Comparative Forensic Analysis of the Ragnar Locker Ransomware

Submitted in partial fulfilment
of the requirements of the degree of

Bachelor in Digital Forensics

of Noroff University College

Marthe Brendefur

Kristiansand, Norway
May 2021

Declaration

I declare that the work presented for assessment in this submission is my own, that it has not previously been presented for another assessment, and that work completed by others has been appropriately acknowledged.

Name: Marthe Brendefur **Date**: May 16, 2021

Abstract

Ransomware is one of the most prominent and costly threats enterprises face. Attacks are estimated to cost organisations more than 20 billion dollars globally in 2021. The rapid development of the ransomware business, and the tools and techniques applied in targeted attacks, requires the defensive side to evolve continuously. When an attack occurs, digital forensics techniques are applied to locate, gather, and preserve information relevant to an investigation. The Hybridised Malware Framework was developed by Schmitt (2019) to facilitate a holistic approach to analyse ransomware forensically. This project details the hybrid analysis of two Ragnar Locker strains. Ragnar Locker is a ransomware family first spotted in the wild in December 2019 and that has since targeted several large companies. The attackers use a double extortion tactic, meaning that they extract the victim's data to add extra pressure on the company. The project is a comparative study, and the findings from the two strains are correlated and mapped. By distinguishing changes and the elements that remain constant between the two strains, researchers are better equipped to understand the development of the family and defend against it. Findings include functionality differences, increased operational security, and that neither strain communicates over the internet.

Keywords: *Ragnar Locker, Ransomware, Hybrid Analysis, Ransomware Forensics, Malware Analysis, Static Analysis, Dynamic Analysis, MITRE ATT&CK*

Acknowledgements

I am eternally grateful for all the amazing people I have in my life. A special thank you to the following people:

- **My closest family** – Ole, Kirsti, Maria, Lars, and Magnhild. Thank you for your love and support both during my studies and all my other adventures. It is a special thanks to mom; you have been supportive even though you no longer understand what I am doing and why it has taken up so much of the time we should have had together. It has been an excruciatingly painful decision to make. Time is precious, and I hope we can still have some moments together.
- **Aleksander Jankov** – Thank you for taking the time to supervise and discuss potential directions. A cup of coffee can change a lot, and for me it changed the direction of my career. I would not have dared to start on this intimidating topic if it had not been for you.
- **Veronica Schmitt** – Thank you for your guidance, understanding and support through the project. It has been a pleasure to have you as my supervisor.
- **My collegues** – Thank you for giving me space and opportunity to explore the universe of ransomware and cybercrime. You make work fun.
- **Fellow students Henrik, Hans, and Runar** – Thank you for your friendship through the studies at Noroff. Neither the learning outcome nor the experience would be the same without you. Thank you for making me laugh when I was down and helping me see solutions when I was stuck.

Contents

List of Figures

List of Tables

List of Code Listings

1

Introduction

Ransomware attacks are one of the most prominent and costly threats that an enterprise can face. It has been the most significant cyber threat since 2016, much due to the high attack frequency and the damage it inflicts upon victims. According to Purplesec (2021), approximately 187.9 million ransomware attacks occurred in 2019. Moreover, targeted ransomware campaigns are predicted to cost organisations more than $20 billion by 2021. It is not only the ransom demand itself that lead to high costs; downtime, broken equipment, time spent on incident management, and a potential loss of reputation are all factors that lead to significant economic losses. According to the security research firm Sophos (2020), in early 2020, the average cost of remediating a ransomware attack was $761,106 when the ransom was not paid. In cases where the ransom was paid, the amount doubled to almost $1.5 million. If personal information is leaked in a breach, organisations could potentially face lawsuits or GDPR fines as well.

Mitigating the threat posed by targeted ransomware is a complex task. There is no single solution that will safeguard systems, so organisations must implement a layered defence. The problem is that many attacks are carried out by human operators. Such skilled hackers will often manage to bypass implemented security mechanisms. In this kind of attack, the ransomware itself is only a tool to

render the data temporarily useless. If ransomware is deployed on a system, it is imperative to detect and terminate it at an early stage. By interrupting the encryption process, it is possible to reduce the cost of resurrecting the system and losing data. Detecting the ransomware is the last tier of defence.

The Ragnar Locker ransomware family is used in Big Game Hunting operations primarily targeting the energy, shipping -and travel sectors. The ransomware is developed and used by a financially motivated cybercriminal group. Although they are not the most active group in terms of attack frequency, the threat actors behind Ragnar Locker are known to be innovative in how they perform attacks. For instance, the group pioneered a technique where ransomware is deployed inside a virtual machine on the victim system to bypass security mechanisms. The virtual machine further enabled the ransomware to connect to shared resources and encrypt data on these. Other prominent cybercriminal groups later adopted the technique. Ragnar Locker also uses a double extortion tactic, where they exfiltrate sensitive data before encrypting the system. The exfiltrated data is later used as extra pressure to make the victim organisation pay the ransom demand (Tavares, 2020). If the demand is not met, the threat actors publish proof of their intrusion on a designated data leak site (DLS). Ragnar Locker's modus operandi makes it suitable for analysis to uncover trending techniques applied by ransomware operators.

This paper details the analysis of two selected strains from the Ragnar Locker family. The strains are analysed using the multi-disciplinary Hybridised Malware Framework developed for malware forensics by Schmitt (2019). The framework concatenates existing methods for reverse-engineering malicious software with forensic examination techniques. The aim is to uncover traits that can be used to profile the Ragnar Locker ransomware family in a Windows 10 environment. The uncovered tactics, techniques and procedures are parsed into the format of the MITRE ATT&CK framework. Using such a widely adopted cyber adversary model facilitates easy adaption and comparison of the findings.

The benefit of performing an in-depth ransomware analysis is that it may provide technical and behavioural information about the examined strain and its operators. On the technical side, extracted Indicators of Compromise (IOC) and forensic artefacts can help security professionals detect and mitigate the threat posed by the ransomware. Such findings can, for instance, be used to write YARA rules. Custom ransomware often bypasses signature-based detection mechanisms, necessitating agile security solutions and defence-in-depth. By eliciting

behavioural traits, one can distinguish actions performed by the ransomware and generate an overview of the adversary's tactics, techniques, and procedures (TTPs). TTP information can be distilled into actionable information and incorporated into the defence.

Ultimately, analysing Ragnar Locker strains at a granular level will contribute to the knowledge of how Ragnar Locker is built and the mechanisms the developers implement to safeguard their creation. This knowledge can later be used as an element in the organisational defence against the Ragnar Locker ransomware.

1.1 Problem Statement

The problem is that few in-depth academic analyses of how a ransomware family develops over time exist. The main objective of this research is to analyse two strains of the Ragnar Locker ransomware using the Hybridised Malware Framework (HMF) combined with the MITRE ATT&CK framework. The thesis is that the discovered characteristics are likely to apply to other members of the Ragnar Locker family. Despite the strains being tailor-made for their victims, most of the code is likely to be reused. By eliciting the key traits of two strains and comparing them, one can single out the attributes that apply to both. Concise characteristics allow for building good rules that can identify several strains of the same ransomware family. One example of such rules is YARA rules, which identifies malware based on textual or binary patterns (YARA, 2020). Tailored ransomware is unlikely to be detected by traditional signature-based indicators, such as antivirus (AV) engines, as each strain is unique to the victim. By implementing solutions that match code segments to specified rules, organisations increase their chance of detecting targeted ransomware strains (NCCIC, 2015).

The research will be conducted following the steps outlined in the HMF and will add to the testing and validating of the framework. By annexing a forensic analysis of a system in addition to the inspection of the ransomware itself, one gain a broader understanding of how the strain behaves on a system and the artefacts that exist due to its execution. However, the forensic aspect does not add much value when analysing a selected strain in a closed environment.Nonetheless, in an actual situation, a forensic examination is of the utmost importance to determine how the system was breached and the attacker's movements and actions before launching the ransomware.

1.2 Research Objectives

This papers primary goal and objective are to perform an extensive analysis of two Ragnar Locker ransomware strains using the HMF and MITRE ATT&CK framework. The following objectives are directly related to, or by-products, of the analysis. These objectives may contribute to the common knowledge of how the threat actors have designed their tools and aid in the detection of the analysed strains:

- Analyse two strains of the Ragnar Locker ransomware using the HMF.
- Compare the findings from the analysis of the two strains.
- Map the discovered TTPs in the MITRE ATT&CK framework.
- Provide a YARA rule that will detect both strains.
- Create a list of Indicators of Compromise for the strains.
- Contribute the discovered IOCs and YARA rule to the MISP Galaxy.

1.3 Scope and Limits

The main scope of this study is to uncover how two strains from the Ragnar Locker ransomware family behaves on a system and the characteristics that apply to both strains. The comparison of findings will provide insight into how the ransomware family has matured over time. The uncovered IOCs will be concatenated into one master list. Knowing the common traits will help build a YARA rule that detects both strains. IOCs discovered through the analysis of Ragnar Locker will also be contributed to the MISP Galaxy instance through GitHub. MISP instances functions as databases for information related to malware artefacts on the tactical, operational, and strategic levels. Contributing findings to a Threat Intelligence Platform (TIP) such as MISP is evaluated to be the best way to share information from the analysis and contribute to the overall knowledge base of the security community. The data gathering period for this research spans from October 12, 2020, to April 1, 2021. The analysis of the ransomware will be conducted between December 2020 and May 2021.

The following Ragnar Locker samples will be analysed:

Strain 1
`SHA-256:9bdd7f965d1c67396afb0a84c78b4d12118ff377db7efdca4a1340933120f376`

Strain 2
`SHA-256:dd79b2abc21e766fe3076038482ded43e5069a1af9e0ad29e06dce387bfae900`

Strain 1 was compiled in January 2020, and strain 2 in November 2020. By analysing strains with such a large time gap between them, one gets a better insight into the constant factors. Moreover, the analysis may uncover how the ransomware family have matured during the eleven months.

The analysis will primarily be conducted in a sandboxed Windows 10 environment. A sandbox is an isolated system with no real connection to a live network or host. The environment will be emulated using a VMWare virtual machine. To account for Ragnar Lockers reaction to a closed environment, internet services will be faked using INetSim and fakedns. Network services and traffic monitoring will be administered using a separate Linux virtual machine running on the same closed network as the Windows host. The Linux based REMnux solution is chosen for this purpose.

In addition to the local analysis of the ransomware, the strains will be run in the complimentary version of three commercial, online sandboxes.

The following sandboxes will be used:

- VirusTotal
- AnyRun
- JoeSandbox Cloud

The sandboxes are designed for malware analysis and implement various tools to analyse the behaviour of different species. VirusTotal is an agent-less sandbox. AnyRun and JoeSandbox Cloud are cloud-based sandbox solutions. Some ransomware samples will not execute properly if they detect that they are being analysed in a closed environment. Therefore, one must have a variety of solutions available to overcome the ransomware's evasion techniques. Agentless and cloud-based solutions are the solutions least likely to be detected by the ransomware (Ali & Papadaki, 2018).

The ransomware strains will be analysed using the Hybridised Malware Framework combined with the MITRE ATT&CK framework. The HMF consists of the following steps and sub-steps:

1. Static analysis
    1. File fingerprinting
    2. Examine hard-coded strings
    3. File format examination
    4. Packer detection
    5. Disassembly of the binary

2. Dynamic analysis
    1. Run in a sandboxed environment
    2. YARA rules
    3. ProcMon
    4. RegShot

3. Network

4. Digital Forensics analysis
    1. Preparation for Acquisition
    2. Memory analysis
    3. Registry analysis
    4. Artefact analysis

When the behaviour of the strain is mapped, it will be linked to the tactics and techniques presented in the MITRE ATT&CK framework. When the strains actions have been mapped to this standard, the findings can be concatenated into a graphical matrix. A matrix is a helpful tool to gain an overview of how a strain operates on a system. This matrix can later be used to compare the modus operandi of the two strains to other malicious code in a standardised manner.

It is, however, essential to note that the HMF includes forensic analysis of the victim system. The aim of such an analysis would, in the case of a real breach, be to trace the attacker's steps and actions on the victim system, as well as the artefacts created by the ransomware. As the analyses in this thesis concern ransomware strains executed on purpose in a sandboxed environment, the forensic findings will be limited to artefacts created by the strains.

Mitigation techniques and suggestions to improve system security on a general basis, beyond the YARA rule and the information shared on the MISP instance, fall outside the scope of this paper. Likewise, attribution of any sort will not be discussed as it is considered irrelevant to the analysis.

1.4 Document Structure

Chapter 1 provides an overview of the project. The chapter details the background for the research, research objectives, scope, and limitations.

Chapter 2 introduce the reader to technological concepts that are prominent within the field of ransomware research and the state of the ransomware business. This serves as a primer and foundation for topics discussed during the analysis section. The chapter also provide insight into relevant research and related work, with a particular focus on analysis areas. Most notably, it outlines the analysis process of the HMF.

Chapter 3 details how the analysis of the Ragnar Locker strains is performed at a general level. The collection of the ransomware samples is described, together with a list of all tools used during the analysis. A section discussing the risks and ethical considerations concludes the chapter.

Chapter 4 presents the details of the analyses and the findings in full. The chapter is structured to follow the outlined steps in the HMF. A comparison section at the end of the chapter summarises the key findings from the analysis and compares the analysis of the two strains.

Chapter 5 concludes this paper and summarises the overall findings, connecting the information from 2 and 4. It also evaluates to which degree the research objectives have been met and discuss to what extent the conducted research contributes to the field of academic ransomware analysis. Finally, the chapter suggests areas for further exploration.

2

Literature Review

This chapter provides a fundamental overview of key topics within the ransomware research domain and explains the technical concepts related to ransomware analysis. As the number of contemporary academic research papers focusing on ransomware analysis is limited, the academic literature has been supplemented with books, white papers, technology documentation, and ransomware reports where necessary.

Comprehensive ransomware attack analyses are seldomly offered freely online as breaches, and threat actor capabilities, are considered sensitive. However, some cursory analyses of dominating threat actors and their TTPs exist. The Ragnar Locker group were behind some breaches of considerable impact in 2020, which qualified them to be classified as a top tier cybercriminal group in the period this research was conducted.

2.1 General Academic Focus

Most academic researchers have focused on creating methods to detect and deter threats before the system is encrypted (Thomas et al., 2019). Exploring possible mitigation mechanisms is a natural focus when aiming to minimise the impact of breaches. Reversing complex cryptography algorithms is often impossible, as the encryption algorithms used by most ransomware families are extremely strong (Schmitt, 2019).

There seem to exist a variety of frameworks addressing the pre-encryption phase, but few forensic oriented frameworks for a holistic approach to post-exploitation ransomware analysis. Some of the focus areas are recurring across detection frameworks and analysis frameworks, like focus on I/O requests, entropy, and Indicators of Compromise - particularly Windows APIs and SHA-256 hashes (Subedi et al. (2018); Latzo et al. (2019)). One observation is that many of the existing pre-encryption frameworks focus on selected IOCs and will only function as a preventive measure if implemented alongside other mechanisms, contributing to establishing defence in depth. However, they are troublesome to implement on a large scale, require various resources, and are time-consuming to keep up to date (Wang & Wang, 2015). Considering that additionally, discovering Advanced Persistent Threats (APTs) requires constant threat intelligence and threat hunting, defence against such adversaries are reserved enterprises that can afford it (Husari et al., 2019). Threat hunting is when one actively monitors the system and looks for activities that are out of place. It builds on the *assume breach* mindset, where one suspects the adversary is already inside the system (Carbon Black, n.d.). Security mechanisms fall outside the scope for this thesis; however, it is vital to understand which functions are generally used to detect ransomware. Only by knowing this can one extract artefacts applicable to assist in the discovery of malicious actions on a system.

2.2 The Ransomware Fundamentals and Business

Ransomware is commonly known as a malicious program that holds data hostage. The name derives from the attackers' practice of demanding a ransom to be paid in order to release access to the data. Several branches of ransomware exist, and their taxonomy can be determined by examining different traits. Al-rimy et al. (2018) developed a universal taxonomy framework based on

market trends and literature, which divides ransomware into three categories; scareware, lockerware, and cryptoware. Scareware acts on the end user's fear to gain access to systems, for instance, by posing as antivirus software. Lockerware denies the user access to the system by displaying the ransom note and nothing else. Cryptoware hinders access to the files on a system by encrypting them.

The very first ransomware was a Trojan, which was spread through floppy disks. It encrypted the content of the `C:\` drive on the infected system. The strain was launched in 1989 and went by the names AIDS and PC Cyborg. The creator, Joseph Popp, had a PO Box set up to receive the ransom demand of $189 (O'Kane et al., 2018). Due to the challenges of paying the ransom, AIDS was not profitable (Waddell, 2016). Since 1989, ransomware has developed from simple malicious software spreading erratically to advanced malicious software used in targeted attacks (Zimba & Chishimba, 2019b). Today, ransomware is the leading cyberthreat and a billion-dollar business, according to Europol (2020). The estimates on ransomware's annual revenue and profit vary, but prominent adversaries are predicted to earn more than $1 billion in ransom payments (Cook, 2020).

The emergence of anonymous and international payment methods such as cryptocurrencies is believed to have helped fuel the growth of cybercriminals utilising ransomware to earn money (Zimba & Chishimba, 2019a). The current trend indicates that targeting individuals are less lucrative for criminals, as individuals often cannot pay large sums to decrypt their files. Therefore, cybercriminal groups increasingly target enterprises to maximise monetisation, a practice that has been dubbed Big Game Hunting (BGH) (Simons (2019); Meyes (2019)). Threat actors go after organisations with high turnovers and who heavily depend on digital assets to create an incentive to pay the ransom.

Threat intelligence analysts have found that cybercriminal groups increasingly adapt and implement Advanced Persistent Threat (APT) tactics to successfully inject ransomware in their chosen target's systems (Frankoff & Hartley, 2018). Skilled cybercriminal groups select their targets and run multistage ransomware campaigns against them, an operation pattern that previously has been associated with nation-state actors. Successfully attacking large organisations requires a higher level of expertise and planning than spreading ransomware at random (Frankoff & Hartley, 2018).

2.2.1 Ransomware-as-a-Service

Some threat actors have started to offer Ransomware-as-a-Service (RaaS) as an addition to their business. Utilising the franchise method, criminals are selling ransomware online (Meland et al., 2020). The RaaS services offered vary in shape and form. Tailor-made attacks, pre-compiled binaries ready for launch, or interface-based services where the buyer provide the intended victim's details, are typical (Meland et al., 2020). The common denominator is that an individual or group buys the ransomware from a provider, removing the requirement of the initiator of a RaaS attack having software development skills. If a RaaS attack is successful, the vendor commonly gets a percentage of the profit. In 2017, researchers from security company Carbon Black (2017) estimated that forums retailing ransomware experienced a growth surpassing 2.500% a year. However, Meland et al. (2020) monitored the open, English speaking markets Dream and Wallstreet during the period 2017-2019. In that time, the researchers only found 69 listings of RaaS for sale instead of the 45.000 listings Carbon Black (2017) argues exists. Meland et al. (2020) found that the RaaS offered openly often is a scam or of bad quality, which makes it plausible that the more decent strains of RaaS are sold at closed markets. Europol (2020) further elaborates on the topic of RaaS, stating that well-established RaaS-vendors have elevated the criteria for buying ransomware. Potential customers must demonstrate sufficient skills and continuous activity to become a trusted affiliate (Europol, 2020). Prominent ransomware groups only work with trusted affiliates if they decide to rent out their tools. From a business point of view, this is logical, as having skilled and trusted affiliates is likely to increase operational security and income. For ransomware families operated as RaaS, the TTPs used in attacks may vary greatly, as affiliates have different methods.

2.2.2 Advanced Persistent Threat Actors

Advanced Persistent Threats (APT) are defined as threat groups that perform continuously, targeted operations with a clear goal (Bahrami et al., 2019). Advanced Persistent Threat actors can be divided into two categories, namely nation-state actors and cybercriminal groups (Gupta, 2018). The key difference is their goals. According to Bahrami et al. (2019)s description, criminal groups perform attacks purely for economic gain, while nation-state actors are generally not financially driven. Tsakalidis and Vergidis (2019) elaborate that operations conducted by APTs show signs of a well-organised actor with advanced technical skills that operate covertly and leaves minimal traces behind, regardless of

object.

Wen et al. (2018) states that the following characteristics apply to APTs:

- APTs aim at specific targets
- APTs use advanced techniques
- APTs are likely to have access to zero-day exploits
- APTs are persistent in gaining access to their targets
- APTs can operate covertly for longer periods

MITRE keeps track of threat actors that are frequently discussed in media and technical reports. Some are believed to be state-sponsored, and some are independent criminal groups (MITRE, n.d.-b). Prominent nation-state APTs are generally assigned a number that identifies the APT across different platforms as common denominators, leaving aside additional names. One example is APT28, which is also known as Fancy Bear, Sofacy, and SNAKEMACKEREL, amongst other things (MITRE (n.d.-a); Crowdstrike (2019)).

Cybercriminal groups are organised units that aim to profit economically from their activities. The groups focusing on ransomware operations are highly skilled and perform attacks over multiple stages (IBM Security, 2020). Although notorious cybercriminal groups are both advanced and persistent, they are commonly referred to as cybercriminal or eCrime.

2.2.3 Human-Operated Ransomware

Attacks where the threat actors are hands-on-keyboard during large parts of the ransomware campaign has been a growing threat in the past years (Microsoft Security, 2020). So-called human-operated ransomware (HumOR) attacks differ significantly from automatically spread ransomware as the cybercriminals leverage TTPs previously associated with nation-state threat actors. HumOR attacks may use commodity malware to infiltrate a system, but the attackers perform the lateral movement and subsequent actions. However, many HumOR attacks are malware-free up until the ransomware is deployed (Turedi, 2020). The human knowledge of the system and common security misconfigurations, paired with the ability to adapt to the environment, makes such attacks pernicious (Simpson, 2021).

All the most successful cybercriminal threat actors in 2020 ran HumOR attacks. There is a direct link between the evolution of BGH to the increase in HumOR. According to PwC (2020) the perceived profitability from such attacks is one major driver for the continuous growth of cybercriminal groups effectuating such attacks. Not only do they target organisations they can elicit large sums from, but contrary to, e.g., banking scams, the payment goes directly to cryptocurrency wallets controlled by the attackers. This eliminates the need for intermediaries and complex money-laundering operations.

Another side effect of HumOr is increased time spent inside compromised environments. The attackers can spend everything from a couple of days to months inside a breached network (Meijerink et al., 2019). During this time, they may move laterally, download sensitive files, or just stay dormant. Threat actors seeking to develop their tools further to avoid detection and generate havoc may also use such opportunities to do additional reconnaissance (PwC, 2020).

2.2.4 Ransomware Cartels as a Business Model

Profiled groups seem to join forces and operate in a cartel-like structure (Bisson, 2020). To date, only one such ransomware cartel is known. One prominent effect of these coalitions is that the different actors exchange tactics and techniques (Scroxton, 2020). For instance, in June 2020, it was announced that the Ragnar Locker group had joined forces with the group behind the now-defunct Maze ransomware group (Abrams, 2020b). Shortly after, the Maze group started to reuse the technique of deploying ransomware from a virtual machine to avoid detection (Scroxton, 2020). The ransomware cartel was estimated to consist of Maze, Ragnar Locker, LockBit, and SunCrypt. Ties between the groups include shared techniques, cross-leaking data, and shared infrastructure (Bisson, 2020). The claimed cartel seized operations and public appearances in November 2020. Both Maze and SunCrypt announced that they would retire around the same time. Maze is believed to have moved on to using Egregor (Abrams, 2020c). Some researchers speculate that the sudden cease of operations is due to the increased public attention. The exposure also led to more law enforcement scrutiny. On February 9, 2021, Ukrainian law enforcement conducted a joint operation with the US and French authorities against several Ukrainian nationals believed to be deeply involved with Egregor ransomware operations (Constantin, 2021).

Ragnar Locker's achievements have also been featured on MountLocker's DLS. Despite these affiliations, there are no indicators that the Ragnar Locker group

cooperates with access brokers (CrowdStrike, 2021). The group likely facilitate and conduct all stages of an operation themselves.

2.2.5 Ransomware Attack Vectors

An attack vector is the method and means used to gain unauthorised access to a target system (Kaspersky, 2020). Threat actors can use several approaches to access a system or launch a full-scale cyber-attack, ranging from exploits and flaws to physical measures (Simmons et al., 2014). A significant number of attacks start with some form of social engineering (Zimba & Chishimba, 2019b). Social engineering is when a threat actor exploits human nature, often by deceiving the victim to perform an action or reveal information (Mitnick, 2011). The most common method is phishing, which is when an adversary gets an unsuspecting person to click on a malicious link or document through an email (IBM Security, 2020). A more advanced threat actor would often have a layered approach towards targeting an enterprise. These tactics often start with a reconnaissance phase on the target to make the communication appear trustworthy (Verizon, 2020). Email is the single most used attack vector for ransomware and accommodates 59% of attacks (Osterman Research, 2016). However, according to IBM Security (2020), operations where threat actors gained access through technical exploits increased drastically through 2020. Such operations consist of scanning the target for vulnerabilities, gathering information about the technical structure, and exploiting the weaknesses (Dahle, 2020). One feature commonly exploited is the Remote Desktop Protocol (RDP). RDP enables a device to be connected to remotely. Threat actors can access open RDP ports if the port is exposed, not adequately secured, or unpatched (Microsoft, 2019). In 2020, RDP exploits were the primary attack vector used to exploit small and medium-sized businesses (Coveware, 2020). For larger organisations, email phishing is still the most prominent attack vector. After the initial compromise, threat actors will try to establish persistence on the system.

2.2.6 Operational Maturity and Stages of Attack

Advanced ransomware attacks are typically conducted over multiple stages (Lord, 2018). Several classifications of such stages exist and ranges from the blanket approach of the Cyber Kill Chain versions to empirical models created by researchers (Lockheed Martin (n.d.); Kumar and Ramlie (2021)). Figure 2.1 displays the Cyber Kill Chain, which aims to identify the steps threat actors use to reach their primary objective. The model is derived from the phases in offensive military operations. The Cyber Kill Chain is one of the most well-known models for mapping the stages of an attack. For instance, MITRE used it as a foundation when developing the ATT&CK framework. However, the steps may not describe the steps of current ransomware operations very well.

