

How to improve efficiency of Linux Forensics?

MSc Research Project
MSc in Cyber Security

Athul Antony
Student ID: 20242239

School of Computing
National College of Ireland

Supervisor: Rohit Verma

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Athul Antony
Student ID: 20242239
Programme: MSc in Cyber Security **Year:** 2021-2022
Module: Research Project
Supervisor: Rohit Verma
Submission Due Date: 15-Aug-2022
Project Title: How to improve efficiency of Linux Forensics?
Word Count: 4156 **Page Count** 15

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ...Athul.....

Date:15/08/2022.....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

How to improve efficiency of Linux Forensics

Athul Antony

20242239

Abstract

The era of digital technology has been rising steadily at a rapid rate. The development of new technologies and software paves way to cybercrimes which has been growing exponentially in recent years. To understand cybercrimes and to prevent it from happening forensic investigation is carried out. Linux forensic triage is in which a forensic investigator collects, assembles, analyses the evidence obtained from Linux machine. Linux operating system provide more secure services than other operating machines and hence investigators need to comb through every piece of information. This is a time consuming and excruciating process. This paper focuses on improving the efficiency of Linux forensics by performing forensic investigation with remote acquisition and collect all relevant information efficiently in much shorter time. The paper shows the forensic investigation carried out by forensic tool and as well as remote acquisition to highlight the differences which helps to show how efficiency has been improved.

1 Introduction

Linux operating system (OS) is one of the popular operating system that is used in almost all types of digital devices. Android which is one of the most popular platforms is powered by Linux operating system. In same way a Linux workstation is considered to be powerful tool for forensic investigation due to their wide support of various files, various availability of advanced tools and ability to develop & compile source code. Linux is capable of supporting multi-tasking and multi-user operations. The chance of Linux OS being affected by viruses are low since Linux is an open-source platform (SearchDataCenter, 2000). Since Linux is being used in various platforms it is necessary that Linux forensic team is established. OS forensics is an art to find evidence, user activities for a specific question. Linux forensics focuses on machines that uses Linux as base operating system. Since there are various Linux distributions Linux forensic conducted for each distribution will be a bit different. Before performing Linux forensic on a Linux machine, it is necessary to understand the working behind the Linux operating system and all the various characteristics it has. Linux architecture is made up of four layers namely hardware, kernel, shell and applications. Kernel acts as intermediate between other layers and is the most important part of any Linux system. A forensic platform system will be set up and will be used as base to conduct forensic investigation on the victim machine. There are various tools and scripts which will help the forensic investigators to perform digital forensic investigation. Each tool will have certain limits, range and capabilities. It is necessary to understand which tools to be used in which situations. When conducting forensic investigation on Linux machine it is necessary collect

memory dumps from various location to clearly understand the activities that has been carried out in that machine. Linux forensic tools can be used to collect the memory dump, but it will be time consuming. If a cyber-attack has been happened in an organization the forensic investigators need to seize all the affected machines and perform forensic investigation. The time taken by the forensic tools in this case will be very high and at the same time the size of the memory dump will be large that it can cause performance issues in the system. Examiners need a solution that brings together all that data and automates some of the tedious tasks in acquisition and processing – freeing up time for deeper analysis. For this purpose and to avoid wastage of spaces for the output produced by forensic tools a remote acquisition tool can be used. A system called RAFT (Remote Acquisition Forensic Tool) was created to make it easier for forensic investigators to obtain digital evidence. A secure, verifiable client/server imaging architecture is used to accomplish this. The RAFT system is made to be quite simple to use and requires the user to have little technical expertise. One of the main goals of RAFT is to make sure that the data it collects remotely may be used in court. This is accomplished by confirming that the image captured via RAFT matches the original evidence on a questionable machine. This remote acquisition tool will help in conducting forensic investigation on victim machine from different part of the world. And only the information that is needed by the forensic investigator can be acquired rather than collecting huge memory dumps to go through. This will not only help in saving the space but also will risk the Linux system from getting crashed during forensic investigation. This kind of remote acquisition will help in increasing the efficiency in which Linux forensics is carried out.

2 Related Work

To understand the concept of Linux forensics and remote acquisition on Linux forensics four papers are chosen as literature review. These research papers will help in understand the need for the research question. The following subsection will give a brief overview of the four chosen research papers.

2.1 Literature Review

The Linux forensic triage (Adelkovic, Hausknecht, and Sirovatka) research paper focuses on various process available in Linux and as well as various tools available to perform Linux Forensics. Linux system architecture is made up of 4 layers which are Hardware, Kernel, Shell, Application. Hardware is the center layer and consists of all physical devices connected to the system. Second layer is the Kernel which is the core component and is responsible for managing the interactions that happen between above & below layers. Third is the Shell layer which acts as bridge between user and kernel. Shell taken input from user & sends to Kernel likewise takes output from Kernel & sends to user. The last is the Application layer which provides users with functionalities. The research paper also focuses on various forensic tools that can be used to perform Linux forensic triage. Acquire Volatile Memory for Linux (AVML) can be used to collect memory details of Linux system without the need for knowing the kernel version. This tool is written in rust programming language. IR Triage

toolkit script collects various information from various sources and dumps the output in a specified location. FastIR collector tool is used to collect information from /etc/shadow and /etc/passwd files, and it doesn't collect memory information. This tool is written in python programming language. Live Response tool can be used to collect information according to the user and output will be given in various location according to the information obtained.

