Lab 1:

physical disk

info and imaging

Forensic Analysis 2023-2024

© Daan Pareit

## Introduction

# Lab concept

During this lab, we will create our forensics workstation as Virtual Machine (VM), learn about disk devices and attach a suspect’s disk safely, create a forensically sound image copy, disconnect the disk and continue to work on the image copy.

# Learning goals

* Using/installing the Kali Linux OS
* Finding block device files which represent disks in Linux
* Obtaining physical disk information via tools such as lsblk, blockdev, lsscsi, udevadm, udisksctl
* Understanding the role of udev and udisks2 in linux for disk detection
* Creating a disk image with dd
* Creating and verify hashes of a disk/image

# Practicalities and prerequisites

You’ll need the following:

* A “Kali Linux” VM
* Suspect’s (virtual) disk
  + scsi-disk.vmdk

On campus or via VPN, you can find the files at the NAS:

* 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.

We will use VMware to work with Virtual Machines. If you don’t have it already, you need to install:

* VMware Workstation Pro (on Windows or Linux)
  + <https://www.vmware.com/be/products/workstation-pro/workstation-pro-evaluation.html>
* or VMware Fusion (on Mac)
  + <https://www.vmware.com/be/products/fusion/fusion-evaluation.html>

You can request a license key for activation via <https://www.academicsoftware.eu/> .

## Lab assignments

# Prepare a Kali VM for forensic analysis

### Create a new Kali VM

You can create your virtual forensics workstation as follows:

* Download the latest Kali ISO file:
  + Go to <https://cdimage.kali.org/current/>
  + Download the kali-linux-<year.number>-live-amd64.iso file
* Create a new VM of type “Other Linux 6.x or later kernel 64-bit” with 20 GB hard disk and 2GB memory.
* Boot it with the downloaded ISO file.
* Choose the “Start installer” option in the boot menu.
* Follow the installation procedure and choose default settings for partitioning etc.
* **IMPORTANT: use your firstname as the username and your lastname as the hostname of the machine. This is to verify your individual assignments for grading!**

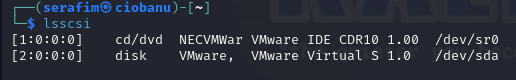
# Analyze your own ‘kali’ disk

### Disks as ‘block devices’ in linux

**/dev/sda** is the representation of our first SCSI hard disk (a block device) in Linux. A second, third, … disk will be represented by **/dev/sdb , /dev/sdc** , …

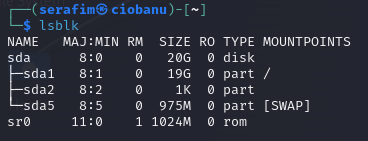
* Note 1: suffix numbers to /dev/sda, such as /dev/sda*1*, /dev/sda*2*, etc are representations of *partitions* on the /dev/sda disk, which we’ll play with during the next lab.
* Note 2: These /dev/sdX files are considered as ‘**block devices**’ in Linux: a continuous series of bits, ordered in blocks/sectors, representing the hard disk. You’ll learn a lot more about Linux’ view on ‘block devices’ later in this course.

Use the lsblk command to view the names of the block device files currently present in your kali machine and check your different SCSI drives with the lsscsi tool. What file represents your hard disk where your kali OS is installed on?



It is installed on the **/dev/sda** file, as the VMware “mark” is there, and it makes sense that it is the main drive that is then becoming partitioned

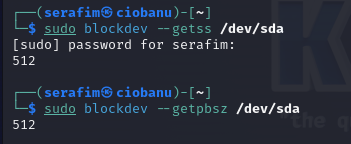
Note that you cannot immediately read/view files from these block device files. Therefore, you’ll need to ‘mount’ the file system on it. This is already done for the files of your kali OS, as shown with ‘mount point’ / in the lsblk output. What block device file represents the partition where the root folder ( / ) is installed on?



