

## SEGURIDAD EN SISTEMAS OPERATIVOS

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

Práctica [1]. Encriptación/desencriptación en Linux

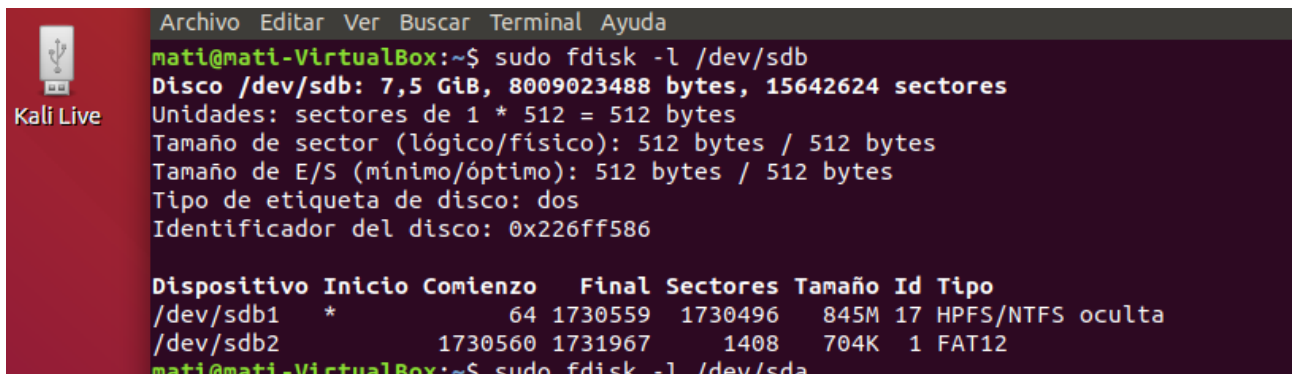
Sesión [6]. Criado de sistemas de archivos. Esteganografía y estegoanálisis.

Autor<sup>1</sup>: Matilde Cabrera González

### Ejercicio 1.

Utilizar cryptsetup para crear una partición encriptada en un pendrive. Escribir un archivo en él. Desmontarlo y extraerlo. ¿Qué ocurre cuando volvemos a conectarlo?

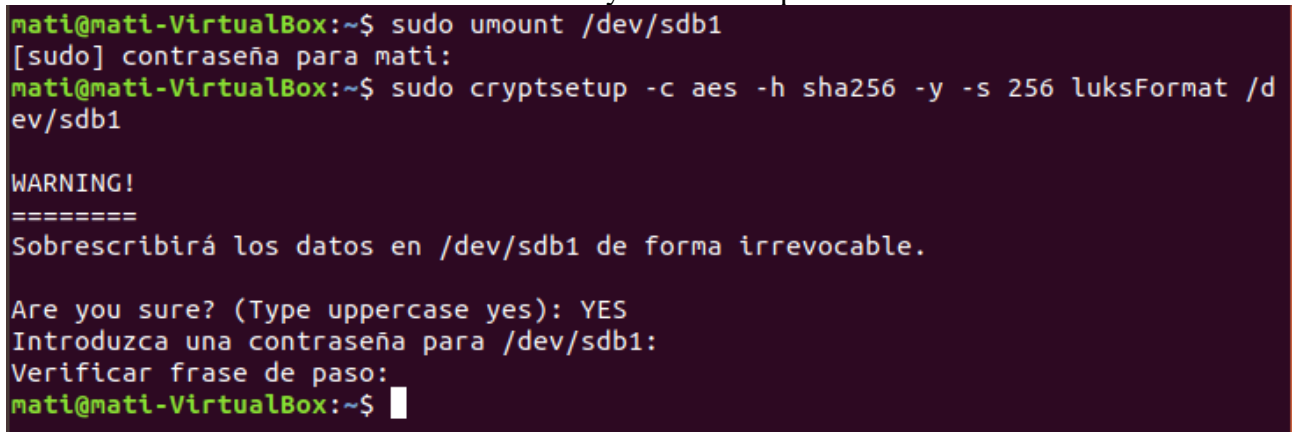
Instalamos la herramienta con “sudo apt install cryptsetup”. Tenemos un pendrive antiguo con “Kali live”. Introducimos el pendrive y lo abrimos en la máquina virtual donde hacemos esta práctica. Buscamos el dispositivo:



```
Archivo Editar Ver Buscar Terminal Ayuda
mati@mati-VirtualBox:~$ sudo fdisk -l /dev/sdb
Disco /dev/sdb: 7,5 GiB, 8009023488 bytes, 15642624 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x226ff586

Dispositivo Inicio Comienzo Final Sectores Tamaño Id Tipo
/dev/sdb1 * 64 1730559 1730496 845M 17 HPFS/NTFS oculta
/dev/sdb2 1730560 1731967 1408 704K 1 FAT12
mati@mati-VirtualBox:~$ sudo fdisk -l /dev/sda
```

