Lab 3:

Windows FS

Forensic Analysis 2023-2024

© Daan Pareit and Hendrik Derre

## Introduction

# Lab Concept

During this lab, we will be using some tools to investigate a suspect’s hard drive. This hard drive will be provided as a virtual disk (vmdk) and you can use your own machine to execute the analysis.

**Your Tasks**

* Examine the drive, what can you tell about the usage?
* Recover all sensitive information that can help our investigation!

# Learning goals

Knowledge

* Structure and content of the FAT file system
* Structure and content of the NTFS file system

Skills

* Use Linux based tools for analysis of the file system
* Restore deleted files

Insight

* Analyze the file system structure

# Practicalities

### Get the images and software

We will use Kali Linux as user environment for analyzing the disk (although any linux distro could do the job). Use the Kali machine from the first lab (which you installed or reused from the ‘Network and System Pentesting’ module).

We will work with an additional Virtual Disk (vmdk) which contains (deleted/hidden) data (“the suspect’s disk”). You can find the vmdk at:

* the samba share: \\nas.ti.howest.be\TI-StudentShare\TI-S4-Forensics or via HTTP at <https://nas.ti.howest.be> in the TI-S4-Forensics folder.
* Alternative download via OneDrive: <https://studenthowest-my.sharepoint.com/:f:/g/personal/daan_pareit_howest_be/ErrJ8sYav4tPjpJW8H3kYBoB_b0u1Yqdo96O1Wv8ajehAg>

On kali, we’ll use tools from the Sleuthkit package. You should already have this installed, as the mmls tool from the previous lab is also part of Sleuthkit. If not, you can find the sleuthkit software here:

<http://sleuthkit.org/sleuthkit/download.php>

*remark: this is preinstalled on kali*

## Lab assignments

### HANDLE WITH CARE – ACQUISITION PHASE

You should treat the virtual disk as a hard disk from a suspect and should thus handle it forensically correct. Therefore, *have a look at the section “Handle with care – Acquisition phase of external disks” of the first lab* on how to make a forensic image and create one from the suspect’s disk (i.e. from the extra vmdk we provided you with).

A reminder of the steps you’ll have to take

1. Stop/mask the udisks2 service (to prevent automounting)

Open 2 terminals – udevadm monitor + udisksctl monitor

systemctl stop udisks2.service

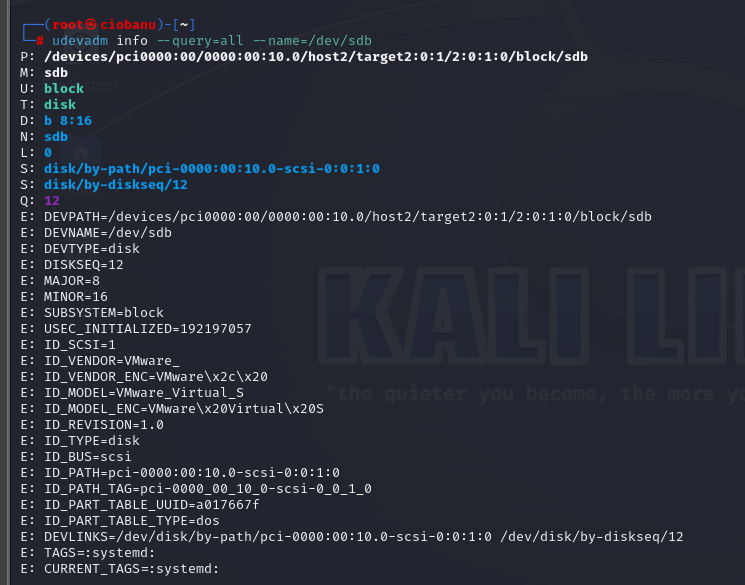
systemctl mask udisks2.service

1. Physically add the disk (the extra vmdk) to your Kali (adding scsi disks is hot swappable)
2. Rescan the SCSI bus to discover the new disk

echo "- - -" > /sys/class/scsi\_host/host2/scan (as root)

1. (optional: read disk metadata with udevadm/hdparm)

udevadm info --query=all --name=/dev/sdb



1. Create an MD5 hash of the suspect’s disk with md5sum.

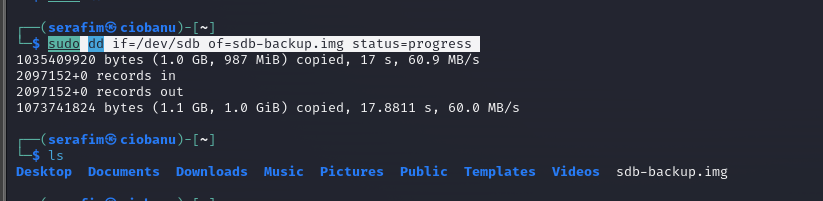
sudo md5sum /dev/sdb

22c2d2ab21a26813ba6479d4874bdcb5 /dev/sdb



1. Make a backup image of *the full disk* using dd (e.g. sdb-backup.img).

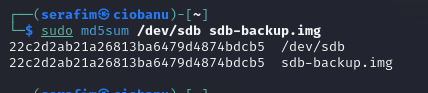
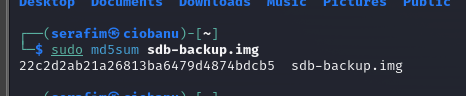
sudo dd if=/dev/sdb of=sdb-backup.img status=progress



