Linux Forensic Triage: Overview of Process and Tools

A. Anđelković[1], K. Hausknecht[1], G. Sirovatka[2]
[1] INsig2 d.o.o., Zagreb, Croatia
[2] Zagreb University of Applied Sciences, Zagreb, Croatia
Anja.Andjelkovic@insig2.com, Kresimir.Hausknecht@insig2.com, Goran.Sirovatka@tvz.hr

***Abstract* - Digital forensics dates back into the 1980s, but the importance of Linux forensics was not taken into place until recently. Linux forensics is a distinctive world compared to example Microsoft Windows forensics. Although it is commonly used as a name for the entire operating system, Linux is just the name of the kernel, a piece of software that handles interactions between the hardware and end-user applications. Its popularity has not reached the popularity of the Windows operating system, therefore, without many reliable tools on the market, it represents a bigger challenge for digital forensics investigators.**

Digital triage is the process in which an investigator collects, assembles, analyzes, and prioritizes digital evidence from a crime. Since there are not many available tools on the market for performing Linux triage, the most important part is to understand the tool and its capabilities in order to know which one to use for a certain situation.

This paper will describe how the Linux system is structured, what its architecture contains, how should one correctly approach and acquire the system, and how to understand the tools and results they provide.

Keywords – Linux forensics; digital triage; Linux architecture

I. INTRODUCTION

Digital forensic examiners use various scripts and tools that are available on the market for performing Linux forensic triage. The usage of the Linux system is constantly growing which increases the need for Linux forensic investigators, as well as the creation of proper tools. Throughout this paper, the value of manually acquiring Linux artifacts will be shown, as well as how to understand the tools and scripts that are available on the market. Before starting to use any tool, one must understand how Linux is structured in order to know where to search for useful data and to understand the output that the tool provides.

The paper is divided into 7 sections. Section 2 and 3 provide a brief overview of the Linux system and its architecture. Within section 4, the introduction to digital forensic triage is given, along with the different types and characteristics of forensic triage tools. Section 5 describes important files for Linux triage and commands for a manual search for data of forensic value, while section 6 provides detailed information on a few scripts and tools for Linux forensic triage and memory capture. Section 7 concludes the paper.

II. LINUX OVERVIEW

Linux is a family of open-source operating systems based on the Linux kernel. It was released on September 17, 1991, by the Finnish software engineer, Linus Torvalds. The idea for the creation of Linux derived from and was based on the UNIX operating system [1].

It became common to use the naming 'Linux' when referring to the entire operating system, but in fact, Linux is the name of the kernel. A kernel is a core of the computer system that loads first and manages interactions between the hardware and applications. The correct expression when wanting to refer to the entire operating system would be 'Linux distribution' or 'Linux flavor'. This part is built on top of the Linux kernel and, depending on the distribution, it can include already installed applications or have the barest minimum [2].

Different Linux distributions offer different ways of being configured. Adjusting the system for personal needs is much more flexible than on other operating systems, but also requires higher technical knowledge. This is why Linux is often thought of as being an operating system for coders and programmers. However, over the years, Linux became more appealing to general users and therefore its usage increased [3].

As mentioned, the main difference among different Linux distributions is the kernel on which the system is built on. Almost six hundred Linux distributions exist nowadays, and the number will most likely keep increasing. Linux can be configured for different kinds of interests and needs. It can also be compiled and assembled from scratch. Some distributions include numerous pre-installed basic applications, while others have the simplest minimum. There are also distributions created for a specific purpose, for example, Digital Forensic and Forensics Toolkit (DEFT) distribution which is created for conducting digital forensic investigations with various pre-installed forensic tools. Some of the most popular widely used distributions are Debian, Fedora, openSUSE, Gentoo, Linux Mint, Ubuntu, and CentOS [3].

Its reliability evolved from being more resistant to viruses, crashes, slow operation, and expensive repairs than other operating systems. Therefore, most of the

Internet, all of the world's top supercomputers, and the world's stock exchanges are run on the Linux system. Also, some of the most popular platforms, like Android and iOS, are established on the Linux operating system [4].