Figure 2.1: The Cyber Kill Chain (Lockheed Martin, n.d.).

The challenge with models is that they may be too broad, too narrow, or outdated. The ransomware threat landscape has shifted and evolved drastically through 2020. Researchers at Trend Micro have developed a model for the stages of a ransomware attack anno 2020-2021 (Logan et al., 2021). Figure 2.2 shows an overview of the identified ransomware attack stages used in current ransomware campaigns.

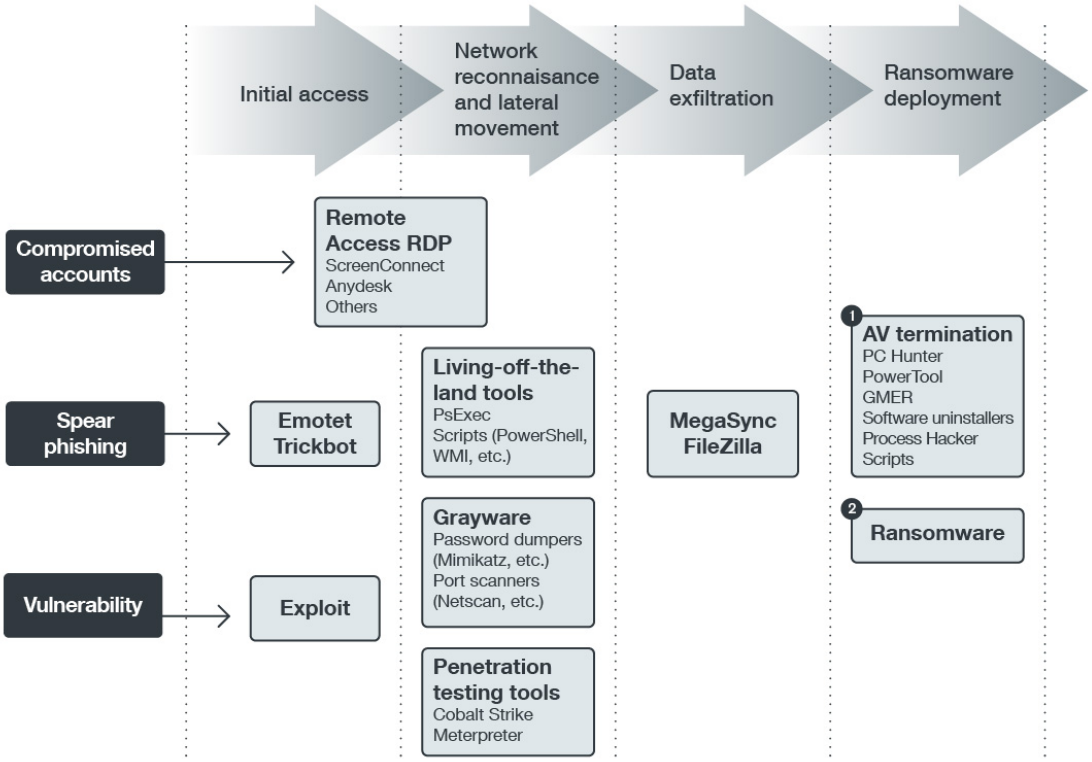


Figure 2.2: Common stages in a 2020 ransomware attack (Logan et al., 2021).

The model considers the growth of targeted HumOR operations and the increased complexity of ransomware attacks. In comparison, security firm Exabeam did similar research on the state of ransomware in 2016, and the ubiquitous stages of attack they discovered are profoundly different (Exabeam, 2016). In 2016, the phases in Figure 2.3 made up the average ransomware attack:

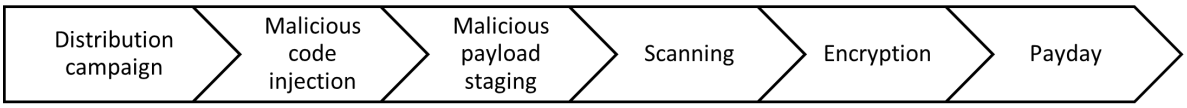


Figure 2.3: Common stages in a 2016 ransomware attack (Exabeam, 2016)

Regarding Ragnar Locker, the stages proposed by Logan et al. (2021) indisputably the most accurate.

The capabilities of a strain and the TTPs used can also reflect the operational maturity of a threat actor. Human-operated ransomware often has a covert and overt phase (Young & Yung, 2017). Attackers aim to stay undetected while moving laterally and exfiltrating data. The visibility in the overt phase is caused by

the encryption of data, which, together with the ransom demand, comprise the main characteristics of this form of malicious software (Young & Yung, 2017). Operational maturity is a general measure of the overall consistency, resilience, and sophistication of the software design, management, and operation (Churchman, 2017). For ransomware strains, this is judged based on how it is coded and the implemented obfuscation and evasion techniques (Lord, 2018). The maturity of the threat actors is generally assessed based on how they manage attacks and their applied TTPs.

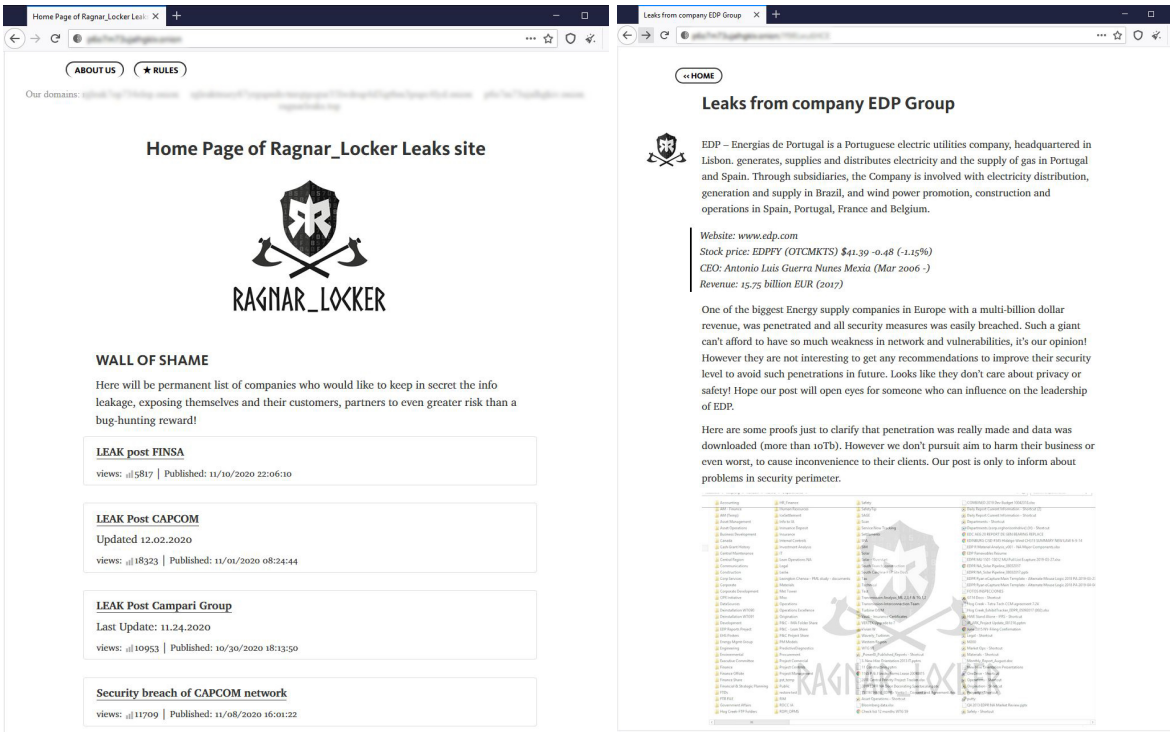
2.2.7 Attack Infrastructure

Threat groups must have a suitable infrastructure in place to facilitate extensive attacks (Steffens, 2020). Command-and-Control servers (C2 or C&C) are core elements in such a setup. The C2 structure generally consists of numerous servers situated at different locations, and threat actors tend to reuse much of their established infrastructure in different attacks. Threat actors can establish C2 servers in numerous ways, from owning the hardware themselves to renting or using compromised hosts' networks. The latter is often referred to as proxy servers or botnets (Steffens, 2020). One abiding trend is Bulletproof Hosting (BPH), where a service provider has little to no restriction on how their services are used (Europol, 2020). BPH services can be hardware-, software-, or application-based, and the providers often spread their resources to increase their resilience if law enforcement shuts one service down (Goncharov, 2015).

Fundamentally, C2 servers are used to communicate with beacons in breached environments and drop tools(Steffens, 2020). However, C2s have several use-cases, such as domain fronting, temporary storage of stolen data, key generation, intelligence gathering on victim IDs, and masking the attacker's location (Cabaj and Mazurczyk (2016); Johnson (2018)). For instance, some threat groups use fileless attacks; this is a method where malware payloads are placed directly into the memory of victim hosts using C2 servers. Such an approach leaves few traces once the memory is overwritten, or the system loses power (Europol, 2020). The attack infrastructure is the backbone of most large operations, which means that threat actors will take steps to secure it. Such steps often include redirecting traffic through several nodes and obfuscating it (Steffens, 2020).

2.2.8 The Ragnar Locker Group and Novel Approaches to Cybercrime

A financially motivated cybercriminal group operates Ragnar Locker. According to Abrams (2020d), Ragnar Locker first surfaced in December 2019. Both the group and the ransomware are referred to as Ragnar Locker by most researchers and organisations. The exception is Crowdstrike, which have dubbed the group Viking Spider. Ragnar Locker is classified as a advanced ransomware due to how it is coded and the techniques it employs (Lord, 2018). The group acquires and exfiltrates the targets data before manually deploying the Ragnar Locker payload on the victim's domain (Tavares, 2020). Figure 4.12 displays how the cybercriminals publicly extort their non-paying victims on their DLS "Wall of Shame".



(a) Ragnar Lockers "Wall of Shame". (b) Victim extortion.

Figure 2.4: Screenshots from the Ragnar Locker home page.

Examinations of completed attacks have shown that the Ragnar Locker operators often gain access to a targeted entity's network through RDP and MSP exploits. However, the threat actors will adapt to use different attack vectors if necessary; in the early days, the group used phishing, and recent attacks show that they have targeted vulnerable VPN devices belonging to a wanted victim

(Ilaschu, 2021). During the data-gathering period of this paper, the threat actors mainly targeted Remote Management Software used by MSPs or RDP exploits to gain access to the victim system. Once a foothold is established on the system, the threat actors have been observed using native Windows tools such as PowerShell and RDP, combined with credential-stealing software such as Mimikatz, to move laterally inside the network. The operators will obtain administrator rights and create Group Policy Objects (GPOs). The GPO will then be used to launch the ransomware (Loman, 2020). The group behind Ragnar Locker used variations of these techniques until April 2020. In May 2020, the first reports of the group using virtual machines to deploy the Ragnar Locker ransomware surfaced (Loman, 2020).

The novel approach of placing an Oracle VirtualBox running Windows XP machine on the target system and then deploying Ragnar Locker from within it, was the first technique that set Ragnar Locker apart from other ransomware families. According to Loman (2020), in attacks where the virtual machine (VM) technique is used, a Microsoft Windows Installer (MSI) package is copied to `C:\ProgramFiles(x86)\VirtualAppliances` on the victim machine. Key components of the MSI package are an Oracle VirtualBox hypervisor and a virtual disk image (VDI) of a MicroXP OS. MicroXP is a stripped-down version of the Windows XP SP3 OS. The ransomware executable is located at `C:\` in the VM. A batch file located in `C:\DocumentsandSettings\Administrator\StartMenu\Programs\Startup` contains the commands to initiate the execution of the payload.

The Ragnar Locker group has pioneered attack techniques and tested new methods to pressure victims into paying the ransom. In November 2020, the threat actors used a hacked Facebook account to buy ads (Krebs, 2020). The ads concerned the attack on the Italian Campari Group, and the threat actors pushed the message that they had stolen vast amounts of confidential data and that the Campari Group are liars trying to claim otherwise. This is allegedly the first time cybercriminal threat actors have bought social media advertising to pressure victims (Krebs, 2020).

A high-level overview of a Ragnar Locker attack is displayed in figure 4.11.

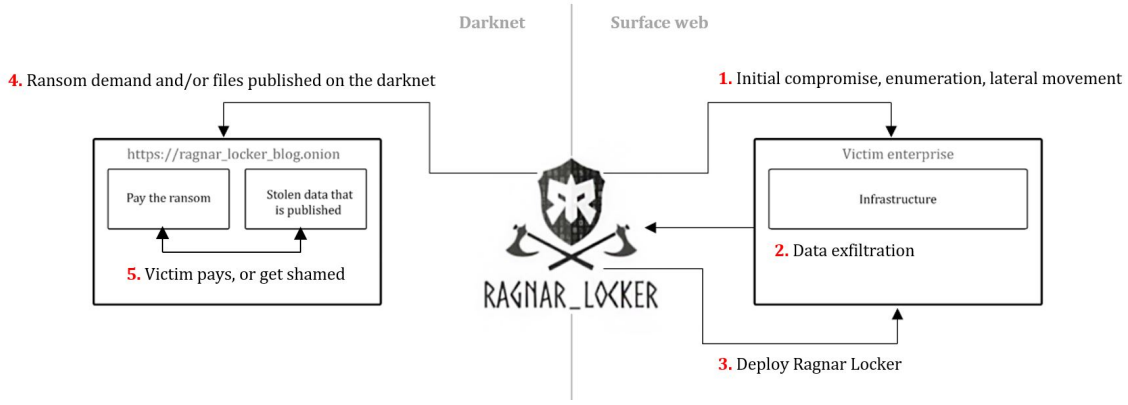


Figure 2.5: The motions of a Ragnar Locker attack.

Reviewing the Ragnar Locker DLS postings shows a significant difference in high activity periods and low activity periods. Figure 2.6 exhibits post statistics for the period June 2020 to March 2021. Due to the DLS being upgraded and redesigned in late May 2020, no earlier posts are available. The peaks in posting activity can be due to multiple reasons. The threat actors may have decided to release breach data in bulks or take it easy in periods after monetising their efforts. They may use the quieter periods to develop their craftsmanship. Another theory is that they might assist affiliated cybercriminal groups in their operations, resulting in little time to conduct campaigns themselves (Bisson, 2020).

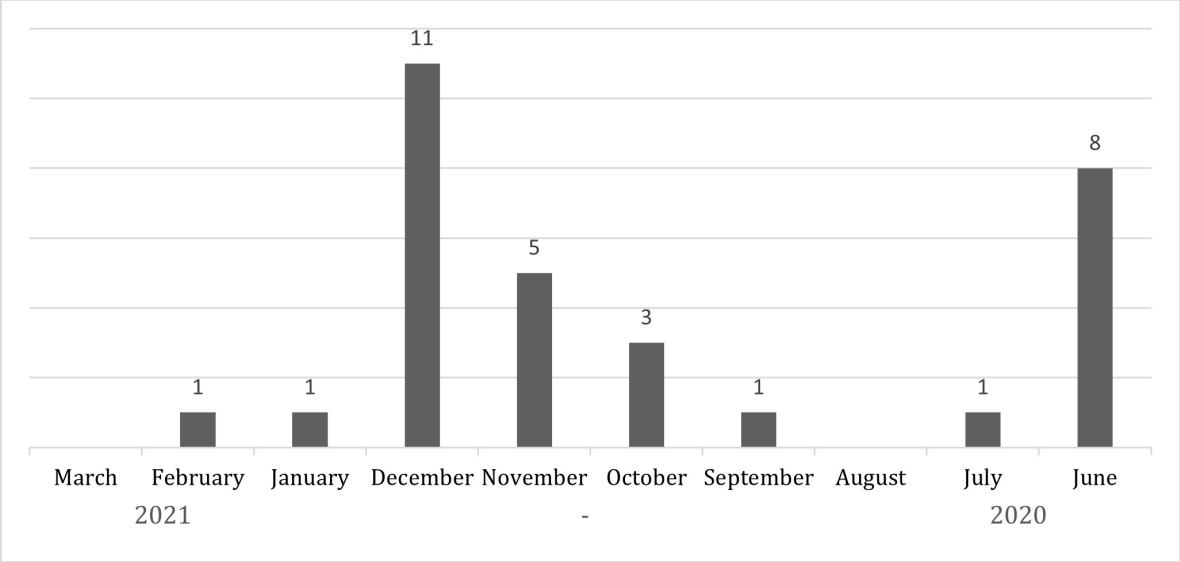


Figure 2.6: The number of leaks posted on Ragnar Lockers DLS from June 2020 to March 2021.

2.2.9 Cryptography

The majority of ransomware families make use of encryption algorithms to obscure data, thus making it inaccessible without the correct decryption key (Al-rimy et al., 2018). To encrypt and decrypt data, cryptographic algorithms are used. Cryptographic algorithms can be divided into two main categories according to Paar and Pelzl (2010), namely symmetric key encryption and asymmetric key encryption. The practice of combining symmetric key encryption algorithms and asymmetric key encryption algorithms to achieve more robust cryptosystems is termed hybrid key cryptography (Paar & Pelzl, 2010). Figure 2.7 exhibits a high-level overview of the different categories of cryptographic algorithms.

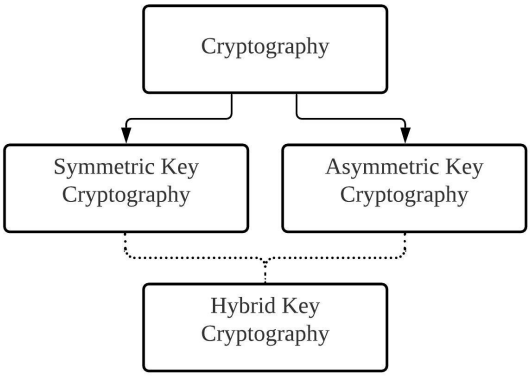


Figure 2.7: Overview of cryptosystems (Paar & Pelzl, 2010)

Symmetric key encryption (SCR) is the simplest encryption method. SCR uses the same key to encrypt and decrypt the files (Al-rimy et al., 2018). The main benefit of using an SCR algorithm is that it is fast. Since only one key is in use, the encryption and decryption process is quicker, as it does not require the same amount of complexity and computer resources (Anton, 2019). Speed and modest use of computing power are beneficial in ransomware attacks, as slow and resource-heavy operations are more likely to be detected and interrupted by the victim (Arabo et al., 2020). The drawback of SCR is that using one single key makes it easier to break the encryption.

Asymmetric key encryption (ACR) offers a significantly more robust encryption algorithm. ACR uses different keys; a public key is used for encryption and a private key for decryption (Ahmadian et al., 2016). Due to the increased complexity, this encryption method is considerably slower than SCR. The encryption process requires a substantial amount of computing power, which makes it stand

out from normal operations (Arabo et al., 2020). The advantage is that items encrypted with ACR are computationally infeasible to decrypt without the correct private key (Saravanan & Krishnan, 2014).

To solve the challenges of speed and complexity, ransomware developers started to utilise Hybrid key encryption (HCR). This method first benefits from the SCR speed by encrypting files using an SCR algorithm (Symmantec, 2018). When the data is encrypted, the SCR key is encrypted using an ACR algorithm. Such an approach is fast, and the encryption is virtually impossible to break without the ACR algorithm's private key. Using HCR also mitigates the risk of a private key being unintentionally disclosed to victims, as the private ACR key never leaves the attackers systems (Davies et al., 2020).

2.3 Analysis Methods

The practice with hybrid analysis arose to better interpret how malicious software operates on a system. The hybrid analysis combines the strengths of static and dynamic analysis techniques with additional focus areas, such as memory forensics and network analysis (Galal et al., 2016). Whichever approach to hybrid analysis is used, a combination of techniques will be applied. Analysing ransomware using a hybrid analysis framework will uncover more of the strains traits than what is possible with just using one technique (Roundy & Miller, 2010).

2.3.1 Hybrid Analysis Frameworks

Eskandari et al. (2013) developed a technique where the dynamic analysis is performed first to map the actual Application Programming Interface (API) calls made. The extracted results from the dynamic analysis are combined and further analysed using static methods. One of the main benefits of this approach is that it minimises time spent on dynamic examination and bypasses the challenges with packing. Code that is packed will exist in a compressed state while stored and extract itself in memory once executed (Bayer et al., 2006).

Nonetheless, most approaches to hybrid analysis start with static analysis. The methodology applied by the HMF is to start with static analysis, followed by the dynamic, network, and forensic analyses (Schmitt, 2019). The framework concatenates existing methods for reverse engineering malicious code with forensic examination techniques. The first key element added by the HMF is net-

work analysis, which helps map what and how a malware strain communicates (Schmitt, 2019). The second notable trait of the HMF is that the framework has a segment dedicated to forensics. The forensics approach adds value to practitioners who deals with real-life incidents by empathising sound forensic practices and procedures. The phases outlined in the framework is a mix between tools and analysis focus areas. To summarise the findings, the behavioural traits of the analysed species are plotted into the MITRE ATT&CK framework. Although the ATT&CK is not a direct part of the HMF, Schmitt (2019)s decision to utilise the framework to highlight findings adds extra value. MITRE have indexed offensive techniques and assigned them an ID. Applying standardised definitions is convenient when generating overviews and comparing TTPs. The HMF has been successfully tested on Cerber, Locky, and Maktub in an academic study. It is unknown if the framework has been used during real-life investigations. A visual representation of the framework is presented in Figure 2.8.

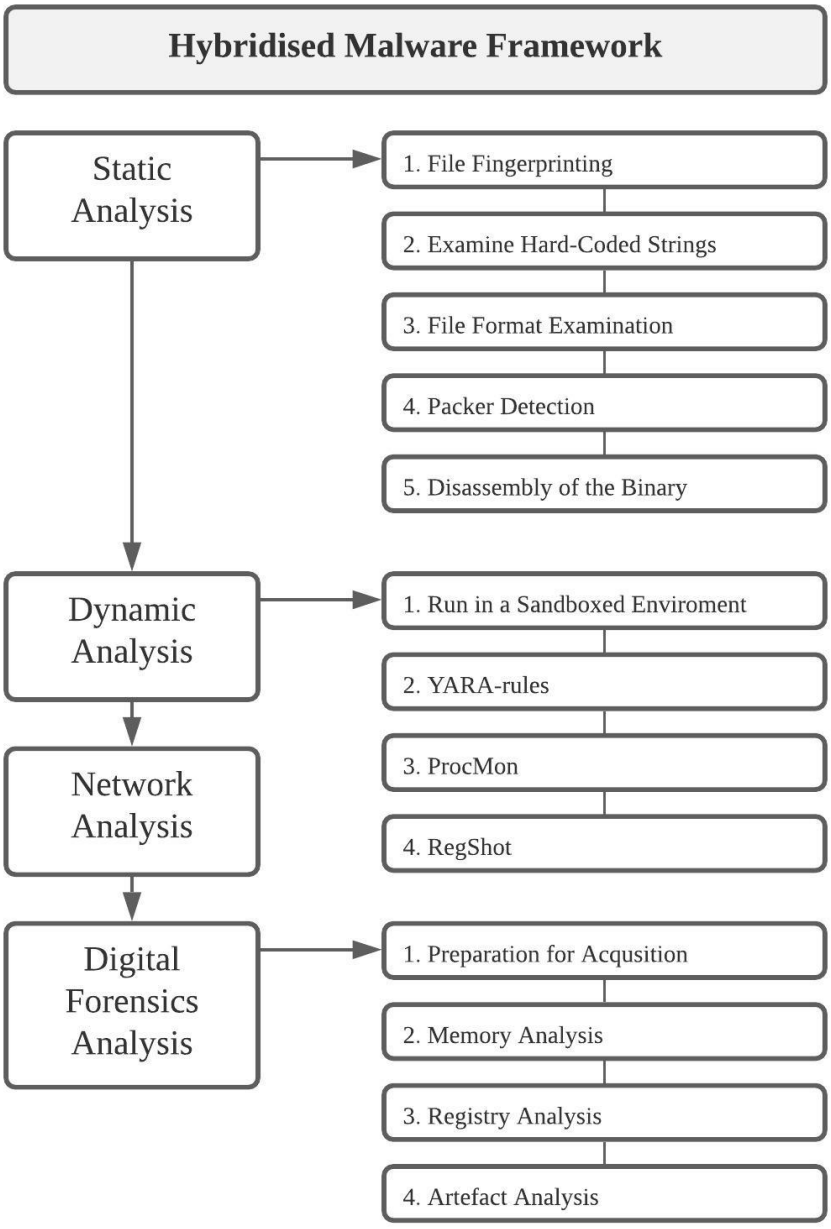


Figure 2.8: Diagram representation of the HMF.

2.3.2 Static Analysis

Static analysis is an umbrella term for techniques that inspect malware without executing it. Static analysis techniques are useful for examining the source code, binary strings, Windows Portable Executable (PE) files, and the execution paths of the malicious software (Wang & Wang, 2015). The PE file header contains information that the program needs to run and can provide an analyst with helpful

information about the strain (Saxe & Sanders, 2018). Program files intended to run on Windows-based machines follow the PE structure. This includes files with extensions such as EXE, DLL, and SYS. In the HMF, the static analysis also includes verifying the file using a cryptographic hash algorithm, preferably SHA-256. This practice is referred to as file fingerprinting in the framework (Schmitt, 2019). The cryptographic hash value of a file is a unique identifier, much like a person's fingerprint. In malware research, MD5, SHA1 and SHA-256 are the most commonly used cryptographic algorithms (Dunham, 2013). The main difference is their length. The SHA-256 is the longest and most complex of the three. The cryptographic hash value is based on the file's content, meaning that metadata such as file name does not affect the hash. However, if the content changes just the slightest, the hash value will change. Researchers such as Sarantinos et al. (2016) argues that although cryptographic hashes are unprecedented in detecting identical files and confirm their integrity, they will not detect files that are merely similar or related. To identify similarities between files, fuzzy hashing can be used. Fuzzy hashing combines cryptographic hashes, rolling hashes, and piecewise hashes (Sarantinos et al., 2016). Simply put, this segments the hash and compares the bulks rather than the entire value. Based on this, cryptographic hashes are elemental when analysing ransomware and conducting forensic examinations, as they are unique to the file in question. If the aim is to detect similarities, however, fuzzy hashing is more effective.