1. Create an MD5 hash of your backup. Verify that both suspect’s disk and the backup are the same.

sudo md5sum sdb-backup.img

22c2d2ab21a26813ba6479d4874bdcb5 sdb-backup.img



sudo md5sum /dev/sdb sdb-backup.img

1. Disconnect the disk in linux from your device list

echo 1 > /sys/block/sdb/device/delete (as root)

1. Physically detach the disk (the extra vmdk) from your Kali
2. (optional: restart udisks2 service)

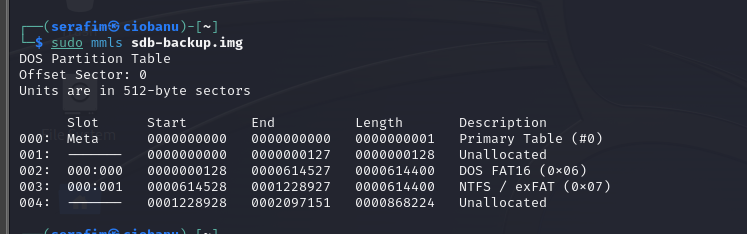
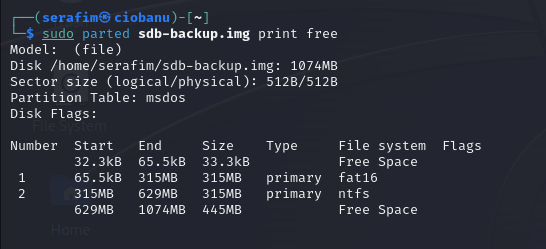
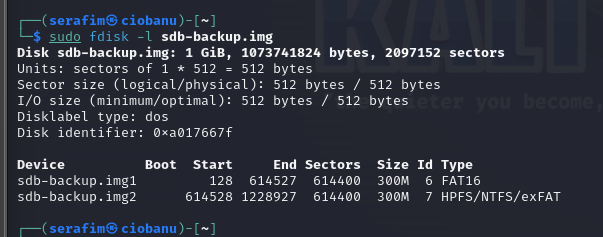
sudo systemctl unmask udisks2.service

sudo systemctl restart udisks2.service

You should not mount the disk itself, because you could alter its data. For the next sections you will always need to perform your investigation on your forensic backup. ***You thus shouldn’t use /dev/sdb in any of the following steps (as it should be disconnected with the steps above) but use your image file instead!***

# FULL DISK ANALYSIS

* Have a look at your backup image and list all the active partitions therein using fdisk, parted and/or mmls (see previous lab)



* + How many active partitions can you find?

2 partitions

* + What file systems are they using?

FAT16 and NTFS

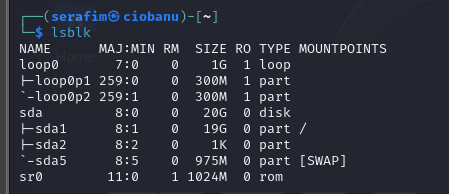
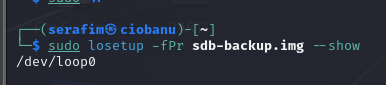
* + Any other useful information from these tools?

It still has some space free

* We now want to inspect these individual partitions. You can further create backup files for each partition, using dd, **or** you can mount your disk backup file as a “loop device”.

Option 1 (preferred): Mount disk backup as loop device using losetup

losetup -fPr sdb-backup.img --show



Your backupfile can now be read as block device at e.g. /dev/loop0 . The partitions therein can be accessed as e.g. /dev/loop0p1 , /dev/loop0p2, etc. (Instead of using the actual /dev/sdb , /dev/sdb1, /dev/sdb2 , etc. on the suspect’s hard disk.)

This thus has the advantage that you do not have to create additional large image files for each partition and that you can work with /dev/loop0 in a similar way as if you would work directly on /dev/sdb .

Option 2: Create backup file for each partition using dd

dd if=/dev/sdbX of=sdb-partitionX.img status=progress

# FAT PARTITION ANALYSIS

Goal: recover all useful artifacts from an FAT filesystem. The FAT partition has three flags hidden.

Tools: xxd, sleuthkit (fsstat, fls, icat, mactime)

### analysis of the files currently present

* Have a look at the filesystem details using fsstat on the partition
  + Based on this output: how many files do you expect to find on the disk?