Desmontamos /dev/sdb1: “umount /dev/sdb1” y ciframos la partición:



```
mati@mati-VirtualBox:~$ sudo umount /dev/sdb1
[sudo] contraseña para mati:
mati@mati-VirtualBox:~$ sudo cryptsetup -c aes -h sha256 -y -s 256 luksFormat /dev/sdb1

WARNING!
=====
Sobrescribirá los datos en /dev/sdb1 de forma irrevocable.

Are you sure? (Type uppercase yes): YES
Introduzca una contraseña para /dev/sdb1:
Verificar frase de paso:
mati@mati-VirtualBox:~$
```

---

<sup>1</sup> 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 ...”

Contraseña: 0123456789. La palabra YES tiene que estar en mayuscula.

Creamos un sistema de archivos mapeado llamado ps6.

```
mati@mati-VirtualBox:~$ sudo cryptsetup luksOpen /dev/sdb1 p1s6
Introduzca una contraseña para /dev/sdb1:
mati@mati-VirtualBox:~$ sudo mkfs /dev/mapper/p1s6
mke2fs 1.44.1 (24-Mar-2018)
Se está creando un sistema de ficheros con 215800 bloques de 4k y 53984 nodos-i
UUID del sistema de ficheros: 5af892b8-c7c3-4b00-adc7-a0db8ac74c0e
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Escribiendo superbloques y la información contable del sistema de archivos: hecho
```

Preparamos el entorno para montar el dispositivo mapeado:

```
mati@mati-VirtualBox:~$ sudo mkdir /mnt/pruebasso
mati@mati-VirtualBox:~$ sudo chmod 777 /mnt/pruebasso
mati@mati-VirtualBox:~$ mount /dev/mapper/p1s6 /mnt/pruebasso
mount: sólo el usuario root puede efectuar esa acción
```

Desmontamos el dispositivo mapeado y quitamos el pendrive:

```
mati@mati-VirtualBox:~$ sudo umount /dev/mapper/p1s6
mati@mati-VirtualBox:~$ sudo cryptsetup luksClose /dev/mapper/p1s6
mati@mati-VirtualBox:~$
```

Quitamos el pendrive y lo volvemos a poner. No logramos acceder a ningún dato

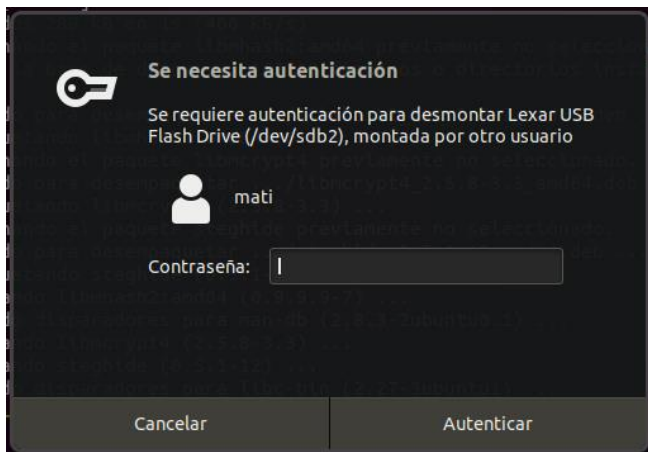
```
mati@mati-VirtualBox:~$ sudo fdisk -l /dev/sdb
Disco /dev/sdb: 7,5 GiB, 8009023488 bytes, 15642624 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x226ff586

Dispositivo Inicio Comienzo Final Sectores Tamaño Id Tipo
/dev/sdb1 * 64 1730559 1730496 845M 17 HPFS/NTFS oculta
/dev/sdb2 1730560 1731967 1408 704K e W95 FAT16 (LBA)
mati@mati-VirtualBox:~$ cat /media/mati/8EE5-FE94/System\ Volume\ Information\IndexerVolumeGuid
{C05DAE63-7BCB-4C3E-8481-323A71DDC63F}
mati@mati-VirtualBox:~$
```

