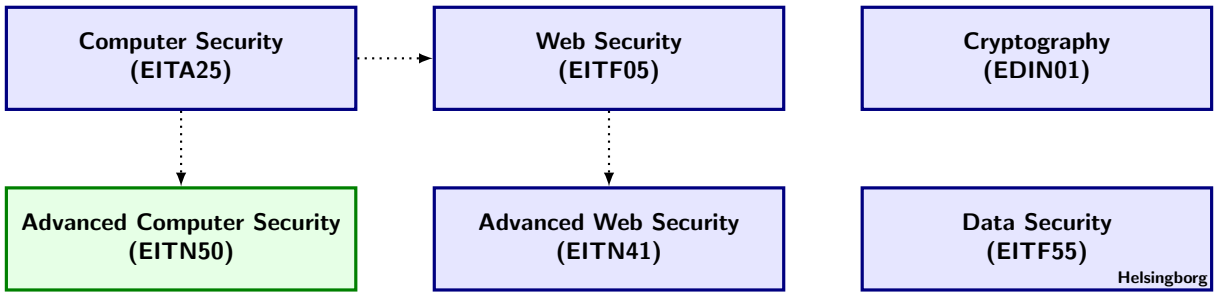
Advanced Computer Security 2021

Department of Electrical and Information Technology
Lund University

Project A: Data Forensics

Learning goals:

- Introduce the steps of data forensics analysis of storage media.
- Analyze FAT type media.
- Get an understanding for how to repair media.



Preparations

- Get and glance through two FAT12 specifications, see references on web page.
- Do and pass the course Quiz Area "Forensics" (not the Project quiz) in Course Canvas (`https://canvas.education.lu.se/courses/13677/quizzes/21523?module_item_id=379592`),
- Read the entire project description (this document) carefully before you begin.

Instructions for Project Examination

The project consists of a number of assignments that guide your work and you should take meticulous notes to later help you pass the project quizzes "Project A Quiz" Part A and partB at `https://canvas.education.lu.se/courses/13677/quizzes/21522?module_item_id=379590`.

For a grade 3, complete Part 1 of the project.

For a grade 4, complete both Part 1 and 2.

1 Introduction

How secure a computer system may be, the fact that you have to cope with attacks from legal users (so called insider attacks) is unfortunately a threat that always will be present. Thus one always needs to be prepared to analyze an attack whether that it is ongoing or (more likely) whether it has occurred in the past. Such an analysis involves the gathering of information about the system(s) that is(are) interacting. This is an important part of Computer Forensics and Data Forensics. This project will illustrate some of the steps of Data Forensics. The project has two parts; part one is on the analysis of an image file from a floppy device, and part two is on the analysis of a USB flash device.

2 Part 1: Evidence of a Crime

In the first part of the project you have to find the evidence of "crime": a firm bought an illegal drug machine "Niagara". You are given a truncated image file `image.dat` of the first 197200 bytes from a recovered floppy memory device. The "evidence" of the illegal purchase is to be extracted from this image.

In a series of small problems we guide you to the final solution and trigger your data forensics skills to find the evidence. Data Forensics is becoming an important discipline in computer engineering. Data Forensics techniques allow investigators to recover data from computers and storage media that are hacked or used by hackers or just used by a person that is subjected to an investigation of unauthorized or illegal computer use. This project, as the other projects in this course, is to be seen as an "appetizer" rather than a regular introduction in the problem area, in this case the area of Data and Computer Forensics. Most important is that you actively work in your group to come through every step. Furthermore, you have to get familiar with situation that it may not be beforehand known how you have to solve your problem and that external information that you collect for getting further is not always consistent or is using the same terminology. Real analysts do not seldom meet situations where they have to tackle not earlier observed obstacles in the analysis that must be overcome. This project tries somehow to create such a situation.

You will be given a virtual machine with everything included to complete the project. The VM image is located at "S:\Courses\eit\EITN50\Project - Forensics". The login credentials for the VM is "root:toor". You should copy the VM to a local folder on the computer (not a network drive). Remember to **clean up** after you are done.

2.1 Hardware Properties of the Memory Device

Consider the image file of the memory. Normally you have either to guess what kind of image file you got or you have to have an understanding of how the memory device works. The `image.dat` file is located in the "/root/Desktop" folder.

> **Assignment 1**
>
> Use a hex editor to open the image file. In appendix A, we present the beginning of what you will see. It looks like we have a FAT12 file system. As a first exercise, you need to analyze and gather information about the FAT system. We have provided an excerpt of useful information to look for, see Table 1. Note that there is more information to be gathered. In order to analyze the image, you may use a hex editor tool to look at the `image.dat` file. You may use the online tool `http://hexed.it` or, the built-in tools `xxd` [1] or `hexeditor` [2].
>
> **Note**: The tools are not guaranteed to give you the answer you are looking for. It is to your advantage to write your own program to extract the data to get a deeper understanding. C or Python are viable options when performing low-level analysis.