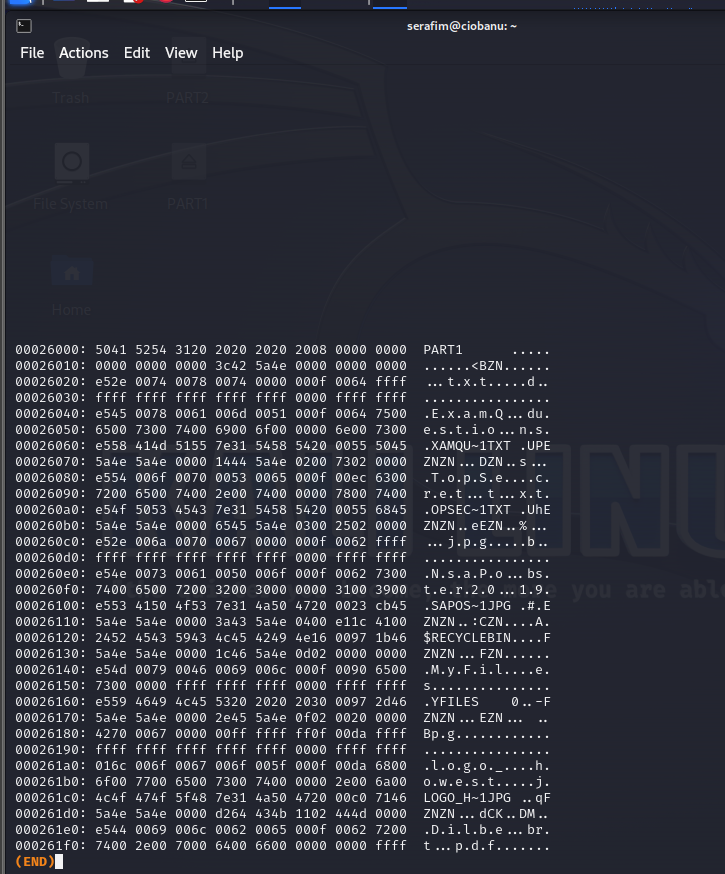
sudo fsstat /dev/loop0p1

sudo fsstat /dev/loop0p2



I suppose there are going to be 5 files.

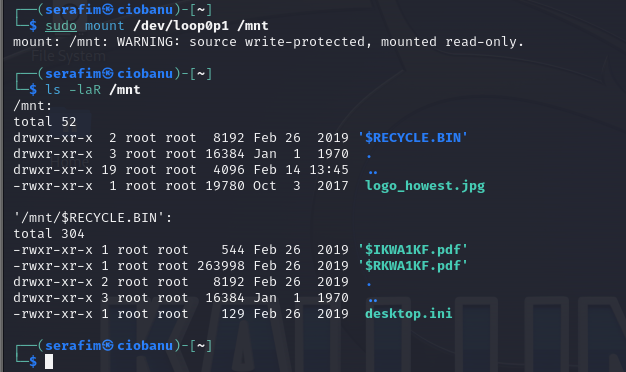
* Look at the FAT in a hex editor, such as xxd. How many files are present?
  + **Hint**: use the data for the offset and length from fsstat output. Remember xxd works with bytes not sectors!



I still believe it is aboit 5 or 6 files. More like 6

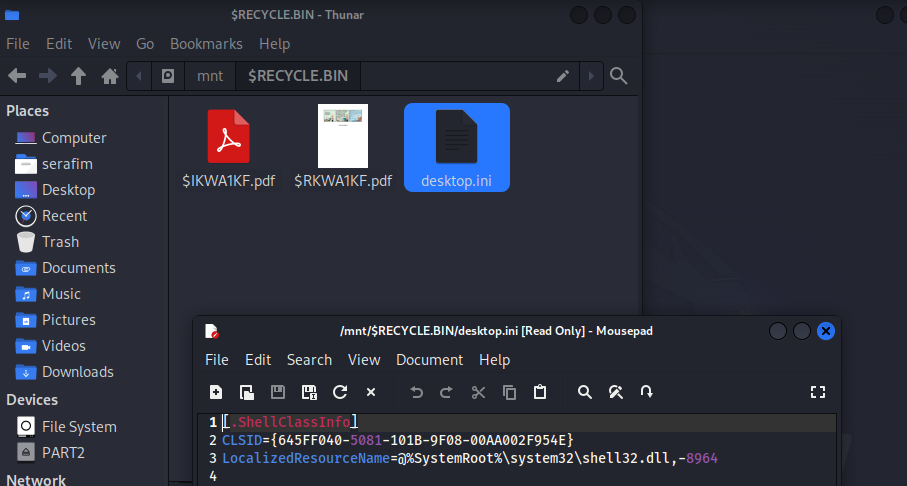
xxd –s offset[Bytes] –l length[Bytes] FILE\_OR\_PARTITION | less

* Mount the FAT partition and have a look at the files still present. Does the number of files match your earlier expectation, based on fsstat output?



Yes, it does

* Can you find any interesting artifacts in the mounted FAT filesystem? Try to explain them.

