SEGURIDAD EN SISTEMAS OPERATIVOS

4º Grado en Informática – Complementos de Ing. del Software
Curso 2018-19

Práctica 1. Administración de la seguridad en Linux

Sesión 2. Herramientas básicas de seguridad

Autor[1]: Víctor García Carrera

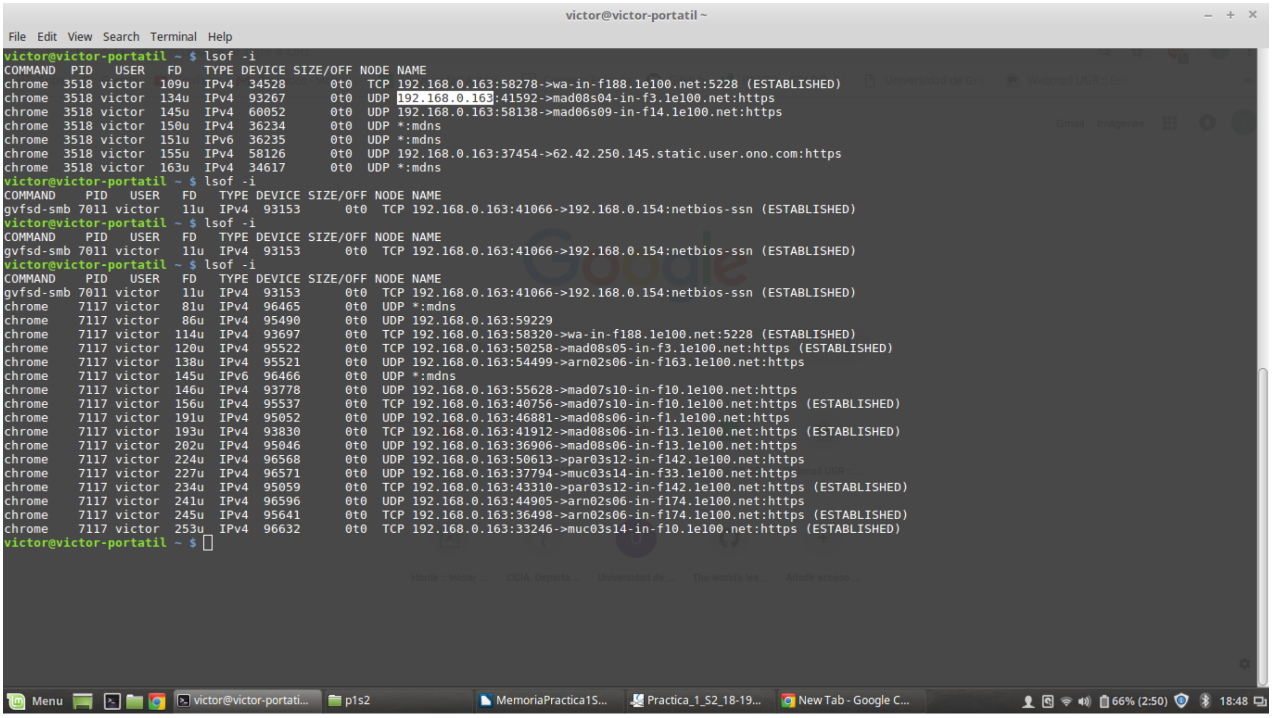
Ejercicio 1.

En este apartado trabajamos con el comando *lsof*, que nos permite listar los ficheros abiertos en nuestro sistema. Como establece la propia entrada en el manual de este comando (*man lsof*), los ficheros o archivos abiertos pueden ir desde un fichero regular a una librería o hasta un fichero de red (sockets), por lo que nos permite incluso ver si hay procesos o servicios de nuestro sistema que están accediendo a la red, algo especialmente útil para valorar si hay tráfico sospechoso. En concreto, el comando lsof -i nos permite listar todos los archivos de red de Interet y x.25 (HP-UX). Si añadimos como parámetro -i4 or -i6 sin ninguna dirección específica, solo aquellos archivos de la versión IP indicada, IPv4 or IPv6, se muestran.

Al utilizar el comando *lsof* vemos que aparecen multidud de entradas y de ficheros, donde cada entrada consta de la siguiente información:

COMMAND PID TID USER FD TYPE DEVICE SIZE/OFF NODE NAME

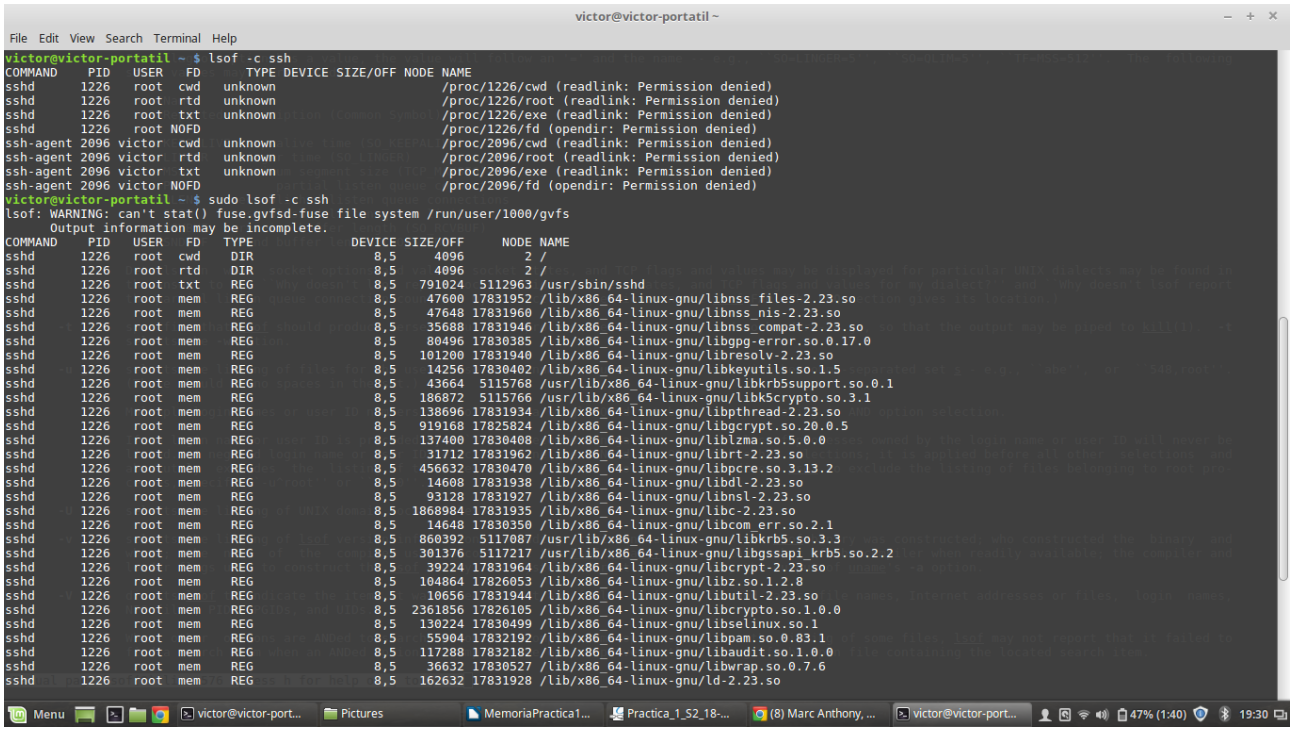
Utilizamos el comando *lsof -i* para ver sólo aquellos archivos relacionados con conexiones abiertas en nuestro sistema, que devuelve la siguiente salida:



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 ..."

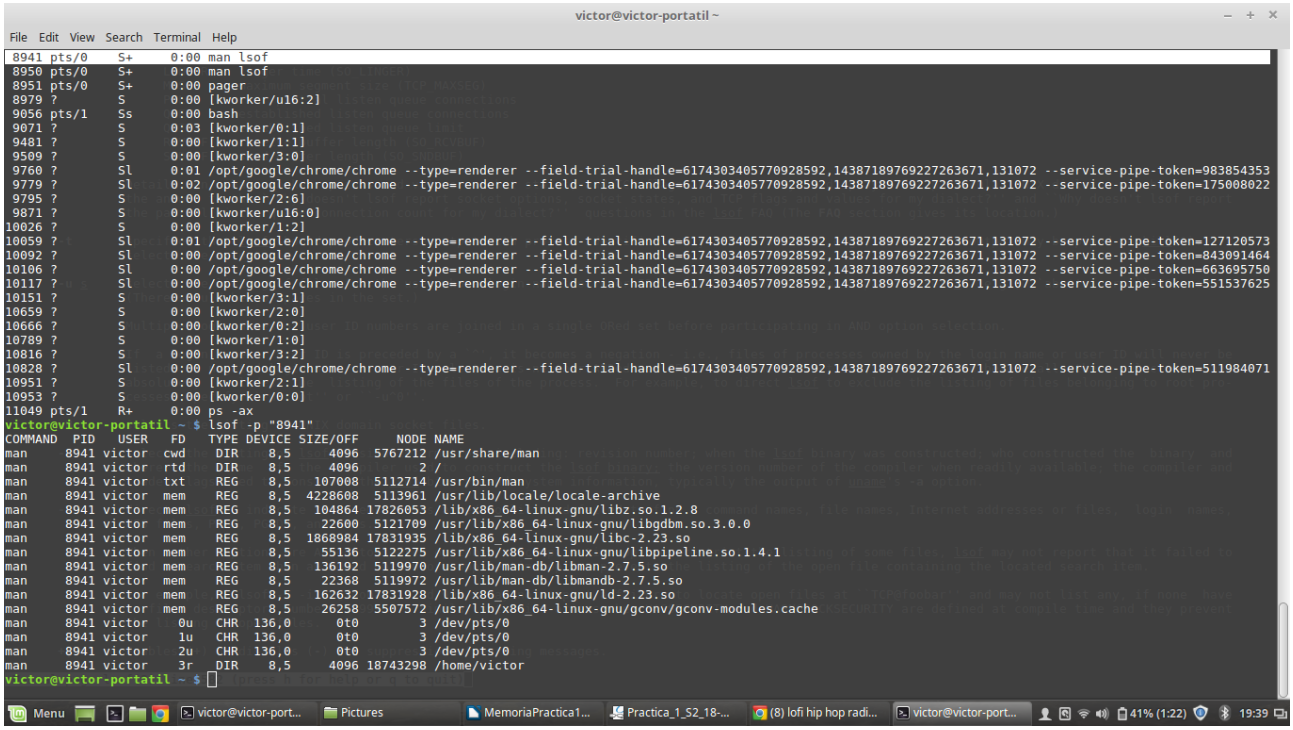
En la primera llamada estaba reproduciendo en google chrome un video de youtube. Vemos como hay un proceso TCP que parece que se utiliza para establecer la conexión con el servidor, y otros procesos que utilizan UDP que seguramente se traten de aquellos destinados al tráfico del video en si. En la segunda y tercera llamada había cerrado el buscador, y sólo aparece una entrada asociada a un proceso que monta SAMBA, utilizado para compartir archivos a través de la red (gvfsd-smb). En las últimas llamadas de nuevo he abierto el buscador y reproducido un video de youtube, apareciendo entradas similares a las primeras. La herramienta, como he descrito previamente, proporciona información como el PID del proceso asociado (podemos ver los PID de los procesos del sistema con el comando *ps*), el usuario del mismo, la versión IP...

