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

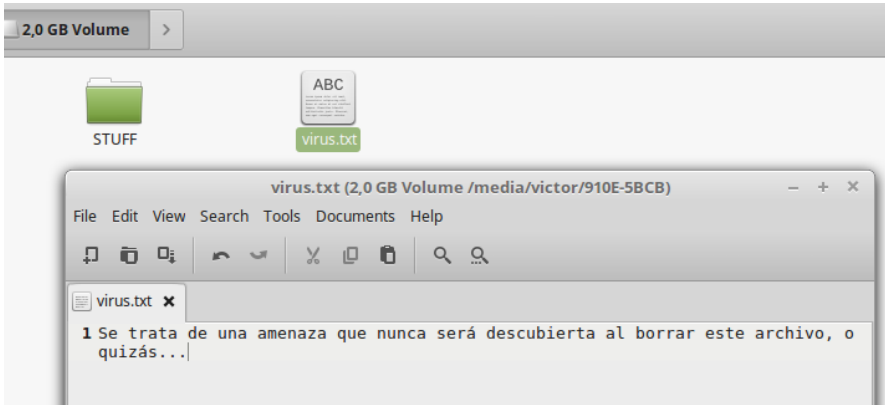
Práctica 3. Auditoría Informática e Informática forense

Sesión 1. Análisis forense en Linux

Autor[1]: Víctor García Carrera

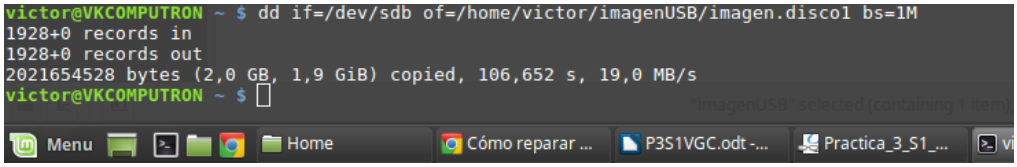
Ejercicio 1.

En este primer ejercicio vamos a simular el caso de un análisis forense sobre un USB en busca de un fichero borrado que simula ser una posible amenaza borrada para evitar que se detectara. Comprobaremos que, mediante diversas técnicas, es posible recuperar el contenido del mismo mediante la búsqueda de texto en espacio disperso o no asignado, pues aunque el estafador "borrara" el archivo, aun puede estar presente en la memoria del USB en bloques de disco no en uso. Creamos en el pendrive el archivo virus.txt que contiene la cadena mostrada a continuación y borramos el archivo.



Primero comenzamos creando una imagen forense del USB. A fin de no modificar ni influir en ningun aspecto sobre el USB original, para realizar en análisis forense creamos una imagen del USB ubicado en */dev/sdb* mediante el comando:

dd if=/dev/sdb of=/home/victor/imagenUSB/imagen.disco1 bs=1M (MEJOR /dev/sdb1)



Por cuestiones de seguridad y prácticas, cambiamos los permisos del mismo para impedir que se ejecute cualquier archivo con el comando *chmod 444 imagen.disco1*

Una vez que disponemos de la imagen forense del pendrive en la ruta */home/victor/imagenUSB/imagen.disco1*, queremos montarla en el directorio creado */mnt/analisis* para poder ver su contenido. A continuación se muestra este proceso y cómo se encuentra, efectivamente, en la ruta */mnt/analisis* la imagen del USB. Cabe

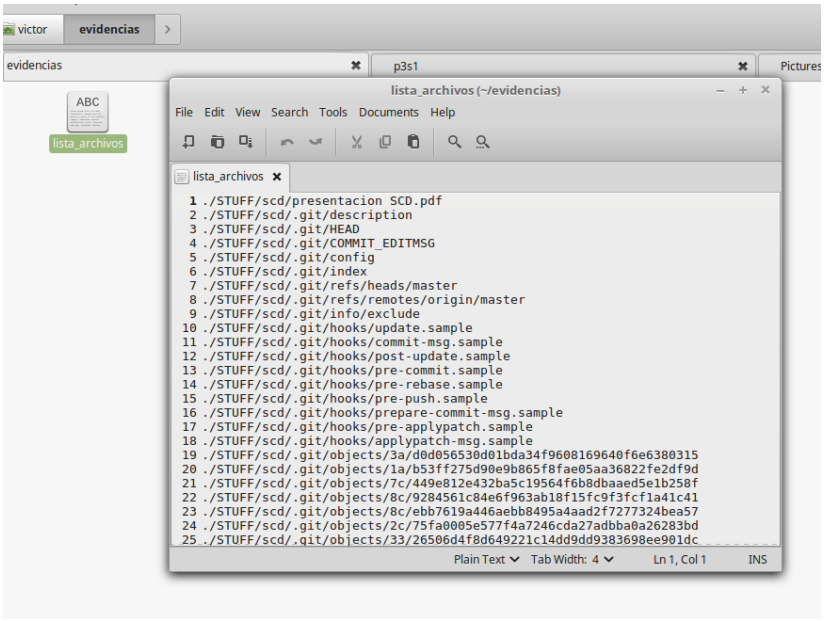
---

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 ..."

