indica en la imagen la opción **-c** la podemos emplear para saber que archivos están siendo usados por un determinado usuario:

```
/x86_64-linux-gnu/ld-2.27.so
lsof      13102 jose    0u      CHR              136,0       0t0          3 /dev
/pts/0
lsof      13102 jose    1u      CHR              136,0       0t0          3 /dev
/pts/0
lsof      13102 jose    2u      CHR              136,0       0t0          3 /dev
/pts/0
lsof      13102 jose    3r      DIR                0,5         0          1 /pro
c
lsof      13102 jose    4r      DIR                0,5         0     136014 /pro
c/13102/fd
lsof      13102 jose    5w     FIFO               0,13       0t0     136019 pipe
lsof      13102 jose    6r     FIFO               0,13       0t0     136020 pipe
lsof      13103 jose  cwd       DIR                8,3      4096    8650754 /hom
e/jose
lsof      13103 jose  rtd       DIR                8,3      4096          2 /
lsof      13103 jose  txt       REG                8,3    163224   10879633 /usr
/bin/lsof
lsof      13103 jose  mem       REG                8,3     47568    9439429 /lib
/x86_64-linux-gnu/libnss_files-2.27.so
lsof      13103 jose  mem       REG                8,3     97176    9437378 /lib
/x86_64-linux-gnu/libnsl-2.27.so
lsof      13103 jose  mem       REG                8,3     47576    9439431 /lib
/x86_64-linux-gnu/libnss_nis-2.27.so
lsof      13103 jose  mem       REG                8,3     39744    9437379 /lib
/x86_64-linux-gnu/libnss_compat-2.27.so
lsof      13103 jose  mem       REG                8,3   5586544   10888179 /usr
/lib/locale/locale-archive
lsof      13103 jose  mem       REG                8,3    144976    9439532 /lib
/x86_64-linux-gnu/libpthread-2.27.so
lsof      13103 jose  mem       REG                8,3     14560    9437374 /lib
/x86_64-linux-gnu/libdl-2.27.so
lsof      13103 jose  mem       REG                8,3    464824    9442381 /lib
/x86_64-linux-gnu/libpcre.so.3.13.3
lsof      13103 jose  mem       REG                8,3   2030544    9437371 /lib
/x86_64-linux-gnu/libc-2.27.so
lsof      13103 jose  mem       REG                8,3    154832    9442404 /lib
/x86_64-linux-gnu/libselinux.so.1
lsof      13103 jose  mem       REG                8,3    170960    9437367 /lib
/x86_64-linux-gnu/ld-2.27.so
lsof      13103 jose    4r     FIFO               0,13       0t0     136019 pipe
lsof      13103 jose    7w     FIFO               0,13       0t0     136020 pipe
jose@jose-CX61-2PC:~$
```

La salida resultante es bastante amplia es por ello que solo he capturado el final.

Ejercicio 2.

Instalar y ejecutar la citada herramienta en vuestro sistema de cara a:

Para instalarlo he introducido en consola:

```
jose@jose-CX61-2PC:~$ sudo apt-get install lynis
[sudo] contraseña para jose:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Una vez instalado compruebo que se ha instalado correctamente para comprobarlo, introduzco **man lynis** el resultado obtenido es:

```
Lynis(8)              Unix System Administrator's Manual              Lynis(8)

NAME
       Lynis - System and security auditing tool

SYNOPSIS
       lynis [scan mode] [other options]

DESCRIPTION
       Lynis is a security auditing tool for Linux, Mac OSX, and UNIX systems.
       It checks the system and the software configuration, to see if there is
       any  room for improvement the security defenses. All details are stored
       in a log file. Findings and other discovered data is stored in a report
       file. This can be used to compare differences between audits. Lynis can
       run interactively or as a cronjob. Root permissions (e.g. sudo) are not
       required, however provide more details during the audit.
```

Las ordenes más importantes que serán útiles en este ejercicio las he sacado del guión, son las siguientes:

- **audit system:** Realiza una auditoría del sistema.
- **show commands:** Muestra las órdenes disponibles.
- **show help:** Suministra una pantalla de ayuda.
- **show profiles:** Muestra los perfiles descubiertos.
- **show settings:** Lista los ajustes activos de los perfiles.
- **show version:** Muestra la versión actual de Lynis.

a) Mostrar que vulnerabilidades hay en vuestro sistema, asignarle un grado de severidad (en una escala: alta, media o baja) e indicar qué pasos debemos dar para eliminarlas.

Para comprobar que vulnerabilidades existen en mi ordenador voy a usar el comando **audit system** ya explicado anteriormente:

```
jose@jose-CX61-2PC:~$ sudo lynis audit system
```

El resultado obtenido es:

```
================================================================================

  Lynis security scan details:

  Hardening index : 52 [##########          ]
  Tests performed : 231
  Plugins enabled : 1

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

  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 2.6.2

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

  2007-2018, CISOfy - https://cisofy.com/lynis/
  Enterprise support available (compliance, plugins, interface and tools)

================================================================================

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

jose@jose-CX61-2PC:~$
```

Como se puede apreciar en la captura todo el reporte se ha almacenado en **/var/log/lynis-report.dat,** para ver su contenido:

```
jose@jose-CX61-2PC:~$ sudo cat /var/log/lynis-report.dat
# Lynis Report
report_version_major=1
report_version_minor=0
report_datetime_start=2020-10-15 22:45:11
auditor=[Not Specified]
lynis_version=2.6.2
os=Linux
os_name=Ubuntu Linux
os_fullname=Ubuntu 18.04
os_version=18.04
linux_version=Ubuntu
os_kernel_version=5.4.0
os_kernel_version_full=5.4.0-48-generic
hostname=jose-CX61-2PC
test_category=all
test_group=all
```

Es un archivo bastante grande, he capturado solo el inicio de la salida resultante. Pero revisando la salida de forma más detallada encontramos los peligros y las sugerencias que nos aconseja **lynis,** por ejemplo:

```
warning[]=PKGS-7392|Found one or more vulnerable packages.|-|-|
suggestion[]=PKGS-7392|Update your system with apt-get update, apt-get upgrade, apt-get dist-upgrade and/or unattended-upgrades|-|-|
suggestion[]=PKGS-7394|Install package apt-show-versions for patch management purposes|-|-|
```

En la imagen anterior **lynis** nos indica que ha encontrado uno varios paquetes vulnerables, para solucionarlo nos sugiere que realicemos **apt-get update, apt-get upgrade, apt-get dist-upgrade, etc.**

```
warning[]=NETW-2705|Couldn't find 2 responsive nameservers|-|-|
suggestion[]=NETW-2705|Check your resolv.conf file and fill in a backup nameserver if possible|-|-|
```

En la imagen anterior se muestra otro warning esta vez relacionado con los **nameservers,** indica que no ha encontrado 2 responsive nameservers y como sugerencia propone revisar el archivo **resolv.conf** y rellenarlo en un nameserver backup.

