Ransomware: Analysis and Evaluation of Live Forensic Techniques and the Impact on Linux Systems

Salko Korac

Submitted in partial fulfilment of
the requirements of Edinburgh Napier University
for the degree of Master of Science in
Advanced Security and Digital Forensics

Edinburgh Napier University
School of Computing
November 2023

MSc dissertation check list

Milestones	Date of completion	Target deadline
Proposal	Week 2	Week 3
Initial report	Week 13	Week 12
Full draft of the dissertation	1.5 weeks before final deadline	2 weeks before final deadline

Learning outcome	The markers will assess	Pages[1]	Hours spent
Learning outcome 1 Conduct a literature search using an appropriate range of information sources and produce a critical review of the findings.	• Range of materials; list of references • The literature review/exposition/background information chapter	4-17	115
Learning outcome 2 Demonstrate professional competence by sound project management and (a) by applying appropriate theoretical and practical computing concepts and techniques to a non-trivial problem, <u>or</u> (b) by undertaking an approved project of equivalent standard.	• Evidence of project management (Gantt chart, diary, etc.) • Depending on the topic: chapters on design, implementation, methods, experiments, results, etc.	18-40	230
Learning outcome 3 Show a capacity for self-appraisal by analysing the strengths and weakness of the project outcomes with reference to the initial objectives, and to the work of others.	• Chapter on evaluation (assessing your outcomes against the project aims and objectives) • Discussion of your project's output compared to the work of others.	42-46, 31, 32, 34, 38, 40	75
Learning outcome 4 Provide evidence of the meeting learning outcomes 1-3 in the form of a dissertation which complies with the requirements of the School of Computing both in style and content.	• Is the dissertation well-written (academic writing style, grammatical), spell-checked, free of typos, neatly formatted. • Does the dissertation contain all relevant chapters, appendices, title and contents pages, etc. • Style and content of the dissertation.		150
Learning outcome 5 Defend the work orally at a viva voice examination.	• Performance • Confirm authorship		1 hour

Have you previously uploaded your dissertation to Turnitin? [✓ Yes]/No

Has your supervisor seen a full draft of the dissertation before submission? [✓ Yes]/No

Has your supervisor said that you are ready to submit the dissertation? [✓ Yes]/No

[1] Includes time spent on research area that was later rejected

Authorship Declaration

I, Salko Korac, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained informed consent from all people I have involved in the work in this dissertation following the School's ethical guidelines

Signed: Salko Korac

Date: 22. November 2023

Matriculation no: 40517266

General Data Protection Regulation Declaration

Under the General Data Protection Regulation (GDPR) (EU) 2016/679, the University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

The University may make this dissertation, with indicative grade, available to others.

Salko Korac

Abstract

The first ransomware was discovered in 1989. To get back the data, payment should be made by mail. With the numerous cryptocurrencies today, such as Bitcoin, enormous payment options have emerged. Ransomware became one of the most dangerous cybersecurity threats.

Until now, ransomware seemed to be a Windows-only phenomenon. However, Windows' market shares are continually declining. Linux and Unix-based systems are becoming increasingly popular and are often used to operate critical assets in the cloud. Cybercriminals have also recognised this and are increasingly developing ransomware for Linux-based systems.

Recommendations and instructions for action have so far resulted from Windows-based systems. These assumptions may not apply to Linux-based ransomware. The aim of this project is to research ransomware for Linux and to work out the differences to Windows-based ransomware.

For this purpose, several Linux ransomware variants were executed in an isolated virtual environment. The Icefire, Cl0p and Blackbasta ransomware variants were tested on two different Linux operating systems (Ubuntu and Debian).

However, the results of this work differ significantly from the results of similar work for Windows-based ransomware. While Windows-based ransomware predominantly uses RSA and AES for key management, a variety of approaches was identified for Linux. Cybercriminals appear to be deliberately moving away from RSA and AES to make forensic investigations more difficult. RC4 and ChaCha20 are also used. Linux ransomware also appears to be in a development stage. The samples examined always serve a predefined goal and do not exploit the full potential for damage. Linux ransomware development appears to be in its early stages and is expected to progress and reach a similar level of maturity to Windows-based malware.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisor Leandros Maglaras for the constructive support during this thesis. The discussions and challenges were very helpful for me, while at the same time, I was given the freedom to pursue my approaches.

And I would like to thank the University. The MSc course was practically oriented, at a high level and the tutors were professional and committed. I am grateful for the support and can warmly recommend to study at the Edinburgh Napier University.

Studying at Edinburgh Napier University was one of my best decisions.

Chapter 1

1. Introduction

The aim of this work was to examine the current maturity level of ransomware on Linux operating systems. During the experiments, commonly used live forensic techniques were applied and the results were compared with ransomware for Windows operating systems.

The techniques described in this work are intended to demonstrate techniques that can be used to mitigate an ongoing ransomware attack on Linux operating systems. This topic was chosen because ransomware for Windows operating systems is currently widespread, but ransomware for Linux is expected to increase significantly (Trend Micro, 2022). Security researchers such as Terefos have observed Windows-based Cl0p ransomware being expanded to attack Linux systems (Terefos, 2023).

1.1. History and Background

The 1989 AIDS Trojan was the first ransomware. This ransomware claimed to educate about the autoimmune disease and was distributed via floppy disk. Before installation, users had to agree to a license, which required a payment of few hundred US dollars. Once the installation was complete, the ransomware encrypted all data on the computer and demanded payment by mail to an address in Panama. The author Dr. Popp worked for the WHO and was accused of abusing his reputation to get users to trust the software. The malware uses a symmetric key to encrypt the data.

A few years later, Young and Moti (Young & Moti, 1996) examined the next generation of extortion attacks and described the use of public-key encryption to increase the effectiveness of ransomware attacks. This invention remains the basis of all modern ransomware to this day.

Nowadays, ransomware has become a major threat with revenue of $765.6 million in 2021 (Chainalysis, 2023). Cainalysis estimated that ransomware earnings temporarily fell by 41% to $456.8 million. Willingness to pay the ransom also fell from 76% in 2019 to 41% in 2022. Ransomware payments became more legally risky as some of the ransomware gangs were linked to sanctioned organisations. According to criminal investigations by Europol, the Ukrainian-Russian war forced cybercriminal gangs to relocate their activities into other jurisdictions (Europol, 2023). For 2023, the trend is expected to reverse. According to estimates, ransomware attacks in the first half of 2023 already achieved the revenue of the previous year 2022.

Ransomware gangs base their activities on the value of data. In recent years, Linux and Unix-based systems have increasingly challenged Windows operating system dominance. The share of Windows fell from 95.42% in January 2009 to 69.52% in 2023 (Statista, 2023). In addition, more and more data is being managed by Linux-based systems. The first ransomware variants for Linux systems have been spotted. And the ransomware trend for Linux is expected to continue (Trend Micro, 2022). Also other industry actors observe that Linux systems are becoming a prime target for ransomware (G Data Software, 2022).

Previous scientific projects focused mainly on Windows-based ransomware. There is extensive scientific work on ransomware forensic methods, memory analysis and network communication. Part of this work dealt with the recovery of cryptographic keys.

In order to do justice to the increasing importance of Linux-based systems, reliable scientific work on ransomware for Linux systems is necessary. It is therefore of importance to research the current status of Linux ransomware in a thesis and to compare it with ransomware for Windows.

In the event of a Windows based ransomware attack, a common recommendation is to take computers offline or to shut down (Sittig & Singh, 2019). To prevent further spread, most PCs connected to the network via cable can be easily disconnected from the network. However, this is not always possible. Particularly in a business environment, notebooks automatically reconnect to the wireless network and must be turned off. These approaches can instinctively also be discussed for Linux-based systems.

However, for Linux servers it seems advisable to turn off the system. Fast physical access is not always possible as these are often operated in the cloud or a dedicated data centre. Therefore, the only way to prevent the ransomware from spreading further is to shut down the physical or virtual servers. However, shutting down Linux-based server systems can also result in valuable information about the encryption keys being lost and further forensic work becoming impossible. It is therefore of great relevance for research which recommendations for action can be derived for Linux-based systems.

1.2. Aims and objectives

The aim of this work is to identify forensic techniques that can be recommended for Linux ransomware. It is also important to analyse the effects of Linux ransomware on the respective system. The following goals and objectives are pursued:

1. Conduct a literature review on live forensics for Linux systems and ransomware. This covers the research of current ransomware methods, trends, delivery methods and crypto analysis.
2. Considering the outcome of the literature review during experiment definition.
3. Design and implement an adequate test environment for the experiment execution. The design of the infrastructure should consider common Linux servers operating systems as well.
4. Execute the experiments as defined, using at least 2 ransomware samples and different forensic tools.
5. Critically evaluate the experiment results. Compare the results to similar research, especially the impact compared to Windows. Critically evaluate the project.

At the end of this work, the achievement of these goals will be critically assessed and analysed.

1.3. Ethical safeguards

Malware research can be dangerous and should be conducted with caution. Mistakes can lead to unintended harm to organisations and people. McDonald et al. emphasise the importance of environmental controls to prevent the malware from spreading outside the virtual environment (McDonald et al., 2022).

The British Computer Society (BCS, 2022) emphasises the responsibility of security researchers to avoid injury to others, property, reputation or employment. Security researchers can also be liable to prosecution if they fail to comply with certain legal limits, regardless of whether the damage was committed negligently.

In general, criminal liability is assumed if a security researcher overcomes sufficient protective barriers. What exactly is meant by this can be interpreted in different ways. In general, a comparison is often made with a “normal user without any special technical knowledge”. If the protective measures are so simple that a normal user can overcome them without any special technical know-how, criminal liability is controversial. It becomes clearer when a security researcher can be accused of overcoming protective measures to prove a security gap. As a precaution, the assumed limit value should be set low. Like Schlag (Schlag, 2021) and Balaban et al. report, if a security researcher gains access to a password-protected area by trying out several username/password combinations, this is considered under German regulations to be overcoming protective barriers and could therefore be criminally relevant (Balaban et al., 2021).

It is necessary that every security researcher is clear about the legal responsibilities and ensures that no harm is caused to third parties.

For this reason, the following ethical safeguards were designed in this project:

1. The environment will be completely isolated. The connections and passwords to other networks are discarded before the test is executed. If internet connection is required, it will be ensured, that the host is operated in an stand-alone network setup.
2. No personal data is stored on the host computer. A separate hard drive is used for test execution.
3. The ransomware samples, machine snapshots and files were deleted after successful testing.
4. After testing was completed, the hard drive was reformatted.
5. The files were transferred to the test device via removable storage media that were reformatted after each transfer.
6. Endpoint protection is running on the host computer.
7. It has been proven that the security measures have been successfully implemented.

This measures shall provide a sufficient security level to avoid unintentional damage by the ransomware samples.

2. Literature Review

2.1. Definitions

Davies defined ransomware as malware, which in simplified form is software that prevents users from accessing data (Davies, 2020). Other researchers adopted a similar definition. In an older research paper, Salvi defined ransomware very specifically as a program that prevents the use of documents, computers or online storage (Salvi, 2015) and links the definition of ransomware to unbreakable encryption of data. Berardi et al. introduced subclasses of ransomware, with the crypto subclass being just one term among many (Berardi et al., 2023). In recent studies by Robles-Carrillo & García-Teodoro, ransomware is defined from a legal perspective as an "autonomous crime" (Robles-Carrillo & García-Teodoro, 2022).

Symmetric Ransomware is a subclass (Kong, Ang & Seng, 2015). This type of ransomware uses the same key for encryption and decryption. The main weakness of this type is that it is difficult to keep the key secret. The threat actors must take additional measures to keep the key secret. Modern ransomware uses asymmetric keys, which improves key security (Al-rimy et al., 2018). The use of asymmetric encryption methods only is comparatively slow. To increase effectiveness for the attacker, each victim should receive an individual public key. In a hybrid approach, attackers use a symmetric key for faster encryption and an asymmetric approach for key exchange. The longer the encryption key remains in RAM, the more likely forensic investigations are to be successful. To avoid this, some attacks simply encrypt the beginning of a file to get to the next file more quickly.

The German Federal Ministry for Information Security describes ransomware as malicious programs that attack the security goal of availability (Bundesamt für Sicherheit in der Informationstechnik, 2022). This definition is also not entirely accurate, as ransomware also aims to attack the security goal of confidentiality by threatening the victim with data leaks.

Liska and Gallo define ransomware as a broad term for malware that aims to digitally extort victims into paying a certain fee (Liska & Gallo, 2017). Although this definition is comparatively general and old, it is still very valid. There are some real-world examples that show that hackers try to obtain a ransom payment in different ways. Loss of data access is just one example. Hackers also copy data before extortion to threaten victims with data leaks if the ransom is not paid. In some cases, attackers used fictitious data encryption to trick the victim into paying a ransom. Numerous examples in the recent past have demonstrated various new methods in addition to the classic data encryption extortion. Today, the definition of ransomware includes a wide range of all types of malware aimed at forcing payment. These include, among other things, encryption Trojans, data leak Trojans, but also fictitious attacks that did not actually take place.

Research into ransomware definition has shown that the definition is a moving target as attacks are constantly adapting. Until now, it was assumed that ransomware was autonomous software that could encrypt a PC or network. However, ransomware can change during execution time and download add-on files from a command and control server to continue the attack.

Cybercriminals tend to be selective when choosing the victims. The more complex the victim's infrastructure, the less likely a single piece of ransomware software is to be successful. Most ransomware attacks are supported by criminal threat actors in the background, interacting intensively, spying on the victim's infrastructure, and moving laterally within the infrastructure before initiating the encryption process. Ransomware became the term for the entire attack process: for a piece of software, but also for the services of a cybercriminal group before, during and after encryption. The cybercriminal groups are also organised in a franchise-like manner with sub-licensing structures. Laypeople might misunderstand ransomware as the software that carries out the initial attack. However, a ransomware attack is multi-stage, with the actual encryption being initiated remotely by the cybercriminal gang weeks or months later. Ransomware should be renamed “ransomattack” to cover the complex processes and structures in the background.

It can be assumed that the understanding of ransomware will have to be regularly redefined in the future. Cybercriminal business models are changing. Nowadays, hackers are deeply involved in the attack process to ensure the success.

Where does “ransomware” begin and where does “ransomware” end? In the initial phase of an attack, the main aim is to gain access to the victim. It is initially unclear whether the victim will be infected with ransomware or another attack will be carried out. Initial access can be sold and used for espionage, crypto mining, password theft or data leaks, among other things. At the time of the initial attack, the intent of a ransomware attack may still be open and will be clarified later. It is interesting to further investigate how cybercriminals sell and trade credentials.

It is important to understand the attack chain and current infection routes. The current research results on this are described in the following chapter.

2.2. Relevance for Linux operating systems

The market share of Windows operating system decreased from 95.42% in January 2009 to 69.52% in July 2023 (Statista, 2023). At the same time, the inherently similar Unix-based macOS and Linux-based desktop systems increased the cumulative share from 4.33% to 23.54%.

Another Statista market research focused on the TOP 500 supercomputers, which are primarily powered by Linux-based systems (Statista, 2023). The statistics show Linux as TOP 1, but do not provide any information about which distributions are categorised here. CentOS, also a Linux system, was identified as TOP 2. Cray Linux Environment was rated TOP 3. Especially in the enterprise sector, Linux is often used to operate databases, file shares and other software.

Linux holds a significant market share, especially in the server market (Statista, 2019). Linux achieved a market share of 70.4% in 2019. The Linux provider Red Hat sees itself as the market leader for Linux server operating systems (Red Hat, 2019). Other market research confirms that Linux is the market leader with a market share of 62.4% (Fortune Business Insights, 2023).

Scientific figures on server operating system market shares are rare. Therefore, it was necessary to obtain information from private research institutes such as Statista and Fortune Business Insights. The cybersecurity industry has also noticed that Linux and Unix-based systems are becoming a target of attackers (G Data Software, 2022). The numbers above are enough to illustrate the importance of Linux as a server operating system, but leave room for improvement. Future scientific research is possible in analysing the market shares of operating systems.

The numbers show that Linux and Unix have a small market share in personal computers, but are also continually growing. Linux has a significant market share in server operating systems. Newsletters, shops, web apps, social media networks, forums and many other applications are hosted on Linux systems. Linux ransomware is also expected to increase. The aim of this master's thesis is to contribute to Linux ransomware research.

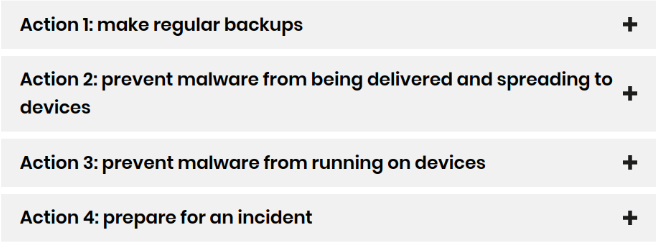
Ransomware attacks have increased significantly on Linux, increasing by 75% (Trend Micro, 2022). Other industry players are also observing that Linux systems are becoming a prime target for ransomware (G Data Software, 2022).

2.3. Current infection paths and impact

Ransomware aims to infect a target and force the victim to pay a ransom. This is often achieved by encrypting the files on the system. Usually the files are also copied before encryption. If the ransom is not paid, the attackers threaten to publish all or part of the data. Some ransomware encrypts data with a symmetric key, while advanced ransomware uses asymmetric keys and attempts to exchange with a command and control server. Most of the infected systems were reportedly Windows operating systems.

The attackers improved the organisation and industrialisation of ransomware extortion by adopting a Ransomware as a Service (RaaS) model (Johansen, 2022). The ransomware groups focus on developing the ransomware and operating the infrastructure, while specialised threat actors gain first access and others inject the payload into the victim's network. Typically, the ransomware operators receive the ransom payment in the background and pass on a defined commission to the threat actor. The RaaS model lowers the hurdles and enables even novice hackers to run effective ransomware campaigns. Initial access often occurs via spear phishing attacks or Visual Basic applications in Office documents (Johansen, 2022). The following paragraphs discuss the recommendations of the British, German and American authorities.

UK NCSC recommendations. The U.K. National Cyber Security Center published some recommendations for action without suggesting a specific priority as seen in Figure 2.1. Even if there is obviously no prioritisation, it can be assumed that an inexperienced reader understands the structure as such. First of all, the NCSC recommends regular backups. In a second step, it should be avoided that ransomware can be delivered on end devices and spread further. In this context, it is particularly recommended to improve email filtering. Furthermore, remote access, multi-factor authentication, reduction of permissions and security patches are recommended. Finally, the NCSC recommends to prepare for an incident.



Action 1: make regular backups +

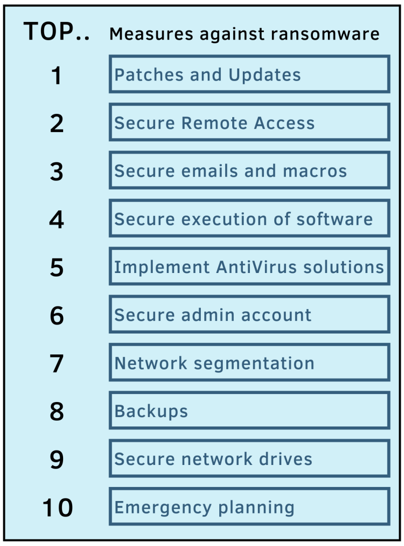
Action 2: prevent malware from being delivered and spreading to devices +

Action 3: prevent malware from running on devices +

Action 4: prepare for an incident +

Figure 2.1: TOP ransomware measures recommended by U.K. NCSC (NCSC, 2021)

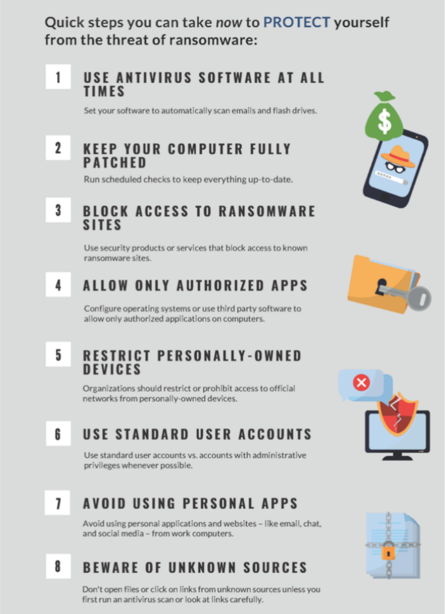
German Federal Office recommendations. In contrast, the German Federal Ministry for Information Security rates the priority of backups relatively low. All recommended TOP 10 measures against ransomware are shown in Figure 2.2 (Federal Office for Information Security, 2023). The German Federal Office gives Patches and Updates the highest priority.