As far as I understand the question – it means what is the partition name, where the root (/) is installed on. In this case it is **/dev/sda1**

Use the blockdev tool to find out some physical information of this (virtual) disk. What argument do you need to use to find the physical bytes per sector and what argument for the logical bytes per sector? Knowing this information, is this a 4Kn, 512n or 512e disk type (cfr lecture slides)? Note: to know this for your own laptop disk, you can use the powershell command: get-disk | format-list

.



sudo blockdev —getss /dev/sda (logical)

sudo blockdev —getpbsz /dev/sda (physical)

It is a 512n (512 native because 512 and 512)

### Detecting disks in linux: udev and udisks2 [[1]](#footnote-3)

**udev**: The Linux kernel can send notifications to a user space process (called udevd ) upon detecting a new device on the system. For example, when someone attaches a drive (SCSI, USB, etc). The udevadm program is an administration tool for udevd .

* Use udevadm to get information about what your linux kernel detected about your kali os disk with: udevadm info --query=all --name=/dev/sda . What is the value of ID\_VENDOR?

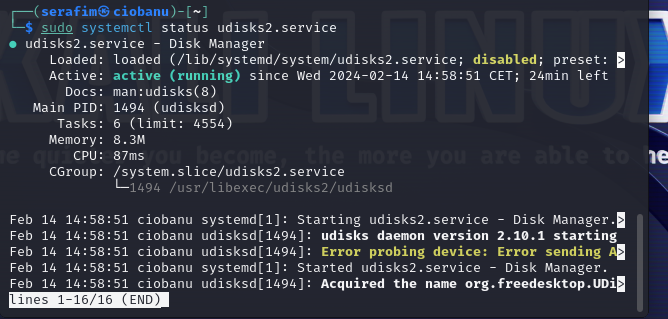


The ID\_VENDOR = VMware\_ (that is probably the same as seeing the information of who made the storage itself, like Samsung or Crucial or etc.)

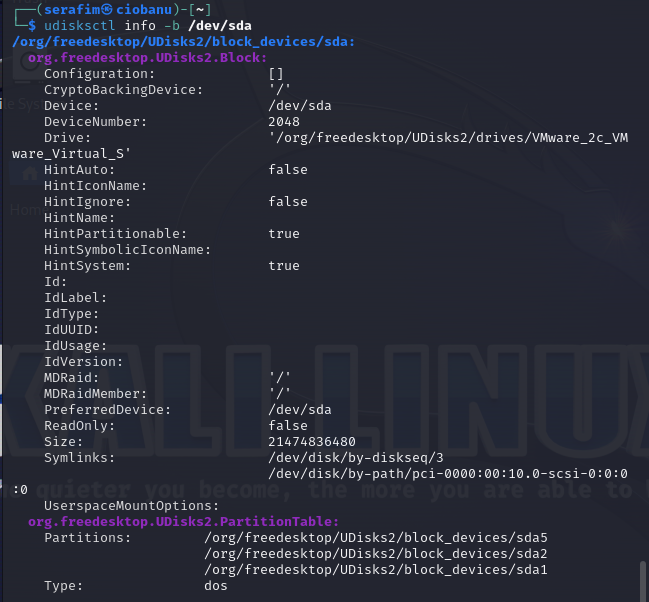
**udisks2**: D-Bus is an interprocess communication mechanism that allows linux desktop applications to talk to each other. Most Linux systems use it to notify processes of system events, such as inserting a USB drive. The D-Bus system has a daemon called udisksd, which is started by the udisk2.service . This one listens to udevd events to automatically attach disks and to further notify other desktop software that a new disk is now available. The udisksctl program is an administration tool for udisksd .

* Check the status of the udisk2.service with: systemctl status udisks2.service. Is it running?

**Yes**, it is



* Use the udisksctl tool to get D-Bus info for your disk: udisksctl info -b /dev/sda



The flag **-b** stands for block device, and **info** is just there as an option

# Handle with care – Acquisition phase of external disks

You should treat the virtual disk as a hard disk from a suspect and should thus handle it forensically correctly (to avoid tampering artifacts which could be evidence). We want to connect the suspect’s disk (here: the extra vmdk) to our workstation. Carefully follow the steps below.

### Disabling automounting

In reality one would put a hardware ‘write blocker’ in between the suspect device and the forensics workstation. If you don’t, you need to make sure no software tries to access/mount the disk automatically to avoid writing to it.