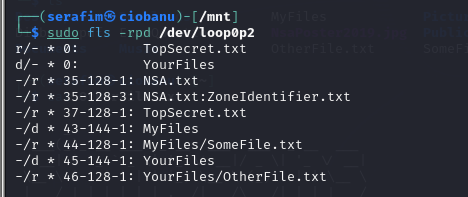
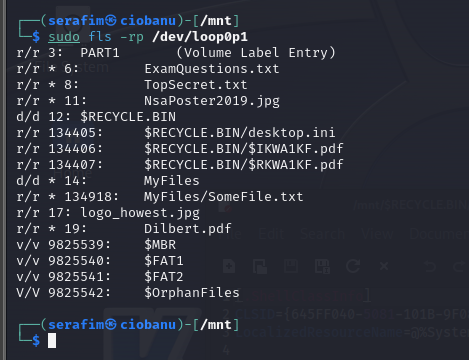


It does contain some files, which can also not be read (probably a separate string)

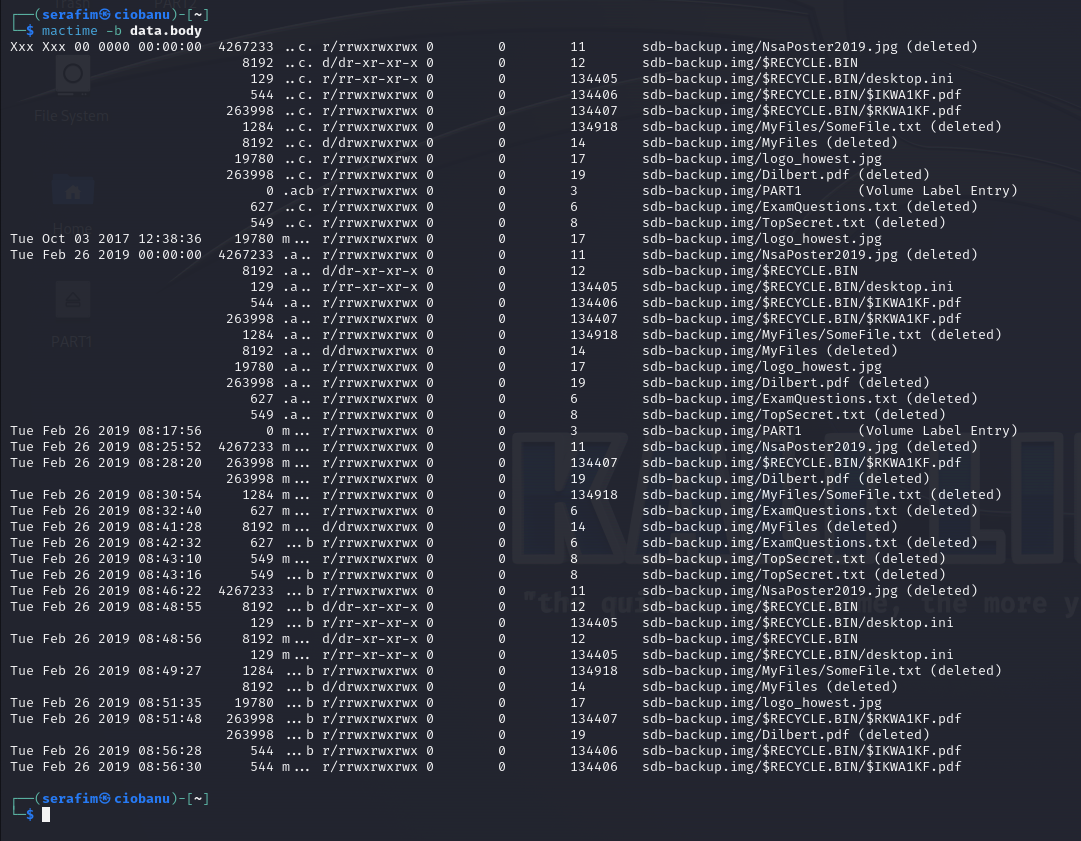
### data recovery of deleted files

* Use fls (look at manpage) to recover other artifacts from the partition

sudo fls -rpd /dev/loop0p1



* + **Extra:** Can you reconstruct a timeline of what happened on the partition? (Hint: mactime)



* Try to restore the content of the deleted files with icat (look at manpage)
  + What files can be restored?