TOP..	Measures against ransomware
1	Patches and Updates
2	Secure Remote Access
3	Secure emails and macros
4	Secure execution of software
5	Implement AntiVirus solutions
6	Secure admin account
7	Network segmentation
8	Backups
9	Secure network drives
10	Emergency planning

Figure 2.2: TOP 10 ransomware measures recommended by German Federal Office for Information Security (2023)

US NIST recommendations. Contrary to this, the United States National Institute of Standards and Technology puts a strong focus on end point security, as shown in Figure 2.3



Quick steps you can take *now* to PROTECT yourself from the threat of ransomware:

1 USE ANTIVIRUS SOFTWARE AT ALL TIMES

Set your software to automatically scan emails and flash drives.

2 KEEP YOUR COMPUTER FULLY PATCHED

Run scheduled checks to keep everything up-to-date.

3 BLOCK ACCESS TO RANSOMWARE SITES

Use security products or services that block access to known ransomware sites.

4 ALLOW ONLY AUTHORIZED APPS

Configure operating systems or use third party software to allow only authorized applications on computers.

5 RESTRICT PERSONALLY-OWNED DEVICES

Organizations should restrict or prohibit access to official networks from personally-owned devices.

6 USE STANDARD USER ACCOUNTS

Use standard user accounts vs. accounts with administrative privileges whenever possible.

7 AVOID USING PERSONAL APPS

Avoid using personal applications and websites – like email, chat, and social media – from work computers.

8 BEWARE OF UNKNOWN SOURCES

Don't open files or click on links from unknown sources unless you first run an antivirus scan or look at links carefully.

Figure 2.3: TOP ransomware measures recommended by U.S. NIST (NIST, 2022)

As shown in Figure 2.3, the US NIST also emphasises the importance of security patches. In another NIST article, Barker et al. recommend starting to train employees (Barker et al., 2022).

The British NCSC, the German Federal Office for Information Security and the US NIST prioritise the recommended measures differently. Collaboration between these institutions could be helpful to exchange mitigation strategies and improve governance. However, all three institutes agree that ransomware is a complex, widespread problem and mitigation measures affect almost all IT areas.

Based on professional experience, there seemed to be broad understanding that ransomware is primarily distributed through a few common methods, but primarily through email and remote access. This no longer appears to be valid. Any vulnerability that can gain access to an IT system or network is also a potential attack vector for a ransomware. Improving the security in software products is therefore considered a TOP 1 measure (Federal Office for Information Security, 2023). Based on professional experience, no ransomware incident was observed, which initially occurred via email. Rather, drive-by infections were observed. Employees tend to use software from unsafe sources that contain manipulated software. In a separate analysis, 122 documented cybersecurity incidents in the last 2 years in a large, global corporation were analysed. All four ransomware attacks were caused by installing third-party software on clients. Only 3 cases of email phishing were documented, albeit in connection with the targeted compromise of business emails to fraudulently carry out a financial transaction. This data may not be disclosed for reasons of confidentiality and is therefore described based on professional experience.

In one recently publicised ransomware incident, a victim asked the cybercriminals how they gained access to the system. The cybercriminals then responded that access had been bought via the dark web (Guenni, 2023). An example of specialisation in the attack chain.

Another textile company from Germany also fell victim to a ransomware attack. Forensic analysis revealed that the cause was a manipulated browser update. The employee was able to install software due to the low security regulations on the notebook (Schreiber, 2023).

These examples confirm the assessment that the previously widespread and well-known assumption that ransomware was primarily distributed via few common methods seems no longer valid. Many companies have now improved the email systems, mark external emails and warn the user to open encrypted zip files accordingly.

Ransomware gangs achieve an immensely fast and effective success through the division of work. Every gap in software products is eligible to be exploited. If access to an IT system is already successful, the hackers use emails and phishing attacks as supportive measures.

There are cases where ransomware is designed and delivered as an software artefact in such a way that it can infect a computer by it's own capabilities in an autonomous way.

Due to the increasing division of tasks and industrialisation of various work steps in the field of cybercrime, cases often arise in which attacks simply aim to initially gain access to the victim's network. The attack does not have to be aimed directly at ransomware at the beginning of the process and the specific use of the initial access will only be decided later. It is conceivable that additional data may be stolen, that the victim may be spied on, or that the data may be sold to ransomware groups who then attempt to make the attack successful through manual assistance. This means that ransomware "lives" during the attack and the attackers explicitly adapt it to the victim's environment. Ransomware should no longer be viewed as a stand-alone software product. Increasingly, entire chains of attacks take place before the ransomware is executed.

Based on the professional experience and the above explanation, an attack chain is introduced as shown in Figure 2.4 below. This attack chain model considers a conscious decision milestone by the hackers, where further use is discussed and decided.

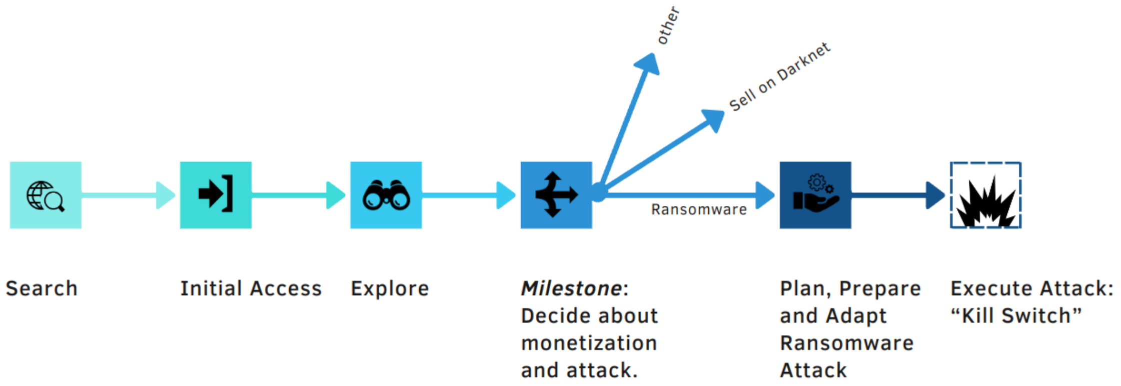

other
Sell on Darknet
Ransomware
Search
Initial Access
Explore
Milestone: Decide about monetization and attack.
Plan, Prepare and Adapt Ransomware Attack
Execute Attack: "Kill Switch"

Figure 2.4: Attack chain from the attacker's perspective with decision steps

Specialised cybercriminals focus on searching for new victims and gaining initial access. This is achieved through targeted attacks using individual methods and through non-targeted attacks using automated processes (e.g. brute force techniques, exploits, zero-day exploits).

Initial access includes all types of access to information and systems that can be beneficial to cybercriminals. This could be access to a network, cloud accounts, data, systems or others. After initial access is confirmed, attackers explore the situation to continue the attack. The exploration includes a scan of surrounding systems, documents and accessible information. According to professional experience, the decision-making process includes the attacker's ability to self-monetise the access, the personal risk of being caught or fail, the region, the value of the data, and willingness of the victim to pay a ransom. There is widespread agreement that Russian cybercriminals do not target victims in Russia. According to Glenny, the Russian cybercriminal group Cl0p avoids attacks on government institutions, cities or the police (Glenny, 2023). Cl0p promises that in such cases it is not interested in disclosing relevant information. It is precisely the exclusion of state victims that can initially protect cybercriminals from aggressive prosecution.

The ransomware attack from the victim's perspective is shown in following Figure 2.5.

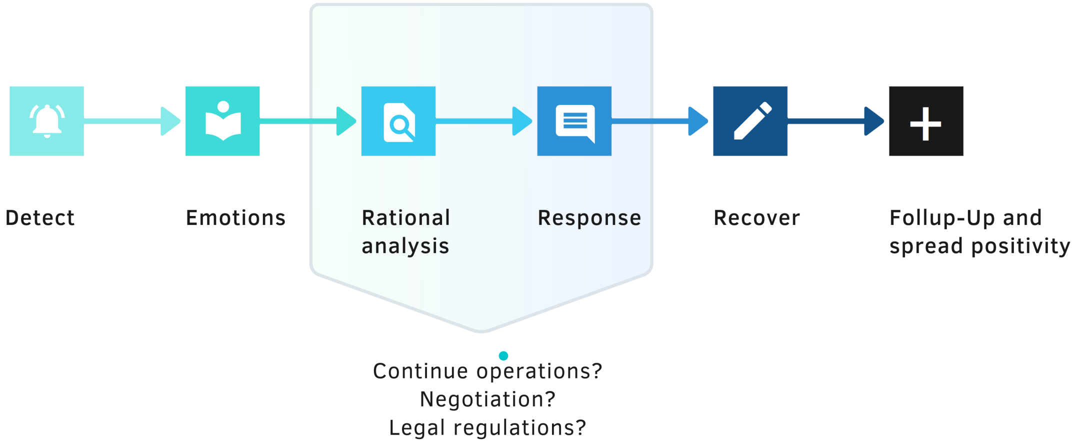


Figure 2.5: Attack chain from the victim's perspective

An attack is usually discovered by an employee during regular working hours, often after weekends. This employee reports the incident internally to the responsible colleagues, triggering a frenzy, and the incident escalates at all levels, including management.

It is known from professional experience that ransomware victims initially react emotionally and categorically refuse to negotiate. After taking stock, victims react more soberly, are more likely to consider paying the ransom and are more open to recognising this process as a deal. After the emotional phase is completed, a response is prepared. In both phases, the victim examines the possible options and checks whether it is possible to continue business operations, how promising negotiations would be and whether paying a ransom is legal.

Ransom Payment Considerations and Legal Requirements. Bart et al. emphasise that paying a ransom is not generally prohibited under US law (Bart et al., 2018). However, in some countries there is a risk of committing a crime if the cybercriminals receiving the payment are classified as terrorists. New regulations have been enacted in North Carolina and Florida to prohibit ransom payments to certain groups, primarily government institutions (Boyd et al., 2023). UK regulations do not prohibit the payment of a ransom, but do provide few clear exceptions. Ransom payments are prohibited when sanctions are imposed (HM Treasury Office of Financial Sanctions Implementation, 2023). In practice, it is very difficult to check the applicability of sanctions before making a payment. Therefore, in the event of an attack, it is imperative to contact local authorities. It is generally recommended that the negotiation is not carried out by those affected themselves, but rather is supported by professional negotiators.

The United States White House wants to take a financially-focused approach to curbing the ransomware problem and has launched the International Counter Ransomware Initiative (CRI). As of October 2023, 50 member countries, including the United Kingdom, have pledged not to pay a ransom (The White House, 2023). This initiative would urge member countries to take appropriate proactive measures instead of paying the ransom.

There is a general warning against paying ransoms. Even if paid, there is no guarantee that the attackers will keep the promise, that the key is fully functional, or that the decryption software is fast enough. As Wood reports, Colonial Pipeline was attacked by ransomware (Wood, 2023). Even after paying the ransom, decryption was very slow (Sanger & Perlroth, 2021), importing the backups was actually faster than decryption (Kaspersky ICS CERT, 2021).

Linux ransomware. Security researcher Donauer analysed early versions of a Linux ransomware variant Linux.Encoder.1 (Donauer, 2015). The ransomware started as a Linux daemon and used a 2048-bit encryption key. The ransomware encrypted files in `/home`, `/root`, `/var/lib/mysql`, `/var/www`, `/etc/nginx`, `~/apache2` and `/var/log`. The work showed that ransomware developers are making targeted attempts to encrypt user files, databases and web server folders. In addition, folders containing "*backups*" or source codes, media data and documents were also encrypted. To avoid loss of access, configuration files and binaries were not encrypted.

Previous ransomware research was focused on analysing ransomware on Windows-based systems. Beaman et al. analysed the challenges and future research directions of ransomware. However, the focus of the work was on Windows (Beaman et al. 2021). Linux variants were not considered in this investigation.

Davies experimented with the NotPetya, Bad Rabbit, and Phobos ransomware for Windows (Davies, 2020). The work used static and dynamic analysis as well as memory capturing techniques. The experiments focused on figuring out how the cryptographic key was stored in memory and extracting it accordingly. This included analysing whether the key in memory could be used to decrypt the files and bypass the attacker's encryption processes. The old Windows version 7 and the newer version 10 were used for the tests. The experiments showed that the keys were only temporarily available in memory and could not be recovered after the ransomware completed encryption. In summary, ransomware experiments and forensic data extraction must be done in a timely manner to analyse the cryptographic keys in memory.

McDonald et al. analysed the impact of WannaCry, TeslaCrypt, and Jigsaw ransomware samples on Windows Active Directory Domain Services (McDonald et al., 2022). The directory service was operated on a Windows Server 2016 machine. WannaCry uses file shares to spread across the network. It was determined that WannaCry did not have access to the command and control servers and therefore immediately began to spread across the network. In conclusion, the encryption process can be triggered by disabling the internet connection in the test environment. Results showed that all ransomware samples attacked nearby file shares and web server (IIS), which shows that ransomware for Windows is capable of successfully attacking various Windows services. The work underlines the importance of defining environmental controls to prevent the malware from spreading outside the virtual environment. In addition, the definition and operation of vulnerable services is recommended. For example, a web server could be used with a simple HTML page to easily test ransomware execution.

There are different ways of lateral movement for Windows ransomware and for the collection and exfiltration of the victim's data (Skulkin, 2022). Known attack vectors include remote and file sharing services. Data is collected and exfiltrated via email, shared network drives, and web services. The investigation revealed that the services in the test environment must be carefully selected and the selection should take into account the data value for a potential attacker. This includes web services, databases, and `$HOME` directories.

2.4. The emotional aspect during a ransomware attack

Current understandings of ransomware attacks had painted a very rational picture how the attacks happen. Little or no consideration is given to the emotional component of both the victim and the attacker during the literature review. The influence of emotional factors on cyber incidents would be an interesting further field of research.

On the victim side, a hacker attack can lead to internal competition and hierarchical thinking, especially in the initial phase when the organisation forms an emergency response team to analyse the incident. Multi-layered reporting chains are created. One reason for this may be that no emergency plan has been drawn up, but also that, for career and profile reasons, the situation is being taken over by people who cannot make a significant contribution to the solution. This can be perceived as unfair by the respective experts who take care of the IT systems.

After closing the incident, many managers look forward and want to quickly put the situation behind. They then try to convey a feeling of strength. Some information security officers often consider such an approach to be unsustainable. As soon as the closure of the incident is in sight and operations resume, interest in IT security will also decrease.

IT professionals can feel undervalued because responsibility is taken away from them as soon as a problem arises. At the same time, there is a risk of loss of face because company management could assume that those responsible have not done their job well in recent years.

This permanently undermines trust, because in times of crisis, people want to make a name for themselves. Roth reports on a ransomware case in a city where poor collaboration and communication massively delayed resolution time (Roth, 2023). Two concurrent emergency teams were deployed, which made coordination and communication difficult overall. It is criticised that the restoration could have been quicker and cheaper if those involved had worked better together.

Future research could be conducted in the area of emotional aspects during cybersecurity incidents or ransomware attacks.

2.5. Linux live forensic analysis

Andelkovic, Hausknecht, and Sirovatka introduced a forensic triage technique for Linux forensics (Andelkovic, Hausknecht, and Sirovatka, 2020). This technique is known from medicine, in which a doctor has to decide which patient will survive anyway, which has no chance of survival at all and which will only survive with help. A similar approach is taken in forensic triage. Investigators must avoid collecting all possible data or too much data. It is better to only collect what is necessary for the purpose of the investigation. This illustrates how important it is to define clear goals.

Dynamic and static methods. Ransomware can be analysed using two main approaches. Static analysis does not execute the ransomware (Umara Urooj et al., 2022). This approach can be used to analyse the contents of the ransomware sample, including reverse engineering the code. This procedure is usually complex and requires a clear definition of the goal. The dynamic approach runs the ransomware sample under controlled conditions to collect the data and observe the behaviour. However, a hybrid approach utilises both and results in more accurate data (Umara Urooj et al., 2022).

McDonald et al. (2022) and Arfeen et al. (2022) used VirtualBox virtual machines for the test environment. Both research papers show that the use of virtual machines reduces the risks when dealing with malware during analysis. McDonald also highlights the value of simple tools like the in-system Windows Process Monitor for dynamic malware analysis.

However, forensic frameworks can also be very useful in ransom analysis and speed up the work (Akbanov et al., 2019). Akbanov used a REMnux Linux distribution, to easily configure DNS and HTTP services and to capture all network communications via Wireshark. In summary, it is important that the forensic environment can also capture DNS communications.

Ransomware typically communicates with a command and control server, most likely via port 445 (Kara & Aydos, 2022). Kara & Aydos also recommend the Autopsy tool as a framework for conducting comprehensive malware analysis. Also virtual machines were used for the forensic analysis. File directory logs were useful to identify which files were accessed by the malware sample and which files were written to the system.

The open source framework Xplico has been successfully used for network analysis (Parasram, 2020). The main advantage is that the tool can automate the analysis process and find relationships. This tool could be used during research to analyse network communication. Hampton et al. analysed ransomware behaviour using Windows API calls (Hampton et al., 2018).

Cryptographic key management is a success factor for ransomware. Macfarlane et al. described how ransomware manages cryptographic keys using either a single key system or a hybrid key system (Macfarlane et al., 2023). The single-key system directly uses a symmetric or asymmetric key, while the ransomware uses both simultaneously in a hybrid key management approach. Another study identified AES and RSA as the most commonly used ciphers in ransomware (Bajpai et al., 2018).

2.6. Memory forensics

2.6.1. Identifying keys in memory

Malin et al. (2014) and Davies et al. (2020) describe forensic tools for malware analysis on Linux systems. Both highlight aeskeyfind and rsakeyfind as established tools. But Malin et al. recommend the volatility framework for dump analysis. Volatility can also link storage processes to network communications. The SecondLook tool was introduced to detect hidden kernel modules.

Davies selected the tools specifically and saw findaes and ransomAES as suitable forensic tools for research (Davies, 2020). Both tools were able to correctly identify AES keys in storage and did not require the use of a powerful framework. This is consistent with other work that emphasises the importance of a clear forensic goal rather than collecting extensive data (Andelkovic et al., 2020).

2.6.2. Issues

Linux operating systems differ in details between distributions. Some forensic tools also need to be compiled for and on a specific kernel version (Andelkovic, Hausknecht & Sirovatka, 2020). To improve reliability, the authors recommend using different tools for the same purpose.

Free tools rely on community development. Commercial tools were not considered for this project due to lack of budget.

The Volatility Framework is already equipped with several profiles for Windows, but the profiles for Linux are very diverse and must be created individually (Volatility Foundation, 2020). This is time-consuming during a security incident.

Problems can arise when carrying out the experiments and collecting data. During ransomware execution and when tests are run repeatedly, problems with the correct timing of data collection may occur. Furthermore, storing the extensive amount of RAM and hard drive data remains a challenge. The tests are expected to take up a lot of disk space. In addition, the number of experiments represents a project risk. Consequently, a minimum of 3 experiments are planned, which will be expanded to up to 5 experiments if the aforementioned risks do not occur.

2.7. Conclusion and summary

The literature review began by presenting the relevance of ransomware to Linux operating systems and later discussed the infection paths and impacts on Windows. The review also discussed the emotional aspects of a ransomware attack. Previous analyses and research always present ransomware attacks in technical terms, although the emotional aspects are hardly taken into account in current research. Exploring the emotional aspects during an attack and the influence would be very interesting further research.

Live forensic analysis for Linux can depend heavily on the respective kernel version. Some tools need to be compiled with the specific kernel version to enable reliable forensic investigations, which increases the effort and reduces the chances of reliable memory forensics.

The research found that it is better to define a specific forensic goal and select targeted tools. Collecting large amounts of data and using complex platforms is not recommended unless necessary.

The literature review also revealed that most ransomware programs use hybrid key management, which involves the use of symmetric AES keys backed by an asymmetric public key.

The goal of this project is to analyse memory and network traffic. In summary, this project will use simple tools such as aeskeyfind and findaes as specific tools for memory forensics and avoid excessive data collection and analysis. The reliable Wireshark tool is used for network analysis.

