Computer Systems Forensic Analysis AFSC

Obtaining Evidences

Artur Varanda

School Year 2023-2024

- It is normal forensic practice to remove a hard drive from a computer, write-block it and then image that hard drive
- But sometimes that is not possible:
  - ✓ some thin laptops have SSD chips soldered to the motherboard
  - ✓ the storage device has a non standard data interface and the examiner doesn't have the appropriate connector:
    - in these cases the imaging of the storage device needs to be done with the drive connected to the computer;

Use a forensic boot device on the computer:

- boot diskette, bootable CD-ROM/DVD, or bootable USB device

  to ensure the storage drive is not altered either during the boot or the acquisition phase.

The normal startup of a computer alters data on the primary storage drive during the boot process

- it is required to protect the integrity of the original evidence
- we must modify the start-up process in order to prevent any changes to the data on the storage drive

Boot process
- the normal boot process begins within the computer's hardware and moves to the boot device
- there are no changes made until the computer transfers control to the boot device

Boot process steps

- most systems have 2 phases:

  1. configure and start the hardware

  2. find the operating system and run it

- boot code – machine instructions used by the computer

- when it is starting when power is applied to a CPU:

  ✓ it reads instructions from a specific location in memory – typically ROM

  ✓ the instructions in ROM force the system to probe for and configure hardware

  ✓ then searches for a device that may contain additional boot code
    - disks reserve space for boot code, but it isn't always used
    - its boot code is executed, and attempts to locate and load an operating system
    - the process after the bootable disk is found is platform-specific

Boot code characteristics

- has a specific location
- the instructions are in machine code

    0xB400   *// machine code*

    MOV AH,00 *// machine code representation in Assembly*

    on a storage device it is difficult to distinguish random data from machine code

Systems with BIOS (Basic Input/Output System):

- BIOS boots by reading the first sector on a hard disk and executing it
  which has space limitations
- this boot sector code in turn locates and runs additional code in the first sector of the partition
- which locates and loads the actual operating system

UEFI (Unified Extensible Firmware Interface)

- boots by loading EFI program files (with .efi filename extensions)
  - ✓ stored in a special disk partition, known as the EFI System Partition (ESP)
- it can read files from a partition on the hard disk
- it is not limited to the size of the first sector
- it allows booting OS on disks with more than 2TB
  - ✓ regardless of the CPU architecture
- it supports graphics user interfaces on startup
- **secure boot** – is a feature of UEFI
  - ✓ the boot code must be digitally signed to prevent the installation of malware in the boot code
  - ✓ the examiner must use a forensic boot device that supports secure boot
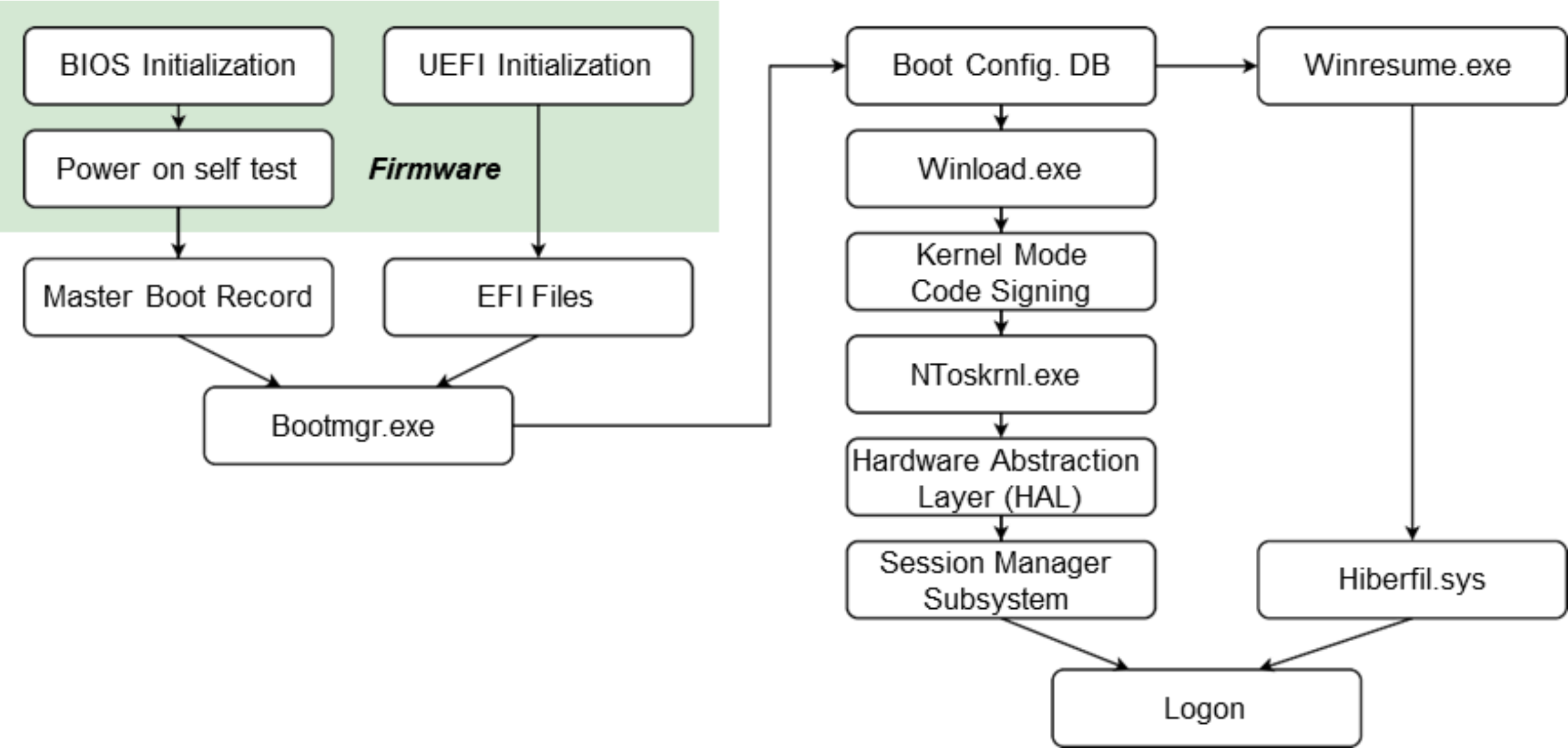
Forensic Boot Tools

DOS boot disk (obsolete, but some times required)

- there are three files required to boot a computer into MS-DOS:

  `O.SYS`, `MSDOS.SYS` and `COMMAND.COM`

- if present are also used in the boot process:

  `DRVSPACE.BIN` or `DBLSPACE.BIN`, `CONFIG.SYS` and `AUTOEXEC.BAT`

How to create a **forensic** bootable diskette:

- on the command line of Windows 98: `format a: /U /S`

  `/U` unconditional format

  `/S` copy the necessary system files over to the diskette, in order to make it a boot disk

- then remove every file from the diskette except the mandatory three (see above)

- remove special attributes from the files to be deleted: `attrib -H -R -S filename`

  later, it is possible to customize the forensic boot disk by adding `CONFIG.SYS` and `AUTOEXEC.BAT` files

  write-blocking utilities and other forensic tools

If you don't have a Windows 98 running

- HP makes an easy to use utility called HP USB Disk Format Tool, which includes a "Create a DOS Startup Disk" option
  - ✓ It's available for free download at http://www.19systems.net/HP-USB-Tool-v2.1.8.exe along with the Windows 98/DOS boot files http://www.19systems.net/Win98-Boot-Files.zip
- once the bootable diskette is created follow the same procedures to make it "forensic":
  - ✓ remove every file from the diskette except the mandatory three `O.SYS, MSDOS.SYS` and `COMMAND.COM` later, it is possible to customize the forensic boot disk by adding `CONFIG.SYS` and `AUTOEXEC.BAT` files write-blocking utilities and other forensic tools

There are many Linux based bootable CD-ROMs (or Live CDs) with forensic tools, such as:

- Paladin (www.sumuri.com) – linux distro with many forensic GUI tools, including Autopsy
- DEFT (Digital Evidence & Forensic Toolkit https://www.deftlinux.it/index.html) is a customized distribution of the Ubuntu live Linux CD
- Caine (Computer Aided INvestigative Environment https://www.caine-live.net) is an **Italian** GNU/Linux live distribution created as a Digital Forensics project (Installation and resolving Stuck Boot-Repair)
- Kali Linux (https://www.kali.org) is an open-source, Debian-based Linux distribution geared towards various information security tasks, such as Penetration Testing, Security Research, Computer Forensics and Reverse Engineering

Create a bootable CD-ROM:

- find and download the `ISO` file, *e. g.* paladin_edge_64.iso
- use a CD-burning program to write the `ISO` file to the CD

Linux based bootable CD-ROM disadvantage

Not all first responders are comfortable using Linux

BOOTABLE CD-ROMS – WINDOWS BASED

There are several Windows based bootable CD-ROMs (or Live CDs):

- WinFE (Windows Forensic Environment) created by Brett Shavers – Free

- System Acquisition Forensic Environment (SAFE) Boot Disk by Forensic Soft – Commercial

- Gargoyle Investigator by Wetstone – Commercial

WinFE (https://www.winfe.net)

Advantages

- it's free, but requires a Windows license
- runs windows software, *e. g.* FTK Imager, RegRipper, . . . can be customized

Disadvantages

- requires configuration on the part of the user prior to use
- must be built to customize with your own set of tools

Nowadays, most computers don't have CD/DVD drives.

Tools to create a bootable USB device:

on Windows

- UNetBootIn (https://unetbootin.github.io/)

- Rufus (https://rufus.ie/)

on Linux

- Gnome Multi-Writer (https://gitlab.gnome.org/GNOME/gnome-multi-writer)

- Etcher – USB and SD Card Writer (https://etcher.download/)

- UnetBootIn for Linux (https://unetbootin.github.io/linux_download.html)

- DD command line tool: `sudo dd bs=4M if=input.iso of=/dev/sdx conv=fdatasync` (replace `sdx` by the drive letter of the USB device)
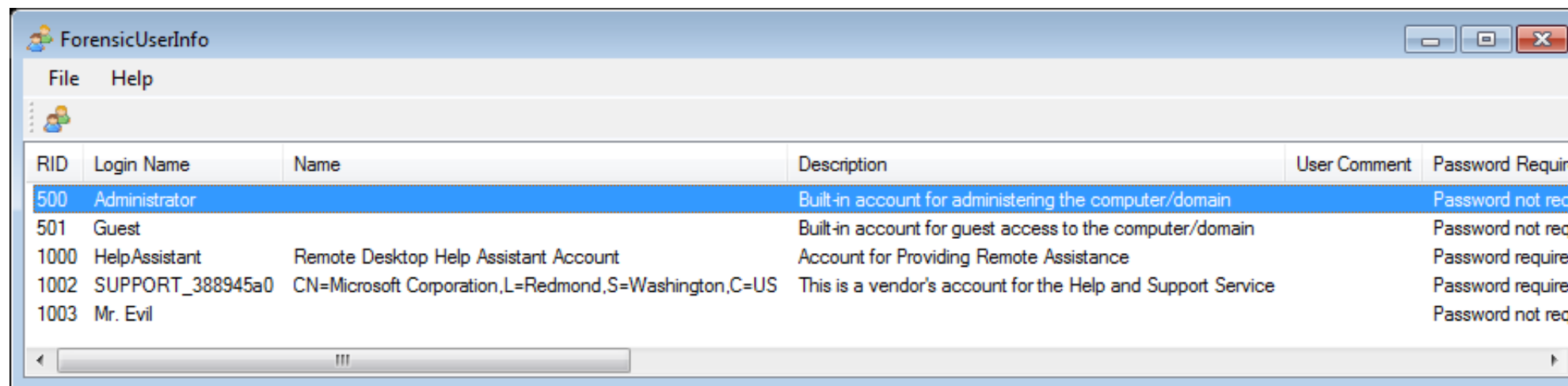
Forensic Sorting Tools

RegistryReport

- homepage: https://www.gaijin.at/en/software/downloads

- requires the `SAM`, `SOFTWARE`, `SYSTEM` and `NTUSER.DAT` registry files

- doesn't process the registry files of the running operating system

- shows information about (Windows 2000 or higher)

    ✓ the operating system

    ✓ installed software

    ✓ the last user activity

    ✓ the user settings

    ✓ and many other details

- the amount of information for each category can be configured in the settings dialog

- it allows to save, print and search the generated report

ForensicUserInfo

- homepage: https://github.com/woanware/ForensicUserInfo

- requires the `SAM`, `SOFTWARE` and `SYSTEM` files

- extracts the following information:
  - ✓ RID, Login Name, Name, Description, User Comment
  - ✓ LM Hash, NT Hash
  - ✓ Last Login Date, Password Reset Date, Account Expiry Date, Login Fail Date
 - ✓ Login Count, Failed Logins, Profile Path, Groups

Mitec WRR (Windows Registry Recovery)

- homepage: https://www.mitec.cz/wrr.html

- for crashed machine, registry configuration, data recovery

- it allows to explore:
  - ✓ File Information
  - ✓ SAM
  - ✓ Security Record Explorer
  - ✓ Windows Installation
  - ✓ Hardware
  - ✓ User Data
  - ✓ Startup Applications
  - ✓ Services and Drivers
  - ✓ Network Configuration
  - ✓ Windows Firewall Settings
  - ✓ Environment,
  - ✓ Shell Folders
  - ✓ Outlook Express,
  - ✓ Raw Data

RanSack

- homepage:

  https://www.mythicsoft.com/agentransack/
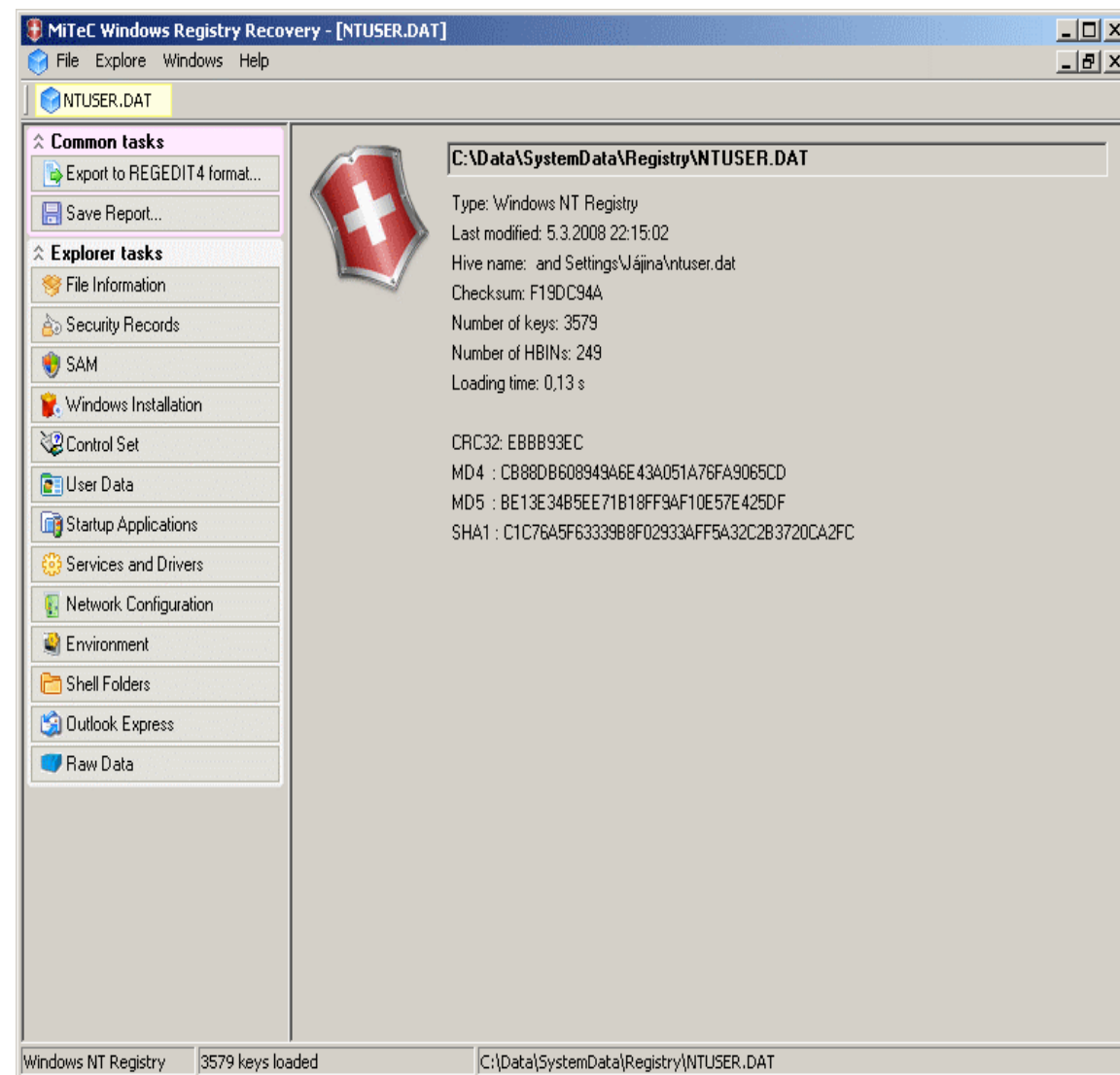
- free software program for finding files on your PC or network drives
  - ✓ fast search (less time waiting)
  - ✓ powerful search capabilities (Boolean expressions, Perl regex)
  - ✓ supports Microsoft Office and Libre Office files formats
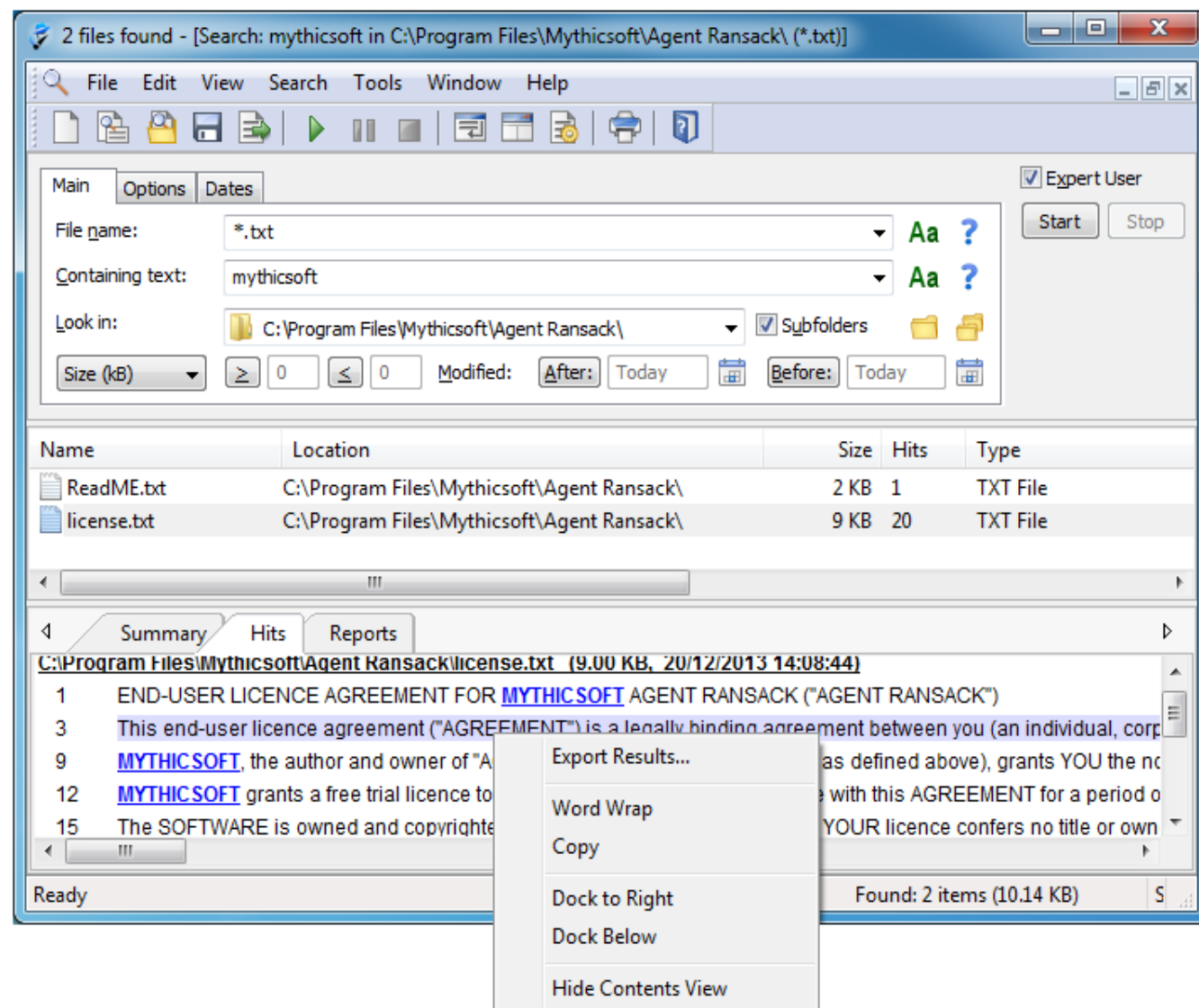
Portable Forensic Tools

- collection of freeware tools, such as:

  - ✓ DataProtectionDecryptor – decrypts passwords of Microsoft Outlook accounts, credentials files of Windows, wireless network keys, passwords in some versions of Internet Explorer, passwords and cookies of Chrome Web browser

  - ✓ JumpListsView – displays the information stored by the 'Jump Lists'

  - ✓ Windows File Analyzer – decodes and analyzes to provide cached information

  - ✓ BinText – extracts strings from binary files

  - ✓ Data Converter – converts numbers, hexadecimal values or dates

  - ✓ EXIF Viewer – displays EXIF informations from JPEG images

  - ✓ eMule MET Viewer – shows various information from the eMule

  - ✓ …

- to find more portable tools: https://www.portablefreeware.com

FTK Imager

- homepage: https://www.exterro.com/ftk-imager

- very powerful and user-friendly tool

  ✓ runs as portable application, ideal to include in WinFE

  ✓ search files

  ✓ look for deleted files

  ✓ copy files (*e. g.* cache and registry files)

  ✓ identify ADS (Alternate Data Stream)

  ✓ acquire storage devices and RAM

  ✓ mount `E01` files

  ✓ …

Forensic Acquisition

Data Acquisition

- typically occurs in the "system preservation" phase

- although it might also occur on a running system

- this is an import phase

  ✓ if not done properly data can be lost forever

  ✓ it must be done in a way that does not undermine its legal validity

What to do if:

- the computer is off → remove power cord

- the computer is on:

    ✓ take a picture of the screen

    ✓ are destructive processes running? → remove power cord

    ✓ do a memory dump and get network connections status → this may destroy or contaminate evidences

