

SIT282/SIT703 Computer Forensics and Investigations

Workshop Session 4

Through this session, you will practice three forensic software tools – VeraCrypt, dd, and the Sleuth Kit (TSK). By solving forensic tasks, you will be able to perform simple operations on a file and acquire dd images from the mounted VeraCrypt volume. You will also practice some advance usage of the Sleuth Kit. The Ubuntu VM is used in this session.

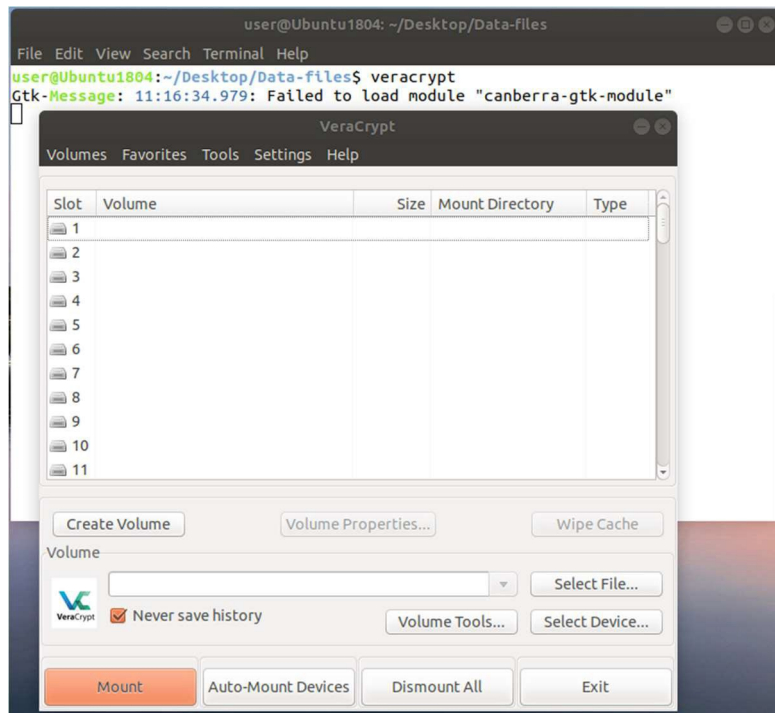
Learning Objectives

1. Explain the purpose of volume encryption and how it is used in both computer crime and digital forensics.
2. Determine whether there is a fixed pattern to detect the existence of encrypted volumes on an evidence drive.
3. Demonstrate you can recover items from a mounted encrypted volume.
4. Practice some advanced features of the “Sleuth Kit” for further analysis of disk images and file recovery.
5. Demonstrate the skills to recover forensic data by applying the combined knowledge of digital forensic procedure and tools learned over the past weeks.

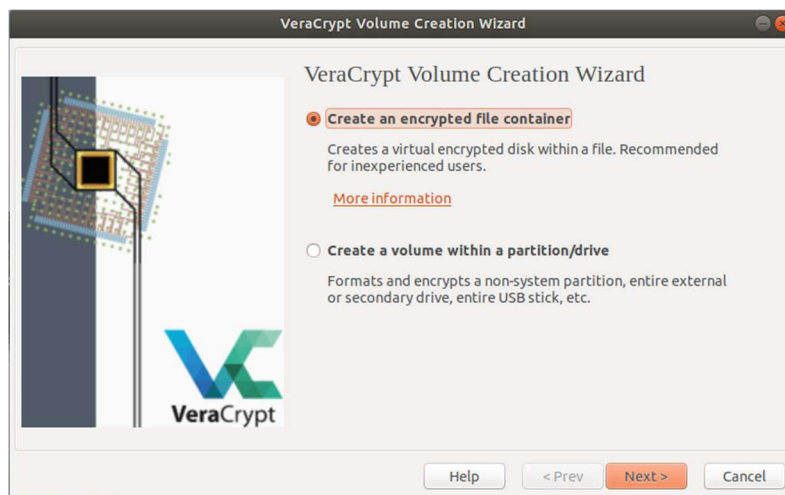
1. Introducing VeraCrypt

VeraCrypt is a successor of the famous tool TrueCrypt which was retired due to many reasons. You may find some interesting stories from some mainstream media at https://www.theregister.co.uk/2015/08/04/truecrypt_decrypted_by_fbi/ and [medium.com](https://www.medium.com).