destacar que el montado de esta imagen supuso algunos problemas, puesto que hay que tener en cuenta que la imagen de un disco duro comienza con un bootloader y la tabla de particiones, por lo que hay que calcular el offset de la partición o unidad y montarlo con la opción de offset del comando mount. Para averiguar el valor del offset, multiplicamos el tamaño de bloque (512 bytes) por el sector de inicio del disco (2048), información que nos muestra fdisk -l, resultando en 1048576. Este offset se debe a que la copia se realizó de todo el disco duro */dev/sdb*, que cuenta con el bootloader y la tabla de particiones, y debemos saltarnos estas para llegar a la unidad */dev/sdb1*, que es realmente donde se encontraba el contenido del pendrive.

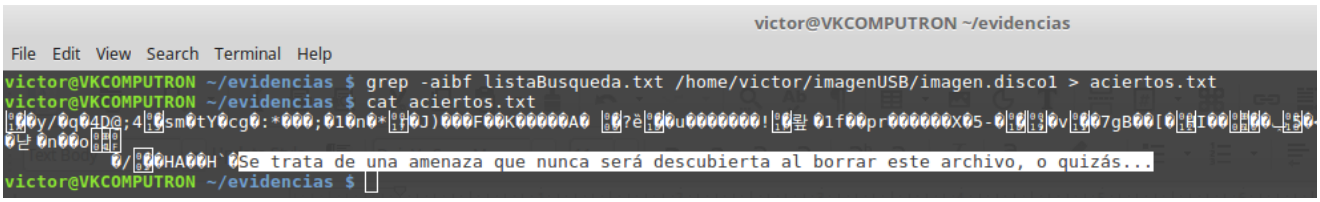
```
victor@VKCOMPUTRON / $ fdisk -l /home/victor/imagenUSB/imagen.disco1
Disk /home/victor/imagenUSB/imagen.disco1: 1,9 GiB, 2021654528 bytes, 3948544 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xd348c306

Device                                Boot Start     End Sectors  Size Id Type
/home/victor/imagenUSB/imagen.disco1p1       2048 3948543 3946496  1,9G  c W95 FAT
victor@VKCOMPUTRON / $ sudo mount -t vfat -o,noexec,loop,offset=1048576 /home/victor/imagenUSB/imagen.disco1 /mnt/analisis
victor@VKCOMPUTRON / $ cd /mnt/analisis/
victor@VKCOMPUTRON /mnt/analisis $ ls
STUFF
victor@VKCOMPUTRON /mnt/analisis $ cd STUFF/
```

Una vez montada la imagen del disco, guardamos en un archivo en la ruta */home/victor/evidencias/lista_archivos* todos los archivos de la imagen mediante el comando: *find . -type f -print > /home/victor/evidencias/lista_archivos*



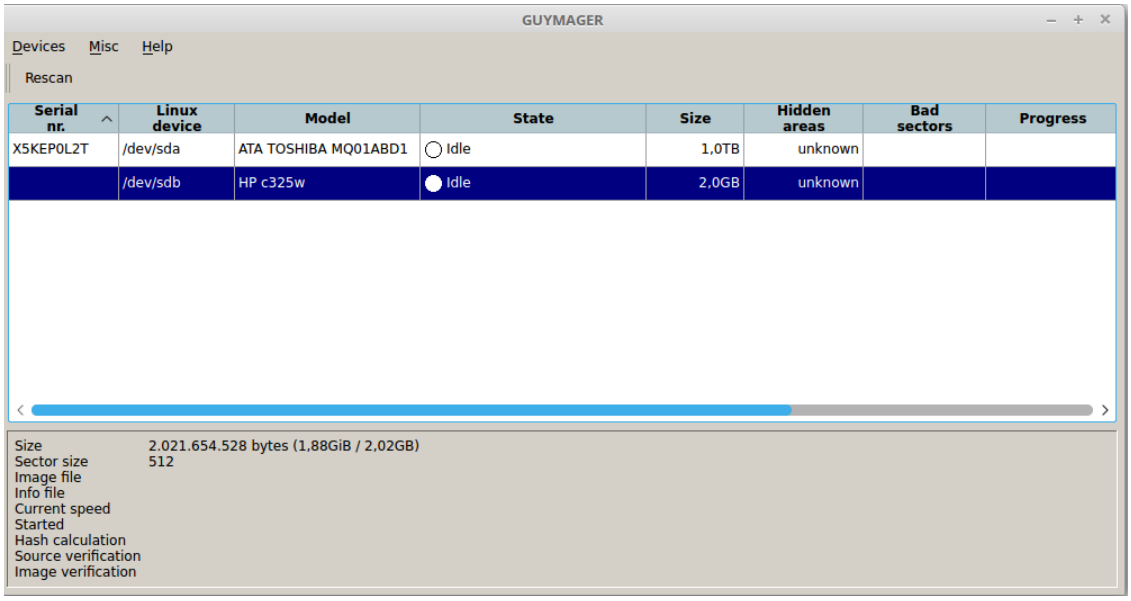
Ahora llegamos al punto clave: la búsqueda de texto en espacio disperso o no asignado. Este es el caso que nos concierne pues estamos tratando de encontrar el texto de un archivo borrado, por lo que debemos buscar en aquellos bloques de disco no en uso. Para ello utilizamos la conocida orden grep. Creamos una lista de palabras claves a buscar en el fichero */home/victor/evidencias/listaBusqueda.txt* donde cada linea contiene una palabra o frase (ha de ser un fichero sin lineas en blanco). Incluimos en la búsqueda la palabra *amenaza*. Mediante el siguiente comando volcamos los aciertos del comando grep en su búsqueda de estas palabras clave en el fichero aciertos.txt (ejecutamos desde la ruta /home/victor/evidencias):