Table 1: FAT12 image information excerpt.

Information	Offset (Decimal)	Value
Device name	?	?
Serial number	?	?
Filesystem type	?	?
Media descriptor	?	?
Bytes per sector	?	?
Number of reserved sectors	?	?
Number of sectors per allocation	?	?
Number of sectors per FAT	?	?
Number of sectors per track	?	?
Number of heads on the diskette	?	?
Number of hidden sectors	?	?
Start of bootstrap routine	?	?
Number of FATs	?	?
Boot signature	?	?
Size of the device (bytes)	-	?
Offset to start of FAT(s)	-	?
Root Directory Offset	-	?
Offset to data area	-	?

2.2 FAT Investigation

It seems that the infamous virus, VirFAT, has damaged and corrupted the FAT table on the disk.

> **Assignment 2**
>
> Find out where the virus has corrupted the FAT table and suggest a way to correct it.
>
> **Hint**: Write a program to visualize the FAT table entries to make it easier for you to analyze it. It should be enough to print the first 200 entries.

2.3 Investigation of Directories

The next step in the analysis is to understand what is on the disk and we start by looking at the directories.

> **Assignment 3**
>
> Give detailed information for each directory and file that you can find, using Table 2. Date information should be decoded in human readable form.
>
> **Hint**: Write a program to extract directory and file information. Consult the mandatory readings for details regarding the directory and file formats.

The directory structure has been damaged by another virus, VirROOT.

> **Assignment 4**
>
> Find and analyze the anomalies.
>
> **Hint**: Find all directories and draw a tree structure of the layout.

Table 2: Directory/file information.

Information	Offset (Size)	Value
Directory/File Name	?	?
Attributes	?	?
Creation Time and Date	?	?
Last Access Date	?	?
Time and Date Stamp	?	?
Cluster's Chain in FAT	?	?
Absolute Offset	?	?
Size of the file	?	?

2.4 Attack on a Zip-archive

An experienced analysts easily recognizes at this stage that there is Zip file in the image. It may contain the evidence we are looking for.

Assignment 5

To retrieve the data in the Zip file, you will need to break the password of a Zip-archive file. In practice there are many tools for this purpose, here we suggest to simply use a brute-force attack using John the Ripper [3]. A compiled version of John is located in the "/root/john" folder.

Note that for every command, you must be located in the `john` folder. Also, place the extracted Zip file in the `john` folder as well. Check carefully that the file you compiled is a sane zip file. A common mistake is to have the ending wrong which might give a crc error. If you can click on the zipfile and it will ask for the password when you want to extract the document the zip file is ok to work with. Extract the Zip password hash by running

```
./zip2john <Zip file> > ziphash
```

Then, to crack the password, run

```
./john -i=adsec ziphash
```

To view the password, run

```
./john --show ziphash
```

The output format is

```
zipfile.zip:XXX:::::zipfile.zip
```

where the `XXX` is the password. Make sure to take notes of all the details about the Zip file and the evidence information that you can find.

3 Part 2: USB Flash Analysis

The second part of this project concerns the analysis of a USB Flash memory device. The device, when attached to a PC, shows up as two drives and for each of the drives an image set has been made using a forensic image tool; flash0.E01 and flash1.E01. The files are in a standard form which can be read and fed into different analysis tools. In this project we use the Autopsy forensic browser that you find on the Kali virtual image with various tools for simple computer forensic analysis.

3.1 Preparation

For analyzing the images we use the Kali Linux Suite. It is a Linux distribution with lots of penetration testing tools that can be found at `https://www.kali.org/`. There is also a live-cd than can be transformed into a bootable pendrive. That would be the type of solution you would use when analyzing a complete PC.

You simply start autopsy by entering "`autopsy`" in a terminal. Autopsy runs in a browser on address "`http://localhost:9999`". Use the Firefox browser (there are issues with the Chrome browser). One may easily add the image files into the Kali virtual machine by just drag-and-drop. For convenience, we have already placed three files on the desktop: flash0.E01, flash1.E01 and flash2.E01. Note also that the virtual machine can read from the USB ports of your host machine.

In autopsy, you may try to open the image files as a disk or a partition. The Autopsy engine is here a bit annoying here for our purpose as it does not allow users to simply delete an added image. It is simpler here just to add an new host to which you add the same image in another fashion (say partition instead of disk). Run the analysis so you can conveniently browse the files.

