# Global Information Assurance Certification Paper

# Windows 10 as a Forensic Platform

*GIAC GCFA*

Author: Ferenc Kovacs, ferii.kovacs@gmail.com
Advisor: *Sally Vandeven*

Accepted: *June 4, 2018*

Abstract

Microsoft Windows is widely used by forensic professionals. Windows 10 is the latest version available today. Many popular forensic packages such as FTK, Encase, and Redline are only running on Windows. Other packages such as Python, Volatility, The Sleuth Kit and Autopsy have Windows versions. This paper will detail the process of configuring a Windows 10 computer as a forensics investigation platform. It will show the necessary steps to set up the operating system, install Windows Subsystem for Linux, Python, VMware, and VirtualBox. The research will examine the setup of dd.exe, FTK Imager, Encase Forensic Imager, Redline, The Sleuth Kit, Autopsy, the SANS SIFT workstation, Volatility and Log2Timeline. This research will also highlight the external devices that will be used such as write blockers and external drives. Metrics will be collected to show the effectiveness of the software tools and hardware devices. By following the described steps, the reader will have a configured Windows 10 workstation that provides a useful platform for conducting forensic investigations.

# 1. Introduction

Microsoft Windows is an operating system widely used by home, education, government, and business users as well as forensic professionals. Many popular forensic packages such as FTK, Encase, and Redline are only available on Windows. Other packages such as Python, Volatility, The Sleuth Kit and Autopsy have Windows versions. Virtualization tools such as VMware and VirtualBox are available which allow virtualized computers to run within Windows. The aim of this research is to provide a guide for analysts who want to create a Windows 10 environment for computer forensics. This includes selecting the hardware for the platform, installing the operation system, setting it, choosing the necessary software tools, and installing these tools.

The forensic investigation process includes three main stages: the evidence acquisition, the analysis of the collected evidences, and reporting the results. The majority of the forensic tools provide solutions for the first two stages, as the reporting stage is not tool-dependent but rather depends on the personal skills of the analyst. The investigation process and the required reporting format also depend on the organization. An effective forensic platform provides tools that will allow the analyst to perform each of these tasks in a timely manner. Using the tools included in such a platform, the investigator should be able to collect all of the necessary evidence and analyze it to create the report of investigation (Martin, 2017). An effective forensic investigation platform will comply with the requirements of the forensic analysis tools. The results generated by using the platform should be usable, comprehensive, accurate, deterministic, and verifiable (Carrier, 2003).

This paper will detail the process to configure a Windows 10 computer as a forensics investigation platform. It will show the necessary steps to set up the operating system and install forensic tools on the machine. The tools will be grouped by their functions. The system setup will cover the configuration of Windows and the installation of Python scripting environment. The sections which follow will discuss the configuration of Virtualization software, evidence acquisition tools, and analysis tools. The performance comparison will demonstrate the effectiveness of the software tools and hardware devices based on the collected metrics. By following the described steps, the

forensic analysts will have a configured Windows 10 workstation that provides a useful platform for conducting forensic investigations.

## 2. System Setup

The physical computer that will be used for testing the environment in this study is an HP Z640 mini tower workstation containing 2.4 GHz Intel Xeon CPU with two processors (2x12 virtual CPUs) and 32 GB RAM. The operating system installed is 64-bit x64 Windows 10 Professional. The system is installed on a 250 GB SCSI hard disk. For data storage, there is an Intel Raid 5 SCSI disk array with 5.18 TB capacity.

Windows 10 is the latest version running on desktop computers. Microsoft designed it to be safer and more user-friendly than previous versions. It collects diagnostic and personal- related information to improve user experience and to help Microsoft in troubleshooting problems. Microsoft published their privacy statement for Windows 10 that explains what data is collected and also not collected (Myerson, 2015), (Nadella, 2018). Microsoft collects Safety and Reliability Data (such as device ID, device type, and application crash data) to provide a secure and reliable experience. They collect Personalization Data to provide users with a customized look and feel of Windows. Microsoft aims to deliver targeted advertising, but they will not collect the content of emails, other communications, or files. Microsoft states that the user remains in control to determine what information is collected.

The operating system has to be set to make sure the information acquired for analysis will not leak from the workstation. Keeping the workstation off the network would stop any data leakage but it would also prevent the installation of security updates.

The current installation is a clean install of Windows 10 Professional. Windows 10 installation media can be downloaded from https://www.microsoft.com/en-gb/software-download/windows10. The activation code is not included in the package; it has to be purchased separately.

Microsoft encourages users to connect the machine to their Microsoft Live account that allows synchronizing the settings between various computers and devices

that belong to the user. This workstation will be used to perform forensic analysis. All the tasks intended to run on the machine are related to that function so personalized setup is not needed. The options for customizing privacy settings are summarized in The Complete Guide to Windows 10 Privacy Settings (Phillips, 2016).

During installation, the analyst should choose to use a local account on the computer and should not sign in with a Microsoft account. Cortana is not activated by default. To verify it is disabled click the Cortana icon on the Taskbar and check the status.

It is possible to modify the privacy settings. In order to do so, the analyst should open the Windows menu and navigate to "Settings > Privacy" as shown in Figure 1.

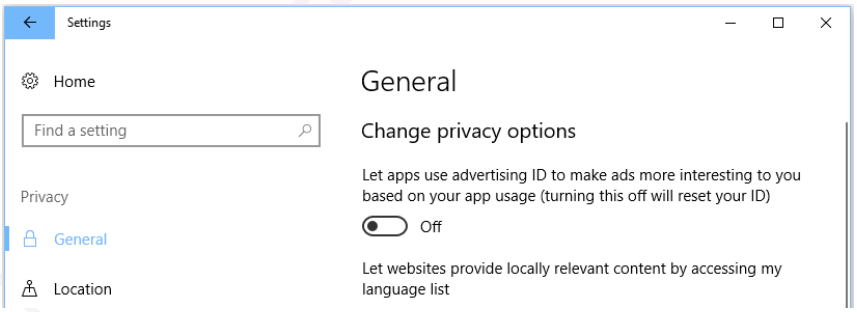



**Figure 1:** Setting Up Privacy Options

To disable the advertising integration, the analyst should navigate to the General Tab and uncheck the radio button "Let apps using Advertising ID". Then he or she should open the browser, go to https://choice.microsoft.com/en-us/opt-out and set "Off" for "Interest-based ads in this browser" as shown in Figure 2. The analyst should do this for each browser that is installed and used on the machine.

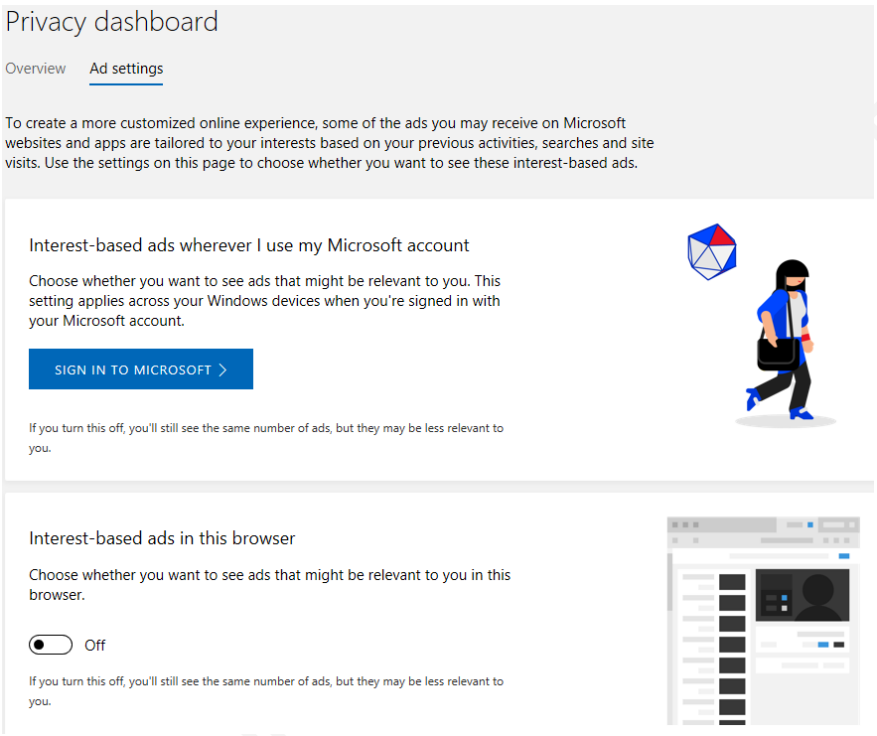



**Figure 2:** Disabling the Advertising Integration

In the menu "Settings > Privacy" uncheck the Location, Camera Microphone, Notifications and Speech services. Switch Wi-Fi Sense off, go to Settings > Network & Internet > Wi-Fi > Manage Wi-Fi Settings. Then slide the option "Connect to suggested open hotspots" to "Off".

Microsoft updates can no longer be turned off. It is possible to pause updates for a maximum of 35 days. It can be set it in "Settings > Update &Security > Advanced options" to pause the updates during daily operation on the machine and activate it manually for idle time. The update cycle will automatically resume after 35 days. Set "Restart options" "Off" to avoid an unwanted reboot of the machine.

## 2.1. Windows Subsystem for Linux

Windows Subsystem for Linux (WSL) is a compatibility layer integrated into Windows 10 that enables Linux binary executables to be run on the Windows machine. WSL provides a Linux-compatible kernel interface developed by Microsoft. The analyst

can install it through PowerShell by opening PowerShell as an Administrator and then running the command: `Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux.` After installation, the computer should be rebooted. Another way of installation is through the Windows GUI. Open "Control Panel > Programs and Features > Turn Windows features on or off" and select the checkbox "Windows Subsystem for Linux" as indicated in Figure 3.