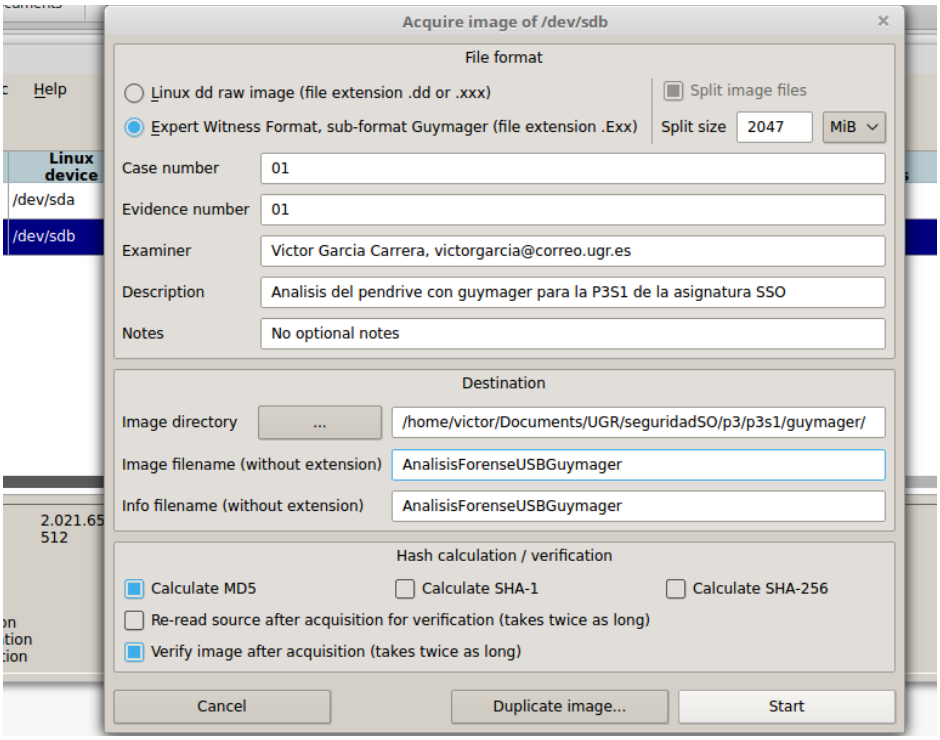
grep -aibf listaBusqueda.txt /home/victor/imagenUSB/imagen.disco1 > aciertos.txt