Now, during all next steps, keep two extra terminals open to monitor the output of udevd and udisksd:

* One terminal with: udevadm monitor
* and another terminal with: udisksctl monitor

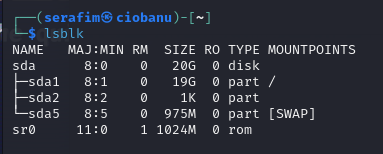
We want to stop the udisks2.service from listening to udev changes.

* Stop the service via: systemctl stop udisks2.service
* Next, prevent the service from being started via any other process:  
   systemctl mask udisks2.service

### Attaching the disk

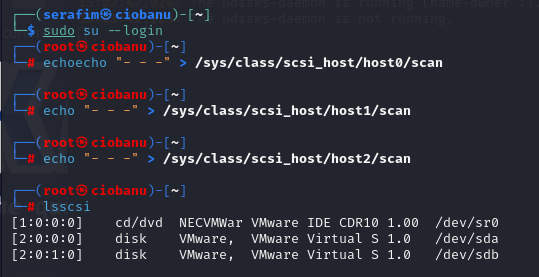
Now, we’ll add the suspect’s vmdk disk to your Kali (adding scsi disks is hot swappable).

* In reality you’ll physically attach the disk to your workstation. Here we attach it virtually by adding the vmdk to your Kali VM via the “Settings” of your VM in “VMware workstation/Fusion”.
* Is it immediately detected by your Kali Linux OS? Check with the lsblk tool

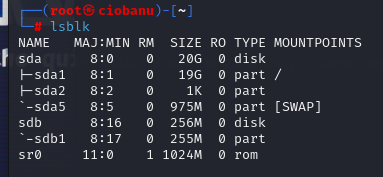


Apparently, there is nothing mounted on the system

* Rescan the SCSI bus to discover the new disk. (The “- - -“ tells the SCSI host to rescan all controllers, targets and LUNs.) Note: this might not work with sudo. You’ll need to switch a terminal shell to root (su --login):  
  echo "- - -" > /sys/class/scsi\_host/host0/scan  
  echo "- - -" > /sys/class/scsi\_host/host1/scan  
  echo "- - -" > /sys/class/scsi\_host/host2/scan



I believe it should be the host2 scan, as the **[ ] values** represent the H:C:T:L and the H is for host, hence I may consider it is the **echo "- - -" > /sys/class/scsi\_host/host2/scan command that worked**



* Which of the lines above made it work for you? Or to ask the question in another way: on what SCSI host (0,1 or 2) was the new disk added? You can check this with the lsscsi tool. (Check the man page for the meaning of the 4 digits in brackets.)

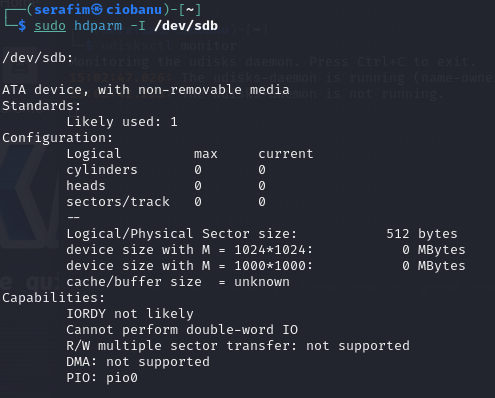
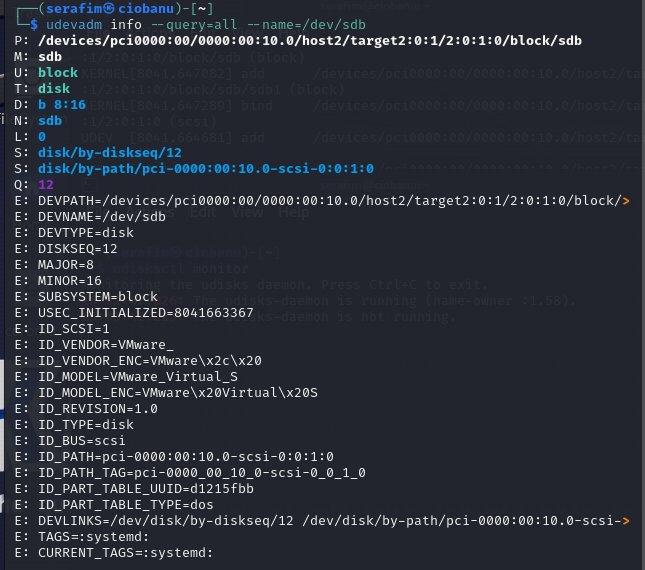
I believe it should be the host2 scan, as the **[ ] values** represent the H:C:T:L and the H is for host, hence I may consider it is the **echo "- - -" > /sys/class/scsi\_host/host2/scan command that worked**

### Reading disk metadata

Now you can query the firmware of the disk about its metainformation which is not part of the actual disk data.