III. LINUX ARCHITECTURE

Linux system architecture contains four layers, as it is shown in Figure 1. The central layer consists of all the physical devices that are connected to the system. The central layer is called hardware and is contained of a hard disk drive, Random Access Memory, Central Processing Unit, motherboard, etc. Secondly goes the Kernel, which is the core component that manages interactions with the layers above and below it. It is responsible for all important activities. The last two layers are Shell and Applications. Shell is the bridge between the user and the Kernel. It takes inputs from users and forwards them to the upper layer. Likewise, it takes outputs from Kernel and sends results back to the Shell, for users to be seen. The last layer, Applications, is the closest one to users. Applications utilize programs that run on Shell and provide the user with the functionalities of the system [5].

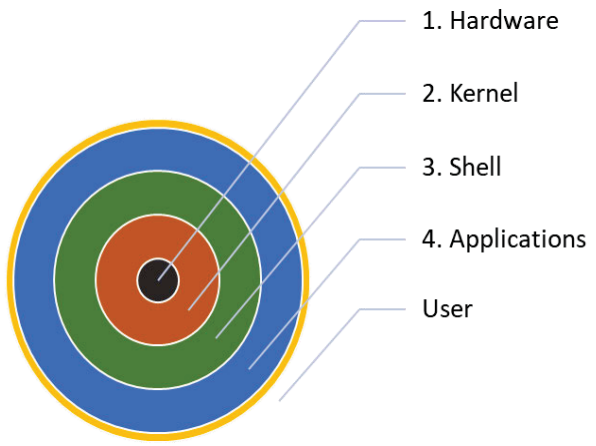

Figure 1 Linux architecture overview

A. *Linux structure*

The Linux File Hierarchy Structure or the Filesystem Hierarchy Standard (FHS) defines the structure and content of the directories in Unix-like operating systems. The FHS project started in 1993, and the purpose was to agree on how directories should be systematized, and which files should be stored where so that distributions could have a single reference point from which to work [6].

In Linux, everything starts from the *root* directory, which can be seen in Figure 2. Root directory would be equivalent to the Microsoft Windows C:\ folder. Every file and directory start from there and only root users can write under this directory. Root users are the ones with the administrative privileges, such as changing the ownership over the files. In other words, root users have the highest level of access rights on the system [7].

Some of the most common top-level directories, which can contain forensically sound evidence are listed and described in Table 1.

Table 1 Most common top-level directories

Directory	Description
/bin	Essential system binaries
/etc	System configuration files
/home	User directories
/opt	Optional or third-party application
/tmp	Temporary files (cleared on reboot)
/usr	User related software
/var	Variable data, log files

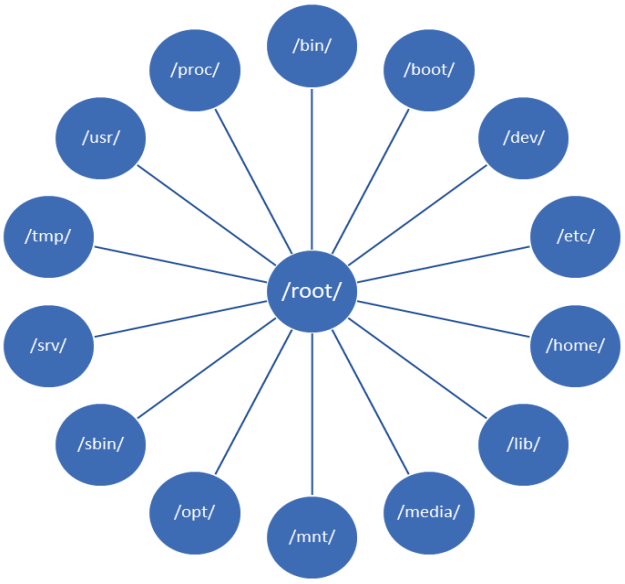

Figure 2 Linux directory structure

B. *Journaling file system*

To investigate the Linux system, a forensic investigator should know the system's file structure. Being able to understand how a certain file system handles files can be helpful when recovering evidence such as deleted files and file activity.