I could not understand whether I need to try and fix the broken files, but I did not manage to do anything fancy with them

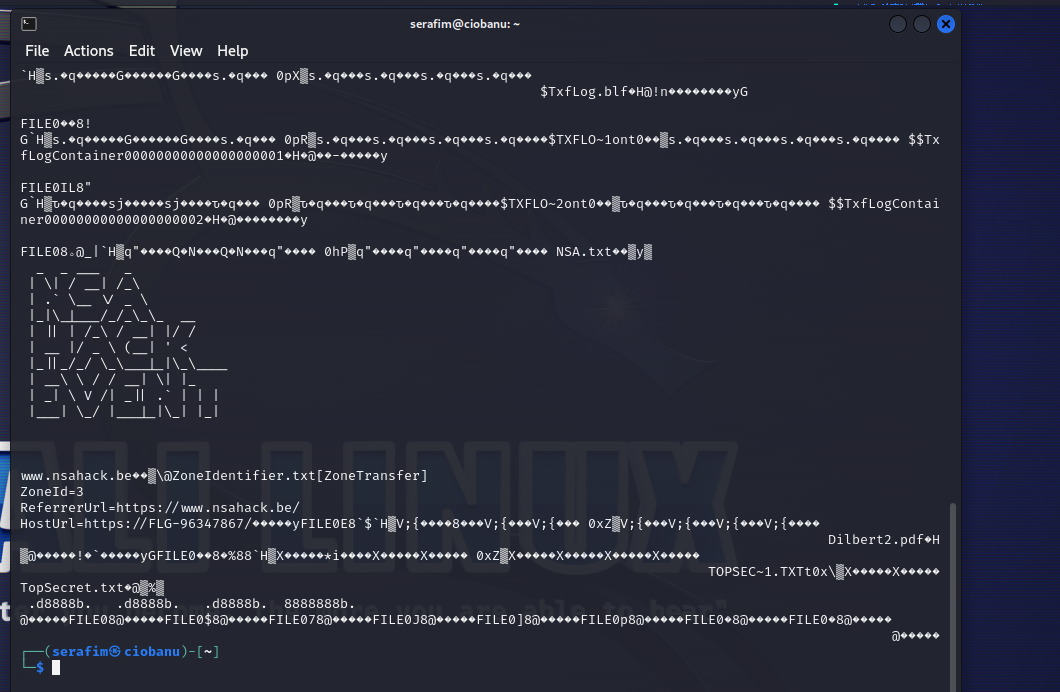
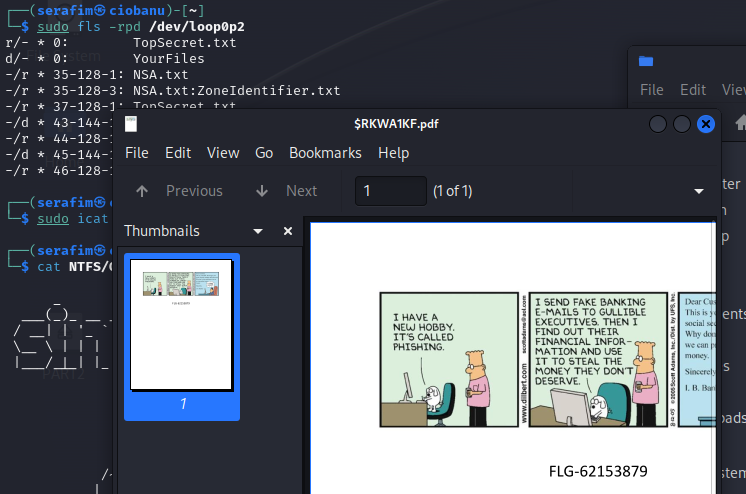
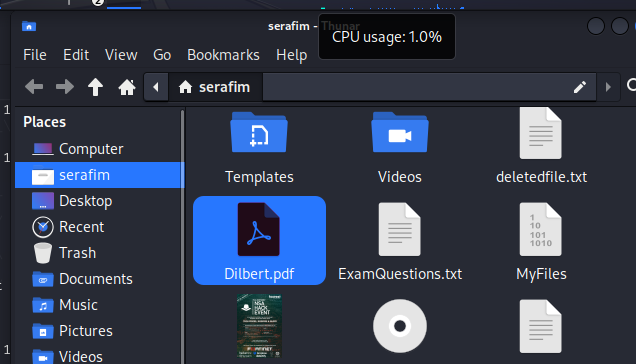
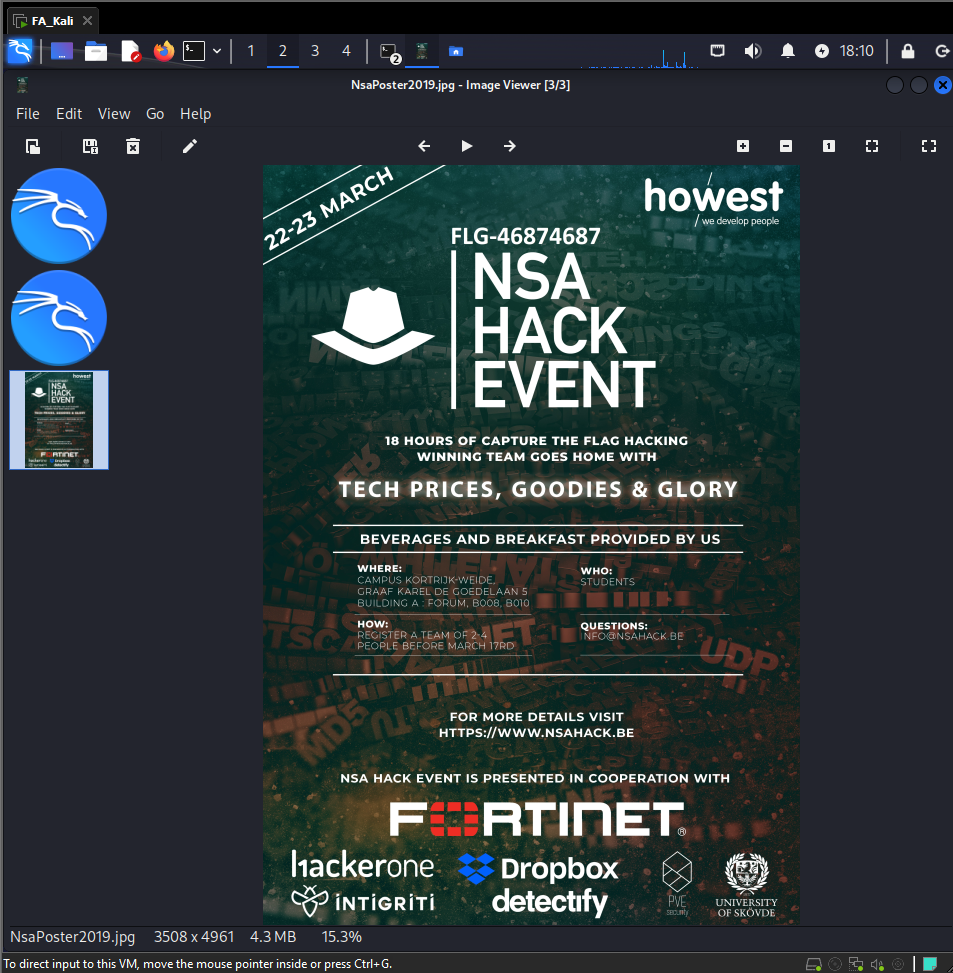
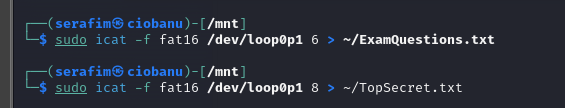
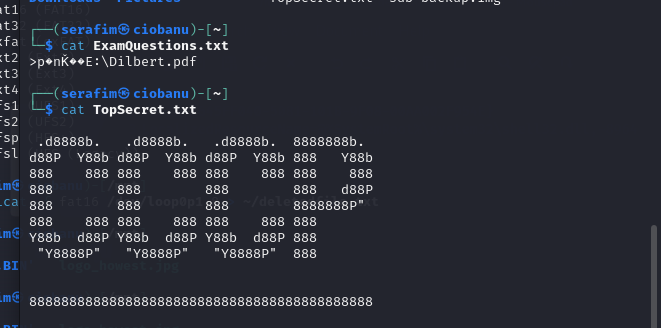
**FLAGS FOUND:**