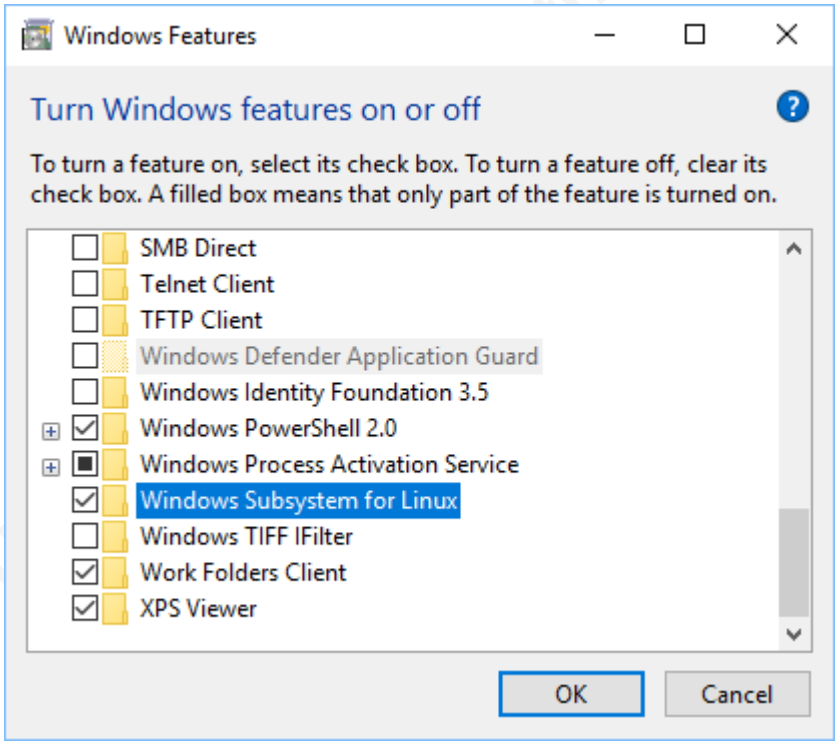



**Figure 3:** Turning on WSL in Programs and Features Menu

Several Linux packages are available in the Microsoft store. Installation from the Microsoft store does not require the linked Microsoft account. The configured system for this research uses the Ubuntu package, which provides a Terminal window and runs Ubuntu command line utilities such as bash, ssh, and apt. The Ubuntu command line is shown in Figure 4. The command line version of SANS SIFT workstation will also be installed under the Ubuntu package. The details of the SIFT setup are described in section 5.3.

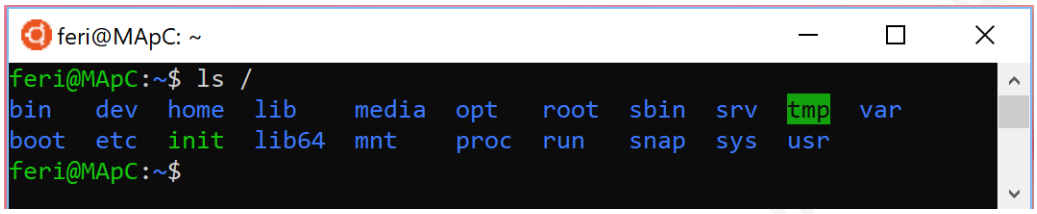



**Figure 4:** Ubuntu Command Line

## 2.2. Python

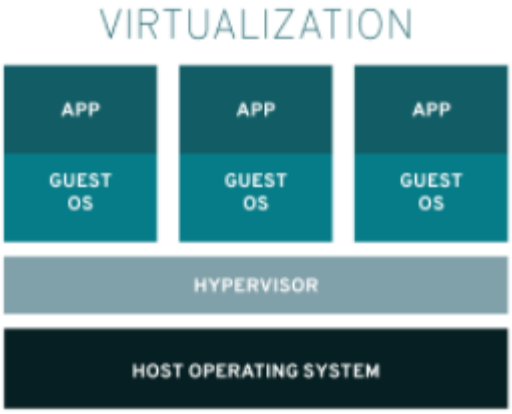
The Python scripting language is the foundation of several open-source forensic tools including Volatility and the Plaso/Log2Timeline tool. Since its release in 1991, Python has been gaining popularity as a scripting language (Donaldson, 2003). The downside of Python is its relative slowness and its rigid conventions compared to other languages. The development of Python diverged in 2008 with the introduction of version 3.0 which is incompatible with the predecessor 2.x. Both versions have updated releases as of the time of this writing (Python Software Foundations, 2017). The main reason to stick with the older version is that Volatility requires Python 2.7.x to operate.

The Python releases for Windows can be downloaded from www.python.org; the current versions are Python 3.6.4 and Python 2.7.14. The Windows 64-bit install package can be found at https://www.python.org/ftp/python/2.7.14/python-2.7.14.amd64.msi.

# 3. Virtualization Software

Virtualization techniques are commonly used in computer forensics. Running multiple instances of operating systems as a virtual machine provides the opportunity to run and analyze malicious software on the isolated guided environment. In the event of an issue, it is easy to stop the infected virtual machine and revert to a previous snapshot of the VM. The tradeoff for using virtual machines is the degraded performance and limited resources as the host operating system must allocate its memory and disk to the guest. In this case, Microsoft Windows is the host operating system that will be used, and which will host one or more virtual machines that look like separate computers with

Windows or Ubuntu Linux operating systems. In the test environment set up for this research, VMware and VirtualBox will also be installed. SANS Investigative Forensics Toolkit (SIFT) will be running on both Virtual Machines. The installation of SIFT is detailed in section 5.3. Measurements will be made to compare the speed of the tools running natively versus the tools running on the VMs.




The performance of virtualization will depend on the tasks executed, the type of virtualization and the virtualization platform used. The differences in response time are not so visible when the operator performs shorter tasks but are much more significant when long batch processes are running.

## 3.1. VMware Workstation

VMware Inc. is a subsidiary of Dell and provides cloud computing and platform virtualization software and services. It was the first company that successfully virtualized the x86 architecture. VMware isn't a single product but rather an ecosystem of connected tools and applications. VMware Workstation, their desktop virtualization tool for Windows and Linux, is a commercial product. VMware Workstation player is free for personal, non-commercial use. The latest release is version 14.1.1 and can be downloaded from the URL https://www.vmware.com/go/tryplayer.

## 3.2. Oracle VirtualBox

VirtualBox is the host-based virtualization solution owned by Oracle. Originally developed by Innotek Gmbh, it was first acquired by Sun Microsystems in 2008 then by Oracle in 2010. VirtualBox is a free and open-source solution that works with all x86 and x64 platforms including Windows, Mac, Linux, and Solaris. The latest version 5.2.8 can be downloaded from the URL https://www.virtualbox.org/wiki/Downloads.

# 4. Evidence Acquisition Tools

This section covers the installation and setup of different evidence acquisition tools that will be used on the workstation. Software tools such as Dd.exe, FTK Imager and EnCase forensic imager are used to acquire forensic images of system memory (RAM) and Hard Disks (HDD). Write blockers are used to protect the source media from accidental modification. The terminology used for evidence acquisition is discussed in the Appendix.

## 4.1. Dd.exe

The '*dd'* utility was originally developed for UNIX back in the 1970's. The free '*dd'* is available on UNIX, Linux, OS X and Windows. It is a simple utility that reads the source disk and writes it to a bit-by-bit raw image.

The Windows version '*Dd.exe'* is part of the Forensic Acquisition Utilities (FAU) package written by George Garner (Garner, 2016). FAU can be downloaded free from the URL http://www.gmgsystemsinc.info/fau/. The package comes with a ZIP compressed file that contains the executable and all the *".dll*" files needed to run it. There are separate file sets for x86 and x64 environments. To use DD.exe in Windows, the analyst only needs to decompress the content of the appropriate folder. The Windows-based *Dd.exe* has additional functions compared to the UNIX *dd*. It can send its output directly to a TCP or UDP port which allows it to perform remote acquisition over the network. The built-in help can be viewed using:

C:\> dd.exe --help or C:\> dd.exe /?

## 4.2. FTK Imager

The Forensic Toolkit (FTK) created by AccessData is a commercial product, but the company also offers a free of charge acquisition tool. Using FTK Imager the analyst can acquire disk images, memory images, and review them to determine if further

forensic analysis is required (AccessData, 2016). It has a graphical GUI and command line version as well.

FTK Imager can be downloaded from the URL https://accessdata.com/product-download. The latest 4.2.0 version was released on December 11, 2017. The instructions to operate the software are detailed in the *FTKImager_UG.pdf* that is included in the download package.

The program creates images from hard drives and other types of storage devices. FTK can create images in four different file formats: .E01, SMART, AFF, and raw. These images can be written in a single file or can be split into file segments that can be constructed at a later point. When the file is split into segments, the files can be moved and stored in several locations. With the compression option, the image size can be decreased by 50% or more at the cost of slower compression speed (AccessData, 2016).

With FTK Imager the user can create a forensic image of a physical disk or a logical volume by selecting the menu item "File > Create Disk Image". The wizard walks through the possible options for selecting the type of the source drive, the destination image type, filename, and compression as shown in Figure 5 and Figure 6.