Most of the Linux systems use a journaling file system that keeps track of system changes which are still not committed to the file system. Recorded changes are written down in a data structure, known as "journal". In case a power failure occurs and the system unexpectedly shuts down, such file systems can be brought back more quickly with less chance of becoming corrupted [8].

The most commonly used journaling file system for Linux is the third extended filesystem (ext3 or ext3fs), which was added in 2020 to the kernel from version 2.4.15. The newest journaling file system for Linux is ext4 (fourth extended filesystem). It is the default file system for many Linux distributions and can support volumes with sizes up to one exbibyte [9].

IV. DIGITAL FORENSIC TRIAGE

The term "triage" comes from the field of medicine, in which medical personnel determine the priority of patients due to their condition. Triage in digital forensics can be the crucial step as its purpose is to quickly identify data that are likely to contain the evidential value. During the

process of digital forensic triage, investigators collect, extract, and analyze the acquired digital evidence. By using triage, investigators can decide not to perform a full disk image but only capture what is necessary to prove or disprove investigation goals [10]. Though the purpose of triage is to quickly identify data of evidential value, these tools have certain limitations. Issues that could occur when working with triage tools are that they will not bypass the encryption or passwords and most of them need administrator privileges [11].

The digital triage comes in two forms, based on the type of data that is being investigated and methods used – live triage and post-mortem triage. Traditional forensic investigations were focused on analyzing data from a powered-off computer. Investigators would take the disk from a powered-off computer, create its duplicate and start the analysis. Since certain data gets lost when the device gets powered-off, investigators started to analyze the live system. When dealing with the live computer, running processes, notifications, recent email messages, recently visited web sites, passwords, etc. can be captured [11].

A. *Forensic triage tools characteristics*

Not every tool is appropriate for every situation, nor will every tool produce all of the needed results. The most important part is to understand the tool and its capabilities in order to understand the results they produce. Triage tools and tools' updates must be validated before being used. Tool validation is performed in order to preserve the integrity of evidence and to state the correctness of the results. Tools should not modify data in any way. Investigators must not only know what the tools output, but also how they do it. That way they will also be able to describe and demonstrate how these tools are relevant to their investigation and how they changed the inspected system without setting their investigation in danger. The precondition that every triage tool should have is the capability to collect data in a reasonably short time without changing the data [12].

Tools can be divided into three categories – open-source, free, and commercial tools. Open-source tools have a freely available program code and support is based on the community. Commercial tools offer plenty of options and customizations, and have excellent support, but are usually extremely expensive. Free tools are a subcategory of commercial tools. They are made free to use by a company that is providing support for them. In comparison to standard commercial tools, free tools do not have full functionality or they are free to use for a certain period of time As Linux is a family of open-source operating systems, so are most of the tools used for them [12].

All tools and scripts that will be presented throughout this paper are open-source and free for download.

V. LINUX TRIAGE COMMANDS AND FILES

Tools used for Linux triage are usually built out of Linux commands. Investigators should know these commands to understand how these tools perform the work and collect information. It is not always enough just to run the tool and wait for it to produce gathered information. In the following chapters, some of these commands will be mentioned, along with their importance for the investigation.

A. *Gathering user account information*

When analyzing the system, volatile data should be captured first. Volatile data is the type of data that would be lost if the system were powered off. This type of data is stored in a computer's temporary memory while it is running. Computer's memory is investigated since it can lead to identifying attacks and malicious behaviors, gather information on recently performed actions (visited web sites, received/sent email messages, etc.), bypass the encryption, collect password information, and many more.

Collection of the user account information is of great importance while performing forensic investigation. Best way to gather information from the Linux system is by using the terminal in which the commands are inputted. The command "w", when typed into the terminal, will display users that are logged on the system. This command will also display what is the user doing on the system and when the user logged into the system [13].