A study of Linux Forensics (Gustavo, Keanu, and Munn) discusses about various open source and closed source tools that can be used to perform Linux forensics. To provide better result and credible evidence it is best to use open-source tools to perform Linux forensics. The main things that need to be checked while performing forensics on Linux system are Auditing, logging, and file system journal. These are the main sources from where various information and activities carried out in the Linux machine can be obtained. Linux's audit system was created to aid in user security and make it possible to audit and record important data. Logging is the process of compiling records of every action taken within a Linux system. They include details on the system events that have taken place as well as details about the users that initiated the events. The investigators will be able to learn what actions have been taken on that specific machine with the use of both auditing and logging. Linux employs the file system journal to cache data that is stored on disk to guarantee that nothing is lost in the event of a system fault. Data files that have been deleted or overwritten can be analysed and recovered using the file system journal.

The State-of-the-art tools and techniques for remote digital forensic investigations (Uma, Shobana) discusses about the various methodologies that is currently available for the remote acquisition in the digital forensics world. For carrying out different stages of research, several cutting-edge software and hardware tools and approaches are contrasted. Comparison charts are shown to help readers comprehend the benefits, drawbacks, obstacles, and possibilities associated with certain strategies. The overall goal of this paper is to conduct comparative analysis based on qualitative outputs observed from memory, timeline, and live forensics imaging on an incident, which can streamline the process of determining the most appropriate technique under different circumstances for an efficient remote forensic investigation.

Enabling the remote acquisition of digital forensic evidence through secure data transmission and verification (Mark) provides an overall approach to help the law enforcement officer to collect and conduct forensic investigation directly from the victim system without being physically present with the help of remote acquisition forensic tools. The practicality of a digital forensic evidence acquisition tool that can be utilized by any law enforcement officer was proposed in this thesis, and it was also validated. As soon as possible during the inquiry, the tool will have the evidence in the forensic laboratory in an "investigation-ready" state. According to the conventional methodology for gathering digital evidence, the digital investigator must go from the forensic lab and go to the crime scene to gather the suspect machines. These devices might then spend a considerable amount of time unattended (and unimaged) in a storage facility for evidence. Information that could be crucial to the case may go undetected during this time. This conventional model can be greatly enhanced by the usage of the RAFT system.

2.2 Literature review summary

Table 1: Literature review matrix

Related Works	Strengths	Result Format
(Adelkovic et al., 2020)	Various tools used in Linux forensics	AVML
(Gustavo et al., 2018)	Complete study of Linux forensics	Areas needed to check for forensic investigation
(Uma et al., 2021)	Different methodologies for remote acquisition are analysed	Method for remote acquisition
(Mark., 2018)	Remote acquisition tools and approaches in forensics	Limits of tool and analysis of remote acquisition tools

3 Research Methodology

Linux forensic analysis involves analysing forensic images obtained after post-mortem forensic investigation that have been abused, misused, or has been the target of malicious attacks. It is necessary to understand how to locate and interpret digital evidence that are obtained from Linux machines. After analysing it is evident to draw logical conclusions and reconstruct past timeline activities or security incidents. The main things while conducting Linux forensics are to perform analysis of partition tables, volume management, Linux filesystems, directory layout, reconstruction of Linux start-up process, initialization of kernel & boot system, analysis of time & local settings, geolocation services available within Linux, reconstruction of user login sessions, analysis of desktop artifacts, identification of external devices.

To show how the efficiency has improved for Linux forensics the forensic analysis will be conducted on Linux machine using two different methods. The first method will be conducting forensic analysis by using already available open-source Linux forensic tools. The second method will be conducting forensic analysis by using data sets that will perform the necessary forensic actions without manual interactions. The analysis obtained from the two methods will be compared. This will help in showing how efficiency for Linux forensics has improved. Figure 1 shows the overall flow chart of the project.