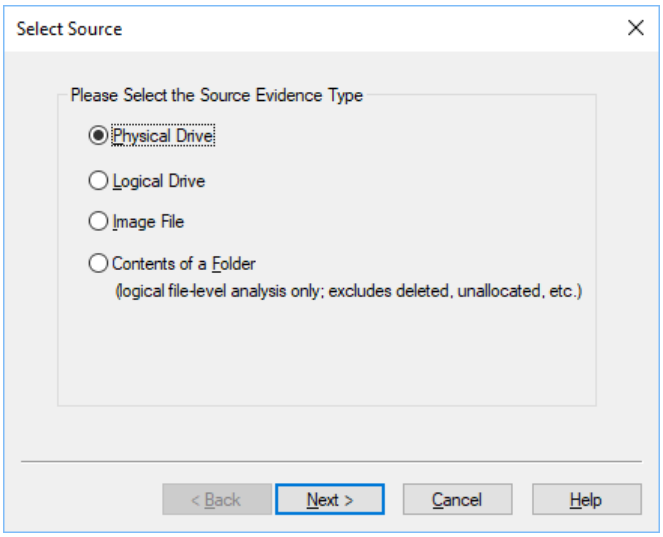



**Figure 5:** Selecting Source Evidence Type in FTK Imager

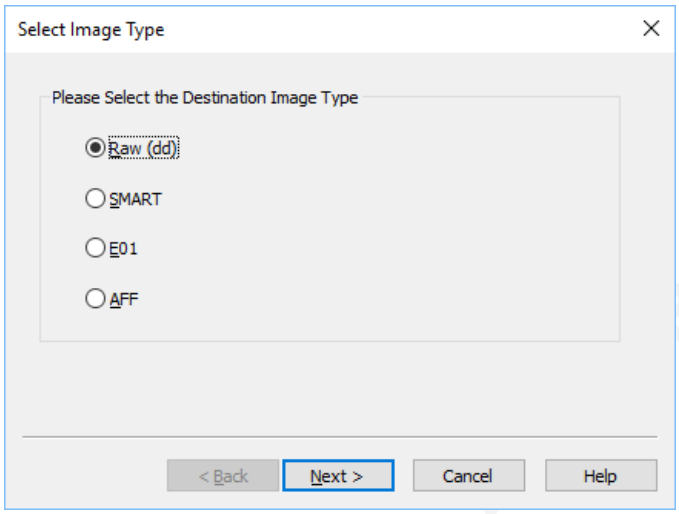



**Figure 6:** Selecting Destination Image Type in FTK Imager

FTK Imager can also create a logical image that only holds the contents of a folder. There is a warning message indicating that resulting image file isn't considered a forensic image as it doesn't contain any file system metadata, deleted files, or unallocated space. The message is shown in Figure 7.

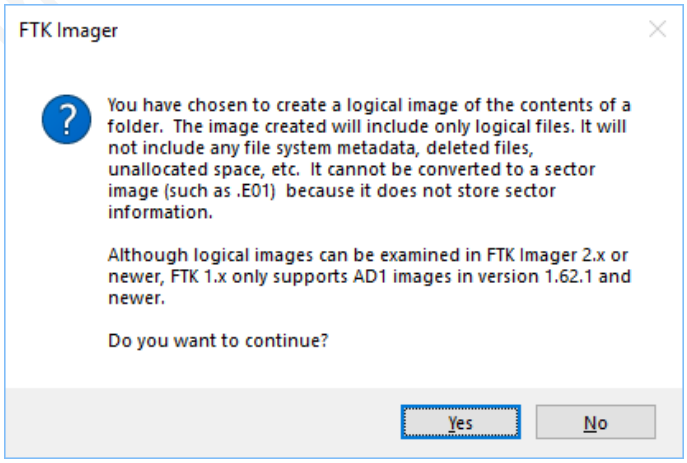



**Figure 7:** Message from FTK Imager when imaging a folder

The RAM image of the computer can be captured by selecting the menu item "File > Capture Memory". It will write the memory in raw format with the option of including the pagefile.

FTK Imager can also mount physical and logical disks and disk images. The function can be found when selecting the menu item “File >Image Mounting.” It supports mounting of raw images and E01 images but does not include support for the newer EnCase Ex01 image format. FTK Imager recognizes the file systems FAT 12, FAT16, FAT32, Ext2FS, Ext3FS, Ext4FS, ReiserFS3, exFAT, NTFS, HFS, HFS+, CDFS and VXFS according to the User Guide. Figure 8 shows the three mounting types available in FTK Imager:

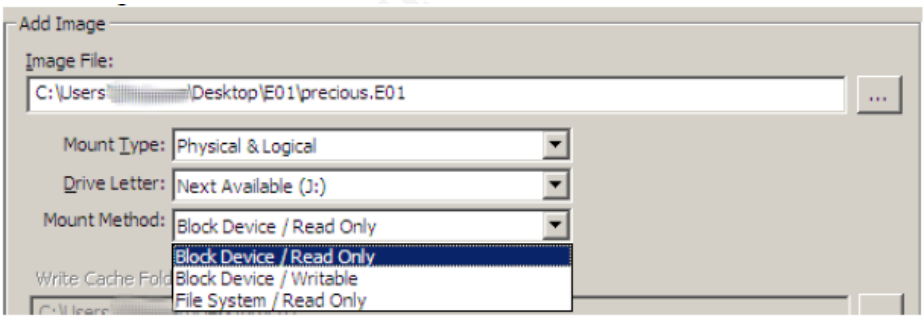



**Figure 8:** Image Mounting Types in FTK Imager

Selecting *Block Device/Read Only* reads the evidence as a block device. The mounted device must be viewed using any Windows application that performs Physical Name Querying. Selecting the *"Block Device/Writable*" will allow writing to the evidence. The *"File System/Read Only*" method will read the evidence as a read-only device that can be viewed using Windows Explorer.

AccessData offers a standalone version of the imager that doesn't need to be installed on the computer. The FTK Imager Lite has the latest version 3.1.1 that was released on October 16, 2010. It can be downloaded from the URL https://accessdata.com/product-download. The package comes in a single ZIP file. The compressed file can be extracted to an external drive or its content can be written to a CD. For using the tool the analyst should connect the external drive to the computer to be acquired and execute the FTK Imager.exe file.

The FTK Imager has command line versions for Windows, OSX, and Linux. The command line version is also 3.1.1 similar to the Lite version but was released in 2012. The Windows version is 32-bit only; the Linux versions are available in 32-bit and 64-bit as well. The command line version has most of the same functionality as the GUI version with few differences. The download package doesn't include a user manual, but there is built-in help with instructions for usage. The help option can be accessed using the command:

```
C:\> ftkimager –help
```

## 4.3. Encase Forensic Imager

Encase Forensic from OpenText is a full-featured forensic package that can create images and perform forensic analysis on them. It is an industry-accepted tool used by law enforcement and private companies. EnCase is used to acquire, analyze, and report on evidence. OpenText also offers its imaging module, the Encase Forensic Imager, as a free product. It can be downloaded from the URL https://www.guidancesoftware.com/encase-forensic-imager/forensic-imager-download. The latest 7.10 version was released on June 30, 2014. The usage instructions to operate the software are detailed in the *EnCase_Forensic_Imager_v7.10_User's_Guide.pdf* that is included in the download package.

The Encase Forensic Imager creates images of physical drives, logical volumes, individual files, folders or other image files. The output image is the Expert Witness Format E01 or Ex01. The newer Ex01 format is available starting with Encase version 7. It provides compression AES256 encryption and options for MD5 and SHA-1 hashing (Guidance Software, 2014).

As Figure 9 indicates, evidence can be acquired by selecting the menu item “Add Evidence” then selecting “Add Local Device” “Add Evidence file” or “Add Raw Image”. The supported evidence files for acquisition are EnCase evidence files (.E01, .Ex01), logical evidence files (.L01, .Lx01), DD images, VMware files (.vmdk), or Virtual PC files (.vhd).

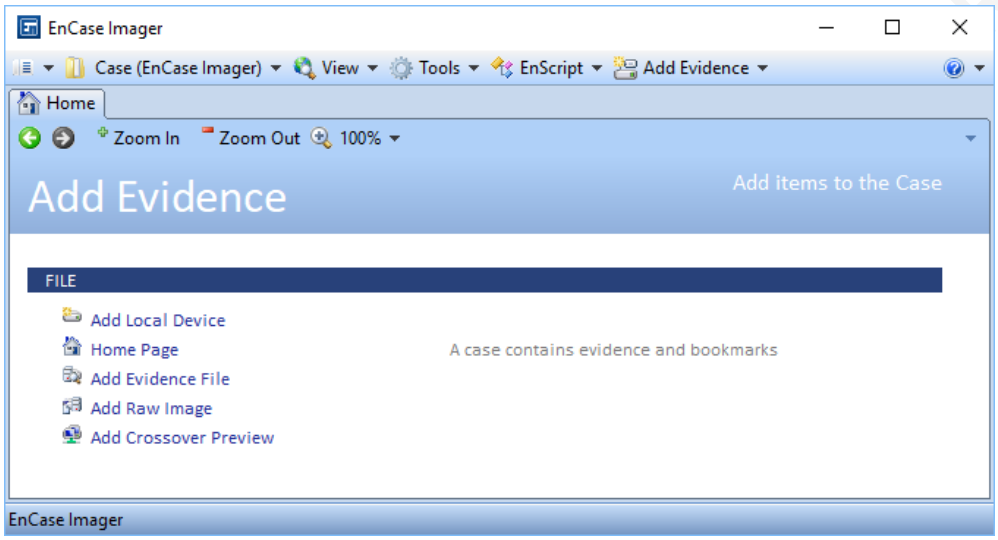



**Figure 9:** Adding Evidence in EnCase Imager

Encase Forensic Imager doesn't provide mounting functionality, but rather provides a simple preview of the image within that utility only. Encase Forensic is the full commercial package that allows one to perform analysis and reporting on the evidence.