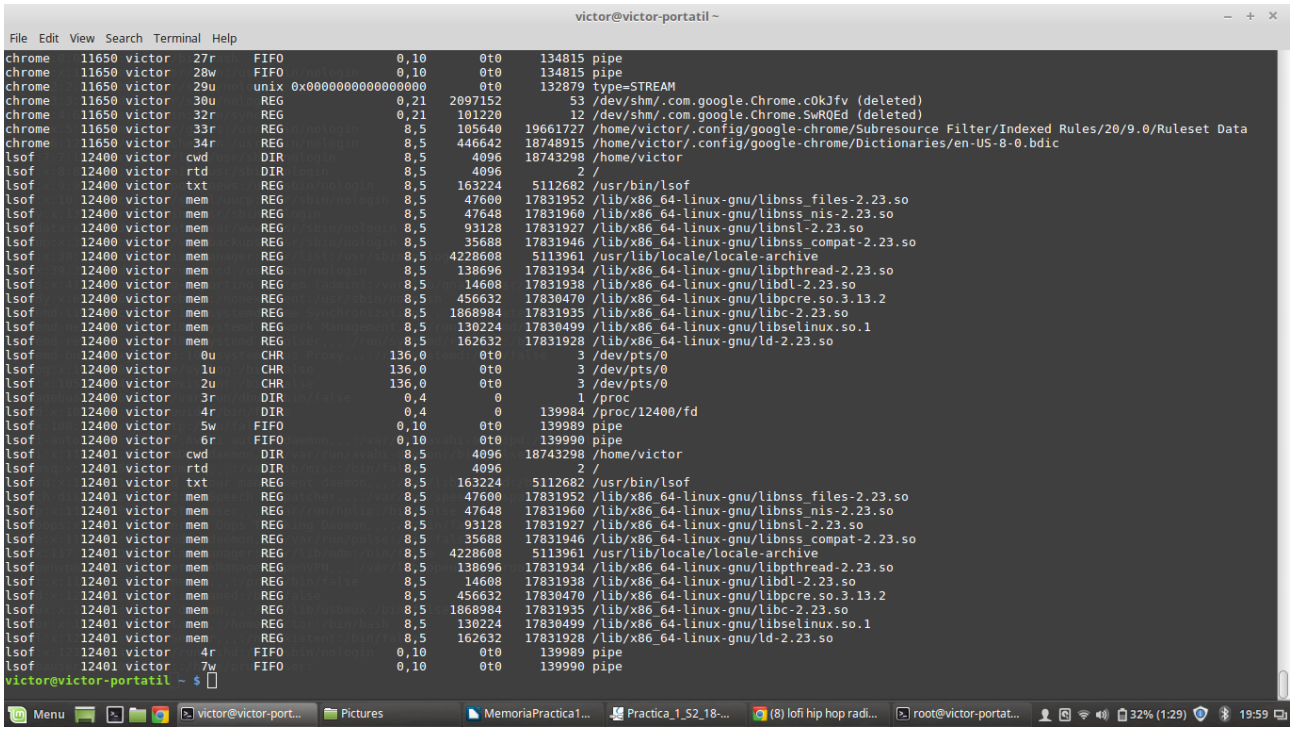
Cabe destacar que estos comandos proporcionan una información más completa cuando se ejecutan en modo superusuario (sudo). Si queremos conocer por ejemplo si hay tráfico ssh, podemos utilizar el comando *lsof -c ssh* para así listar los archivos abiertos por procesos activos que utilizan el comando ssh. De esta manera, podemos ver no sólo la ruta de estos archivos, sino además el usuario del proceso que está ejecutando ese comando (generando el tráfico ssh).



Finalmente, si queremos conocer los archivos a los que está accediendo un proceso con PID *x* en concreto, utilizamos el comando *lsof -p "x"*. Si queremos conocer los archivos en uso por un usuario con nombre *y*, utilizamos el comando *lsof -u "y"*.

A continuación se encuentran 2 imagenes donde se ejemplifica esto, mostrando los archivos del proceso con PID 8941 y los archivos del usuario "victor" (que son bastantes al ser el usuario principal del sistema):

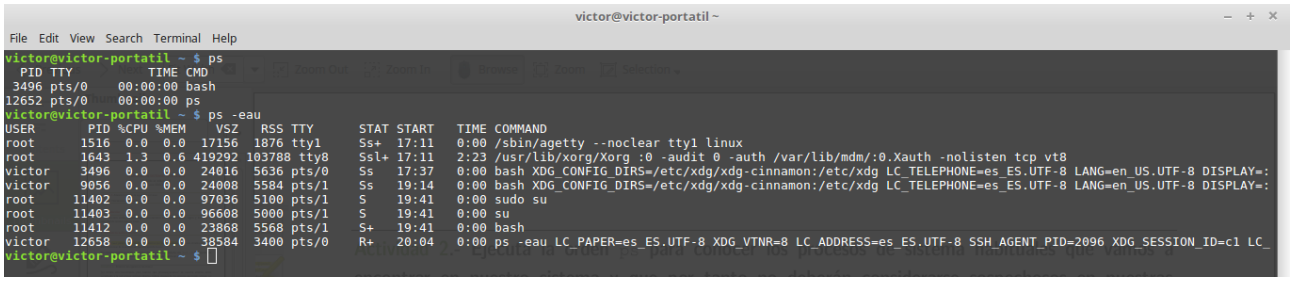


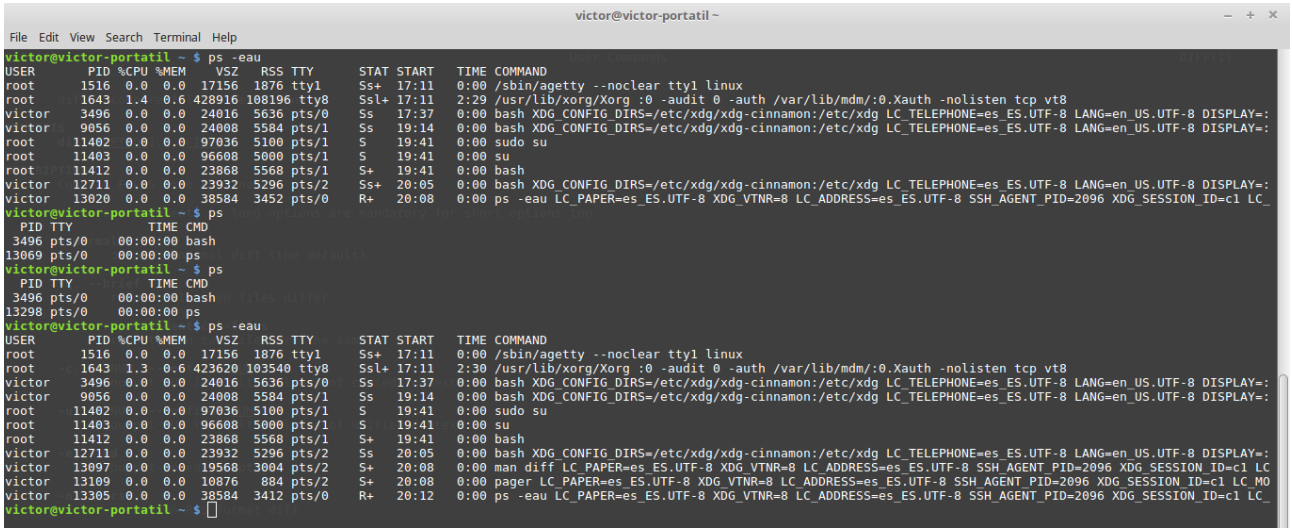


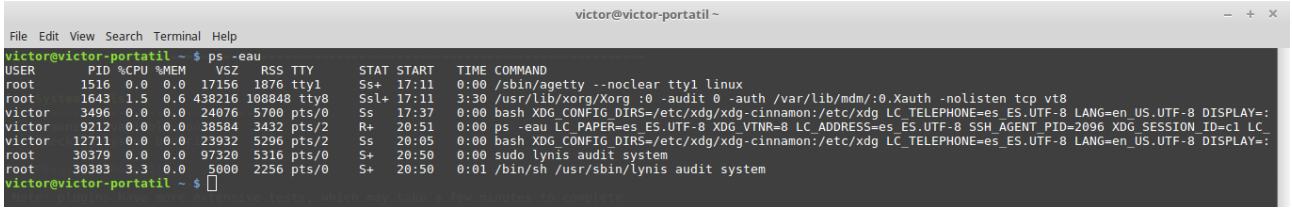
Ejercicio 2.