Nevertheless, VeraCrypt offers a similar level of user experience to TrueCrypt. Since it supports multiple OS platforms, we have installed a version on our virtual machine. Launch a “Terminal” and change directory to “cd ~/Desktop/Data-files/week04”, and type the command “veracrypt”.



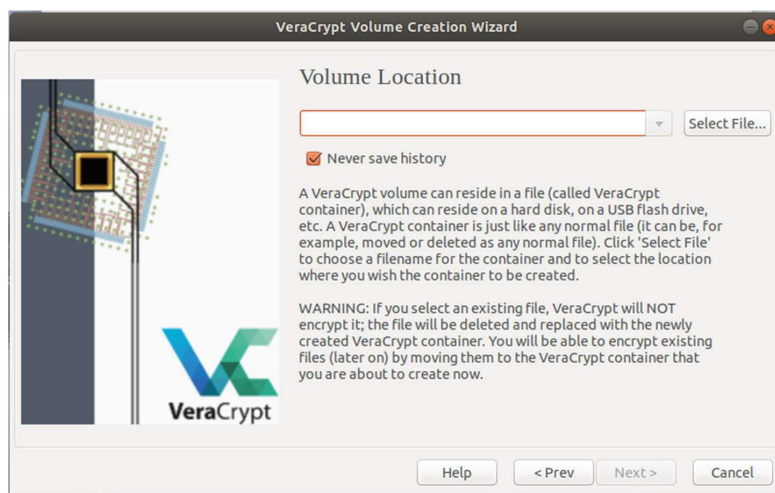
Click the button “Create Volume” and choose the default option:



Click Next, choose the standard volume option:

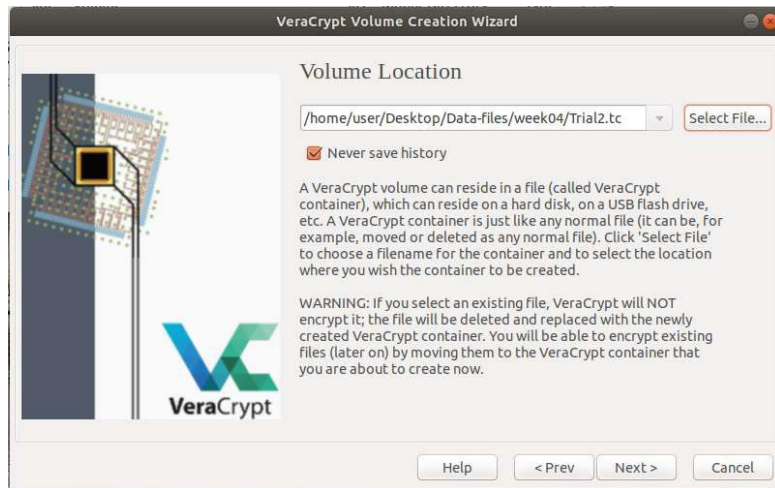


Click Next, you are asked to select file to encrypt. Read the text in the dialog box **carefully**.



Note that VeraCrypt will not encrypt any existing files. If you select an existing file, it will be deleted and replaced by the newly created container (so the existing file will be lost, not encrypted). You can encrypt existing files (later on) by moving them to the VeraCrypt container that you are about to create now.

Select “~/Desktop/Data-files/week04” in the file selector. Type the container filename as “Trial2.tc” in the File name box. Click Save. Then the file selector window should disappear.