```
suggestion[]=HTTP-6640|Install Apache mod_evasive to guard webserver against DoS/brute force attempts|-|-|
suggestion[]=HTTP-6643|Install Apache modsecurity to guard webserver against web application attacks|-|-|
suggestion[]=SSH-7408|Consider hardening SSH configuration|AllowTcpForwarding (YES --> NO)|-|
details[]=SSH-7408|sshd|desc:sshd option AllowTcpForwarding;field:AllowTcpForwarding;prefval:NO;value:YES;|
suggestion[]=SSH-7408|Consider hardening SSH configuration|ClientAliveCountMax (3 --> 2)|-|
details[]=SSH-7408|sshd|desc:sshd option ClientAliveCountMax;field:ClientAliveCountMax;prefval:2;value:3;|
suggestion[]=SSH-7408|Consider hardening SSH configuration|Compression (YES --> (DELAYED|NO))|-|
details[]=SSH-7408|sshd|desc:sshd option Compression;field:Compression;prefval:(DELAYED|NO);value:YES;|
suggestion[]=SSH-7408|Consider hardening SSH configuration|LogLevel (INFO --> VERBOSE)|-|
details[]=SSH-7408|sshd|desc:sshd option LogLevel;field:LogLevel;prefval:VERBOSE;value:INFO;|
suggestion[]=SSH-7408|Consider hardening SSH configuration|MaxAuthTries (6 --> 2)|-|
details[]=SSH-7408|sshd|desc:sshd option MaxAuthTries;field:MaxAuthTries;prefval:2;value:6;|
suggestion[]=SSH-7408|Consider hardening SSH configuration|MaxSessions (10 --> 2)|-|
details[]=SSH-7408|sshd|desc:sshd option MaxSessions;field:MaxSessions;prefval:2;value:10;|
suggestion[]=SSH-7408|Consider hardening SSH configuration|PermitRootLogin (WITHOUT-PASSWORD --> NO)|-|
details[]=SSH-7408|sshd|desc:sshd option PermitRootLogin;field:PermitRootLogin;prefval:NO;value:WITHOUT-PASSWORD;|
suggestion[]=SSH-7408|Consider hardening SSH configuration|Port (22 --> )|-|
details[]=SSH-7408|sshd|desc:sshd option Port;field:Port;prefval:;value:22;|
suggestion[]=SSH-7408|Consider hardening SSH configuration|TCPKeepAlive (YES --> NO)|-|
details[]=SSH-7408|sshd|desc:sshd option TCPKeepAlive;field:TCPKeepAlive;prefval:NO;value:YES;|
suggestion[]=SSH-7408|Consider hardening SSH configuration|X11Forwarding (YES --> NO)|-|
details[]=SSH-7408|sshd|desc:sshd option X11Forwarding;field:X11Forwarding;prefval:NO;value:YES;|
suggestion[]=SSH-7408|Consider hardening SSH configuration|AllowAgentForwarding (YES --> NO)|-|
details[]=SSH-7408|sshd|desc:sshd option AllowAgentForwarding;field:AllowAgentForwarding;prefval:NO;value:YES;|
```

Además se muestran muchas sugerencias relacionadas con el servicio **ssh.**

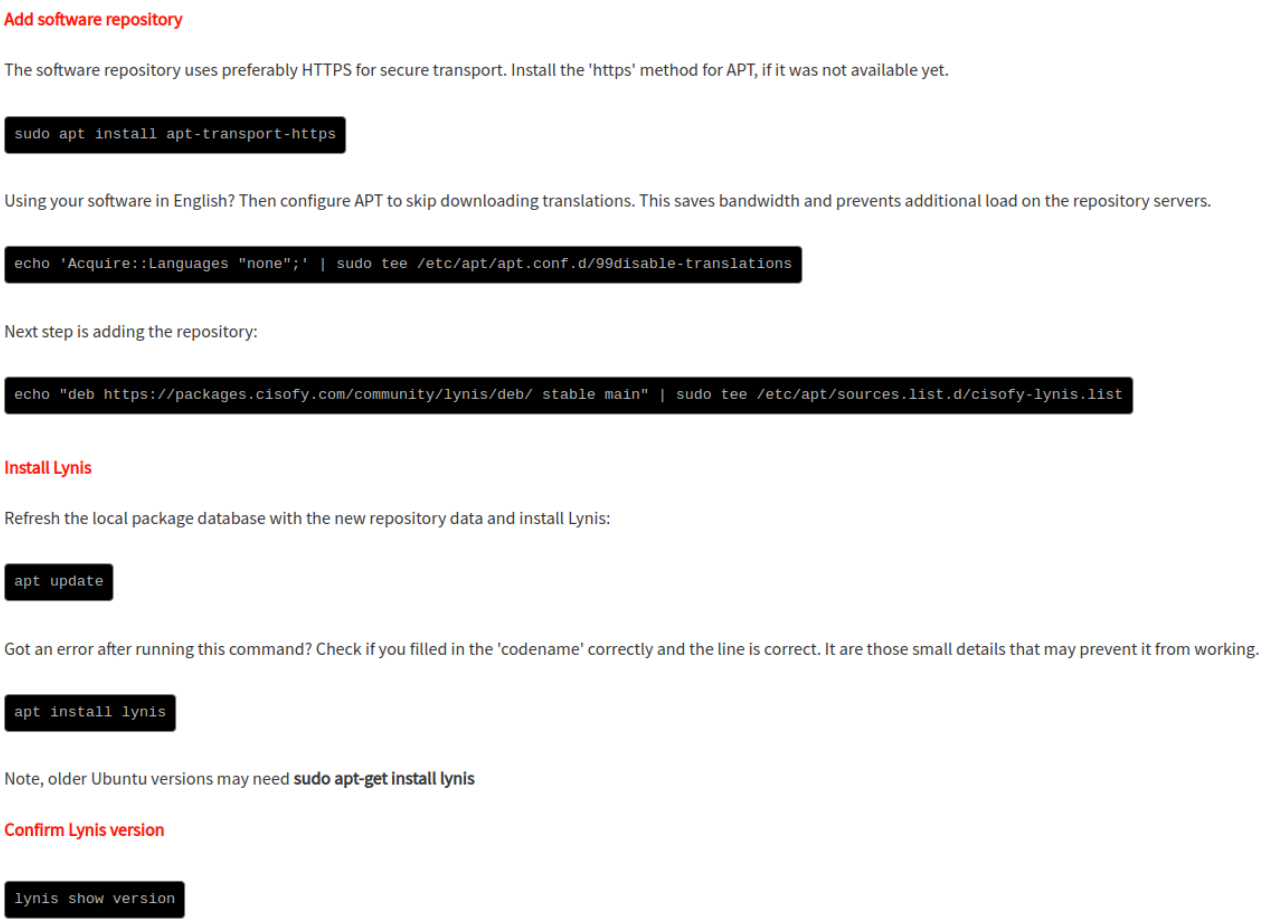
Por último, voy a mostrar un peligro con **mongoDB** aunque hay más warnings y suggestions:

```
warning[]=DBS-1820|MongoDB instance allows any user to access databases|-|-|
suggestion[]=PHP-2320|Harden PHP by disabling risky functions|-|-|
suggestion[]=PHP-2379|Harden PHP by enabling suhosin extension|-|-|
suggestion[]=PHP-2379|Harden PHP by deactivating suhosin simulation mode|-|-|
```

En este caso **lynis** nos indica que la instancia de MongoDB permite acceder a cualquier usuario a las bases de datos. Como sugerencias para solucionar el problema nos indica 3 soluciones para solventar este warning.

b) En clase de teoría, vimos la vulnerabilidad Shellshock (CVE-2014-6271), indicar si la herramienta citada comprueba dicha vulnerabilidad y explicar cómo lo hace (esto nos servirá para conocer como podríamos desarrollar nuestro propio test). Consejo, revisar el contenido del archivo de la herramienta *include/tests_shells.*

Durante la realización de este ejercicio he comprobado que había instalado una versión antigua y procedí actulizarlo pues no me mostraba ningún tipo de información en la script asociada al error **shellshock,** para actualizarlo he seguido los pasos siguientes:



Add software repository

The software repository uses preferably HTTPS for secure transport. Install the 'https' method for APT, if it was not available yet.

```
sudo apt install apt-transport-https
```

Using your software in English? Then configure APT to skip downloading translations. This saves bandwidth and prevents additional load on the repository servers.