3. Design

3.1. Introduction

At the beginning of the design phase, some research methods are analysed and considered to achieve the objectives of the thesis. The experiments are then defined and a suitable environment is designed to conduct the experiments and validate the hypotheses regarding Linux ransomware. The results of the literature search are also taken into account when designing the experiments and the test environment.

3.2. Research Methodology

Chandra & Hareendran (2018) as well as Edgar & Manz (2017) described different research approaches.

In most cases, observational research is a qualitative approach in which a test subject is observed to gain new insights. Observational research is useful when the sample base is small. Logical-theoretical research is based on mathematical-formal proofs and is useful when experiments or observations cannot be carried out. Chandra & Hareendran (2018) highlight a participatory research approach, but do not describe this approach further.

Both research teams describe an experimental research approach that focuses on quantitative findings. In these cases, the researcher can control or monitor and interpret all variables of interest. Chandra & Hareendran (2018) point out that experimental research has the limitation that although the results could be considered true facts, they may still lack interpretation. In addition, experimental research may focus only on specific problems, work by isolating unknown variables, and may not be useful when complex relationships need to be evaluated.

This work is classified as experimental research and is based on the hypothesis that the maturity level of Linux ransomware is lower compared to Windows operating systems, but uses comparable key management systems.

3.3. Experiments Design

3.3.1. Experiment 1 – Is the key in memory

Most ransomware uses a hybrid key management system. Both symmetric and asymmetric keys are used to carry out the attack. The private key is intended to be securely stored by the attacker, while the ransomware is delivered with a predefined public key. It may also be possible for the ransomware to communicate with a command and control server to obtain an individual public key. The ransomware is believed to contain AES-based encryption. In both cases, it is expected that the keys are visible in memory during encryption and that a symmetric key is used for the encryption process itself.

This experiment runs the ransomware and creates a memory dump for analysis using live forensic techniques. Before the experiment execution, the memory will be captured for before-after analysis. A visualisation of the experiment is shown in Figure 3.1.

3.3.2. Experiment 2 – How long is the key present

In this experiment a memory dump is taken in 7 time intervals with the goal to identify how long the key is present in the memory. Memory dump is also taken before the experiment and after reboot. To increase the chances to identify the encryption keys in memory, some big files will be stored on the machine, to prolong the encryption time.

3.3.3. Experiment 3 – Does the key decrypt the files

If keys are found in memory, they are used to decrypt the files. A visualisation of this experiment is shown in Figure 3.2.

3.3.4. Experiment 4 – Does the sample spread throughout the network

This experiment examines whether and how the ransomware attempts to spread across the network. Network communications must be captured outside the virtual machine to prevent detection and evasion by the ransomware. To simulate a realistic attack surface, a second server is placed on the same network providing SSH, FTP, Web services, MySQL databases and a Samba file share. During the experiment execution, client and server will have open connections through these protocols.

The files will be captured using Virtual Box virtual interfaces and Wireshark.

3.3.5. Experiment 5 – What is the impact of the encryption

In this experiment the impact on the system will be analysed. Honeypot files will be placed in `/home`, `/root`, `/var/lib/mysql`, `/var/www`, `/etc/nginx`, `~/apache2` and `/var/log`. Also a web server with a small web page will be created as honeypot. Furthermore a server will provide FTP, SSH and Samba file shares and will have open connections with a client.

File hashes will reveal which files were encrypted by the ransomware.

3.3.6. Test plan

A multi-dimensional combination was developed that takes into account different operating systems and permission levels. The complete test plan is shown in Table 3.1.

Experiment	Combination / Test Cycle
Experiment 1 – Is the key in memory	Combo 1: Icefire on **Client-Debian** (Debian) as normal user Combo 2: Icefire on **Client-Debian** (Debian) as root user Combo 3: Cl0p on **Client-Debian** (Debian) as normal user Combo 4: Cl0p on **Client-Debian** (Debian) as root user Combo 5: Blackbasta on **Client-Debian** (Debian) as normal user Combo 6: Blackbasta on **Client-Debian** (Debian) as root user Combo 7: Icefire on **Server-Clean** (Ubuntu) as normal user Combo 8: Icefire on **Server-Clean** (Ubuntu) as root user Combo 9: Cl0p on **Server-Clean** (Ubuntu) as normal user Combo 10: Cl0p on **Server-Clean** (Ubuntu) as root user Combo 12: Blackbasta on **Server-Clean** (Ubuntu) as normal user Combo 13: Blackbasta on **Server-Clean** (Ubuntu) as root user + Retest of Icefire on Server-Clean-CMD (Ubuntu) Retest of Cl0p on Server-Clean-CMD (Ubuntu) Retest of Blackbasta on Server-Clean-CMD (Ubuntu)
Experiment 2 – How long is the key present	Same like in Experiment 1 and 2
Experiment 3 – Does the key decrypt the files	Same like in Experiment 1 and 2
Experiment 4 – Does the sample spread throughout the network	Same like in Experiment 1 and 2 + Retest of Icefire on Server-Clean-CMD (Ubuntu) Retest of Cl0p on Server-Clean-CMD (Ubuntu) Retest of Blackbasta on Server-Clean-CMD (Ubuntu)
Experiment 5 – What is the impact of the encryption	Same like in Experiment 1 and 2

Table 3.1: Experiment and combinations plan

3.3.7. Experiment playbook

To ensure consistent experiment quality, a playbook was defined that is used to prepare and conduct each experiment. The general steps in the playbook are shown in Figure 3.1 below.

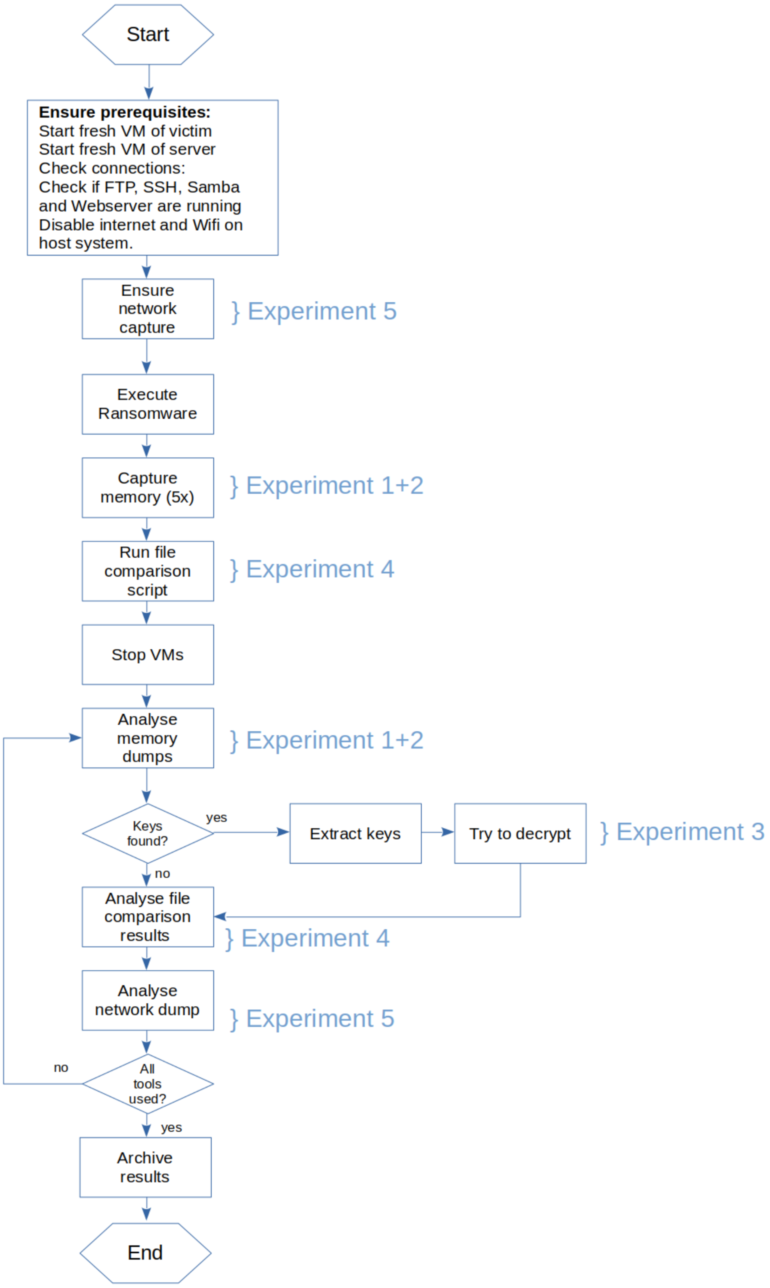


Figure 3.1: Experiment playbook

Before each experiment, the prepared virtual machines are copied and started as a new virtual machine. This includes allowing the victim's VM to connect to the Linux server via FTP, SSH, Samba and HTTP. The connection to the Internet is being tested and must be deactivated. The Wi-Fi on the host system is disabled.

Ransomware execution. The ransomware samples are already available on the VM template. The example is unpacked from the archive before execution. The ransomware is executed via the command line.

Memory capture and analysis. During execution, memory is captured outside the virtual machine, as suggested by McLaren et al. (2019) and Nissim et al. (2019). This has the advantage that the ransomware sample cannot detect when it is analysed. At least two tools are used to identify the symmetric keys in the memory dumps. `aeskeyfind` and `findaes` are used to analyse the memory dumps. This approach reduced the risk of wrong results, unknown bugs or vulnerabilities of a specific tool.

Network capture. The network is captured outside the VM to prevent detection and evasion by the ransomware sample. This is done by the Virtual Box internal nictrace function, as shown below.

Activation of network trace

```
# VBoxManage modifyvm "ubuntu" --nictrace1 on --nictracefile1
out.pcap
# VirtualBox -startvm "ubuntu"
```

Evaluation of impact. To determine the impact on the file system, a set of files were defined and compared before and after using MD5 and SHA checksums. To avoid accidental encryption, this file was stored under `/usr/bin`. The ransomware is expected not to encrypt binaries to ensure continuous operating system connectivity.

Honeypot files for encryption. To analyse the impact on the file system, some files were predefined and placed in different folders across the file system, as shown in Table 3.2.

Location	File	Size in	On client / server?
`/home`	a.doc/jpg	each 2,5mb	Server and client
`/root`	a.doc/jpg	each 2,5mb	Only server
`/var/lib/mysql/a`	Persons.ibd/Persons2.ibd	each 112kb	Only server
`/var/www/html`	a.html/jpg	2,5mb	Only server
`/etc/nginx`	a.html/jpg	2,5mb	Only server
`/etc/apache2`	a.html/jpg	2,5mb	Only server
`/var/log`	a.log	1mb	Only server

Table 3.2: Overview of honeypot files for encryption

3.4. Environment design

When designing the test environment, care must be taken to ensure that it is realistic. The design should only deviate from real conditions if necessary. When designing a test environment, a careful balance must always be struck between effort and the creation of realistic conditions. Conditions that are too realistic make forensic work more difficult, conditions that are too lax are easier to implement, but on the other hand can make essential research opportunities impossible. Any research work has a certain number of unknowns that the researcher should consider.

It is therefore advisable to design the test environment with a certain amount of resources and time in order to reduce the risk of restricting the research results. It is possible for malware to detect when being analysed using forensic tools and therefore behave differently. Forensic analyses should be structured in such a way that the malware has the lowest possible chance to detect ongoing forensic analysis. However, an environment that is too realistic can pose a danger. On the one hand, it is possible that a ransomware will report to the cybercriminal authors or operators when a forensic analysis is carried out. Sensitive information may be transmitted to the attackers. An environment that is too realistic makes it difficult to control and measure all dependencies. It can also increase the effort required for analysis and lead to false findings due to uncontrolled interference.

In the course of this work, a main test environment was initially set up. This test environment was realistically structured and offered various services in the network. This in turn increased the risk of interference. For this reason, another simplified virtual environment was set up later to exclude interference and retest unexpected results.

Virtual machine technology is used in this project because cybercriminals have to take into account that Linux is often operated in virtual cloud environments. This provides a convenient way to perform external memory dumps as well as external network capture, without noticeable risk of being discovered by the ransomware.

Operating a virtual environment also carries the risk of infection of the host system and host network. This requires appropriate environmental controls. In this project, the main test environment is completely isolated from the host network. The virtual machines are operated in a separate virtual network. The virtual and physical network interfaces on the host were disabled. All Wi-Fi password credentials on the host were deleted during the experiments and the host system data was backed up before the experiments start. These measures significantly reduced the risk of an outbreak.

3.5. Analysis design

The literature review focused on the results and comparison with academic research by Davies (2020) and McDonald et al. (2022). The results of this project will be compared primarily with this both research papers. To compare the results with Windows forensic methods, a results matrix is used as shown in Table 3.3.

Research	**Ransomware for Windows by other research**			**Ransomware for Linux by the author**		
Sample	**Sample A**	**Sample B**	**Sample C**	**Icefire**	**Cl0p**	**Blackbasta**
Criteria 1						
Criteria 2						
Criteria *n*						

Table 3.3: Example of results matrix for comparative analysis with other research

If the aforementioned research does not allow a comparison, a comparison will be made inbetween the respective Linux ransomware examples. Table 3.4 is used for this, as shown below.

Research	**Linux ransomware research by the author**		
Ransomware Sample	**Icefire**	**Cl0p**	**Blackbasta**
Criteria 1			
Criteria 2			
Criteria n			

Table 3.4: Example of results matrix for the selected Linux ransomware samples

4. Implementation

4.1. Ransomware samples

Recent reports of ransomware attacks on Linux were researched in the news. The Icefire ransomware was known to abuse IBM Aspera Faspex (Delamotte, 2023). The Cl0p Linux variant is known to successfully attack the University of Columbia (Terefos, 2023). And Blackbasta ransomware focused on VMWare ESXi virtual technology (Umawing, 2022). The analysis mainly considered newspapers and non-scientific security research work from other security companies. The following ransomware examples have been selected.

Icefire. This ransomware was primarily designed and developed for Windows, but in the recent months also attacks on Linux operating systems have been observed misusing IBM Aspera Faspex file transfer software (Delamotte, 2023).

Cl0p. This ransomware is known to be similar to it's Windows variant and appears to be still in initial development (Terefos, 2023).

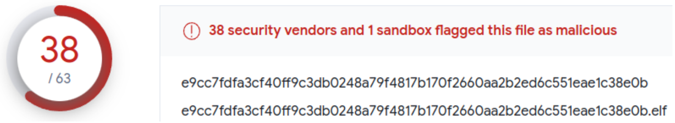
Blackbasta. This example is known to be a sub-variant of the Conti ransomware and focuses on exploiting VMware's ESXi virtual machine technology (Umawing, 2022).

An overview of the obtained ransomware samples is given in Table 4.1.

Name	HSA256 Checksum	Date
Icefire	e9cc7fdfa3cf40ff9c3db0248a79f4817b170f2660aa2b2ed6c551eae1c38e0b	2023-07-06
Cl0p	09d6dab9b70a74f61c41eaa485b37de9a40c86b6d2eae7413db11b4e6a8256ef	2023-07-06
Blackbasta	0d6c3de5aebbbe85939d7588150edf7b7bdc712fceb6a83d79e65b6f79bfc2ef	2023-07-06

Table 4.1: Checksum values for selected ransomware samples

The samples were obtained from Malware Bazaar (www.bazaar.abuse.ch) and uploaded to VirusTotal (www.virustotal.com) to ensure that they were valid ransomware samples. The scan results are shown in Figures 4.1, 4.2 and 4.3.

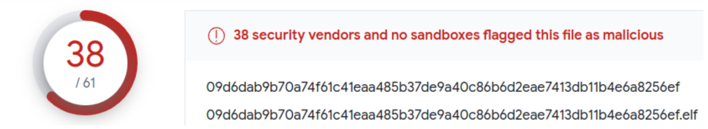


38 security vendors and 1 sandbox flagged this file as malicious

e9cc7fdfa3cf40ff9c3db0248a79f4817b170f2660aa2b2ed6c551eae1c38e0b

e9cc7fdfa3cf40ff9c3db0248a79f4817b170f2660aa2b2ed6c551eae1c38e0b.elf

Figure 4.1: VirusTotal result for Icefire

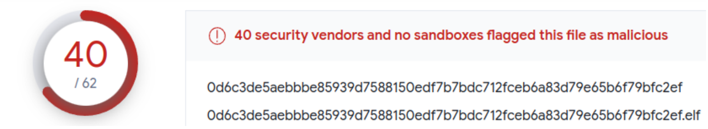


38 security vendors and no sandboxes flagged this file as malicious

09d6dab9b70a74f61c41eaa485b37de9a40c86b6d2eae7413db11b4e6a8256ef

09d6dab9b70a74f61c41eaa485b37de9a40c86b6d2eae7413db11b4e6a8256ef.elf

Figure 4.2: VirusTotal result for Cl0p



40 security vendors and no sandboxes flagged this file as malicious

0d6c3de5aebbbe85939d7588150edf7b7bdc712fceb6a83d79e65b6f79bfc2ef

0d6c3de5aebbbe85939d7588150edf7b7bdc712fceb6a83d79e65b6f79bfc2ef.elf

Figure 4.3: VirusTotal result for Blackbasta

4.2. Hardware, software and network configuration

Host hardware. The experiments will place high demands on system performance, memory and storage space. A laptop with premium specification was available as listed in Table 4.2.

Hardware	**Details**
CPU	AMD Ryzen 7 4800H
Memory	64 GB (2x32GB, 3200 MT/s)
Storage	2 TB
Software	**Details**
Operating System	Ubuntu 23.04
VM Software	Virtual Box 7.0.6

Table 4.2: Laptop system specification

Virtual hardware configuration. As drawn in Figure 4.5, two guest machines were defined in the virtual environment. A machine (Server-Clean) was configured as a typical server and offered various services (e.g. MySQL, Samba shares, FTP, SSH, web server). Another machine was configured as a client (Client-Debian). The ransomware was run on both machines to cover different operating systems. It was ensured that both machines can connect to each other and are isolated in a separate network. A third machine was prepared for retesting purposes without providing a GUI with a very simplified setup to avoid interference. During the experiments, Server-Clean and Client-Debian were active at the same time. Server-Clean-CMD was only used for retesting critical results if necessary.

The guest machine configuration is visible in Table 4.3.

Guest Machine	**System**	**CPU Threads**	**Memory**	**IP Address**	**Purpose**
Server-Clean	Ubuntu 20.04.6 with GUI	2	4 GB	192.168.56.101	Victim Machine
Client-Debain	Debian 12.1.0	2	4 GB	192.168.56.102	Victim Machine
Server-Clean-CMD	Ubuntu 20.04.6 without GUI	2	4 GB		Retest, confirm critical findings

Table 4.3: Virtual machine configuration

The architecture of the software stack is shown in Figure 4.4.

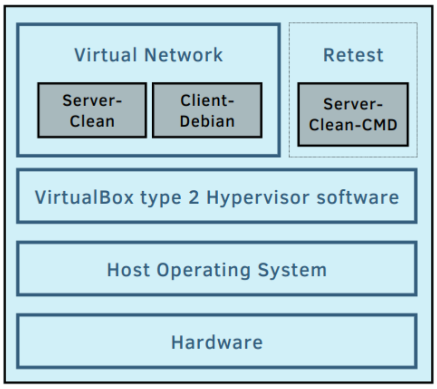


Figure 4.4: Software stack architecture

The virtual test environment had no connection to the Internet. During each test, all Wi-Fi passwords on the host machine were deleted and the host machine was not connected to any network during testing. The virtual box network interface was disabled to avoid outbreaks from the virtual test environment. The retest environment was allowed to connect to the Internet to track certain outcomes (e.g., communication with a command and control server).

The network topology is shown in Figure 4.5. The main test environment was used for the experiments. The retest environment was used to reconfirm critical results. Some experiments showed unexpected results. Therefore, another testing machine was needed to perform repeat testing at a strictly reduced complexity.