## 4.4. Write blockers

Write blockers are small hardware devices that are used to prevent any data from being written back to the source during the imaging process. To acquire hard drives that are connected to Windows, the analyst should always use write blocker hardware. According to Guidance Software, "Windows writes to any local hard drive visible to it. Windows will, for example, put a Recycle Bin file on every hard drive that it detects and will also change Last Accessed date and time stamps for those drives" (Guidance Software, 2014 page 22). The source disk is connected to the input port of the write blocker. The computer which performs the acquisition is connected to the output. These devices connect an IDE or Serial ATA hard drive to the analysis computer through one of several interfaces. Most write blockers support USB, FireWire and eSATA connections, though the speed of these interfaces varies depending on the age of the device.

The performance of two write blocker devices was tested in this research. One is the Forensic UltraDock (FUD) v5 from Wiebetech by CRU as shown in Figure 10. The device was distributed with the SANS FOR408 course materials and supports IDE, SATA disks with eSATA, Firewire, USB 2.0 / 3.0 interfaces (Wiebetech by CRU, 2018).


**Figure 10:** Forensic UltraDock v5

The second device that will be used for the test is Tableau Forensic T35u-RW from Guidance as shown below in Figure 11. It is an older device that supports IDE, SATA disks, and USB 2.0 / 3.0 interfaces. The device can be configured to operate read-only or read/write mode by setting a 4-position DIP switch (Guidance Software /Tableau Hardware, 2017).

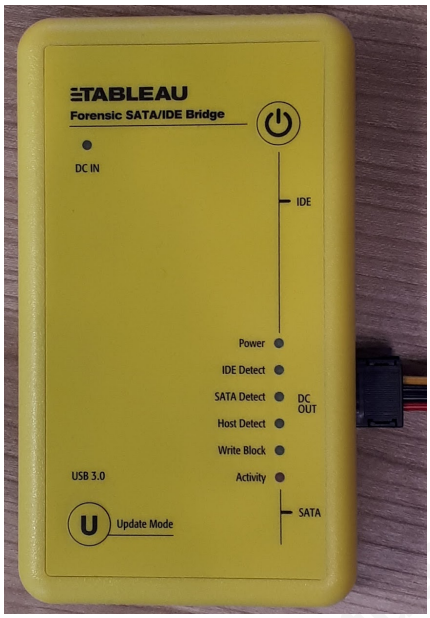



**Figure 11:** Tableau Forensic T35u-RW

## 4.5. Destination media

The acquired image files then have to be stored on the analysis computer either on the internal Hard Drive or on an external Disk connected by the USB interface. The workstation used for this research has an internal Raid storage disk with 5.18 TB of capacity. The acquired images can be stored on this internal disk which is feasible for storing relatively small images. The downside is that storing multiple images can use up available space. To acquire images of 250 GB to 2 TB hard drives, it is necessary to have enough storage space on the destination media. Additional external storage can also be connected to the computer to increase the available capacity.

# 5. Evidence Analysis Tools

There are two main types of evidence analysis: disk forensics and memory forensics. Memory forensics is important when dealing with memory-resident malware. Early memory analysis was mainly limited to byte and string searches. Current memory structures are better understood, and new tools exist that allow more detailed analysis of the contents of memory. RAM is now the best place to discover malware running on the system. Malware “wants to hide”, but it also has to be executed to be effective. As the

size of system's memory is steadily increasing it is becoming less volatile and looks like a secondary file system. The tracks of running processes, threads, and DLLs are visible in the RAM (Lee, Tilbury, 2016).

## 5.1. Mandiant Redline

Mandiant Redline is probably the most user-friendly free memory analysis software available. It guides investigators through the process of evaluating the system for finding compromise or infection. It provides a series of investigation steps to find malware to review processes, network connections, memory sections, untrusted handles, hooks, and drivers. Redline attempts to quickly point out potentially malicious objects via behavioral analysis and known bad heuristics. The latest 1.20.1 version was released on October 23, 2017. It can be downloaded from https://www.fireeye.com/services/freeware/redline.html. The following options are provided when an analyst opens Redline, as Figure 12 indicates:

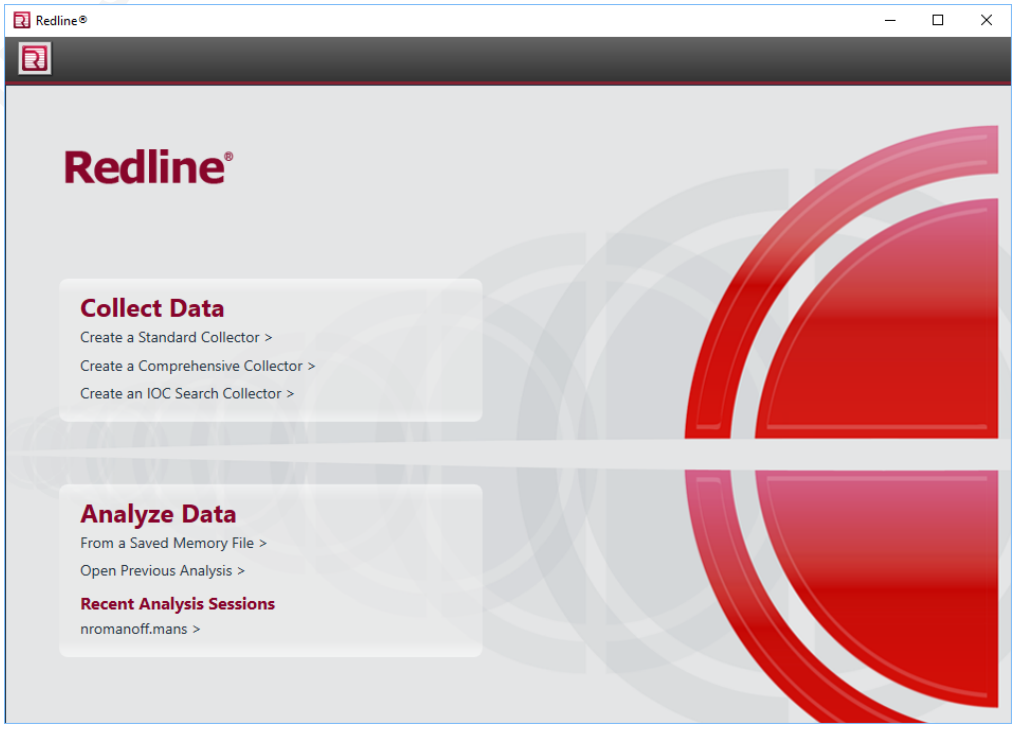



**Figure 12:** Redline data collection and analysis options

**Create a Standard Collector**. A "collector" is a portable agent used to collect data from a live system. It is implemented as a batch script that should be run from a removable media such as a USB drive. It collects the data necessary for a full live memory analysis. The result is written back to the USB drive and can then be imported back to Redline.

**Create a Comprehensive Collector.** This option is similar to the previous function but additionally collects file metadata, Registry hives, Event Logs, and Prefetch files.

**Create an IOC Search Collector.** This option collects the host-based artifacts to perform a scan for specified Indicators of Compromise (IOC).

**Analyze Data / From a Saved Memory File**. This option allows the analyst to open a raw memory image file for analysis. It is the main menu that should be used for memory forensics.

**Open Previous Analysis.** This option is used to review previous analysis results.

## 5.2. The Sleuth Kit and Autopsy

The Sleuth Kit (TSK) and the Autopsy Forensic Browser are open-source forensic analysis tools that were first released by Brian Carrier in 2001. They are available on Linux, Windows, and OSX platforms. TSK is a C library and a collection of more than twenty command line tools that can analyze disk and file system evidence. The tools are organized into five groups called layers (Carrier, 2017). These layers are: *File System Layer*, *Filename Layer, Metadata Layer*, *Data Unit Layer*, and *File System Journal* tools. Each tool name has two parts, where the first part identifies its group and the second part identifies its function. The *File System Layer* includes the data that describes the layout and general information about a file system. The tool *fsstat* belongs to this layer. *Fsstat* reads the boot sector or superblock and other data structures that are specific to the different types of file systems. The *File Name Layer* tools (such as *ffind* and *fls*) process the file name structures, which are typically located in the parent directory. The *Metadata Layer* tools (such as *icat, ifind, ils, istat*) process the metadata structures, which store the details about a file. The *Data Unit Layer* tools (such as *blkcat, blkls, blkstat,* and *blkcalc*) process the data units where file content is stored. The *File System Journal* tools (such as

*jcat* and *jls*) process the journals of some filesystems. The journal records the metadata (and sometimes content) updates that were made which could help recover recently deleted data. The TSK supports Ext2/3 (Linux-ext2, linux-ext3), FAT (fat, fat12, fat16, fat32), NTFS (ntfs), and UFS1/2 (freebsd, netbsd, openbsd, Solaris) file system formats.

Autopsy is a graphical frontend for TSK which allows browser-based access to the TSK tools as shown in Figure 13. Autopsy has an intuitive design. Its main modules are *Timeline Analysis, Hash Filtering, Keyword Search, Web Artifacts, Data Carving, Multimedia, and Indicators of Compromise* (Carrier, 2017).