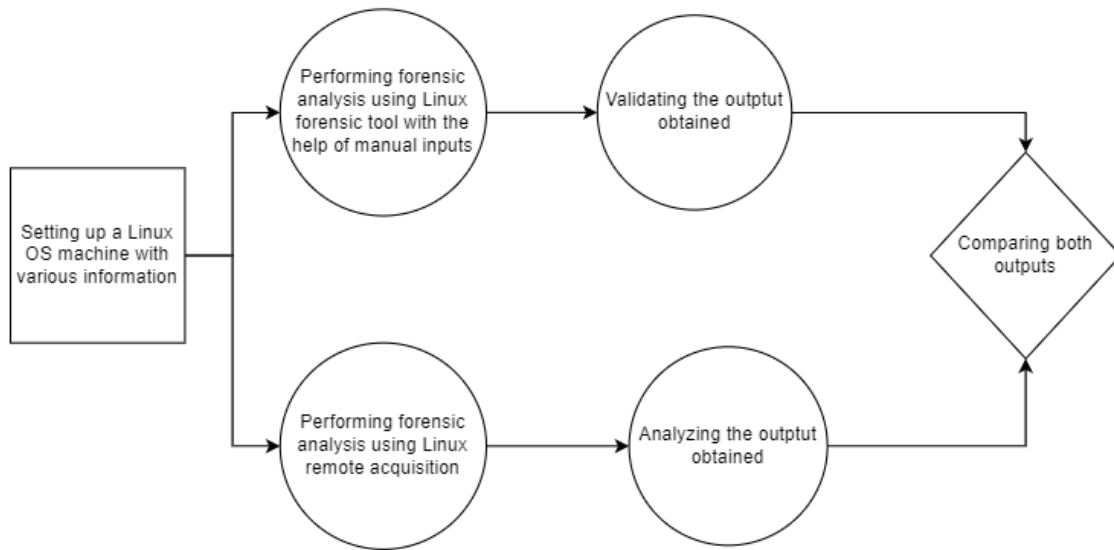


Figure 1: Flow Chart of the project

A Linux system will be set up to conduct forensic analysis. The Linux OS machine will be set up with the help of Oracle VM virtual box rather than using a separate system. Setting up Linux OS inside virtual box will provide with various advantages and in case of any issues everything can be set back to default without causing any damage. This will help in performing forensic analysis multiple times without the risk of any damage or corruption to file since it is possible to revert to the default state of the Linux machine anytime needed. Setting up a Linux system separately and performing forensic analysis can or may lead to corruption of files. Hence to avoid this dilemma a Linux OS machine will be setup inside the Oracle VM virtual box. The machine that will be used as platform to perform forensic analysis will be Ubuntu 18.04 system. Ubuntu 18.04 is a free, customizable, and highly usable alternative to both macOS and Windows. Various files, images will be added to the kali Linux and as well as various network activities will be carried out. This is to understand that while conducting forensic analysis how many of this information can be recovered.

4 Design Specification

The forensic investigation on the Ubuntu 18.04 is conducted by two methods using forensic tool and remote acquisition. They will be used to get as much information possible from the machine for the forensic investigation.

The forensic tool that will be used to perform Linux forensic analysis on Ubuntu system for the first method will be Acquire Volatile Memory for Linux (AVML) introduced by Microsoft. AVML is a memory acquisition tool that is written in Rust programming language. The main advantage of using AVML is that AVML can acquire volatile memory without the need of knowing OS or kernel distribution and will not require additional libraries in the target machine (Son, 2019). Three memory sources `/dev/crash`, `/proc/kcore`, and `/dev/mem` will be used by the AVML to acquire memory. If these memory sources are not specified AVML will iterate over to find functional sources. It uses the LiME output

format, when not using optional compression. AVML allows to save recorded image to external location and has features like page level compression & automatic retry.

The remote acquisition tool that will be used is the GRR. Google Rapid Response is a framework for incident response that emphasizes remote live forensics. GRR aims to facilitate investigations and forensics in a swift, scalable manner so that analysts may quickly triage threats and carry out analysis remotely (Sindhuja, 2021). GRR consists of 2 parts: client and server. On systems that one would want to investigate, GRR client is deployed. Once deployed on each of these systems, the GRR client frequently queries the GRR frontend servers for tasks. "Work" refers to performing a specified task, such as downloading a file or listing a directory. The frontends, workers, and UI servers that make up the GRR server infrastructure provide a web-based graphical user interface and an API endpoint that let analysts schedule client actions, view, and process collected data.

5 Implementation

For the research question the implementation is done via two methods. The forensic analysis on Ubuntu machine using Linux forensic tool AVML and using remote acquisition will be carried out in this section. Some files, images will be downloaded into the Linux machine as well as various network functions will be carried out before conducting forensic analysis. All the files downloaded before the forensic analysis will be noted down.

The implementation steps involved for the forensic investigation using forensic tool is as follows:

- Downloading the AVML tool from GitHub and running the tool to create memory dump.
- Cloning volatility into the Ubuntu machine and compiling the Makefile in volatility folder to create module.dwarf file.
- A zip file containing the newly created module.dwarf file and system debug symbols will be created. This will act as the volatility profile.
- Placing the zip file into the volatility folder so that volatility will be able to access it.
- Using python to feed memory dump into the volatility profile so that memory dump can be analysed.
- Required information can be obtained by modifying commands and using python to feed memory dump into volatility profile.

The implementation steps involved for the forensic investigation using remote acquisition is as follows:

- GRR server will be set up on the machine which will act as forensic platform.
- Before installing GRR MySQL needs to be installed and grr database needs to be created in MySQL.
- Once database is set grr can be installed directly from the official site with the help of terminal.

- While installing GRR it will ask options to set localhost, port details, admin credentials for the GRR server, hostname, or IP address of the machine to be used as GRR server.
- GRR server can be launched by entering IP address and port details into the web browser. To login admin credentials need to be given.
- In the GRR server launch page we can download the GRR client for Linux from the Binaries section. This then can be set up in the machine where forensic investigation needs to be conducted.
- Once client is set up GRR server page will show the client details and we can get relevant information easily.