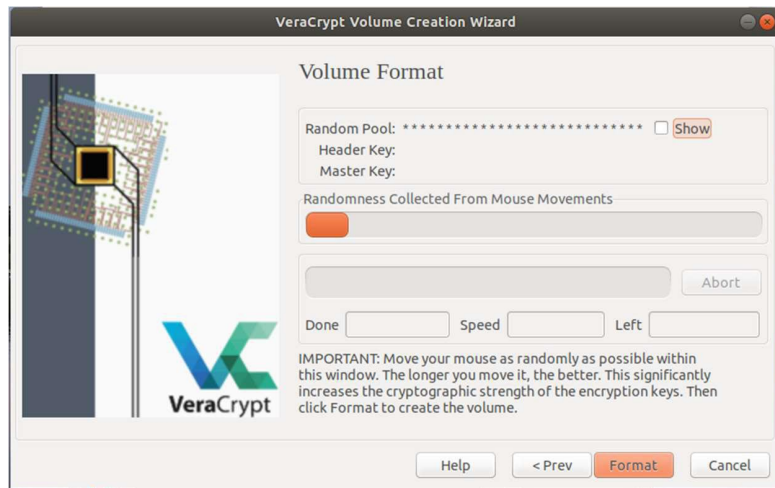


In the Volume Creation Wizard window, click Next. (Choose the default cipher scheme **AES** and the default hash algorithm **SHA-512**), click Next. Choose a small size, e.g. 1MB, at this point, click Next.

Use “deakin” as the volume password before you proceed to the next step.



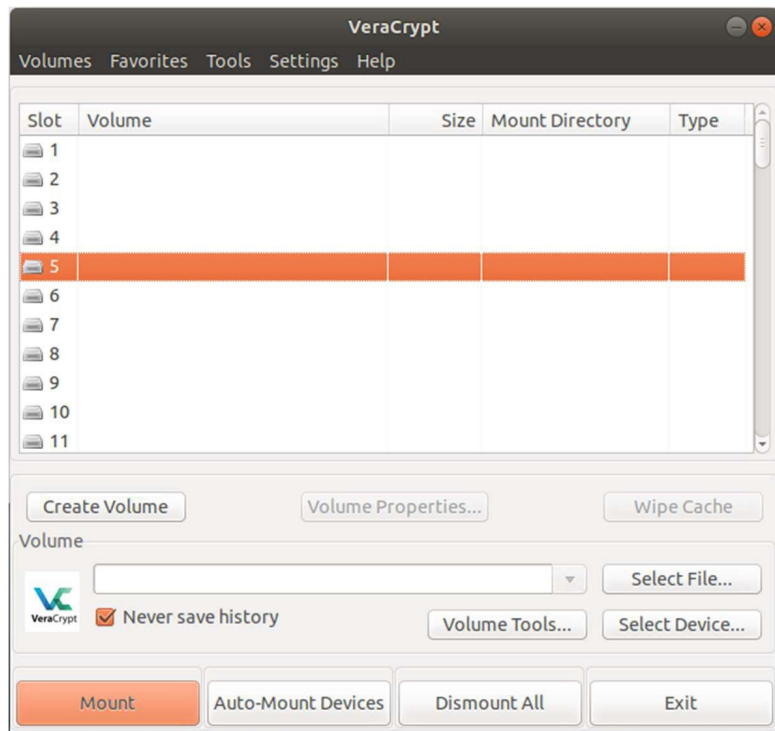
Then choose the file system format. Use the default settings as FAT. Click Next. Move your mouse cursor around within the window for approx. 30 seconds to generate a random key. Click the “Format” button and the creation should be finished in a short while.



After the volume is created, click “Exit”.