```
echo 'Acquire::Languages "none";' | sudo tee /etc/apt/apt.conf.d/99disable-translations
```

Next step is adding the repository:

```
echo "deb https://packages.cisofy.com/community/lynis/deb/ stable main" | sudo tee /etc/apt/sources.list.d/cisofy-lynis.list
```

Install Lynis

Refresh the local package database with the new repository data and install Lynis:

```
apt update
```

Got an error after running this command? Check if you filled in the 'codename' correctly and the line is correct. It are those small details that may prevent it from working.

```
apt install lynis
```

Note, older Ubuntu versions may need **sudo apt-get install lynis**

Confirm Lynis version

```
lynis show version
```

Ahora sí ya tenía **lynis** actualizado:

```
jose@jose-CX61-2PC:~$ lynis show version
3.0.1
```

Una vez actualizado vuelvo a lanzar el **audit** de **lynis:**

```
jose@jose-CX61-2PC:~$ lynis audit system

[ Lynis 3.0.1 ]

################################################################################
  Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
  welcome to redistribute it under the terms of the GNU General Public License.
  See the LICENSE file for details about using this software.

  2007-2020, CISOfy - https://cisofy.com/lynis/
  Enterprise support available (compliance, plugins, interface and tools)
################################################################################
```

Ahora compruebo el contenido del archivo **/usr/share/lynis/include/tests_shells:**

```
jose@jose-CX61-2PC:~$ sudo cat /usr/share/lynis/include/tests_shells
```

El contenido del script es:

```
#!/bin/sh

#################################################################################
#
#   Lynis
# ------------------
#
# Copyright 2007-2013, Michael Boelen
# Copyright 2007-2020, CISOfy
#
# Website  : https://cisofy.com
# Blog     : http://linux-audit.com
# GitHub   : https://github.com/CISOfy/lynis
#
# Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
# welcome to redistribute it under the terms of the GNU General Public License.
# See LICENSE file for usage of this software.
#
#################################################################################
#
# Shells
#
#################################################################################
#
   IDLE_TIMEOUT=0
   InsertSection "Shells"
#
#################################################################################
#
   # bash
   # Files (interactive login shells):     /etc/profile $HOME/.bash_profile
   #                                       $HOME/.bash_login $HOME/.profile
   # Files (interactive non-login shells): $HOME/.bash_rc
   #
   # csh/tcsh
   # Files: /etc/csh.cshrc /etc/csh.login
   #
   # zsh
   # Files: /etc/zshenv /etc/zsh/zshenv $HOME/.zshenv /etc/zprofile
   #        /etc/zsh/zprofile $HOME/.zprofile /etc/zshrc /etc/zsh/zshrc
   #        $ZDOTDIR/.zshrc /etc/zlogin /etc/zsh/zlogin

   SHELL_LOGIN_FILES="${ROOTDIR}etc/csh.cshrc ${ROOTDIR}etc/csh.login ${ROOTDIR}etc/zshenv $
{ROOTDIR}etc/zsh/zshenv
                      ${ROOTDIR}etc/zprofile ${ROOTDIR}etc/zsh/zprofile ${ROOTDIR}etc/zshrc
${ROOTDIR}etc/zsh/zshrc
                      ${ROOTDIR}etc/zlogin ${ROOTDIR}etc/zsh/zlogin"
#
#################################################################################
#
# Test        : SHLL-6202
   # Description : check all console TTYs in which root user can enter single user mode without password
   Register --test-no SHLL-6202 --os FreeBSD --weight L --network NO --category security --description "Check
console TTYs"
   if [ ${SKIPTEST} -eq 0 ]; then
      LogText "Test: Checking console TTYs"
FIND=$(${EGREPBINARY} '^console' ${ROOTDIR}etc/ttys | ${GREPBINARY} -v 'insecure')
      if [ -z "${FIND}" ]; then
         Display --indent 2 --text "- Checking console TTYs" --result "${STATUS_OK}" --color GREEN
         LogText "Result: console is secured against single user mode without password."
      else
```

```
Display --indent 2 --text "- Checking console TTYs" --result "${STATUS_WARNING}" --color RED
        LogText "Result: Found insecure console in ${ROOTDIR}etc/ttys. Single user mode login without password allowed!"
        LogText "Output ${ROOTDIR}etc/ttys:"
        LogText "${FIND}"
        ReportWarning "${TEST_NO}" "Found unprotected console in ${ROOTDIR}etc/ttys"
        LogText "Possible solution: Change the console line from 'secure' to 'insecure'."
      fi
   fi
#
################################################################################
#
   # Test       : SHLL-6211
   # Description : Determine available shell according /etc/shells
   Register --test-no SHLL-6211 --weight L --network NO --category security --description "Available and valid shells"
   if [ ${SKIPTEST} -eq 0 ]; then
      LogText "Test: Searching for ${ROOTDIR}etc/shells"
      if [ -f ${ROOTDIR}etc/shells ]; then
         LogText "Result: Found ${ROOTDIR}etc/shells file"
         LogText "Test: Reading available shells from ${ROOTDIR}etc/shells"
         SSHELLS=$(${GREPBINARY} "^/" ${ROOTDIR}etc/shells)
         CSSHELLS=0; CSSHELLS_ALL=0
         Display --indent 2 --text "- Checking shells from ${ROOTDIR}etc/shells"
         for I in ${SSHELLS}; do
            CSSHELLS_ALL=$((CSSHELLS_ALL + 1))
            Report "available_shell[]=${I}"
            # TODO add check for symlinked shells
            if [ -f ${I} ]; then
               LogText "Found installed shell: ${I}"
               CSSHELLS=$((CSSHELLS + 1))
            else
               LogText "Shell ${I} not installed. Probably a dummy or non existing shell."
            fi
         done
         Display --indent 4 --text "Result: found ${CSSHELLS_ALL} shells (valid shells: ${CSSHELLS})."
      else
         LogText "Result: ${ROOTDIR}etc/shells not found, skipping test"
      fi
   fi
#
################################################################################
#
 # Test       : SHLL-6220
   # Description : Check for idle session killing tools or settings
   Register --test-no SHLL-6220 --weight L --network NO --category security --description "Idle session killing tools or settings"
   if [ ${SKIPTEST} -eq 0 ]; then