One of the main benefits of static analysis is that it is relatively safe, as the malicious program is never executed (Galal et al., 2016). It can also be reasonably quick, which is convenient during triage. However, if the code is packed or obfuscated, an in-depth analysis using static techniques will be complicated and time-consuming (Saxe & Sanders, 2018). Code that is packed will exist in a compressed state while stored and then extract itself in memory once executed (Arntz, 2017). According to Al-rimy et al. (2018), advanced strains often use packers to conceal their payload. Although packing of code is common practice in benign programs to reduce the storage size, a packer can also be used as a form of anti-analysis technique. By compressing and sometimes encrypting, the malicious segment of the code, the threat actor complicates code disassembly (Ligh et al. (2011); Assis et al. (2019)). Verma et al. (2020) propose using a grey-level co-occurrent matrix (GLCM), a way of visualisation using greyscale images, to classify obfuscated malware quickly. GLCM proved effective during testing on known families but was less successful on unknown strains and if the malicious code was hidden within bigger benign binaries.

2.3.3 Dynamic Analysis

Dynamic analysis is performed by executing programs and analysing their behaviour at runtime. Ransomware is commonly run and monitored in a controlled environment, such as a sandbox (Al-rimy et al., 2018). One of the main benefits of dynamic analysis is that the researcher can observe how the strain interacts with the system, such as API calls made and behaviour in memory (Bayer et al., 2006). For a process to call an API function, it must first load the right Dynamic-Link Library (DLL) into memory. By masking malicious functions within DLLs, threat actors can avoid detection mechanisms such as antivirus (Barkly, 2018). The reason for this is that DLLs run in the system's memory, leaving few traces elsewhere. Thus, threat actors can use DLL instead of EXE files to mask the ransomwares actions inside processes (Ligh et al., 2011). Such attacks are often referred to as *fileless attacks* (Kerr & Ewing, 2018).

Arabo et al. (2020) monitored the DLL API calls, CPU, and memory usage by different ransomware families. The memory usage was relatively stable and low for the strains tested while the CPU would be higher, bench-marking between 20-40%. The key indicator was a significant increase in write counts. The sudden increase in disk writes reveals when the strains start encrypting the system. In addition to interaction with the OS and resource usage, dynamic analysis allows researchers to reproduce some of the artefacts dropped by ransomware strains. One example is ransom notes. On the other hand, a dynamic code analysis does not always produce an impression that is transferrable to an actual situation (Ali & Papadaki, 2018).

Many ransomware strains are aware of their environment. If such a strain detects that it is run in a sandbox or debugger, it will likely refrain from executing the malicious payload (Bayer et al., 2006). To avoid the ransomware detecting that it is in an artificial environment, sandboxes often emulate everyday events, such as internet traffic. A variety of sandboxes exist, from agent-based to agentless and cloud-based (Ali & Papadaki, 2018). With an assortment of solutions available, the chance of successfully overcome the ransomware's evasion techniques increases.

YARA Rules

YARA is used to identify and classify malware samples (Sikorski & Honig, 2012). It is an open-source project and allows for creating custom AV-like signatures. The YARA syntax resembles that of the C language (YARA, 2020). The rules are logical descriptions based on strings and binary patterns found in malware. YARA strings can be either hexadecimal values, plain text, or regular expressions. In general, every YARA rule has two sections, namely a strings description and a condition (Balci et al., 2020). The rule will be triggered if a string in a scanned file matches a string in the YARA rule so that the condition is met. Listing 1 provides a basic example of the structure of a YARA rule. As with code in general, it is considered good practice to add a metadata section with the author's name, date of creation, and a description of the rule in the header.

```
rule Name
{
    strings:
       $string = "string"
    condition:
       $string
}
```

Listing 1: Fundamental YARA rule syntax.

YARA rules can be generated using automated tools or be written manually based on static analysis of the malware in question. The rules are highly customisable. However, the larger and more complex a rule is, the more computing power it needs. The complexity depends on the number and type of strings (Naik et al., 2020b). There are two main caveats when writing YARA rules. If the rule is so specific that it only detects one particular malware sample, it is no more valuable than a cryptographic hash value. Suppose the rule is too broad or include conditions that are likely to appear in benign files as well, one risk generating many false positives. Upon further research on methods to improve the effectiveness of YARA-rules, Naik et al. (2020a) suggest implementing fuzzy hashing alongside YARA to minimise the complexity and increase the detection of malicious files. Such advanced solutions fall outside the scope of this paper. Nevertheless, writing basic and effective YARA rules that detects a ransomware family is possible (Roth, 2015). YARA rules can be shared and accessed on many platforms, but the most common is to use a vetted Threat Intelligence Platform. Such platforms are discussed in Section 2.3.7.

2.3.4 Network Analysis

Network analysis entails investigating collected network traffic for suspicious events and determine the consequences of discovered artefacts (Khan et al., 2016). Uncovered data commonly includes contacted C2 servers, IP addresses used, and the IP of targeted systems. Network activity and artefacts can provide a wealth of forensic evidence (Ligh et al., 2014). According to Khan et al. (2016), most network analysis aims to uncover malicious activity either in the form of packets or irregular traffic patterns. Several tools exist for these purposes, but packet capturing tools such as Wireshark appears to be the most used, along with device logs (Sammons, 2015). For malware, network communication can take many forms, resulting in detection being one of the main challenges. Prominent malware, including ransomware, often utilises script obfuscation techniques and modifies network traffic to emulate benign requests to avoid detection (Sanders, 2017). For instance, attackers may encode shellcode and hide it between two `script` tags in an HTTP packet. This way, they can exploit vulnerable services with a higher chance of staying undetected.

The style and content of ransomware C2 communication may vary greatly throughout a breach (Sanders, 2017). However, one key indicator is network traffic occurring at regular intervals, indicating that the communication stems from a beacon. Most attackers install beacons on breached systems and use them to communicate back to their C2 servers. Such implants commonly masquerade as legitimate binaries and mimic normal encrypted network traffic. The communication intervals will depend on the operation and the attackers' goals; in brief operations, short intervals are used. In more pervasive attacks, longer intervals are commonly used to lessen the risk of being detected (Borchani, 2020). To further enhance the stealth of the beacon, advanced threat actors add extra layers of distortion. Such distortion techniques include appending a random number of milliseconds to the beaconing interval and rotating IP addresses. Research performed by Borchani (2020) show that distorted beaconing, beaconing with skipped exchanges, and distorted beacons that skip exchanges is possible to detect by creating sorted clusters.

Many benign functions also beacon to web servers, so discovering communication occurring in somewhat regular intervals does not necessarily mean that it is a malevolent beacon. However, threat actors make their traffic appear like legitimate network traffic. One example of masked traffic is HTTP GET and POST request modified with the Cobalt Strike Malleable C2 profiles (Kim, 2018). The

standard Malleable profile sets the host header to `www.amazon.com` and the uniform resource identifier (URI) is set to `/s/ref=nb_sb_noss_1/167-3294888-0262949/field=keywords=books`. Most attackers do not use the standard profiles, as several detection signatures exist, but may mask traffic in a similar manner (Kim, 2018).

In the case of actual ransomware incidents, network data can be obtained from numerous devices and solutions. Endpoints, firewalls, SIEM solutions and ISPs are some entities that may hold valuable information about network activities (Borchani, 2020). All the discussed factors can influence how network analyses are performed.

2.3.5 Digital Forensic Analysis

A digital forensic examination is a process that uses science and technology to assess digital objects to answer questions about events that occurred (Abulaish & Haldar, 2020). From this definition, one can determine that a forensic investigation happens after an event has occurred or to develop and test theories. On a high level, digital forensics examinations will often try to answer what has happened, how it happened, and what specific artefacts are related to the incident.

Another aspect of digital forensics that is often highlighted is that the uncovered evidence should be admissible to a court of law (Malik, 2018). To be classified as valid evidence, the data must be obtained and examined scientifically. Montasari (2016) details the four fundamental principles of digital forensic investigations: audibility, repeatability, reproducibility, and justifiability. In addition, the process of how the evidence has been handled, the Chain of Custody, is of importance. To comply with the Chain of Custody requirements, factors related to the *who, how, why, where,* and *when* must be documented (Prayudi & SN, 2015). Several digital forensics investigation models have been developed to facilitate evidence extraction and analysis of electronic evidence.

Montasari (2016) critically reviewed eleven digital forensics investigation models used in law enforcement, incident response, and commerce against the Daubert criteria. The Daubert criteria is commonly used by judges, primarily in the US, to determine the relevance and validity of evidence. The Daubert criteria consist of five requirements (Montasari, 2016):

1. The theory or technique in question can, and has been, tested.
2. The technique has been subject to peer review and publication.
3. The potential error rate is known.
4. Standards controlling its operation exists and are maintained.
5. It is widely accepted within the relevant scientific community.

The study performed by Montasari (2016) concludes that none of the analysed models can be regarded as generic. Moreover, the assessed models were deemed too abstract to provide sufficient guidance. Several models were constructed based on the creators' personal experiences rather than scientific processes. Based on this, one can deduce that as long as the principles of a forensic investigation and the Chain of Custody is governed, the framework chosen as an investigation guideline is of less judicial importance.

Indicators of Compromise

Such an approach will produce artefacts and knowledge that can be used to improve the security posture of the victim organisation and hinder future incidents (Monnappa, 2018). Artefacts that exist due to a strain of malicious code are often concatenated into a list with identifiable traits, or IOCs (Obbayi, 2018). IOCs are a subtype artefact; an artefact is a piece of forensic data related to an event, while an IOC is a forensic item directly related to the threat (Greten, 2020). Therefore, IOCs can be used to identify threats. IOCs are often in the form of cryptographic hashes or strings (Makrushin, 2015). Known IOCs can later be used to fine-tune a system's security mechanisms so that they will detect known strains. The IOCs are matched against the files and traffic on a system. However, as they are based on known bad files, they are unlikely to discover new threats (Kerr & Ewing, 2018). Considering the rapid changes in the threat landscape, IOCs are usually quickly deprecated. IOCs are also unlikely to help prevent targeted attacks by HumOR, as such attacks often rely on custom tools or tools already present on the system. Despite this, IOCs from a specific attack can be helpful when assessing its impact and cleaning systems. Collecting, systemising, and implementing IOCs is a part of the threat intelligence cycle (Zanoramy Zakaria et al., 2017).

Memory Forensics

Memory forensics constitutes a set of investigative techniques that extract forensic artefacts from a computer's physical memory (Monnappa, 2018). Memory forensics can provide unprecedented insights into the runtime state of a system (Ligh et al., 2014). All actions on an OS generates specific changes in the Random Access Memory (RAM), which can be captured and analysed. The RAM comprises the main memory of a computer. It is volatile, meaning that it requires constant power and is not written to disk (Ligh et al., 2014). If the system loses power, the RAM will be lost. The memory contains information such as logged on users, running processes, recently executed commands, and open network connections (Uroz & Rodríguez, 2020). Advanced ransomware mainly resides in the RAM, which introduces new challenges to security professionals (Palutke et al., 2020). Due to this, it is essential that the memory of the compromised system is extracted for analysis. There are three main memory capturing techniques; software-based, hardware-based, and virtualisation based (Ruff, 2008). Ruff (2008) argues that although hardware-based acquisition preserves the memory with non to little interference, few clients support this method. As for software-based acquisition, memory dumps can be acquired from a running system by various tools, such as DumpIt[1], Belkasoft Ram Capturer[2], and FTK Imager Lite[3] (Joseph & Norman, 2020). If the ransomware is dynamically analysed on a virtual system, the investigator can take snapshots of the RAM using the visualisation software (Davies et al., 2020).

Davies et al. (2020) have found that it is possible to extract AES-keys generated by ransomwares such as Bad Rabbit, Phobos, and NotPetya from memory, given that a snapshot is taken of the memory while the key is present. AES is an SCR algorithm. Capturing the key could help restore encrypted documents, but it is a long shot for most enterprises due to the timing and locating issues. Nonetheless, the first instinct people often have when attacked with ransomware is pulling the plug on the system to minimise the spread of the infection. Shutting down the system leads to a loss of volatile information that could be valuable for investigative purposes (Davies et al., 2020).

[1] https://www.comae.com/dumpit-memory-forensics-malware-analysis/
[2] https://belkasoft.com/ram-capturer
[3] https://accessdata.com/product-download/ftk-imager-lite-version-3-1-1

2.3.6 MITRE ATT&CK Framework

The MITRE ATT&CK framework is used to map and index the behavioural traits of adversaries and the tools they use (MITRE, n.d.-c). ATT&CK is an abbreviation for Adversarial Tactics, Techniques, and Common Knowledge. The tactic is the aim – what the adversary or tool wants to achieve. The technique is the method used to accomplish the tactic, and the procedure is how the technique was performed. It is primarily used to index adversarial activity. The framework is public and consists of common tactics used by threat groups. Encompassed in the tactics are over 440 attack techniques with a description of each exploit (Al-Shaer et al. (2020); MITRE (n.d.-c)). The tactics and techniques are assigned an ID and described using common language, which makes them easy to understand (Strom et al., 2018). Having a common taxonomy for TTPs makes it easy to compare them across different platforms and organisations, using the same terminology.

The framework has several use cases, from assisting in red teaming and penetration testing assessments to cyber threat intelligence enrichment (Strom et al., 2018).

The framework has divided the motions of an attack into fourteen tactics, from reconnaissance to exfiltration and impact (MITRE, n.d.-c). These phases are derived from the four last stages in the Cyber Kill Chain. Al-Shaer et al. (2020) highlights that MITRE ATT&CK structure makes the framework valuable for investigators focusing on the forensic aspects despite not being the primary target audience. By correlating the found artefacts and IOCs with the information in the framework, one can create an overview of the attacker's TTPs. TTPs are important for recognising an ongoing campaign as they might reveal a combination of techniques that are not common in benign operations (Husari et al., 2019). Threat actors usually reuse strategies, and by monitoring the characteristics of their actions, it is sometimes possible to attribute new attacks to known threat actors (Hovmark & Schüldt, 2020). The MITRE ATT&CK framework collects information about known APTs, criminal groups, and malware. Figure 2.9 presents a high-level overview of how the different elements in MITRE ATT&CK correlate to one another.

Nevertheless, it is essential to note that the value of MITRE information depends on the recipient. Many organisations are not mature enough to make use of it, and some organisations use different methods and models, such as Microsoft's STRIDE and the Diamond Model (Petters, 2020).

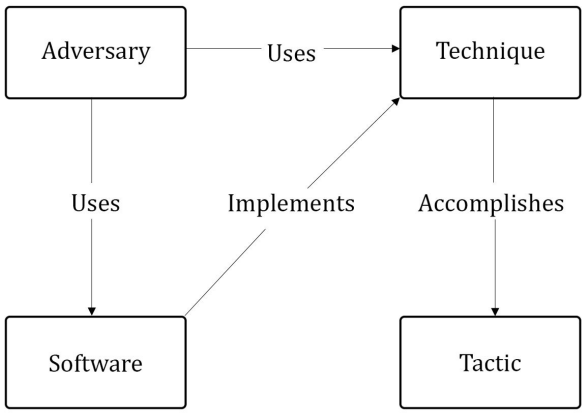


Figure 2.9: Relationship between components in the MITRE ATT&CK framework (Strom et al., 2018).

2.3.7 Threat Intelligence Platforms

Threat Intelligence Platforms (TIP) are resources that collect and structure threat intelligence, such as IOCs and TTPs, in a standardised manner (Tounsi & Rais, 2018). Serketzis et al. (2019) states that utilising information from TIPs is an integral part of proactive and reactive security despite that the update frequency needed, along with the emergence of targeted attacks, might impact their effectiveness. MISP, formerly known as the Malware Information Sharing Platform, is an example of open-source TIP widely used by the industry (MISP, n.d.). The MISP platform has four sharing levels: organisation only, community only, connected communities, and all (Wagner et al., 2016). The platform relies on volunteers and organisations contributing information, and it is the contributor who decides who should be allowed access to the data (Wagner et al., 2016). The MISP framework does not follow one set taxonomy when classifying threat intelligence but allows the users to customise it to their needs (Vinot, 2020). MISP requires some configuration, and not all have the resources available to run or engage with a MISP instance. One alternative to MISP is the AlienVault Open Threat Exchange. AlienVault is a free threat intelligence platform and can be used very simply directly in the browser. It also gives the user the option to leverage additional functionalities and APIs (AlienVault, n.d.).

2.4 Existing Analyses of the Ragnar Locker Ransomware

Tavares (2020) and Blaze Information Security (2020) analysed a Ragnar Locker strain with the SHA-256 hash `68eb2d2d7866775d6bf106a914281491d23769a9eda88fc078328150b8432bb3`. This is the strain that hit Energias de Portugal (EDP) on April 13th, 2020. The EDP attack was Ragnar Lockers first high-profile breach, putting the group in the spotlight.

Both researchers found that Ragnar Locker will terminate itself on systems where the strings in `LCIDLOCALE_SYSTEM_DEFAULT` matches that of a language used in the Commonwealth of Independent States (CIS) countries. Listing 2 details the countries avoided by Ragnar Locker.

Belorussian	Azerbaijani	Ukrainian	Moldavian
Georgian	Armenian	Turkmen	Russian
Kyrgyz	Kazakh	Uzbek	Tajik

Listing 2: Countries avoided by Ragnar Locker.

After checking the language settings, Ragnar Locker starts the encryption process (Tavares, 2020). The ransomware enumerates directories on the victim's system and copies the ransom note named `RGNR_<ID>.txt` to all directories except for twelve locations which are ignored (Tavares, 2020). `<ID>` represent an eight-digit victim ID. The directories that will remain unaffected are specified in Listing 3. The reason for not encrypting the listed files is that the threat actors still need the operating system (OS) to function normally. If system files are encrypted, threat actors' risk permanently rendering the computer useless, which would keep them from holding up their part of the deal. A bad reputation would impact their business negatively. Additionally, the victim needs access to a web browser to contact the threat actor (Tavares, 2020).

Windows	Windows.old	Tor Browser	Internet Explorer
Google	Opera	Opera Software	Mozilla
Mozilla Firefox	$Recycle.bin	ProgramData	All Users

Listing 3: Directories ignored by Ragnar Locker

Onwards, Ragnar Locker will enumerate all files, checking if they match the file names in listing 4 or have an extension that matches the extensions in listing

5. If the file does not match any of these criteria, a pointer to the file is added onto a stack array (Blaze Information Security, 2020). The files referenced in the stack array will be encrypted once Ragnar Locker has finished iterating through the system.

`RGNR_*`	`autorun.inf`	`boot.ini`	`bootfont.bin`
`bootsect.bak`	`bootmgr`	`bootmgr.efi`	`ntldr`
`bootmgfw.efi`	`desktop.ini`	`iconcache.db`	`thumbs.db`
`ntuser.dat`	`ntuser.dat.log`	`ntuser.ini`	

Listing 4: Files ignored by Ragnar Locker

`.db`	`.sys`	`.dll`	`.lnk`
`.msi`	`.drv`	`.exe`	

Listing 5: Extensions ignored by Ragnar Locker

According to Blaze Information Security (2020), the encryption algorithm used seems to be a modified version of the Salsa20 stream cypher. Salsa20 is an SCR algorithm developed for software where the encryption needs to be fast and encrypts each bit individually (Paar & Pelzl, 2010). Continuing, the files are encrypted by an RSA 2048-bit public key, and `_RAGNAR_` is added as a footer within the files (Blaze Information Security, 2020). The RSA algorithm is an ACR algorithm, which means the victim must use a private key to decrypt it (Paar & Pelzl, 2010). At this stage, the extension of the encrypted files has changed to `.ragnar_<ID>`. Tavares (2020) found that the Ragnar Locker strain did not check if a file had already been encrypted, so if invoked again, the file would be encrypted a second time. However, the analysis performed by Blaze Information Security (2020) contradicts this, stating that Ragnar Locker will not encrypt a file if the footer `_RAGNAR_` is detected in it.

Blaze Information Security (2020), indicates that the examined strain does not communicate over the network. In that case, it may be seen as proof that the Ragnar Locker group has operated hidden in the company's network and extorted data manually before launching the ransomware (Blaze Information Security, 2020). The date and time stamps of the unpacked executable displayed *Monday, 06.04.2020 19:57:20 UTC*. The date and time stamp can easily be modified, but if it is correct, it is seven days before Ragnar Locker started to encrypt EDP's data (Blaze Information Security, 2020).

Done (2020) analysed the encryption phase of another Ragnar Locker strain with the SHA-256 hash `EC35C76AD2C8192F09C02ECA1F263B406163470CA8438D054DB7ADCF5BFC0597`, which allegedly is one of the strains that make use of a VM on the victim system to evade security mechanisms. Based on the presented results, the abort parameters and avoided directories, files, and extensions are similar to what was found by Blaze Information Security (2020) and Tavares (2020). However, the analysis by Done (2020) primarily focuses on the encryption process and does not detail the differences between the two outlined evasion techniques used by Ragnar Locker.

2.5 Conclusion

The growth of agile and tailored ransomware is likely to continue, necessitating procedures that will facilitate thorough analysis using a hybrid approach. Openness and sharing of threat intelligence are essential for building technical defences and mitigate the risk before breaches. Much of the literature published on the different aspects of forensics in the last decade focus on niche approaches to solve a specific analysis aspect, often using machine learning and artificial intelligence. These approaches have not been discussed as they fall outside the scope of this paper. Despite the myriad of techniques, there seem to be few frameworks that endorse an agile and hybrid approach to forensic analysis of malicious software. The Hybridised Malware Framework by Schmitt (2019) has been discussed as it appears to be the most up to date and comprehensive framework. The MITRE ATT&CK framework has been introduced, along with the two main groupings of threat actors.

The MISP Threat Intelligence Platform is an open-source for contributing and receiving information about threats, which should be a natural step after analysing a ransomware strain. Lastly, three public analyses focusing on how Ragnar Locker encrypts data have been reviewed. The approach of the surveyed analyses allows IOCs to be extracted but hardly touch upon how Ragnar Locker evades security mechanisms, operate in memory, or interacts with the network.

3
Analysis

The analysis of the selected Ragnar Locker strains was, to the best of the researchers understanding, conducted in line with the HMF developed by Schmitt (2019). The procedure utilised by the HMF is outlined in Section 1.3 and Section 2.3.1. Although the analysis follows the HMF, the steps taken were more in accordance with the analysis result section divisions of Schmitt (2019) than the proposed framework. Figure 3.1 demonstrate how the analysis was carried out. The content and analysed aspects match the original framework, but the layout of the analysis pattern is slightly shifted. This is done to accommodate the analysis of Ragnar Locker better. The changes are not considered to be significant. Two steps in the framework are greyed out in Figure 3.1. The reason for this is that they are deemed only partially fulfilled when applying the framework in a controlled, academic setting. As the strains were run in local and cloud-based sandboxes, the need to forensically acquire data as described by Schmitt (2019) subsides. The forensic analysis of artefacts is greyed out due to the expected differences in artefacts generated by an actual Ragnar Locker attack and those generated in a sandboxed environment. The operators of the ransomware likely leave more valuable traces than those formed by the ransomware itself.

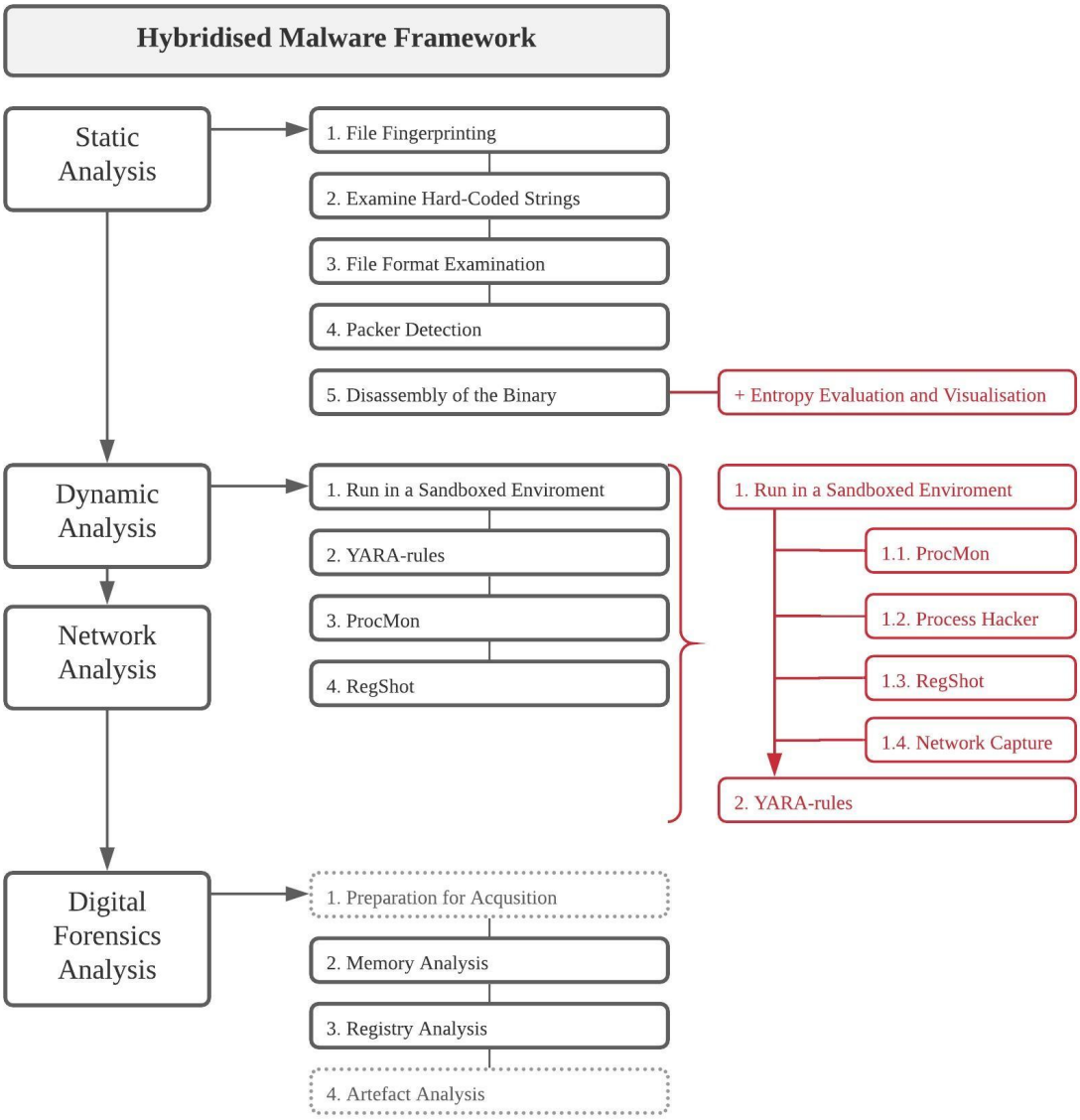


Figure 3.1: Diagram representation of the analysis based on HMF.