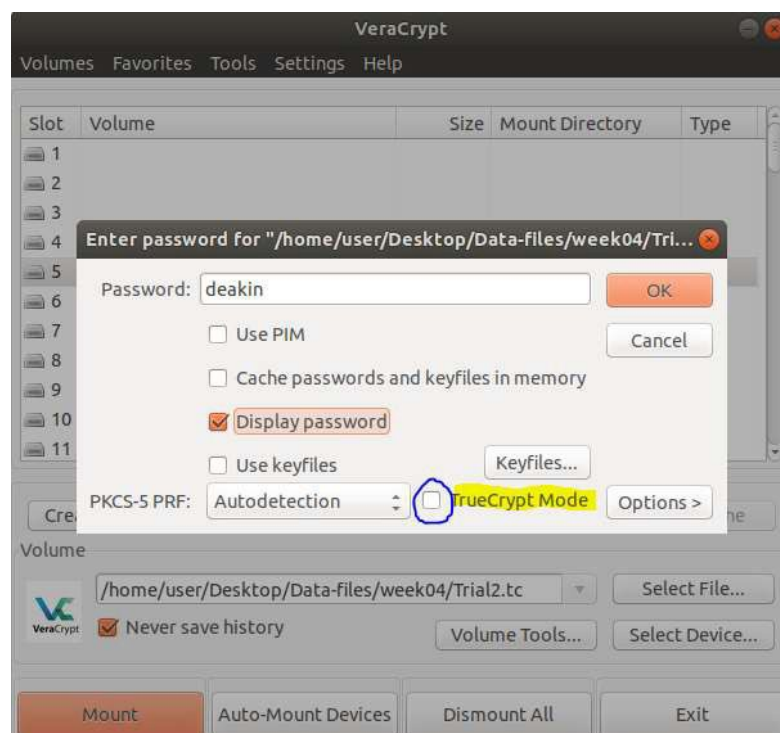
Back to the main menu of VeraCrypt, and mount the volume that you just created. Choose Drive 5 as shown below.



Select your new volume by clicking “Select File...” and navigate to the file “~/Desktop/Data-files/week04/Trial2.tc”, click “Mount”.

Type in the password “deakin”, you should be able to access the new drive 5. If the system asks for the administrator password, it is “user”.

NOTE: To avoid error, ensure you **deselect** “TrueCrypt Mode” as shown below, since this is selected by default.



Make a random text file and save it on your mounted drive.

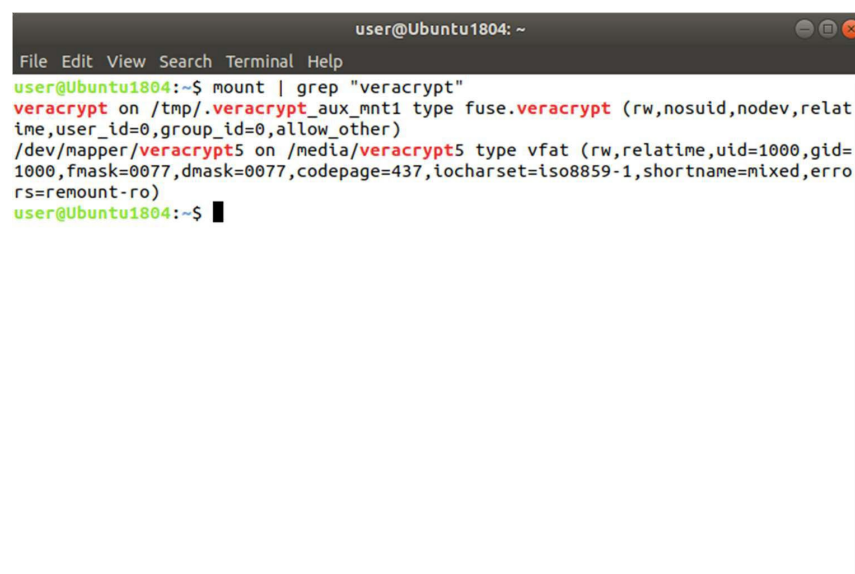
Dismount the drive 5 and quit VeraCrypt.