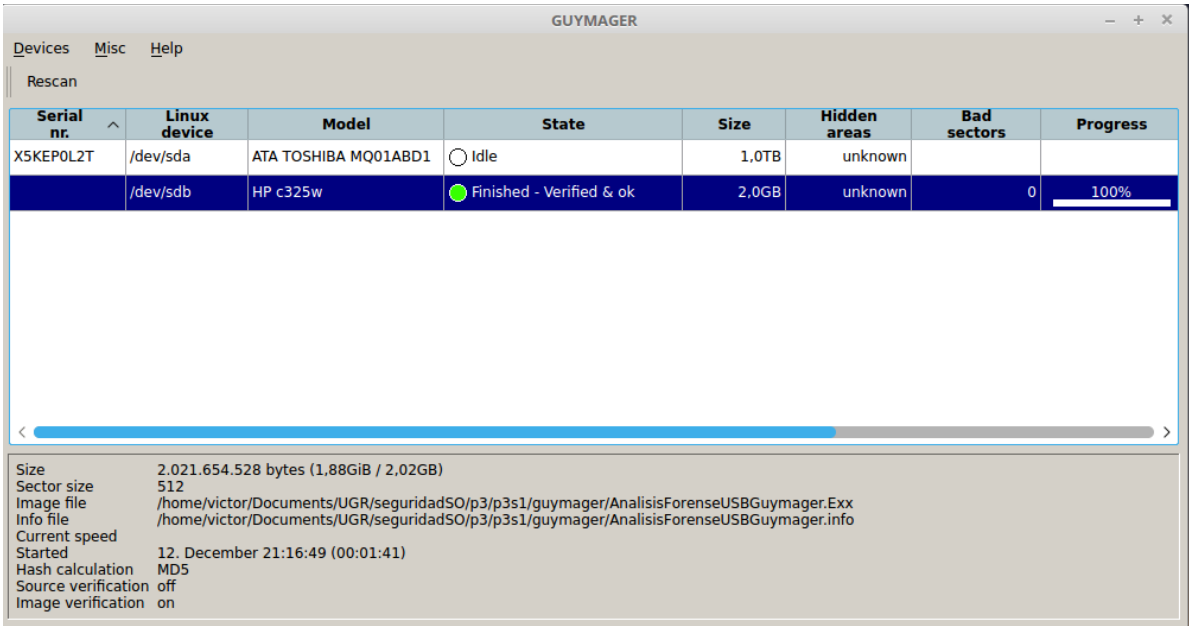


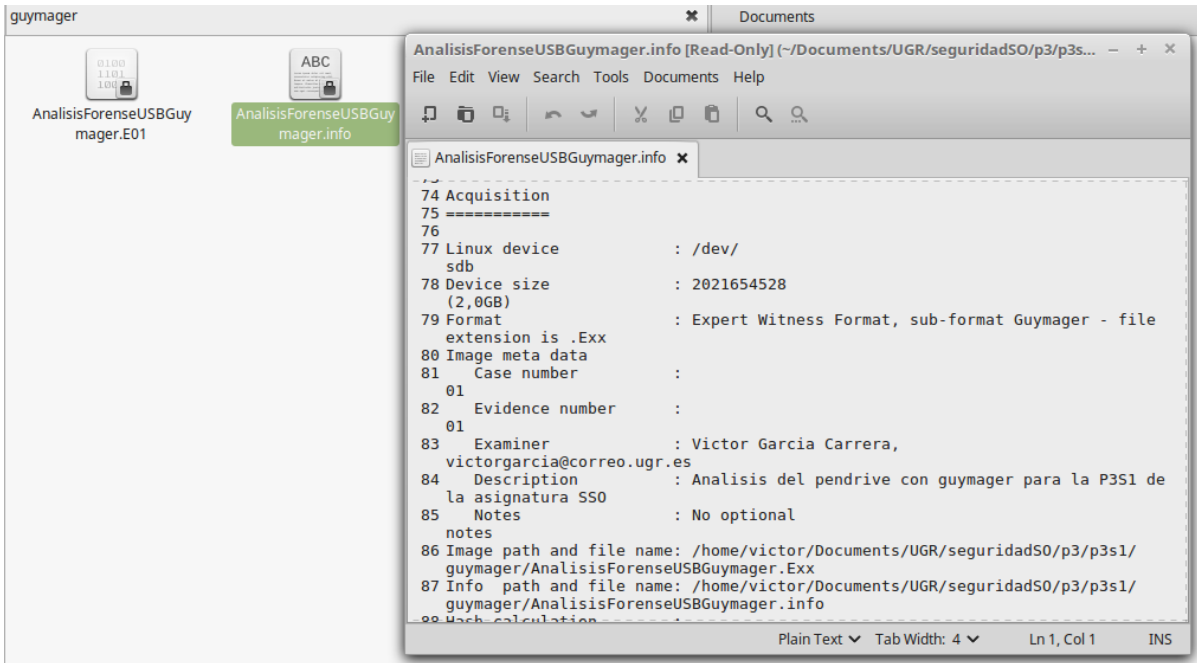
Ejercicio 2.

En este ejercicio utilizamos la herramienta *guymanager*, disponible mediante el comando *sudo apt install guymanager*, para crear una imagen forense del pendrive utilizado en el ejercicio anterior. A continuación se muestra el proceso seguido.