Besides these commands, there are important files under the /etc directory that should be investigated. All this information could be helpful when building an event timeline. The /etc directory contains application's configuration files, startup, shutdown, start and stop script for every individual program. A configuration file is a local file used to manage the operation of a program – it must be static and cannot be an executable file [14]. Inside this directory are many files of interest for digital forensic investigators when performing Linux triage. System configuration, like hostname, time zone, operating system information, etc., could be found. For reading each of these files, the command "cat", along with the file name, is used. Knowing the hostname can be useful in the further investigation to connect other information with the specific action or user. Hostname file can be opened from the terminal with the command "cat /etc/hostname" [15]. To adjust the event timeline, useful information is to know in which time zone is the system set. This information can be found with the command "cat /etc/timezone" [15].

Another two important files are /etc/passwd and /etc/shadow. These files contain information on users and their passwords. The /etc/passwd includes basic information about each user account on the system, including the root user who has full administrative rights, system service accounts, and actual users. [16].

The /etc/shadow file stores the actual password in an encrypted format for the user's account with the additional properties related to the user's password. In other words, it stores secure user account information. [17].

B. Network information

Part of digital triage is to check the network information. Network information includes the local network information, a list of ports opened to the outside that could be used to compromise the network, related programs to those ports, and many more.

In order to check the IP address that is assigned to the system, "ifconfig -a" command is used. This command lists out the number of received and transmitted packets and assigned IP address [18].

During this process, it is also important to review each of the network connections and open sockets. This information can be gathered with the "netstat" command. To use this command, the user must be in root mode in order to receive the information. This command will list out all the active internet connections which can be very useful for the investigators. There are additional options with this command. For example, with the "netstat -tulpn" command, only servers will be listed, but when using the "netstat -natp" command, established connections will also be listed. It will give out the protocol that is used, the number of received and sent packets, local and foreign address, the state, PID (Process Identifier) number and program name (e.g. Firefox) [18].

C. Running programs and processes

Running programs and processes can point to a certain activity that can prove or dispute certain actions. The command "top" will show all running processes, while command "ps -aux" will display all processes [19]. Listing out the processes can point to certain activities performed by the user or lead to the suspicious activity running in the background. To check for opened files on a Linux system, the "lsof" command will provide information on it. In its output, this command returns information about files that are opened by processes. Usage of this command can be very useful in malware investigations since it can discover what was used by the user. This command can be modified with different options to obtain a different output. To display IPv4 or IPv6 files, either "lsof -i -4" or "lsof -i -6" commands are used. Also, it can be specified to display processes that are belonging to a certain user. For that, the username must be included in the command, in a way that "lsof -u [USERNAME]" command is used [20].

Most of Linux triage tools are based on these commands but can also be used to create own scripts and personalize them according to what needs to be acquired.

VI. LINUX TRIAGE SCRIPTS AND TOOLS

The main purpose of digital triage is to identify the most relevant artifacts. Two important characteristics that triage tools must have is that they are reliable and do the work fast. These requirements are imperative for building triage tools.

As mentioned before, useful information can often be found in the computer's Random Access Memory (RAM). Digital investigators can search for information simply by reviewing the readable text and performing keyword searches. Since physical memory in modern computers has increased and will continue to increase, such an approach is not efficient. That is why it is important to have specialized knowledge of data structure in memory and associated tools.

When talking about Linux triage, not many tools are available on the market, but those that are, showed justifying results. A great number of tools and scripts are open-source and available for download.

A. Acquire Volatile Memory for Linux (AVML)

When coming across a live computer, capturing memory is imperative. When talking about Linux, the traditional approach was to use "dd" and point to a device file such as /dev/mem or /dev/kmem. Modern Linux system restricted access to these virtual devices so the new approach was required. The current method and the most popular tool for capturing memory is Linux Memory Extractor (LiME), which is, in fact, Loadable Kernel Module (LKM). The problem with this method is it requires knowledge of the target distribution or kernel, which can sometimes present an issue. Also, it requires to be compiled for every kernel version separately. This is where Microsoft introduced its memory acquisition tool called AVML (Acquire Volatile Memory for Linux).