Encrypt two 1MB VeraCrypt volumes with two different passwords of your choice. Compare the hash values of the two volumes and determine whether there is a fixed pattern to detect the existence of VeraCrypt volumes.

2. Recovering Items of a Mounted VeraCrypt Volume

To demonstrate how to recover items from a mounted VeraCrypt volume, we created a sample “Trial.tc” volume. Originally it had four files: three image files logo.jpg, logo.gif and logo.bmp and a text file before we deleted the three image files. Now let us see if we can recover them correctly.

Assume the “Trial.tc” volume has been mounted using VeraCrypt on a drive (e.g. drive 5). In a Terminal, we first find the mounted volume by typing the command “**mount | grep “veracrypt”**”. In this case, it is mounted to /media/veracrypt5, and it is located in /dev/mapper/veracrypt5. (Yours output might be different as it depends on where you mounted “Trial.tc”).



```
user@Ubuntu1804: ~  
File Edit View Search Terminal Help  
user@Ubuntu1804:~$ mount | grep "veracrypt"  
veracrypt on /tmp/.veracrypt_aux_mnt1 type fuse.veracrypt (rw,nosuid,nodev,relatime,user_id=0,group_id=0,allow_other)  
/dev/mapper/veracrypt5 on /media/veracrypt5 type vfat (rw,relatime,uid=1000,gid=1000,fmask=0077,dmask=0077,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro)  
user@Ubuntu1804:~$
```

Then, we use the “dd” tool to acquire a forensic copy of the mounted volume by typing the command “**sudo dd if=/dev/mapper/veracrypt5 of=Trial.dd**”


```
user@Ubuntu1804: ~/Desktop/Data-files/week04
File Edit View Search Terminal Help
user@Ubuntu1804:~$ mount | grep "veracrypt"
veracrypt on /tmp/.veracrypt_aux_mnt1 type fuse.veracrypt (rw,nosuid,nodev,relatime,user_id=0,group_id=0,allow_other)
/dev/mapper/veracrypt5 on /media/veracrypt5 type vfat (rw,relatime,uid=1000,gid=1000,fmask=0077,dmask=0077,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro)
user@Ubuntu1804:~$ cd Desktop/Data-files/week04/
user@Ubuntu1804:~/Desktop/Data-files/week04$ sudo dd if=/dev/mapper/veracrypt5 of=
f=trial.dd
[sudo] password for user:
1536+0 records in
1536+0 records out
786432 bytes (786 kB, 768 KiB) copied, 0.00563112 s, 140 MB/s
user@Ubuntu1804:~/Desktop/Data-files/week04$ ls -l
total 1792
-rw-r--r-- 1 root root 786432 Jun 28 12:17 trial.dd
-rw----- 1 user user 1048576 Jun 28 12:15 trial.tc
user@Ubuntu1804:~/Desktop/Data-files/week04$
```

Once the dd dump is ready, we can use the Sleuth kit's recover facility "tsk_recover" to extract all items from the dd image. Type in the command "**tsk_recover -e trial.dd ./recover**", the option "-e" helps us extract all items including the deleted ones. Then we list the contents in the recover folder. In this case, we find all the four items.

```
user@Ubuntu1804: ~/Desktop/Data-files/week04
File Edit View Search Terminal Tabs Help
user@Ubuntu1804:~/Desktop/Data-files/week04$ tsk_recover -e trial.dd ./recover
Files Recovered: 4
user@Ubuntu1804:~/Desktop/Data-files/week04$ ls -l recover/
total 752
-rw-r--r-- 1 user user 659456 Jun 28 12:19 logo.bmp
-rw-r--r-- 1 user user 30280 Jun 28 12:19 logo.gif
-rw-r--r-- 1 user user 72147 Jun 28 12:19 logo.jpg
-rw-r--r-- 1 user user 70 Jun 28 12:19 logo.txt
user@Ubuntu1804:~/Desktop/Data-files/week04$
```