**FLG-46874687 (NSAPoster), FLG-35168547 (SomeFile.txt), FLG-62153879 ($RKWA1KF.pdf)**

Seemingly, the files which have a \* before them, should be able to res

also can try `sudo fls –rpd /dev/loop0p1`

sudo icat -f fat16 /dev/loop0p1 <start cluster> > ~/<name of file>



* + What files are destroyed?

ExamQuestions.txt, and Dilbert.pdf as far as I can understand, and can not open them, TopSecret (0) from /dev/loop0p1. Another PDF from inside the directory also seems to be broken.

* + **Extra**: Can you tell what was the cause for not bein able to recover certain files?

Maybe because something inside them was altered, or I am confused. I also decided to check some of the files, that being one of the damaged PDF’s, and apparently it did not even have the needed magic bytes. I decided to check it, and then it still would not work, after I changed it

# NTFS PARTITION ANALYSIS

Goal: recover all usefull artifacts from an NTFS filesystem. The NTFS partition has two flags hidden.

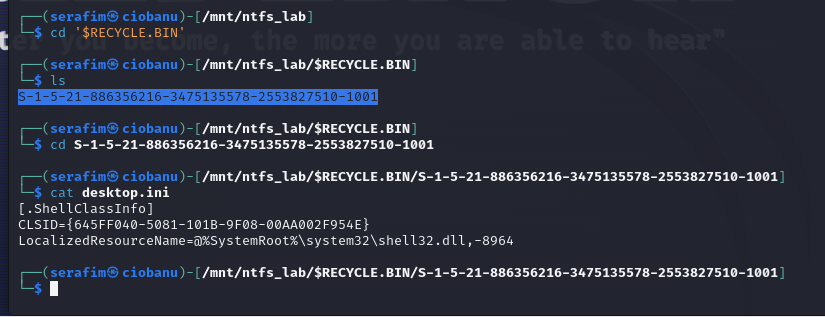
Tools: sleuthkit

### analysis of the files currently present

* Have a look at the filesystem details using fsstat on the partition



* Mount the NTFS partition and have a look at the files still present. Can you find any interesting artifacts? (Note: in case of ntfs-3g-mount error, use the -o loop option.) Try to explain them.