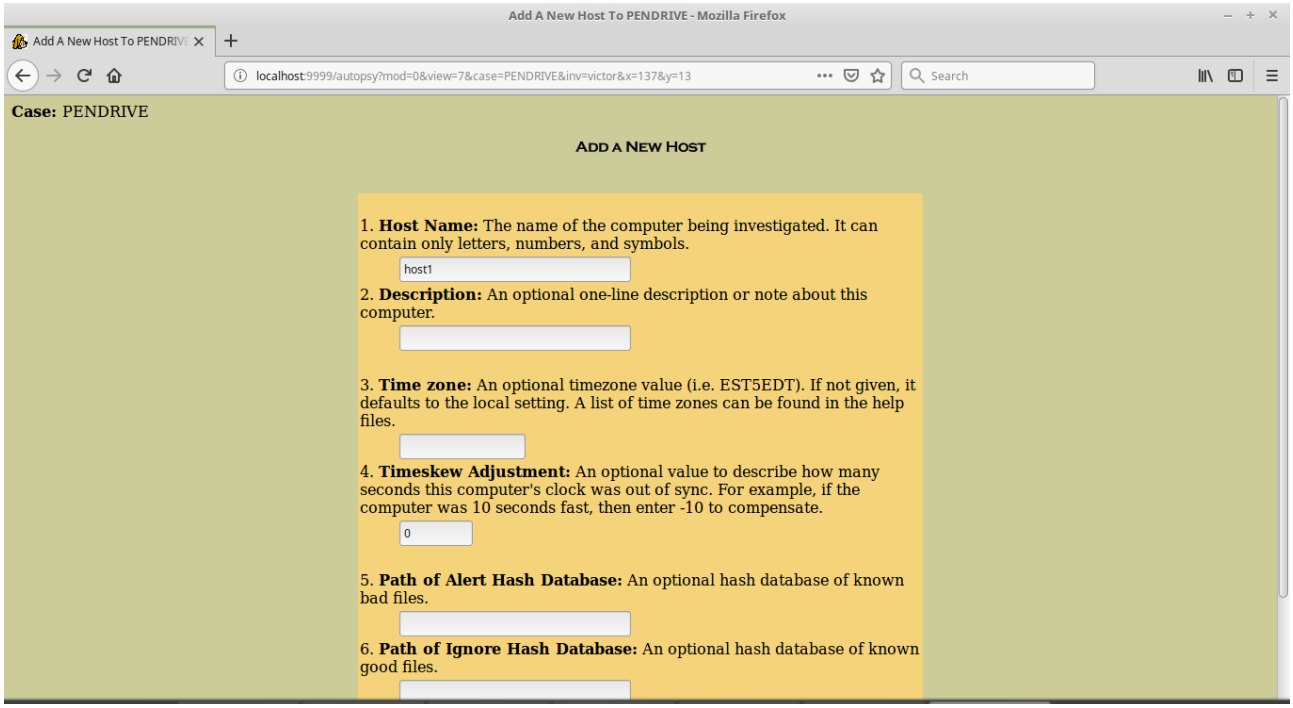


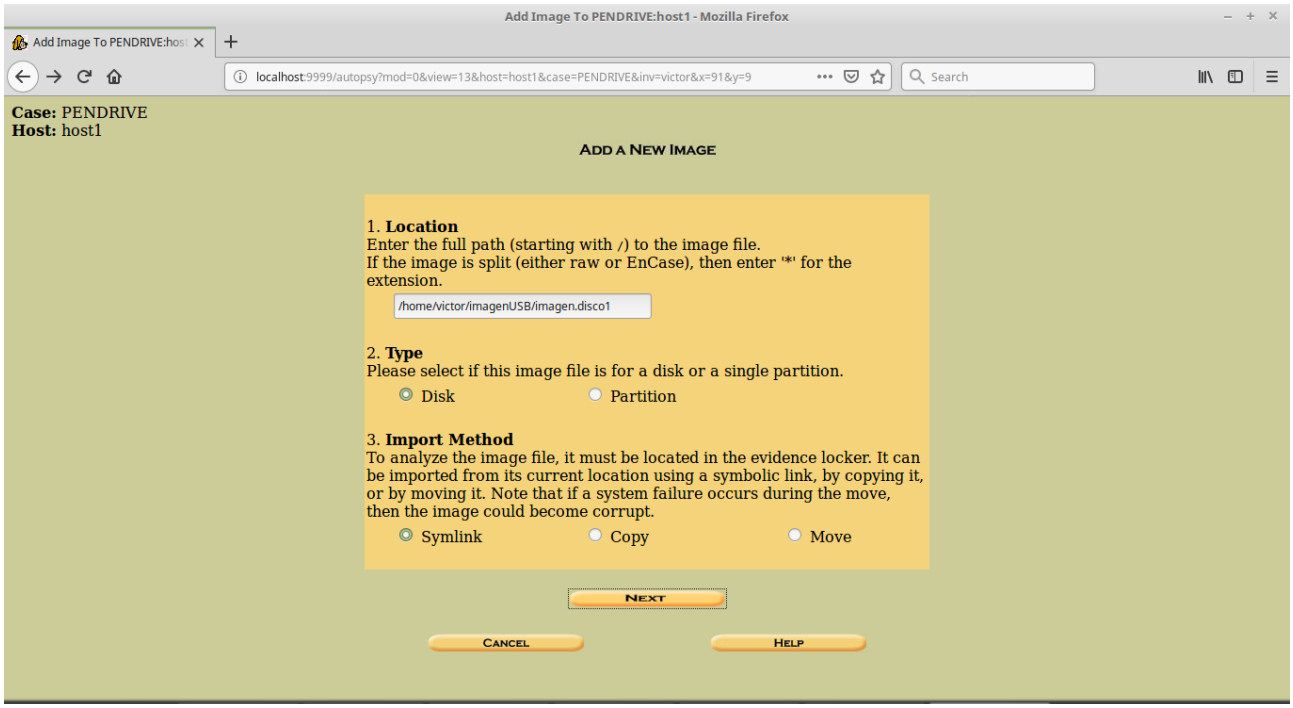




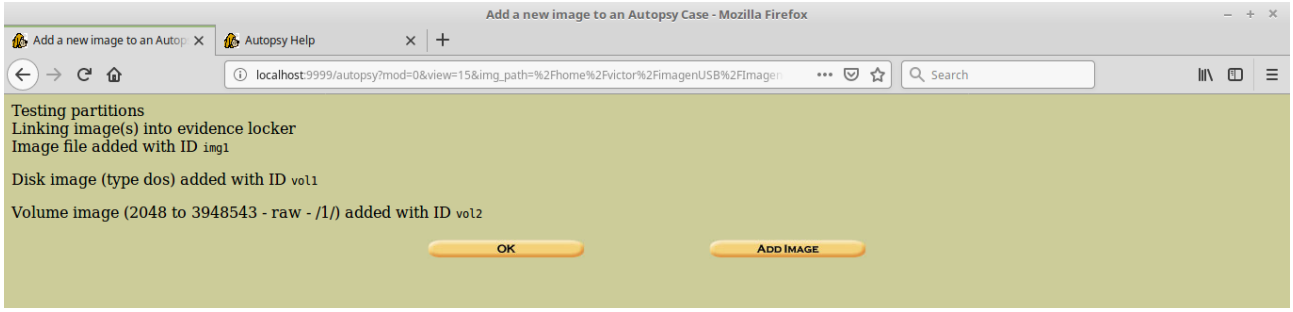
Ejercicio 3.

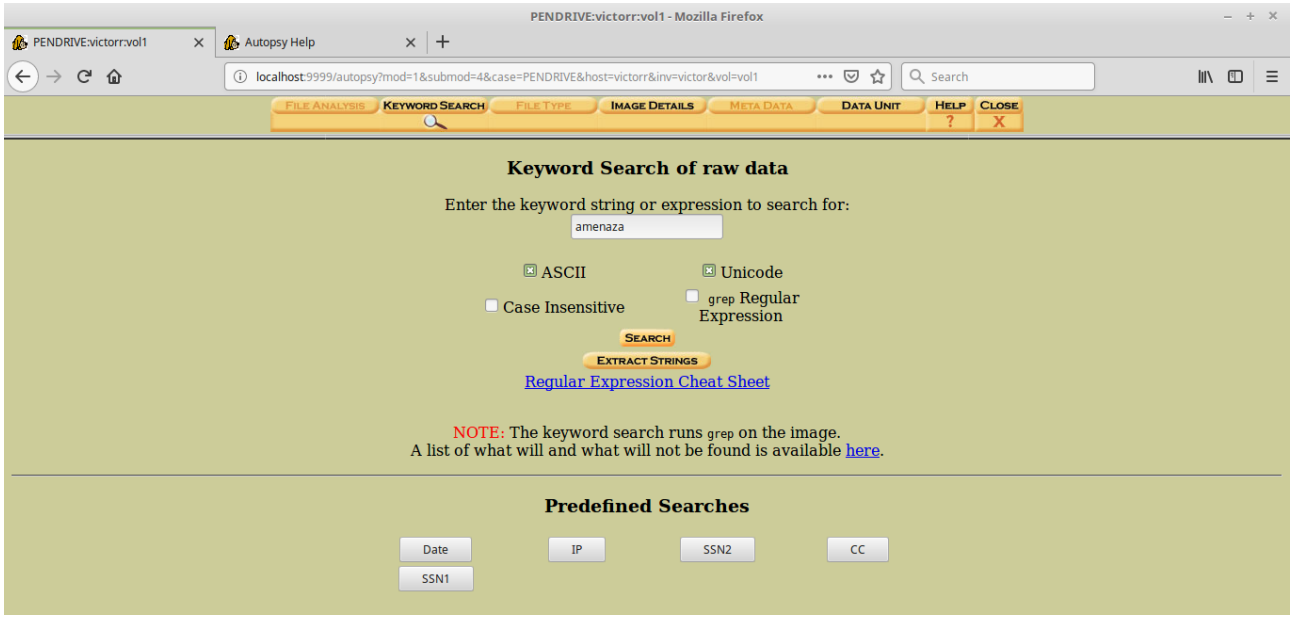
En este último ejercicio vamos a trabajar con la herramienta Autopsy. Siguiendo los pasos indicados en el documento de la práctica, no consigo descargarla, y recurro a instalar el repositorio mediante *sudo apt install autopsy* y ejecutando *sudo autopsy* nos aparece en terminal la dirección que introducir en un buscador HTML como Firefox: *http://localhost:9999/autopsy*. Una vez dentro, creamos un nuevo caso con la imagen forense del anterior ejercicio y buscamos evidencias por la palabra clave *amenaza* con el fin de encontrar con Autopsy la misma.