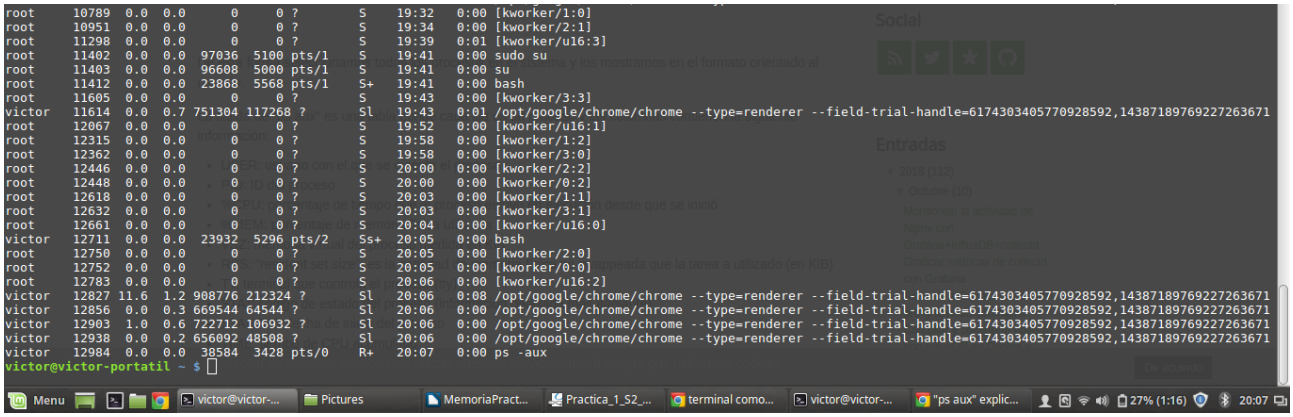
En el siguiente apartado trabajamos con el comando *ps*, que permite listar los procesos activos en el sistema. Tomamos capturas de la salida de este comando en diversos momentos, con diversas opciones para visualizar unos procesos u otros (*ps -aux* para ver todos los procesos activos, *ps -eau* para ver el momento de inicio de los mismos), y comprobamos con el comando *diff* que las diferencias básicamente radican en los nuevos procesos creados (y la finalización de otros) a raiz de un uso normal del

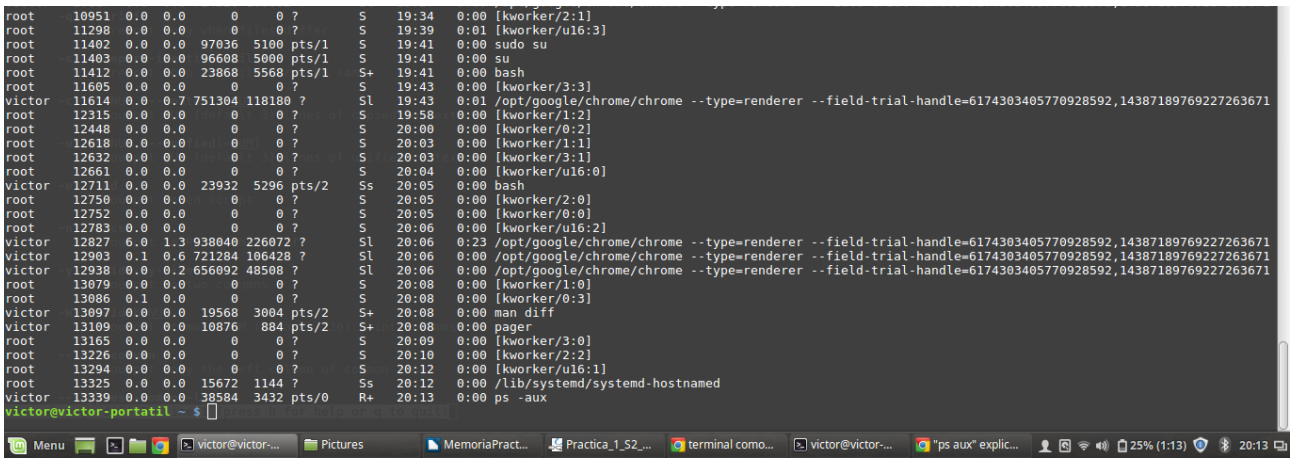
sistema, como puede ser ver una entrada de un comando en el manual (*man diff*), ejecutar en otra terminal la herramienta Lynis para explorar vulnerabilidades del sistema...