These are different useful tools/commands, but they might not give you a lot of information on vmdks, as there’s no real firmware there.

* udevadm info --query=all --name=/dev/sdb
* sudo /lib/udev/ata\_id --export /dev/sdb
* hdparm -I /dev/sdb



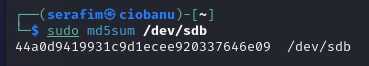
One of the commands did not really work for me for some reason.

### Creating an image backup and hash

We’ll now calculate a hash value of the full disk and create an image copy of the full disk into a single file.

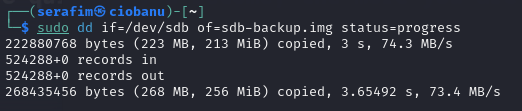
* Create an MD5 hash of the suspect’s disk with md5sum.

sudo md5sum /dev/sdb



44a0d9419931c9d1ecee920337646e09

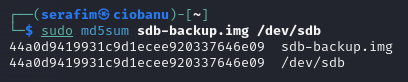
* Make a backup of *the full disk* using dd:  
   dd if=/dev/sdb of=sdb-backup.img status=progress



* Create an MD5 hash of your backup file with md5sum. Verify that both suspect’s disk and the backup are the same. What is the hash value?

md5sum sdb-backup.img

sudo md5sum sdb-backup.img /dev/sdb



I have not easily found a flag (if there is any) that gives output with “Yes” or “No”. But according to the results, it is the same.

### Detaching the disk

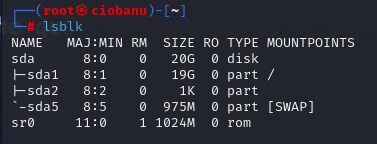
Now that we have a full copy, we should disconnect the suspect’s original disk.

* This can be done this way:  
   echo 1 > /sys/block/sdb/device/delete

sudo su —-login



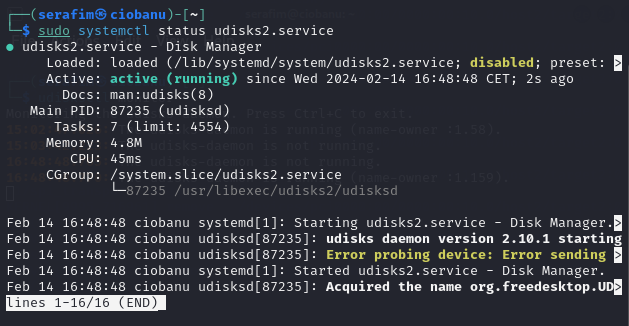
* Verify with lsblk that /dev/sdb isn’t known by your kernel anymore



* You can now physically detach the disk. In our virtual environment: remove the vmdk file in the settings of your VM in VMware Workstation/Fusion
* You can now unmask and start the udisks2.service again via systemctl.

sudo systemctl unmask udisks2.service

sudo systemctl start udisks2.service



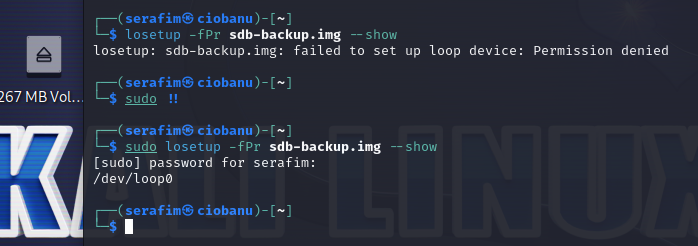
# Inspecting the disk image copy

### Using the image copy as a fake disk

You will need to perform your investigation on your forensic backup.