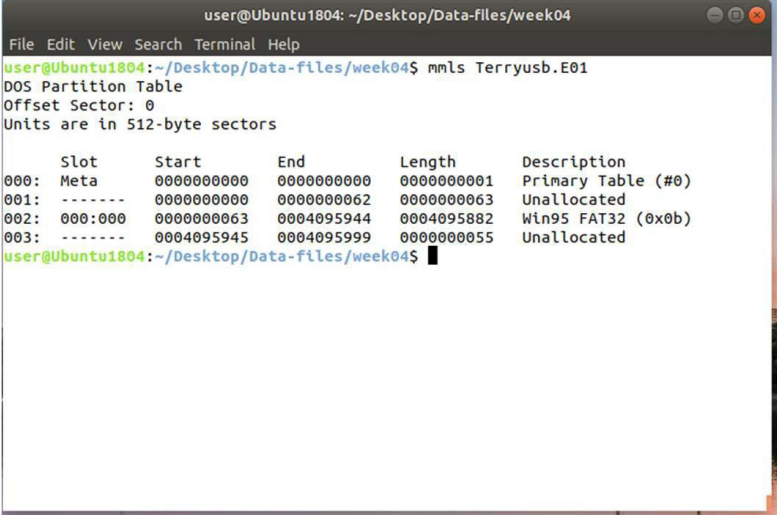
Spend 30 minutes or so to create a VeraCrypt volume of your own with some deleted files. Use this method to recover your files from the mounted volume.

3. Advanced Usage of the Sleuth Kit

In Section 2, we have used one of the automated tools in the Sleuth Kit to extract data. In this section, we use some more advanced tools so that you may recover files under your control.

Launch a “Terminal” if you have closed the previous one. Change the directory to “~/Desktop/Data-files/week04”. This time, we investigate a USB drive image named “Terryusb.E01”. The image is acquired by using EnCase forensic image format, which has several more features than the standard dd images.

Type command¹ “**mmls Terryusb.E01**” to see the partition¹ information of the USB drive. In the screenshot below, the drive has a FAT partition with the offset 63 sectors². Take the note of the offset number as it will be used throughout this section.



```
user@Ubuntu1804: ~/Desktop/Data-files/week04
File Edit View Search Terminal Help
user@Ubuntu1804:~/Desktop/Data-files/week04$ mmls Terryusb.E01
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

   Slot   Start      End      Length  Description
000: Meta  0000000000  0000000000  0000000001 Primary Table (#0)
001: ----- 0000000000  0000000062  0000000063 Unallocated
002: 000:000 0000000063  0004095944  0004095882 Win95 FAT32 (0x0b)
003: ----- 0004095945  0004095999  0000000055  Unallocated
user@Ubuntu1804:~/Desktop/Data-files/week04$
```

Next, use the tool “fsstat³” to obtain the information of the files and directories of this image. “fsstat” displays general details of a file system. Type in command “**fsstat -o 63 Terryusb.E01**”. The USB drive has a FAT32 file system as shown below.

¹ <http://www.sleuthkit.org/sleuthkit/man/mmls.html>

² <https://knowledgebase.macrium.com/display/KNOW/Understanding+partition+alignment>

³ <http://www.sleuthkit.org/sleuthkit/man/fsstat.html>

```
user@Ubuntu1804: ~/Desktop/Data-files/week04
File Edit View Search Terminal Help

user@Ubuntu1804:~/Desktop/Data-files/week04$ fsstat -o 63 Terryusb.E01
FILE SYSTEM INFORMATION
-----
File System Type: FAT32

OEM Name: BSD 4.4
Volume ID: 0x4a741208
Volume Label (Boot Sector): TERRY'S WORK
Volume Label (Root Directory):
File System Type Label: FAT32
Next Free Sector (FS Info): 158074
Free Sector Count (FS Info): 3937808

Sectors before file system: 0

File System Layout (in sectors)
Total Range: 0 - 4095881
* Reserved: 0 - 31
** Boot Sector: 0
** FS Info Sector: 1
** Backup Boot Sector: 6
* FAT 0: 32 - 4024
* FAT 1: 4025 - 8017
* Data Area: 8018 - 4095881
** Cluster Area: 8018 - 4095881
*** Root Directory: 8018 - 8025

METADATA INFORMATION
-----
Range: 2 - 65405830
Root Directory: 2

CONTENT INFORMATION
-----
Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 2 - 510984
```