Strains of Ragnar Locker will be obtained from two different sources, namely *www.tutorialjinni.com* and *www.bazaar.abuse.ch*.

Strain 1

`SHA-256:9bdd7f965d1c67396afb0a84c78b4d12118ff377db7efdca4a1340933120f376`

Strain 2

`SHA-256:dd79b2abc21e766fe3076038482ded43e5069a1af9e0ad29e06dce387bfae900`

The static analysis is, in its entirety, performed on a local virtual machine running Windows 10. The dynamic analysis is primarily conducted using the same Windows 10 virtual machine. As an addition, it is supplemented with a local virtual Linux machine and the online solutions VirusTotal, AnyRun, and Joe Sandbox Cloud. The Linux machine is used to generate and capture network traffic. The online solutions will be used to validate and enrich the findings. Data for the network analysis and memory analysis segments will be collected during the execution of the ransomware on the local system. Figure 3.2 gives a simplified overview of the local sandbox environment used.

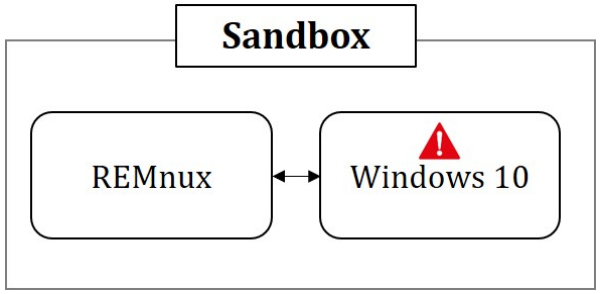


Figure 3.2: The local sandbox environment.

For the network analysis, the REMnux machine is used to generate the network services and capture traffic. The IP of the Windows machine is set to be `192.168.126.128`, with the default gateway as `192.168.126.129`. The last address is the IP of the REMnux machine. INetSim is used to emulate common network services so that the ransomware can interact with the resources it is most likely to need. INetSim simulates the following services (service:port):

- HTTP:80
- SMTP:25
- POP3:110
- FTP:21
- HTTPS:443
- SMTPS:465
- POP3S:995
- FTPS:990

Additionally, fakedns is used to respond to DNS queries and resolve domain names. Wireshark is used to listen to the network traffic from the infected machine. To circumvent potential sleep-mechanisms, the plan is to monitor the network for twenty-four hours post-infection.

Volatility3 will be used to analyse the memory dumps. The baseline commands used will be those detailed in Listing 6. Other commands may be used to pivot on uncovered information.

```
> python vol.py -f C:\Users\volatility3\memory.raw windows.pstree.PsTree
> python vol.py -f C:\Users\volatility3\memory.raw windows.cmdline.CmdLine
> python vol.py -f C:\Users\volatility3\memory.raw windows.netscan.NetScan
> python vol.py -f C:\Users\volatility3\memory.raw windows.mutantscan.MutantScan
> python vol.py -f C:\Users\volatility3\memory.raw timeliner.Timeliner
> python vol.py -f C:\Users\volatility3\memory.raw windows.malfind.Malfind
```

Listing 6: Fundamental Volatility3 commands.

The MITRE ATT&CK framework is used in addition to the HMF to present the uncovered TTPs. To ensure a data-driven overview of the TTPs, the ATT&CK matrix generated for each strain by Joe Sandbox Cloud is used as a foundation. The tactics and techniques are plotted in the ATT&CK navigator tool, using version 8.2 of the enterprise layer. The corresponding procedures are explained and linked to the findings in the text.

3.1 Tools used to Analyse Ragnar Locker

Despite that the methodology suggested in the HMF mention different tools in some sections, some of the tools used when analysing Ragnar Locker may deviate from those. An exhaustive list of the tools used in the analyses is provided in Table 3.1.

Table 3.1: Tools Used in the Analysis

Tool	Version	Use
HashMyFiles	v2.25	Static Analysis
BinText	3.0.3	Static Analysis
PeStudio	8.74	Static Analysis
DetectItEasy	1.01	Static Analysis
Exeinfo PE	0.0.4.9	Static Analysis
CFF Explorer	VIII	Static Analysis
binvis	www.binvis.io	Static Analysis
x32dbg	Mar 4 2018	Static Analysis
IDA Freeware	7.0.180201	Static Analysis
Process Hacker	2.39	Dynamic Analysis
RegShot	1.9.0	Dynamic Analysis
Process Monitor	3.50	Dynamic Analysis
VirusTotal	www.virustotal.com	Dynamic Analysis
AnyRun	www.any.run	Dynamic Analysis
Joe Sandbox Cloud	www.joesandbox.com	Dynamic Analysis
Wireshark	3.4.4	Network Analysis
INetSim	1.3.2	Network Analysis
Fakedns	www.github.com/SocialExploits/fakedns	Network Analysis
DumpIt	1.3.2. 20110401	Memory Analysis
Volatility	3	Memory Analysis
Python	3.9.4	Memory Analysis
MITRE ATT&CK Navigator	v8.2	Presentation

3.2 Risks and Ethical Considerations

The dynamic analyses present some risk, as one is dealing with live ransomware. Using a correctly set up and air-gapped lab when conducting the analysis will mitigate the hazard. Obtaining and dealing with malicious code demands solid security mechanisms and ethical considerations. Under no circumstances should live ransomware be set free nor used for anything other than scientific research. The researcher is obligated to notify relevant instances if live ransomware leaves its confined space.

The strains are most likely created for particular organisations. The data of the victim organisations may be visible on the Ragnar Locker DLS and display sensitive information. This data will not be republished or shared in the thesis, despite the DLS being classified as open-source. However, the victim organisation name and validation of the breach is considered exempt from this. The reason for this is that the company name is likely to be hardcoded into the strain. It is also assumed that the victim organisations are done with the crucial phases of incident management and have notified relevant instances of the breach. The address to DLS sites will not be shared. This decision is based on the wish to limit the traffic to such sites as they often expose innocent individual's information. IPs and other URLs found during the analysis will be "fanged", meaning that parts of the address will be within square brackets. The reason for this is to hinder accidental access to potentially malicious sites.

4

Results

The strain does not appear to be masquerading as legitimate software, nor claim to be anything other than Ragnar Locker. As it does not hide or attempt to lure a victim into executing it, this may indicate that the operators manually deploy Ragnar Locker on the compromised system. According to the DLS, the operators view themselves as penetration testers and bug-hunters. The group claims to search for weaknesses in corporate networks and are willing to provide penetration reports to victims who pay the ransom. They also offer to help fix the issues. Such a mindset corresponds with the emerging trend of HumOR and a wholesome ransomware business.

4.1 File Fingerprinting

To verify the cryptographic hash value of the files, the program HashMyFiles was used. Both strains had their SHA-256 hash as their filename. The generated hashes match the values provided by the download sources. Table 4.1 and Table 4.2 list the verified hash values of the Ragnar Locker strains.

Table 4.1: The hash values of Ragnar Locker strain 1

Strain 1	
Filename	9bdd7f965d1c67396afb0a84c78b4d12118ff377db7efdca4a1340933120f376 (.exe)
MD5	6171000983cf3896d167e0d8aa9b94ba
SHA1	b155264bbfbad7226b5eb3be2ab38c3ecd9f3e18
SHA-256	9bdd7f965d1c67396afb0a84c78b4d12118ff377db7efdca4a1340933120f376

Table 4.2: The hash values of Ragnar Locker strain 2

Strain 2	
Filename	dd79b2abc21e766fe3076038482ded43e5069a1af9e0ad29e06dce387bfae900 (.exe)
MD5	1195d0d18be9362fb8dd9e1738404c9d
SHA1	ef02cc3dbdff915c9149aef86911b4e780b89ae6
SHA-256	dd79b2abc21e766fe3076038482ded43e5069a1af9e0ad29e06dce387bfae900

4.2 Static Analysis

The static analysis was performed using single-purpose tools to determine the hash, PE information, entropy, and strings. These are listed in Section 3.1. The x32dbg debugger and the IDA Freeware disassembler were used to analyse the strain's structure and functions at the assembly level.

4.2.1 PE Structure

According to the file header, strain 1 was compiled at 16:36:20 Friday 31. January 2020. The second strain was compiled at 10:31:44 Monday 31. August 2020. In both instances, the creators have deleted the debugging path, implying a focus on operational security. If present, the debugging path can reveal where the program is located on the creator's machine. User account names and directory names can be valuable clues when trying to create an overview of the attacker's intentions and related breaches. Neither of the strains contains any version number or indicator of version control, exempt for the strains being modified to each victim.

The executables are 32-bit PE files, indicating that they are created for 32-bit processors. Although 64-bit processors are more common for the Windows 10

OS, using the 32-bit architecture ensures better backwards compability. 32-bit programs also use fewer memory resources, making them suitable for computers with 4GB of RAM or less. The use of 32-bit indicates that the developers designed Ragnar Locker to infect a wide range of OS versions. The strains are compressed when at rest and unpacks themself in memory when executed. Profile-guided Optimisation (PGO), a C++ compiler from Intel, is likely the compile and compression method used. PGO compresses files in three steps and is suitable for processor-intensive applications with few variations in applied data sets. The header information in the strains suggests that Ragnar Locker is written in C++. In C++, the source code is first compiled into object files before being linked together into one executable file (Pomeranz, 2018). The linker used seems to be Microsoft Visual C++ (MSVC) version 14.16. MSVC is easily accessible on Windows machines. Pivotal information from the PE headers is listed in Table 4.3.

Table 4.3: The PE information of the analysed Ragnar Locker strains

	Strain 1	**Strain 2**
Compilation date	Friday 31. January 2020 at 16:36:20	Monday 31. August 2020 at 10:31:44
Collected date	Monday 10. February 2020 at 14:34:03	Unknown.
Compiler	Unknown. Written in C++ in Visual Studio Code 15.	Unknown. Written in C++ in Visual Studio Code 15.
Format	PGO	PGO
File type	32-bit Portable Executable	32-bit Portable Executable
Entropy, compiled	6.66313	6.286
Size, compiled	39.50 KB	55.00 KB
Size, decompiled	60.00 KB	72.00 KB
Address-space-layout-randomisation	Yes	Yes
Terminal-server-aware	Yes	Yes
Contains self-modifying sections	Yes	Yes
Library imports	crypt32.dll kernel32.dll user32.dll advapi32.dl shell32.dll shlwapi.dll	crypt32.dll kernel32.dll user32.dll advapi32.dl shell32.dll shlwapi.dll
Number of DLLs imported	87	110
Manifest MD5	1E4A89B11EAE0FCF8BB5FDD5EC3B6F61	1E4A89B11EAE0FCF8BB5FDD5EC3B6F61

4.2.2 Windows API Class

The Ragnar Locker strains imports six native Windows libraries, as seen in Table 4.4. The Windows application programming interfaces (APIs) are standard

for Windows machines. Windows APIs are used to invoke lower-level functions. Relying on these API Classes help Ragnar Locker blend with the environment and reduce the need to import additional tools and libraries. Strain 2 imports more functions from these libraries, which indicates that the developers have added extra functionality in the newer strain.

Table 4.4: DLL libraries used by Ragnar Locker

Library	DLL Imports, Strain 1	DLL Imports, Strain 2	Description
crypt32.dll	4	4	Crypto API32. Implements Certificate and Cryptographic functions.
kernel32.dll	57	72	Windows NT BASE API Client DLL. Handles memory management, I/O operations, and interrupts.
user32.dll	2	2	Multi-User Windows USER API Client DLL. Handles the different user preferences.
advapi32.dl	20	26	Advanced Windows 32 Base API. Handles restarts and shutting down the system, the Windows registry, user accounts and Windows services.
shell32.dll	1	3	Windows Shell Common DLL. Is used when opening web pages and files.
shlwapi.dll	3	3	Shell Light-Weight Utility Library. Contains functions for URL paths, Windows registry, and colour settings.

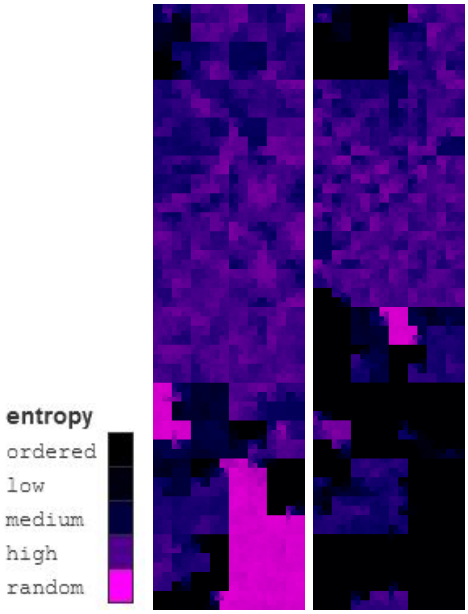
4.2.3 Entropy

Entropy can be described as the amount of information, and its randomness, contained within a variable (Vajapeyam, 2014). Entropy analysis can help determine whether an executable file is compressed or obfuscated. When analysing digital files, the entropy is measured by calculating the randomness of bytes. The result is measured on a scale from 0 to 8. As a byte consist of 8 bits, the maximum entropy of a file is 8. Repetitive and conform data will have a low entropy, where 0 is the lowest and signifies completely uniform data. Ordinary files usually have an entropy of around 5. By examining the Ragnar Locker strain using PeStudio and DetectItEasy, it becomes evident that parts of the strain are obfuscated or compressed. The `.text`-sections have an entropy above 6. The same is true for `.keys` and `.didata`, respectively. The `.text`-section contains the program code, and the high entropy indicates that the code is packed. As for the `.keys`-section and `.didata`-section, it is expected that the entropy is high, as they contain the cryptographic keys. The sections and their entropy are listed in Table 4.5.

Table 4.5: Entropy of the different sections of Ragnar Locker

Entropy		
Section	**Stain 1**	**Strain 2**
.text	6.521	6.480
.rdata	5.354	5.242
.data	n/a	0.206
.keys	6.442	Not present
.didata	Not present	4.615
.rsrc	4.702	4.696
.reloc	4.810	6.001

The entropy can also be visualised. There are several ways to do this. For this analysis, two methods were used. Both strains were visualised using binvis.io in their compressed and unpacked state. Binvis.io calculates entropy using Shannon's Entropy Metric, and displays the entropy using a scale from ordered to random. The Hilbert space-filling curve is used for generating the layout of the entropy images, as it is the leading curve in regards to preserving the locality of the ransomware features (Cortesi, 2015). Preserving the locality ensures that related functions in the analysed binary are presented close to each other, which gives an overview of how the program looks- Additionally, the entropy timeline function of DetectItEasy was used to visualise the entropy of the different sections in a graph. DetectItEasy is also based on Shannon's Entropy Metric but displays the entropy using digits. Figure 4.1 exhibits the entropy of strain 1 when compressed at rest and unpacked during the execution of the ransomware as generated by binvis.io.



(a) Strain 1 compressed. (b) Strain 1 unpacked.

Figure 4.1: Strain 1 visualised using binvis.io.

From the graph generated by DetectItEasy it is possible to visualise the entropy of the different sections. When browsed in the tool, the different areas and their boundaries can be selected. Figure 4.2 displays the entropy graph of strain 1, where the `.keys`-section is the cause of the most significant peak.

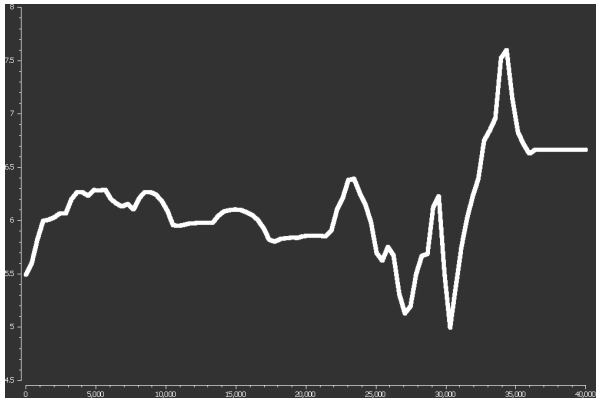
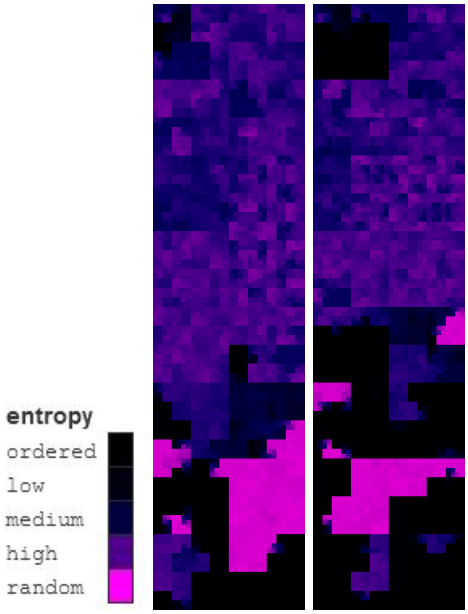


Figure 4.2: Entropy in strain 1, as displayed in DetectItEasy.

Figure 4.3 displays the entropy of strain 2 compressed and unpacked according to binvis.io.



(a) Strain 2 compressed. (b) Strain 2 unpacked.

Figure 4.3: Strain 2 visualised using binvis.io.

For strain 2, the graph generated by DetectItEasy is similar to strain 1 in its general structure. However, it displays an overall higher entropy with more considerable differences between low and high points. The top and bottom points both belong to the `.didata`-section, which appears to be the successor to the `.keys`-section in strain 1. Figure 4.4 displays the entropy graph of strain 2.

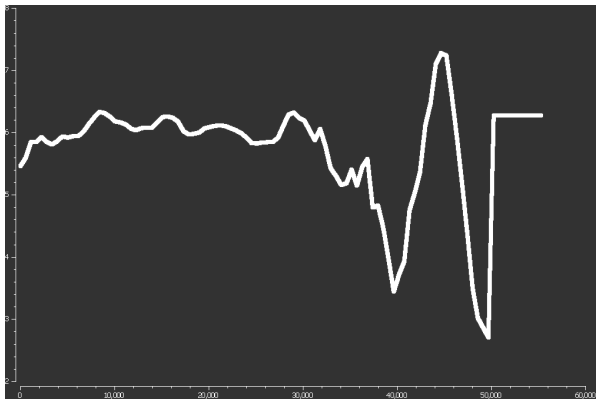


Figure 4.4: Entropy in strain 2, as displayed in DetectItEasy.

4.2.4 Termination Criteria

Strain 1 will abort its execution based on the computer locale. Ragnar Locker uses the `GetLocaleInfoW()` function to determine the OS language settings. The `GetLocaleInfoW()` function use the `LOCALE_SYSTEM_DEFULT` to get the local language strings of the system and compare them to the strings that have been pushed to the memory stack. If it detects that it is in an environment that matches one of the pre-defined languages, it will obtain its own process and terminate itself. The OS language settings that will make Ragnar Locker 1 stop executing are presented in Table 4.6.

Table 4.6: OS languages that will cause strain 1 to terminate

Language settings that will cause termination		
Azerbaijani	Armenian	Belorussian
Kazakh	Kyrgyz	Moldovian
Tajik	Russian	Turkmen
Uzbek	Ukrainian	

Strain 2 contains the `GetLocaleInfoW()` function as well, but lacks the list of pre-defined countries. Instead, the function calls undefined variables. It is possible to use the function with numeric values, but the default setting for the function is to retrieve information in text format. As the rest of the Ragnar Locker code does not apply any form of obfuscation, it is considered unlikely that the threat actors would take extra measures to hide the abortion criteria of the strain. One possible hypothesis is that since the attacks are targeted, there is little need for a fail-safe mechanism. Another possibility is that the broken function is an unintended bug.

4.2.5 Behaviour and File Encryption Process

Both Ragnar Locker strains read the registry keys in Figure 4.5 to collect the ID of the victim computer, the current user, and the OS product name. The function `GetComputerNameW` reads the registry key `\SOFTWARE\Microsoft\Cryptography\MachineGuid` which contain the Globally Unique Identifier (GUID). The GUID is a 128-bit integer, and the MachineGUID is used to uniquely identify a machine (Microsoft Docs, n.d.). The `GetUserNameW`

function will retrieve the name of the currently logged-in user from the buffer. Lastly, the `\SOFTWARE\Microsoft\WindowsNT\CurrentVersion.ProductName` registry key is read to obtain the OS product name.

```
.text:000241C0              call    ds:GetComputerNameW
.text:000241C6              lea     eax, [ebp-6Ch]
.text:000241C9              push    eax             ; pcbBuffer
.text:000241CA              lea     eax, [ebp-0D40h]
.text:000241D0              push    eax             ; lpBuffer
.text:000241D1              call    ds:GetUserNameW
.text:000241D7              push    offset SubKey   ; "SOFTWARE\\Microsoft\\Cryptography"
.text:000241DC              lea     eax, [ebp-1050h]
.text:000241E2              push    eax             ; lpString1
.text:000241E3              call    ds:lstrcpyW
.text:000241E9              push    offset ValueName ; "MachineGuid"
.text:000241EE              push    offset SubKey   ; "SOFTWARE\\Microsoft\\Cryptography"
.text:000241F3              call    sub_234D0
.text:000241F8              push    offset aProductname ; "ProductName"
.text:000241FD              push    offset aSoftwareMicros_0 ; "SOFTWARE\\Microsoft\\Windows NT\\Curren"...
```

Figure 4.5: Ragnar Locker obtaining the name of the computer and user.

The collected values are concatenated into one string, using the format `%s-%s-%s-%s-%s` where `%s` represents a single value. The formatted string is then *GUID-OS name-username-machine name*.

The Ragnar Locker strains will iterate through the volumes and drives on the computer. Logical volumes are also mapped and assigned a drive letter, indicating that shared access resources will also be encrypted. To do this, Ragnar Locker use `CreateFileW` combined with `\\.\PHYSICALDRIVE%d`. The enumeration of the drives follows the flow visualised in Figure 4.6.

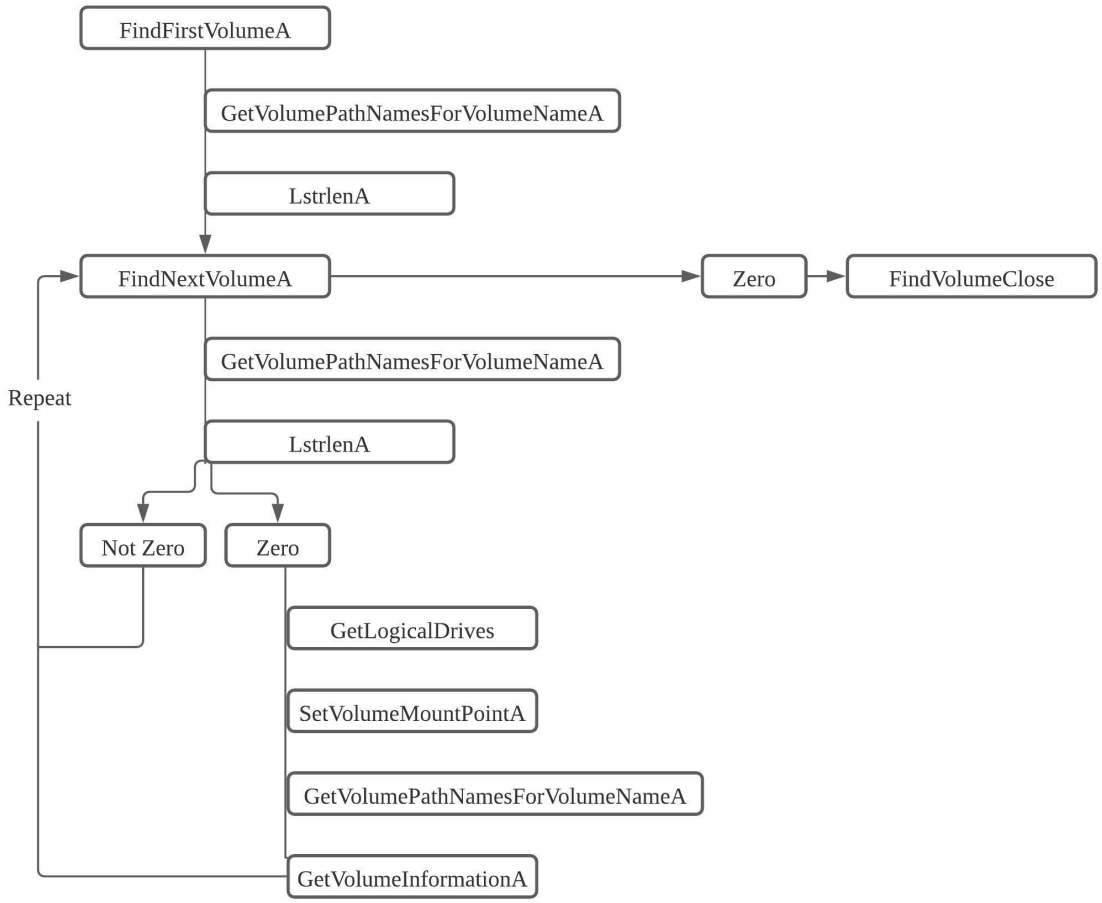


Figure 4.6: Volume and drive mapping process of Ragnar Locker.