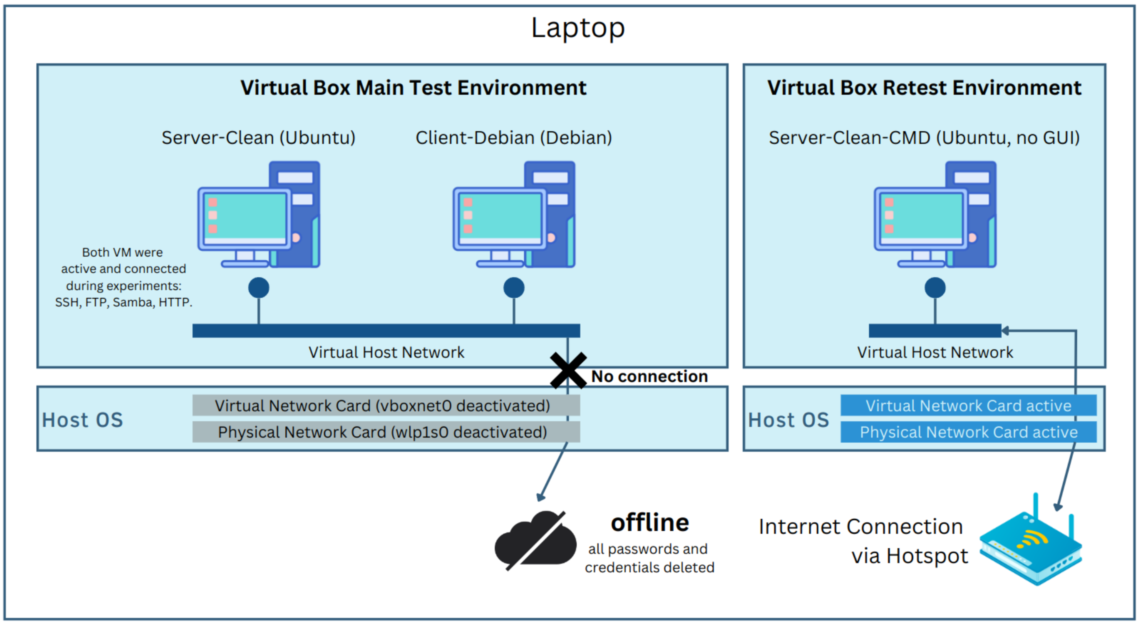

Laptop
Virtual Box Main Test Environment
Server-Clean (Ubuntu)
Client-Debian (Debian)
Both VM were active and connected during experiments: SSH, FTP, Samba, HTTP.
Virtual Host Network
No connection
Host OS
Virtual Network Card (vboxnet0 deactivated)
Physical Network Card (wlp1s0 deactivated)
offline
all passwords and credentials deleted
Virtual Box Retest Environment
Server-Clean-CMD (Ubuntu, no GUI)
Virtual Host Network
Host OS
Virtual Network Card active
Physical Network Card active
Internet Connection via Hotspot

Figure 4.5: Virtual Box main test environment and retest environment

The retest environment was a single virtual machine with no graphical user interface (GUI). This machine was very simplified and created to avoid interferences.

4.3. Experiment execution and data acquisition

Before each test, the VirtualBox configuration was adjusted that the filename of the memory dump contains the name of the ransomware sample as shown in the following command. This made the identification and management of the files easier.

Activation of virtual machine network trace

```
# VBoxManage modifyvm "Client-Debian" --nictrace1 on --nictracefile1 Client-Debian-
blackbasta-networktrace.pcap
# VBoxManage modifyvm "Server-Clean" --nictrace1 on --nictracefile1 Server-Clean-
blackbasta-networktrace.pcap
```

A small script was created that unloads the virtual machine memory after 5 seconds and then every 30 seconds, as shown in Appendix B.2. The last dump took place after 15 minutes.

The Virtual Box Machine system network interface has been disabled on the host to further improve isolation, as shown below.

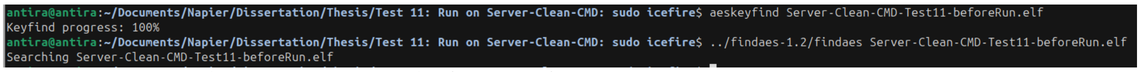
Deactivation of virtual network interface

```
# sudo ifconfig vboxnet0 down
```

5. Results evaluation and comparative analysis

5.1. Experiment 1 – Is the key in memory

Before executing the ransomware, the memory was dumped initially to ensure a clean situation. This pretest dump showed that there are no AES keys in the memory before start of the experiment, as shown as an example in Figure 5.1 below.

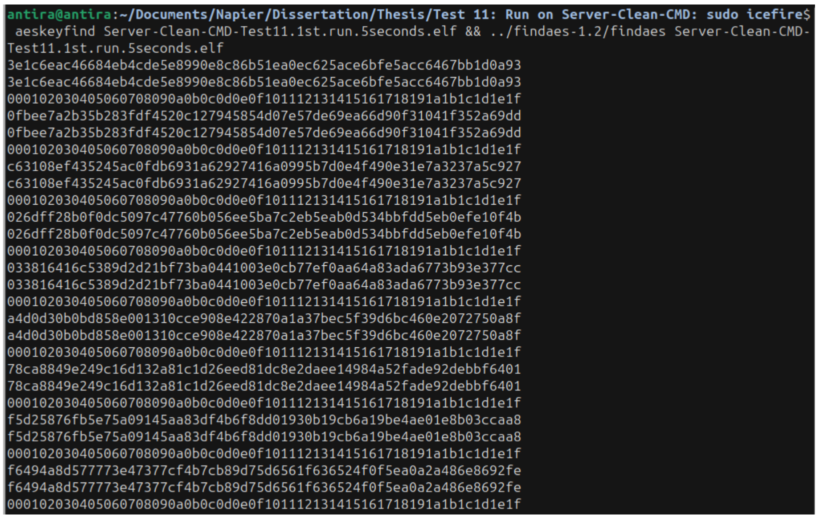


```
antira@antira:~/Documents/Napier/Dissertation/Thesis/Test 11: Run on Server-Clean-CMD: sudo icefire$ aeskeyfind Server-Clean-CMD-Test11-beforeRun.elf
Keyfind progress: 100%
antira@antira:~/Documents/Napier/Dissertation/Thesis/Test 11: Run on Server-Clean-CMD: sudo icefire$ ../findaes-1.2/findaes Server-Clean-CMD-Test11-beforeRun.elf
Searching Server-Clean-CMD-Test11-beforeRun.elf
```

Figure 5.1: No keys in memory before Icefire ransomware execution

5.1.1. Icefire

During execution, several AES keys were loaded into RAM by the ransomware, as shown in Figure 5.2. Both aeskeyfind and findaes found the same AES keys in memory.

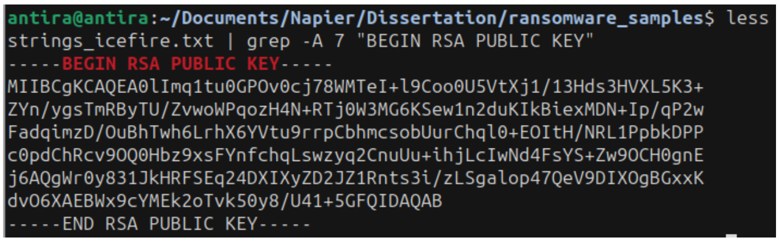


```
antira@antira:~/Documents/Napier/Dissertation/Thesis/Test 11: Run on Server-Clean-CMD: sudo icefire$
 aeskeyfind Server-Clean-CMD-Test11.1st.run.5seconds.elf && ../findaes-1.2/findaes Server-Clean-CMD-
Test11.1st.run.5seconds.elf
3e1c6eac46684eb4cde5e8990e8c86b51ea0ec625ace6bfe5acc6467bb1d0a93
3e1c6eac46684eb4cde5e8990e8c86b51ea0ec625ace6bfe5acc6467bb1d0a93
000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
0fbee7a2b35b283fdf4520c127945854d07e57de69ea66d90f31041f352a69dd
0fbee7a2b35b283fdf4520c127945854d07e57de69ea66d90f31041f352a69dd
000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
c63108ef435245ac0fdb6931a62927416a0995b7d0e4f490e31e7a3237a5c927
c63108ef435245ac0fdb6931a62927416a0995b7d0e4f490e31e7a3237a5c927
000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
026dff28b0f0dc5097c47760b056ee5ba7c2eb5eab0d534bbfdd5eb0efe10f4b
026dff28b0f0dc5097c47760b056ee5ba7c2eb5eab0d534bbfdd5eb0efe10f4b
000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
033816416c5389d2d21bf73ba0441003e0cb77ef0aa64a83ada6773b93e377cc
033816416c5389d2d21bf73ba0441003e0cb77ef0aa64a83ada6773b93e377cc
000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
a4d0d30b0bd858e001310cce908e422870a1a37bec5f39d6bc460e2072750a8f
a4d0d30b0bd858e001310cce908e422870a1a37bec5f39d6bc460e2072750a8f
000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
78ca8849e249c16d132a81c1d26eed81dc8e2daee14984a52fade92debbf6401
78ca8849e249c16d132a81c1d26eed81dc8e2daee14984a52fade92debbf6401
000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
f5d25876fb5e75a09145aa83df4b6f8dd01930b19cb6a19be4ae01e8b03ccaa8
f5d25876fb5e75a09145aa83df4b6f8dd01930b19cb6a19be4ae01e8b03ccaa8
000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
f6494a8d577773e47377cf4b7cb89d75d6561f636524f0f5ea0a2a486e8692fe
f6494a8d577773e47377cf4b7cb89d75d6561f636524f0f5ea0a2a486e8692fe
000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
```

Figure 5.2: Icefire memory dump contains 9 AES keys

The second snapshot taken 30 seconds later showed no AES keys in memory. After the ransomware completed the encryption, the keys were lost forever. The observations showed that the ransomware generated various AES keys and encrypted the AES keys for the specific file with the hard-coded RSA key in the ELF executable. This protection measure makes it difficult to identify the correct AES key for each file during forensic investigations. The encryption cannot be traced back to a single key. The extraction of a single key is therefore almost worthless as the other keys remain unknown.

The Icefire ransomware is delivered with a hard-coded RSA public key, as shown in Figure 5.3.

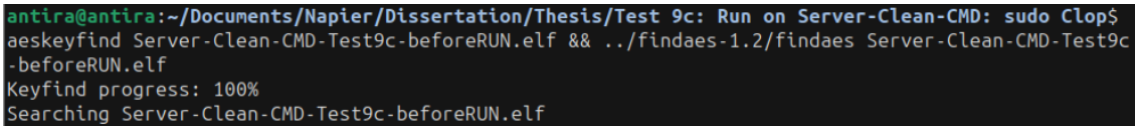


```
antira@antira:~/Documents/Napier/Dissertation/ransomware_samples$ less
strings_icefire.txt | grep -A 7 "BEGIN RSA PUBLIC KEY"
-----BEGIN RSA PUBLIC KEY-----
MIIBCgKCAQEA0lImq1tu0GPOv0cj78WMTeI+l9Coo0U5VtXj1/13Hds3HVXL5K3+
ZYn/ygsTmRByTU/ZvwoWPqozH4N+RTj0W3MG6KSew1n2duKIkBiexMDN+Ip/qP2w
FadqimzD/OuBhTwh6LrhX6YVtu9rrpCbhmcsobUurChql0+EOItH/NRL1PpbkDPP
c0pdChRcv9OQ0Hbz9xsFYnfchqLswzyq2CnuUu+ihjLcIwNd4FsYS+Zw9OCH0gnE
j6AQgWr0y831JkHRFSEq24DXIXyZD2JZ1Rnts3i/zLSgalop47QeV9DIXOgBGxxK
dvO6XAEBWx9cYMEk2oTvk50y8/U41+5GFQIDAQAB
-----END RSA PUBLIC KEY-----
```

Figure 5.3: Icefire hardcoded RSA public key

5.1.2. Cl0p

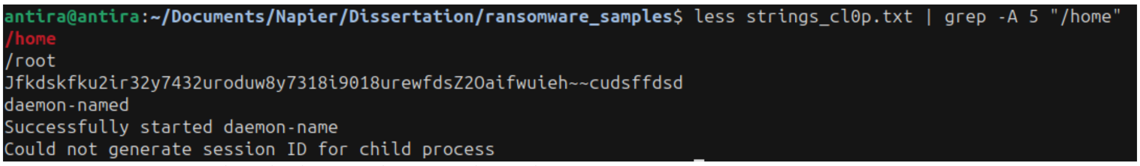
Before executing the ransomware, the memory was dumped to ensure a clean situation. The pretest dump showed no AES keys in memory, as shown in Figure 5.4 below.



```
antira@antira:~/Documents/Napier/Dissertation/Thesis/Test 9c: Run on Server-Clean-CMD: sudo Clop$
aeskeyfind Server-Clean-CMD-Test9c-beforeRUN.elf && ../findaes-1.2/findaes Server-Clean-CMD-Test9c
-beforeRUN.elf
Keyfind progress: 100%
Searching Server-Clean-CMD-Test9c-beforeRUN.elf
```

Figure 5.4: No keys in memory before Cl0p ransomware execution

All dumps captured showed no AES keys in memory. Analysis of the ELF executable revealed that the ransomware uses a hard-coded RC4 symmetric key to encrypt the files, as shown in Figure 5.5 and confirmed by other researchers (Terefos, 2023).

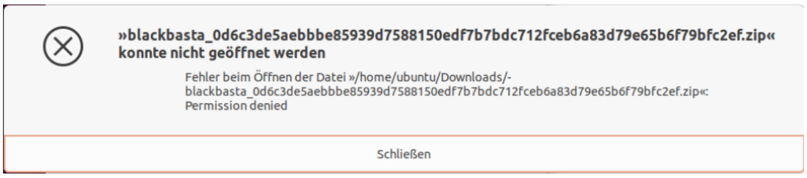


```
antira@antira:~/Documents/Napier/Dissertation/ransomware_samples$ less strings_cl0p.txt | grep -A 5 "/home"
/home
/root
Jfkdskfku2ir32y7432uroduw8y7318i9018urewfdsZ2Oaifwuieh~~cudsffdsd
daemon-named
Successfully started daemon-name
Could not generate session ID for child process
```

Figure 5.5: Hardcoded RC4 key in the Cl0p ransomware

According to Terefos the Cl0p Linux ransomware variant uses individual RC4 keys for each file (Terefos, 2023).

This example required administrative privileges to successfully encrypt the files on the system. In the first experiment cycle, the example was run as a standard "Ubuntu" user. The example successfully revoked access permissions for files, but did not encrypt them. Navigation through the directories was no longer possible. Attempts to open some files resulted in a "Permission denied" error message. As shown in Figure 5.6, it was no longer possible to open another ZIP file (Blackbasta) because the permissions were revoked.

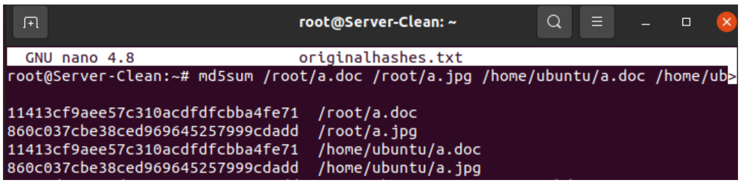


»blackbasta_0d6c3de5aebbbe85939d7588150edf7b7bdc712fceb6a83d79e65b6f79bfc2ef.zip« konnte nicht geöffnet werden

Fehler beim Öffnen der Datei »/home/ubuntu/Downloads/-blackbasta_0d6c3de5aebbbe85939d7588150edf7b7bdc712fceb6a83d79e65b6f79bfc2ef.zip«: Permission denied

Figure 5.6: Cl0p ransomware manipulates permissions before encryption

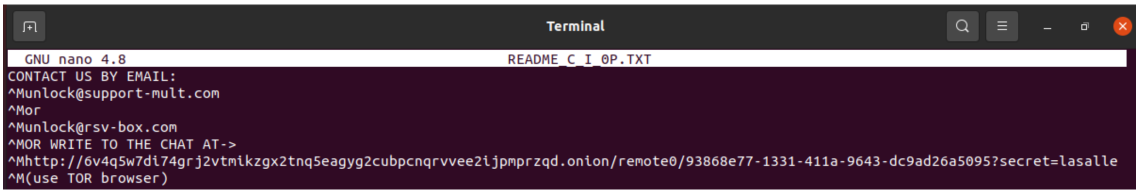
Comparing the hashes of the a.jpg and a.doc files revealed that the files have the same original hash values, which is evidence that encryption was not present, as shown in Figure 5.7.



```
  GNU nano 4.8                  originalhashes.txt
root@Server-Clean:~# md5sum /root/a.doc /root/a.jpg /home/ubuntu/a.doc /home/ub>

11413cf9aee57c310acdfdfcbba4fe71  /root/a.doc
860c037cbe38ced969645257999cdadd  /root/a.jpg
11413cf9aee57c310acdfdfcbba4fe71  /home/ubuntu/a.doc
860c037cbe38ced969645257999cdadd  /home/ubuntu/a.jpg
```

Figure 5.7: Cl0p ransomware did not encrypt the files when run as normal user

The experiments were repeated with administrative privileges (i.e., `sudo`). After this customisation, the ransomware successfully encrypted the files and left the ransom message as a text file in the `/home` directory, but not in `/home/ubuntu`, as shown in Figure 5.8.

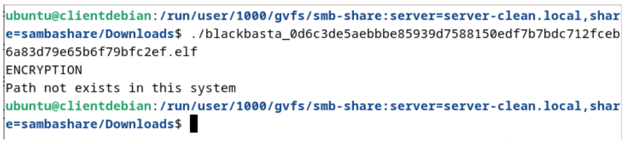


```
  GNU nano 4.8                                    README_C_I_0P.TXT
CONTACT US BY EMAIL:
^Munlock@support-mult.com
^Mor
^Munlock@rsv-box.com
^MOR WRITE TO THE CHAT AT->
^Mhttp://6v4q5w7di74grj2vtmikzgx2tnq5eagyg2cubpcnqrvvee2ijpmprzqd.onion/remote0/93868e77-1331-411a-9643-dc9ad26a5095?secret=lasalle
^M(use TOR browser)
```

Figure 5.8: Cl0p ransom message

5.1.3. Blackbasta

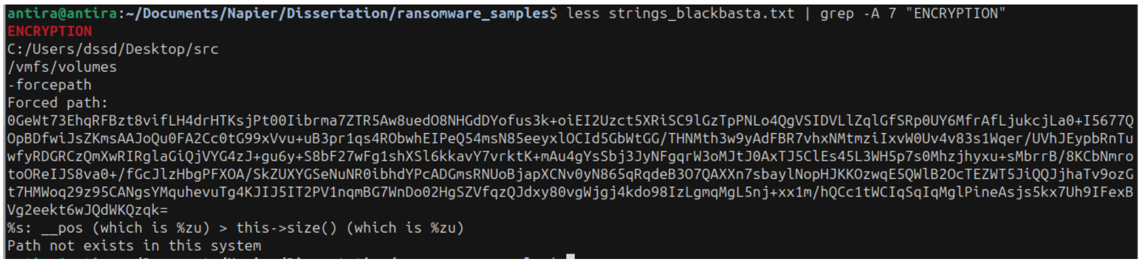
The Blackbasta ransomware did not run in the initial experiments and resulted in a "*Path does not exist in this system*" error on the command line, as shown in Figure 5.9. Once executed, there was no impact and no encryption was detected on the system.



```
ubuntu@clientdebian:/run/user/1000/gvfs/smb-share:server=server-clean.local,shar
e=sambashare/Downloads$ ./blackbasta_0d6c3de5aebbbe85939d7588150edf7b7bdc712fceb
6a83d79e65b6f79bfc2ef.elf
ENCRYPTION
Path not exists in this system
ubuntu@clientdebian:/run/user/1000/gvfs/smb-share:server=server-clean.local,shar
e=sambashare/Downloads$ 
```

Figure 5.9: Blackbasta requires predefined path for encryption

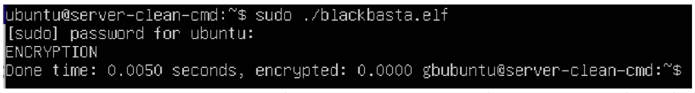
An analysis of the ELF strings revealed that the ransomware requires the `/vmfs/volumes` folder, as shown in Figure 5.10 below and confirmed by other researchers (Umawing, 2022).



```
antira@antira:~/Documents/Napier/Dissertation/ransomware_samples$ less strings_blackbasta.txt | grep -A 7 "ENCRYPTION"
ENCRYPTION
C:/Users/dssd/Desktop/src
/vmfs/volumes
-forcepath
Forced path:
0GeWt73EhqRFBzt8vifLH4drHTKsjPt00Iibrma7ZTR5Aw8uedO8NHGdDYofus3k+oiEI2Uzct5XRiSC9lGzTpPNLo4QgVSIDVLlZqlGfSRp0UY6MfrAfLjukcjLa0+I5677Q
OpBDfwiJsZKmsAAJoQu0FA2Cc0tG99xVvu+uB3pr1qs4RObwhEIPeQ54msN85eeyxlOCId5GbWtGG/THNMth3w9yAdFBR7vhxNMtmziIxvW0Uv4v83s1Wqer/UVhJEypbRnTu
wfyRDGRCzQmXwRIRglaGiQjVYG4zJ+gu6y+S8bF27wFg1shXSl6kkavY7vrktK+mAu4gYsSbj3JyNFgqrW3oMJtJ0AxTJ5ClEs45L3WH5p7s0Mhzjhyxu+sMbrrB/8KCbNmro
toOReIJS8va0+/fGcJlzHbgPFXOA/SkZUXYGSeNuNR0ibhdYPcADGmsRNUoBjapXCNv0yN865qRqdeB3O7QAXXn7sbaylNopHJKKOzwqE5QWlB2OcTEZWT5JiQQJjhaTv9ozG
t7HMWoq29z95CANgsYMquhevuTg4KJIJ5IT2PV1nqmBG7WnDo02HgSZVfqzQJdxy80vgWjgj4kdo98IzLgmqMgL5nj+xx1m/hQCc1tWCIqSqIqMglPineAsjs5kx7Uh9IFexB
Vg2eekt6wJQdWKQzqk=
%s: __pos (which is %zu) > this->size() (which is %zu)
Path not exists in this system
```

Figure 5.10: Strings analysis of Blackbasta ELF executable

This folder was created manually to ensure the prerequisites of the ransomware. The tests were repeated with the existence of the `/vmfs/volumes` folder. After this adaption of the environment, the ransomware started successfully as seen in Figure 5.11 below.



```
ubuntu@server-clean-cmd:~$ sudo ./blackbasta.elf
[sudo] password for ubuntu:
ENCRYPTION
Done time: 0.0050 seconds, encrypted: 0.0000 gbubuntu@server-clean-cmd:~$
```

Figure 5.11: Successful encryption after added /vmfs/volumes folder

The Blackbasta ransomware started the encryption and encrypted the files under */vmfs/volumes*. No AES keys were detected in the RAM. After analysing the strings of the ELF executable, no indicators were found that could lead to a conclusion, that the AES cipher were implemented. Instead, ChaCha20 cipher were identified as underlying cipher. Other researchers confirmed this finding (Umawing, 2022). The selected tools were not able to identify any keys related in the memory.