Damos los mismos pasos para acceder a la información cifrada:

```
mati@mati-VirtualBox:~$ sudo cryptsetup luksOpen /dev/sdb1 s1p6
Introduzca una contraseña para /dev/sdb1:

mati@mati-VirtualBox:~$ sudo mount /dev/mapper/s1p6 /media
```



## Ejercicio 2.

Utilizar la herramienta *Steghide* para ocultar un mensaje dentro de una imagen, tal como acabamos de ver. Comparar los archivos portadores antes y después de usar la técnica para ver las diferencias.

Instalamos la herramienta:

```
mati@mati-VirtualBox:~$ sudo apt install steghide
[sudo] contraseña para mati:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libmcrypt4 libmhash2
Paquetes sugeridos:
  libmcrypt-dev mcrypt
Se instalarán los siguientes paquetes NUEVOS:
  libmcrypt4 libmhash2 steghide
```

Me descargo una imagen, la duplico, tendremos “imagenoriginal.jpeg” y “imagencopia.jpeg”, vamos a ocultar el texto en “imagencopia.jpeg” y hago un documento para ocultar en la misma

```
mati@mati-VirtualBox:~$ touch toculto.txt
mati@mati-VirtualBox:~$ sudo nano toculto.txt
mati@mati-VirtualBox:~$ ls
Descargas  Escritorio  imagen.jpeg  p1s5  Público  toculto.txt
```

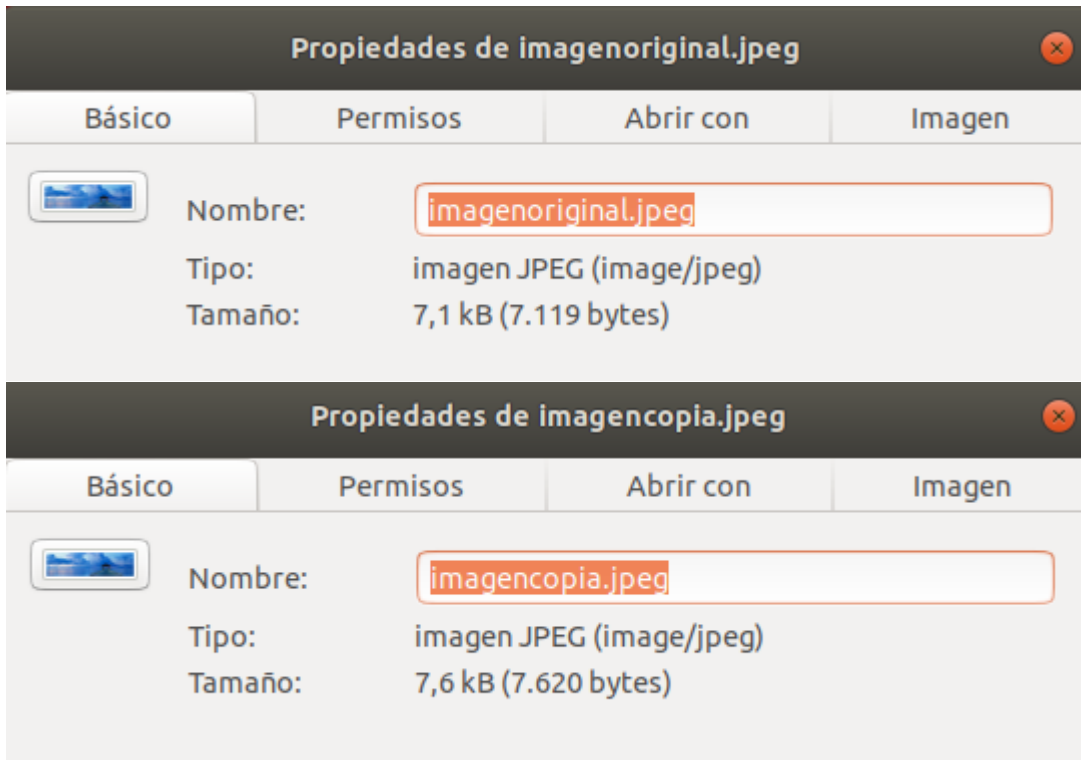
```
mati@mati-VirtualBox: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.9.3  toculto.txt

Vamos a ocultar este mensaje en una imagen
```