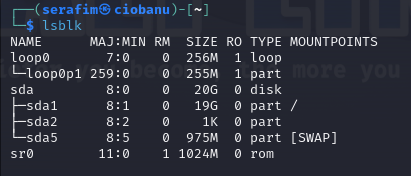
* You can now load the disk image backup as “loop” device using losetup  
  losetup -fPr sdb-backup.img --show

**Take a screenshot of this and paste it here as well as in the Leho quiz.**



Your backup file can now be read as a ‘fake’ block device. Which one?

/dev/loop0 is the disk, but the information is stored on the partition /dev/loop0p1



The partitions therein can also be accessed, thanks to the -P option. What is the file name which represents the first partition?

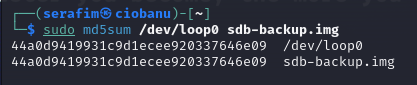
/dev/loop0p1 seems to be the only partition, so it works

This thus has the advantage that you can work with forensic tools in a similar way as if you would work directly on /dev/sdb without risking damaging the artifact/evidence.

* Now take the md5 hash of the loopback device. Is it the same as the original disk?

sudo md5sum /dev/loop0 sdb-backup.img

They are the same (also because we made the loopback device from the backup itself)



### Mounting filesystems

You’ll learn more about partitions and file systems in the next labs. But now that you have taken a disk image and loaded it as a loopback device, let’s explore the disk. Therefore, you need to ‘mount’ the partition on a folder in your linux directory tree in order to access the files.

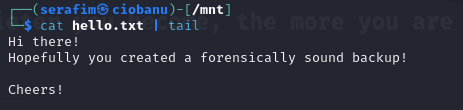
* Mount the file system on the partition of the loopback disk on e.g. the /mnt directory using the mount command.

sudo mount –t ext4 /dev/loop0p1 /mnt



* What file is present on the disk? What is the last word in the file?

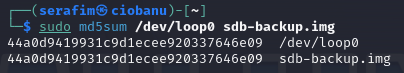
Hello.txt and lost+found directory. The last words should be “Cheers!”



* Unmount the partition again. Is the md5sum hash still the same?

sudo umount /mnt

sudo md5sum /dev/loop0p1 sdb-backup.img

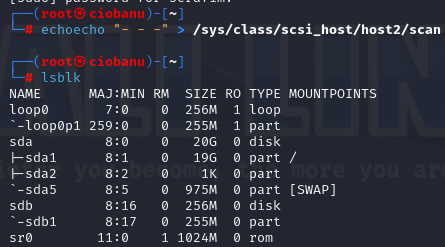


It seems to be the same, since I believe it is based on the intel inside, and we did not touch it, hence it should be logic that it stays intact

### Why mounting the original disk is a bad idea

Now we’re at the end of the lab, let’s do something wrong. Assume you wouldn’t create a backup of the suspect’s disk.

* Attach the suspect disk again.
* Scan the SCSI bus again and verify with lsblk the linux kernel knows about the disk.

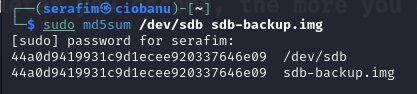


sudo su —login

echo “- - -“ > /sys/class/scsi\_host/host2/scan

* Calculate the md5 hash again (which should not be different than previously)

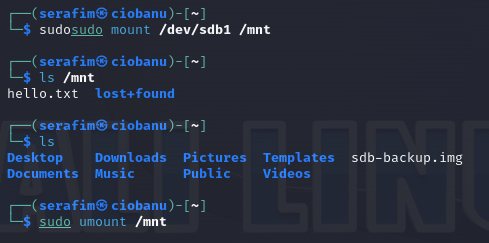
sudo md5sum /dev/sdb sdb-backup.img



* mount and unmount the suspect’s disk, without even looking at the file.

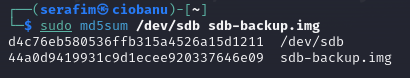
sudo mount /dev /sdb1 /mnt

sudo umount /mnt



* Calculate the md5 hash again. Did it change? You just wasted the artifact and the possible evidence!

It is different, because we did not have the udisks2.service disabled, and it was still working, and hence probably something changed on it. So working with a backup is a better idea.



1. Source and more info: ebook “How Linux Works, 2nd edition” [↑](#footnote-ref-3)