One of them looks like something specific from a user, because the structure of S-1-5 etc looks like one of the similar things from Windows realm. Probably it is the SID I recall.

### data recovery of deleted files

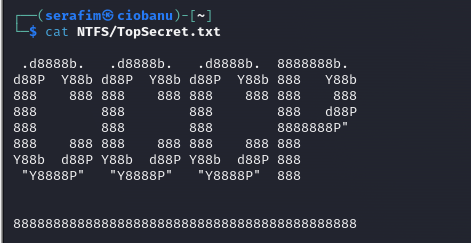
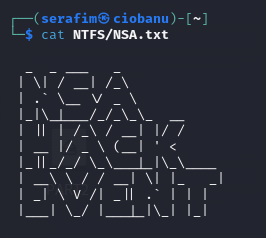
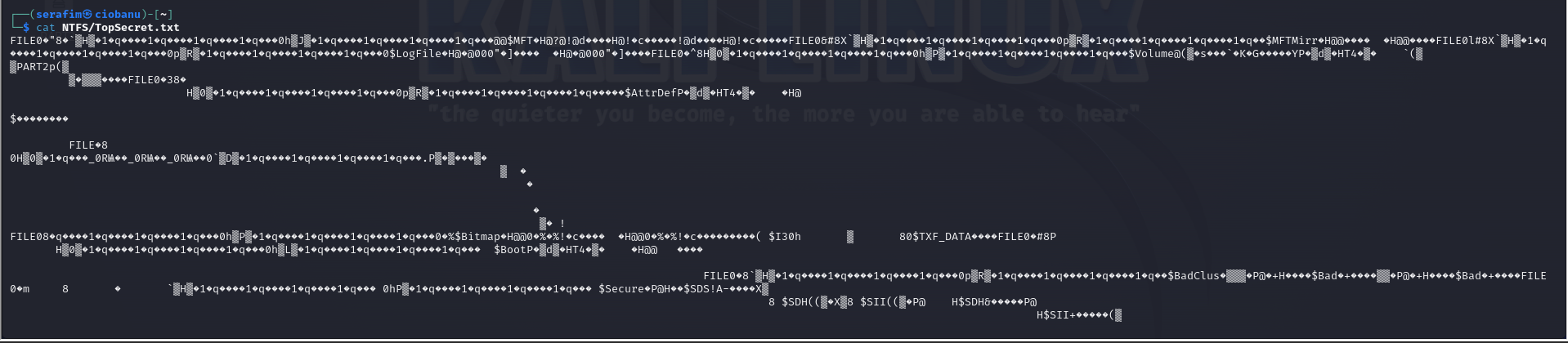
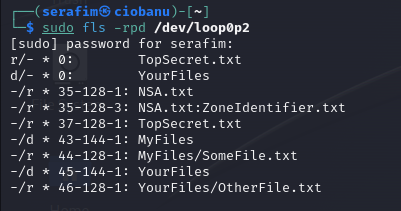
* Use fls (look at manpage) to recover other artifacts from the partition

FLAGS FOUND:

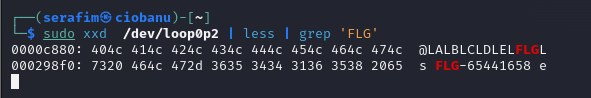
FLG-96347867(NSA:ZoneIdentifier.txt), FLG-65441658 (xxd – grep ‘FLG’), FLG-68691246 (recovery of 3rd partition)