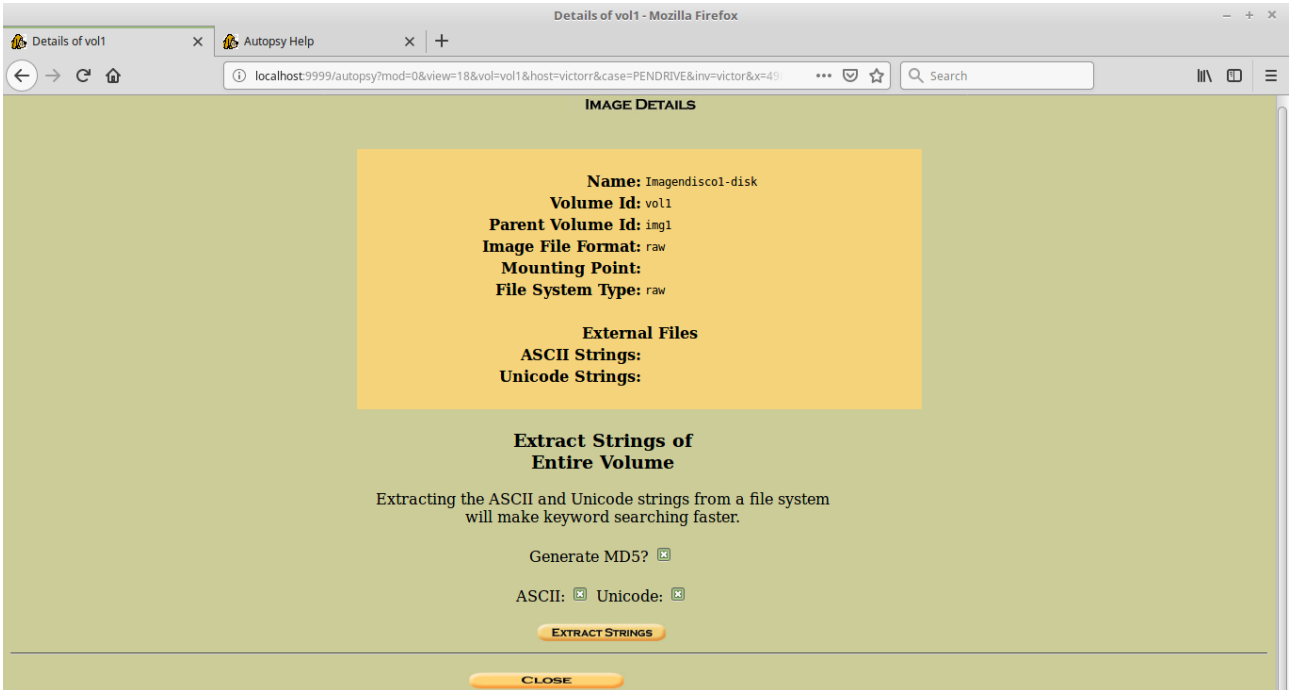


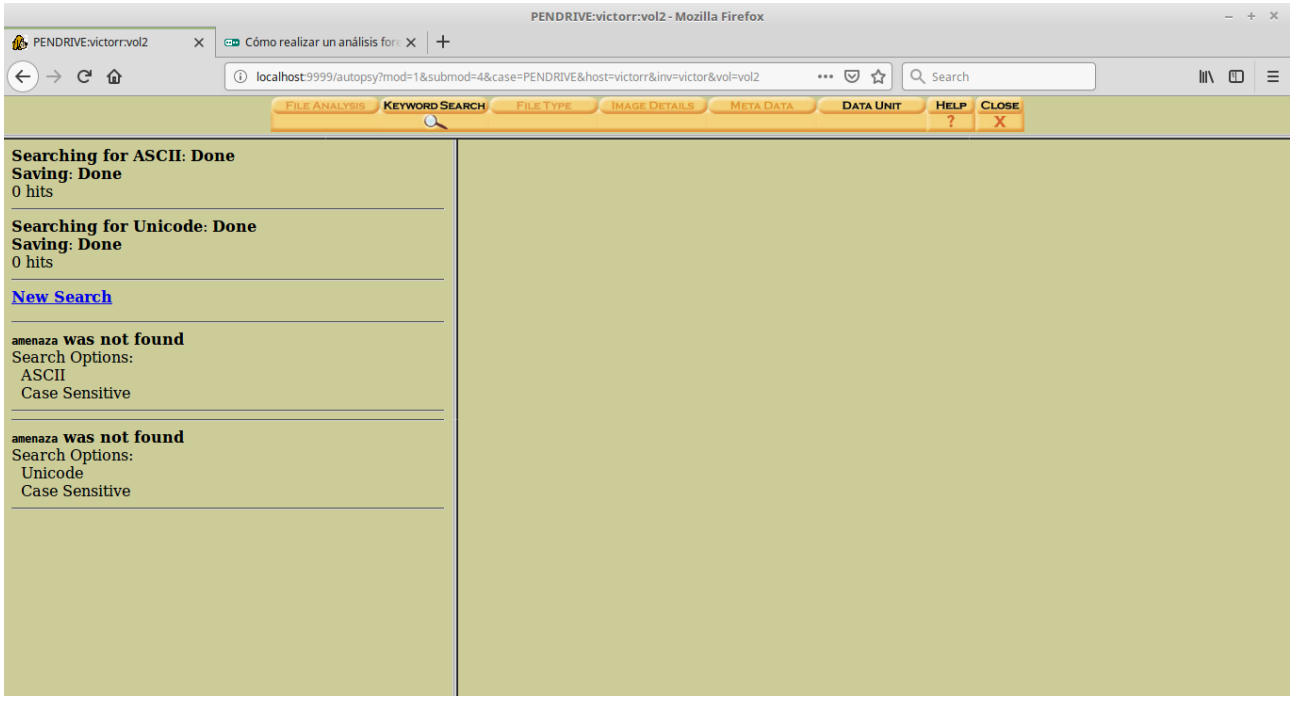


Desgraciadamente, no obtenemos resultados.




PENDRIVE:victorr:vol1 - Mozilla Firefox
PENDRIVE:victorr:vol1
Autopsy Help
localhost:9999/autopsy?mod=1&submod=4&case=PENDRIVE&host=victorr&inv=victor&vol=vol1
Search
FILE ANALYSIS
KEYWORD SEARCH
FILE TYPE
IMAGE DETAILS
META DATA
DATA UNIT
HELP
CLOSE
Keyword Search of raw data
Enter the keyword string or expression to search for:
amenaza
ASCII
Unicode
Case Insensitive
grep Regular Expression
SEARCH
EXTRACT STRINGS
Regular Expression Cheat Sheet
NOTE: The keyword search runs grep on the image.
A list of what will and what will not be found is available here.
Predefined Searches
Date
IP
SSN2
CC
SSN1


Details of vol1 - Mozilla Firefox
Details of vol1
Autopsy Help
localhost:9999/autopsy?mod=0&view=18&vol=vol1&host=victorr&case=PENDRIVE&inv=victor&x=49
Search
IMAGE DETAILS
Name: Imagendisco1-disk
Volume Id: vol1
Parent Volume Id: img1
Image File Format: raw
Mounting Point:
File System Type: raw
External Files
ASCII Strings:
Unicode Strings:
Extract Strings of Entire Volume
Extracting the ASCII and Unicode strings from a file system will make keyword searching faster.
Generate MD5?
ASCII:
Unicode:
EXTRACT STRINGS
CLOSE


PENDRIVE:victorr:vol2 - Mozilla Firefox
PENDRIVE:victorr:vol2
Cómo realizar un análisis for
localhost:9999/autopsy?mod=1&submod=4&case=PENDRIVE&host=victorr&inv=victor&vol=vol2
Search
FILE ANALYSIS
KEYWORD SEARCH
FILE TYPE
IMAGE DETAILS
META DATA
DATA UNIT
HELP
CLOSE
Searching for ASCII: Done
Saving: Done
0 hits
Searching for Unicode: Done
Saving: Done
0 hits
New Search
amenaza was not found
Search Options:
ASCII
Case Sensitive
amenaza was not found
Search Options:
Unicode
Case Sensitive

Para simular un caso más realista, utilizamos dos imagenes forenses descargadas de un caso real en el que se produjo una filtración de documentos corporativos desde un ordenador de un alto ejecutivo de la empresa M57 y se publicaron en un foro del sitio web de la competencia. Utilizando Autopsy, debemos obtener información a partir de su análisis.

Por un fallo con la aplicación, no es posible obtener resultados.