When a volume is discovered, Ragnar Locker retrieves the pathname and the length of the name. The strains will skip volumes where the path is three characters or less in length. When the first volume is mapped, the strains continue to enumerate physical and logical volumes but adds a check to see if the volume has already been mapped.

The files that are to be encrypted is added to the memory stack. The `.keys` and `.didata`-sections are called several times to decrypt the strings that reference the files exempt encryption. Folders, files, and extensions refrained from encryption is listed in Table 4.7.

Table 4.7: Folders exempt encryption

Folders exempt encryption	
Strain 1	**Strain 2**
Windows	Windows
Windows.old	Windows.old
Tor browser	Tor browser
Internet Explorer	Internet Explorer
Google	Google
Opera	Opera
Opera Software	Opera Software
Mozilla	Mozilla
Mozilla Firefox	Mozilla Firefox
§Recycle.Bin	§Recycle.Bin
ProgramData	ProgramData
	All Users
	Sysvol

Table 4.9: File extensions exempt encryption

Extensions exempt encryption	
Strain 1	**Strain 2**
.db	.db
.lnk	.lnk
.sys	.sys
.msi	.msi
.dll	.dll
.drv	.drv
	.exe

Ragnar Locker will also utilise strings to search for certain running processes. Processes that match these strings will be terminated. This behaviour is likely a security mechanism to avoid detection and termination, as the strings are related

Table 4.8: Files exempt encryption

Files exempt encryption	
Strain 1	**Strain 2**
autorun.inf	autorun.inf
boot.ini	boot.ini
bootfont.bin	bootfont.bin
bootsect.bak	bootsect.bak
bootmgr	bootmgr
bootmgr.efi	bootmgr.efi
desktop.ini	desktop.ini
iconcache.db	iconcache.db
ntldr	ntldr
ntuser.dat	ntuser.dat
ntuser.dat.log	ntuser.dat.log
ntuser.ini	ntuser.ini
	thumbs.db
	!!!_READ ME_ID_!!!.txt

to several known MSPs, anti-virus (AV), and security solution providers. The strings called are listed in Table 4.10.

Table 4.10: Processes terminated by Ragnar Locker

Process termination strings	
Strain 1	**Strain 2**
vss	vss
sql	sql
memtas	memtas
mepocs	mepocs
sophos	sophos
veeam	veeam
backup	backup
pulseway	pulseway
logme	logme
logmein	logmein
connectwise	connectwise
splashtop	splashtop
kaseya	wu auserv

Once Ragnar Locker finishes enumerating the OS and adds file references to the stack, the encryption process will begin. The strains have two hardcoded keys, one for symmetric encryption and one for asymmetric encryption. The private key used in the asymmetric encryption is never present in the system. The organisation of the keys is similar in both strains, the main difference being slightly different naming conventions. The symmetric key is referred to as `RAGNAR SECRET` in strain 1, and `RAGN KEY` in strain 2. The asymmetric key is named `PUBLIC KEY` regardless of strain. The keys are encoded using Base64. Listing 7 show how the symmetric key for strain 2 is presented in the unpacked assembly code. Ragnar Locker uses the Salsa20 stream cypher for symmetric encryption.

```
"**************************************************************************"
"---BEGIN RAGN KEY---"
"NjRlQ0M5M0ZBQkEzQUFjRThEZjBEMWQ0RUJEOUZiMDRkZUU0NGVDMTExYzQwMUQyYmNjRUEyM
zdhNTQ1NDE5Mw=="
"---END RAGN KEY---"
"**************************************************************************"
```

Listing 7: The symmetric key as displayed in the assembly code of strain 2.

The data is first encrypted using Salsa20, then the Salsa20 key is encrypted using the RSA algorithm. Figure 4.7 depicts the assembly portion where the `CryptAcquireContextW` function is called. The `dwProvType` is set to 1, referring to `PROV_RSA_FULL` which is the Cryptographic Provider Type for RSA public key algorithms (Microsoft Docs, 2020).

```
.text:00022A90    push    0F0000040h      ; dwFlags
.text:00022A95    push    1               ; dwProvType
.text:00022A97    push    0               ; szProvider
.text:00022A99    push    0               ; szContainer
.text:00022A9B    push    ebx             ; phProv
.text:00022A9C    call    ds:CryptAcquireContextW
.text:00022AA2    test    eax, eax
.text:00022AA4    jz      loc_22BD2
```

Figure 4.7: The `CryptAcquireContextW` function in strain 2.

The Base64 encoded public RSA key is displayed in assembly as follows in Listing 8. When the data is encrypted, Ragnar Locker calls the `CryptDestroyKey` function to release the handle for the key, so it cannot be used again.

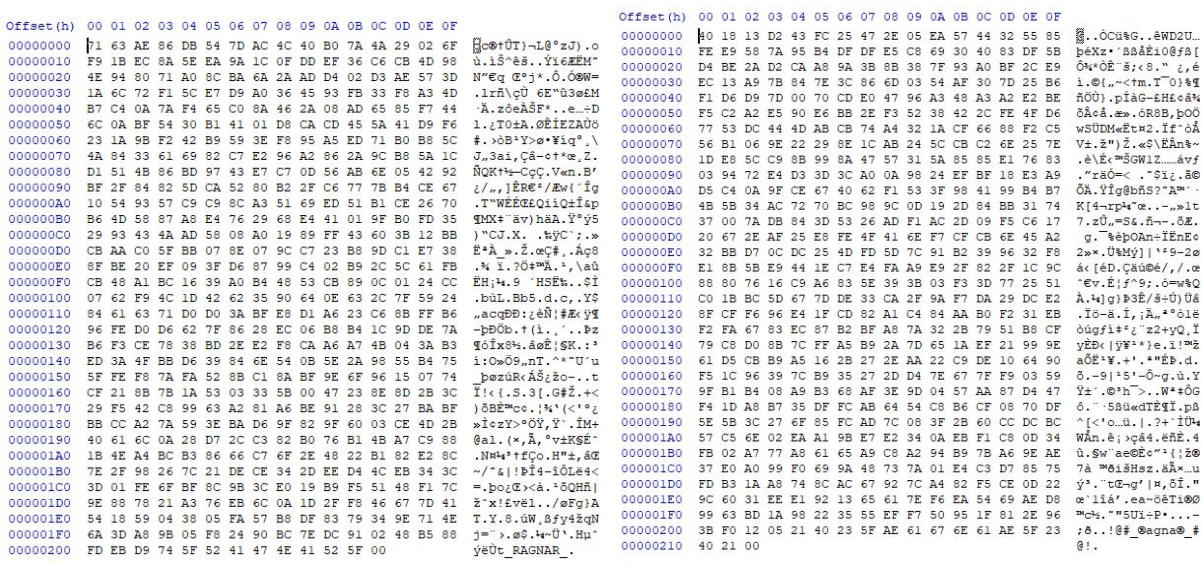
```
"-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA6
sbq10jIwo0QT80CSAil\nl/YpWWrZJ5noHg7iAmqYVPRiLpPUfJ1AF++78CPQzzjh7LJm2Q646
JPhdijwU2u8\nsujjf1ZwWgdYuigMzqgNA/g/TKhvl1X4YT0o+px1YxAG5rWIJjMtimTxoLF8N
WI0\nAlekPpLgV+piZTgRTVy7JY3NzTSK1VKVgZnzoZBYqNahx8Fqi0F98zs+tozaCsBV\noVi
8WIHrMi4kP+Fuvpr6VadocsoPYHwYhrmGQEiXlm/qZ+VJ4d15WR4lo2CZ3eTM\nOAw6TKjtqGW
UVoAWxzfbol6EdoBQbyilvgKhw8C/etcz2IkzDgPVykh/EYzhgVi\ntwIDAQAB\n
-----END PUBLIC KEY-----\n"
```

Listing 8: The asymmetric key as displayed in the assembly code of strain 2.

The encryption process is the same for both strains. Ragnar Locker encrypts program files and similar data before moving on to user data. A ransom note is generated in all directories except for the ignored folders. Strain 1 appends the extension `.ragnar_<ID>` to the encrypted files, while strain 2 use `.__r4gN4r_`

_<ID>. The ID is a hash of the computers NETBIOS name (FBI, 2020). Once the encryption process is finished, all running processes are terminated, and a notepad window opens to display the ransom note.

As part of the encryption, a new footer is added to the affected files. For strain 1, the new file footer is _RAGNAR_. Strain 2 have further obfuscated the new footer, resulting in !@#_agna_#@!. Moreover, due to the encryption, the file size is considerably larger. The file affected by strain 1 grew with 928.57%. Stain 2 provided a 964.29% increase in file size. Figure 4.8 compares the hexadecimal and ANSI of a text file encrypted by the two strains.



(a) Text file encrypted with strain 1.

(b) Text file encrypted with strain 2.

Figure 4.8: A sample file encrypted by Ragnar Locker.

4.2.6 Code Related to the Virtual Machine Technique

Strain 2 have a segment of code that is not found in strain 1. A snippet of this assembly code is displayed in Figure 4.9. The -vm, -vmback, and -share_network are command-line options used by the ransomware executable when it is deployed in a virtual machine (Loman, 2020).

```
.text:00024CC0                push    offset aVm      ; "-vm"
.text:00024CC5                push    dword ptr [edi+4] ; lpString1
.text:00024CC8                call    ds:lstrcmpiW
.text:00024CCE                test    eax, eax
.text:00024CD0                mov     eax, 1
.text:00024CD5                cmovz   esi, eax
.text:00024CD8                cmp     [ebp-8], eax
.text:00024CDB                jle     short loc_24D15
.text:00024CDD                push    offset aVmback  ; "-vmback"
.text:00024CE2                push    dword ptr [edi+4] ; lpString1
.text:00024CE5                call    ds:lstrcmpiW
.text:00024CEB                test    eax, eax
.text:00024CED                jnz     short loc_24CF7
.text:00024CEF                mov     esi, 1
.text:00024CF4                mov     [ebp-4], esi
.text:00024CF7
.text:00024CF7 loc_24CF7:                             ; CODE XREF: start+B6D↑j
.text:00024CF7                cmp     dword ptr [ebp-8], 1
.text:00024CFB                jle     short loc_24D15
.text:00024CFD                push    offset aShareNetwork ; "-share_network"
.text:00024D02                push    dword ptr [edi+4] ; lpString1
.text:00024D05                call    ds:lstrcmpiW
.text:00024D0B                test    eax, eax
.text:00024D0D                mov     ecx, 1Fh
.text:00024D12                cmovz   esi, ecx
```

Figure 4.9: Assembly code related to the virtual machine used to deploy Ragnar Locker.

The presence of this functionality may indicate that the technique of using an Oracle VirtualBox hypervisor to deploy the ransomware and bypass security mechanisms was used in the attack on CMA CGM.

4.2.7 The Ransom Note

The ransom note is hardcoded in both strains. The first strain appears to be tailor-made for American PSE Credit Union. PSE Credit Union is not mentioned on the DLS, nor have the company made a public statement or implied that it has suffered a ransomware attack. As PSE Credit Union were hit in January 20, a potential leak most likely happened before June 2020. If data from PSE Credit Union was leaked, it was removed the redesign of the DLS. Strain 2 is intended for the French container line company Compagnie Maritime d'Affrètement, abbreviated CMA CGM. CMA CGM has confirmed that they suffered a ransomware attack by Ragnar Locker on September 28, 2020 (Shen & Baker, 2020). Data from the container line company has not been leaked on the DLS per March 2021.

The ransom note has changed significantly from strain 1 to strain 2. Figure 4.13 displays the demand delivered by strain 1, in a note named `RGNR_<ID>.txt`. The full ransom note can be found in Appendix A.


RGNR_08B528AC.txt - Notepad

File Edit Format View Help

Hello PSE_CREDIT_UNION !

****

If you reading this message, then your network was PENETRATED and all of your files and data has been ENCRYPTED

by RAGNAR_LOCKER !

****

********What happens with your system ?************

Your network was penetrated, all your files and backups was locked! So from now there is NO ONE CAN HELP YOU to get your files back, EXCEPT US.
You can google it, there is no CHANCES to decrypt data without our SECRET KEY.

But don't worry ! Your files are NOT DAMAGED or LOST, they are just MODIFIED. You can get it BACK as soon as you PAY.
We are looking only for MONEY, so there is no interest for us to steel or delete your information, it's just a BUSINESS $-)

HOWEVER you can damage your DATA by yourself if you try to DECRYPT by any other software, without OUR SPECIFIC ENCRYPTION KEY !!!

Also, all of your sensitive and private information were gathered and if you decide NOT to pay,
we will upload it for public view !

****

**********How to get back your files ?******

To decrypt all your files and data you have to pay for the encryption KEY :

BTC wallet for payment: 1E6EjTqYPHLj1uovPKKRXzMpPCcpAcVuiU
Amount to pay (in Bitcoin): 60

****

Figure 4.10: The `RGNR_<ID>` note in strain 1.

The ransom note belonging to strain 1 instructs the victim to pay 60 bitcoins (BTC) to a bitcoin wallet. Onwards it tells the victim company to install the messaging application qTOX and add a specific user. qTOX offers peer-to-peer encryption. The victim is asked to identify themselves by providing the `RAGNAR SECRET` key appended in the ransom note.

The ransom note in strain 2 is fundamentally different. The file name syntax of it is `!!!_README_<ID>_!!!.txt`. Much of the information is similar to that in strain 1, such as warning the victim not to tamper with the files, urging the company to make contact within two days, and the `RAGN KEY`. Part of the second ransom note is presented in Figure 4.13, and the full ransom note can be found in Appendix B.

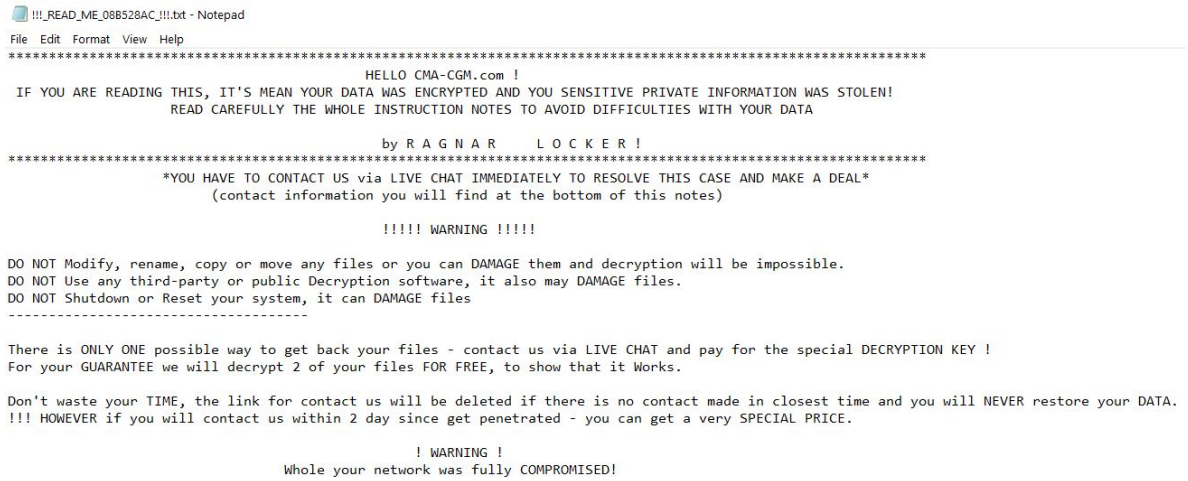


Figure 4.11: The `!!!_README_<ID>_!!!` note in strain 2.

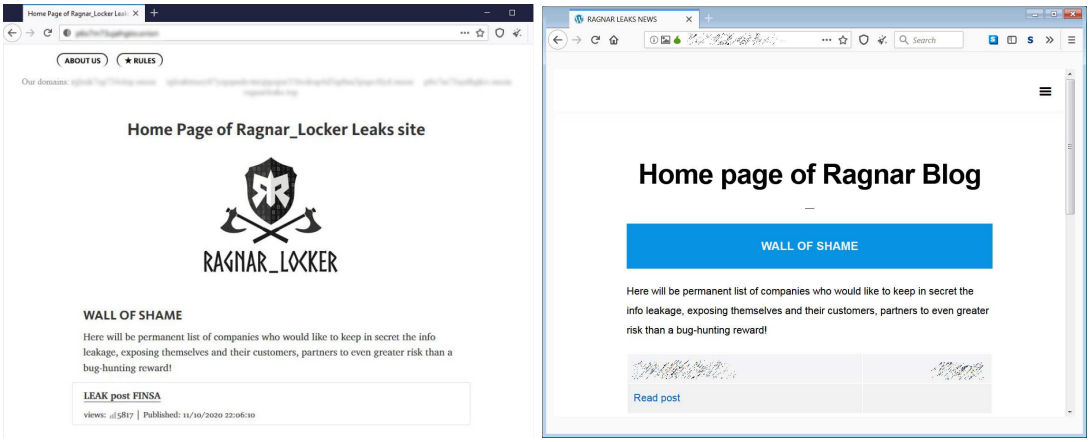
The first difference is that the note does not list the price for decrypting the data. It simply states that the victim company should contact the threat actors within two days if they want a special price. The author of the note is also clear that victims should not waste the groups time, indicating that the operators are tired of cheap negotiators. Continuing, the note list types of data the attackers presumably have extracted, empathising private and sensitive documents. To demonstrate that they possess the victim's files, links to print screens from the breached system are added as additional proof. The free and open *LightShot screen capture tool* is used for this purpose. A password-protected onion site with the same function is also listed.

Instead of asking the victim to download qTOX, strain 2 directs the victim to download the Tor browser and visit a password restricted page on their DLS. There, the attackers have set up a live chat service on which the victims should contact them.

There are two key takeaways from the changes in the ransom note. The first is the increased focus on operational security by removing the BTC wallet address from the note. Law enforcement agencies are becoming increasingly skilled at tracking BTC transactions, so removing the wallet's address from public view is likely a security consideration. The second takeaway is the business maturity of the Ragnar Locker group. In addition to developing their tools, the threat actors have taken the time to create a proof of concept pages and chat solutions on their own domain. The improvements make Ragnar Locker appear more trustworthy and professional. For ransomware distributors, it is eminent that the victim

trusts them enough to pay for the decryption key. If the group comes off as dishonest and immature, they are less likely to monetise on their efforts.

Figure 4.12 displays the evolution of the design for the DLS. Based on the site's metrics, the new design was implemented in early June 2020.



(a) "Wall of Shame", November 2020.

(b) "Wall of Shame", January 2020 (Abrams, 2020a).

Figure 4.12: Victim extortion.

4.3 Dynamic Analysis

The strains have been run in both a local virtual environment and commercial sandboxes to observe how they behave when executed. The sandboxes used is VirusTotal, JoeSandbox Cloud, and AnyRun. The findings from the static analyses were validated when the samples were executed locally. One function offered by VirusTotal is to check the detection rate of a strain against a large base of AV software. 64 out of 71 AV engines flagged strain 1 as a malicious file the 31.01.2021, giving it a 90.14% detection rate one year after its creation. In contrast, 37 AV engines reacted on strain 2, indicating a 52.11% detection rate three months after it was discovered.

4.3.1 Initiated processes

Strain 1 initiates three processes: `vssadmin.exe`, `wmic.exe`, and `notepad.exe`. Figure 4.13 presents a hierarchal overview of the behaviour. The processes are started to perform actions on the system that Ragnar Locker cannot conduct independently.

Figure 4.13: Processes initiated by Ragnar Locker strain 1.

`Vssadmin.exe` is the main Windows function that administers the volume shadow copies. In `vssadmin.exe`, the command `vssadmin delete shadows /all /quiet` is run. This command deletes all shadow copies on systems running Windows 8.1 and Windows Server 2008 and newer. The volume shadow copies are deleted to prevent a user from recovering information by rolling the system back to a known good version. Ragnar Locker launches `wmic.exe` in addition to `vssadmin.exe`. `Wmic.exe` is a command-line utility used to access the Windows Management Instrumentation and can only be used by the local system administrator. When the strain has obtained access to the command-line, it runs the command `shadowcopy delete`. As some systems have mechanisms to secure the shadow copies, this double function is likely a redundancy mechanism for Ragnar Locker. When initiating the same process using two different methods, the threat actors increase their chance to succeed. Notepad.exe is initiated after the system is encrypted and is used to display the ransom note.

Strain 2 commence four processes: `wmic.exe`, and three instances of `bcdedit.exe`. They are all executed with the admin user account. Figure 4.14 depicts the processes spawned by strain 2.

The `wmic.exe` function is the same as in strain 1. Similarly to the previously discussed tools, `bcdedit` is a default Windows command-line tool. It is used for managing the Boot Configuration Data (BCD) (Microsoft Docs, 2017). Each instance of `bcdedit` run one command. The first command is `bcdedit /set {globalsettings} advanced options false`, which disables the Advanced Startup Options and hinders the startup boot menu from auto-loading. Disabling the startup boot menu may make it harder for the victim to boot computers into

Figure 4.14: Processes initiated by Ragnar Locker strain 2.

safe mode. However, the Windows default is not to have the boot menu show at every start up, so the impact of this action depends on the victim system configuration. The second command `bcdedit /set {default} bootstatuspolicy IgnoreAllFailures` changes the boot policy to ignore errors if there is a failed boot, failed shutdown, or failed checkpoint. The computer will boot normally once it can. The third parameter set is the `bcdedit /set {default} recoveryenabled No`, which disables the Automatic Startup Repair. Automatic Startup Repair is a built-in feature that is supposed to fix boot-related issues. The `bcdedit` operations are likely added to make it harder for victims to boot systems into safe mode once they detect that something is amiss in their systems and to disable recovery.

4.3.2 YARA Rules

The YARA rule to detect the two analysed strains is relatively simple. Where fitting, the strings to search for are provided in hexadecimal. Using the hexadecimal value allows for wildcards, which in YARA is `??`, and removes the need to convert the string to ASCII. Although YARA supports regular expressions, these have been avoided to keep the rule fast and compatible with large networks and rulesets (Roth, 2015). The classification of the conditions is based on the following naming conventions:

$x = Very specific strings. One of these is enough to detect the threat.
$a = Specific string. The string appears like a malicious indicator but may occur in benign programs as well.
$s = Other strings. Miscellaneous strings found in the ransomware samples.

```
rule Ragnar_Locker
{
 meta:
    author = "Marthe Brendefur"
    date = "04.04.2021"
    description = "Basic YARA-rule for the Ragnar Locker ransomware family"
    hash1 = "9bdd7f965d1c67396afb0a84c78b4d12118ff377db7efdca4a1340933120f376"
    hash2 = "dd79b2abc21e766fe3076038482ded43e5069a1af9e0ad29e06dce387bfae900"

 strings:
    $x1 = { 52 41 47 4e 20 4b 45 59 } // RAGN KEY
    $x2 = { 52 41 47 4e 41 52 20 53 45 43 52 45 54 } // RAGNAR SECRET
    $a1 = "1E4A89B11EAE0FCF8BB5FDD5EC3B6F61" // MD5 of Manifest
    $s1 = { 62 6f 6f 74 ?? ?? ?? 74 2e 62 ?? ?? } // bootfont and bootsect
    $s2 = { 62 6f 6f 74 6d 67 72 2e 65 66 69 } // bootmgr.efi
    $s3 = { 53 44 3b 53 44 77 } // SD;SDw
    $s4 = { 4b 3c 5e 5f 5b 5d } // K<^_[]
    $s5 = "PHYSICALDRIVE%d" fullword wide
    $s6 = { 25 73 2d 25 73 2d 25 73 2d 25 73 2d 25 73 } // %s-%s-%s-%s-%s

 condition:
    filesize > 35000 and filesize < 75000 and
    (
        ( 1 of ($x*) ) or
        ( 1 of ($a*) and all of ($s*) )
    )
}
```

Listing 9: YARA rule for detecting Ragnar Locker.

The YARA rule was first tested on the two strains analysed in this paper, including a decompressed version of each of them. Figure 4.15 depicts the result, showing that the rules detected all instances of the strains.

```
PS C:\Users\REM\Downloads\YARA> .\yara64.exe C:\Users\REM\Downloads\YARA\ragnar.yar 'C:\Users\REM\Desktop\RagnarLocker'
Ragnar_Locker C:\Users\REM\Desktop\RagnarLocker\9bdd7f965d1c67396afb0a84c78b4d12118ff377db7efdca4a1340933120f376
Ragnar_Locker C:\Users\REM\Desktop\RagnarLocker\9bdd7f965d1c67396afb0a84c78b4d12118ff377db7efdca4a1340933120f376_dump
Ragnar_Locker C:\Users\REM\Desktop\RagnarLocker\dd79b2abc21e766fe3076038482ded43e5069a1af9e0ad29e06dce387bfae900
Ragnar_Locker C:\Users\REM\Desktop\RagnarLocker\dd79b2abc21e766fe3076038482ded43e5069a1af9e0ad29e06dce387bfae900_dump
```

Figure 4.15: The YARA rule run against a folder containing the analysed strains.

Another essential aspect of testing is that the YARA rule does not produce many false positives. Ideally, a YARA rule should be specific enough to detect the

threat it is supposed to catch. To validate that the YARA rule primarily identify strains from the Ragnar Locker family, and preferably more than the two analysed strains, the rule was tested on a folder containing different malware. Figure 4.16 shows the content of the second folder. In addition to the analysed strains, it contains two other strains of Ragnar Locker and three strains of unrelated malware. The analysed strains are highlighted in the dark blue boxes, whilst the additional strains of Ragnar Locker are accentuated in light blue boxes. The other samples are miscellaneous malware, mainly trojans and crypto miners.