      IDLE_TIMEOUT_METHOD=""
      IDLE_TIMEOUT_READONLY=""

      LogText "Test: Search for session timeout tools or settings in shell"
      if IsRunning "timeoutd"; then
         IDLE_TIMEOUT=1
         LogText "Result: found timeoutd process to kill idle sessions"
         IDLE_TIMEOUT_METHOD="timeout-daemon"
      fi
      if IsRunning "autolog"; then
         IDLE_TIMEOUT=1
         LogText "Result: found autolog process to kill idle sessions"
         Report "session_timeout_method[]=autolog"
         IDLE_TIMEOUT_METHOD="autolog"
      fi
```

```
if [ -f ${ROOTDIR}etc/profile ]; then
        # Determine if we can find a TMOUT value
        FIND=$(${GREPBINARY} 'TMOUT=' ${ROOTDIR}etc/profile | ${TRBINARY} -d ' ' | ${TRBINARY} -d '\t' | ${GREPBINARY} -v "^#" | ${SEDBINARY} 's/export//' | ${SEDBINARY} 's/#.*//' | ${AWKBINARY} -F= '{ print $2 }')
        # Determine if the value is exported (with export, readonly, or typeset)
        FIND2=$(${GREPBINARY} '\(export\|readonly\|typeset -r\)[ \t]*TMOUT' ${ROOTDIR}etc/profile | ${GREPBINARY} -v "^#" | ${SEDBINARY} 's/#.*//' | ${AWKBINARY} '{ print $1 }')
        if [ -n "${FIND}" ]; then
           N=0; IDLE_TIMEOUT=1
           for I in ${FIND}; do
              LogText "Output: ${I}"
              Report "session_timeout_value[]=${I}"
              N=$((N + 1))
           done
           if [ ${N} -eq 1 ]; then
              LogText "Result: found TMOUT value configured in ${ROOTDIR}etc/profile"
           else
              LogText "Result: found several TMOUT values configured in ${ROOTDIR}etc/profile"
           fi
           IDLE_TIMEOUT_METHOD="profile"
        else
           LogText "Result: could not find TMOUT setting in ${ROOTDIR}etc/profile"
        fi

        if [ -n "${FIND2}" ]; then
           N=0;
           for I in ${FIND2}; do
              LogText "Output: ${I}"
              if [ "${I}" = "readonly" -o "${I}" = "typeset" ]; then
                 N=$((N + 1))
              fi
           done
           if [ ${N} -gt 0 ]; then
              LogText "Result: found readonly setting in ${ROOTDIR}etc/profile (readonly or typeset -r)"
              IDLE_TIMEOUT_READONLY=1
           else
              LogText "Result: NO readonly setting found in ${ROOTDIR}etc/profile (readonly or typeset -r)"
              IDLE_TIMEOUT_READONLY=0
           fi
        else
           LogText "Result: could not find export, readonly or typeset -r in ${ROOTDIR}etc/profile"
        fi
     else
        LogText "Result: skip ${ROOTDIR}etc/profile test, file not available on this system"
     fi

     if [ -d ${ROOTDIR}etc/profile.d ]; then
        FIND=$(${LSBINARY} ${ROOTDIR}etc/profile.d/*.sh 2> /dev/null)
        if [ -n "${FIND}" ]; then
           # Determine if we can find a TMOUT value
           FIND=$(${FINDBINARY} ${ROOTDIR}etc/profile.d -name "*.sh" -type f -exec cat {} \; 2> /dev/null | ${GREPBINARY} 'TMOUT=' | ${TRBINARY} -d ' ' | ${TRBINARY} -d '\t' | ${GREPBINARY} -v "^#" | ${SEDBINARY} 's/export//' | ${SEDBINARY} 's/#.*//' | ${AWKBINARY} -F= '{ print $2 }')
           # Determine if the value is exported (with export, readonly, or typeset)
           FIND2=$(${FINDBINARY} ${ROOTDIR}etc/profile.d -name "*.sh" -type f -exec cat {} \; 2> /dev/null | ${GREPBINARY} '\(export\|readonly\|typeset -r\)[ \t]*TMOUT' | ${GREPBINARY} -v "^#" | ${SEDBINARY} 's/#.*//' | ${AWKBINARY} '{ print $1 }')
           if [ -n "${FIND}" ]; then
              N=0; IDLE_TIMEOUT=1
              for I in ${FIND}; do
                 LogText "Output: ${I}"
                 Report "session_timeout_value[]=${I}"
                 N=$((N + 1))
```