Assignment 6

1. Load one of the image files, e.g. flash0.E01, in the Autopsy system and collect as much information about the USB flash device (e.g. what kind of USB flash device it is), see Figure 1. Describe how this USB flash worked/behaved when it was inserted into a windows PC.
2. List the files that you find on the image, including files that were erased (as much as possible of course).
3. What can you tell about the user of the USB disk? Did he/she left interesting traces?
4. Repeat with the other image file, e.g. flash1.E01.

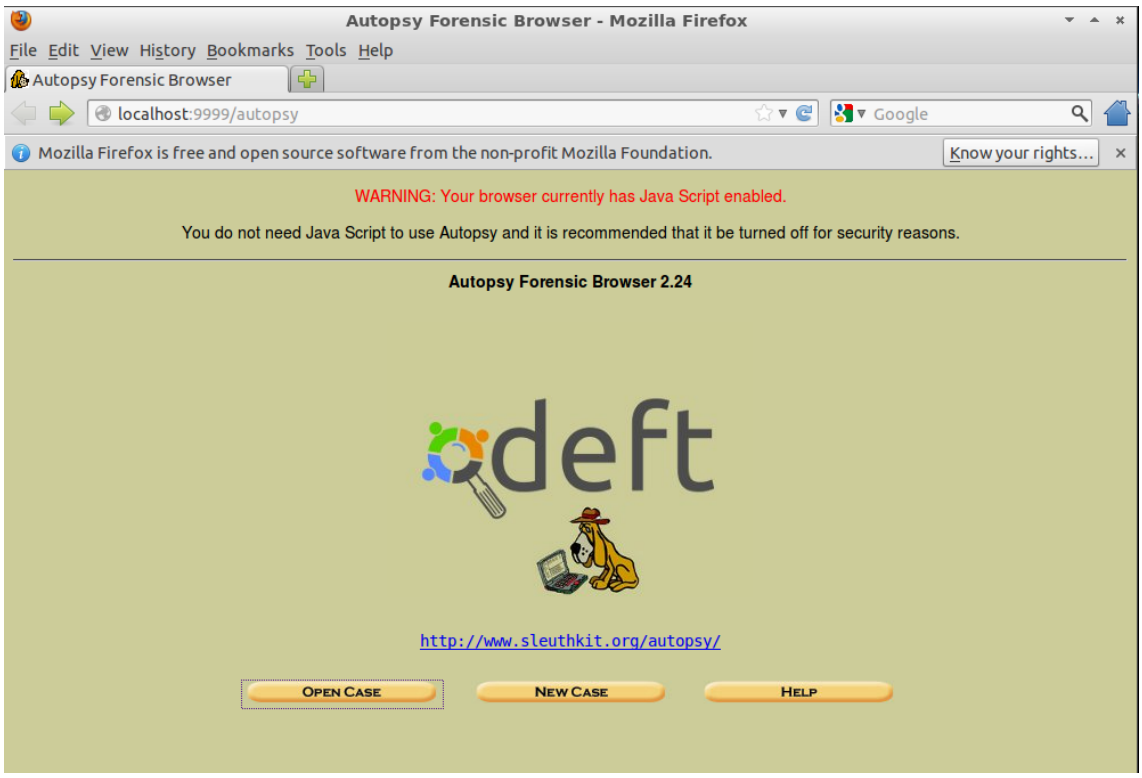


Figure 1: Autopsy from the Kali virtual image

.

Assignment 7

Optional Make an image of each other's USB flash drives or SD cards (if the PC has a reader for it). Take a small one as this may take some time. Determine and document the characteristics of your friend's memory device. Run a file analysis, to what extend you can recover the files? Check the results with your friend.