5.1 Implementation of forensic tool

After the volatility profile is created the memory dump is fed into the volatility profile to be analysed. Using python we will be able to get the needed result from the memory dump. By using “lsdf” we will be able to see the list of opened files that has been in the ubuntu system. This is shown in Figure 2. By using “grep” we will be able to get details of particular applications. Figure 3 shows result of using Firefox with “grep” command.

```

Offset          Name          Pid      FD      Path
-----
0xfffffa0d41b1b9780 systemd      1        0      /dev/null
0xfffffa0d41b1b9780 systemd      1        1      /dev/null
0xfffffa0d41b1b9780 systemd      1        2      /dev/null
0xfffffa0d41b1b9780 systemd      1        3      /dev/kmsg
0xfffffa0d41b1b9780 systemd      1        4      anon_inode:
[10394]
0xfffffa0d41b1b9780 systemd      1        5      anon_inode:
[10394]
0xfffffa0d41b1b9780 systemd      1        6      anon_inode:
[10394]
0xfffffa0d41b1b9780 systemd      1        7      /sys/fs/cgr
oup/unified
0xfffffa0d41b1b9780 systemd      1        8      anon_inode:
[10394]
0xfffffa0d41b1b9780 systemd      1        9      socket:[145
68]
0xfffffa0d41b1b9780 systemd      1        10     anon_inode:
[10394]
0xfffffa0d41b1b9780 systemd      1        11     /proc/1/mou
ntinfo
0xfffffa0d41b1b9780 systemd      1        12     anon_inode:
[10394]
0xfffffa0d41b1b9780 systemd      1        13     /proc/swaps
0xfffffa0d41b1b9780 systemd      1        14     socket:[431
71]

```

Figure 2: Using lsdf to list all opened files

```

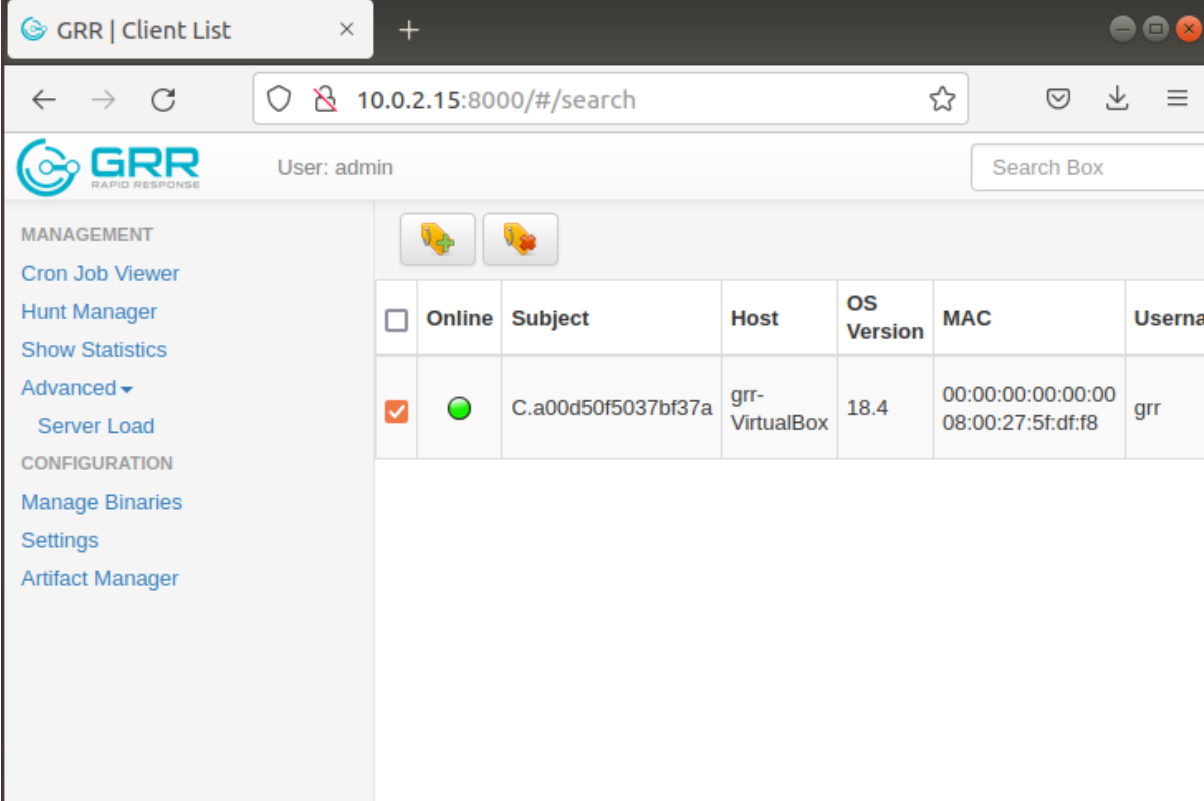
ubuntu@ubuntu-VirtualBox:~/Downloads/volatility$ python vol.py -f ../memory.dmp
--profile=Linuxubuntu_5_4_0-124-genericx64 linux_pslist | grep -i firefox
Volatility Foundation Volatility Framework 2.6.1
0xfffffa0d38c241780 firefox          1754          1          1000
          1000 0x0000000117b9a000 2022-08-10 14:26:35 UTC+0000
ubuntu@ubuntu-VirtualBox:~/Downloads/volatility$ █

```

Figure 3: Using grep command to get details of Firefox

5.2 Implementation of remote acquisition

The GRR server is hosted and GRR client is set to make connection with the server. Once connection is established we can see the details of the client machine. Basic details like MAC address, all the usernames on the client machine, version, OS details will be first visible as shown in Figure 4. In the left side of the GRR GUI we can see various management links. These will be used to get the required details from the GRR client or the targetted machine. It is also possible to run jobs on the client machine and see the result of it. Clicking on the client machine will give the overview of that particular client. This is shown in Figure 5.