Ocultamos el mensaje:

```
mati@mati-Lenovo-50-70:~$ steghide embed -cf imagencopia.jpeg -ef toculito.txt
Anotar salvoconducto:
Re-ingresar salvoconducto:
adjuntando "toculto.txt" en "imagencopia.jpeg"... hecho
mati@mati-Lenovo-50-70:~$
```

Podemos comprobar como las imágenes son iguales pero la copia pesa más, es más grande.



Visualmente son iguales, permisos iguales.

```
-rw-rw-r-- 1 mati mati 7620 nov 27 12:08 imagencopia.jpeg
drwxr-xr-x 2 mati mati 4096 sep 27 13:11 Imágenes
-rw-rw-r-- 1 mati mati 7119 nov 27 12:07 imagenoriginal.jpeg
drwxr-xr-x 2 mati mati 4096 sep 27 13:11 Música
```

Generamos hash y son diferentes.

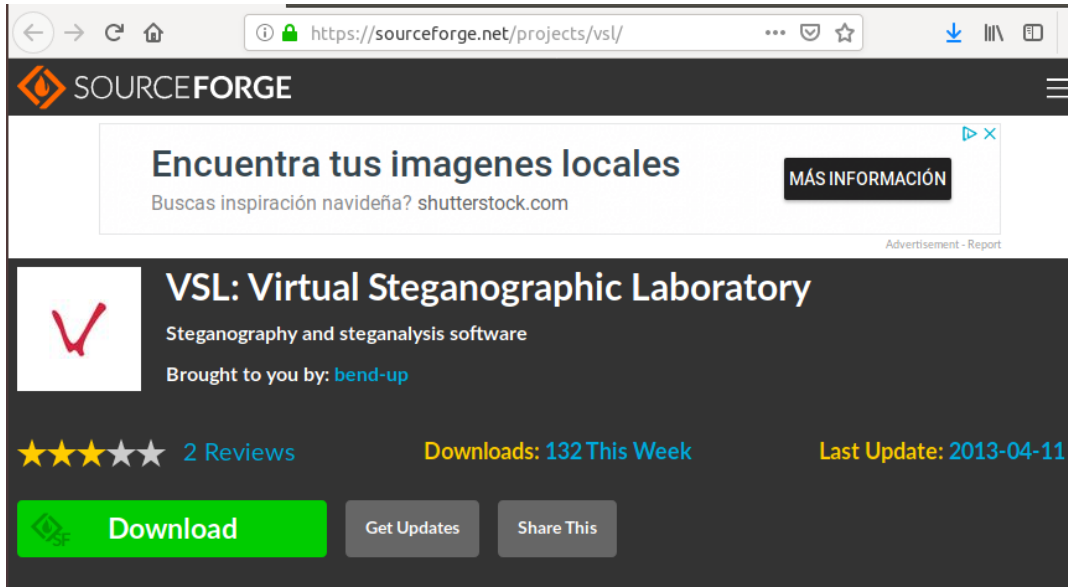
```
mati@mati-VirtualBox:~$ md5sum imagenoriginal.jpeg
26eeab9beeac725d33007458ed355de3 imagenoriginal.jpeg
mati@mati-VirtualBox:~$ md5sum imagencopia.jpeg
7d4b5821dbbd0d53b344720c284ce8b4 imagencopia.jpeg
mati@mati-VirtualBox:~$
```

En conclusión, para saber si una imagen tiene algo oculto tenemos que tener la original para comparar y tenemos que usar métodos como la generación de hash.