        ▪ when you cannot turn off a server

        ▪ to get passwords or encryption keys stored in RAM

        ▪ to monitor malicious software network activities
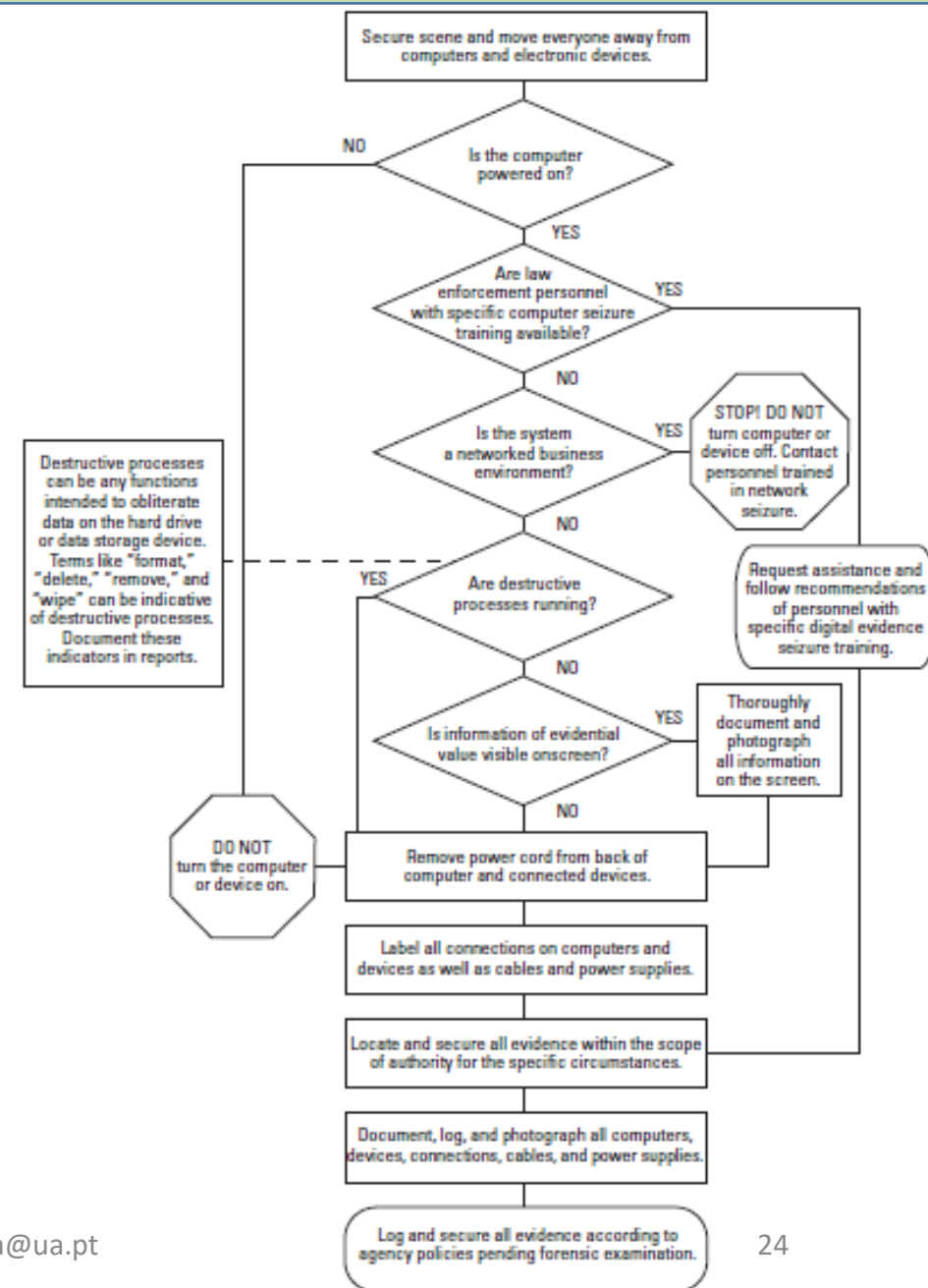
Source: Electronic Crime Scene Investigation: A Guide for First Responders, Second Edition (U.S. Department of Justice)

Flowchart:
- Secure scene and move everyone away from computers and electronic devices.
- Is the computer powered on? NO / YES
- Are law enforcement personnel with specific computer seizure training available? YES / NO
  - YES → STOP! DO NOT turn computer or device off. Contact personnel trained in network seizure.
  - Request assistance and follow recommendations of personnel with specific digital evidence seizure training.
- Is the system a networked business environment? YES / NO
- Are destructive processes running? YES / NO
  - Destructive processes can be any functions intended to obliterate data on the hard drive or data storage device. Terms like "format," "delete," "remove," and "wipe" can be indicative of destructive processes. Document these indicators in reports.
- Is information of evidential value visible onscreen? YES / NO
  - YES → Thoroughly document and photograph all information on the screen.
- DO NOT turn the computer or device on.
- Remove power cord from back of computer and connected devices.
- Label all connections on computers and devices as well as cables and power supplies.
- Locate and secure all evidence within the scope of authority for the specific circumstances.
- Document, log, and photograph all computers, devices, connections, cables, and power supplies.
- Log and secure all evidence according to agency policies pending forensic examination.

Information analysis layers on storage media:

- physical – from the first to the last bit of the storage media

- volume – it is not possible to get unallocated sectors, partition table or hidden areas

- file – file copies (*e. g.* backup tools) less likely to retrieve deleted files

- application – each application has its own encoding or file format

The higher the acquisition layer, the less information can be retrieved

Whenever possible, data acquisition should be made at the physical level.

Other media:

- network and volatile memory

- each medium as its own recommend procedures

Copying storage media

- the bigger the block size, the faster the acquisition,

- but if there are sectors with errors, the all block will be invalid

- the acquisition block size should match the sector size

  - ✓ for HDD the sector size is 512 bytes

  - ✓ for SSD sector size depends on the brand, model and capacity

- data acquisition should include the complete storage medium (physical level)

  - ✓ including unallocated sectors, and

  - ✓ hidden areas: HPA or DCO – in this case 2 acquisitions are recommended

    - ▪ one with the hidden area in place, and another with the hidden areas disabled

Data acquisitions from storage media

make a storage medium forensic copy

- requires another storage medium of equal or bigger size, although many tools can create compressed images files
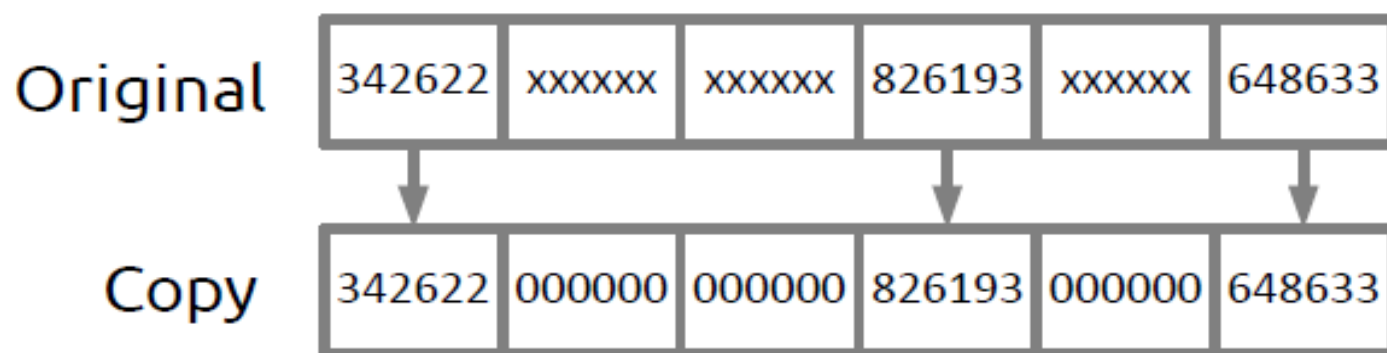
reading the data

- through the BIOS – old BIOS don't support large storage drives → may report wrong drive size
- direct access – the best choice, but not supported by all tools

Post mortem versus alive data

- acquisition post mortem
  - ✓ the OS is shutdown
  - ✓ suspect hardware can be used using a trusted OS to boot it
    - **Caution**: new PCs boot too fast and we might not be able to change boot order
  - ✓ the NSA scandal showed that we cannot always trust the hardware
    - spyware inside HDDs' firmware
  - ✓ although it is less likely to happen than software tampering
- alive
  - ✓ the OS is running and used to perform the acquisition
    - there is the risk of the OS have been tampered and return wrong data
    - *e. g.* rootkits that hide processes and files to avoid detection
  - ✓ online acquisition should be performed only in special situations

What happens if the drive has bad sectors?