5.1.4. Comparative analysis

Davies demonstrated that all 3 Windows ransomware samples have used AES based encryption (Davies, 2020). Thus it was possible to extract the keys using standard tools. In contrast, this project showed that different algorithms were used in all Linux ransomware samples, as shown in Table 5.1 below. Icefire used AES in combination with RSA and generated different AES keys during encryption. Cl0p used a hardcoded RC4 master-key and Blackbasta used ChaCha20. A proposal for the determination of ChaCha20 key material in memory dumps was found (McLaren et al., 2019), but reliable implementations are pending. The results were surprising and not expected.

Research	**Ransomware for Windows by Davies (2020)**			**Ransomware for Linux by the author**		
Sample	NotPetya	Bad Rabbit	Phobos	Icefire	Cl0p	Blackbasta
Key found?	Yes	Yes	Yes	Yes	No (not in RAM, but hardcoded)	No
Symmetric cipher	AES	AES	AES	AES	RC4	ChaCha20
Number of keys generated	*1*	*1*	*1 + n*	*n*	*n*	unknown
	(global key for all files)	(global key for all files)	(initial encryption with global key, new files encrypted with individual keys)	(different keys, for each file)	(different keys, for each file)	

Table 5.1: Comparative analysis for experiment 1

5.2. Experiment 2 – How long is the key present

The Icefire ransomware has shown that the AES keys were deleted from memory after encryption of the specific file is complete. While the first dump showed 9 different AES keys immediately after execution, the second dump 30 seconds later showed no AES keys in memory. For the Cl0p ransomware, analysis of the duration of key presence is not relevant because the RC4 key was hardcoded in the ELF binary. For Blackbasta, an analysis of the duration of the key was not measurable because the literature search for the underlying operating system did not find any ready-to-use forensic tools for ChaCha20 key determination.

5.2.1. Comparative analysis

The duration of the key presence is shown in Table 5.2.

Research	**Ransomware for Windows by Davies (2020)**			**Ransomware for Linux by the author**		
Sample	NotPetya	Bad Rabbit	Phobos	Icefire	Cl0p	Blackbasta
Time until key becomes present	< 2 minutes	< 1 minute	< 1 minute	< 5 seconds	hardcoded	unknown
Key presence duration	until reboot	removed after whole encryption completed	removed when ransom message displayed	removed for each file after encryption	hardcoded	unknown

Table 5.2: Comparative analysis for experiment 2

5.3. Experiment 3 – Does the key decrypt the files

5.3.1. Icefire

This study evaluated a similar decryption approach demonstrated by other security researchers (Davies, 2020). Unlike Davies' studies, the position of the initialisation vector (IV) and the structure of the encrypted files were unknown in this work. Additionally, each file was encrypted with an individual AES key. The scripts developed by Davies and the assumptions regarding the position of the IV were not successful in this work.

The decryption experiment was very complex because the ransomware encrypted each file with an individual key and removed this key immediately after encryption. Therefore, there is no way to ensure that the keys obtained in the dump were the correct ones. In the experiments, the ransomware encrypted several hundred files, as opposed to only nine AES keys extracted. Decryption was complicated by various factors. Only 9 AES keys could be extracted from storage. Identifying the corresponding file was a major challenge, as was identifying the IV length and location.
Given the time available and the other goals of this project, the decryption attempt ended without a positive result. Future research is recommended in Section 6.4.

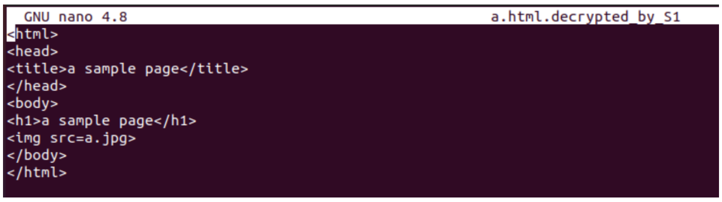
5.3.2. Cl0p

The Cl0p ransomware was found to use a hardcoded RC4 master-key. A published decryptor was used to successfully decrypt the files (Sentinel One, 2023). First, it was needed to identify all encrypted files. In a second step, the decryptor was executed to decrypt the files like shown below.

Command to run Cl0p decryptor

```
# find / -name *.$cl0p_extension -print 2>/dev/null > cl0p_keys.txt
python3 clop_linux_file_decr.py --elfie clop.elf --keys cl0p_keys.txt
```

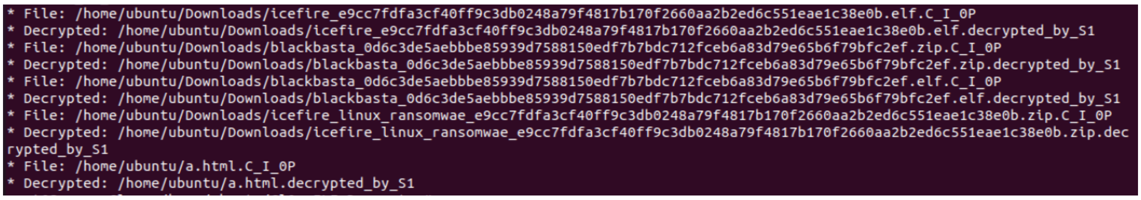
The decryptor identified the RC4 key from the binary and successfully decrypted the files, as proofed for `a.html.decrypted_by_S1` in Figure 5.12.



```
  GNU nano 4.8                              a.html.decrypted_by_S1
<html>
<head>
<title>a sample page</title>
</head>
<body>
<h1>a sample page</h1>
<img src=a.jpg>
</body>
</html>
```

Figure 5.12: HTML file successfully decrypted

The decryptor saved each file with the extension “decrypted_by_S1”, as shown in Figure 5.13.



```
* File: /home/ubuntu/Downloads/icefire_e9cc7fdfa3cf40ff9c3db0248a79f4817b170f2660aa2b2ed6c551eae1c38e0b.elf.C_I_0P
* Decrypted: /home/ubuntu/Downloads/icefire_e9cc7fdfa3cf40ff9c3db0248a79f4817b170f2660aa2b2ed6c551eae1c38e0b.elf.decrypted_by_S1
* File: /home/ubuntu/Downloads/blackbasta_0d6c3de5aebbbe85939d7588150edf7b7bdc712fceb6a83d79e65b6f79bfc2ef.zip.C_I_0P
* Decrypted: /home/ubuntu/Downloads/blackbasta_0d6c3de5aebbbe85939d7588150edf7b7bdc712fceb6a83d79e65b6f79bfc2ef.zip.decrypted_by_S1
* File: /home/ubuntu/Downloads/blackbasta_0d6c3de5aebbbe85939d7588150edf7b7bdc712fceb6a83d79e65b6f79bfc2ef.elf.C_I_0P
* Decrypted: /home/ubuntu/Downloads/blackbasta_0d6c3de5aebbbe85939d7588150edf7b7bdc712fceb6a83d79e65b6f79bfc2ef.elf.decrypted_by_S1
* File: /home/ubuntu/Downloads/icefire_linux_ransomwae_e9cc7fdfa3cf40ff9c3db0248a79f4817b170f2660aa2b2ed6c551eae1c38e0b.zip.C_I_0P
* Decrypted: /home/ubuntu/Downloads/icefire_linux_ransomwae_e9cc7fdfa3cf40ff9c3db0248a79f4817b170f2660aa2b2ed6c551eae1c38e0b.zip.dec
rypted_by_S1
* File: /home/ubuntu/a.html.C_I_0P
* Decrypted: /home/ubuntu/a.html.decrypted_by_S1
```

Figure 5.13: Execution of the Cl0p RC4 decryptor

5.3.3. Blackbasta

The decryption of Blackbasta was impossible, because the ChaCha20 key extraction failed.

5.3.4. Comparative analysis

The results were mainly compared to similar research for Windows (Davies, 2020). As Table 5.3 shows below, Davies achieved to decrypt the files for all Windows ransomware variants. For Linux, only for 1 Linux ransomware the decryption was successful The decryption for Icefire and Blackbasta failed. The reason for this is that Icefire uses a unique key for each file. The determination of the IV position also failed. Blackbasta used ChaCha20 symmetric ciphers.

For ChaCha20 no reliable, ready-to-use implementations of forensic tools were found.

Research	**Ransomware for Windows by Davies (2020)**			**Ransomware for Linux by the author**		
Sample	NotPetya	Bad Rabbit	Phobos	Icefire	Cl0p	Blackbasta
Files decrypted?	Yes	Yes	Yes	No	Yes	No
						But, decryption possible according to other research (Lamsouber, 2023)

Table 5.3: Comparative analysis for experiment 3

During the completion of this project, a BSc thesis related to the Blackbasta ransomware example was discovered. Lamsouber was able to successfully revert the Blackbasta encryption with a public decryptor (Lamsouber, 2023). His work enabled a comparative analysis with Experiment 3 and covered the execution of one ransomware sample, setting up a test environment, and a corresponding decryption attempt. The creation date of the PDF document is November 14, 2023.

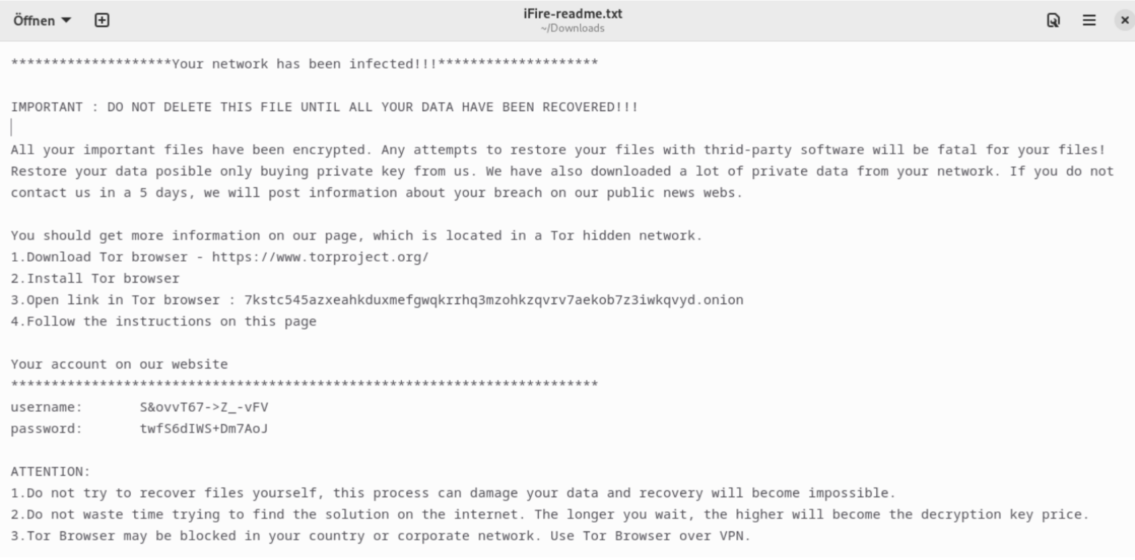
Since both this work and Lamsouber were based on the same Blackbasta ransomware with identical checksum values, it is reliable that his approach would also be successful in this work.

5.4. Experiment 4 – Does the sample spread throughout the network

The network was recorded outside the virtual machine to avoid the risk of detection and evasion by the ransomware samples. The network was isolated as described in Section 4 and contained a client and a server with different services (e.g. SSH, Web server, Samba) without hardened security. During each execution, the client was connected to the server and the latest package ID in Wireshare was noted beforehand.

5.4.1. Icefire

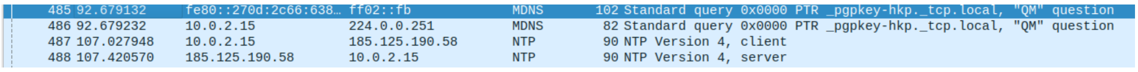
Attempts to communicate with command and control server were not detected. Regardless of whether the ransomware was running on the server or the client, the network scan did not identify any signs of compromise. As shown in Figure 5.14, the ransomware states that the network has been infected. However, there is no evidence that the ransomware is capable of spreading laterally through the network and services. The services and files on the other system in the network continued to run without disruption and were not encrypted. The experiment showed that the ransomware was specifically designed to encrypt the files after the first delivery on the respective computer.



```
*******************Your network has been infected!!!*******************

IMPORTANT : DO NOT DELETE THIS FILE UNTIL ALL YOUR DATA HAVE BEEN RECOVERED!!!

All your important files have been encrypted. Any attempts to restore your files with thrid-party software will be fatal for your files!
Restore your data posible only buying private key from us. We have also downloaded a lot of private data from your network. If you do not
contact us in a 5 days, we will post information about your breach on our public news webs.

You should get more information on our page, which is located in a Tor hidden network.
1.Download Tor browser - https://www.torproject.org/
2.Install Tor browser
3.Open link in Tor browser : 7kstc545azxeahkduxmefgwqkrrhq3mzohkzqvrv7aekob7z3iwkqvyd.onion
4.Follow the instructions on this page

Your account on our website
***********************************************************************
username:       S&ovvT67->Z_-vFV
password:       twfS6dIWS+Dm7AoJ

ATTENTION:
1.Do not try to recover files yourself, this process can damage your data and recovery will become impossible.
2.Do not waste time trying to find the solution on the internet. The longer you wait, the higher will become the decryption key price.
3.Tor Browser may be blocked in your country or corporate network. Use Tor Browser over VPN.
```

Figure 5.14: Icefire ransom message in iFire-readme.txt

After execution of the ransomware there were no evidence that the ransomware sample was contacting a command and control server as seen in Figure 5.15.



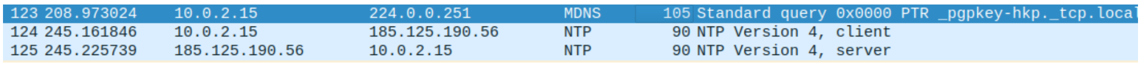
```
485 92.679132    fe80::270d:2c66:638… ff02::fb         MDNS      102 Standard query 0x0000 PTR _pgpkey-hkp._tcp.local, "QM" question
486 92.679232    10.0.2.15            224.0.0.251      MDNS       82 Standard query 0x0000 PTR _pgpkey-hkp._tcp.local, "QM" question
487 107.027948   10.0.2.15            185.125.190.58   NTP        90 NTP Version 4, client
488 107.420570   185.125.190.58       10.0.2.15        NTP        90 NTP Version 4, server
```

Figure 5.15: Icefire ransomware did not contact any command and control servers

5.4.2. Cl0p

Communication attempts to command and control servers were not detected. Independently if the ransomware was run on the server or on the client, no indicators of compromise were identified in the network capture. The services and files on the other network system continued to run without impact and were not encrypted. The experiment demonstrated that the ransomware was build to encrypt the files on the specific machine, after initial delivery and was not able to move laterally utilising further exploitation techniques.

As seen in Figure 5.16, there was no communication to external servers.

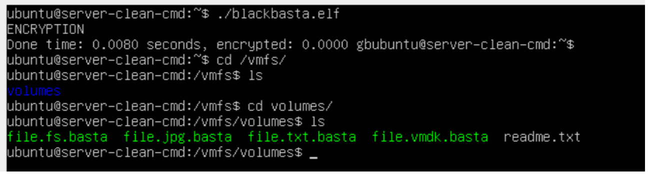


```
123 208.973024   10.0.2.15        224.0.0.251      MDNS      105 Standard query 0x0000 PTR _pgpkey-hkp._tcp.local
124 245.161846   10.0.2.15        185.125.190.56   NTP        90 NTP Version 4, client
125 245.225739   185.125.190.56   10.0.2.15        NTP        90 NTP Version 4, server
```

Figure 5.16: Cl0p ransomware did not contact any command and control servers

5.4.3. Blackbasta

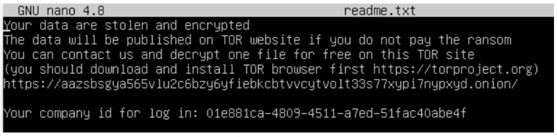
Blackbasta ransomware is a specialised malware for VMWare ESXi. The ransomware successfully encrypted all files under `/vmfs/volumes`, including non-VMFS files (e.g. file.txt, file.jpg), as shown in Figure 5.17.



```
ubuntu@server-clean-cmd:~$ ./blackbasta.elf
ENCRYPTION
Done time: 0.0080 seconds, encrypted: 0.0000 gbubuntu@server-clean-cmd:~$
ubuntu@server-clean-cmd:~$ cd /vmfs/
ubuntu@server-clean-cmd:/vmfs$ ls
volumes
ubuntu@server-clean-cmd:/vmfs$ cd volumes/
ubuntu@server-clean-cmd:/vmfs/volumes$ ls
file.fs.basta  file.jpg.basta  file.txt.basta  file.vmdk.basta  readme.txt
ubuntu@server-clean-cmd:/vmfs/volumes$ _
```

Figure 5.17: Blackbasta successfully encrypted all files in volumes folder

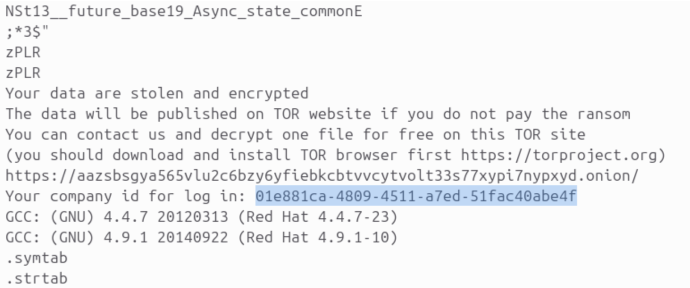
As shown in Figure 5.18, blackbasta has created a readme.txt in `/vmfs/volumes`. Regardless of whether the experiments were repeated, the same company ID was always created. It can be concluded that the ransomware payload is compiled individually for each victim.



```
  GNU nano 4.8                                   readme.txt
Your data are stolen and encrypted
The data will be published on TOR website if you do not pay the ransom
You can contact us and decrypt one file for free on this TOR site
(you should download and install TOR browser first https://torproject.org)
https://aazsbsgya565vlu2c6bzy6yfiebkcbtvvcytvolt33s77xypi7nypxyd.onion/

Your company id for log in: 01e881ca-4809-4511-a7ed-51fac40abe4f
```

Figure 5.18: Blackbasta ransomware places a company id in the readme.txt

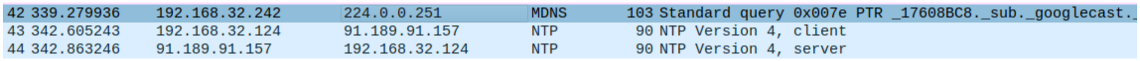
A `strings` analysis revealed that the company ID is hardcoded in the ELF binary, as shown in Figure 5.19.



```
NSt13__future_base19_Async_state_commonE
;*3$"
zPLR
zPLR
Your data are stolen and encrypted
The data will be published on TOR website if you do not pay the ransom
You can contact us and decrypt one file for free on this TOR site
(you should download and install TOR browser first https://torproject.org)
https://aazsbsgya565vlu2c6bzy6yfiebkcbtvvcytvolt33s77xypi7nypxyd.onion/
Your company id for log in: 01e881ca-4809-4511-a7ed-51fac40abe4f
GCC: (GNU) 4.4.7 20120313 (Red Hat 4.4.7-23)
GCC: (GNU) 4.9.1 20140922 (Red Hat 4.9.1-10)
.symtab
.strtab
```

Figure 5.19: Blackbasta ransomware contains hardcoded company ID

Once executed, no signs were detected that the Blackbasta ransomware contacted a command and control server or attempted to move laterally through the network. Network tracking was observed in real time. Immediately after execution, the machine contacted the Canonical NTP service, as shown in Figure 5.20.



```
42 339.279936    192.168.32.242    224.0.0.251       MDNS     103 Standard query 0x007e PTR _17608BC8._sub._googlecast._
43 342.605243    192.168.32.124    91.189.91.157     NTP       90 NTP Version 4, client
44 342.863246    91.189.91.157     192.168.32.124    NTP       90 NTP Version 4, server
```

Figure 5.20: Blackbasta did not contact any command and control servers

There were no attempts to communicate with other external IP addresses.
This is in line with the targeted shipping method and the assignment of an individual company ID. Most likely, the cybercriminals will use other means to ensure that the ransomware runs successfully.