“fls” lists file and directory names in a disk image. To list the items of this partition in the image, type the command “**fls -o 63 -rd Terryusb.E01**”, the option “-rd” will recursively list all the items⁴.

⁴ <http://www.sleuthkit.org/sleuthkit/man/fls.html>

```

user@Ubuntu1804: ~/Desktop/Data-files/week04
File Edit View Search Terminal Help
user@Ubuntu1804:~/Desktop/Data-files/week04$ fls -o 63 -rd Terryusb.E01
d/d * 133:      .Trashes/_01
r/r * 135:      .Trashes/_501
d/d * 10:       .fsevents
r/r * 1433:     .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/_7.IND
r/r * 1483:     .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/journalAttr.1
r/r * 1493:     .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/0.shadowIndexHead
r/r * 1525:     .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/tmp.0.cmpt.indexPostings
r/r * 1528:     .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/tmp.0.cmpt.indexHead
r/r * 1531:     .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/tmp.0.cmpt.indexPositions
r/r * 1534:     .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/tmp.0.cmpt.shadowIndexHead
r/r * 1535:     .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/_MP0CM-4.IND
r/r * 1538:     .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/tmp.0.cmpt.indexDirectory
r/r * 13958:    .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/tmp.0.cmpt.indexCompactDirectory
r/r * 13961:    .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/tmp.0.cmpt.indexArrays
r/r * 13964:    .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/tmp.0.cmpt.newTermIDMap
r/r * 13968:    .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/tmp.0.cmpt.compactPayloads1.idx
r/r * 13972:    .Spotlight-V100/Store-V1/Stores/7680DE76-88D9-43B3-AC7E-3B02E7F3
8194/tmp.0.cmpt.compactPayloads2.idx
r/r * 61:       _54402.EXE
d/d * 63:       _461531_
d/d * 64:       _604468_
d/d * 65:       _078421_
d/d * 66:       _452781_
d/d * 67:       _189812_
r/r * 70:       xpadvancedkeylogger.exe
user@Ubuntu1804:~/Desktop/Data-files/week04$

```

There are many files and items in the USB image. Let us try to manually extract one spotlight item with the inode⁵ number 1493. “icat” outputs the contents of a file based on its inode number. Type in the command⁶ “icat -o 63 -r Terryusb.E01 1493”, then the contents with some URLs will be extracted from the image:

```

user@Ubuntu1804: ~/Desktop/Data-files/week04
File Edit View Search Terminal Help
user@Ubuntu1804:~/Desktop/Data-files/week04$ icat -o 63 -r Terryusb.E01 1493
[ac OS X] 2[TTTTR;*****]com.apple.metadata:kMDItemWhereFrom
sbplist00[https://domex.nps.edu/domex/svn/src/m57patents/s_copyright.txt_]
https://domex.nps.edu/domex/svn/src/m57patents/
L[his resource fork intenti
onally left blank] user@Ubuntu1804:~/Desktop/Data-files/week04$

```

Spend 20 minutes or so to read the documentations of the Sleuth Kit at http://wiki.sleuthkit.org/index.php?title=The_Sleuth_Kit_commands. Try to use other commands that are not given in this workshop.

⁵ <https://en.wikipedia.org/wiki/Inode>

⁶ <http://www.sleuthkit.org/sleuthkit/man/icat.html>