- acquisitions is still possible, if the percentage of bad sectors is small

- the tool must be able to deal with the errors:
  - ✓ place zeros on the bad sectors, so the data keeps its alignment
  - ✓ otherwise the forensic copy would be smaller and the analysis tool could trigger errors
  - ✓ the tool must register in a log file all the identified bad sectors
  - ✓ tools should automatically decrease the acquisition block size to the sector size

| Original | 342622 | xxxxxx | xxxxxx | 826193 | xxxxxx | 648633 |

| Copy | 342622 | 000000 | 000000 | 826193 | 000000 | 648633 |

Host Protected Area (HPA)

- this area should be copied also, it may contain hidden data
- few tools support reading HPA
  - ✓ we can use the `hdparm` tool to temporarily access it (a reboot will restore the HPA)
  - ✓ as a precaution measure, we should make first one forensic copy with HPA in place

Device Configuration Overlay (DCO)

- the removal of the DCO is permanent, so as a precaution measure, we should make first one forensic copy with DCO in place
- few tools support reading DCO areas
  - list of tools is available in https://forensics.wiki/dco_and_hpa/ (old site)

Tableau Imager is able to identify and read both HPA and DCO

Write blockers stop any write operation to the storage media under investigation

- Hardware

  - ✓ specific for each medium interface: ATA, SATA, SCSI, Firewire (IEEE 1394) or USB

  - ✓ stops write operations regardless of the used OS

  - ✓ specialized hardware provides better acquisition performance

  - ✓ some hardware works like a proxy and monitors all operations

  - ✓ this is the best option, but it is also the most expensive

  - ✓ list of tests on hardware write blockers:

    - ▪ https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt/cftt-technical/hardware

- Software

  - ✓ this is the less expensive solution

  - ✓ but may be less effective, some apps can bypass the software write block

  - ✓ list of tests on software write blockers:

    - ▪ https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt/cftt-technical/software

There are 2 main approaches to acquire data:

- cloning

  - ✓ it is recommended to use drives of the same size

  - ✓ if the clone driver is bigger, where the cloned data ends?

  - ✓ it is highly recommended to zero out first the drive before cloning

  - ✓ drive geometry of the clone might be different

  - ✓ some OS, namely Windows, by default *auto mount* drives, so you need write blockers to analyze the clones

- imaging the drive – the most common approach

  - ✓ it is not vulnerable to *auto mount* by the OS

  - ✓ an image will always be mounted as read only, no need for a write blocker

  - ✓ it is possible to simulate read/write operation

  - ✓ the changes will be stored in a cache file, leaving the original intact

  - ✓ this way one drive can store image files from several different media

  - ✓ the image file can be split into smaller files to fit in a DVD

There are several image formats for acquired data:

- *raw image* – most flexible format, supported by all analysis software

- *raw image* + external metadata – like the raw image, but adds another file with description data, hash values and time

- *embedded image* – proprietary format, the metadata is embedded inside the image

- some image file formats support compression

  - ✓ save storage space, but the acquisition process takes longer to complete

  - ✓ not all tools support compressed image files – it might be required to uncompress first

  - ✓ good solution for long term storage of the image files

Acquisition

- local acquisition – implies physical access to the storage drive

- remote acquisition

  - ✓ when it is not possible to have physical access to the storage drive

  - ✓ when there are no adequate adapters for the storage medium

  - ✓ this process is slower, usage of compression is recommended

  - ✓ if there are no full control of the network encryption should be used

To guarantee integrity, hash values should be stored

- hash blocks of small size to prevent a single error to invalidate all drive image

    - ✓ *e. g.* the same size of the acquisition block

- in a RAW file; hash values are stored in a separate file

- **It is always recommended to do a digital signature on the hash values files**

    - ✓ Why do you think this is good practice?

Digital signatures

- best tool is GnuPG - https://www.gnupg.org/

- if possible, use also a time stamping server

Tools for data acquisition on storage media there are many tools

- Windows – with graphical user interface

  ✓ FTK imager, Tableau imager, . . .(https://www.exterro.com/ftk-imager)

- Linux – many are command line

  ✓ GUI – Guymager (https://guymager.sourceforge.io/)

  ✓ CMD – ewftools, dd and derived tools: dcfldd, dc3dd, ddrescue, dd_rescue, rdd . . .

Computer Forensic Tool Testing (CFTT) project

- develops test-cases for digital forensic tools

- tests the tools and publishes the results

  https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt

Expert Witness Format (EWF)

- proprietary format from EnCase tool, supports compression

  ✓ file extension: `E01,E02,...`

- there are also [open-source tools](#): `apt-get install ewf-tools`

  ✓ `ewfacquire` – acquire drive data locally

  ✓ `ewfacquirestream` – acquire drive data remotely

  ✓ `ewfmount` – mount EWF images (will be mounted as a RAW file)

  ✓ `ewfexport` – convert image file formats

  ✓ `ewfinfo` – get info from a an EWF image file

  ✓ `ewfrecover` – tries to recover corrupted `EWF` files

  ✓ `ewfverify` – validate EWF file integrity → **very important** before start an analysis

```
sudo apt install ewf-tools        # Install EWF tools
ewfexport -f raw filename.E??     # EnCase → RAW
```

Acquisition example:

```
ewfacquire /dev/sdd                                          # issued command line
ewfacquire 20130416                                          # info generated by the tool

Device information:
Bus type:
Vendor:                                 ATA
Model:                                  VMware Virtual I
Serial:                                 00000000000000000001

Storage media information:
Type:                                   Device
Media type:                             Fixed
Media size:                             53 MB (53477376 bytes)
Bytes per sector:                       512

Acquiry parameters required, please provide the necessary input
Image path and filename without extension: example          # interactive part
Case number: 001
Description: This is just a test
Evidence number: 000001
Examiner name: Miguel Frade                                  # who did the acquisition
Notes: Visrtual disc drive
(...)
Compression level (none, empty-block, fast, best) [none]: best  # compress option
(...)
```

Example

- download `EWF` files:

  http://downloads.digitalcorpora.org/corpora/drives/nps-2008-m57-jean/nps-2008-jean.E01

  http://downloads.digitalcorpora.org/corpora/drives/nps-2008-m57-jean/nps-2008-jean.E02

- get info: ewfinfo nps-2008-jean.E0*

- mount the EWF image is a 2-step process:

  ✓ mount the EWF to be seen as RAW disk

  ✓ mount the raw disk

- Tools:

  ✓ Windows: `FTK Imager` (very simple)

  ✓ Linux has several options:

    ▪ `ewfmount` or `xmount + mount` → requires `sudo`, but allows to see `MFT` as a file

    ▪ `xmount + udisksctl` → the safest option, can be done by a regular user

Mount EWF image with `ewfmount` on Linux

1. `sudo ewfmount nps-2008-jean.E?? /mnt/raw`

   - maps as a RAW file

   - read only, it is not possible to emulate write operations

2. `sudo mount -t ntfs -o`

`ro,loop,show sys files,streams interface=windows,offset=$[63*512]`

`/mnt/raw/ewf1/mnt/loop/`

   - mounts file system in the RAW image

   - `show sys files` – allows to see NFTS structures as files

   - `streams interface=windows` – allows access to *Alternate Data Streams* (ADS) data

   - `offset=$[63*512]` – beginning of the partition to mount (in bytes)

Mount EWF with `xmount` on Linux (without `sudo`)

1. `xmount --in ewf nps-2008-jean.E?? --out raw --cache cachefile /mnt`

   - maps EWF as a RAW file

   - `xmount` emulates write operations using a cache file

2. `udisksctl loop-setup -f /mnt/nps-2008-jean.dd`

   - creates a loop device for each partition

   - `ls /dev/loop0*` → check how many partitions were identified

   - this command requires the user to belong to the `fuse` and `disk` groups:

     ✓ `sudo usermod -a -G fuse username`

     ✓ `sudo usermod -a -G disk username`

3. `udisksctl mount -b /dev/loop0p1`

   - mount the first partition

dcfldd

- developed by *Department of Defense Computer Forensics Lab* (DCFL)

- forensic tool derivate from the `dd` command line tool

- differences with `dd`

  - ✓ calculates the hash values and supports `md5, sha1, sha256` and `sha512`

  - ✓ can write the identified bad sectors to a separate file

  - ✓ can aggregate bad sectors errors `Had 1,023 'Input/ouput errors' between blocks 17-233'`

  - ✓ checks the hash values

  - ✓ reports the progress

  - ✓ allows to split the image file in smaller files

  - ✓ to ensure reproducibility bad sectors are written with zeros in the image file

- there are many forensic tools derivate from `dd`

  - ✓ `dc3dd` (very similar), `ddrescue, dd_rescue, rdd,...`

To get more info about the `dcfldd` tool

- `dcfldd --version` → installed version

- `dcfldd --help` → list all the supported options

- `man dcfldd` → man page

You can get more info:

https://dcfldd.sourceforge.net

https://forensics.wiki/dcfldd/

complement with info from https://en.wikipedia.org/wiki/Dd_(Unix)

dcfldd `if=/dev/sourcedrive hash=md5,sha256 hashwindow=1G md5log=md5.txt`

`sha256log=sha256.txt hashconv=after bs=512 conv=noerror,sync split=10G`
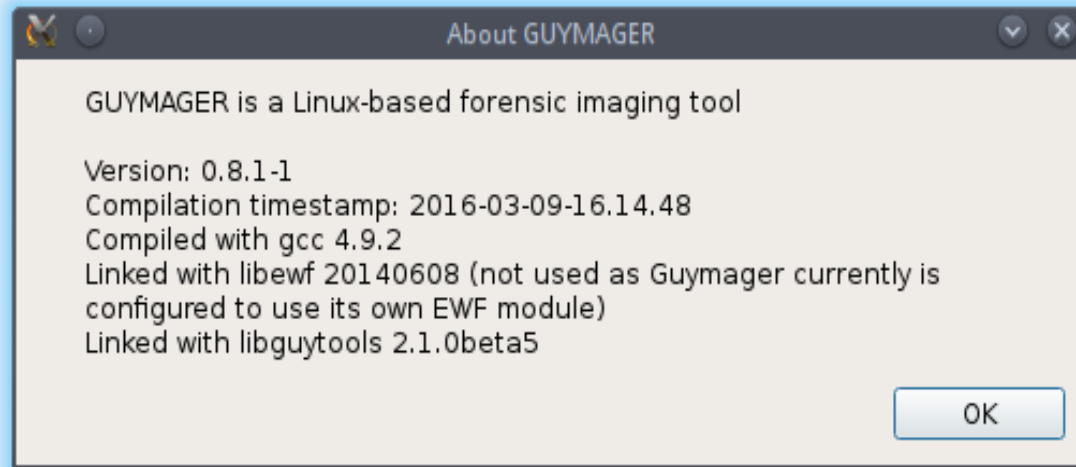
`splitformat=aa  of=image.dd`

- `if=/dev/sourcedrive` → file that represents the drive to acquire
- `hash=md5,sha256` → request `md5` and `sha256` hash values
- `hashwindow=1G` → calculate hash values at each `1 GB`
- `md5log=md5.txt sha256log=sha256.txt` → files name to store hash values
- **`hashconv=after`** → calculate hash values after error checking
- `bs=512` → use block size of 512 bytes
- **`conv=noerror,sync`** → `noerror` – doesn't stop in case of reading errors, `sync` – keeps image synced when errors show up
- `split=10G` → split RAW image into 10 GB files
- `of=image.dd` → base filename
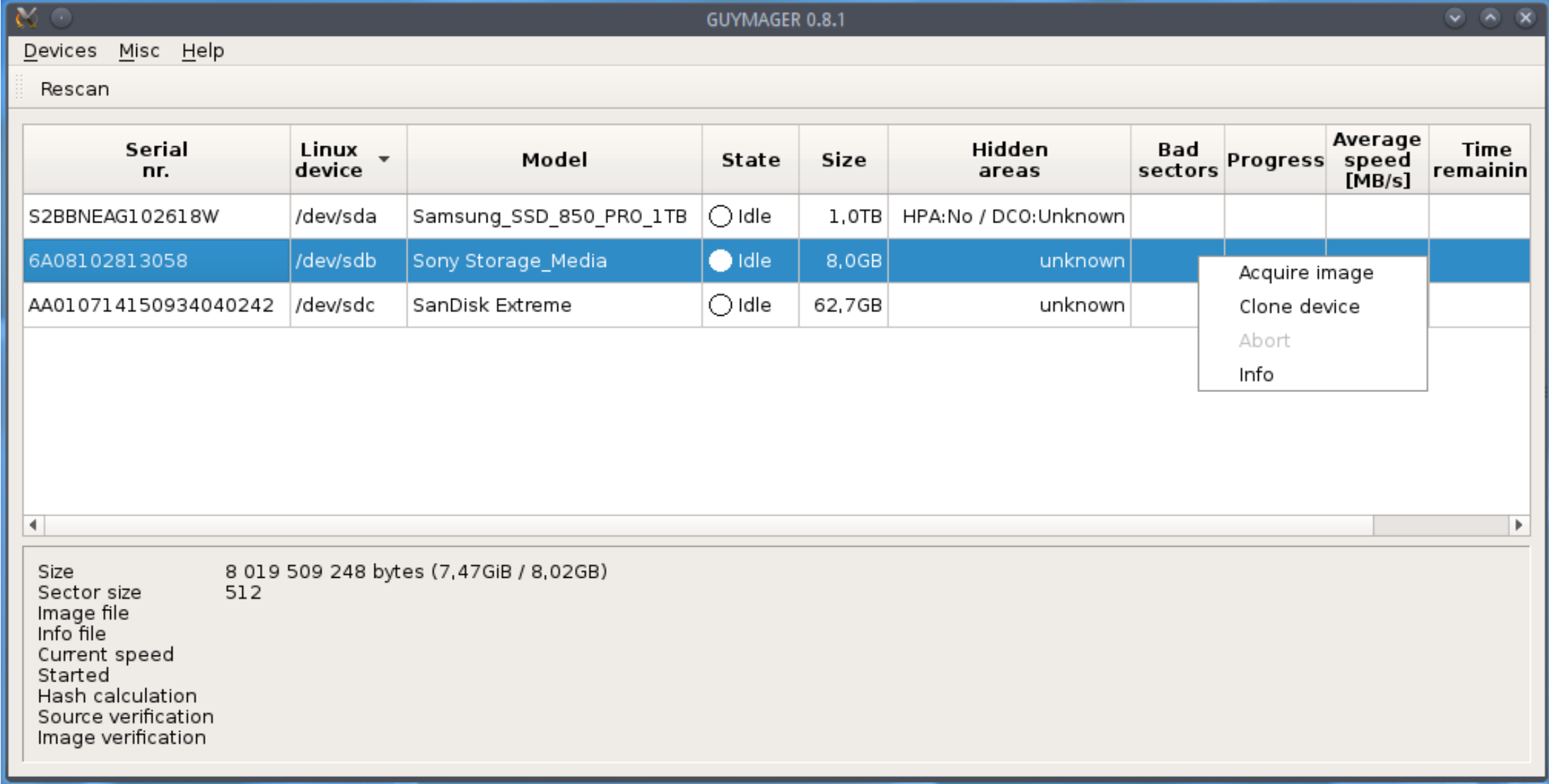- `splitformat=aa` → format of individual filenames `image.dd.aa, image.dd.ab,...`

```
cat SHA256.txt
```

0 - 1048576: 1756ad07245b68744644fa147bbed4dbf5b148ba61839d01e7421ff20098c681

1048576 - 2097152: 908348ee7e44372531d1311143d9ef3a2829fe30f93831b5a81e450a9d366168

2097152 - 3145728: 30e14955ebf1352266dc2ff8067e68104607e750abb9d3b36582b8af909fcb58

3145728 - 4194304: 30e14955ebf1352266dc2ff8067e68104607e750abb9d3b36582b8af909fcb58

4194304 - 5242880: 30e14955ebf1352266dc2ff8067e68104607e750abb9d3b36582b8af909fcb58

5242880 - 6291456: 30e14955ebf1352266dc2ff8067e68104607e750abb9d3b36582b8af909fcb58

...