5.4.4. Comparative analysis

The analysis in Table 5.4 showed that all ransomware samples did not attempt to spread across the network. The tests were repeated in the retest environment to ensure the accuracy of the results. The same behavior for all 3 samples was confirmed by the retests. To summarise, the ransomware samples were primarily designed to perform the encryption step for a specific victim. Delivery, lateral movement and feedback of successful encryption therefore occur outside of ransomware capabilities.

Research	**Linux ransomware research by the author**		
Ransomware Sample	**Icefire**	**Cl0p**	**Blackbasta**
Communication to command and control server?	No	No	No
Indicators of lateral movement?	No	No	No
SSH attacked?	No	No	No
FTP attacked?	No	No	No
Web services attacked?	No	No	No
Samba attacked?	No	No	No

Table 5.4: Comparative analysis for experiment 4: Linux ransomware

The results were also compared with the results for Windows Active Directory domain services (McDonald et al., 2022). As shown in table 5.5, Windows ransomware was capable to detect attached file shares and to encrypt the data inside. Linux ransomware was not capable of doing this. WannaCry is known to contact command and control servers (C&C). If this step fails, the ransomware usually starts the encryption immediately (McDonald et al., 2022). Communication to C&C servers is also confirmed for TeslaCrypt (Skuratovich, 2016) and Jigsaw ransomware (Ashdown, 2021).

Research	**Ransomware for Windows by McDonald et al. (2022), Skuratovic (2016) and Ashdown (2021)**			**Ransomware for Linux by the author**		
Sample	WannaCry	TeslaCrypt	Jigsaw	Icefire	Cl0p	Blackbasta
Communication to C&C server?	Yes	Yes	Yes	No	No	No
Files encrypted in file shares?	Yes	Yes	Yes	No	No	No
Negative impact on network file shares	No	No	No	No	No	No
Impact on network services (e.g. DNS, DHCP)	No	No	No	No	No	No

Table 5.5: Comparative analysis for experiment 4: Comparison with Windows ransomware

5.5. Experiment 5 – What was the impact on the system

5.5.1. Icefire

The Icefire ransomware encrypted user data and some system configuration files. The Pulseaudio and Desktop File Search Service tracker-store crashed. The ransomware creates an individual symmetric key for each file. Icefire encrypted the original files and saved them, along with the encrypted symmetric key, as a new ".iFire" file. When Icefire was run as a normal user, the encryption focused only on the `$HOME` folder. With `root` privileges, the ransomware also included `/root` and specific system data folders in `/usr`. Other folders or databases were not encrypted (e.g. MySQL, `/var/www`). The behaviour was the same on both Debian and Ubuntu.

5.5.2. Cl0p

The impact of Cl0p ransomware depended heavily on the permissions of the user. When the ransomware was run with the normal user, it had limited impact on the system. The Cl0p ransomware started by manipulating permissions, but the encryption step failed as long as it was a normal - non-privileged - user.

A re-login via GUI was not possible any-more, because the ransomware also manipulated the user Desktop configuration files and `bashrc`. The password was still detected as valid or invalid, but the login was not successful. No files were put on the file system and no file extensions were changed. All files were able to be opened when accessed with privileged permissions (i.e. sudo/root)

When executed with root permissions, the ransomware successfully achieved to encrypt the original file in `/root` and `/home/$USER`. After successful encryption, the ransomware created a new file with the extension *.C_l_0p* to store the RC4 encrypted file-specific symmetric key.

Other folders or databases were not encrypted (e.g. `MySQL`, `/var/www`). The behaviour was same on Debian as well as Ubuntu.

5.5.3. Blackbasta

Blackbasta ransomware encrypted all files in `/vmfs/volumes`, regardless of the file extension. This required the user to have permission to `/vmfs/volumes` or `sudo` privileges. When running as a normal user without write permission to `/vmfs/volumes`, encryption was not successful. The ransomware also did not check the file headers to see if they were actually VMWare ESXi files. The Blackbasta ransomware added the `*.basta` extension to all encrypted files. Other data folders (e.g. the user's home folder) were not encrypted even when running as root.

Blackbasta demonstrated, that it was developed to fulfil a certain goal. Even when the ransomware ran with full system privileges, it did not take advantage of these full privileges and limited itself only to `/vmfs/volumes`. After the encryption, the user was still able to login.

Other folders or databases were not encrypted (e.g. `MySQL`, `/var/www`). The behaviour was same on Debian as well as Ubuntu.

5.5.4. Comparative analysis

The impact on the system was limited by the defined target of the ransomware authors. As shown in Table 5.6, the Cl0p and Icefire ransomware encrypted files only in the user directory, even when running with administrative privileges.

Research	**Linux ransomware research by the author**		
Ransomware Sample	**Icefire**	**Cl0p**	**Blackbasta**
Login possible?	Yes	No	Yes
Impact on communication?	No	No	No
Impact on applications?	Yes, Pulseaudio and tracker-store crashed.	Yes, Nautilus and office applications crashed due to manipulated file permissions.	No

Encrypted folders/files if executed as **normal** user	`/home/ubuntu`
	`/Downloads/*`
	`/Documents/*`
	`/Music/*`
	`/Videos/*`
	`/Desktop/*`
	`/Pictures/*`
	`/.cache/`
	`/.mozilla/firefox/*`
	`/.gnupg/*`
	`/.config/pulse/*`
	`/.config/evolution/*`
	`/.config/gtk-3.0/*`
	`/.config/libreoffice/*`
	`/.local/share/*`
		none, only manipulated permissions	none

Encrypted folders/files if executed as **root** user	`+`
	`/etc/gdm3/*`
	`/usr/share/cups/*`
	`/usr/share/doc/*`
	`/usr/share/groff/*`
	`/usr/src/*`
	`/usr/lib/libreoffice/*`
	`/root/*`
	`/root/.local/share*`
	`/root/.cache/tracker/*`
		`/root/*`
		`/snap/*`
		`/.profile`
		`/.config/dconf/*`
		`/.config/enchant/*`
		`/.dbus/*`
		`/.cache/tracker/*`
		`/.bashrc`
		`/home/ubuntu`
		`/Downloads/*`
		`/Documents/*`
		`/Music/*`
		`/Videos/*`
		`/Desktop/*`
		`/Pictures/*`
		`/.mozilla/firefox/*`
		`/.config/libreoffice/*`
			`/vmfs/volumes/*`

		`/.config/dconf/*`
		`/.config/enchant/*`
		`/.cache/*`
		`/.bashrc`
		`/.profile`
		`/.dbus/*`	

Table 5.6: Comparative analysis for experiment 5: Linux ransomware

All ransomware samples did not utilise the full potential of the permissions being executed. In all cases, the ransomware was only executed according to the use cases predefined by the ransomware authors. Even when running with full permissions, important paths were not affected. The MySQL databases, SSH data, FTP data and Samba shares were not affected. But especially in the enterprise sector, it is common practice to connect external databases and storage and not to store any data under `/home` or `/root`.

As shown in Table 5.7 below, Windows ransomware tends to restart after successful encryption and to display a visual message. For Linux such behaviour was not observed.

Research	**Ransomware for Windows by Davies (2020)**			**Ransomware for Linux by the author**		
Sample	**NotPetya**	**Bad Rabbit**	**Phobos**	**Icefire**	**Cl0p**	**Blackbasta**
Automatic reboot?	Yes	Yes	No	No	No	No
Visual ransom message?	Yes - on console	Yes - on console	Yes - in window	No - only text file	No - only text file	No - only text file

Table 5.7: Comparative analysis for experiment 5: Comparison with Windows ransomware

The results were also compared with the research by McDonald et al., who analysed the impact of Windows ransomware on Windows Active Directory Domain Services. As shown in Table 5.8, Windows ransomware had no impact on logon services. Users were still able to log in. Windows ransomware was also able to move laterally to some extent, to detect web server files (e.g. wwwroot of the IIS web server) and encrypt them as well. Both the Linux and Windows ransomware had no impact on the operation of the web server.

Research	**Ransomware for Windows by McDonald et al. (2022)**			**Ransomware for Linux by the author**		
Sample	**WannaCry**	**TeslaCrypt**	**Jigsaw**	**Icefire**	**Cl0p**	**Blackbasta**
Logon possible?	Yes	Yes	Yes	Yes	No	Yes
Web server files encrypted?	Yes	Yes	Yes	No	No	No
Web server still working?	Yes	Yes	Yes	Yes	Yes	Yes

Table 5.8: Comparative analysis for experiment 5: Comparison with Windows ransomware

6. Conclusions

This work began with the hypothesis that the ransomware samples would use variations of RSA and AES keys to encrypt the data, similar to common Windows variants. This hypothesis turned out to be invalid for Linux. The experiments demonstrated diverse key management and encryption methods. Only Icefire ransomware is confirmed to use AES encryption. Cl0p used a hardcoded RC4 symmetric master-key and Blackbasta used ChaCha20 in combination with RSA. The results contradicted those of other researchers compared to Windows operating systems.

As a result, the different use of encryption methods poses considerable difficulties for forensic investigations. Previously proven methods and tools that were mainly used for RSA and AES can no longer be used. Suitable, ready-to-use implementations for obtaining RC4 or Chacha20 keys from memory dumps were not found during the literature search. The attackers diversified the encryption methods.

Another hypothesis was that Linux ransomware is not as sophisticated compared to Windows-based ransomware. This proved to be valid. But surprisingly, the maturity of Linux ransomware fell far short of expectations. The results also showed that the Linux ransomware was not used to its full potential. Even when the ransomware was executed with root privileges, important file paths (e.g. web server content, databases, file shares) remained untouched. The Linux ransomware each serves a specific purpose and leaves out potential. No signs of lateral movement or communication with the command and control servers were detected in all three samples. This suggests that Linux ransomware is distributed through other means and requires significant manual assistance to succeed. Due to the different distributions, Linux ransomware is designed for a specific niche and goal.

6.1. Protective measures

Avoid $HOME directories. To reduce the risk of a ransomware attack, it is recommended not to store important data under `/home/user` or `/root`. Instead, it is advisable to determine unusual, individual paths for data storages. A very unusual but effective measure is to hide important data in system subfolders (e.g. `/bin/*`), as the ransomware most likely needs to keep this folder clear in order to keep the system functional. Mount configurations can also be helpful during setup. The experiments showed that the ransomware precisely defines which folders to encrypt and cannot distinguish anomalies.

Separate and restrict permissions and data access. It also became apparent that the ransomware was unable to independently take over other processes or user rights. This means that the ransomware is limited to the specified user, data and permission scope. Each application should have separate storage space and run with different users. A separate technical user should be created for each application who only has the absolutely necessary rights. Access to the memory of other applications must not be permitted.

Avoid using privileged users. Cl0p and Blackbasta could only encrypt the data if they were run with admin rights. When the Icefire ransomware was run as a normal user, the attack was only limited to the user's specific home directory. Therefore, it is extremely important not to operate or install any software with root privileges.

Focus to identify backdoors. All 3 ransomware variants were unable to spread independently across the network. Other computers on the network were unaffected. This suggests that cybercriminals are targeting the victims and may have already created backdoors to maintain access. Therefore, forensic resources should be focused on identifying such backdoors.

Do not hesitate to shut down systems. Disconnet internet connection centally. Weighing up the chances of forensic investigations and the risks, it is advisable to shut down the infected Linux machine and surrounding systems as quickly as possible and not to act hesitantly. Shutting down the system increases the chances of obtaining the ELF ransomware binary for static analysis. Static analysis in particular proved to be helpful in the experiments.

The chances that the key can be obtained during forensic investigations of the RAM are slim. The risk that further system damage will occur despite the interruption of the network connection remains high. In its current state, Linux ransomware cannot escalate through the network alone. It is very likely that the attackers themselves are responsible for an advanced network infection and are maintaining access through other means. This recommendation depends on the development progress of cybercriminals and must be reviewed again by the end of 2025 at the latest.

6.2. Aims and Objectives

The original goals of this project should be critically questioned.

The thesis aimed to:

1. Conduct a literature review on live forensics for Linux systems and ransomware. This covers the research of current ransomware methods, trends, delivery methods and crypto analysis.
2. Considering the outcome of the literature review during experiment definition.
3. Design and implement an adequate test environment for the experiment execution. The design of the infrastructure should consider common Linux servers operating systems as well.
4. Execute the experiments as defined, using at least 2 ransomware samples and different forensic tools.
5. Critically evaluate the experiment results. Compare the results to similar research, especially the impact compared to Windows. Critically evaluate the project.

Goal one – Literature review. The literature review revealed that most Windows ransomware programs use a symmetric key for encryption, supported by asymmetric key management. Specifically, RSA and AES. The project was based on the hypothesis that this could also apply to Linux ransomware. This hypothesis was refuted by the experiments. The results of this work are a potential reference for other researchers. All objectives proposed for this goal have been achieved and fulfilled.

Goal two – Experiment design. The experiments were carefully designed and aimed at testing the hypothesis. To ensure comparison with Windows ransomware and other research, the experiments were compared to the work of Davies (Davies, 2020) and McDonald et al. (McDonald et al., 2022). The fact that the experiments cumulatively covered the experiments of two scientific papers is a success. The experiment scope and design proved to be adequate and eligable draw a comprehensive picture regarding Linux ransowmare. to All objectives proposed for this goal have been achieved and fulfilled.

Objective Three – Environment design and implementation. The aim of this project was to find an appropriate balance between effort and a realistic test environment. A realistic and high-quality test environment was created. The environment was comprehensive, but potential interference from external influences was noted during the experiments. To avoid uncertainties, a second virtual environment with simplified configuration was created for retesting. This allowed critical and unexpected results to be checked again. The need for a retest environment is a positive result and is recommended for any other scientific work. It turns out that the combination of a realistic, comprehensive environment and a second, smaller environment is very effective and provides high reliability. All objectives proposed for this goal have been achieved and fulfilled.

Objective Four – Test execution. The test execution was underestimated and took significantly longer than expected. The reason for this was that the need for repeat testing. In addition, the work dealt with a completely new area that had to be carried out under many assumptions. Due to these uncertainties, a more realistic but more complex environment was chosen after consideration. This meant that if unexpected or unknown results were encountered, a retest was required to validate the results. For example, it seemed common for Windows ransomware to communicate with a command and control server. The fact that not a single Linux ransomware exhibited such behaviour raised doubts as to whether the test environment could have an influence. To reduce the risk of interference, tests were repeated in a very simple test environment, but with an existing internet connection, and the results were successfully confirmed. This careful cross-checking resulted in considerable additional effort. For new types of scientific work, it is strongly recommended to allow sufficient time for validation and cross-checking of the results.

Experiment 1 was challenging because many AES keys were present in RAM prior to test execution in the complex test environment. It was difficult to track the changes in RAM. A retest in a simplified environment subsequently made the analysis easier.

The tests were carried out with different types of permission levels and different operating systems. Overall, instead of 2 ransomware variants, 3 samples were tested. Instead of the planned minimum of 3 experiments, all 5 experiments were carried out. At the same time, the network behaviour was analysed on the client and the server, which both were running at the same time. Therefore, it can be said proudly with good knowledge and conscience that the results were not only met, but exceeded.

Objective Five – Evaluation and comparative analysis. This project provided valuable insights into the current maturity level of Linux ransomware and highlighted differences from Windows. The work of Davies (Davies, 2020) and McDonald et al. were used for comparison (McDonald et al., 2022). It turns out that the Linux ransomware appears to be comparatively unfinished, as all samples had limited impact. Bugs were also observed, such as some ransomware programs failing when not run as root. Assumptions and recommendations that applied to Windows do not apply to Linux. Comparisons were made with a total of 4 scientific papers, 2 of which were used primarily for comparison and 2 were compared on individual points. Therefore, it can be said proudly with good knowledge and conscience that the results were not only met, but exceeded.

6.3. Self Appraisal

This project is a comprehensive analysis of Linux ransomware that identified key challenges in forensic investigations and differences from Windows-based ransomware. The result that Linux ransomware uses different key management and encryption methods, is tailored and limited to a specific target is a success. At its current stage of development, Linux ransomware appears to have neither a command and control infrastructure nor an automated payment infrastructure (e.g. Bitcoin). The Linux ransomware samples did not exploit additional vulnerabilities that spread through the operating system or network. Compared to Windows-based ransomware, some basic assumptions and incident response recommendations were revised.

The work enables concrete recommendations for protective measures, forensic investigations and incident response.

The broad scope of the work enabled a representative assessment of Linux ransomware. However, if the project becomes repetitive, the environment should be adjusted. The chosen environment was safe, robust and universal, but was far from sufficient for every experiment. Overall, a comprehensive amount of data was collected, as shown in Table 6.1.

	Amount of files	**Storage needed**
Virtual machines	32	329 GB
Memory dumps	88	363 GB

Table 6.1: Summary of collected raw data

During testing, risks of possible side effects were identified. To rule out such risks, some retesting was required to confirm critical results in isolated environments. For the future, it is recommended to rely on multiple, but smaller environments that are quickly and precisely tailored to the respective test. This would provide more certainty about possible side effects right from the start. The overall effort required to manage the environments and carefully conduct the experiment represented a significant workload for the project.

Due to personal circumstances, individual milestones were temporarily delayed by 4 weeks compared to the project plan attached in A.2. This delay was offset by vacation and overtime. I am proud that I mastered the large scope and amount of work.

In professional life, security incidents often raise the question of how to deal with Linux-based systems. At the same time, there is a lack of representative research in this area. This work provided the right answers and contributed significantly to the development of knowledge in this field.

In summary, the following questions were researched. Ransomware execution, key management and ciphers used, key extraction, comparison with Windows operating systems, ransomware execution on different operating systems with different permissions, ability to spread throughout the network, communication with command and control servers as well as the impact of the encryption process.

Potential future work was documented during the project and is described in section 6.4.

6.4. Future work

The investigation revealed that ransomware for Linux is in its early stages. Most ransomware programs have focused on diversifying encryption methods and avoiding the use of AES encryption to hinder forensic investigations. The Linux ransomware samples did not utilise the full potential and were designed to achieve a specific, defined goal. Cybercriminals are expected to further increase the impact, build background infrastructures and develop more universal ransomware.

Icefire decryption. Decryption of Icefire encrypted files was not successful in this project. Future work could follow up on the decryption.

Development of forensic tools for new ciphers. There are several standard tools to reliably extract RSA and AES keys from memory dumps. However, the literature search only yielded design suggestions for determining the ChaCha20 keys. Further work is required to develop reliable and ready-to-use forensic tools for ChaCha20 and/or RC4.