The screenshot shows the GRR Client List interface in a web browser. The browser address bar shows the URL `10.0.2.15:8000/#/search`. The GRR logo and "User: admin" are visible at the top. A search box is present on the right. The left sidebar contains navigation links under "MANAGEMENT" and "CONFIGURATION". The main content area displays a table with the following data:

<input type="checkbox"/>	Online	Subject	Host	OS Version	MAC	Username
<input checked="" type="checkbox"/>	●	C.a00d50f5037bf37a	grr-VirtualBox	18.4	00:00:00:00:00:00 08:00:27:5f:df:f8	grr

Figure 4: GRR client connection with GRR server

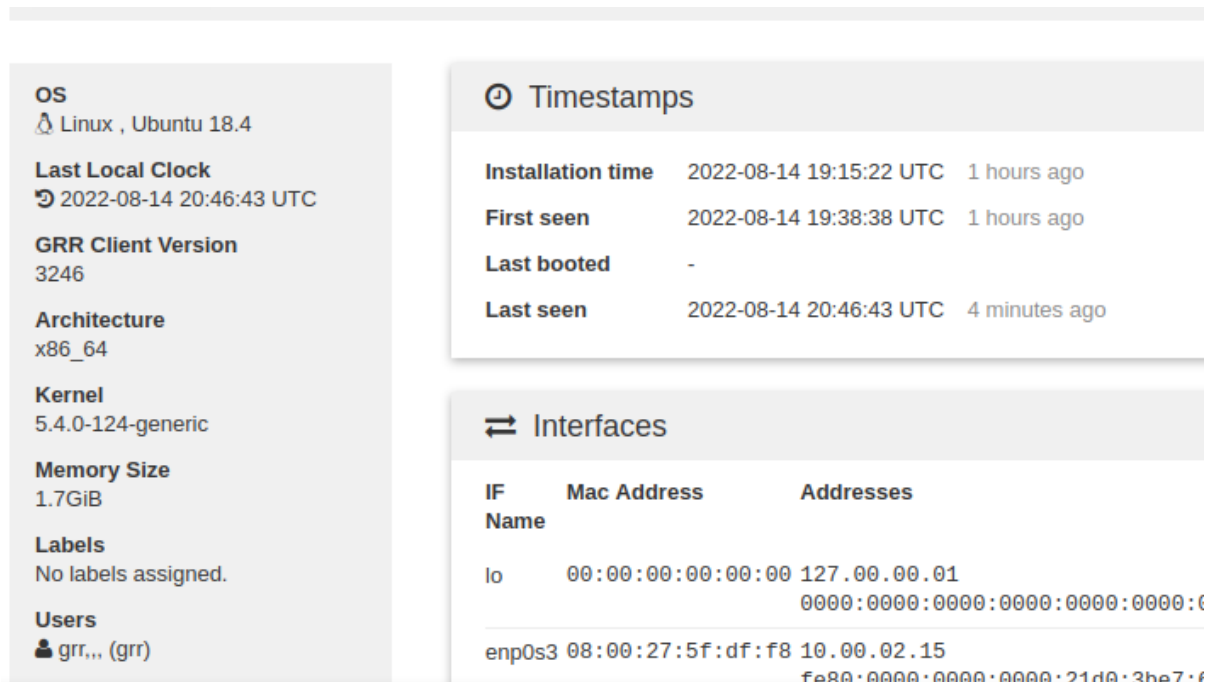


Figure 5: Overview of the GRR client

6 Evaluation

In this section the various results obtained from both methods will be discussed further. By comparing the results obtained from the two methods it will be able to show how the efficiency has improved.

6.1 Comparison

Using AVML memory dump will be generated, and volatility will be used to analyse it. From the memory dump we can get information about various activities in the system. However, the information obtained are in the form of huge cluster. To get practical results deeper analysis needs to be done. The information obtained are not very much user friendly and pleasing. Unless the investigator has good technical knowledge extracting the output to get required information will be difficult. Figure 6 shows the results obtained using “pslist” command.