```
done
                if [ ${N} -eq 1 ]; then
                   LogText "Result: found TMOUT value configured in one of the files in ${ROOTDIR}etc/profile.d
directory"
                else
                   LogText "Result: found several TMOUT values configured in one of the files in
${ROOTDIR}etc/profile.d directory"
                fi
                IDLE_TIMEOUT_METHOD="profile.d"
             else
                LogText "Result: could not find TMOUT setting in ${ROOTDIR}etc/profile.d/*.sh"
             fi
             # Check for readonly
             if [ -n "${FIND2}" ]; then
                N=0;
                for I in ${FIND2}; do
                   LogText "Output: ${I}"
                   if [ "${I}" = "readonly" -o "${I}" = "typeset" ]; then
                      N=$((N + 1))
                   fi
                done
                if [ ${N} -gt 0 ]; then
                   LogText "Result: found readonly setting in ${ROOTDIR}etc/profile (readonly or typeset -r)"
                   IDLE_TIMEOUT_READONLY=1
                else
                   LogText "Result: NO readonly setting found in ${ROOTDIR}etc/profile (readonly or typeset -r)"
                   IDLE_TIMEOUT_READONLY=0
                fi
             else
                LogText "Result: could not find export, readonly or typeset -r in ${ROOTDIR}etc/profile"
             fi
          fi
       else
          LogText "Result: skip ${ROOTDIR}etc/profile.d directory test, directory not available on this system"
       fi

       if [ -n "${IDLE_TIMEOUT_METHOD}" ]; then
          Report "session_timeout_method[]=${IDLE_TIMEOUT_METHOD}"
       fi
       if [ -n "${IDLE_TIMEOUT_READONLY}" ]; then
          Report "session_timeout_set_readonly=${IDLE_TIMEOUT_READONLY}"
       fi

       if [ ${IDLE_TIMEOUT} -eq 1 ]; then
          Display --indent 4 --text "- Session timeout settings/tools" --result "${STATUS_FOUND}" --color GREEN
          AddHP 3 3
       else
          Display --indent 4 --text "- Session timeout settings/tools" --result "${STATUS_NONE}" --color YELLOW
          AddHP 1 3
       fi
    fi
#
##############################################################################
#
# Test       : SHLL-6230
    # Description : Check for umask values in shell configurations
    SHELL_CONFIG_FILES="${ROOTDIR}etc/bashrc ${ROOTDIR}etc/bash.bashrc
${ROOTDIR}etc/bash.bashrc.local ${ROOTDIR}etc/csh.cshrc ${ROOTDIR}etc/profile"
    Register --test-no SHLL-6230 --weight H --network NO --category security --description "Perform umask check
for shell configurations"
    if [ ${SKIPTEST} -eq 0 ]; then
```

```
FOUND=0
      Display --indent 2 --text "- Checking default umask values"
      for FILE in ${SHELL_CONFIG_FILES}; do
         HARDENING_POSSIBLE=0
         FIND=""
         if [ -f ${FILE} ]; then
            LogText "Result: file ${FILE} exists"
            FOUND=1
            FIND=$(${GREPBINARY} umask ${FILE} | ${SEDBINARY} 's/^[ \t]*//g' | ${SEDBINARY} 's/#.*$//' |
${GREPBINARY} -v "^$" | ${AWKBINARY} '{ print $2 }')
            if IsEmpty "${FIND}"; then
               LogText "Result: did not find umask configured in ${FILE}"
               Display --indent 4 --text "- Checking default umask in ${FILE}" --result "${STATUS_NONE}" --color
YELLOW
            else
               for UMASKVALUE in ${FIND}; do
                  LogText "Result: found umask ${UMASKVALUE} in ${FILE}"
                  case ${UMASKVALUE} in
                     027|0027|077|0077)
                           LogText "Result: umask ${UMASKVALUE} is considered a properly hardened value"
                     ;;
                     *)
                           LogText "Result: umask ${UMASKVALUE} can be hardened "
                           HARDENING_POSSIBLE=1
                     ;;
                  esac
               done
               if [ ${HARDENING_POSSIBLE} -eq 0 ]; then
                  Display --indent 4 --text "- Checking default umask in ${FILE}" --result "${STATUS_OK}" --color
GREEN
                  AddHP 3 3
               else
                  Display --indent 4 --text "- Checking default umask in ${FILE}" --result "${STATUS_WEAK}" --
color YELLOW
                  AddHP 1 3
               fi
            fi
         else
            LogText "Result: file ${FILE} not found"
         fi
      done
   fi
#
################################################################################
#

Report "session_timeout_enabled=${IDLE_TIMEOUT}"