### Ejercicio 3.

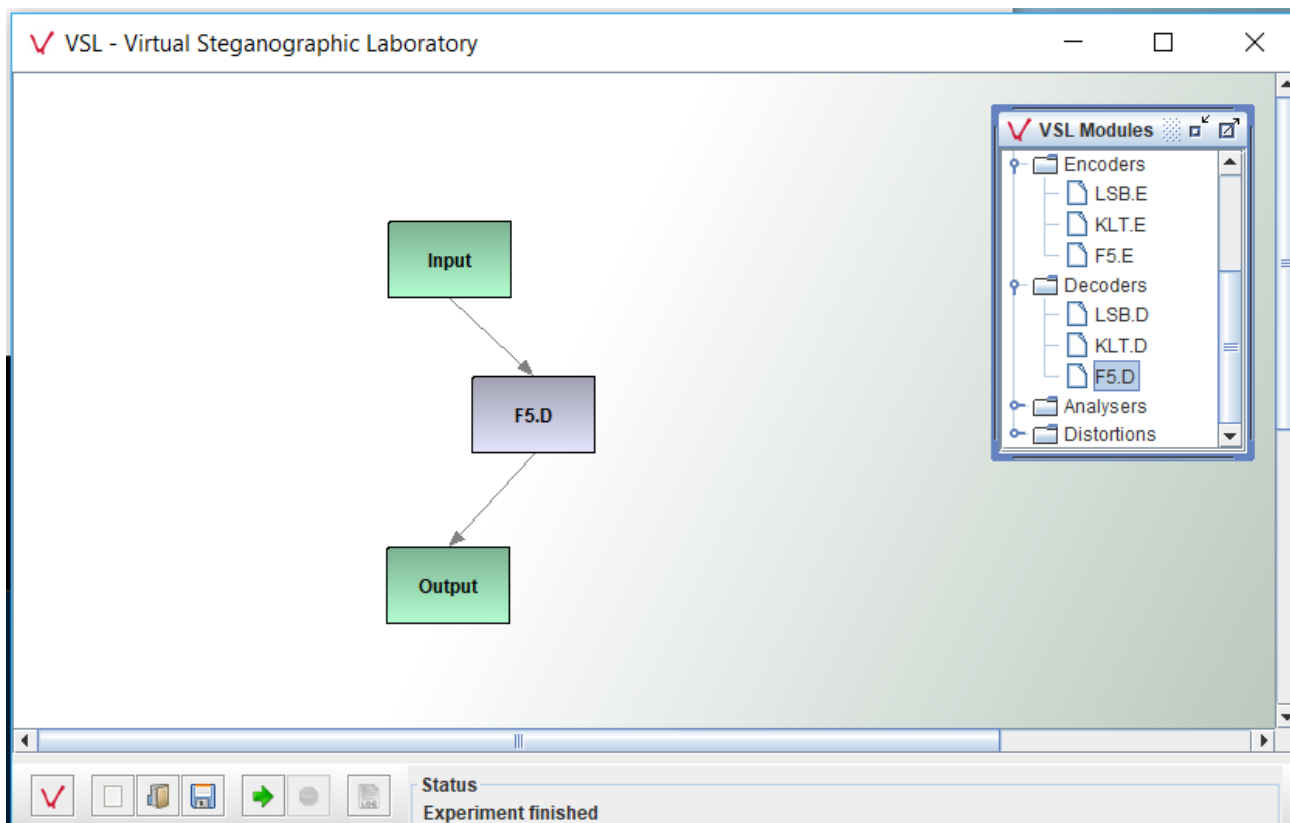
Utilizar VSL para analizar la imagen esteganográfica generada en el ejercicio anterior para detectar información oculta.

Buscamos la herramienta a descargar:



No logro hacer que ejecute en Ubuntu, paso mis archivos a la maquina anfitrión y ejecuto el programa.

```
C:\Users\chipi>ssh mati@192.168.56.200 "gzip -c /home/mati/imagen.zip" > C:\Users\chipi\Desktop\VB0X\IMG.gz
mati@192.168.56.200's password:
C:\Users\chipi>
```



Solo nos ha funcionado F5D

En input hemos conectado la imagen “imagencopia.jpeg”. Asociamos la salida a la carpeta VBOX. Nos da como salida un fichero llamado result0000. El cual tiene el mensaje cifrado que ocultemos en la imagen.

> VBOX > 2018-11-27-16-57-38-409 > input000				Buscar en input000
Nombre		Fecha de modifica...	Tipo	Tamaño
result00000		27/11/2018 16:57	Archivo	1 KB