AVML tool is a volatile memory acquisition tool written in Rust programming language. Its advantage among other acquisition tools is its possibility to acquire memory without knowing the target operating system or even its kernel and does not require any additional libraries on the target machine. This is the issue with most other acquisition tools since they require a loadable kernel module to be built on the exact same kernel version as the target [21]. This tool also has certain limitations as it does not support every target distribution or kernel.

It is enough to build the tool once and place it on any storage device for further use. When wanting to capture a memory from a target machine, it can be done without any on-target compilation needed.

AVML acquires memory from three memory sources - /dev/crash, /proc/kcore, and /dev/mem. It uses the LiME output format, when not using optional compression.

B. IR Triage Toolkit script

IR Triage Toolkit is an open-source collection of scripts that can be used to create a toolkit for incident response and volatile data collection. It can be used for both Linux and Windows operating systems and downloaded for free from a GitHub webpage [22].

When the script is run, it dumps the content of the system's memory and outputs several other system's information in a text file. Additionally, it creates detailed timestamped logs of the given command and creates a SHA-256 checksum hash of the files saved to the storage directory [22].

After the toolkit is downloaded, it needs to be created in the desired directory. In order to do so, the bash script "create-toolkit" is started with the command "sudo bash create-toolkit", as it is shown in Figure 3. Following, a bash script named "run" will be generated inside the chosen directory. When running the "run" script, volatile data will be saved and ready for further analysis.

When looking at the background of the toolkit by opening its bash file, it is visible what the toolkit does. The toolkit first acquires a full memory dump by using the integrated LiME tool. Afterward, it collects various system information and outputs it into separate text files.

The output files contain information on the network, opened files, running processes, users, system, and mounted devices. At the end of the process, the toolkit creates SHA-256 checksum for all files.

Gathering information with IR Triage Toolkit is very fast with the well-structured output. It collects the main basic information on the system and users, and the bash script can be easily modified and adjusted for specific needs. The script is based on numerous Linux commands, some of them which were mentioned in the chapter before. The tool is small in size when downloaded but requires larger storage capacity for dumping the content – depending on the memory size of the target device.

```
@ubuntu:~/ir-toolkit$ sudo bash create-toolkit
Directory to save toolkit: ./create-toolkit /tool
```

Figure 3 Command for creating IR Triage Toolkit

C. *FastIR Collector Linux*

The tool FastIR Collector Linux is a script based on Linux commands which collects various artifacts on the live Linux system and can be downloaded for free from a GitHub webpage. There is also a separate script created for collecting artifacts from Windows operating systems. Some of the artifacts it collects are system information, last logins, connections, handles, user's data, and many more. Collected results are outputted in .csv and .txt files [23].

The tool is written in Python programming language and must be run as a root user with a "python" command, as it is shown in Figure 4. It will output the mentioned information in a separate directory.

Information collection is performed and outputted within seconds. This tool also collects information from /etc/passwd and /etc/shadow files. Its disadvantage, in comparison to IR Triage Toolkit, is that it does not dump the system's memory, and is harder to adjust the script's code for personal needs.

```
@ubuntu:~/Fastir$ sudo python fastIR.py
```

Figure 4 Running FastIR Collector python script

D. *IR Rescue*

IR Rescue consists of two scripts that collect numerous forensic data from both 32-bit and 64-bit Windows systems and Unix systems. The scripts are intended for incident response and forensic investigators for use at various stages in the analysis and investigation process. [24].

The script also comes with an executable for dumping memory, called "linpmem-2.1.post4" [24].