WaitForKeyPress

#
#================================================================================
# Lynis - Copyright 2007-2020, CISOfy - http://cisofy.com
```

He adjuntado todo el script que contiene dicho archivo pero en ningún momento se realiza una comprobación del problema de **shellshock,** creía que el problema era la versión de **lynis,** y procedí a actualizarlo a la última versión, la versión 3.0.1 pero aún así no aparece dicho test, considero que se debe a que mi versión de ubuntu es la 18.0 y este error ya está solucionado y no es necesario ningún test acerca de ello. He de mencionar que en la

auditoría tampoco se hace ninguna comprobación acerca de este tipo de vulnerabilidad luego yo creo que este tipo de vulnerabilidad está ya resuelto en versiones 18.0 o superiores.

c) Suponiendo que nuestro sistema tiene un antivirus, Avx, no contemplado por la herramienta. Indicar qué debemos hacer para que la herramienta lo detecte y que no muestre en el informe final que no tenemos solución antivirus).

Lo primero que realizo es comprobar que tests disponibles tiene **lynis,** para ello:

```
jose@jose-CX61-2PC:~$ sudo ls /usr/share/lynis/include/ | grep test
tests_accounting
tests_authentication
tests_banners
tests_boot_services
tests_containers
tests_crypto
tests_custom.template
tests_databases
tests_dns
tests_file_integrity
tests_file_permissions
tests_filesystems
tests_firewalls
tests_hardening
tests_homedirs
tests_insecure_services
tests_kernel
tests_kernel_hardening
tests_ldap
tests_logging
tests_mac_frameworks
tests_mail_messaging
tests_malware
tests_memory_processes
tests_nameservices
tests_networking
tests_php
tests_ports_packages
tests_printers_spoolers
tests_scheduling
tests_shells
tests_snmp
tests_squid
tests_ssh
tests_storage
tests_storage_nfs
tests_system_integrity
tests_time
tests_tooling
tests_usb
tests_virtualization
tests_webservers
Mostrar aplicaciones
jose@jose-CX61-2PC:~$
```

En la imagen anterior se pueden ver todos los test de **lynis,** pero si analizamos cada uno de ellos de forma detallada se puede ver que hay un llamado **tests_malware,** voy a comprobar su contenido:

```
jose@jose-CX61-2PC:~$ sudo cat /usr/share/lynis/include/tests_malware
[sudo] contraseña para jose:
#!/bin/sh

#################################################################################
#
#   Lynis
# -----------------
#
# Copyright 2007-2013, Michael Boelen
# Copyright 2007-2020, CISOfy
#
# Website  : https://cisofy.com
# Blog     : http://linux-audit.com
# GitHub   : https://github.com/CISOfy/lynis
#
# Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
# welcome to redistribute it under the terms of the GNU General Public License.
# See LICENSE file for usage of this software.
#
#################################################################################
#
# Malware scanners
```

No voy a copiar todo su contenido ya que el script es bastante grande pero para confirmar que efectivamente este test se encarga de comprobar si nuestro sistema dispone de antivirus se puede ver en la siguiente imagen:

```
#
#################################################################################
#
    # Test        : MALW-3280
    # Description : Check if an anti-virus tool is installed
    Register --test-no MALW-3280 --weight L --network NO --category security --description "Check if anti-virus tool is installed"
    if [ ${SKIPTEST} -eq 0 ]; then
        FOUND=0

        # Avast (macOS)
        LogText "Test: checking process com.avast.daemon"
        if IsRunning "com.avast.daemon"; then
            FOUND=1
            AVAST_DAEMON_RUNNING=1
            MALWARE_SCANNER_INSTALLED=1
            if IsVerbose; then Display --indent 2 --text "- ${GEN_CHECKING} Avast daemon" --result "${STATUS_FOUND}" --color GREEN; fi
            LogText "Result: found Avast security product"
            Report "malware_scanner[]=avast"
        fi

        # Avira
        LogText "Test: checking process Avira daemon"
        if IsRunning "avqmd"; then
            FOUND=1
            AVIRA_DAEMON_RUNNING=1
            MALWARE_SCANNER_INSTALLED=1
            if IsVerbose; then Display --indent 2 --text "- ${GEN_CHECKING} Avira daemon" --result "${STATUS_FOUND}" --color GREEN; fi
            LogText "Result: found Avira security product"
            Report "malware_scanner[]=avira"
        fi
```

Efectivamente se comprueba si nuestro sistema dispone de alguna herramienta de antivirus, luego si queremos que el antivirus **Avx** sea detectado por **lynis** debemos añadir lineas de comprobación de **Avx** en el test **MALW-3280**, concretamente en el script que se muestra en la imagen. Este script actualmente comprueba si los siguientes antivirus están instalados:

- **Avast (macOS)**
- **Avira**
- **Bitdefender (macOS)**
- **CrowdStrike falcon-sensor**
- **Cylance (macOS)**
- **ESE security producs**

- **Karpersky products**
- **McAfee products**
- **Sophos savscand/SophosScanD**
- **Symantec rtvscand/smcd/symcfgd**
- **TrendMicro (macOS)**

Luego deberíamos añadir a esta lista **Avx** para que sea detectado por **lynis.**

Ejercicio 3.

Instalar y ejecutar la citada herramienta en vuestro sistema de cara a:

Para instalarlo he introducido en consola:

```
jose@jose-CX61-2PC:~$ sudo apt install rkhunter
```

Una vez instalado compruebo que está en su última versión:

```
jose@jose-CX61-2PC:~$ sudo rkhunter --version
Rootkit Hunter 1.4.6

This software was developed by the Rootkit Hunter project team.
Please review your rkhunter configuration files before using.
Please review the documentation before posting bug reports or questions.
To report bugs, provide patches or comments, please go to:
http://rkhunter.sourceforge.net

To ask questions about rkhunter, please use the rkhunter-users mailing list.
Note this is a moderated list: please subscribe before posting.