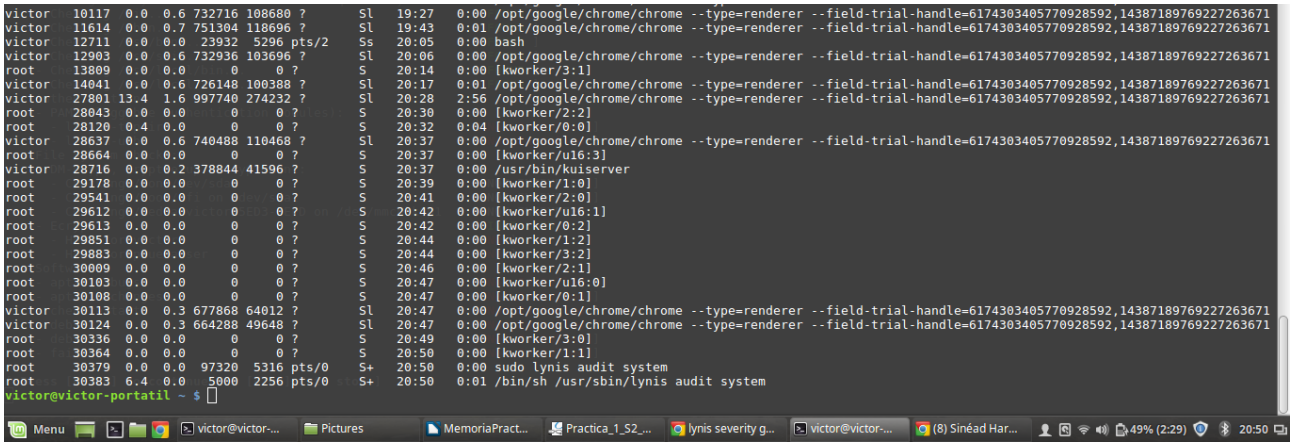






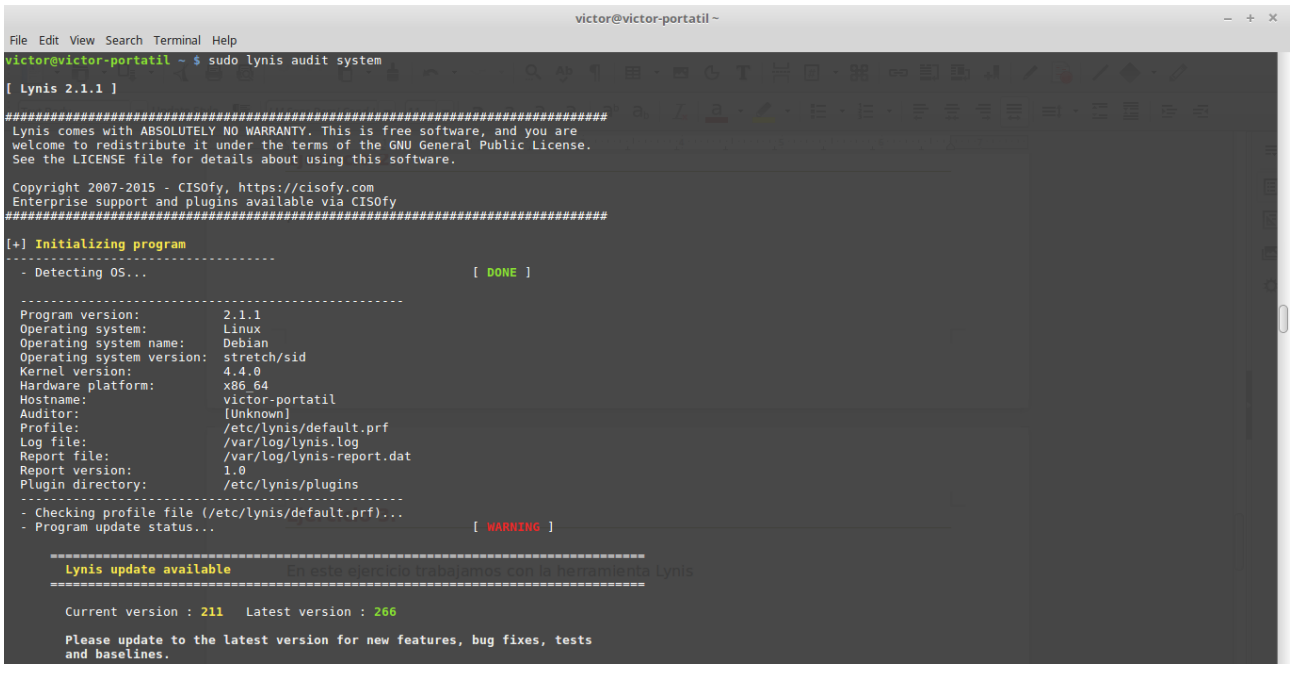




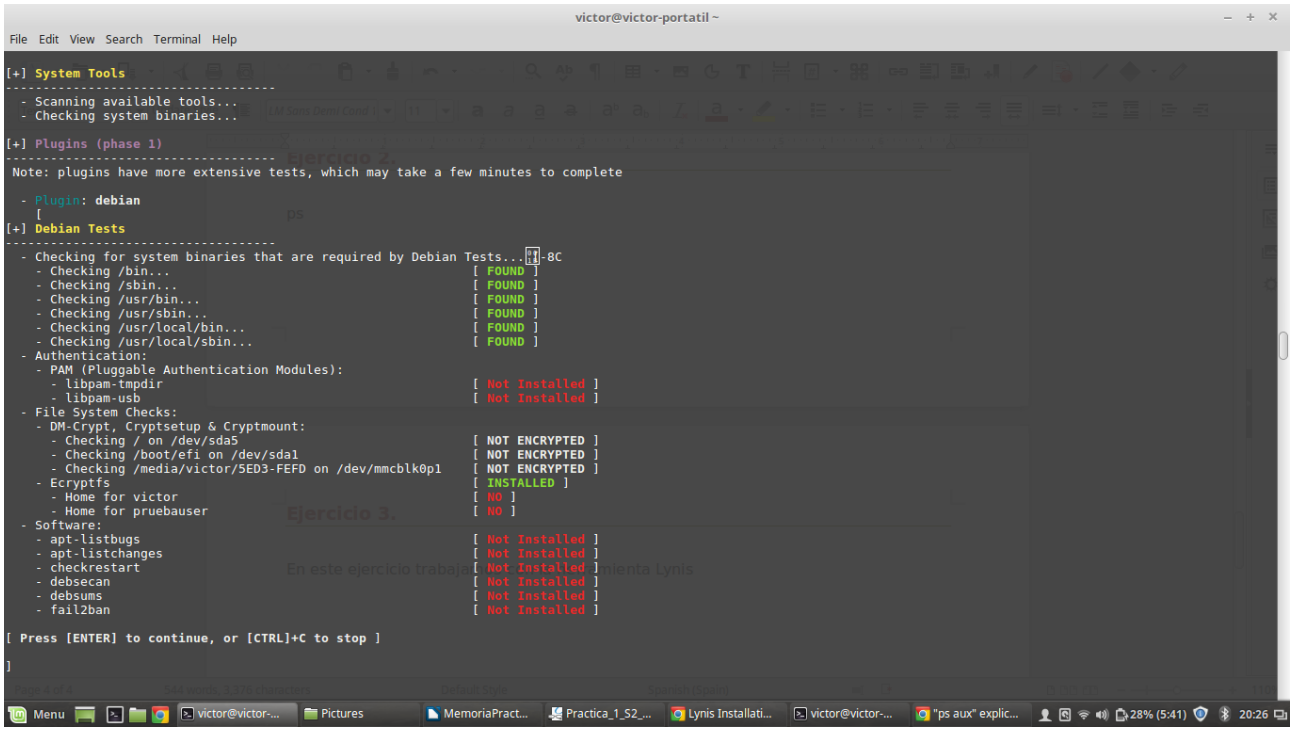


Ejercicio 3.