The file is composed of binary directives – true or false. By choosing between true and false options, data types to collect and advanced tools for running will or will not be performed. The configuration file can be modified in the way it is needed for a certain purpose or situation. For example, a memory dump is set to "false" by default, and therefore will not be performed unless otherwise is not defined. This may make a procedure a bit more complicated but also gives the examiners different possibilities to adjust the script for specific needs. Examples of setting options to "true" or "false" are shown in Figures 5 and 6. For example, if the web browser's data is needed to be collected, it will be set to state "true", as shown in Figure 5. On the other hand, if a complete memory dump is not needed, it can be set to false, as seen in Figure 6. After the configuration file is modified and the script is run, outputted data will be structured in directories, in either a textual (.txt) format or as a database (.sqlite). Further on, .sqlite files, which are created for web browsers, can be opened with the command "sqlitebrowser [FILENAME]".

```
# web
web-all=true
        # collect browser information
        # "mozilla", "firefox"
web-browers=true
```

Figure 5 Collecting web browser information with IR Rescue

```
# memory
memory-all=false
        # dump the RAM to AFF4 format
        # Pmem "linpmem-2.1.post4" (64-bit ELF)
memory-dump=false
```

Figure 6 Turning off memory dump creation in IR Rescue script

E. *Live Response Tool Collection*

Live Response Tool Collection is a set of tools for collecting data from a live system. It covers Windows, OS X, and Unix/Linux operating systems. It is created in a way that a non-technical person can easily use it. The built-in triage option collects volatile data, using a variety of scripts. The script saves data to a directory with the computer name and timestamp under the directory from which the script was run. It also generates text files containing the MD5 and SHA-256 hashes of every collected file and the full path to that file. There is also a log text file that contains each command that was run by the script and, if present, any error message from running a certain command.

The script should be run under administrative privileges as it allows larger data collection. When running the Linux bash script, the tool collects information and stores it under five subdirectories. Information is collected very fast and outputted in a well-

structured way, where each file is automatically named according to the content it contains.

"BasicInfo" subdirectory contains alternate data streams, last activity view, running processes, list of hidden files, etc. Under "NetworkInfo" subdirectory information about ARP tables, allowed/denied hosts, routing tables, Internet settings, and socket statistics are stored. "PersistanceMechanisms" subdirectory contains information related to persistence mechanisms on the system, including autoruns, loaded drives, and scheduled tasks. "UserInfo" subdirectory includes user's information, like logins, a listing of the users, and information on the current user [25].

The bash script can be adjusted for specific needs and purposes, and it is well-documented, giving the information on what each section does and where it is outputted.

VII. Conclusion

The consumption of the Linux operating system is expanding, especially among coders and programmers, but, unfortunately, it is also widely used for performing criminal activities. In order to investigate activities on a Linux system, forensic investigators must understand the system's architecture and file structure.

When approaching a live computer, it is imperative to capture volatile data. Once the system is powered off, volatile data becomes irretrievable. Linux distributions may differ in the kernel version on which they are built, making the memory capture harder since the tool must be built on the same kernel version. Microsoft company presented a tool for capturing the memory of a Linux system, called AVML. This tool distinct from others as it can acquire memory without knowing the kernel version, making the whole process easier.

Throughout this paper, various commands were described. Tools and scripts that are being used for performing Linux triage are based on these commands. Since there are various tools and scripts available on the market, it is not enough to just run them but to understand how they work and the results they produce. Few free-for-download scripts were described throughout this paper. Their support is mainly based on the community, making the tool validation process even more important. The best practice is to use multiple different tools for the same action, as not every tool will produce the same amount of results, nor will one tool be appropriate for every situation. Sometimes the used tool can only be an investigator's matter of preference. As can be seen in Section VI., all of these scripts are easy for use. This is one more reason for forensic investigators to be familiar with the Linux commands and file structure so that they can adjust the existing scripts as they need them for a particular situation. Knowing the file structure leads to knowing how to navigate yourself through files that could have forensic value.