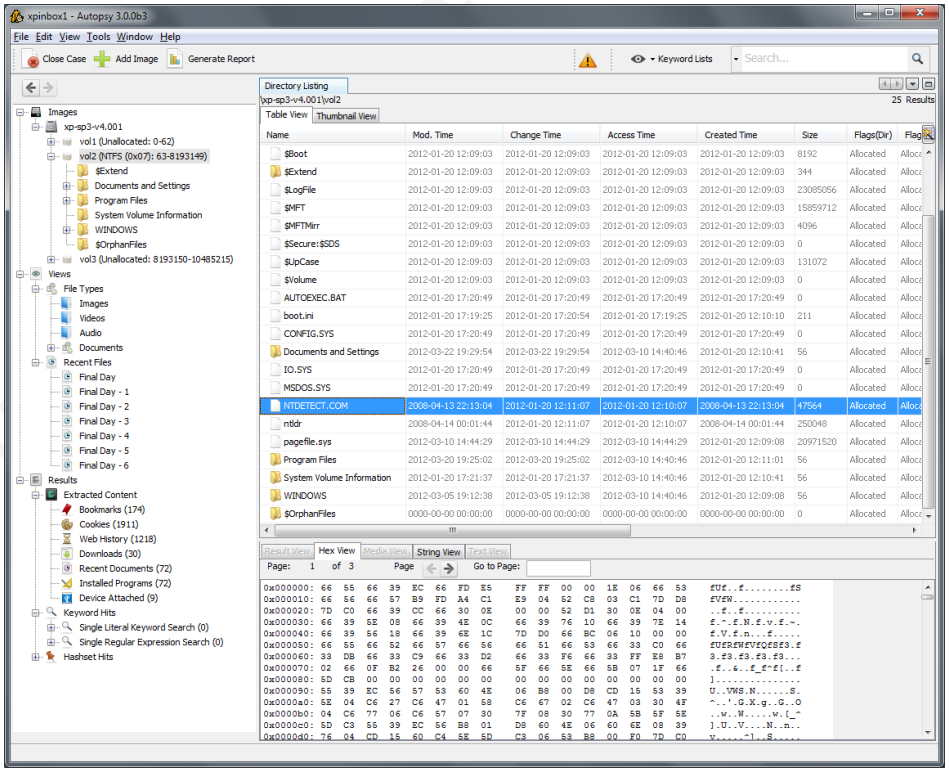



**Figure 13:** The Graphical Interface of Autopsy

The latest 4.6.0 version of TSK and Autopsy was released on February 23, 2018. The TSK Windows binaries and Autopsy can be downloaded from the URL http://sleuthkit.org/sleuthkit/download.php and http://sleuthkit.org/autopsy/download.php respectively.

## 5.3. SANS SIFT

As the SANS Investigative Forensics Toolkit (SIFT) homepage explains "The SIFT Workstation is a group of free open-source incident response and forensic tools designed to perform detailed digital forensic examinations in a variety of settings" (SANS, 2018). SIFT was created and maintained by Rob Lee at the SANS Institute.

The current version is based on Ubuntu LTS 16.04. It can be downloaded from SANS as a pre-configured Virtual Machine appliance in ".ova" format. Manual installation can also be done on an existing Ubuntu system running the SIFT-CLI command line tool. SIFT can be installed on Windows 10 Creators Edition or later version using the "Windows Subsystem for Linux" tool. Installation of the tool and the Ubuntu bash shell was discussed in section 2.1. The SIFT-CLI command line tool has to be used from the *bash* shell. The latest v1.5.2 version of SIFT was released in 2018. It can be downloaded from http://digital-forensics.sans.org/community/downloads.

After downloading the preconfigured "SIFT-Workstation.ova" virtual appliance, it has to be imported to the Virtualization software. The settings used for importing SIFT virtual appliance into VMware are shown in Figure 14.

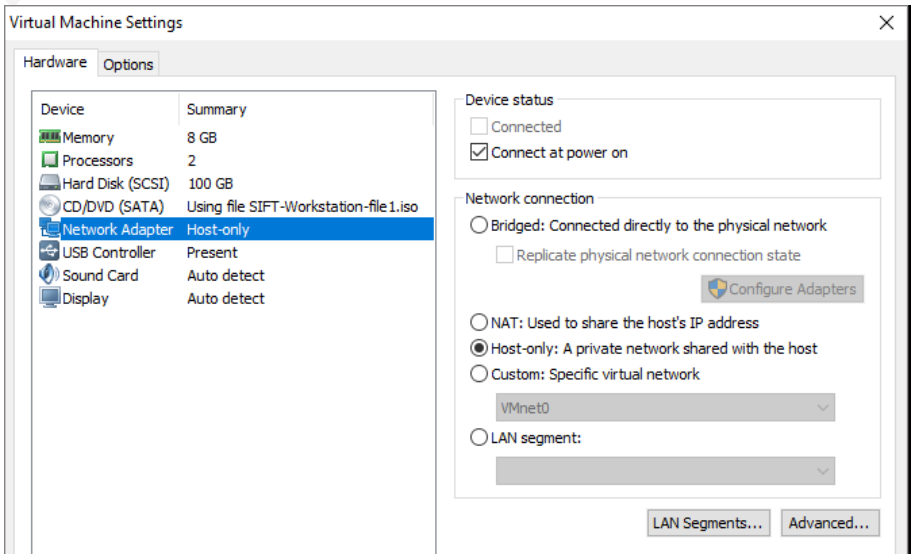



**Figure 14:** SIFT Virtual appliance in VMware

The settings used for importing SIFT virtual appliance into VirtualBox are shown in Figure15 below.

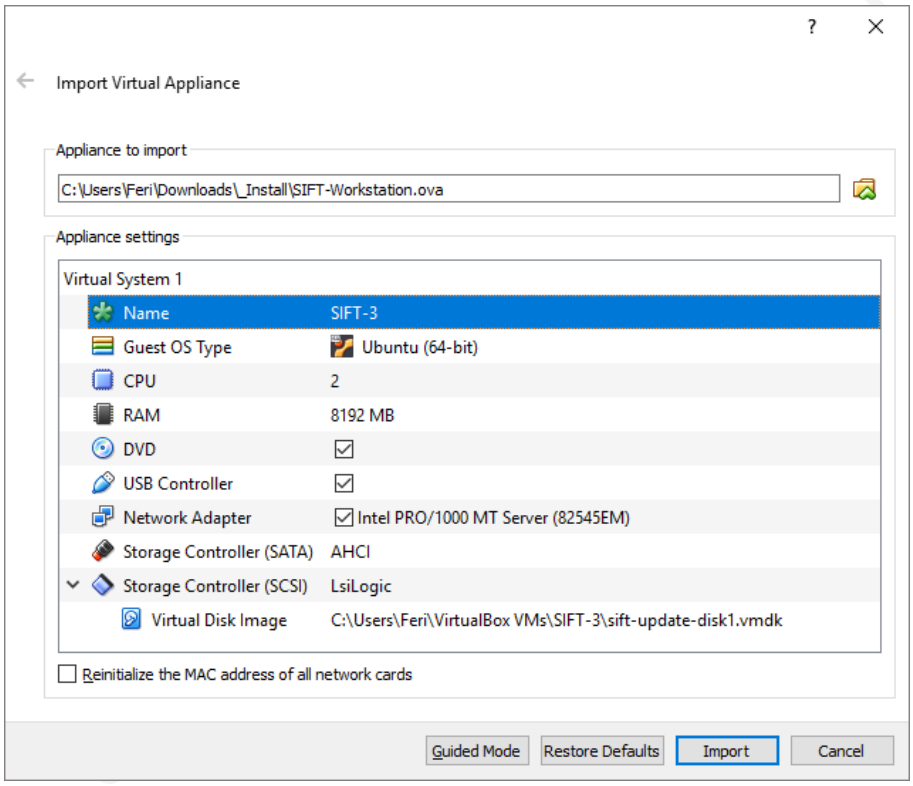



**Figure 15:** Importing the SIFT Virtual appliance in VirtualBox

SANS provides extensive documentation for installing and using SIFT including the description and usage of all the embedded packages. Cheat sheets are also available to provide concise reference material for the daily use of forensic investigations. The SIFT documentation can be found on the URL http://sift.readthedocs.io/en/latest/.

SIFT supports the NTFS, iso9660, HFS, RAW, swap, memory, FAT (fat12, fat16, fat32), EXT (ext2, ext3, ext4), UFS(ufs1, ufs2), and VDMK file system formats. The following evidence image formats are supported: raw (Single raw file (dd)), aff (Advanced Forensic Format), afd (AFF Multiple File), afm (AFF with external metadata), afflib (All AFFLIB image formats including beta ones), ewf (Encase Expert Witness format), split raw (Split raw files), affuse, and split ewf (Split E01 files).

The SANS SIFT workstation includes the mounting utilities *mount*, *ewfmount*, *xmount,* and *affuse*. There are 100+ software tools included in SIFT such as *log2timeline* (Timeline Generation Tool), *Rekall Framework* (Memory Analysis), *Volatility Framework* (Memory Analysis) with 3rd Party Volatility Plugins, *bulk_extractor*, *SleuthKit* with *Autopsy*, *log2timeline*, *Plaso*, *Qemu*, and *regripper* with its plugins.

## 5.4. Volatility

The Volatility Framework is the most well-known memory analysis tool. The first version was released publicly at Black Hat DC in 2007 and has recently evolved dramatically with improved functionality. The latest version, Volatility 2.6 was released in December 2016. It can be downloaded from http://www.volatilityfoundation.org/releases.

Volatility supports 32 and 64-bit Windows 10, 8, 7, XP, Vista, Windows Server 2012 (64bit), 32 and 64-bit Windows Server 2008, 2003, Linux kernels from 2.6.11 to 4.2.3+, and Mac OSX. The supported memory formats include Raw Physical Memory, Expert Witness Format (.E01), Windows Crash Dump, Windows Hibernation, VirtualBox Core Dumps, VMware Saved State (.vmss) and Snapshot (.vmsn).