52428800 - 53477376: 30e14955ebf1352266dc2ff8067e68104607e750abb9d3b36582b8af909fcb58

Total (sha256): 5767d9dcd2e48b3a0dce1b9a143ecf7a664364660637c5f348d1054afb4a1784

Guymager https://guymager.sourceforge.io

- very user friendly

- suppports `RAW`, `EWF` and `AFF` file formats

- faster than known commercial imagers running under Windows.

- does not support logical acquisitions:

  - ✓ `/etc/guymager/guymager.cfg` – default configuration file, do not change!

  - ✓ `/etc/guymager/local.cfg` – do all your configuration in this file, *e. g.* `EwfCompression=BEST`

About GUYMAGER

GUYMAGER is a Linux-based forensic imaging tool

Version: 0.8.1-1
Compilation timestamp: 2016-03-09-16.14.48
Compiled with gcc 4.9.2
Linked with libewf 20140608 (not used as Guymager currently is
configured to use its own EWF module)
Linked with libguytools 2.1.0beta5

OK

GUYMAGER 0.8.1

Devices Misc Help

Rescan

Serial nr.	Linux device	Model	State	Size	Hidden areas	Bad sectors	Progress	Average speed [MB/s]	Time remainin
S2BBNEAG102618W	/dev/sda	Samsung_SSD_850_PRO_1TB	◯ Idle	1,0TB	HPA:No / DCO:Unknown				
6A08102813058	/dev/sdb	Sony Storage_Media	● Idle	8,0GB	unknown				
AA010714150934040242	/dev/sdc	SanDisk Extreme	◯ Idle	62,7GB	unknown				

Acquire image
Clone device
Abort
Info

Size   8 019 509 248 bytes (7,47GiB / 8,02GB)
Sector size   512
Image file
Info file
Current speed
Started
Hash calculation
Source verification
Image verification

Recommended option for Linux

Paladin Toolbox https://sumuri.com/software/paladin

- included in the live Linux Paladin from Sumuri (forensics distro)

- free, but not open source and requires registration to download

- supports logical acquisitions, which is the recommend option for SSDs
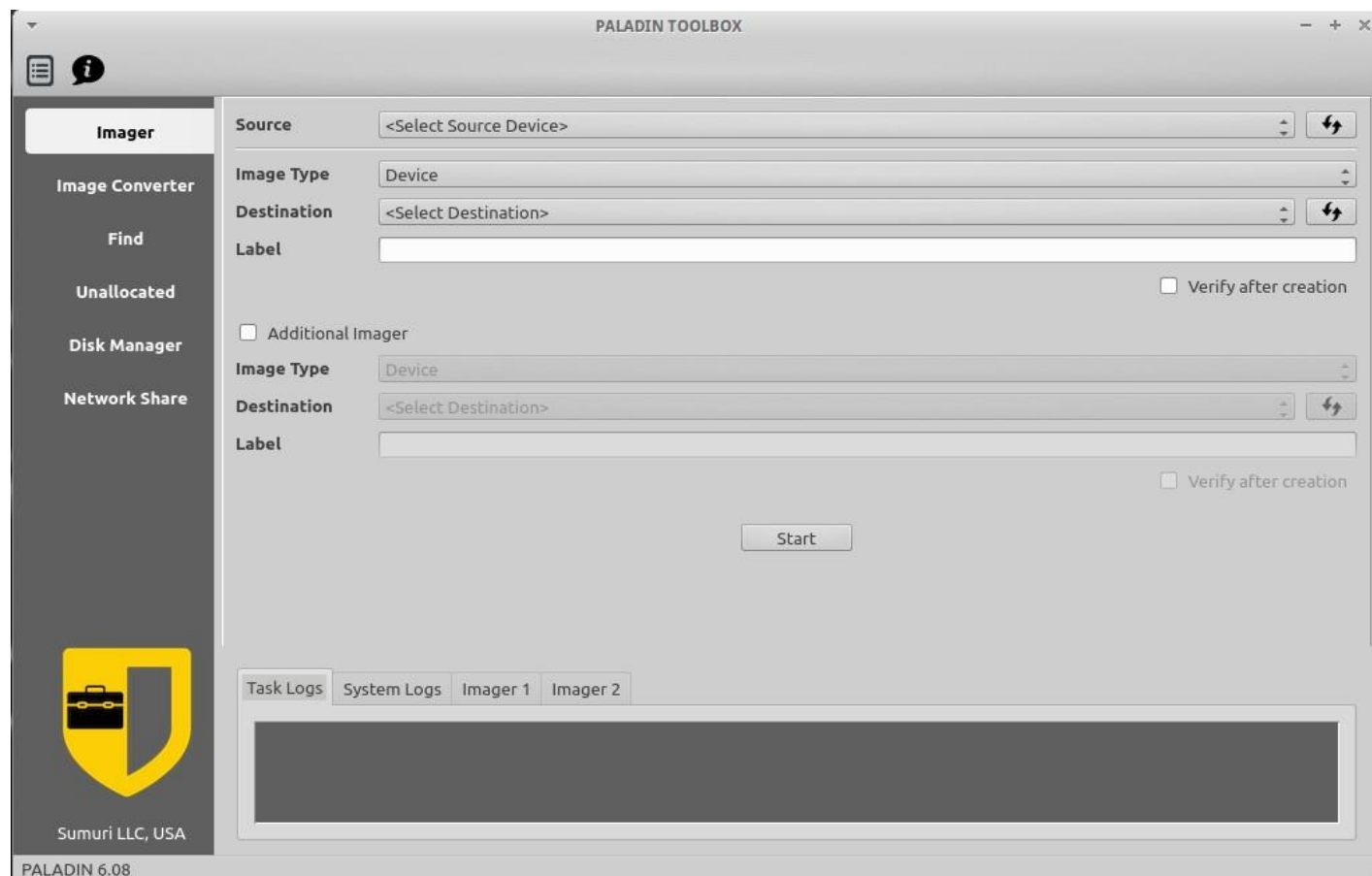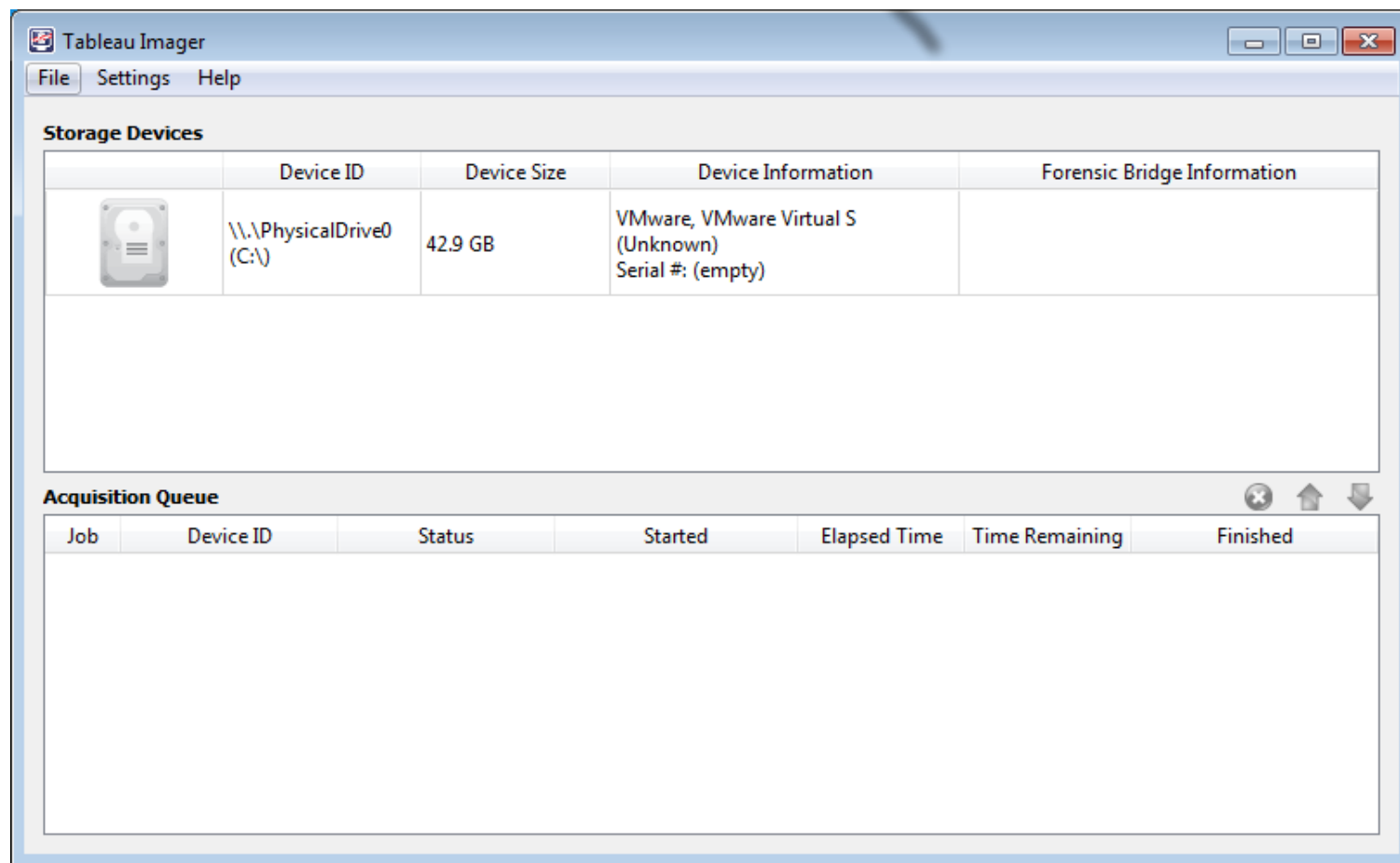
Tableau Imager https://security.opentext.com/tableau/hardware/details/tx1