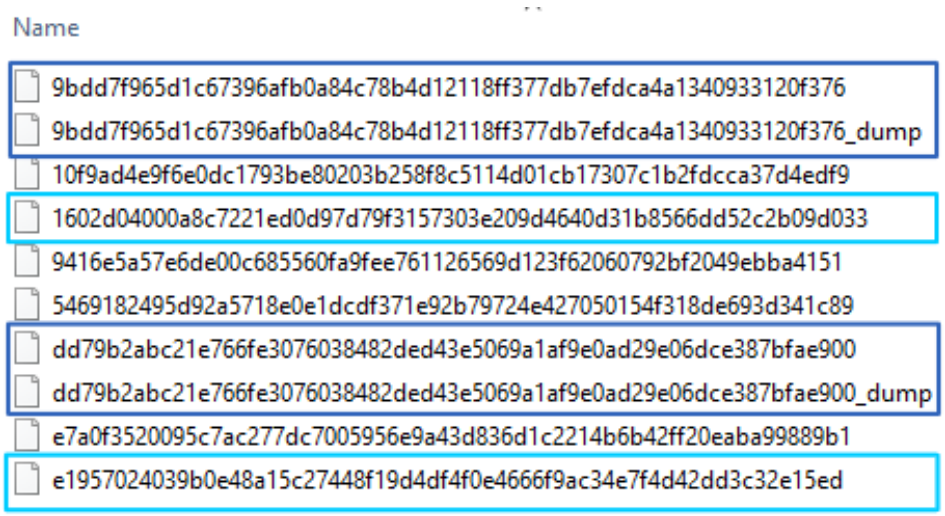


Figure 4.16: Folder containing various malware samples.

When the YARA rule was run against the folder, it only detected the Ragnar Locker strains, as can be seen in Figure 4.17.

```
PS C:\Users\REM\Downloads\YARA>  .\yara64.exe C:\Users\REM\Downloads\YARA\ragnar.yar 'C:\Users\REM\Desktop\RagnarLocker'
Ragnar_Locker C:\Users\REM\Desktop\RagnarLocker\1602d04000a8c7221ed0d97d79f3157303e209d4640d31b8566dd52c2b09d033
Ragnar_Locker C:\Users\REM\Desktop\RagnarLocker\9bdd7f965d1c67396afb0a84c78b4d12118ff377db7efdca4a1340933120f376
Ragnar_Locker C:\Users\REM\Desktop\RagnarLocker\9bdd7f965d1c67396afb0a84c78b4d12118ff377db7efdca4a1340933120f376_dump
Ragnar_Locker C:\Users\REM\Desktop\RagnarLocker\dd79b2abc21e766fe3076038482ded43e5069a1af9e0ad29e06dce387bfae900
Ragnar_Locker C:\Users\REM\Desktop\RagnarLocker\dd79b2abc21e766fe3076038482ded43e5069a1af9e0ad29e06dce387bfae900_dump
Ragnar_Locker C:\Users\REM\Desktop\RagnarLocker\e1957024039b0e48a15c27448f19d4df4f0e4666f9ac34e7f4d42dd3c32e15ed
```

Figure 4.17: The YARA rule run against a folder containing different malware.

4.3.3 Indicators of Compromise

The uncovered IOCs are listed in Table 4.11. Although the links to the DLS can be considered as IOCs, especially since one of them is listed in strain 2, these have been excluded from the list due to the ethical considerations discussed in Section 3.2. In total, the conducted research has uncovered four verified links to the Ragnar Locker DLS. As most of the generated artefacts and ransomware

actions are highly host-specific, few of the listed IOCs will work on other Ragnar Locker strains. The only exception may be the hash of the manifest, which is the same in both analysed strains.

Table 4.11: Identified Indicators of Compromise

Type	Indicator	MD5 Hash	Strain
File	E3E8FF39C3359C873A06E0F7E6C394AF	E3E8FF39C3359C873A06E0F7E6C394AF	1
File	9808E52A5928A709A480CEC75FE88326	9808E52A5928A709A480CEC75FE88326	2
File	1E4A89B11EAE0FCF8BB5FDD5EC3B6F61	1E4A89B11EAE0FCF8BB5FDD5EC3B6F61	Both
Imphash	6A3E7314BD4201552084C30FB976959E	6A3E7314BD4201552084C30FB976959E	1
Imphash	2C2AAB89A4CBA444CF2729E2ED61ED4F	2C2AAB89A4CBA444CF2729E2ED61ED4F	2
Strain	6171000983CF3896D167E0D8AA9B94BA	6171000983CF3896D167E0D8AA9B94BA	1
Strain	1195D0D18BE9362FB8DD9E1738404C9D	1195D0D18BE9362FB8DD9E1738404C9D	2
String	7D509C5BB14B1B8CB0A3338EEA9707A D31075868CB9515B17C4C0EC6A0CCCA 750CA81606900D		1
String	!@#_®agna®_#@!		2
String	_RAGNAR_		1

4.4 Network Analysis

The examined strains do not appear to communicate with external entities. The static analyses found that the strains do not contain or call any web-related APIs, nor do they contain any named pipes. Executing Ragnar Locker whilst monitoring the internet activity confirmed that no attempts to connect to the internet was made. The original plan for the network analysis was to monitor each strain for 24 hours post-infection. The time was reduced to 8 hours due to computing resources and a lack of indicators that Ragnar Locker would communicate on the network. Figure 4.18 show some of the domain names resolved by strain 1 while it was monitored.

```
remnux@remnux:~$ fakedns
pyminifakeDNS:: dom.query. 60 IN A 192.168.126.129
Respuesta: settings-win.data.microsoft.com. -> 192.168.126.129
Respuesta: ctldl.windowsupdate.com. -> 192.168.126.129
Respuesta: watson.telemetry.microsoft.com. -> 192.168.126.129
Respuesta: sls.update.microsoft.com. -> 192.168.126.129
Respuesta: ctldl.windowsupdate.com. -> 192.168.126.129
Respuesta: v20.vortex-win.data.microsoft.com. -> 192.168.126.129
```

Figure 4.18: Domain names resolved by strain 1 over a 8 hour period.

The lack of network connectivity indicates that the threat actor's communication mechanisms are not related to, or connected to, the ransomware. The beacons used are likely independent implants on the breached system. The sole purpose of the ransomware is to encrypt the system and inform the victim of the compromise. Analyses from the web-based sandboxes further confirmed this finding.

4.5 Forensic Artefacts

When conducting a digital forensics examination of a system, it is essential to note that each interaction with the system will change it. In forensics, Locard's exchange principle stating that everyone interacting with a crime scene will leave something behind and take something with them, must always be considered (Sammons, 2015). The principle is also true in regards to digital forensics. As Ragnar Locker is a ransomware, it will produce many artefacts, both in the registry and the file system. The number of artefacts will depend on the system. The reason for this is that it checks and affect most files on a system.

4.5.1 Registry Analysis

Strain 1 of Ragnar Locker adds 17 keys to the Windows registry. It adds the `HKLM\SOFTWARE\Microsoft\IdentityCRL\ThrottleCache`-key and appends five accounts to it. The ThrottleCashe is used to store request counts. All the accounts added have the SID S-1-5-18, which originally belongs to the local system's service accounts. Throttles are akin to permissions, so these actions indicate that the strain tries to change the privilege of the accounts it is using. It then adds the provisioning package `UbYfyvbG8E+B6zqO.0` to the `HKLM\SOFTWARE\Microsoft\Provisioning\Sessions`-key. The provisioning packages contains settings used to configure the device. Adding a provisioning package in this location indicates that the strain will change runtime settings on applications, which allow the ransomware to customise its environment.

The remaining ten keys are added with the SID S-1-5-21-domain-1000, which is a user account on the domain. Five of the keys is associated to the `Microsoft\Speech_OneCore\Recognizers`, and the remaining five values edit the `ApplicationViewManagement\W32`. Three registry keys are deleted, the old `Provisioning\Sessions`-key and two keys related to system updates. These keys are believed to be a coincidence and not related to Ragnar Locker. The

Ragnar Locker strain also adds several values to existing keys. One significant value is stored in the newly created `Provisioning\Sessions`-key. The `NextSessionID` is the same as the key added previously, and the `BeginTime` within that key holds the timestamp of when Ragnar Locker was executed. Knowing when the strain was executed will help build a timeline of events.

Strain 2 do not appear to add any keys to the registry, nor affect the same keys as strain 1. In registry, the most notable with strain 2 is that it deletes the `C:\$Recycle.Bin\S-1-5-21-{domain}-1000\$RE8GXZ1`, which is sorted under Deleted Folders by RegShot. In Windows, all users have their own Recycle Bin. `C:\$Recycle.Bin` is used in Windows 7, 8, and 10. When a file is deleted, its content is stored in an `$R`-file and the metadata in an `$I`-file. The filename is randomly generated, but the original file extension is usually preserved. The `$I`-file corresponding to `$RE8GXZ1` is located under Deleted Files in the RegShot output. The file's content is unknown, but it is not further investigated due to time constraints and a hypothesis that the information gained from the file will be of little practical value.

4.5.2 Memory Analysis

The memory was dumped during the execution of each strain. The tool DumpIt was used for this purpose. DumpIt captures the host's physical memory, meaning that all the data used by Windows in memory at the time of acquisition will be preserved. The capture was started right after the strain was initiated. To ensure that DumpIt finished after Ragnar Locker so that the memory dump was not encrypted, the accessible RAM was reduced to 4GB. By slowing down the operational speed of the virtual machine, Ragnar Locker took up most resources so that DumpIt finished last.

Strain 1 runs as a child process of `explorer.exe`, as shown in Figure 4.19. The process name, `9bdd7f965d1c67`, are the first fourteen characters of the SHA-256 hash. The process spawns Notepad. The ransom note is a `.txt` file and hardcoded to be opened using Notepad, so this behaviour is expected.

The memory was also checked for potential mutexes. Mutexes, or mutual exclusions, locks resources so that only the thread who locked it, can access it (Farber, 2017). When the thread is done with the process, it is unlocked again. The use of mutexes ensures thread safety by reserving shared variables and hindering threads accessing the exact memory location simultaneously (Ta et al.,

PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime
***** 3284	3268	explorer.exe	0xd18a639ff080	79	-	1	False	2021-04-16 13:30:22.000000	N/A
****** 4176	3284	DumpIt.exe	0xd18a639ff080	2	-	1	True	2021-04-16 13:31:19.000000	N/A
******* 4656	4176	conhost.exe	0xd18a639ff080	5	-	1	False	2021-04-16 13:31:19.000000	N/A
****** 4916	3284	vm3dservice.ex	0xd18a639ff080	4	-	1	False	2021-04-16 13:30:37.000000	N/A
****** 3668	3284	9bdd7f965d1c67	0xd18a639ff080	3	-	1	True	2021-04-16 13:32:25.000000	N/A
******* 4516	3668	notepad.exe	0xd18a639ff080	5	-	1	True	2021-04-16 13:32:47.000000	N/A
****** 4932	3284	vmtoolsd.exe	0xd18a639ff080	8	-	1	False	2021-04-16 13:30:37.000000	N/A
**** 908	756	fontdrvhost.ex	0xd18a639ff080	5	-	1	False	2021-04-16 13:30:09.000000	N/A

Figure 4.19: Processes hierarchy in memory showing Ragnar Locker strain 1.

2017). Mutexes created by benign programs tend to be readable and have reasonably self-explanatory names. Ransomware that creates mutexes often uses more scrambled names. Strain 1 did not contain any mutexes created by Ragnar Locker.

Strain 2 showed that the ransomware is consistent in executing as a child process of `explorer.exe`. The main difference in the process tree is that strain 2 does not appear to spawn `notepad.exe` as a child process of `dd79b2abc21e76`. If it does, the frequency and length of the initiation is considerably smaller. Correlative to strain 1, strain 2 do not create any mutexes.

The memory dump from both strains was queried for network connection information, command line parameters, mutants, and hidden or injected code. These queries did not provide any notable results. The lack of network connections somewhat reinforces the findings from the network analysis. As for the lack of significant events in the other operations, there may be several reasons. The three most prominent are:

1. The events were missed due to the timing of the dumps.
2. The events were not initiated, or is not visible, due to the ransomware being executed in a sandboxed environment.
3. Ragnar Locker does not create such events.

Regardless of the reason, the analysis of the Ragnar Locker memory dumps did not yield any significant results. The processes in the dumps were either benign or expected, and unmasked behaviour that correlates to the discoveries in 4.2.

4.6 MITRE ATT&CK

The MITRE ATT&CK framework allows for a standardised presentation of a strain's tactics, techniques, and procedures. Version 8.2 of the ATT&CK Enterprise layer is used in this paper. There are many ways of presenting the TTPs of ransomware or threat actors using ATT&CK. The two most used methods are using a graphical matrix outlining the tactics and the techniques, or by listing the tactics, techniques, and corresponding procedures in a table. As the graphical matrix is better suited for a high-level overview and comparison, this method will be used to present how the strains map to ATT&CK. The procedures are explained in the text. However, it is essential to note that the TTPs discussed in this section are purely elicited from the ransomware strains and may therefore not account for any human actions in a Ragnar Locker campaign.

4.6.1 MITRE ATT&CK and Strain 1

The first Ragnar Locker strain exploits legitimate user accounts to gain access and maintain persistence on systems. Using legitimate accounts also help the attackers evade defence and detection mechanisms, as the activity is less likely to stand out. To further mask their activity as benign, the attackers use ordinary user accounts where possible and do not try to escalate their privileges to a level that would attract attention. One notable exception is the command line arguments run, which requires administrator privileges. The deletion of shadow copies is one example of such use.

The strain makes use of Windows APIs when executing behaviours. These will primarily blend with benign API calls made. Ragnar Locker monitors good API and DLL calls and spawns' processes from them. It may also try to inject arbitrary code in the Windows Explorer process. Process spawning and injection are used to conserve a low profile on the system and escalate privileges.

To maintain persistence on the system, Ragnar Locker is equipped with functionalities to infect the boot sector. Such functions are often referred to as bootkits. Bootkits reside in the Master Boot Record, on the layer below the file system, making them hard to discover and remove. The bootkit was not recovered and analysed during the analysis.

The strain is aware of its environment, and queries disk information to compare the user and computer. These actions are common to use when determining whether the system is real or virtual. Strain 1 collect the ID of the vic-

tim computer, the current user, and the OS product name. It also runs the `GetProcessHeap` function to check if it is run in a debugger. This behaviour was evident when analysing the strain in assembly, and the exact actions are detailed in-depth in Section 4.2. Strain 1 queried several registry keys and fused the results.

Strain 1 deletes the system shadow copies to hinder the system from being recovered to a known good state. To ensure the deletion of shadow copied do not fail, strain 1 uses both `vssadmin.exe` and `wmic.exe` for this purpose. The process of assessing files for encryption is automated, which means that MITRE flags the *Automated Collection* and *Data from Local System*-tags. The strain contains a public RSA-key and calls the Windows Crypto API but shows no sign of attempting to connect to a C2 when run. If Ragnar Locker runs uninterrupted on a system, and its termination criterion is not triggered, Ragnar Locker will encrypt the file system. Table 4.20 presents a high-level overview of the tactics and techniques as classified by MITRE ATT&CK.

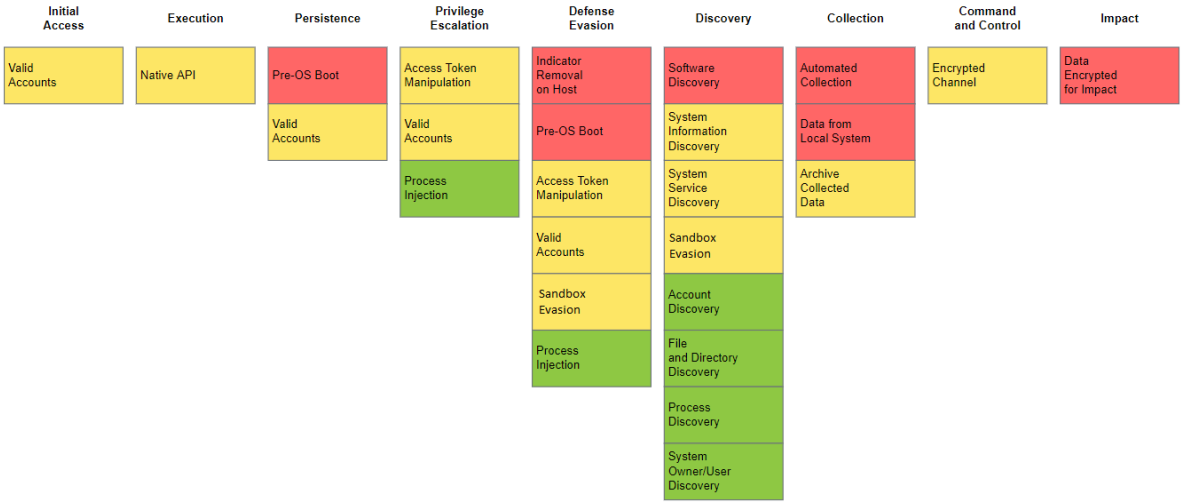


Figure 4.20: MITRE ATT&CK map of Ragnar Locker strain 1.

4.6.2 MITRE ATT&CK and Strain 2

Strain 2 utilise the same techniques as strain 1 for initial access and execution. For persistence and privilege escalation, strain 2 is flagged to create a start menu entry and try to load missing DLL files by JoeSandbox Cloud. Startup items are shell scripts or executable files that are run during the final phase of the computer boot process. The notable thing about the creation of a startup item is that according to MITRE, this is a technique used on Mac OS, not Windows. The start menu entry is likely related to the VM technique used in

Ragnar Locker attacks. To execute the ransomware from the VM, a batch file called `vrun.bat` is run from `C:\DocumentsandSettings\Administrator\StartMenu\Programs\Startup` on the VM (Loman, 2020).

The next tactic that differentiates the two strains is that strain 2 creates a DirectInput-object as a credential access tactic. The Windows OS uses DirectInput objects to communicate with input mechanisms. In situations involving malware, this may indicate an attempt to log user input and keystrokes. In the case of strain 2, the input capture technique was triggered by the processing of command line parameters which most likely refers to the use of the command line discussed in Section 4.2.

Strain 2 also have some added discovery tactics. Querying certain registry keys is often used to determine the behaviour of the strain or to monitor keys for changes. In the case of strain 2, this may relate to the decision to terminate or execute based on the system OS. Strain 2 also reads the hostname and correlated IP addresses.

The command-and-control tactics proxy technique is triggered because the ransom note contains the Tor onion address of Ragnar Lockers DLS. A graphical overview of strain 2's tactics and techniques are presented in Figure 4.21.

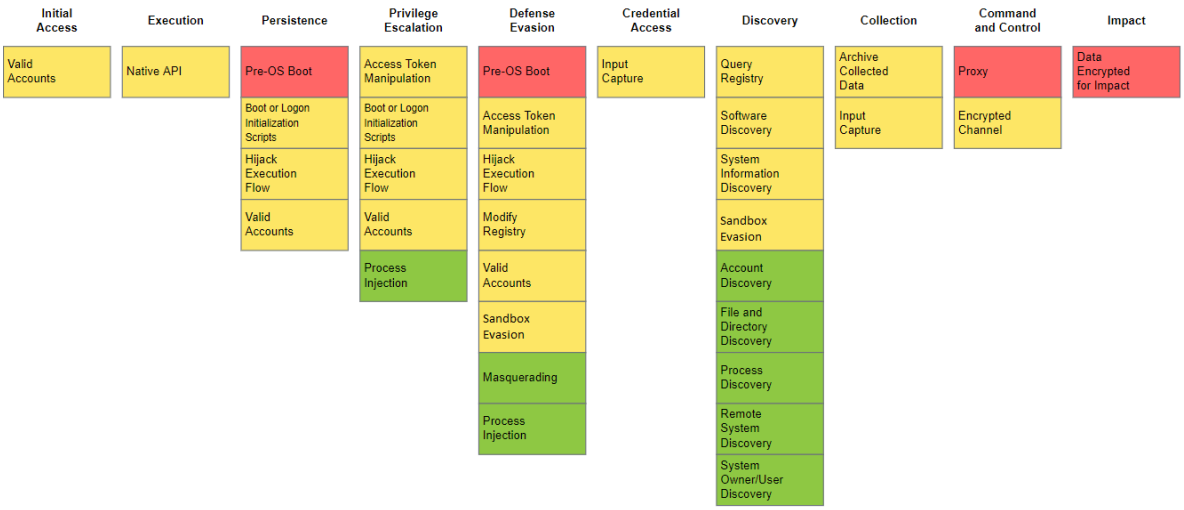


Figure 4.21: MITRE ATT&CK map of Ragnar Locker strain 2.

4.6.3 MITRE ATT&CK Comparison

Most of the tactics and techniques used by the Ragnar Locker ransomware have remained constant from January 2020 to November 2020. Some of the variations

are simply because the ransomware creators have tweaked how the ransomware performs the same actions. One example of this is the proxy technique, which results from the change in the ransom note. Figure 4.22 shows the techniques that both strains employ in green. The techniques in blue are unique to one of the strains.

The severity level of the applied techniques is outlined in Figure 4.23. The heatmap can be useful for determining where additional mitigation efforts should be applied if one wishes to build resilience towards Ragnar Locker attacks. The heatmap can also easily be used to compare Ragnar Locker to other prominent ransomware families.

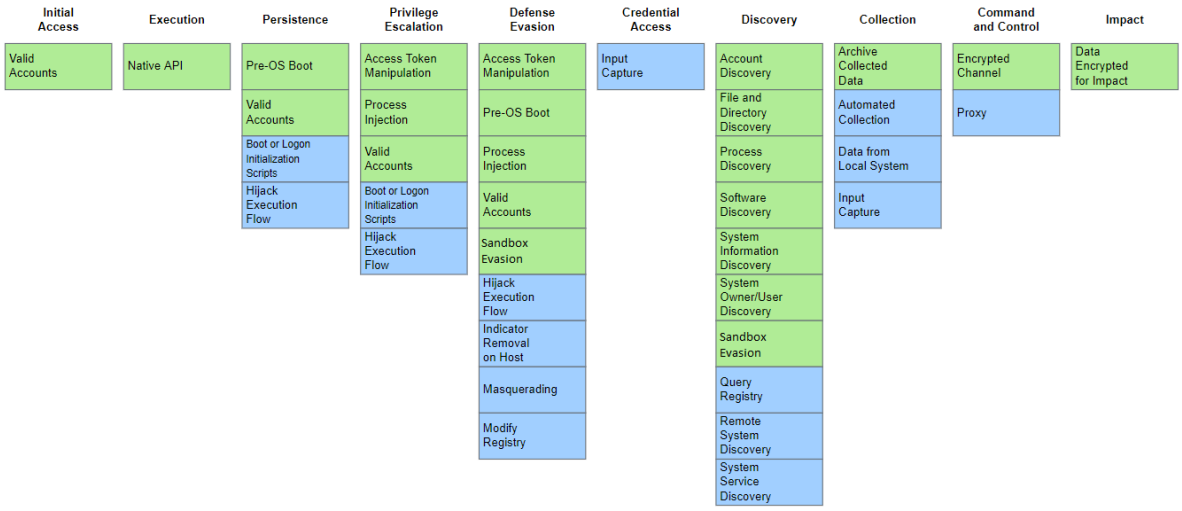


Figure 4.22: MITRE ATT&CK map of similarities and differences between strain 1 and 2.

Figure 4.23: MITRE ATT&CK map of the severity of Ragnar Lockers tactics and techniques.

4.7 MISP

One of the main objectives for the research was to share artefacts, TTPs, and the YARA rule using the MISP Threat Intelligence Platform. Due to time constraints and practicalities, the MISP instance used for this purpose was changed to the MISP maintained by the Norwegian Maritime Cyber Resilience Centre (NORMA Cyber). The sharing level was set to “Connected Communities”. This MISP instance is available to 420 Norwegian ship owners, which in turn operates 3400 vessels, drilling rigs and similar movable units (NORMA Cyber, 2020). Figure 4.24 displays how the indicator section in the MISP instance looks when populated with the data.

Figure 4.24: MISP indicator section.

The MISP instance also allows for mapping MITRE ATT&CK data, attaching analysed files and share various relevant resources and discussions.

4.8 Comparison of Findings and Conclusion

This chapter has detailed the analysis of a Ragnar Locker strain used in January 2020 and one Ragnar Locker strain used in November 2020. In both strains, parts of the code were packed. The packed sections held the encryption keys, the ransom note, and the function parameters, such as services to stop and language settings that would cause the ransomware to terminate itself. No segments were obfuscated or otherwise concealed, indicating that the threat actors may be indifferent to researchers analysing their destructive tool. Both strains execute as a child process of `explorer.exe`. Strain 1 additionally spawns an instance of `notepad.exe`, a behaviour that was not detected in strain 2.

The PE structure and use of Windows APIs have remained consistent from strain 1 to strain 2. Strain 2 is slightly larger in size and call 23 extra DLL files, but this is expected as the strain have some additional functions. The ransomware creators have changed the naming convention of one section of the code; the `.keys`-section has been renamed to the less descriptive name `.didata`. Following this restructuring of the code, the entropy layout for the strains is slightly different. The main difference in entropy is that strain 2 has noticeable higher entropy when unpacked. This difference is likely caused by the way the strain handles the encryption keys. Strain 2 "opens and closes" the keys more, whereas strain 1 unpacks the keys section and does not close it until it is no longer needed.

Stain 1 will obtain its process and terminate itself if it finds that the OS language matches one of the languages used in the Commonwealth of Independent States. The list of languages to check for is hardcoded in the strain. Strain 2 contains the same function but lacks the list of criteria. The hypothesis is that the change may be an unintended bug or that the attackers found the function excessive as they target their victims. The method used to delete potential backup files has been developed during the eleven months separating the two strains. Whereas stain 1 merely deletes the shadow copies, strain 2 disables the computers automatic repair options and instructs it to ignore failure errors. These operations are likely added to make it harder for victims to boot systems into safe mode once they detect that something is amiss in their systems and to disable recovery.

The method applied when encrypting the files is similar in both strains. The only difference is that strain 2 has added some folders, files, and file types to be exempt from encryption. One example is that strain 2 will not encrypt EXE files, whereas strain 1 did. The consequence of this change is that the computer and its programs are somewhat usable even though all files are encrypted, which was not the case with strain 1. Furthermore, the file extension and file footer of encrypted files have changed. Strain 1 uses plain text language, but strain 2 has added numbers, non-standard characters, and a mix of uppercase and lowercase letters. This is likely done to avoid security mechanisms, as it is considerably harder to create rules that will flag the naming conventions used by strain 2.