Offset	Gid	Name	Pid	PPid	Uid
		DTB	Start Time		
0xfffffa0d41b1b9780	0	systemd	1	0	0
		0x00000000119fc4000	2022-08-10 14:26:18	UTC+0000	
0xfffffa0d41b1baf00	0	kthreadd	2	0	0
		-----	2022-08-10 14:26:18	UTC+0000	
0xfffffa0d41b1b8000	0	rcu_gp	3	2	0
		-----	2022-08-10 14:26:18	UTC+0000	
0xfffffa0d41b1bc680	0	rcu_par_gp	4	2	0
		-----	2022-08-10 14:26:18	UTC+0000	
0xfffffa0d41b1bde00	0	kworker/0:0	5	2	0
		-----	2022-08-10 14:26:18	UTC+0000	
0xfffffa0d41b1f2f00	0	kworker/0:0H	6	2	0
		-----	2022-08-10 14:26:18	UTC+0000	
0xfffffa0d41b1f0000	0	kworker/0:1	7	2	0
		-----	2022-08-10 14:26:18	UTC+0000	
0xfffffa0d41b1f4680	0	kworker/u6:0	8	2	0
		-----	2022-08-10 14:26:18	UTC+0000	
0xfffffa0d41b1f5e00	0	mm_percpu_wq	9	2	0
		-----	2022-08-10 14:26:18	UTC+0000	
0xfffffa0d41b1f1780	0	ksoftirqd/0	10	2	0
		-----	2022-08-10 14:26:18	UTC+0000	
0xfffffa0d41b1faf00	0	rcu_sched	11	2	0
		-----	2022-08-10 14:26:18	UTC+0000	
0xfffffa0d41b1f8000		migration/0	12	2	0

Figure 6: Output from pslist command

In the case of GRR the main advantage is that we can get the information from anywhere and anytime. Once the client is set up the investigator can get the required details according to the needs. The user interface is easy, and all the details are shown clearly. Figure 7 shows the virtual file system and the option to download all the files collected to check the details if needed. It is also possible to get the load status of the client machines by using GRR which is shown in Figure 8. GRR has concepts of flows and hunts. Flows are used to gather data by providing path. Flows are the pieces of server-side code that invoke client activities. These calls are made in an asynchronous manner. In other words, they are asked for, and the answers come later. The outcomes of client activities returning to the server trigger a transition between stages in a flow, which is similar to a state machine. Figure 9 shows the flow for collecting Firefox history in the client machine. A hunt specifies a Flow, the Flow parameters, and a set of rules for which machines to run the Flow on.