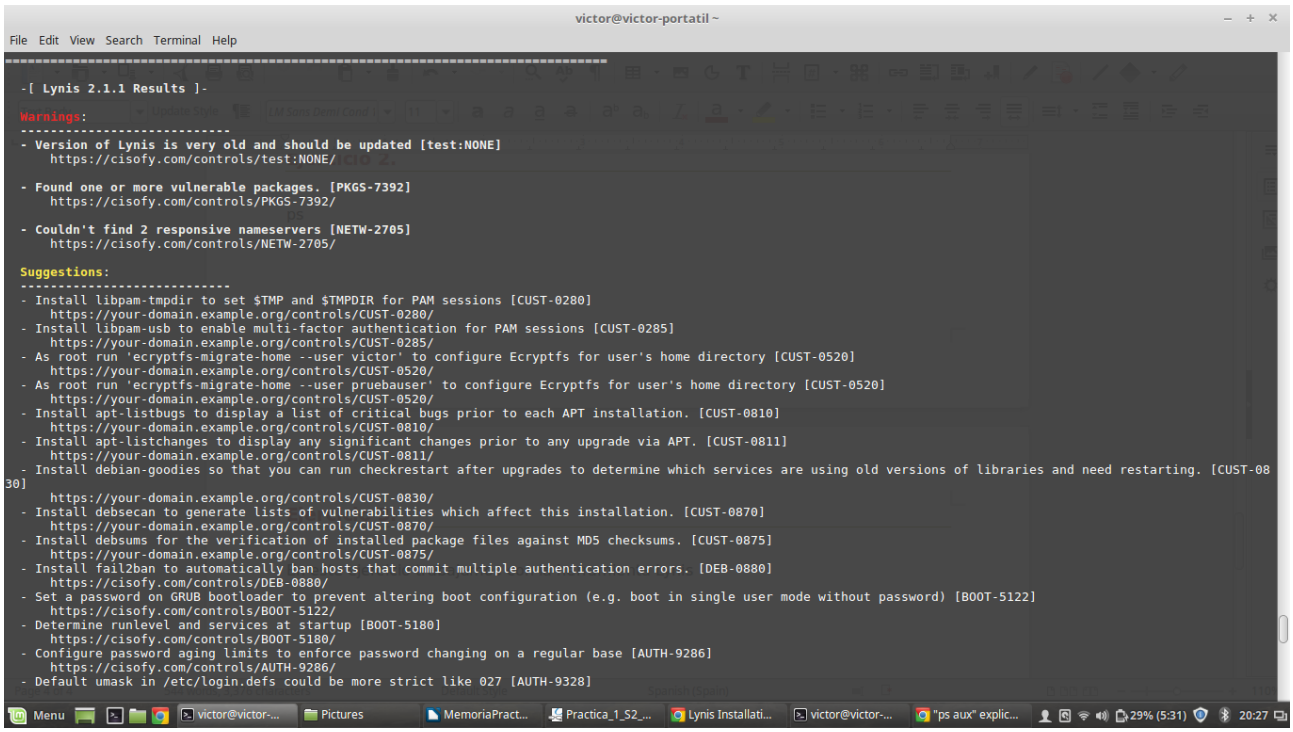
En este ejercicio trabajamos con la herramienta Lynis. Comenzamos utilizando el comando *sudo lynis audit system* para realizar una auditoría del sistema, evaluar los fallos del mismo y obtener recomendaciones para solucionarlos.

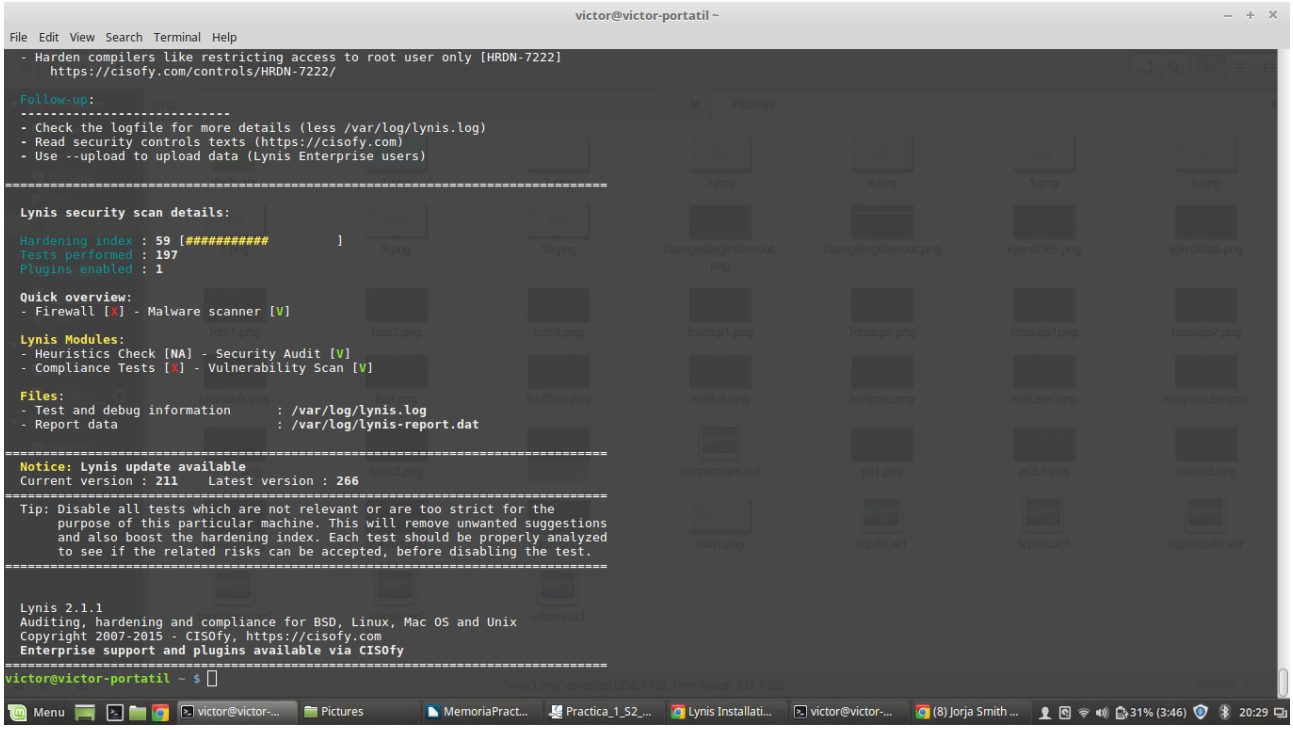


Este es un ejemplo de uno de los tests y comprobaciones que realiza:



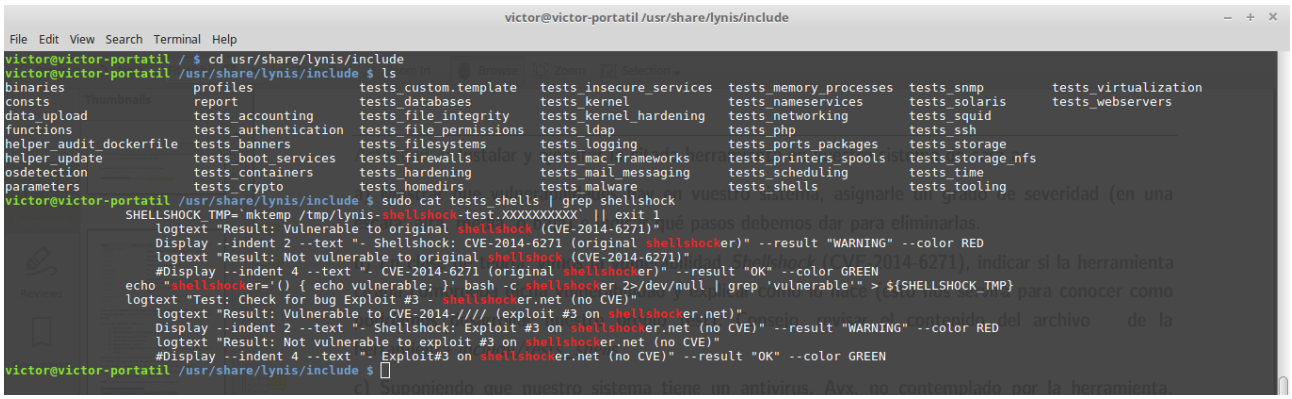
A continuación se muestra el reporte final de la herramienta:





En la primera imagen se muestran algunos warnings y sugerencias acerca de los problemas que ha encontrado y recomendaciones de cómo solventarlos. Estos mismos podemos encontrarlos en el log localizado en la ruta *var/log/lynis.log*. A lo largo de los tests utiliza colores (verde, amarillo, rojo) para indicar el grado de severidad de los mismos. En la última imagen podemos además observar en qué grado de fortalecimiento se encuentra nuestro sistema, un 59% en este caso.

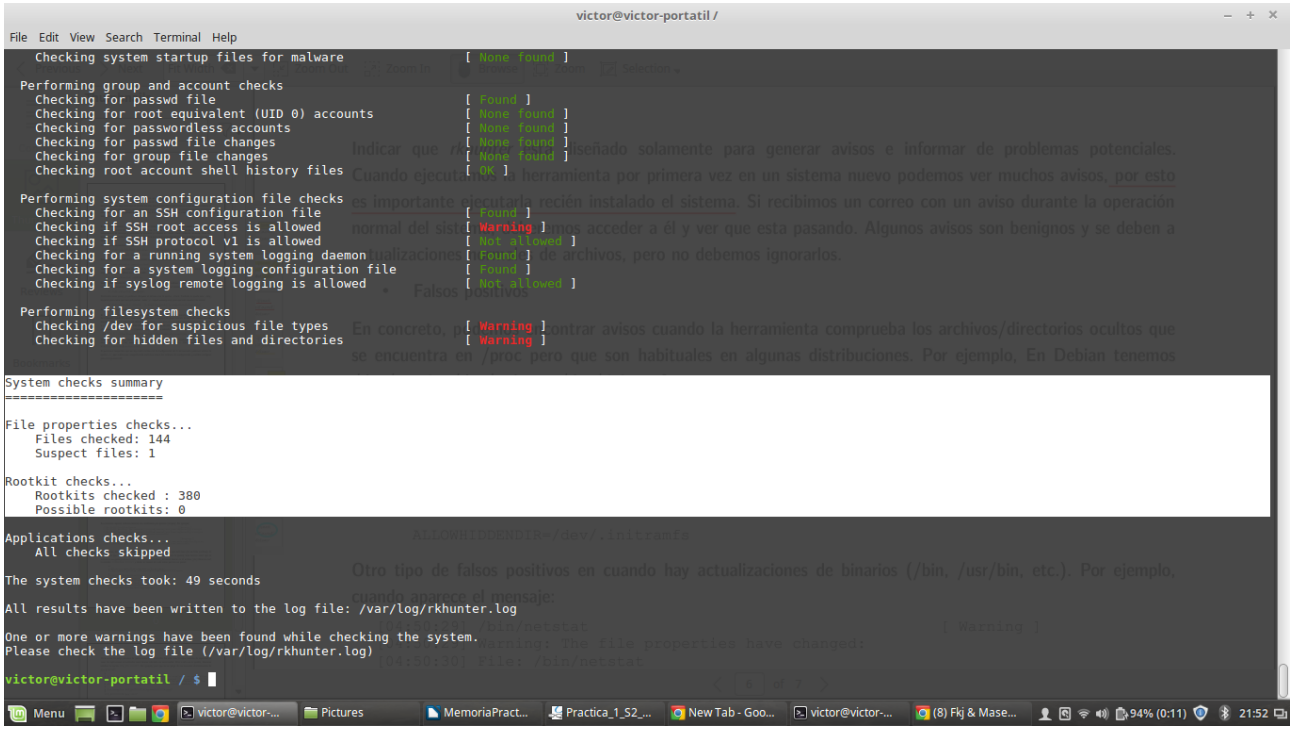
Podemos comprobar si la herramienta detecta una vulnerabilidad en concreto en la ruta *usr/share/lynis/include/tests_shells*. Con ayuda del comando *grep*, buscamos si la herramienta comprueba la vulnerabilidad *Shellshock*. La siguiente salida muestra cómo efectivamente realiza un test para comprobar esta vulnerabilidad.



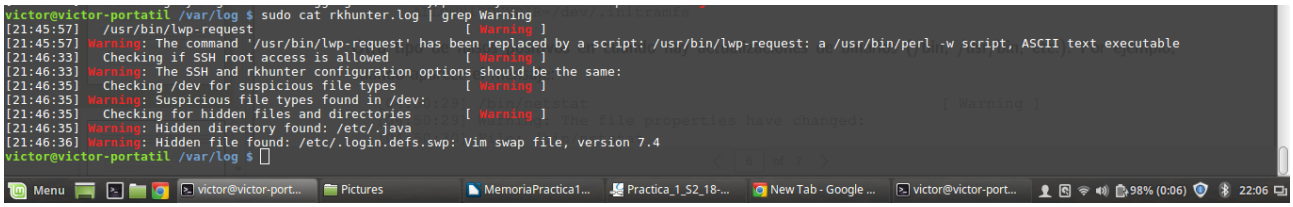
En el archivo *tests_malware* de este mismo directorio podemos ver qué comprobaciones hace acerca del antivirus instalado en el sistema. La herramienta checkea la presencia o no de un antivirus por defecto. Si queremos cambiar esto para que detecte otro diferente, hay que modificar la configuración de uno de los perfiles de Lynis, que por defecto está el archivo *etc/lynis/default.prf*, creando un nuevo archivo de configuración *custom.prf* donde establecemos este cambio en la comprobación del antivirus.