The overall structure and content of the ransomware code is homogeneous. However, one block of functions in strain 2 stand out as different. The code segment contains command-line options used by the ransomware executable when deployed within a virtual machine. From this, one can deter that the technique of

using an Oracle VirtualBox hypervisor to deploy the ransomware was used when executing strain 2. Upon execution, strain 1 adds a provisioning package to the registry. Within this key, it is possible to find the execution time of the strain. Strain 2 does not perform the same changes. However, it does delete a file. The deleted file was not analysed due to time constraints and scope limitations.

The evolution of the ransom note follows the overall trend observed in the assembly code. The threat actors have excluded information about the decryption price and bitcoin wallet in strain 2, likely to improve the operational security. Moreover, the victim is steered to the attacker's web page instead of a third-party application. These changes add a veil of secrecy and protection, making it harder for researchers and law enforcement to track the threat actors based on the ransomware strain. The threat actor's communication mechanisms are not related to, or connected to, the ransomware. Communication mechanisms used during a breach are therefore likely to be independent implants. Neither strain generate any network activity. The sole purpose of the ransomware is to encrypt the system and inform the victim of the compromise. Key TTPs applied by the strains remain somewhat consistent, with 72.73% of the techniques being the same in both. This further confirms the observations from the analysis, stating that most of the code structure was the same in strain 1 and 2.

Based on these findings, one can infer that the Ragnar Locker group focuses their time and effort on developing their skills and approach to the initial stages of an attack rather than their already functional ransomware tool. Such an approach is logical, as the threat actors can optimise both the victim environment and the ransomware as needed once they have established persistence on the network. Moreover, the Ragnar Locker groups affiliation with other cybercriminal groups does not appear to have impacted the design of their ransomware strain.

5

Conclusion

The growth of agile and tailored ransomware is likely to continue, necessitating procedures that will facilitate thorough analysis of detected strains. Analysis of attacks and the tools applied is an essential part of understanding how the threats evolve and elicit technical details that may help detect future attacks. Openness and sharing of threat intelligence are essential for building technical defences and mitigate the risk before breaches occur.

This paper details the analysis of two strains from the Ragnar Locker ransomware family. The threat actors behind Ragnar Locker tailor their attacks to each victim and are actively present when conducting lateral movement and data exfiltration. The number of manual processes applied to each attack is one factor that leads to exorbitant ransom demands, along with the possession of sensitive data. The ransomware is deployed when the attackers decide to end the engagement.

The ransomware itself have not changed significantly during the eleven months that separate strain 1 and 2. The main change is that strain 2 has an extra set of functions to accommodate the group's new execution tactic. In both strains, parts of the code are packed but not obfuscated. The treat actors hardly attempt

to mask the build of their ransomware, which may infer that they are indifferent to the code being analysed and reverse engineered. Such an attitude may be based on two things: firstly, the threat actors gain access to victim networks and facilitate the execution of the ransomware, which allows them to terminate security mechanisms. Secondly, the knowledge that each entity of the ransomware is customised to the victim company, so antivirus solutions will not flag the strain as malicious as the hash is unknown.

5.1 Summary of Research

Chapter 1 provides an overview of the project. The chapter details the background for the research, research objectives, scope, and limitations.

Chapter 2 is a literature review and elaborate on topics relevant to ransomware research. It reviews the technological aspects a reader should be familiar with before reading the rest of the paper. It also provides an overview of existing research on the discussed topics.

Chapter 3 details how the analysis of the Ragnar Locker strains was planned and performed at an executive level. The ransomware samples are introduced, along with a list of all tools used during the analysis. A section discussing the risks and ethical considerations concludes the chapter.

Chapter 4 presents the findings from the analyses. The strains were found to be quite similar; few modifications were added to the more recent strain. The most significant changes were the addition of functions that provide interoperability with the VM used to deploy the newer ransomware. The ransom note and functions related to generating it were modified to improve operational security. Neither strain contain functionalities for network communication.

Chapter 5 concludes this paper and summarises the overall findings. It also evaluates to which degree the research objectives have been met and discuss to what extent the conducted research contributes to the field of academic ransomware analysis. Finally, this chapter suggests areas for further exploration.

5.2 Research Objectives

5.2.1 Analyse two strains of the Ragnar Locker ransomware using the HMF

Two strains of the Ragnar Locker ransomware were analysed in accordance with the author's interpretation of the steps outlined in the HMF. The findings are detailed in dept in Chapter 4. Based on the level of detail and successful analysis of all aspects related to the strains, it is assessed that this research objective is fully accomplished.

5.2.2 Compare the findings from the analysis of the two strains

The technical findings from the two analyses were compared continuously in Sections 4.2, 4.3, 4.5, and 4.6 and summarised in Section 4.8. One key takeaway from the comparison is that the ransomware has not changed drastically during eleven months of operations. Some extra functionalities have been added to facilitate a new approach to avoid security mechanisms. Spesificaly, the threat actors deploy the ransomware using virtual machines, but this does not change the operation of the ransomware itself. From this, one can deduct that the uniqueness of each Ragnar Locker strain primarily is caused by the victim specific ransom notes and occasional tailoring to disrupt processes running on the victim system. Considering these findings, this research objective has been met.

5.2.3 Map the discovered TTPs in the MITRE ATT&CK framework

Both strains were parsed into the MITRE ATT&CK framework to concretise their behaviour into a standardised format. The procedures used to perform the different techniques were detailed in Section 4.6 and linked to the findings in Section 4.2. To further enrich the comparison of the two strains, they were also graphically compared in two ATT&CK matrixes, one outlining the common techniques and one heatmap to display the presumed severity of the actions. As the strains TTPs are mapped and compared in the MITRE ATT&CK framework, this research objective is realised.

5.2.4 Provide a YARA-rule that will detect both strains

A YARA rule that detects Ragnar Locker as per April 2021 has been provided in Section 4.3.2. The rule detects both the analysed strains as well as other public Ragnar Locker strains that have targeted organisations in the period January 2020 to April 2021. This objective is deemed accomplished.

5.2.5 Create a list of Indicators of Compromise for the Strains

Indicators of Compromise elicited from the analysed strains are presented in Section 4.3.3, along with a short evaluation of their future usefulness. However, it is essential to note that these IOCs are based on the ransomware strains alone and will not help identify or detect tools and techniques used before the execution of the ransomware payload. The research objective of creating a list of IOCs directly related to the ransomware is evaluated to be met.

5.2.6 Contribute the discovered IOCs and YARA rule to the MISP Galaxy

The original plan was to submit relevant findings to the MISP Galaxy instance to contribute to the security community. Setting up and connecting to MISP proved to be a tedious task. Due to time constraints, this goal was slightly changed during the project. The solution was to contribute the findings to an already established and configured MISP. The chosen MISP instance was that of the Norwegian Maritime Cyber Resilience Centre due to the researcher's affiliation with the organisation. Although the original objective is not fully fulfilled, the primary intention of the goal is achieved.

5.3 Research Contribution

The research presented in this paper contributes to the common knowledge of how Ragnar Locker operates. Systematically analysing ransomware strains' key traits enables researchers to compare results, both within one family and against other species. Such an approach will facilitate the construction of solid and effective prevention mechanisms. An in-depth analysis of pioneering cybercriminal groups may help to understand how such threat actors operationalise their goals. In the case of Ragnar Locker, it is evident that the group focuses more on locating

targets and applying an arsenal of skills to gain access and move laterally, rather than developing and obfuscating their ransomware. The ransomware itself is just another tool in their kit and is solemnly used as an extortion technique. This knowledge shows how drastically ransomware operations have changed the past few years and underlines the importance of defence-in-depth and sound monitoring. By uncovering similarities between strains and families, one can possibly predict behaviour characteristics in future strains. Many security solutions rely on signature-based detection, which only enables them to detect known threats. Signature-based detection mechanisms are evidently not enough to safeguard a company.

Moreover, this research adds to the academic work focusing on cybercrime and ransomware. Few research papers discuss ransomware operations and analysis, especially human-operated ransomware attacks. This is a contrast to the threat such attacks pose to society. Most such academic papers focus on developing mechanisms to detect and deter attacks.

The Hybridised Malware Framework developed by Schmitt (2019) coalesce existing methods for reverse engineering malicious code into one forensic oriented approach. The framework has previously been tested on automated ransomware operations. By analysing Ragnar Locker using the steps outlined in the framework, this analysis proves that the framework is functional when applied to current ransomware threats as well.

The analysis presented in this paper present some of Ragnar Locker's traits. Artefacts produced include a YARA rule and a list of IOCs. The YARA rule can be implemented as-is to security solutions and may add direct value to the security community. The list of IOCs can be added to threat intelligence platforms, but the IOCs' operationalisation will depend on what each entity considers deprecated. That said, care has been taken to select IOCs that are likely to remain somewhat constant, such as the manifest hash that was the same for both strains.

5.4 Future Work

Future work should seek to obtain actual Ragnar Locker breach data, such as logs from a system where the threat actors have been at work. Analysing such artefacts and comparing them to the findings outlined in this report, or other analyses of Ragnar Locker, would serve two purposes. Scope one should be to determine the actions made by the operators on a system. Findings related to human actions may provide valuable insight into how such treat actors reconnoitre environments. Such an analysis could potentially also uncover communication methods used by the threat actors. The technical analysis of their tools, such as ransomware strains, are only an enrichment of such knowledge. The second purpose is to analyse authentic breach data to help validate the findings from the obtained strains.

Future work should include analysing the VM used to deploy strain 2 and merge the findings to expand on the analysis detailed in this paper. Incorporating a VM analysis would provide a more holistic overview of the tools used by the Ragnar Locker group.

Ragnar Locker is a human-operated ransomware; thus, attacks carried out by this actor is fundamentally different from the Cerber, Maktub, and Locky attacks previously analysed with the HMF. To gain a broader understanding of how contemporary ransomware strains are built, one possibility for future work is to analyse other ransomware families used in big game hunting by human operators.

The Hybridised Malware Framework by Schmitt (2019) is a decent analysis framework outlining crucial steps in a deep analysis of malware. In addition to serving as a guideline for malware analysis, the framework has been academically used to compare different malware strains and families to uncover common traits. Depending on the strain in question, in-depth analysis can be a time-consuming task. Future work should investigate developing a comparison module utilising fuzzy hashing to develop the framework further so it is fit for code analysis and comparison of a multitude of malwares. By incorporating an automated fuzzy hashing module researchers can compare strains to each other at the code level and uncover similarities and relations more accurately.

Bibliography

Abrams, L. (2020a, May). List of ransomware that leaks victims' stolen files if not paid. Retrieved October 8, 2020, from https://www.bleepingcomputer.com/news/security/list-of-ransomware-that-leaks-victims-stolen-files-if-not-paid/

Abrams, L. (2020b, June). Maze Ransomware adds Ragnar Locker to its extortion cartel. Retrieved October 8, 2020, from https://www.bleepingcomputer.com/news/security/maze-ransomware-adds-ragnar-locker-to-its-extortion-cartel/

Abrams, L. (2020c, November). Maze ransomware shuts down operations, denies creating cartel. Retrieved April 22, 2021, from https://www.bleepingcomputer.com/news/security/maze-ransomware-shuts-down-operations-denies-creating-cartel/

Abrams, L. (2020d, March). Ransomware Gangs to Stop Attacking Health Orgs During Pandemic. Retrieved October 9, 2020, from https://www.bleepingcomputer.com/news/security/ransomware-gangs-to-stop-attacking-health-orgs-during-pandemic/

Abulaish, M., & Haldar, N. A. H. (2020, April). *Digital Forensics and Forensic Investigations*. IGI Global. https://doi.org/10.4018/978-1-7998-3025-2

Ahmadian, M. M., Shahriari, H. R., & Ghaffarian, S. M. Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares. In: *12th international isc conference on information security and cryptology, iscisc 2015*. Institute of Electrical; Electronics Engineers Inc., 2016, January, 79–84. ISBN: 9781467376099. https://doi.org/10.1109/ISCISC.2015.7387902.

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

Al-Shaer, R., Spring, J. M., & Christou, E. (2020). *Learning the Associations of MITRE ATT&CK Adversarial Techniques* (tech. rep.).

Ali, M., & Papadaki, M. (2018). *Agent-based Vs Agent-less Sandbox for Dynamic Behavioral Analysis* (tech. rep.).

AlienVault. (n.d.). AlienVault - Open Threat Exchange. Retrieved April 22, 2021, from https://otx.alienvault.com/

Anton, T. (2019, April). The need to manage both symmetric and asymmetric keys. Retrieved December 1, 2020, from https://www.cryptomathic.com/news-events/blog/the-need-to-manage-both-symmetric-and-asymmetric-keys

Arabo, A., Dijoux, R., Poulain, T., & Chevalier, G. (2020). Detecting Ransomware Using Process Behavior Analysis. *Procedia Computer Science*, *168*, 289–296. https://doi.org/10.1016/j.procs.2020.02.249

Arntz, P. (2017, March). Explained: Packer, Crypter, and Protector. Retrieved December 12, 2020, from https://blog.malwarebytes.com/cybercrime/malware/2017/03/explained-packer-crypter-and-protector/

Assis, C. R., Miani, R. S., Carneiro, M. G., & Park, K. J. A comparative analysis of classifiers in the recognition of packed executables. In: *Proceedings - international conference on tools with artificial intelligence, ictai. 2019-Novem*. IEEE Computer Society, 2019, November, 1356–1360. ISBN: 9781728137988. https://doi.org/10.1109/ICTAI.2019.00189.

Bahrami, P. N., Dehghantanha, A., Dargahi, T., Parizi, R. M., Choo, K. K. R., & Javadi, H. H. (2019). Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tactics, techniques, and procedures. *Journal of Information Processing Systems*, *15*(4), 865–889. https://doi.org/10.3745/JIPS.03.0126

Balci, A., Ungureanu, D., & VondruŠka, J. (2020). *Malware Reverse Engineering Handbook* (tech. rep.). NATO Cooperative Cyber Defence Centre of Excellence. Tallinn. www.ccdcoe.org

Barkly. (2018). *The Fileless Attack Survival Guide* (tech. rep.). https://dsimg.ubm-us.net/envelope/395823/551993/FilelessAttackSurvivalGuide.pdf

Bayer, U., Moser, A., Kruegel, C., & Kirda, E. Dynamic analysis of malicious code. In: *Journal in computer virology*. *2*. (1). 2006, August, 67–77. https://doi.org/10.1007/s11416-006-0012-2.

Bisson, D. (2020, June). Maze Gang Forms Ransomware Cartel to Extort From Non-Paying Victims. Retrieved October 8, 2020, from https://securityintelligence.com/news/ransomware-news-maze-gang-forms-extortion-cartel/

Blaze Information Security. (2020, July). Dissecting Ragnar Locker: The Case Of EDP. Retrieved October 9, 2020, from https://blog.blazeinfosec.com/dissecting-ragnar-locker-the-case-of-edp/

Borchani, Y. (2020). Advanced malicious beaconing detection through AI. *Network Security*, *2020*(3), 8–14. https://doi.org/10.1016/S1353-4858(20)30030-1

Cabaj, K., & Mazurczyk, W. (2016). Using software-defined networking for ransomware mitigation: The case of cryptowall. *IEEE Network*, *30*(6), 14–20. https://doi.org/10.1109/MNET.2016.1600110NM

Carbon Black. (2017, October). Dark Web Ransomware Economy Growing at an Annual Rate of 2,500%, Carbon Black Research Finds. Retrieved October 9, 2020, from https://www.carbonblack.com/press-releases/dark-web-ransomware-economy-growing-annual-rate-2500-carbon-black-research-finds/

Carbon Black. (n.d.). What is Cyber Threat Hunting [Last accessed on 2020-09-06]. Retrieved September 6, 2020, from https://www.carbonblack.com/definitions/what-is-cyber-threat-hunting/

Churchman, M. (2017, August). The Importance of Operational Maturity and Being Application-Centric. Retrieved April 22, 2021, from https://www.pagerduty.com/blog/operational-maturity-and-application-centric/

Constantin, L. (2021, February). Egregor ransomware takes a hit after arrests in Ukraine. Retrieved April 22, 2021, from https://www.csoonline.com/article/3608368/egregor-ransomware-takes-a-hit-after-arrests-in-ukraine.html

Cook, S. (2020, June). Ransomware Statistics 2018-2020 : 50+ Ransomware Stats & Facts. Retrieved October 9, 2020, from https://www.comparitech.com/antivirus/ransomware-statistics/

Cortesi, A. (2015). Visualizing binaries with space-filling curves. Retrieved April 3, 2021, from https://corte.si/posts/visualisation/binvis/

Coveware. (2020, August). Ransomware Attacks Split Between Enterprise & RaaS. Retrieved October 10, 2020, from https://www.coveware.com/blog/q2-2020-ransomware-marketplace-report{\#}types

Crowdstrike. (2019, February). Fancy Bear Hackers (APT28): Targets & Methods. Retrieved October 14, 2020, from https://www.crowdstrike.com/blog/who-is-fancy-bear/

CrowdStrike. (2021). Global Threat Report 2021. *CrowdStrike*, 75.

Dahle, T. (2020). *Large Scale Vulnerability Scanning Development of a large-scale web scanner for detecting vulnerabilities* (tech. rep.). Oslo Metropolitan

University. Oslo. https://www.duo.uio.no/bitstream/handle/10852/79552/torjuskd-master-2020v1-4.pdf?sequence=8{\&}isAllowed=y

Davies, S. R., Macfarlane, R., & Buchanan, W. J. (2020). Evaluation of live forensic techniques in ransomware attack mitigation. *Forensic Science International: Digital Investigation*, *33*. https://doi.org/10.1016/j.fsidi.2020.300979

Done, Z. (2020, June). Let's analyze Ragnar Locker. Retrieved October 16, 2020, from https://zawadidone.nl/2020/06/01/lets-analyze-ragnar-locker.html

Dunham, K. (2013). A fuzzy future in malware research. *The ISSA Journal*. https://bluetoadpublishing.co.uk/publication/?m=1336&i=162101&p=32

Eskandari, M., Khorshidpour, Z., & Hashemi, S. (2013). HDM-Analyser: A hybrid analysis approach based on data mining techniques for malware detection. *Journal in Computer Virology*, *9*(2), 77–93. https://doi.org/10.1007/s11416-013-0181-8

Europol. (2020). *Internet Organised Crime Threat Assessment* (tech. rep.). European Union Agency for Law Enforcement Cooperation.

Exabeam. (2016). *THE ANATOMY OF A RANSOMWARE ATTACK* (tech. rep.). Exabeam.

Farber, R. From serial to parallel programming using OpenACC. In: *Parallel programming with openacc*. Elsevier Inc., 2017, January. Chap. 1, pp. 1–28. ISBN: 9780124103979. https://doi.org/10.1016/B978-0-12-410397-9.00001-9.

FBI. (2020). *Indicators of Compromise Associated with Ragnar Locker Ransomware* (tech. rep.).

Frankoff, S., & Hartley, B. (2018, November). Big Game Hunting: The Evolution of INDRIK SPIDER From Dridex Wire Fraud to BitPaymer Targeted Ransomware. Retrieved October 14, 2020, from https://www.crowdstrike.com/blog/big-game-hunting-the-evolution-of-indrik-spider-from-dridex-wire-fraud-to-bitpaymer-targeted-ransomware/

Galal, H. S., Mahdy, Y. B., & Atiea, M. A. (2016). Behavior-based features model for malware detection. *Journal of Computer Virology and Hacking Techniques*, *12*(2), 59–67. https://doi.org/10.1007/s11416-015-0244-0

Goncharov, M. (2015). *Criminal Hideouts for Lease: Bulletproof Hosting Services* (tech. rep.). Trend Micro.

Greten, D. (2020, June). Indicators of Compromise (IOCs) and Artifacts: What's the Difference? Retrieved December 2, 2020, from https://www.vmray.com/cyber-security-blog/indicators-of-compromise-artifacts-whats-the-difference/

Gupta, A. (2018). *The Dark Web as a Phenomenon: A Review and Research Agenda* (tech. rep.). The University of Melbourne. Melbourne.

Hovmark, O., & Schüldt, E. (2020). *Towards Extending Probabilistic Attack Graphs with Forensic Evidence An investigation of property list files in macOS* (tech. rep.).

Husari, G., Al-Shaer, E., Chu, B., & Rahman, R. F. (2019). Learning APT Chains from Cyber Threat Intelligence. https://doi.org/10.1145/3314058.3317728

IBM Security. (2020). *X-Force Threat Intelligence Index* (tech. rep.). IBM X-Force Incident Response and Intelligence Services.

Ilaschu, I. (2021, April). Capcom: Ransomware gang used old VPN device to breach the network. Retrieved April 18, 2021, from https://www.bleepingcomputer.com/news/security/capcom-ransomware-gang-used-old-vpn-device-to-breach-the-network/

Johnson, A. (2018). *Making Backdoor Access Look Like Google Requests* (tech. rep.). https://www.theverge.com/2018/4/18/17253784/google-domain-fronting-

Joseph, P., & Norman, J. (2020). Systematic Memory Forensic Analysis of Ransomware using Digital Forensic Tools. *International Journal of Natural Computing Research*, *9*(2), 61–81. https://doi.org/10.4018/ijncr.2020040105

Kaspersky. (2020). Attack Vector. Retrieved December 16, 2020, from https://encyclopedia.kaspersky.com/glossary/attack-vector/

Kerr, D., & Ewing, P. (2018). *The Endgame Guide to Threat Hunting* (S. Shuttleworth, Ed.; Vol. 53). CyberEdge Group, LLC.

Khan, S., Gani, A., Wahab, A. W. A., Shiraz, M., & Ahmad, I. (2016). Network forensics: Review, taxonomy, and open challenges. https://doi.org/10.1016/j.jnca.2016.03.005

Kim, P. (2018). *THE HACKER PLAYBOOK 3 - Practical Guide To Penetration Testing - Red Team Edition*. Secure Planet.

Krebs, B. (2020). Ransomware Group Turns to Facebook Ads. https://krebsonsecurity.com/2020/11/ransomware-group-turns-to-facebook-ads/

Kumar, P. R., & Ramlie, H. R. E. B. H. Anatomy of Ransomware: Attack Stages, Patterns and Handling Techniques. In: Springer, Cham, 2021, January, pp. 205–214. https://doi.org/10.1007/978-3-030-68133-3_20.

Latzo, T., Palutke, R., & Freiling, F. (2019). A universal taxonomy and survey of forensic memory acquisition techniques. *Digital Investigation*, *28*, 56–69. https://doi.org/10.1016/j.diin.2019.01.001

Ligh, M. H., Adair, S., Hartstein, B., & Richard, M. (2011). *Malware Analyst's Cookbook and DVDTools and Techniques for Fighting Malicious Code* (C. Long & M. Gregg, Eds.; 1st). Wiley Publishing, Inc.

Ligh, M. H., Case, A., Levy, J., & Walters, A. A. (2014). *The Art of Memory ForensicsDetecting Malware and Threats in Windows, Linux, and Mac Memory* (1st). Wiley.

Lockheed Martin. (n.d.). The Cyber Kill Chain. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Logan, M., Mendoza, E., Maglaque, R., & Tamaña, N. (2021, February). *The State of Ransomware: 2020's Catch-22* (tech. rep.). Trend Micro. https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/the-state-of-ransomware-2020-s-catch-22

Loman, M. (2020, April). Ragnar Locker ransomware deploys virtual machine to dodge security. Retrieved August 28, 2020, from https://news.sophos.com/en-us/2020/05/21/ragnar-locker-ransomware-deploys-virtual-machine-to-dodge-security/

Lord, N. (2018, September). What is Advanced Malware? Retrieved December 12, 2020, from https://digitalguardian.com/blog/what-advanced-malware

Makrushin, D. (2015, August). Indicators of compromise as a way to reduce risk. Retrieved September 3, 2020, from https://securelist.com/indicators-of-compromise-as-a-way-to-reduce-risk/71915/

Malik, D. A. A. M. A. A. (2018). Electronic Crime Investigation. *IJECI*, *2*(2), 7–7. www.moneycontrol.com/cryptocurrency

Meijerink, M., Jair, J., De Santanna, C, Sperotto, A., & Perniola, A. (2019). *Anomaly-based Detection of Lateral Movement in a Microsoft Windows Environment Using the Windows security event log for detecting lateral movement techniques executed by a professional red team* (tech. rep.).

Meland, P. H., Bayoumy, Y. F. F., & Sindre, G. (2020). The Ransomware-as-a-Service economy within the darknet. *Computers and Security*, *92*, 101762. https://doi.org/10.1016/j.cose.2020.101762

Meyes, A. (2019, February). Advanced Persistent Threat List. Retrieved October 14, 2020, from https://www.crowdstrike.com/blog/meet-the-adversaries/

Microsoft. (2019, September). CVE-2019-1182 — Remote Desktop Services Remote Code Execution Vulnerability. Retrieved October 10, 2020, from https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2019-1182

Microsoft Docs. (2017, February). BCDEdit Command-Line Options. Retrieved March 29, 2021, from https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/bcdedit-command-line-options

Microsoft Docs. (2020). Cryptographic Provider Type. Retrieved April 4, 2021, from https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-gpnap/e58d0d81-6cb4-4e07-bbc3-1e27978c1e72

Microsoft Docs. (n.d.). Guid Struct (System). Retrieved April 11, 2021, from https://docs.microsoft.com/en-us/dotnet/api/system.guid?view=net-5.0

Microsoft Security. (2020). Human-operated ransomware attacks: A preventable disaster. Retrieved March 29, 2021, from https://www.microsoft.com/security/blog/2020/03/05/human-operated-ransomware-attacks-a-preventable-disaster/

MISP. (n.d.). MISP - Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing. Retrieved October 16, 2020, from https://www.misp-project.org/

Mitnick, K. D. (2011). *Gohst in the wires* (2nd). Hachette Book Group.

MITRE. (n.d.-a). APT28. Retrieved October 14, 2020, from https://attack.mitre.org/groups/G0007/

MITRE. (n.d.-b). Groups. Retrieved October 12, 2020, from https://attack.mitre.org/groups/

MITRE. (n.d.-c). MITRE ATT&CK®. Retrieved October 12, 2020, from https://attack.mitre.org/

Monnappa, K. A. (2018). *Learning Malware Analysis* (G. George, Ed.; 1st). Packt Publishing.