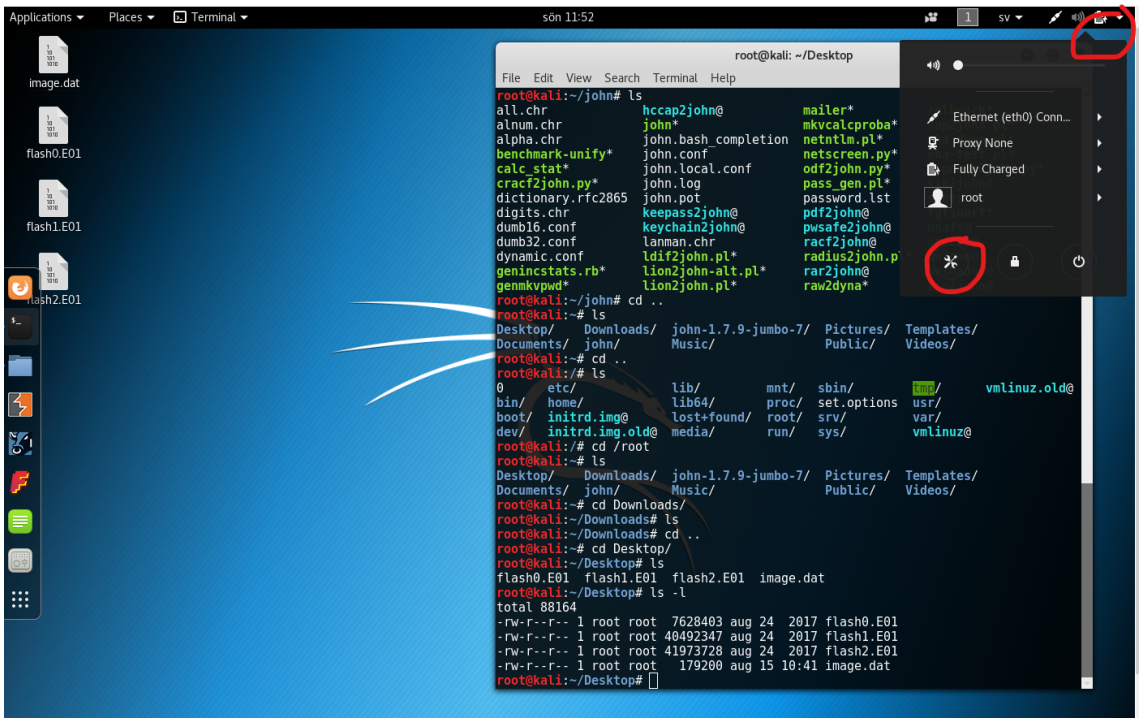


Figure 2: Findings settings for keyboard

.

3.2 Kahli Virtual machine

The Kahli VM is configured for Swedish keyboard but you can change that after you started the image by changing the regional settings and do a reboot, see Figure 2.

References

[1] xxd: https://www.systutorials.com/docs/linux/man/1-xxd/

[2] Hexeditor: http://manpages.ubuntu.com/manpages/zesty/man1/hexeditor.1.html

[3] John: http://www.openwall.com/john/

A Beginning of Image File

Beginning of the image file in hex-mode:

```
00000000  eb 3c 90 4d 53 44 4f 53  35 2e 30 00 02 01 01 00  |.<.MSDOS5.0.....|
00000010  02 e0 00 40 0b f0 09 00  12 00 02 00 00 00 00 00  |...@............|
00000020  00 00 00 00 00 00 29 1b  b1 36 24 4e 4f 20 4e 41  |......)..6$NO NA|
00000030  4d 45 20 20 20 20 46 41  54 31 32 20 20 20 33 c9  |ME    FAT12   3.|
00000040  8e d1 bc f0 7b 8e d9 b8  00 20 8e c0 fc bd 00 7c  |....{.... .....||
00000050  38 4e 24 7d 24 8b c1 99  e8 3c 01 72 1c 83 eb 3a  |8N$}$....<.r...:|
00000060  66 a1 1c 7c 26 66 3b 07  26 8a 57 fc 75 06 80 ca  |f..|&f;.&.W.u...|
00000070  02 88 56 02 80 c3 10 73  eb 33 c9 8a 46 10 98 f7  |..V....s.3..F...|
00000080  66 16 03 46 1c 13 56 1e  03 46 0e 13 d1 8b 76 11  |f..F..V..F....v.|
00000090  60 89 46 fc 89 56 fe b8  20 00 f7 e6 8b 5e 0b 03  |‘.F..V.. ....^..|
000000a0  c3 48 f7 f3 01 46 fc 11  4e fe 61 bf 00 00 e8 e6  |.H...F..N.a.....|
000000b0  00 72 39 26 38 2d 74 17  60 b1 0b be a1 7d f3 a6  |.r9&8-t.‘....}..|
000000c0  61 74 32 4e 74 09 83 c7  20 3b fb 72 e6 eb dc a0  |at2Nt... ;.r....|
000000d0  fb 7d b4 7d 8b f0 ac 98  40 74 0c 48 74 13 b4 0e  |.}.}....@t.Ht...|
000000e0  bb 07 00 cd 10 eb ef a0  fd 7d eb e6 a0 fc 7d eb  |.........}....}.|
000000f0  e1 cd 16 cd 19 26 8b 55  1a 52 b0 01 bb 00 00 e8  |.....&.U.R......|
00000100  3b 00 72 e8 5b 8a 56 24  be 0b 7c 8b fc c7 46 f0  |;.r.[.V$..|...F.|
```