Rootkit Hunter comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it under the
terms of the GNU General Public License. See the LICENSE file for details.
```

a) Relizar un análisis del sistema para ver si esta o no comprometido.

Para realizar un análisis en el sistema, introduzco por consola:

```
jose@jose-CX61-2PC:~$ sudo rkhunter --check --skip-keypress --rwo
Warning: The command '/usr/bin/lwp-request' has been replaced by a script: /usr/
bin/lwp-request: Perl script text executable
Warning: The SSH configuration option 'PermitRootLogin' has not been set.
         The default value may be 'yes', to allow root access.
Warning: Suspicious file types found in /dev:
         /dev/shm/sem.CiscoAcNamedEventPostureISE: data
         /dev/shm/sem.CiscoAcNamedEventOpenDNS: data
         /dev/shm/sem.CiscoAcNamedEventNVM: data
         /dev/shm/sem.CiscoAcMemoryLock: data
         /dev/shm/tmp: data
Warning: Hidden directory found: /etc/.java
jose@jose-CX61-2PC:~$
```

Como se puede ver en la imagen anterior se han reportado 4 warnings, el primer warning indica que un comando ha sido reemplazado por una script, el segundo warning indica que PermitRootLogin de SSH no ha sido asignado, el tercer warning indica que ha encontrado

archivos sospechosos en /dev/shm/sem... Por último, hay una advetencia sobre un archivo oculto de java.

b) De los avisos, soluciona los que sean falsos positivos, bien eliminando los test bien bien ajustándolos adecuadamente.

Voy a solucionar el último archivo warning, lo primero que realizo es comprobar que efectivamente se trata de un falso positivo, para ello:

```
jose@jose-CX61-2PC:~$ ls -lai /etc/.java/
total 20
3940361 drwxr-xr-x   3 root root  4096 nov 27  2019 .
3932161 drwxr-xr-x 142 root root 12288 oct 16 20:01 ..
3940362 drwxr-xr-x   2 root root  4096 nov 27  2019 .systemPrefs
jose@jose-CX61-2PC:~$ cat /etc/.java/.systemPrefs/
cat: /etc/.java/.systemPrefs/: Es un directorio
jose@jose-CX61-2PC:~$ ls -lai /etc/.java/.systemPrefs/
total 8
3940362 drwxr-xr-x 2 root root 4096 nov 27  2019 .
3940361 drwxr-xr-x 3 root root 4096 nov 27  2019 ..
3934843 -rw-r--r-- 1 root root    0 nov 27  2019 .system.lock
3934844 -rw-r--r-- 1 root root    0 nov 27  2019 .systemRootModFile
jose@jose-CX61-2PC:~$ ls -lai /etc/.java/.systemPrefs/.system.lock
3934843 -rw-r--r-- 1 root root 0 nov 27  2019 /etc/.java/.systemPrefs/.system.lo
ck
jose@jose-CX61-2PC:~$ cat /etc/.java/.systemPrefs/.systemRootModFile
jose@jose-CX61-2PC:~$ cat /etc/.java/.systemPrefs/.system.lock
jose@jose-CX61-2PC:~$
```

He comprobado que efectivamente es un falso positivo, para evitar que se vuelva a producir este falso positivo de nuevo voy a realizar lo siguiente:

- Editar el archivo **/etc/rkhunter.conf:**

```
jose@jose-CX61-2PC:~$ sudo gedit /etc/rkhunter.conf
[sudo] contraseña para jose:
```

Dentro de este archivo realizo los siguientes cambios:

```
#ALLOWHIDDENDIR=/etc/.java
#ALLOWHIDDENDIR=/etc/.git
#ALLOWHIDDENDIR=/dev/.lxc
ALLOWHIDDENDIR=/etc/.java
```

- Realizar de nuevo el check para asegurarnos que el falso positivo no vuelve a salir como warning:

```
jose@jose-CX61-2PC:~$ sudo rkhunter --check --skip-keypress --rwo
Warning: The command '/usr/bin/lwp-request' has been replaced by a script: /usr/
bin/lwp-request: Perl script text executable
Warning: The SSH configuration option 'PermitRootLogin' has not been set.
         The default value may be 'yes', to allow root access.
Warning: Suspicious file types found in /dev:
         /dev/shm/sem.CiscoAcNamedEventPostureISE: data
         /dev/shm/sem.CiscoAcNamedEventOpenDNS: data
         /dev/shm/sem.CiscoAcNamedEventNVM: data
         /dev/shm/sem.CiscoAcMemoryLock: data
         /dev/shm/tmp: data
jose@jose-CX61-2PC:~$
```

Efectivamente el falso positivo no ha vuelto a salir.