Monitoring Linux ransomware development activity. This project showed that the Linux ransomware appears to be unfinished, has several flaws, and does not live up to its full potential. Cybercriminals are expected to close these gaps and work to increase impact. Further work should reassess the maturity level to see how quickly cybercriminals can develop ransomware. Cybercriminals are also believed to add command-and-control infrastructure to Linux variants. The re-evaluation should not begin until at least two years after publication of this work.

Real-time key monitoring. The experiments showed that ransomware also creates file-specific encryption keys and cleans RAM immediately after encryption of the single file. In this research, the memory dumps were created using the virtual machine mechanisms. Future research could investigate how to monitor key creation and extraction in real time to increase the likelihood of detecting attacks and decrypting files.

Real-time file access monitoring and expansion. Further research could provide useful insights into whether ransomware encryption processes could be captured by monitoring access to predefined HoneyPot files in real time and tricking such processes (e.g., by dynamically changing file size, dead-end redirects).

Persistence of Linux ransomware. This project did not address whether Linux ransomware is persistent after initial encryption and how it remains persistent on the system and network. There was also no evidence found that current ransomware variants seek long-term persistence. Future work could investigate how Linux ransomware could achieve persistence and mitigate it.

Emotional aspects during an attack. The project found that victims regularly report the emotional impact following a ransomware attack, particularly during negotiations. However, the literature search revealed that emotional aspects are not regularly examined in scientific papers. Future work could examine how the emotional experiences influence the response to the incident and the ransom negotiation.

Ransomware delivery methods and division of labor. It turned out that the classic assumptions about the spread of ransomware, namely via email and phishing, no longer applied. It would be extremely important to examine how delivery methods have changed.

7. References

Akbanov, M. et al. (2019) WannaCry Ransomware: Analysis of Infection, Persistence, Recovery Prevention and Propagation Mechanisms. *Journal of Telecommunications and Information Technology.* [Online] 1 (2019), 113–124.

Al-rimy, B. A. S., Maarof, M. A., & Shaid, S. Z. M. (2018). Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. Computers & Security, 74, 144–166. https://doi.org/10.1016/j.cose.2018.01.001

Andelkovic, A., Hausknecht, K., & Sirovatka, G. (2020). Linux Forensic Triage: Overview of Process and Tools. *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, 1230–1235. https://doi.org/10.23919/MIPRO48935.2020.9245304

Ashdown, D.(2021). *Jigsaw Ransomware Analyses* . Retrieved from https://www.cyberdonald.com/post/jigsaw-ransomware-analyses

Arfeen, A. et al. (2022) Process based volatile memory forensics for ransomware detection. Concurrency and computation. [Online] 34 (4), .

Bajpai, P. et al. (2018). A key-management-based taxonomy for ransomware. 1-12. 10.1109/ECRIME.2018.8376213.

Balaban, S. et al. (2021). *White paper on the legal situation IT security research* [translated from German]. Retrieved from https://www.fzi.de/wp-content/uploads/2021/11/doku-position-itsicherheitsforschung.pdf

Bart, W., et al. (2018). *Is Paying a Ransom to Stop a Ransomware Attack Illegal?*. Retrieved from https://www.sxsw.com/wp-content/uploads/2018/03/Legality-of-Paying-Ransom-FINAL-2018.1.19.pdf

Barker, W. C., Fisher, W., Scarfone, K., & Souppaya, M. (2022). Ransomware risk management : *National Institute of Standards and Technology (U.S.).* https://doi.org/10.6028/nist.ir.8374

Beaman, C. et al. (2021) Ransomware: Recent advances, analysis, challenges and future research directions. Computers & security. [Online] 111102490–102490.

Berardi, D., et al. (2023). Data Flooding against Ransomware: Concepts and Implementations. Computers & Security, 131, 103295–. https://doi.org/10.1016/j.cose.2023.103295

Boyd, A. D., et al. (2023). *The Increasing Risks and Prohibitions Associated With Paying a Ransom After a Ransomware Attack*. Retrieved from https://www.polsinelli.com/publications/the-increasing-risks-and-prohibitions-associated-with-paying-a-ransom-after-a-ransomware attack#:~:text=Existing%20risks%20for%20making%20or,and%20 may%20incentivize%20more%20attacks

BSC. (2022). BCS, *The Chartered Institute for IT. CODE OF CONDUCT FOR BCS MEMBERS*. Retrieved from https://www.bcs.org/media/2211/bcs-code-of-conduct.pdf

Bundesamt für Sicherheit in der Informationstechnik. (2022). *Ransomware. Threat situation 2022.* [translated from German]. Retrieved from https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Ransomware.pdf?__blob=publicationFile&v=5

Chainalysis. (2023). *Ransomware Revenue Down As More Victims Refuse to Pay*. Retrieved from https://www.chainalysis.com/blog/crypto- ransomware-revenue-down-as-victims-refuse-to-pay/

Chandra, V., & Hareendran, A. (2018). *Research methodology* (1st edition). Pearson India Education Services.

Davies, S. R. et al. (2020) Evaluation of live forensic techniques in ransomware attack mitigation. *Forensic Science International: Digital Investigation*. [Online] 33300979–

Delamotte, A. (2023). *IceFire Ransomware Returns | Now Targeting Linux Enterprise Networks*. Retrieved from https://www.sentinelone.com/labs/icefire-ransomware-returns-now-targeting-linux-enterprise-networks/

Donauer, J. (2015). *Linux.Encoder.1: Ransomware mit Verschlüsselung hat es auf Linux-Anwender abgesehen*. Retrieved from https://www.bitblokes.de/linux-encoder-1-ransomware-mit-verschluesselung-hat-es-auf-linux-anwender-abgesehen/

Edgar, T. W., & Manz, D. O. (2017). *Research methods for cyber security*. Syngress.

Europol. (2023). Cyber-attacks: the apex of crime-as-a-service, Europol Spotlight Report series, Publications Office of the European Union, Luxembourg.

Federal Office for Information Security. (2023). *TOP 10 Ransomware measures*. Retrieved from https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Cyber-Sicherheitslage/Analysen-und-Prognosen/Ransomware-Angriffe/Top-10-Ransomware-Massnahmen/top-10-ransomware-massnahmen_node.html

Fortune Business Insights (2023). *Server Operating System Market Volume, Share & COVID-19 Impact Analysis, By Operating System (Windows, Linux, UNIX, and Others), By Virtualization Status (Virtual Machine, Physical, and Virtualized), By Subscription Model (Non-paid Subscription and Paid Subscription), By Enterprise Type (Large Enterprises and Small & Medium Enterprises), and Regional Forecast, 2023-2030.* Retrieved from https://www.fortunebusinessinsights.com/serveroperating-system-market-106601

G Data Software. (2022). *G DATA threat report: Significant increase in Linux ransomware*. Retrieved from https://www.gdatasoftware.co.uk/news/2022/08/37568-g-data-threat-reportsignificant-increase-in-linux-ransomware.

Glenny, M. (2023). *The untold history of today's Russian-speaking hackers*. [Online] Retrieved from https://www.ft.com/content/9ac188be-8bcf-4b5a-8051-10563683b979

Guenni (pseudonym). (2023). *When ransomware groups offers security tips*. [Online] Retrieved from https://borncity.com/win/2023/11/13/when-ransomware-groups-offers-security-tips/

Johansen, G. (2022) Digital Forensics and Incident Response: Incident Response Tools and Techniques for Effective Cyber Threat Response. *3rd ed. Birmingham: Packt Publishing, Limited*

Kara, I. & Aydos, M. (2022) The rise of ransomware: Forensic analysis for windows based ransomware attacks. Expert systems with applications. [Online] 190116198–.

Kaspersky ICS CERT. (2021). *DarkChronicles: the consequences of the Colonial Pipeline attack*. Retrieved from https://ics-cert.kaspersky.com/publications/reports/2021/05/21/darkchronicles-the-consequences-of-the-colonial-pipeline-attack/#:~:text=So%20why%20did%20it%20take, backups%20to%20restore%20its%20systems.

Kong, J. H., Ang, L.-M., & Seng, K. P. (2015). A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments. *Journal of Network and Computer Applications*, *49*, 15–50. https://doi.org/10.1016/j.jnca.2014.09.006

Lamsouber, L. (2023).*What is a ransomware and how does it work?*. Retreived from https://www.theseus.fi/bitstream/handle/10024/810033/Lamsouber_Luca.pdf

Liska, A., & Gallo, T. (2017). Ransomware : defending against digital extortion (1st edition). O'Reilly.

Macfarlane, R. et al. (2023) Evaluation of live forensic techniques, towards Salsa20-Based cryptographic ransomware mitigation.

Malin, C. H. et al. (2014) Malware forensic field guide for Linux systems. Curtis W. Rose (ed.). Waltham, MA: Syngress.

McDonald, G. et al. (2022) Ransomware: Analysing the Impact on Windows Active Directory Domain Services. *Sensors (Basel, Switzerland)*. [Online] 22 (3), 953–.

McLaren, P., et al. (2019). Deriving ChaCha20 key streams from targeted memory analysis. *Journal of Information Security and Applications*, *48*, 102372-. https://doi.org/10.1016/j.jisa.2019.102372

NCSC.(2022).*Mitigating malware and ransomware attacks.* Retrieved from https://www.ncsc.gov.uk/guidance/mitigating-malware-and-ransomware-attacks

NIST.(2022).Ransomware.National Institute of Standards and Technology. Retrieved from: https://www.nist.gov/system/files/documents/2022/02/17/Ransomware.pdf

Or-Meir, O., Nissim, N., Elovici, Y., & Rokach, L. (2019). Dynamic Malware Analysis in the Modern Era-A State of the Art Survey. *ACM Computing Surveys*, *52*(5), 1–48. https://doi.org/10.1145/3329786

Parasram, S. V. N. (2020) Digital forensics with Kali Linux : perform data acquisition, data recovery, network forensics, and malware analysis with Kali Linux 2019.x. Second edition. Birmingham ;: Packt Publishing.

Red Hat .(2019). *Red Hat: Leading the enterprise Linux server market.* Retrieved from https://www.redhat.com/en/blog/red-hat-leading-enterprise-linux-server-market

Robles-Carrillo, M., & García-Teodoro, P. (2022). Ransomware: An Interdisciplinary Technical and Legal Approach. *Security and Communication Networks, 2022, 1–17*. https://doi.org/10.1155/2022/2806605

Roth, M. (2023). *INTERNAL PAPER OF THE BSI. Final report: Anhalt- Bitterfeld paralyzed for too long by cyber attack.* [Translated from German]. Retrieved from https://www.mdr.de/nachrichten/sachsen-anhalt/dessau/podcast-cyberkatastrophe-bsi-kritik-landkreis-100.html

Salvi, H. U. (2015). Ransomware : A Cyber Extortion. Asian Journal of Convergence in Technology, II(III 2350-1146).

Sanger, D. E. & Perlroth, N. (2021).*Pipeline Attack Yields Urgent Lessons About U.S. Cybersecurity*. Retrieved from https://www.nytimes.com/2021/05/14/us/politics/pipeline-hack.html

Schlag, J. (2021). *University of Regensburg. Unauthorized data access is also a punishable offense for administrators*. [Translated from German]. Retrieved from https://www.uni-regensburg.de/informationssicherheit/it-sicherheit/aktuelles/index.html?tx_news_pi1%5Baction%5D=detail&tx_news_pi1%5Bcontroller%5D=News&tx_news_pi1%5Bnews%5D=6875&cHash=cc3aa346a1fcdc8ae3e44206bf1fb687

Schreiber, M. (2023). *Hacking attack on Marc O'Polo - what you can learn from it*. Retrieved from https://www.itsa365.de/en/news-knowledge/2023/interview/hacking-angriff-auf-marc-opolo

Sentinel One. (2023). *Cl0p-ELF-Decryptor*. Retrieved from https://github.com/SentineLabs/Cl0p-ELF-Decryptor/blob/main/clop_linux_file_decr.py

Sittig, D. F. & Singh, H. (2019). *A Socio-Technical Approach to Preventing, Mitigating, and Recovering from Ransomware Attacks (Tech. Rep.).* PWC. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4941865

Skulkin, O. (2022) Incident response techniques for ransomware attacks : understand modern ransomware attacks and build an incident response strategy to work through them. Birmingham: Packt Publishing, Limited.

Skuratovich, S.(2016). *CHECK POINT. LOOKING INTO TESLACRYPT V3.0.1*. Retrieved from https://blog.checkpoint.com/wp-content/uploads/2016/05/Tesla-crypt-whitepaper_V3.pdf

Statista. (2023). Marketshare of leading operating systems worldwide from January 2009 till July 2023 (translated from German). Retrieved from https://de.statista.com/statistik/daten/studie/157902/umfrage/marktanteil-der-genutztenbetriebssysteme-weltweit-seit-2009/

Statista. (2023). *Dstribution of 500 most powerful supercomputers worldwide operating systems in June 2023*. [Translated from German]. Retrieved from https://de.statista.com/statistik/daten/studie/260534/umfrage/verteilung-von-supercomputern-nachbetriebssystem

Statista. (2019). *Share of the global server market by operating system in 2018 and 2019.* Retrieved from https://www.statista.com/statistics/915085/global- server-share-by-os/

Terefos, A. (2023). *Cl0p Ransomware Targets Linux Systems with Flawed Encryption | Decryptor Available*. Retrieved from https://www.sentinelone.com/labs/cl0p-ransomware-targets-linux-systems-with-flawed-encryption-decryptor-available/

The White House. (2023). *International Counter Ransomware Initiative 2023 Joint Statement*. Retrieved from https://www.whitehouse.gov/briefing-room/statements-releases/2023/11/01/international-counter-ransomware-initiative-2023-joint-statement/

Trend Micro. (2022). *Defending the Expanding Attack Surface. Trend Micro 2022 Midyear Cybersecurity Report*. Retrieved from https://documents.trendmicro.com/assets/rpt/rpt-defending-the-expanding-attack-surface-trend-micro-2022-midyear-cybersecurity-report.pdf

Umara Urooj et al. (2022) Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions. Applied sciences. [Online] 12 (172), 172–

Umawing, J. (2022). *BlackBasta is the latest ransomware to target ESXi virtual machines on Linux.* Retrieved from https://www.malwarebytes.com/blog/news/2022/06/blackbasta-is-the-latest-ransomware-to-target-esxi-virtual-machines-on-linux

Volatiltiy Foundation.(2020). *Linux*. Retrieved from https://github.com/volatilityfoundation/volatility/wiki/Linux

Wood, K. (2023). *Cybersecurity Policy Responses to the Colonial Pipeline Ransomware Attack*. Retrieved from https://www.law.georgetown.edu/environmental-law-review/blog/cybersecurity-policy-responses-to-the-colonial-pipeline-ransomware-attack/#_ftn12

Young, A., & Moti Yung. (1996). Cryptovirology: extortion-based security threats and countermeasures. *Proceedings 1996 IEEE Symposium on Security and Privacy, 129–140*. https://doi.org/10.1109/SECPRI.1996.502676

Appendices

Appendix A: Project management

This attachments support that the learning outcome 2 has been successfully achieved. Below is the evidence of project management, including plan and diary.

A.1 Project proposal

A first project proposal was made on 31st of May 2023 and discussed with potential supervisor R. Macfarlane on 08th of June 2023. After discussion this first project proposal was withdrawn. An updated proposal was discussed with new supervisor L. Maglaras and verbally agreed during a meeting held on the 6th of July 2023.

A.2 Project timeline

Duration	Start - **Deadline**	Milestone
3 weeks	24.07 - **14.08.**	Literature research finished
1 week	14.08 - **21.08.**	Test environment and experiments designed
	21.08.	***Initial report submitted***
3 weeks	28.08. - **18.09.**	Implementation & Execution finished
3 weeks	18.09. - **09.10.**	Results analysed
2 weeks	09.10. - **23.10.**	Evaluation and comparison with previous research accomplished
1 Week	23.10. - **30.10.**	Conclusions
2 weeks	06.11. - **21.11.**	Final report reviewed
	22.11.	***Final report submitted***
	01.12.	***Viva finished***

Table A.2 Project and milestone plan (according to initial report)

A.3 Project diary

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Korac, Salko **Supervisor: Maglaras, Leandros**

Date: 31.05.2023 **Last diary date: 22.11.2023**

Objectives:

- Identify research areas
- Prepare initial research proposal

Progress:

23.05.2023

- Collect intersting topics

24.05.2023

- Reread the moodle distertation module
- Watch welcome workshop on moodle
- Screen discussion forum

25.05.2023

- Evaluate own interesting topics and project topic ideas in https://moodle.napier.ac.uk/pluginfile.php/3269939/mod_resource/content/28/MSc%20Dissertation%20Project%20Ideas%202023.pdf
- Conduct first, basic literature review
- Triage the topics to three

27.05.2023

- Selection of research area of "Large-scale empirical analysis of different UK industries in regards to common ransomware root causes and basic cyber security principles."

28.05.2023-31.05.2023

- Preparation of research proposal
- Upload research proposal
- Inform Thomson, Craig about completed research proposal and propose Rich Macfarlane as supervisor

Supervisor's Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Korac, Salko **Supervisor: Maglaras, Leandros**

Date: 06.07.2023 **Last diary date: 22.11.2023**

Objectives:

- Critically review and refine research proposal

Progress:

06.06.2023

- Answer written questions from Rich Macfarlane
- Arrange first supervision meeting

08.06.2023

- First supervision meeting with Rich Macfarlane
- Review research proposal, identify challenges
- Outcome: First proposed scope seems too wide; evaluation of results not described

09.06.2023

- Critically review the research area
- Identify a research focus, “Evaluation of Linux Ransomware”

10.06.2023

- Basic literature research to ensure, that a comparison to other research is possible

11.06.2023

- Update research proposal

12.06.2023

- Update research proposal
- Hand-in updated research proposal after review

13.06.2023

- Assignment of final supervisor (Leandros Maglaras) and internal examiner (Huynh Nguyen)

25.06.2023

- In contact with supervisor

27.06.2023

- Arranged first supervision meeting

06.07.2023

- Meeting with supervisor Leandros Maglaras
- Arranged bi-weekly mentoring sessions

Supervisor’s Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Korac, Salko	**Supervisor: Maglaras, Leandros**
Date: 31.07.2023	**Last diary date: 22.11.2023**

Objectives:

- Literature review
- Obtain ransomware samples

Progress:

11.07.2023

- Linux ransomware research
- Assess the quality (e.g. peer-reviewed?)