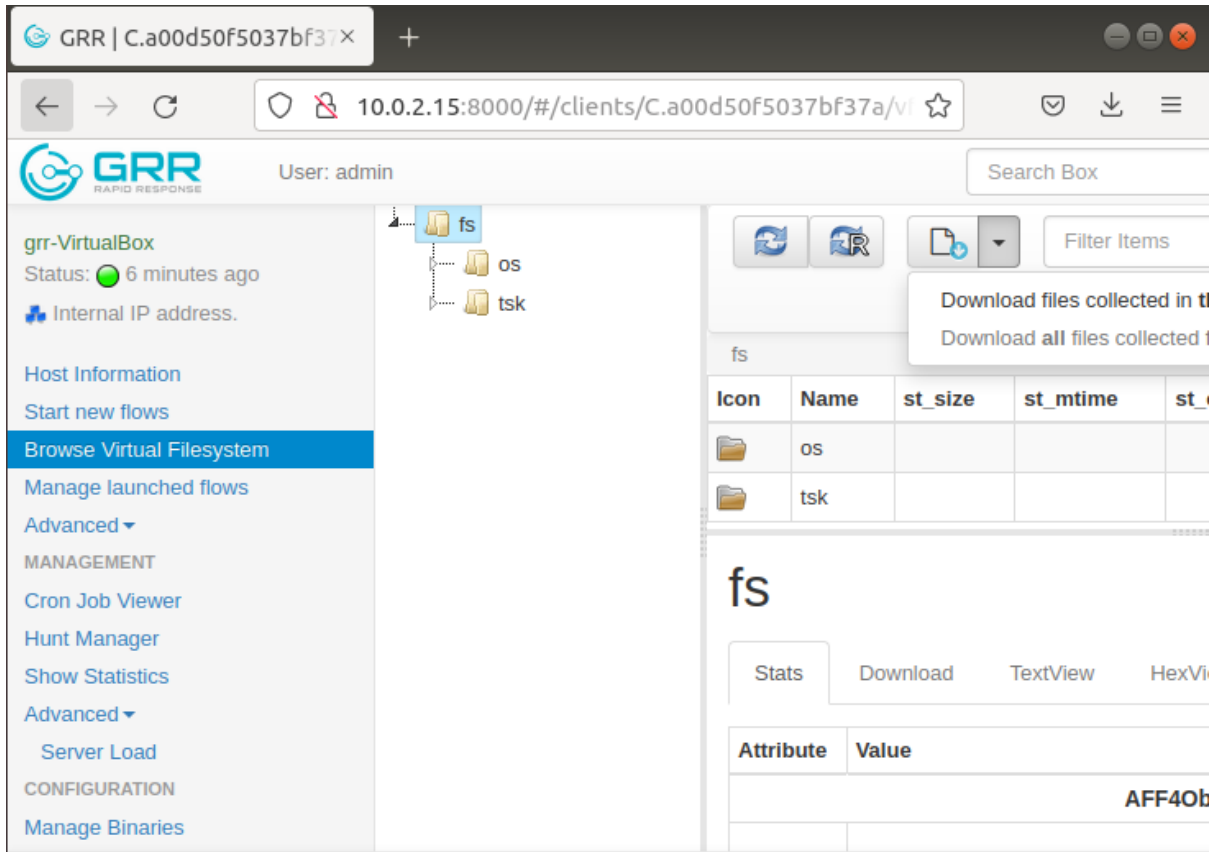


Figure 7: File system and option to download all files from client

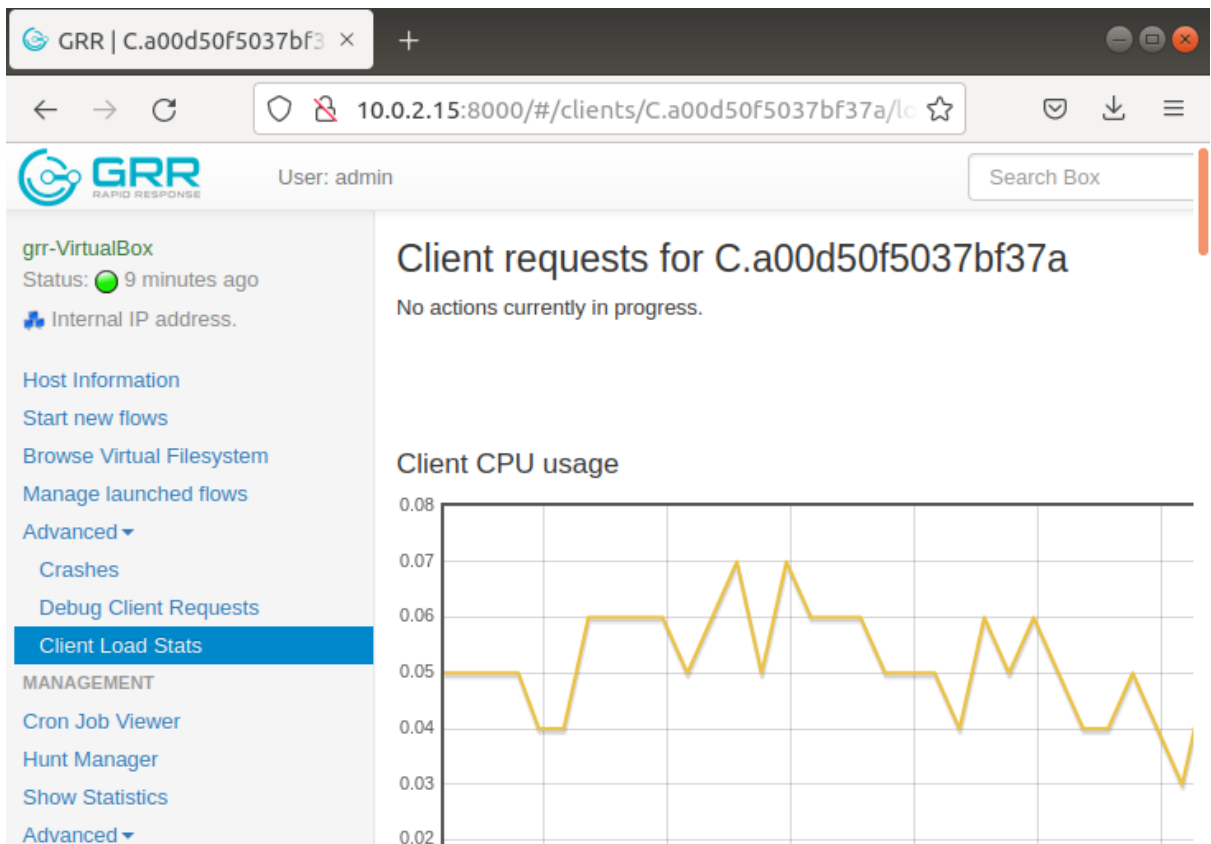


Figure 8: Load status of the GRR client

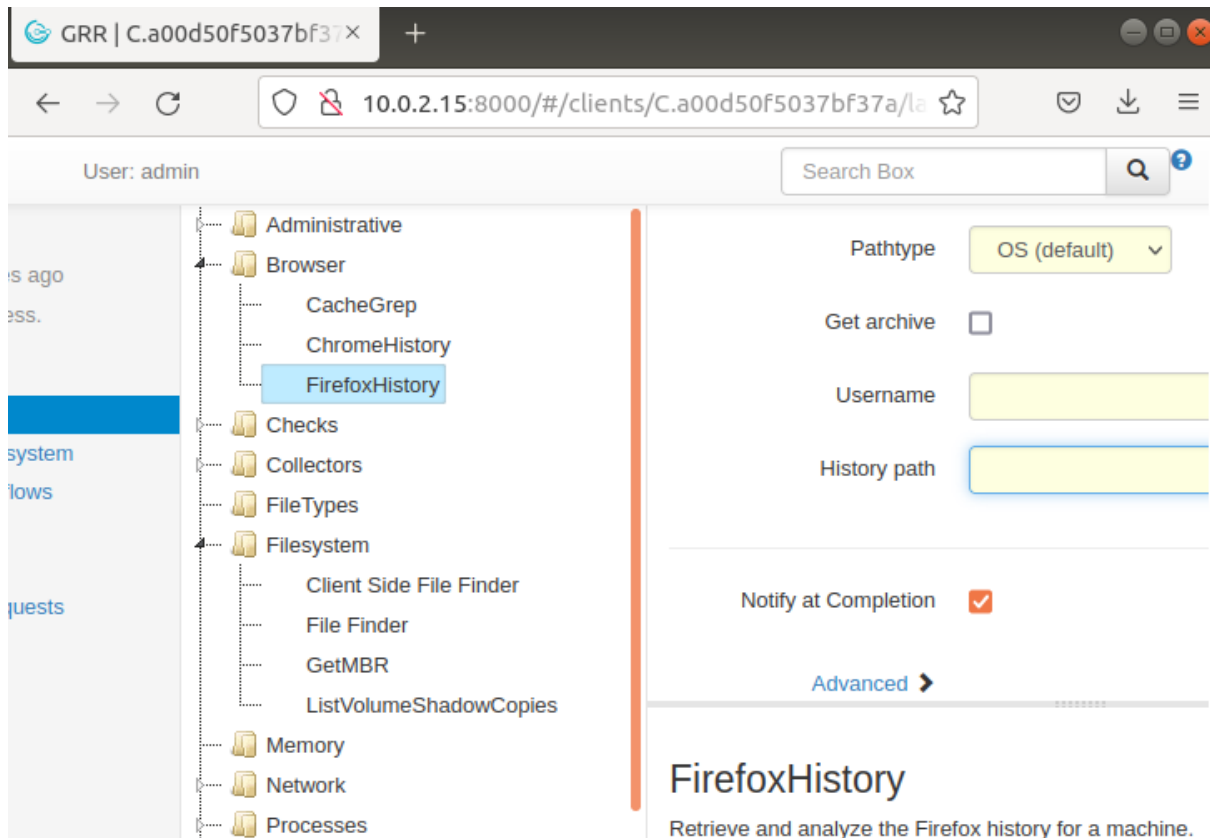


Figure 9: Flow details for Firefox history

6.2 Discussion

From the above results we will be able to see that remote acquisition provides more accurate and relevant information required for the forensic analysis. By using remote acquisition, the time taken to gather forensic details for the investigation is shortened thereby more time can be focussed on analysing the information obtained. The information that is gathered via remote acquisition is precise and will be in order according to the investigator formed. Not only remote acquisition provides all the details that can be obtained from forensic tools but also provides addition information like load status & capability to capture data for a time limit. There will not be a huge dump of data that needed to be combed through. The data obtained will be relevant to the investigation, as the data is collected only from the sources that investigator specifies. However forensic investigation via forensic tools takes a lot of time depending on the size of the system. The output obtained will mostly be a huge memory dump that will consume huge space within the system leading to slowness or crashing of the system. This will halt the forensic investigation whole together. This scenario will not happen if the forensic investigation is done via remote acquisition. Overall, by using remote acquisition the efficiency of forensic investigation in Linux forensics can be improved.

7 Conclusion and Future Work

The forensic investigation conducted on Ubuntu 18.04 by different methods shows how efficiency can be improved for the Linux forensic investigations. The proposed idea is to use remote acquisition for performing Linux forensics rather than using the traditional approach