- works only with Tableau write blockers
- free, but not open source and requires registration to download
- does not support logical acquisitions, which is the recommend option for SSDs

FTK Imager https://accessdata.com/product-download/ftk-imager-version-4-5

- Create forensic images of local hard drives, CDs and DVDs, thumb drives or other USB devices, entire folders, or individual files from various places within the media

- Preview files and folders

- Preview the contents of forensic images

- Mount an image for a read-only view

- Export files and folders from forensic images

- See and recover files that have been deleted from the Recycle Bin

- Create hashes to check the integrity of the files

- Generate hash reports for regular files and disk images

- Lite version runs as portable application

Recommended option for windows
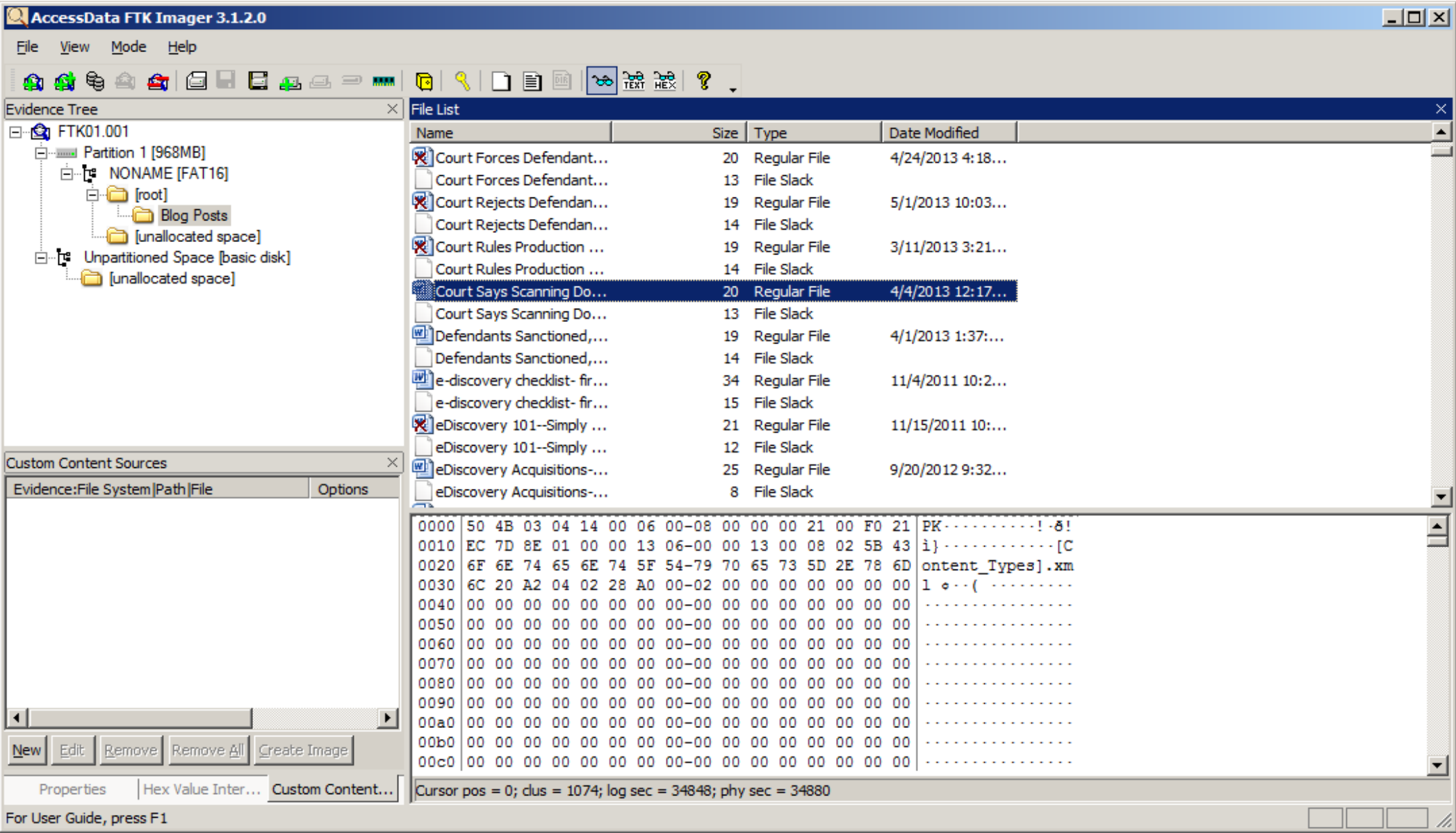
- Lab 01 - Build WinFE and boot a virtual machine with it

Lab 02 – Create a Forensic Image

1. Without extracting the virtual machine assigned to your team, add it as an evidence source to FTK Imager running on your computer

2. Create a .e01 forensic image from the virtual machine with FTK Imager

   - ✓ make sure you have enough disk space for the acquisition
   - ✓ split into 4 096 MB files to fit into FAT32 file systems if needed
   - ✓ enable compression, it's slower, but takes less space
   - ✓ validate the forensic image

• Lab 02 – Create a Forensic Image

3. Sign your forensic imaging acquisition report

As you already know, calculating acquired image hashes using summary functions is always good forensic practice. Don't forget that in addition to guaranteeing the **integrity** of the files, it is also convenient to guarantee their **authenticity** through a digital signature. I suggest you use Gpg4win (Windows) or GnuPG (Linux). They are easy to install and use:

Windows:
https://gpg4win.org/download.html (already contains Kleopatra frontend)

Ubuntu:
https://www.gnupg.org/download/index.html

Frontends:
sudo apt install gpa
sudo apt install kleopatra

Use hkp://keyserver.ubuntu.com server to export your public keys