Volatility is not as user-friendly as Redline, but it is far more powerful. It is a command line tool written completely in Python (Volatility Foundation, 2017). Volatility is included in the SIFT workstation and can be started with the *vol.py* command. It is also available as a Windows standalone executable file, which can run without the need to install the Python environment. The usage syntax of Volatility is:

```
vol.py -f [image] [plugin] --profile=[PROFILE]
```

The –f option is required, and it should be followed by the [image] option that describes the name and path of the memory image file to be analyzed. The [plugin] option tells Volatility which plugin to run to analyze the image. Finally, the system type of the memory image has to be specified using the [PROFILE] parameter.

To begin, the [PROFILE] has to be determined. The *imageinfo* plugin reads the image and recovers metadata such as the system time the image was captured, and the

suggested profile (or profiles) describing Operating System and service pack information. The analyst must always indicate this profile when running Volatility analysis. The KDBG (Kernel Debugger Block) information is also needed to speed up the processing time of some plugins. Figure 16 shows the result of the *imageinfo* plugin running within the Ubuntu *bash* shell:

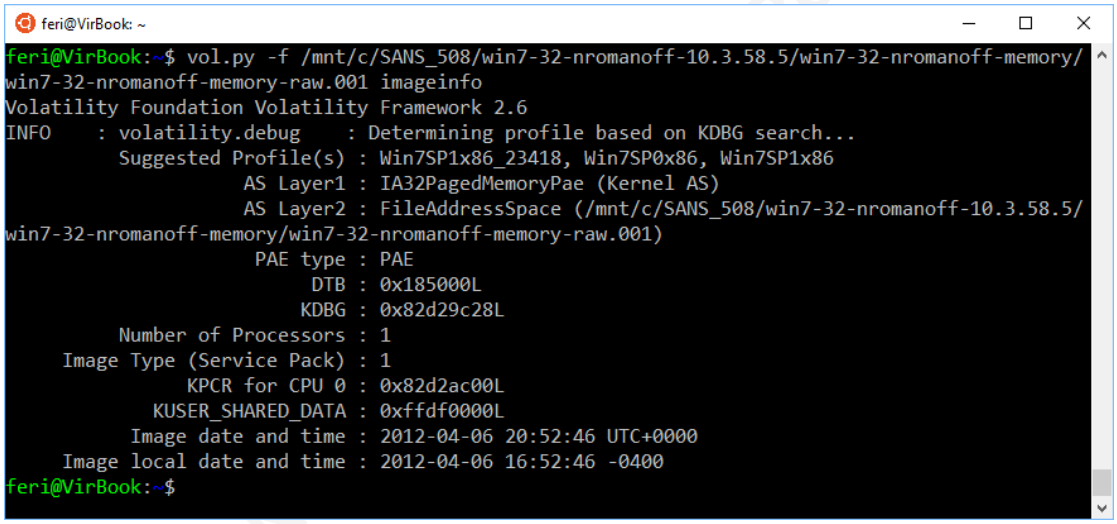



**Figure 16:** Volatility imageinfo in Ubuntu bash

Figure 17 indicates the result of *imageinfo* plugin running within the Command Prompt:

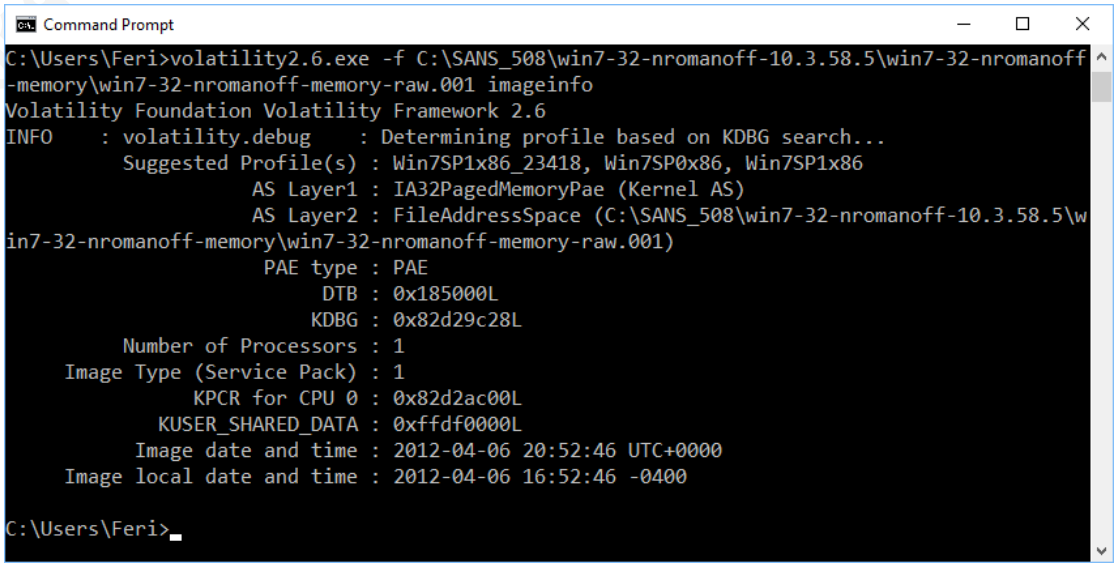



**Figure 17:** Volatility imageinfo in Windows Command prompt

There are several plugins in Volatility; listed below are some important ones that can be used to identify the rogue processes running in the memory and collecting network artifacts:

**pslist** Prints all running processes within the EPROCESS doubly linked list.

**psscan** Scans physical memory for process blocks. Psscan doesn't simply follow the EPROCESS list. Hidden processes may be identified by using this plugin.

**pstree** Prints the process list as a tree showing parent relationships (using the EPROCESS linked list).

**netscan** Identifies network sockets and TCP structures resident in memory.

**malsysproc** Scans system processes for anomalies to find malware "pretending" to be legitimate system processes.

**malfind** Scans process memory sections by looking for indications of code injection.

The help option can be invoked using the -h or –help option to list all of the available options:

$ vol.py –h or $ vol.py –help

## 5.5. Log2timeline with Plaso

Timeline analysis is one of the most important capabilities which helps to find malicious activity in the system. It allows the investigator to automatically collect artifacts from the registry, filesystem, and operating system and arrange them in order of their occurrence time. Intruders are using anti-forensic techniques to cover their tracks on the system. But it is nearly impossible to delete all of the footprints; there are too many of them. Timeline data is extremely useful in tracking the activity of adversaries. Because it is handled in the OS level, using encryption and covert channels is useless. Anything that an attacker does will interact with the system (starting programs, opening, modifying, deleting files, etc.).

Log2timeline is a tool that can extract the timestamps from files found on a computer system and aggregate them to create a Super Timeline. It was created by Kristinn Guðjónsson as part of SANS Gold Paper discussion (Guðjónsson, 2010). Plaso is a Python-based back-end engine used by log2timeline to create Super Timelines. The name *plaso* is an acronym: "Plaso Langar Að Safna Öllu" (in Icelandic) which means "Plaso Want to Collect Everything". The goal of log2timeline is to provide a single tool that can parse various log files and forensic artifacts from computers to produce a single correlated timeline. Plaso is a suite of tools that collects timestamps from a system, archives them, and searches through the data (Metz, 2016).

The key tools in Plaso:

**log2timeline** This is the main front-end that is used to extract events from a file, mount point, or an image file and then saves the events to a *.plaso* storage file.

**pinfo** The *.plaso* storage file contains information about how and when the collection took place. Pinfo prints out this information.

**psort** This tool is used to filter, sort, and process the *.plaso* storage file to convert the results into a human-readable format.

There are numerous plugins/parsers that exist in *plaso* and which are used to collect different Windows, Registry, Webhistory artifacts. Parsers for Linux, Android, and Mac are also available. Plaso and Log2timeline are preinstalled in the SANS SIFT workstation, either in the Virtual Machine appliance or in Ubuntu bash shell under Windows 10. The usage syntax of log2timeline.py:

```
$ log2timeline.py [arguments] [STORAGE FILE] [SOURCE]
```

The Command options:

STORAGE FILE Plaso output database file - /path/to/output.dump

SOURCE Source of device, image, file - /path/to/image.dd

arguments -z TIMEZONE_OF_IMAGE
-f FILE_LIST
--parsers PARSER_LIST

Creating the overall Super Timeline is a time-consuming task. It is also possible to create more focused timelines to collect data from the relevant files based on the interest of the analyst. This method is called Targeted Timeline Collection. The syntax of usage is:

```
$ log2timeline.py -f FILE_LIST --parsers PARSER_LIST
[STORAGE FILE] [SOURCE]
```

The FILE_LIST argument defines the list of files to include for targeted collection in "/path/to/file" format. The PARSER_LIST argument defines the parsers used by the tool. The parser list is a comma separated file where each entry is the name of a parser or another parser list.

The output of Plaso is a ZIP compressed archive that stores several files. Psort is the post-processing tool that reads the *.plaso* file and extracts the events in chronological order. It can remove duplicates and present the output in human-readable format. The output can include several varieties of CSV, JSON, MySQL, and SQLite. The resulting data can then be imported to Excel for analysis.

## 6. Performance Comparison

To measure the evidence acquisition performance, multiple tests were run using different acquisition software with the two write blockers with write destinations to internal and external storage media. The source media to be acquired was a 250 GB Samsung SSD 850 SATA Disk Device. This source disk was used for all tests.

Two kinds of acquisition software were used for the tests, first the FTK Imager then the Encase Forensic Imager. FTK was used creating raw file output and compressed E01 file output. Encase doesn't support raw format, only the compressed E01 output file format was used with it.