of forensic tools to conduct the forensic investigation. The results obtained after design and implementation of the two methods for forensic investigation on Linux OS clearly indicate how efficiency is improved for the Linux forensics. The methods used here checks log activities and OS memory to collect information required for the investigation. The current solution proposed still is based on few manual interactions. In future I'll try to develop trained datasets that will be able to perform the entire forensic investigation automatically without any manual assists and collect relevant information required from the Linux system. I'll also try to explore ways by which same datasets can be used for different kernel version of Linux. In future I'll also focus on forensic investigation on other operating systems like Windows, Mac OS.

References

SearchDataCenter. (n.d.). *What is the Linux operating system?* [online] Available at: <https://www.techtarget.com/searchdatacenter/definition/Linux-operating-system#:~:text=Linux%20is%20used%20as%20an>

Andelkovic, A., Hausknecht, K. and Sirovatka, G. (2020). Linux Forensic Triage: Overview of Process and Tools. *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*.

Amarchand, G., Keanu, Munn and Renicker, S. (2018). A Study on Linux Forensics. [online] *www.semanticscholar.org*. Available at: <https://www.semanticscholar.org/paper/A-Study-on-Linux-Forensics-Amarchand-Keanu/0771b65a7baf22b685f90243ff749ef791e4ae39>.

K. U. Maheswari and G. Shobana, "The State of the art tools and techniques for remote digital forensic investigations," *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, 2021, pp. 464-468, doi: 10.1109/ICSPC51351.2021.9451718.

Scanlon, M. (2009). Enabling the remote acquisition of digital forensic evidence through secure data transmission and verification. *researchrepository.ucd.ie*. [online] Available at: <https://researchrepository.ucd.ie/handle/10197/9276?mode=full>.

Son, D. (2019). *AVML v0.3 releases: Acquire Volatile Memory for Linux • Penetration Testing*. [online] Penetration Testing. Available at: <https://securityonline.info/acquire-volatile-memory-for-linux/>.

Sindhuja (2021). *Google Rapid Response Tool for Remote Live Forensics - Security Investigation*. [online] Available at: <https://www.socinvestigation.com/google-rapid-response-tool-for-remote-live-forensics/>.