Ejercicio 4.

Por último trabajamos con la herramienta rkhunter para comprobar rootkits y backdoors o puertas traseras. Tras instalarlo y actualizarlo utilizamos el comando *sudo rkhunter --check [--skip-keypress]* con esta última opción para ejecutar el diagnóstico, resultando en la siguiente salida:



Podemos ver de forma más detenida el resultado del análisis en el log localizado en la ruta /var/log/rkhunter.log. En la siguiente imagen de ese log podemos visualizar los avisos indicados:



Estos son falsos positivos que podemos corregir modificando el archivo de configuración *etc/rkhunter.conf*.

Para solucionar el primero de *command ... has been replaced by a script*, primero comprobamos que el programa está bien, y a continuación para que no salte de nuevo el aviso en futuros diagnósticos, descomentamos la linea

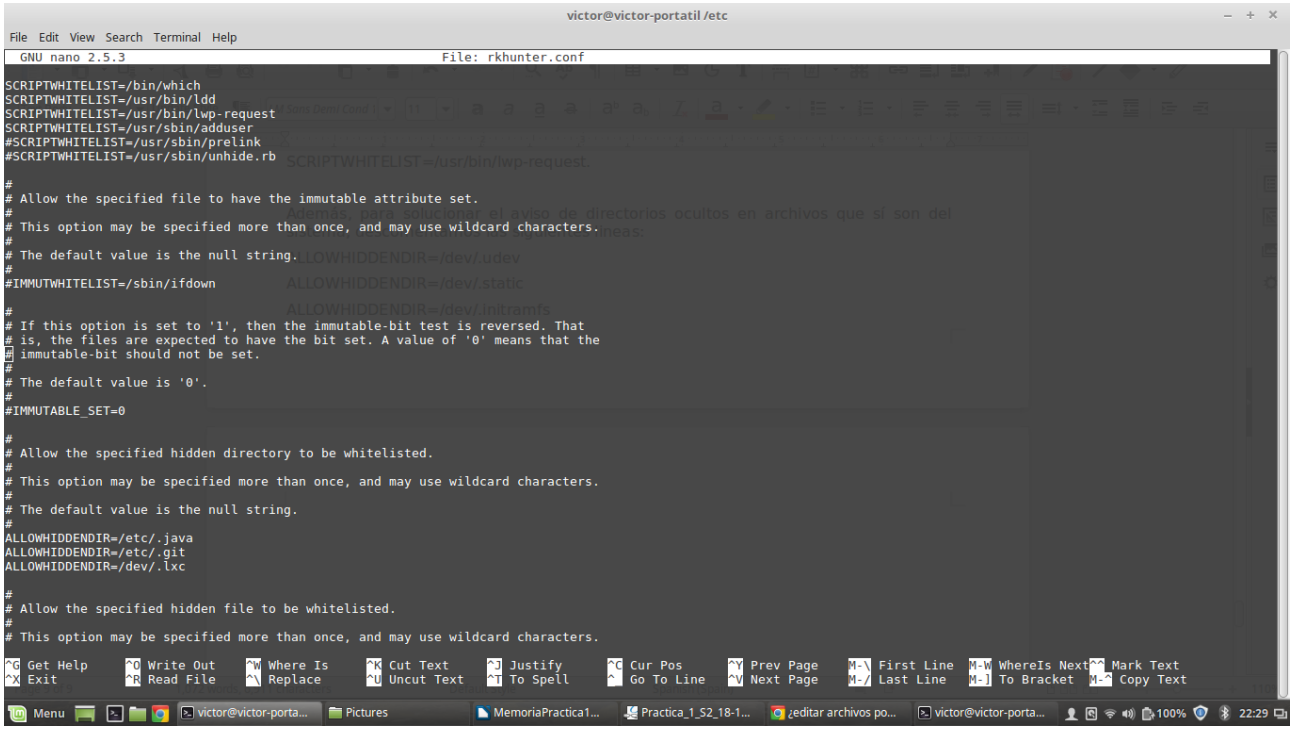
SCRIPTWHITELIST=/usr/bin/lwp-request.

Además, para solucionar el aviso de directorios ocultos en archivos que sí son del sistema, descomentamos las siguientes lineas:

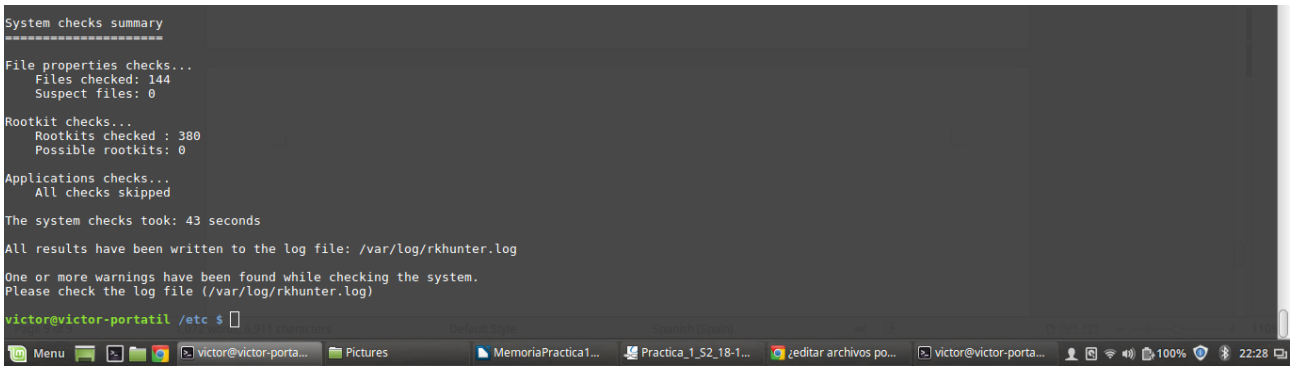
ALLOWHIDDENDIR=/dev/.udev

ALLOWHIDDENDIR=/dev/.static

ALLOWHIDDENDIR=/dev/.initramfs



Finalmente volvemos a ejecutar el diagnóstico de rkhunter y comprobamos que hemos solventado esos falsos positivos:



El único aviso que aparece en el log es el de *The SSH and rkhunter configuration options should be the same.* Hemos eliminado los falsos positivos.