One set of tests were made with using the Forensic UltraDock v5 write blocker. The source disk was connected to the SATA data interface of the UltraDock. The UltraDock is powered by its own power adapter; the source disk was powered by the SATA power interface of UltraDock. The UltraDock device was connected to the forensic workstation through a USB3 port.

Two types of destination media were used for the tests. During the first test, the acquired image files were written to the internal hard disk of the workstation. The second time, the image files were written to a 1.8 TB external Maxtor portable USB3 Hard Disk.

All tests were repeated with using the Tableau Forensic T35u-RW write blocker. This write blocker also has an external power adapter. The source disk was connected to the SATA data input port and SATA power connector of the write blocker. The Tableau device was connected to the forensic workstation through a USB3 port.

Table 1 populates the measurement setups grouped by the Acquisition software tools, Write blockers, Destination media, and Image File formats. The measurement results are indicated in (hours: minutes: seconds) format.

| Acquisition software | Write blocker | Destination media | Image File format | | Acquisition time (h:m:s) |
|---|---|---|---|---|---|
| | | | raw | E01 | |
| FTK Imager | UltraDock | Internal | 1:30:42 | 1:26:33 | |
| | | External USB | 1:24:10 | 1:32:44 | |
| | Tableau | Internal | 1:12:29 | 1:11:34 | |
| | | External USB | 0:56:19 | 1:13:29 | |
| Encase Forensic Imager | UltraDock | Internal | | 2:12:00 | |
| | | External USB | | 1:17:40 | |
| | Tableau | Internal | | 2:03:02 | |
| | | External USB | | 1:15:04 | |

**Table 1:** Image Acquisition Times

The graphical representation of the results of the tests is shown in Table 2.

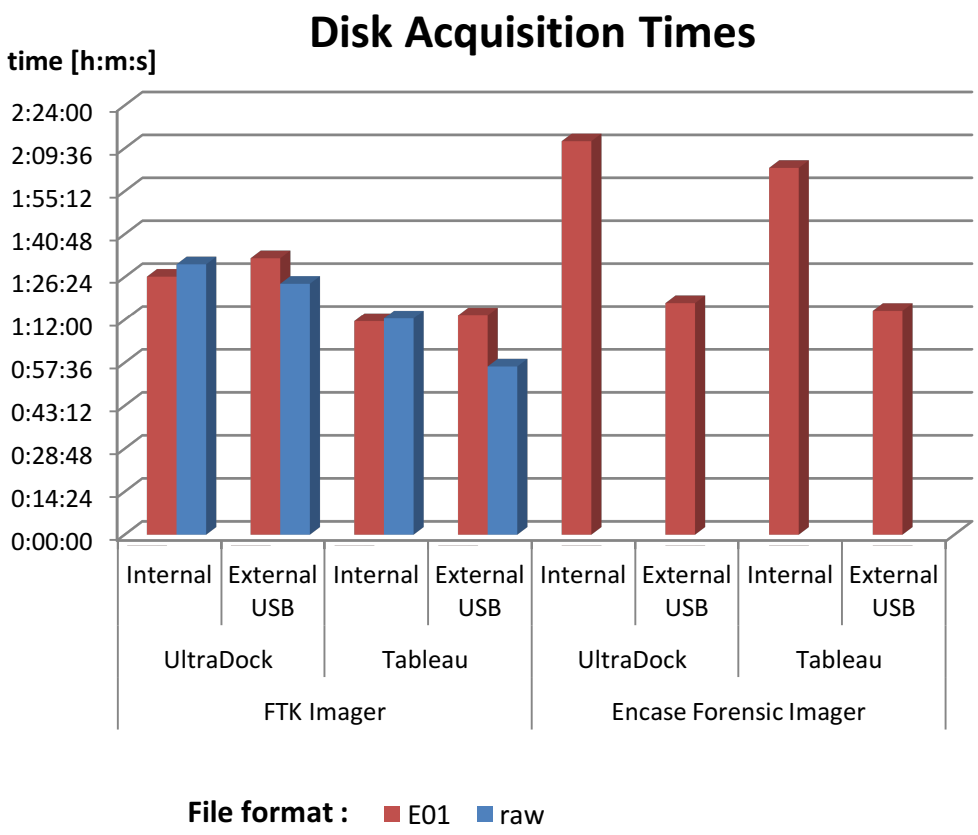



**Table 2:** Disk Imaging Performance

According to the results, there is no significant difference in the acquisition times of the uncompressed raw format and the compressed E01 format files. The 2.4 GHz Intel Xeon CPU provides sufficient computing power for compressing the data. There is a difference in the acquisition times when comparing the Tableau and UltraDock write blockers. Tableau performs slightly better in all measurement setups. There is a significant difference in the processing time of the two imaging software. FTK Imager performs better than Encase except in one case when UltraDock is used with external USB storage. Encase needs more time when writing to the internal Raid storage, more specifically, 53% more time using UltraDock and 72% more time with Tableau. In overall comparison, FTK Imager produces better results in disk acquisition times.

Another set of tests were ran to measure the performance of the forensic workstation performing memory analysis. Two memory images were used for testing.

The first image is originated from a Windows 7 SP1 64-bit computer with 2:00 GB data size. The second image is from a Windows 10 Professional 64-bit computer with 8.00 GB data size. To test the analysis performance of the workstation Volatility *imageinfo* plugin was running against both memory images. For the first test, Volatility2.6.exe was running in the Windows Command prompt. The second test was running under bash shell of Windows Subsystem for Linux fitted with SIFT workstation that includes Volatility. Then to test the impact of virtualization, Volatility was running within the Virtual appliance of Ubuntu based SIFT workstation. An updated copy of SANS SIFT workstation was installed on VMware and on VirtualBox. Both VMs were configured using two processor cores and 8 GB RAM. The results of the tests are shown in Table 3.

Redline was only ran on Windows, its performance was tested with the 2 GB and 8GB memory images. The execution time of the function "Analyze data / From a Saved Memory Image" was measured with both RAM images. The Redline results are also incorporated in Table 3.

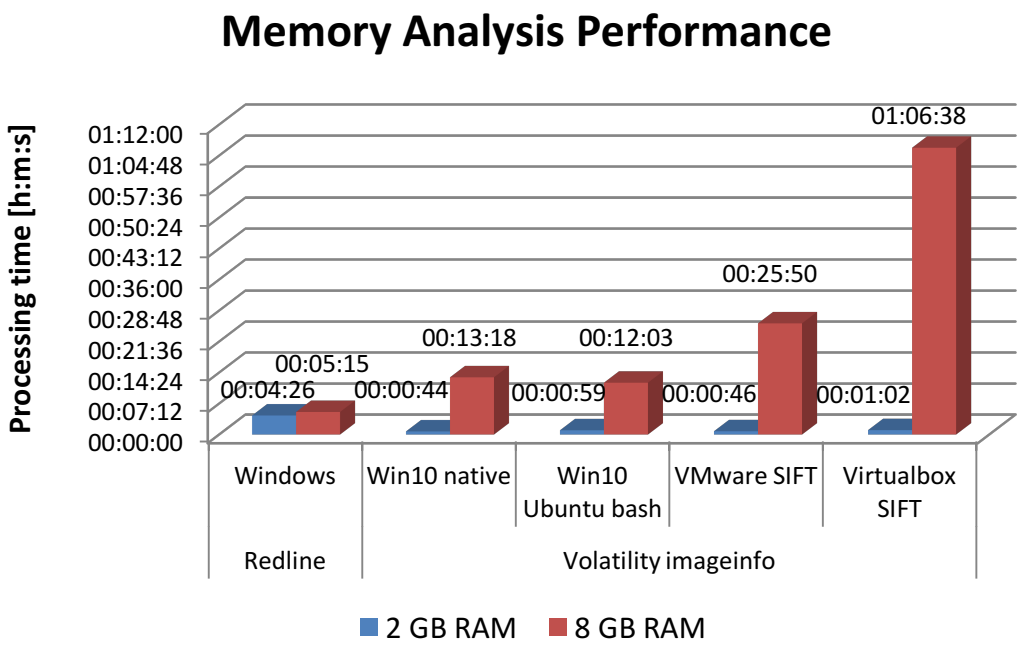



**Table 3:** Memory Analysis Performance

The analysis time of the 2 GB image with Volatility *imageinfo* plugin did not show significant differences in any environment. All results indicated around a one-minute running time. The native Windows command produces the best result followed by the VMware SIFT installation. VirtualBox SIFT VM took the longest time to run. The analysis times needed to process the 8 GB was significantly greater. The Windows environments, either the Command prompt or the Ubuntu *bash* in Windows Subsystem for Linux were close with the shortest 12 minutes and 3 seconds of Ubuntu *bash*. VMware SIFT required 25 minutes to run while VirtualBox SIFT ran for 1 hour and 6 minutes. The assigned 8 GB memory size for both Virtual Machines were the same size as the analyzed 8 GB image that utilized the memory handling capabilities of the VMs. VMware performed significantly better with the larger memory image. Redline produced similar processing times for both memory images.

## 7. Conclusion

It is time and labor -intensive process to create and set up a forensic platform on Windows 10. There are precooked forensic environments available on Linux, such as the SANS SIFT workstation. Using Windows 10 gives analysts the opportunity to use software available on Windows only, and the new Windows Subsystems for Linux environment allows them to run Linux-based tools. Virtualization software such as VMware and VirtualBox are available to run virtualized tools. The Windows native setup provides more system resources such as CPU utilization and allocated RAM to the running programs. Since it is a part of the operating system, WSL can take advantage of the available resources. Programs running on Virtual Machines can use resources limited by the settings of the VM.