sudo icat -f ntfs -r /dev/loop0p2 0 > ~/NTFS/TopSecret.txt

sudo fls -rpd /dev/loop0p2



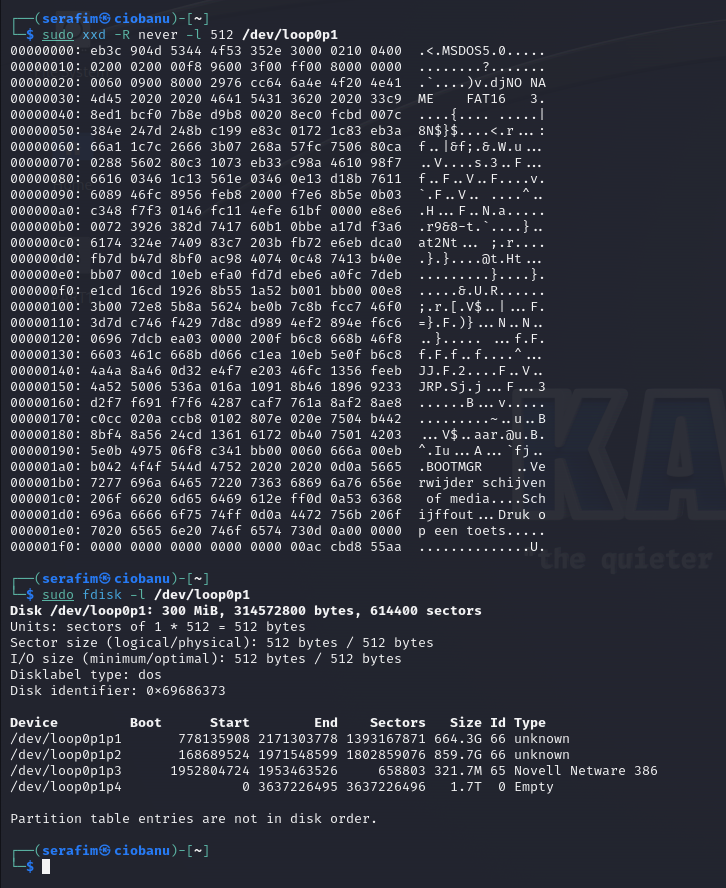
* Try to restore the content of the deleted files with icat (look at manpage)
  + What files can be restored?
* **Extra**: A file ‘LorumIpsum.txt’ was at one time present on this partition and contained a flag. We know it is partly overwritten but the flag remains…How could we find it?



This is cancer. I manually scrolled through the xxd of the partition /dev/loop0p2, to find only the starting text of it. And then, I decided to do a grep. and it worked...

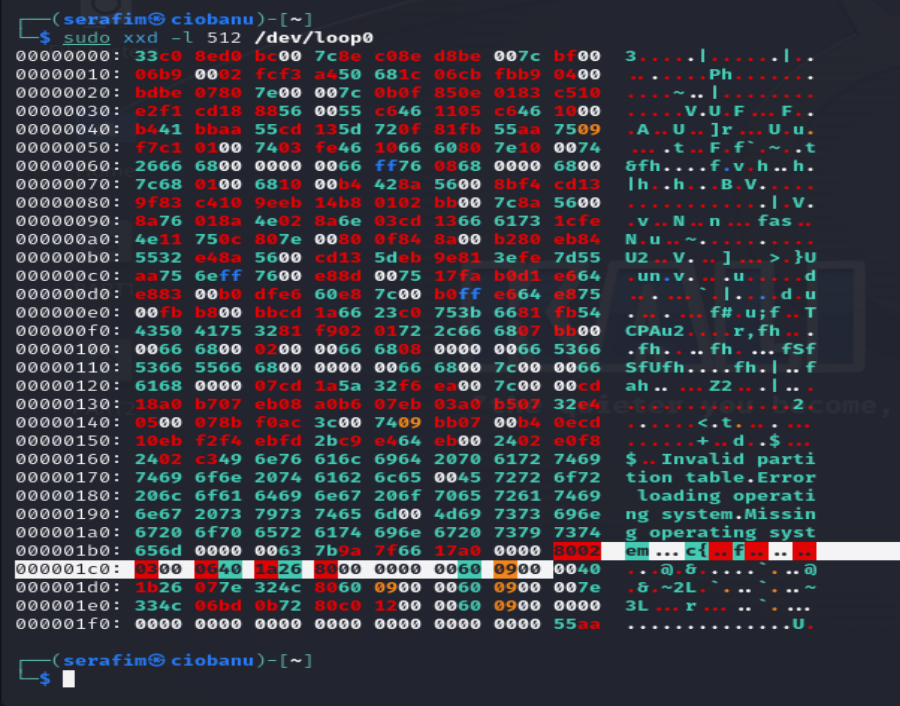
# BONUS

Goal: There is one more Flag hidden in the hard drive… Hint: Examine the partition table closely…. Remember the lab about partitioning…



This is getting the top cancer move. I had to colaborate with other in order to understand a way to find the last flag. The hint to remember the partition lab was not the handiest one.

So apparently, what you are supposed to do, is first of all – take a look at the partition table of **/dev/loop0**



After that, it should ring you a bell about MBR partitions, which have the signature (55aa), and then you have 4 partitions of 16 bytes prior to it. Here we can see 3 out of 4. But when trying to set up the sdb-backup.img as it was, we can only get 2 partitions. This should ring a bell about one of the last parts, which is the recovery of partitions, exactly from lab with partitions.

Next step is to recover that partition, so what you are going to do is run `sudo testdisk sdb-backup.img` (that is the only way I found, but better to also make a backup of backup and test it like that). Once you are there, click **Proceed > Intel > Analyze > Quick search > [Enter] > Write > [Type Y > then Enter], and then you will get changes written to the backup img.**

After a reboot, you will be able to again set up the backup image (sudo losetup -fPr sdb-backup.img --show), and then you can do `lsblk`, and see 3 partitions instead of 2. Now mount the partition (sudo mount /dev/loop0p3 /mnt), and then you can navigate there with Thunar, and open the “EasterEgg.jpg”, and get the flag.