References

[1] "Linux", (2019, December), Retrieved from: https://en.wikipedia.org/wiki/Linux

[2] "What is Linux", (2019, December), Retrieved from: https://opensource.com/resources/linux

[3] A. Williams and B. Turner, "Best Linux distros of 2019: for beginners and advanced users", (2019, December), Retrieved from: https://www.techradar.com/best/best-linux-distros

[4] "What is Linux", (2019, December), Retrieved from: https://www.linux.com/what-is-linux/

[5] "Linux Architecture", (2019, December), Retrieved from: https://tecadmin.net/tutorial/linux/linux-architecture

[6] "Filesystem Hierarchy Standard", (2019, December), Retrieved from: https://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard

[7] "Root User", (2020, January), Retrieved from: https://www.ssh.com/iam/user/root

[8] Sayak Boral, "What is a Journaling File System?", (2019, December), Retrieved from: https://www.maketecheasier.com/journaling-in-file-system

[9] Ramesh Natarajan, "Linux File Systems: Ext2 vs Ext3 vs Ext4", (2019, December), Retrieved from: https://www.thegeekstuff.com/2011/05/ext2-ext3-ext4/

[10] V. Jusas, E. Gahramanov, and D. Birvinsksas, "Methods and Tools of Digital Triage in Forensic Context: Survey and Future Directions", (2020, January), Retrieved from: https://www.researchgate.net/publication/315928408_Methods_and_Tools_of_Digital_Triage_in_Forensic_Context_Survey_and_Future_Directions

[11] INsig2, "Basic digital forensics for IT professionals", Retrieved from: https://insig2-and-zyberglobal.learnworlds.com/author/course?courseid=dfir-it

[12] INsig2, "Fundamentals of digital forensics for lawyers and judges", Retrieved from: https://insig2-and-zyberglobal.learnworlds.com/author/course?courseid=fundamentals-of-dfir

[13] Aaron Kili, "11 ways to find user account info and login details in Linux", (2020, January), Retrieved from: https://www.tecmint.com/find-user-account-info-and-login-details-in-linux/

[14] Linux Foundation, "Chapter 3. The Root Filesystem", (2019, December), Retrieved from: https://refspecs.linuxfoundation.org/FHS_3.0/fhs/ch03s07.html

[15] Tho Le, "Linux Forensics – Some useful artifacts", (2020, January), Retrieved from: https://medium.com/@tho.le/linux-forensics-some-useful-artifacts-74497dca1ab2

[16] V. Gite, "Understanding /etc/passwd File Format", (2019, December) Retrieved from: https://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/

[17] V. Gite, "Understanding /etc/shadow file", (2019, December), Retrieved from: https://www.cyberciti.biz/faq/understanding-etcshadow-file/

[18] Chandan Kumar, "10 Useful Linux Networking Commands", (2020, January), Retrieved from: https://geekflare.com/linux-networking-commands/

[19] Chris Hoffman, "How to manage processes from the Linux terminal", (2020, January), Retrieved from: https://www.howtogeek.com/107217/how-to-manage-processes-from-the-linux-terminal-10-commands-you-need-to-know/

[20] Narad Shrestha, "10 losf command examples in Linux", (2020, January), Retrieved from: https://www.tecmint.com/10-lsof-command-examples-in-linux/

[21] Microsoft, "AVML (Acquire Volatile Memory for Linux)", (2020, January), Retrieved from: https://github.com/microsoft/avml

[22] R. Shipp, "ir-triage-toolkit", (2020, January), Retrieved from: https://github.com/rshipp/ir-triage-toolkit

[23] SekoiaLab, "FastIR Collector Linux", (2020, January), Retrieved from: https://github.com/SekoiaLab/Fastir_Collector_Linux

[24] D. Fernandes, "ir-rescue", (2020, January), Retrieved from: https://github.com/diogo-fernan/ir-rescue

[25] B. Moran, "Live Response Collection Overview", (2020, January), Retrieved from: https://www.slideshare.net/BriMorLabs/live-response-collection-overview