The forensic investigation platform built on the Windows machine is effective if it will comply with the requirements of the forensic analysis tools (Carrier, 2003). The platform is effective if the results of the measurements are usable, comprehensive, accurate, deterministic, and verifiable.

Author Name, email@address:Ferenc Kovacs

As demonstrated by the paper it is possible to install the listed forensic tools on Windows 10 computer that cover all of the forensic functions. The measurement results show that the installed platform is effectively usable. Those professionals in the industry can use this guide to set up their Windows 10 computer to perform forensic analysis.



Author Name, email@address:Ferenc Kovacs

# Figures and Tables



Author Name, email@address:Ferenc Kovacs

# References

Myerson, Terry (September 28, 2015) *Privacy and Windows 10* Retrieved from https://blogs.windows.com/windowsexperience/2015/09/28/privacy-and-windows-10/.

Nadella, Satya (2018) *Privacy at Microsoft* Retrieved from https://privacy.microsoft.com/en-US/.

Phillips, Gavin (October 6, 2016) *The Complete Guide to Windows 10 Privacy Settings* Retrieved from https://www.makeuseof.com/tag/complete-guide-windows-10-privacy-settings/.

Martin, David M. (February 22, 2017) *OS X as a Forensic Platform.* Retrieved from: https://www.sans.org/reading-room/whitepapers/forensics/os-forensic-platform-37637.

Vandeven, Sally (September 19, 2014) *Forensic Images: For Your Viewing Pleasure* Retrieved from https://www.sans.org/reading-room/whitepapers/forensics/forensic-images-viewing-pleasure-35447.

Giesbrecht, Shelly (July 22, 2015) *Coding For Incident Response: Solving the Language Dilemma* Retrieved from https://www.sans.org/reading-room/whitepapers/forensics/coding-incident-response-solving-language-dilemma-36107.

Donaldson, Toby (2003, April 25). *Python as a First Programming Language for Everyone.* Retrieved from https://www.cs.ubc.ca/wccce/Program03/papers/Toby.html.

Python Software Foundations (2017, September 10). *Should I use Python 2 or Python 3 for my development activity?* Retrieved from https://wiki.python.org/moin/Python2orPython3.

Garner, George M. Jr (May 19, 2016.) *Forensic Acquisition Utilities* Retrieved from http://www.gmgsystemsinc.info/fau/.

AccessData (March 31, 2016) *Imager User Guide* Retrieved from http://marketing.accessdata.com/e/46432/FTKImager-UG/531gls/1237892476.

Guidance Software (July 30, 2014) *EnCase Forensic Imager Version 7.10 User's Guide* Retrieved from https://www.guidancesoftware.com/encase-forensic-imager/forensic-imager-download.

Guidance Software /Tableau Hardware (2017) *Tableau Forensic SATA/IDE Read-Write Bridge.* Retrieved from https://www.guidancesoftware.com/tableau/hardware/t35u-rw.

Wiebetech by CRU (2018) *Forensic UltraDock FUDv5.5 product sheet.* Retrieved from https://www.cru-inc.com/products/wiebetech/forensic-ultradock-v5-5/.

Carrier, Brian (2003) *Defining Digital Forensic Examination and Analysis Tools Using Abstraction Layers.* Retrieved from: http://www.dfrws.org/conferences/dfrws-usa-2002/sessions/defining-digital-forensic-examination-and-analysis-tools.

Carrier, Brian (March 17, 2005). *File system forensic analysis.* Addison Wesley Professional.

Carrier, Brian (2017) *The Sleuth Kit.* Retrieved from http://sleuthkit.org/index.php.

Author l@address:Ferenc Kovacs

INFOSEC Institute (n.d.) *Autopsy: a platform overview* Retrieved from http://resources.infosecinstitute.com/category/computerforensics/introduction/free-open-source-tools/autopsy-forensics-platform-overview/.

SANS (2018) *SANS SIFT Workstation: Investigative forensic toolkit download.* Retrieved from http://digital-forensics.sans.org/community/downloads.

SANS Institute. (2014). *SANS Investigative Forensics Toolkit Documentation.* Retrieved February 10, 2018, from SANS DFIR: http://sift.readthedocs.io/en/latest/

Rob Lee, Chad Tilbury (2016) *FOR 508 | Advanced Digital Forensics and Incident response.* The SANS Institute

Mandiant/ FireEye (n.d.) *Redline* Retrieved from https://www.fireeye.com/services/freeware/redline.html.

Mandiant/ FireEye (2017) *Redline User Guide.* Retrieved from https://www.fireeye.com/content/dam/fireeye-www/services/freeware/ug-redline.pdf.

Volatility Foundation. (n.d.). *Volatility Foundation.* Retrieved from http://www.volatilityfoundation.org/.

Volatility Foundation. (October 2, 2017). *Volatility Wiki.* Retrieved from GitHub: https://github.com/volatilityfoundation/volatility/wiki.

Metz, Joachim (September 18, 2016). *Getting started - log2timeline/plaso Wiki.* Retrieved from GitHub: https://github.com/log2timeline/plaso/wiki.

Guðjónsson, Kristinn (June 29, 2010) *Mastering the Super Timeline With log2timeline.* Retrieved from: https://www.sans.org/reading-room/whitepapers/logging/mastering-super-timeline-log2timeline_33438.

# Appendix A. Installed packages

Python version 2.7.14
(https://www.python.org/ftp/python/2.7.14/python-2.7.14.amd64.msi)

VMware Workstation player version 14.1.1
(https://www.vmware.com/go/tryplayer)

VirtualBox version 5.2.8
(https://www.virtualbox.org/wiki/Downloads)

Forensic Acquisition Utilities (FAU)
(http://www.gmgsystemsinc.info/fau/)

FTK Imager version 4.2.0
FTK Imager Lite version 3.1.1
(https://accessdata.com/product-download)

Encase Forensic Imager version 7.10
(https://www.guidancesoftware.com/encase-forensic-imager/forensic-imager-download)

Mandiant Redline version 1.20.1
(https://www.fireeye.com/services/freeware/redline.html)

TSK and Autopsy version 4.6.0
(http://sleuthkit.org/sleuthkit/download.php)
(http://sleuthkit.org/autopsy/download.php)

SIFT appliance version 3.0
(http://digital-forensics.sans.org/community/downloads)

SIFT-CLI command line tool version v1.5.2
(http://digital-forensics.sans.org/community/downloads)

Volatility version 2.6
(http://www.volatilityfoundation.org/releases)

Author Name, email@address:Ferenc Kovacs

# Appendix B. Evidence acquisition terminology

The very first step of the forensic investigation is to obtain a reliable copy of the evidence that will be analyzed. There are different forms of the artifacts of computer systems that can be collected as evidence. The hard drive (HDD) of the computer can hold the operating system files, the installed applications, the log files that have been collected during operation of the computer and the data files that are stored by the user. To preserve the integrity of the evidence, the investigator has to create an exact copy of the disk and has to be able to prove that the copy is identical. The other important artifact is the memory (RAM) image that holds volatile information regarding the running processes. There are different ways of acquiring these pieces of evidence. The easiest is when the forensic examiner has access to the physical machine. The machine can be running or be switched off. If the machine is off, the hard disk can be removed from the box and the analyst can create a forensic image (Vandeven, 2014). This method is referred to as *dead imaging*. In this case, the volatile memory can no longer be acquired, as it is lost. There is a possibility to recover the last content of the volatile memory in case the machine was hibernated. The hibernation file contains the last state of memory and it can be obtained from the HDD. In case the machine is malware infected it is advisable to keep it powered on but disconnected from the network to prevent all external communications.

*Live Imaging* is when the image file is created without powering down the source machine. The machine is kept running to preserve the volatile memory that can hold valuable information of running code. This method is also used when imaging disks of computers with full disk encryption. A user has to be logged on the machine to unlock the disk encryption. The memory and disk of the computer can be accessed through the network connection. Live imaging can be used for machines in remote locations or when the business requirements restrict the analyst from powering down the machine. To acquire the RAM and disk content the computer can be connected to an isolated network where only the target machine present and the forensic machine of the investigator. The machine cannot be powered down during the network change and has to be imaged as

quickly as possible to avoid the loss or change of RAM contents. Software running on the forensic machine can connect to the target machine and obtain the RAM and the disk over the network. There are cases when the investigator cannot travel to the site where the target computer is located; this could be because the computer is in a remote location or there are multiple locations impacted with no local IT personnel available. In this scenario, the network speed to the remote location is the limiting factor for the amount data to be collected. The remote machine has to be connected to the network until the acquisition is finished. That may cause additional risk if the target machine is infected by malware.

A *forensic image* file is created when a physical disk is copied bit-by-bit into a single file. The resulting image file contains a duplicate copy of the source disk (Carrier, 2005, Chapter 3). The forensic images can be made without compression or can be compressed to conserve storage space. They can be separated into multiple files called *split images*. There is metadata such as a timestamp when the image was created and a cryptographic hash providing a fingerprint for the image file. This metadata can be embedded in the image file or stored separately.

There are several different forensic image formats evolved over time. One is the *raw image* format that contains data of the source only without metadata information. The file extensions '.raw', '.dd', and '.img' are the most common for raw images. The metadata may be stored in a separate file mostly with extension '.txt'. The commercial forensic suite EnCase introduced a proprietary format 'EnCase Evidence File' using file extension '.E01" that became de facto standard. E01 files are storing the metadata of the image files in their header and footer. The metadata contains EnCase version, operating system of source disk, timestamps Cyclical Redundancy Check (CRC) and cryptographic hashes of the data. There is a new format the '.Ex01'introduced in Encase 7. It has a different compression method, and the CRCs are written to the end of the file instead of after each data block.