12.07.2023

- Continue Linux ransomware research
- Download ransomware samples

15.07.2023

- Verify validity of ransomware samples using different online platforms
- Execute ransomware samples in an isolated environment to ensure prerequisites for the project

24.07.2023

- Meeting with supervisor

Supervisor's Comments:

-

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Korac, Salko	**Supervisor: Maglaras, Leandros**
Date: 30.08.2023	**Last diary date: 22.11.2023**

Objectives:

- Complete literature review
- Finalize initial report

Progress:

07.08.2023

- Supervision meeting
- Download and make myself familiar with initial report template and contents
- First attempt to fill out initial report template

12.08.2023

- Linux live forensics research
- Install forensic tools on machine for testing purposes

13.08.2023

- Execute forensic tools on machine
- Research in retrieving forensic keys from memory

14.08.-17.08.2023

- Read carefully the research paper from Davies (2020)
- Identify key problems of Davies (2020)
- Identify recommendations regarding the extraction of keys and environmental setup
- Follow-up references made by Davies (2020)

18.08. - 20.08.2023

- Continue research
- Write initial report and finalize
- Spell-check of initial report

21.08.2023

- Supervision meeting, obtain feedback on initial report draft
- Review initial report after feedback, send initial report for feedback to internal examiner
- Upload initial report to Turnitin

Supervisor's Comments:

-

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Korac, Salko **Supervisor: Maglaras, Leandros**

Date: 30.09.2023 **Last diary date: 22.11.2023**

Objectives:

- Complete Literature review
- Prepare Initial Report

Progress:

08.09.2023

- Supervision meeting

13.09.2023

- Ask internal examiner for feedback regarding initial report

15.09.2023

- Got feedback from internal examiner
- Review feedback, main outcomes noted
- Downloaded dissertation template and started writing
- Started experiment definition

17.09.2023

- Showing appreciation to internal examiner regarding feedback
- Started virtual environment design
- Definition of an experiment playbook

18.09.2023

- Supervision meeting
- Continued with environment design (e.g. virtual environment)

19.09.2023

- Definition of ethical safeguards
- Prepare virtual environment setup, download VirtualBox software

22/23.09.2023

- Reworked the sections and complete structure in the dissertation
- Continued with chapter 2.1.1 Definitions

24.09.2023

- Researched for commonly used Linux distributions as server operating system
- Selected Ubuntu and Debian for virtual environment
- Downloaded Ubuntu 20.04.6 AMD64 and Debian 12.1.0 AMD64 distribution

25.09.2023

- First installation of Ubuntu and Dabian as VMs

Progress:

26.09.2023

- Continue on dissertation, create draft version 0.2
- Write chapter 2.1.2 Relevance (of ransomware) for Linux operating systems

27.09.2023

- Continue on dissertation
- Write chapter 2.1.3 Current infection paths and impact (on Windows and Linux OS)
- Install forensic tools on machine for testing purposes

28.09.2023

- Continue on dissertation
- Write chapter 2.1.4 The emotional aspect during a ransomware attack
- Define honeypot files for virtual machines
- Start chapter 2.2 Linux live forensic analysis and 2.3 Cryptographic analysis

30.09.2023

- Continue configuration of virtual machine: ssh, ftp, Samba
- Deactivate virtual network interfaces
- Run ransomware for testing purposes in isolated network

Supervisor's Comments:

-

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Korac, Salko	**Supervisor: Maglaras, Leandros**
Date: 31.10.2023	**Last diary date: 22.11.2023**

Objectives:

- Finish test environment
- Execute tests
- Collect results

Progress:

01.10.2023

- Continue with dissertation, create visuals for attack paths (kill chain)
- Start chapter 3.2 Research Methodology

02.10.2023

- Test memory dumping and network capturing of virtual machines
- Isolate virtual machines from network side

03.10.2023

- Freed disk space on internal SSD, removed big files
- Made a full data backup of laptop
- Obtained external 500 GB SSD to store virtual machines and raw data
- Erased external SSD

04.10.2023

- Backup failed due to big files, restarted data backup
- Meeting with supervisor, first dissertation draft presented, writing style reviewed

08.10.2023

- Moved virtual machines from internal SSD to external SSD
- Confirm that VMs are operational

11.10.2023

- Start dissertation draft version 0.3, started chapter 3.10 and described how capture network and to deactivate virtual network interface

12.10.2023

- Research how to create virtual networks in VirtualBox
- Create a virtual network and move “Client-Debian” and “Server-Clean” into the network
- Research if DNS service is necessary in the network
- Ensure that no connection outside the virtual machines is possible

15.10.2023

- Start experiment 1: blackbasta on Server
- Start experiment 2: Icefire on Server

Progress:

16.10.2023

- Meeting with supervisor
- Starting dissertation version 0.4., improve wording in dissertation

17.10.2023

Experiment in memory dump extraction and timing

Develop script "memory_acquisition_script.sh"

- Experiment execution "Test 3 (retest)"
- Experiment execution "Test 5 Blackbasta as normal user"
- Experiment execution "Test 7 Cl0p on Debian as normal user"
- Experiment execution "Test 8 Cl0p on Debian as sudo"
- Update test results in internal "Experiments_documentation.xlsx"
- 18.10.2023
- Repeat tests to verify approach
- Analyse memory dumps and follow-up on existence of AES keys
- Update test results in internal "Experiments_documentation.xlsx"

22.10.2023

- Experiment execution "Test 6 Icefire on Debian as normal user"
- Experiment retests and cross-checks, validation key existence before encryption starts
- Migrate script decrypt.py to python3 and attempt to decrypt Icefire encrypted files
- Download and compile tool findaes to cross-check existence

23.10.2023

- Continue to decrypt Icefire encrypted files
- Analyse binary content with xdd, hexdump and strings

24.10.2023

- Continue to decrypt Icefire encrypted files
- Identify hard coded keys of 3 ransomware samples
- Research for newspaper articles regarding the ransomware samples

25.10.2023

- Continue to decrypt Icefire encrypted files
- Mark decryption as failed

26.10.2023

- Restart saved virtual machines and verify AES key existence after reboot
- Verify impact on virtual machines

27.10.2023

- Continue with impact analysis on virtual machines, attempt to re-login
- Start dissertation version 0.5, begin section 4. Results evaluation and comparative analysis

28.10.2023

- Result identified, that Cl0p ransomware does not encrypt files when executed as normal user
- Retests to confirm Clop results, re-login not possible once logged out after execution

Supervisor's Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Korac, Salko **Supervisor: Maglaras, Leandros**

Date: 22.11.2023 **Last diary date: 22.11.2023**

Objectives:

- Finish evaluation
- Finish dissertation

Progress:

01.11.2023

- Forensic analysis of AES keys difficult due to complex setup
- Create dedicated virtual machine without any GUI nor any other connections, Server-Clean-CMD with Ubuntu 20.04.6 server without desktop software.
- Repeated tests to confirm that there are no keys in memory before execution, confirmation successful
- Retest of experiment 4 for all 3 ransomware samples on a simplified "Server-Clean-CMD" and activate network trace.

02.11.2023

- Improve section 4, add screenshots
- Start version 0.6, improve tables for comparative analysis, new criteria documented
- Retest experiment 1 and 2 with simplified "Server-Clean-CMD" VM and ensure that there are no AES keys in RAM before test execution

03.11.2023

- Document results of experiment 4
- Adapt virtual environment for Blackbasta ransomware
- Retested blackbasta successfully after creation of required path /vmfs/volumes
- Attempt to decrypt Blackbasta files
- No keys identified

04.11.2023

- Researched on Blackbasta news
- Blackbasta uses an ChaCha20 algorithm
- Identified public key in Blackbasta ELF binary

05.11.2023

- Start version 0.7

06.11.2023

- Continue on version 0.7, improve tables for comparative analysis, new criteria documented
- Start section 5, and chapter 5.1 Protective Measures, 5.3 Self Appraisal
- Supervision meeting, informed about results, provided and presented dissertation draft
- Aligned last supervision meeting to be held on 17.11.2023

Progress:

07.11.2023

- Found decryptor for Cl0p on Github
- Identified hard-coded RC4 key in Cl0p ELF binary
- Retest with fresh, dedicated machine. Mirrored decryptor project on Github on VM, files successfully decrypted

08.11.2023

- Analyse memory dumps and network traces, take screenshots for dissertation
- Uploaded ransomware samples to VirusTotal and took screenshots
- Freeze raw data, archived virtual machines

09.11.2023

- Create dissertation version 0.8, improved aims and objectives

10.11.2023

- Writing evaluation part
- Improve wording, style, fonts and spacing, resize images and tables
- Try to continue with Libreoffice

11.11.2023

- Continue on dissertation
- Writing evaluation part
- Repair corrupt dissertation file from 1h backup before, leave Libreoffice and go back to TextMaker
- Resize images and tables,
- Refresh index for Tables and Figures

12.11.2023

- Continue on dissertation, version 0.92 and 1.0, add line breaks, MSc checklist, add attachments, Roman numbers, improve footer and headers

13.11.2023

- Continue on dissertation, version 1.1, spell-check, wording, cross-check references
- Writing evaluation part

14.11.2023

- Continue on dissertation, version 1.2, spell-check, wording, cross-check references
- Writing evaluation part

15.11.2023

- Continue on dissertation, version 1.3 and 1.4, spell-check, wording, cross-check references

16.11.2023

- Continue on dissertation, version 1.5, spell-check, wording, cross-check references

17.11.2023

- Continue on dissertation, version 1.5, spell-check, wording, cross-check references
- Last supervision meeting

18.11.2023

- Continue on dissertation, version 1.6, spell-check, wording, cross-check references
- Cross-check with feedback list of internal examiner (initial report)

19.11.2023

- Upload to first time to Turnitin
- Continue on dissertation, version 1.7, spell-check, wording, cross-check references

20.11.2023

- Continue on dissertation, version 1.8, spell-check, wording, cross-check references

Progress:

21.11.2023

- Continue on dissertation, version 1.9, spell-check, wording, cross-check references

22.11.2023

- Final upload of version 2.0 to Turnitin and submission via email to internal examiner and supervisor
- Final diary entry

Supervisor's Comments:

-

Appendix B: Code samples

B.1 decrypt.py: Script migrated to Python 3.0

This script was obtained from the previous research done by Davies Simon in 2019. It was used to try to decrypt the Icefire encrypted files. It was necessary to migrate this script to Python 3.0 and to adapt it slightly.

```
"""
Script to decrypt a file encrypted with AES
The script uses a file that can contain multiple AES keys and it will try each in turn
Simon Davies simonrdavies@yahoo.com
Version 1.0
20190822
"""
import os, random, struct, argparse, ntpath
import binascii
from Crypto.Cipher import AES


def decrypt_file(key, in_filename, out_filename=None, chunksize=24*1024):
        """ Decrypts a file using AES (CBC mode) with the
         given key. Parameters are similar to encrypt_file,
         with one difference: out_filename, if not supplied
         will be in_filename without its last extension
         (i.e. if in_filename is ’aaa.zip.enc’ then
         out_filename will be ’aaa.zip’)
         """
        print ("key:", key)
        print ("in_filename: ", in_filename)
        if not out_filename:
        out_filename = os.path.splitext(in_filename)[0]
        with open(in_filename, "r") as infile:
        iv = infile.read(16)
        decryptor = AES.new(key, AES.MODE_CBC, iv)
        with open(out_filename, "wb") as outfile:
                #outfile.write(iv)
                #print "here"
                #exit()
                while True:
                        chunk = infile.read(chunksize)
                        if len(chunk) == 0:
                                break
                        if len(chunk) != 16:
                                break
                        outfile.write(decryptor.decrypt(chunk))
#generate a file name for the decrypted file based on the encrypted filename and the key
def destination_filename(source_filename,key):
        head,tail = ntpath.split(source_filename)
                if len(head) == 0:
                head = "."
                f, e = os.path.splitext(tail)
        newfilename = head+"/"+f+"-decrypted-"+key+e
        print ("newfilename: " , newfilename)
        return newfilename
def process_keyfile(key_file, encrypted_filename):
        if not os.path.exists(key_file):
                print ("Key file doesnot exist: ",key_file)
                return
        with open(key_file, "rb") as infile:
                keyline = str.split(infile.readline())
                while keyline:
                        if keyline.find("Found",0, 50)>=0:
                                print ("found a comment row: ", keyline)
                        elif keyline.find("#",0, 50)>=0:
                                print("found a comment row: ", keyline)
                        else:
                                key_nospace=keyline.replace(" ","")
                                #get rid of the carrage return
                                fkey=binascii.unhexlify(key_nospace)
                                decrypt_file(fkey,encrypted_filename,destination_filename(encrypted
                                _filename,key_nospace),16)
                                keyline = infile.readline()
        infile.close()


def main():
        parser_description = "Decrypt a file encoded with AES encryption"
        parser = argparse.ArgumentParser(description=parser_description)
        parser.add_argument("--file", help="Path to the encrypted file", required=True)
        parser.add_argument("--key", help="Path to the file which holds the AES key(s)",
        required=True)
        args = parser.parse_args()
        process_keyfile(args.key, args.file)


if __name__ == "__main__":
main()
```

B.2 memory_acquisition_script.sh: Script to obtain memory dumps

This script was development to ensure a reliable experiment execution. This script takes a first snapshot within the first 5 seconds. Other memory dumps are taken every 30 seconds. The final memory dump is taken 17 minutes and 5 seconds after execution. Overall this scripts creates 6 memory dumps. The memory dump before execution and after reboot was taken manually.

```
#!/bin/bash
clear
ransomwaresample="Test4_retest_Server-Clean_ubuntu_Cl0p_sudo_user"
echo "Memory acquisition for sample '$ransomwaresample'"

echo ""
echo "1) ---------"
echo "sleeping 5 seconds"
time sleep 5
echo "5 seconds over. dumping first memory for sample '$ransomwaresample'"
time VBoxManage debugvm "Client-Debian" dumpvmcore --filename $ransomwaresample.1st.run.5seconds.elf
echo "done"

echo ""
echo "2) ---------"
echo "sleeping 30 seconds"
time sleep 30
echo "30 seconds over. dumping second memory for sample '$ransomwaresample'"
time VBoxManage debugvm "Client-Debian" dumpvmcore --filename $ransomwaresample.2nd.run.35seconds.elf
echo "done"

echo ""
echo "3) ---------"
echo "sleeping 30 seconds"
time sleep 30
echo "30 seconds over. dumping third memory for sample '$ransomwaresample'"
time VBoxManage debugvm "Client-Debian" dumpvmcore --filename $ransomwaresample.3rd.run.65seconds.elf
echo "done"

echo ""
echo "4) ---------"
echo "sleeping 30 seconds"
time sleep 30
echo "30 seconds over. dumping fourth memory for sample '$ransomwaresample'"
time VBoxManage debugvm "Client-Debian" dumpvmcore --filename $ransomwaresample.4th.run.95seconds.elf
echo "done"

echo ""
echo "5) ---------"
echo "sleeping 30 seconds"
time sleep 30
echo "30 seconds over. dumping fith memory for sample '$ransomwaresample'"
time VBoxManage debugvm "Client-Debian" dumpvmcore --filename
$ransomwaresample.5th.run.125seconds.elf
echo "done"

echo ""
echo "5) ---------"
echo "sleeping 15 minutes"
time sleep 900
echo "15 minutes over. dumping last (15minutes) memory for sample '$ransomwaresample'"
time VBoxManage debugvm "Client-Debian" dumpvmcore --filename $ransomwaresample.6th.run.15min.elf
echo "done"
```

B.3 clop_linux_file_decr.py: Clop decryptor from github

This script was obtained from Github (https://github.com/SentineLabs/Cl0p-ELF-Decryptor/blob/main/clop_linux_file_decr.py). It was successfully used to decrypt Cl0P encrypted files. No adaptions were necessary.

```
"""
Author: @Tera0017/@SentinelOne
Description: Clop-Linux ransomware variant files decryption.
Link: https://s1.ai/Clop-ELF
Execution help: $ python3 clop_linux_file_decr.py --help
"""
import argparse
import glob
import os.path
import struct
from arc4 import ARC4
```

```
def parse_arguments() -> argparse.Namespace:
    """
    Parses commandline parameters if any.
    @return: returns parse_args() result -> argparse.Namespace
    """
    description = """Python3 script which decrypts files encrypted by flawed Cl0p ELF variant.
    More info regarding Cl0p ELF variant and how decryptor was created at https://s1.ai/Clop-ELF
    """
    print('=' * 40)
    print('SentinelOne Cl0p ELF variant Decryptor.\nAuthor: @Tera0017/@SentinelOne\nLink:
https://s1.ai/Clop-ELF')
    print('=' * 40)
    parser = argparse.ArgumentParser(
                        prog='clop_linux_file_decr.py',
                        description=description,
                        epilog='author:@Tera0017/@SentinelOne')

    parser.add_argument('--elfile', default=None, help='ELF Cl0p Binary, is used to retrieve "RC4
master key" else default is used for decryption.')
    parser.add_argument('--keys', default=None, help='File containing result of "$ find / -name
*.$cl0p_extension -print 2>/dev/null > cl0p_keys.txt". Run with sudo if needed.')
    parser.add_argument('--rc4key', default=None, help='RC4 master key for decryption of clop key
files. If --elf is provided script will dynamically retrieve it.')
    return parser.parse_args()


def message(msg: str) -> None:
    """
    @param msg: message to print
    @return: None
    """
    print(f'* {msg}')


class ClopELFDecryptor:
    def __init__(self, filepath=None, clop_find_file=None, rc4_master_key=None):
        """
        @param filepath: str, filepath of cl0p elf variant ransomware found in encrypted machine.
        @param clop_find_file: str, filepath containing result of "$ find / -name *.$cl0p_extension
-print 2>/dev/null > clop_keys.txt"
        @param rc4_master_key: str, rc4 master key is not extracted well from cl0p elf binary.
        """
        # if elf sample does not exist tries with observed key.
        self.elfdata = open(filepath, 'rb').read() if filepath is not None else None
        # result of "$ find / -name *.$cl0p_extension -print 2>/dev/null > clop_keys.txt" containing
clop keys
        self.clop_keys_file = clop_find_file
        self.rc4_master_key = rc4_master_key
        # clop filekeys extension.
        self.clop_ext = ".C_I_0P"
        # RC4 generated key size.
        self.rc4_gen_key_size = 0x75

    def get_rc4_master_key(self) -> bytes:
        """
        Retrieves RC4 master key from ELF binary. If elf is not found returns default observed key.
        @return: bytes, RC4 master key.
        """
        if self.rc4_master_key is not None:
            message('User provided RC4 master key')
            return self.rc4_master_key
        elif self.elfdata is None:
            message('Retrieved previous observed RC4 key.')
            # observed RC4 master key
            return b'Jfkdskfku2ir32y7432uroduw8y7318i9018urewfdsZ2Oaifwuieh~~cudsffdsd'
        # dirty way to retrieve master key.
        f = b'/root'
        idx = self.elfdata.find(f) + len(f) + 1
        return self.elfdata[idx: idx + 100].lstrip(b'\x00').split(b'\x00')[0]

    def get_clop_keys(self) -> list:
        """
        Based on the filekeys clop extension retrieves all the encrypted files from the machine.
        * If you need to speed up process add specific folders where encryption took place.
        * Or pass result of "$ find / -name *.$cl0p_extension -print 2>/dev/null > clop_keys.txt"
        as argument to "--keys".
        @return: list, encrypted filepaths
        """
        if self.clop_keys_file is not None:
            # get clop keys "$ find / -name *.$cl0p_extension -print 2>/dev/null > clop_keys.txt"
            with open(self.clop_keys_file, 'r') as hfile:
                lines = hfile.readlines()
            return [l.strip() for l in lines if l.strip()]

        # enumerate all folders and find clop extension files.
        message(f'Searching for encrypted file extension {self.clop_ext}.')
        message('This operation will take several minutes...')
        message('To speed up process prefer to use "--keys", parameter.')
        return glob.glob(f'/**/*{self.clop_ext}', recursive=True)

    def decrypt(self) -> None:
        """
        Main function decrypts Clop-ELF encrypted files.
        @return: None
        """
        message('Starting decryption process.')
        #   1. Retrieve RC4 "master-key".
        rc4_master_key = self.get_rc4_master_key()
        message(f'RC4 Master Key: "{rc4_master_key}"')
        #   2. Read all $filename.$clop_extension.
        file_keys = self.get_clop_keys()
        message(f'Encrypted Files: {len(file_keys)}')
        for file_key in file_keys:
            message(f'File: {file_key}')
            with open(file_key, 'rb') as hfile:
                file_key_data = hfile.read()
            #   3. Decrypt with RC4 using the RC4 "master-key", the generated RC4 key.
            cipher = ARC4(rc4_master_key)
            file_rc4_key = cipher.decrypt(file_key_data)[:self.rc4_gen_key_size]
            # getting encrypted file size (if file is written again after encryption then
            # encrypted_file_size != file_size
            size_off = 0x75 + 0x58 + 0x8 + 0x4 + 0x4
            try:
                encr_file_size = struct.unpack('Q', file_key_data[size_off: size_off + 0x8])[0]
            except struct.error:
                message(f'[ERROR] Clop key file seems corrupted: {file_key}')
                continue
            encr_file = file_key.replace(self.clop_ext, '')
            # decrypted files have extension '.decrypted_by_S1', once validated can delete and
replace encrypted.
            decr_file = file_key.replace(self.clop_ext, '.decrypted_by_S1')
            if os.path.isfile(encr_file):
                with open(encr_file, 'rb') as hfile:
                    encr_file_data = hfile.read()
            else:
                message(f'[ERROR] Unable to find encrypted file: {encr_file}')
                continue
            #   4. Decrypt $filename with RC4 using the generated RC4 key.
            cipher = ARC4(file_rc4_key)
            decrypted_file_data = cipher.decrypt(encr_file_data[:encr_file_size]) +
encr_file_data[encr_file_size:]
            #   5. Write decrypted to $filename.
            with open(decr_file, 'wb') as hfile:
                hfile.write(decrypted_file_data)
            message(f'Decrypted: {decr_file}')


if __name__ == '__main__':
    # parsing command line arguments for the decryptor. Use --help for more information
    parsed = parse_arguments()
    ClopELFDecryptor(parsed.elfile, parsed.keys, parsed.rc4key).decrypt()
```