Montasari, R. (2016). Review and Assessment of the Existing Digital Forensic Investigation Process Models. *International Journal of Computer Applications*, *147*(7), 41–49. https://doi.org/10.5120/ijca2016911194

Naik, N., Jenkins, P., Savage, N., Yang, L., Boongoen, T., Iam-On, N., Naik, K., & Song, J. (2020a). Embedded YARA rules: strengthening YARA rules utilising fuzzy hashing and fuzzy rules for malware analysis. *Complex Intelligent Systems*, *7*(2), 687–702. https://doi.org/10.1007/s40747-020-00233-5

Naik, N., Jenkins, P., Savage, N., Yang, L., Naik, K., Song, J., Boongoen, T., & Iam-On, N. Fuzzy Hashing Aided Enhanced YARA Rules for Malware Triaging. In: *2020 ieee symposium series on computational intelligence, ssci 2020*. Institute of Electrical; Electronics Engineers Inc., 2020, December, 1138–1145. ISBN: 9781728125473. https://doi.org/10.1109/SSCI47803.2020.9308189.

NCCIC. (2015). *Using YARA for Malware Detection* (tech. rep.). The National Cybersecurity and Communications Integration Center. moz-extension://dd9e2f78-17bf-4522-8319-d3a83c2e695b/enhanced-reader.html?openApp&pdf=https%3A%2F%2Fus-cert.cisa.gov%2Fsites%2Fdefault%2Ffiles%2FFactSheets%2FNCCIC%2520ICS_FactSheet_YARA_S508C.pdf

NORMA Cyber. (2020). About. Retrieved May 11, 2021, from https://www.normacyber.no/about

Obbayi, L. (2018). Threat Hunting: IOCs and Artifacts. Retrieved September 3, 2020, from https://resources.infosecinstitute.com/category/enterprise/threat-hunting/iocs-and-artifacts/{\#}gref

O'Kane, P., Sezer, S., & Carlin, D. (2018). Evolution of ransomware. *IET Networks*, *7*(5), 321–327. https://doi.org/10.1049/iet-net.2017.0207

Osterman Research. (2016). *Best Practices for Dealing With Phishing and Ransomware - DomainTools* (tech. rep.). www.ostermanresearch.com@mosterman

Paar, C., & Pelzl, J. (2010). *Understanding Cryptography* (2nd ed.). Springer Verlag. https://doi.org/10.1007/978-3-642-04101-3

Palutke, R., Block, F., Reichenberger, P., & Stripeika, D. (2020). Hiding Process Memory Via Anti-Forensic Techniques. *Forensic Science International: Digital Investigation*, *33*, 301012. https://doi.org/10.1016/j.fsidi.2020.301012

Petters, J. (2020, June). MITRE ATTCK Framework: Everything You Need to Know. Retrieved March 31, 2021, from https://www.varonis.com/blog/mitre-attck-framework-complete-guide/

Pomeranz, A. (2018). Introduction to the compiler, linker, and libraries. Retrieved April 3, 2021, from https://www.learncpp.com/cpp-tutorial/introduction-to-the-compiler-linker-and-libraries/

Prayudi, Y., & SN, A. (2015). Digital Chain of Custody: State of The Art. *International Journal of Computer Applications*, *114*(5), 1–9. https://doi.org/10.5120/19971-1856

Purplesec. (2021). 2020 Ransomware Statistics, Data, & Trends.

PwC. (2020). *Cyber Threats 2019: A Year in Retrospect* (tech. rep. February).

Roth, F. (2015, February). How to Write Simple but Sound Yara Rules. Retrieved April 10, 2021, from https://www.nextron-systems.com/2015/02/16/write-simple-sound-yara-rules/

Roundy, K. A., & Miller, B. P. Hybrid analysis and control of malware. In: *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). 6307 LNCS*. Springer,

Berlin, Heidelberg, 2010, 317–338. ISBN: 3642155111. https://doi.org/10.1007/978-3-642-15512-3_17.

Ruff, N. (2008). Windows memory forensics. *Journal in Computer Virology*, *4*(2), 83–100. https://doi.org/10.1007/s11416-007-0070-0

Sammons, J. (2015). *The basics of digital forensics, second edition* (C. Katsaropoulos, Ed.; 2nd). Elsevier inc.

Sanders, C. (2017). *Practical Packet Analysis: Using Wireshark to Solve Real-World Problems* (3rd ed.). https://books.google.com/books?hl=en&lr=&id=kNOaDgAAQBAJ&oi=fnd&pg=PP2&dq=what+is+network+packet+capture+and+analysis&ots=fPbergy6EB&sig=chSMo--pJW2jLpQM2EFwu1LU72I#v=onepage&q=whatisnetworkpacketcaptureandanalysis&f=false

Sarantinos, N., Benza¨ıdbenza¨ıd, C., Arabiat, O., & Al-Nemrat, A. (2016). *Forensic Malware Analysis: The Value of Fuzzy Hashing Algorithms in Identifying Similarities* (tech. rep.).

Saravanan, M, & Krishnan, M. (2014). *Forensic Recovery of Fully Encrypted Volume* (tech. rep. No. 7).

Saxe, J., & Sanders, H. (2018). *Malware Data Science - attack Detection and attribution* (W. Pollock, Ed.; 1st). No Starch Press, Inc.

Schmitt, V. (2019). *A comparative study of CERBER, MAKTUB and LOCKY* (Doctoral dissertation). Rohdes University. Grahamstown. http://hdl.handle.net/10962/92313

Scroxton, A. (2020, September). Maze ransomware borrows Ragnar Locker tactics to sneak past defences. Retrieved October 8, 2020, from https://www.computerweekly.com/news/252489198/Maze-ransomware-borrows-Ragnar-Locker-tactics-to-sneak-past-defences

Serketzis, N., Katos, V., Ilioudis, C., Baltatzis, D., & Pangalos, G. (2019). Improving Forensic Triage Efficiency through Cyber Threat Intelligence. *Future Internet*, *11*(7), 162. https://doi.org/10.3390/fi11070162

Shen, C., & Baker, J. (2020). CMA CGM confirms ransomware attack. Retrieved May 15, 2021, from https://lloydslist.maritimeintelligence.informa.com/LL1134044/CMA-CGM-confirms-ransomware-attack

Sikorski, M., & Honig, A. (2012). *Practical malware analysis: the hands-on guide to dissecting malicious software. no starch press*.

Simmons, C. B., Shiva, S. G., Bedi, H., & Dasgupta, D. 9th Annual Symposium on Information Assurance (ASIA '14) General ASIA Chair: conference proceedings. In: *9th annual symposium on information assurance*. University

of Memphis. New York: Annual NYS Cyber Security Conference Empire, 2014.

Simons, J. (2019, November). FBI Issues a PSA About Ransomware: "Big-Game Hunting" is on the Rise! Retrieved October 12, 2020, from https://www.ics-com.net/ransomware-big-game-hunting/

Simpson, D. (2021, August). Human Operated Ransomware. Retrieved March 29, 2021, from https://docs.microsoft.com/en-us/security/compass/human-operated-ransomware

Sophos. (2020). *The State of Ransomware 2020* (tech. rep.). Sophos.

Steffens, T. (2020). *Attribution of Advanced Persistent Threats*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-61313-9

Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., & Thomas, C. B. (2018). *MITRE ATT&CK™: Design and Philosophy* (tech. rep.).

Subedi, K. P., Budhathoki, D. R., & Dasgupta, D. (2018). *Forensic Analysis of Ransomware Families using Static and Dynamic Analysis* (tech. rep.). University of Memphis. Memphis. USA. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp={\&}arnumber=8424649

Symmantec. (2018). *ISTR Internet Security Threat Report Volume 23* (tech. rep.).

Ta, T., Troendle, D., & Jang, B. Thread communication and synchronization on massively parallel GPUs. In: *Advances in gpu research and practice*. Elsevier Inc., 2017, January. Chap. 3, pp. 57–81. ISBN: 9780128037881. https://doi.org/10.1016/B978-0-12-803738-6.00003-3.

Tavares, P. (2020, June). Ragnar Locker malware: what it is, how it works and how to prevent it. Retrieved October 9, 2020, from https://resources.infosecinstitute.com/ragnar-locker-malware-what-it-is-how-it-works-and-how-to-prevent-it-malware-spotlight/

Thomas, J. E., Galligher, R. P., Thomas, M. L., & Galligher, G. C. (2019). Enterprise Cybersecurity: Investigating and Detecting Ransomware Infections Using Digital Forensic Techniques. *Computer and Information Science*, *12*(3). https://doi.org/10.5539/cis.v12n3p72

Tounsi, W., & Rais, H. (2018, January). A survey on technical threat intelligence in the age of sophisticated cyber attacks. https://doi.org/10.1016/j.cose.2017.09.001

Tsakalidis, G., & Vergidis, K. (2019). A Systematic Approach Toward Description and Classification of Cybercrime Incidents. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*(4), 710–729. https://doi.org/10.1109/TSMC.2017.2700495

Turedi, Z. (2020, July). Changing Tactics: Adversaries are Becoming More Brazen and Stealthy in Their Targeting. Retrieved March 29, 2021, from https : / / www . infosecurity - magazine . com / opinions / changing - tactics - adversaries/

Uroz, D., & Rodríguez, R. J. (2020). On Challenges in Verifying Trusted Executable Files in Memory Forensics. *Forensic Science International: Digital Investigation*, *32*, 300917. https://doi.org/10.1016/j.fsidi.2020.300917

Vajapeyam, S. (2014). *Understanding Shannon's Entropy metric for Information* (tech. rep.).

Verizon. (2020). *Data Breach Investigations Report* (tech. rep.).

Verma, V., Muttoo, S. K., & Singh, V. B. (2020). Multiclass malware classification via first- and second-order texture statistics. *Computers and Security*, *97*, 101895. https://doi.org/10.1016/j.cose.2020.101895

Vinot, R. (2020). MISP/misp-taxonomies: Taxonomies used in MISP taxonomy system and can be used by other information sharing tool. Retrieved October 16, 2020, from https://github.com/MISP/misp-taxonomies/

Waddell, K. (2016, May). The Computer Virus That Haunted Early AIDS Researchers. Retrieved October 12, 2020, from https://www.theatlantic.com/technology/archive/2016/05/the-computer-virus-that-haunted-early-aids-researchers/481965/

Wagner, C., Dulaunoy, A., Wagener, G., & Iklody, A. MISP - The design and implementation of a collaborative threat intelligence sharing platform. In: *Wiscs 2016 - proceedings of the 2016 acm workshop on information sharing and collaborative security, co-located with ccs 2016*. New York, New York, USA: Association for Computing Machinery, Inc, 2016, October, 49–56. ISBN: 9781450345651. https://doi.org/10.1145/2994539.2994542.

Wang, P., & Wang, Y. S. (2015). Malware behavioural detection and vaccine development by using a support vector model classifier. *Journal of Computer and System Sciences*, *81*(6), 1012–1026. https://doi.org/10.1016/j.jcss.2014.12.014

Wen, S., Rao, Y., & Yan, H. Information protecting against APT based on the study of cyber kill chain with weighted bayesian classification with correction factor. In: *Acm international conference proceeding series*. New York, New York, USA: Association for Computing Machinery, 2018, March, 231–235. ISBN: 9781450363624. https://doi.org/10.1145/3208854.3208893.

YARA. (2020). Writing YARA rules. Retrieved April 10, 2021, from https://yara.readthedocs.io/en/v4.0.5/writingrules.html

Young, A. L., & Yung, M. (2017). On Ransomware and Envisioning the Enemy of Tomorrow. *Computer*, *50*(11), 82–85. https://doi.org/10.1109/MC.2017.4041366

Zanoramy Zakaria, W. A., Faizal Abdollah, M. O., & Fadillah Mohd Ariffin, A. (2017). The Rise of Ransomware. https://doi.org/10.1145/3178212.3178224

Zimba, A., & Chishimba, M. (2019a). On the Economic Impact of Crypto-ransomware Attacks: The State of the Art on Enterprise Systems. *European Journal for Security Research*, *4*(1), 3–31. https://doi.org/10.1007/s41125-019-00039-8

Zimba, A., & Chishimba, M. (2019b). Understanding the Evolution of Ransomware: Paradigm Shifts in Attack Structures. *Computer Network and Information Security*, *1*, 26–39. https://doi.org/10.5815/ijcnis.2019.01.03

A

Appendix A

Hello PSE_CREDIT_UNION !

****

If you reading this message, then your network was PENETRATED and all of your files and data has been ENCRYPTED

by RAGNAR_LOCKER !

****

*********What happens with your system ?**********

Your network was penetrated, all your files and backups was locked! So from now there is NO ONE CAN HELP YOU to get your files back, EXCEPT US.
You can google it, there is no CHANCES to decrypt data without our SECRET KEY.

But don't worry ! Your files are NOT DAMAGED or LOST, they are just MODIFIED. You can get it BACK as soon as you PAY.
We are looking only for MONEY, so there is no interest for us to steel or delete your information, it's just a BUSINESS $-)

HOWEVER you can damage your DATA by yourself if you try to DECRYPT by any other software, without OUR SPECIFIC ENCRYPTION KEY !!!

Also, all of your sensitive and private information were gathered and if you decide NOT to pay,
we will upload it for public view !

**********How to get back your files ?******

To decrypt all your files and data you have to pay for the encryption KEY :

BTC wallet for payment: 1E6EjTqYPHLj1uovPKKRXzMpPCcpAcVuiU
Amount to pay (in Bitcoin): 60

**********How much time you have to pay?**********

* You should get in contact with us within 2 days after you noticed the encryption to get a better price.

* The price would be increased by 100% (double price) after 14 Days if there is no contact made.

* The key would be completely erased in 21 day if there is no contact made or no deal made.
Some sensetive information stolen from the file servers would be uploaded in public or to re-seller.

***********What if files can't be restored ?******

To prove that we really can decrypt your data, we will decrypt one of your locked files ! Just send it to us and you will get it back FOR FREE.

The price for the decryptor is based on the network size, number of employees, annual revenue. Please feel free to contact us for amount of BTC that should be paid.

****

! IF you don't know how to get bitcoins, we will give you advise how to exchange the money.

!!
! HERE IS THE SIMPLE MANUAL HOW TO GET CONTCAT WITH US !
!!

1) Go to the official website of TOX messenger (https://tox.chat/download.html)

2) Download and install qTOX on your PC, choose the platform (Windows, OS X, Linux, etc.)

3) Open messenger, click "New Profile" and create profile.

4) Click "Add friends" button and search our contact
7D509C5BB14B1B8CB0A3338EEA9707AD31075868CB9515B17C4C0EC6A0CCCA750CA81606900D

5) For identification, send to our support data from ---RAGNAR SECRET---

IMPORTANT ! IF for some reasons you CAN'T CONTACT us in qTOX, here is our reserve mailbox (hello_psecu@protonmail.com) send a message with a data from ---RAGNAR SECRET---

WARNING!

-Do not try to decrypt files with any third-party software (it will be damaged permanently)
-Do not reinstall your OS, this can lead to complete data loss and files cannot be decrypted. NEVER!
-Your SECRET KEY for decryption is on our server, but it will not be stored forever. DO NOT WASTE TIME !

****

---RAGNAR SECRET---
MmE2RjY2N2YwNUZlYmRERjNhZGY4MWY0Y0NiMUEwNEIwRkYyQUZhNDE5QjEwNzYzODhGZjE2QWM5ZGFEYzEwY
g==
---RAGNAR SECRET---
****

B

Appendix B

The ransom note of strain 2 has been modified for publication so that it does not list the URLs for the Ragnar Locker DLS.

****

HELLO CMA-CGM.com !
IF YOU ARE READING THIS, IT'S MEAN YOUR DATA WAS ENCRYPTED AND YOU SENSITIVE PRIVATE INFORMATION WAS STOLEN!
READ CAREFULLY THE WHOLE INSTRUCTION NOTES TO AVOID DIFFICULTIES WITH YOUR DATA

by R A G N A R L O C K E R !

****

YOU HAVE TO CONTACT US via LIVE CHAT IMMEDIATELY TO RESOLVE THIS CASE AND MAKE A DEAL
(contact information you will find at the bottom of this notes)

!!!!! WARNING !!!!!

DO NOT Modify, rename, copy or move any files or you can DAMAGE them and decryption will be impossible.
DO NOT Use any third-party or public Decryption software, it also may DAMAGE files.
DO NOT Shutdown or Reset your system, it can DAMAGE files

There is ONLY ONE possible way to get back your files - contact us via LIVE CHAT and pay for the special DECRYPTION KEY !
For your GUARANTEE we will decrypt 2 of your files FOR FREE, to show that it Works.

Don't waste your TIME, the link for contact us will be deleted if there is no contact made in closest time and you will NEVER restore your DATA.
!!! HOWEVER if you will contact us within 2 day since get penetrated - you can get a very SPECIAL PRICE.

! WARNING !
Whole your network was fully COMPROMISED!

We has BREACHED your security perimeter and DOWNLOADED more than 3 TB of your PRIVATE SENSITIVE Data, including your: Accounting, Financial, Confidential and/or Proprietary Business information, Confidential Contracts, Non-Disclosure Agreements, Administrators directories, SQL Databases and etc.!
Also we have access to Corporate Correspondence, Personal information about your clients such as Social Security Numbers and even more about your partners and your staff.

- There are some screenshots just as a proofs of what we got on you. (you can find more on Leak Page)
Screenshots:
http://prnt.sc/uopg62
http://prnt.sc/uopga3
http://prnt.sc/uopgd3
http://prnt.sc/uopggl
http://prnt.sc/uopgja
http://prnt.sc/uopgmq
http://prnt.sc/uopgpt
http://prnt.sc/uopgts

Also you can check out our previous cases on our Leak Site:
http://xxxxxxxxxx.onion
www.xxxxxxxxx.top

Whole data that gathered from your private files and directories could be published in MASS MEDIA for BREAKING NEWS!
Yours partners, clients and investors would be notified about LEAK, the consequences will have a DISASTROUS effect on your company's reputation!

However if we make a Deal everything would be kept in Secret and all your data will be Restored, so it is much cheaper and easier way for you than lawsuits expenses.

You can take a look for some more examples of what we have, right now it's a private, temporary and hidden page, but it could become permanent and accessable for Public View if you decide NOT pay.
Use Tor Browser to open the link: http://xxxxxxxxxxxx.onion/?xxxxxxx
To view the page's content use password: GD4oXzO0K1

==========

! HERE IS THE SIMPLE MANUAL HOW TO GET CONTACT WITH US VIA LIVE CHAT !
!!

a) Download and install TOR browser from this site : https://torproject.org
b) For contact us via LIVE CHAT open our website :
http://xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.onion/client/?64eCC93FABA3A
AcE8Df0D1d4EBD9Fb04deE44eC111c401D2bccEA237a5454193
c) To visit our NEWS LEAK BLOG with your data, open this website :
http://xxxxxxxxxxxxxxx.onion/?xxxxxxxx (password: GD4oXzO0K1)
d) If Tor is restricted in your area, use VPN

When you open LIVE CHAT website follow rules :

Follow the instructions on the website.
At the top you will find CHAT tab.
Send your message there and wait for response (we are not online 24/7, So you have to wait for your turn).

****

---BEGIN RAGN KEY---
NjRlQ0M5M0ZBQkEzQUFjRThEZjBEMWQ0RUJEOUZiMDRkZUU0NGVDMTExYzQwMUQyYmNjRUEyMzdhNTQ1NDE5Mw==
---END RAGN KEY---

****

C

Appendix C

Hybrid Analysis of the Ragnar Locker Ransomware

1st Marthe Brendefur
Faculty of Applied Computing and Technology
Noroff University College
Kristiansand, Norway
marthe.brendefur@stud.noroff.no

***Abstract*—Ransomware is one of the most prominent and costly threats enterprises face. Attacks are estimated to cost organisations more than 20 billion dollars globally in 2021 [1]. The rapid development of malicious software and custom attacks requires the defensive side to evolve continuously. When an attack occurs, digital forensics techniques are applied to locate, gather, and preserve information relevant to an investigation. The Hybrid Malware Framework was developed to facilitate a holistic approach to analyse ransomware forensically [2]. This paper details the hybrid analysis of a Ragnar Locker strain. Ragnar Locker is a ransomware family that was first spotted in the wild in December 2019 and has since targeted several large companies. The attackers use an extract and extort tactic, meaning that they steal the victim's data to add extra pressure on the company. Findings include termination criteria, files exempt from encryption, forensic artefacts, and that the strain does not communicate over the internet. The uncovered information can be used to improve the defences against future Ragnar Locker attacks.**

***Index Terms*—Ragnar Locker, Ransomware, Hybrid Analysis, Ransomware Forensics, Malware Analysis, Static Analysis, Dynamic Analysis, MITRE ATT&CK**

I. Introduction

Ransomware is a form of malicious software that denies users access to their data. The name derives from the attackers' practice of demanding a ransom to be paid to decrypt the data. According to Purplesec [1], approximately 187.9 million ransomware attacks occurred in 2019. Ransomware has been the most significant cyber threat since 2016, much due to the high attack frequency and the damage it inflicts upon victims.

The Ragnar Locker ransomware family is used in big game hunting (BGH) operations targeting the energy, shipping - and travel sectors. The threat actor behind Ragnar Locker often utilises weaknesses in IT supply chains and managed service providers (MSPs) in their operations. The attackers, for instance, leverage MSP solutions to get access to networks, bypass security solutions, and spread their attacks. Moreover, the group pioneered a technique where ransomware is deployed inside a virtual machine on the victim system to bypass security mechanisms. Other prominent cybercriminal groups later adopted the technique. The encryption of the target system signifies the last stage of the intrusion. Exfiltrated data is then used as extra pressure to make the victim pay the ransom demand [3]. If the demand is not met, the threat actors publish proof of their intrusion on a designated data leak site (DLS). Ragnar Locker's modus operandi makes it suitable for analysis to uncover trending techniques applied by ransomware operators.

The strain will be analysed using the multi-disciplinary Hybridised Malware Framework (HMF) developed for malware forensics by Schmitt [2]. The framework concatenates existing methods for reverse-engineering malicious software with forensic examination techniques. The key element added by the HMF is network analysis, which helps map what and how a malware strain communicates. Investigating Ragnar Locker using a hybrid approach will uncover more of the strains' traits than what is possible with just using one technique.

Extracted Indicators of Compromise (IOC) and forensic artefacts can help security professionals detect and mitigate the threat posed by Ragnar Locker. Such findings can be used to write YARA rules and generate an overview of the adversary's tactics, techniques, and procedures (TTP). Custom ransomware often bypasses hash signature-based detection mechanisms, necessitating agile security solutions and defence-in-depth. By eliciting behavioural traits, one can distinguish actions performed by the ransomware and incorporate this knowledge into the defence.

The paper is structured as follows: Relevant research regarding ransomware and Ragnar Locker is presented under the Related Work-section. The methods used for analysing Ragnar locker is detailed in the Data Gathering-section, and the outcome is presented in Results. The Conclusion summarises the paper.

II. Related Work

Comprehensive ransomware attack analyses are seldomly offered freely online as breaches and threat actor capabilities are considered sensitive. However, some cursory analyses of dominating threat actors and their TTPs exist. Such a group operates Ragnar Locker. The group have previously been affiliated with the now-defunct Maze Cartel, and Ragnar Locker's achievements have been featured on MountLocker's DLS. Despite these affiliations, there are no indicators that the Ragnar Locker group cooperates with access brokers [4]. The group likely facilitate and conduct all stages of an operation themselves. Ragnar Locker has targeted several profiled organisations through 2020. The cybercriminals publicly extort their non-paying victims on their DLS "Wall of Shame" [3]. Ragnar Locker is classified as an advanced ransomware due to how it is coded and the techniques it employ [5]. Ragnar Locker has a somewhat unique code structure and implement

obfuscation and evasion techniques. The capabilities of a strain reflect the operational maturity. Advanced ransomware often has a covert and overt phase [6]. Attackers aim to stay undetected while moving laterally and exfiltrating data. The visibility in the overt phase is caused by the encryption of data, which, together with the ransom demand, comprise the main characteristics of this form of malicious software [6].

A. Attack Vectors

An attack vector is the method used to gain unauthorised access to a system [7]. Threat actors can use several approaches to launch an attack, ranging from exploits and flaws to physical measures [7]. A significant amount of attacks start with some form of social engineering [8]. Social engineering is when a threat actor exploits human nature, often by deceiving the victim to perform an action or reveal information [8]. The most common method is phishing, which is when an adversary gets an unsuspecting person to click on a malicious link or document through an email. However, operations where threat actors gained access through technical exploits increased drastically through 2020. Such operations consist of scanning the target for vulnerabilities, gathering information about the technical structure, and exploiting the weaknesses [7]. One feature commonly exploited is the Remote Desktop Protocol (RDP). RDP enables a device to be connected to remotely. After the initial compromise, threat actors will try to establish persistence on the system.

B. Attack Infrastructure

Threat groups must have a suitable infrastructure in place to facilitate extensive attacks [9]. Command-and-Control servers (C2 or C&C) are core elements in such a setup. The C2 structure generally consists of numerous servers situated at different locations, and threat actors tend to reuse much of their established infrastructure in different attacks. C2s have several use-cases, such as temporary storage of stolen data, communication with beacons in breached environments, key generation, and masking the attacker's location [9]. The attack infrastructure is the backbone of most large operations.

C. Analysis Methods

The practice with hybrid analysis arose to interpret better how malicious software operates on a system. The hybrid analysis combines the strengths of static and dynamic analysis techniques [10]. The hybrid approaches involve additional focus areas, such as memory forensics and network analysis.

1) Analysis Framework: Eskandari, Khorshidpour, and Hashemi [11] developed a technique where the dynamic analysis is performed first to map the actual Application Programming Interface (API) calls made. The extracted results from the dynamic analysis are combined and further analysed using static methods. One of this approach's main benefits is that it minimises time spent on dynamic examination and bypasses the challenges with packing. Code that is packed will exist in a compressed state while stored and extract itself in memory once executed [12]. However, most approaches to hybrid analysis start with static analysis. The methodology applied by the HMF is to start with static analysis, followed by the dynamic, network, and forensic analyses [2]. To summarise, the uncovered behavioural traits of the analysed species are plotted into the MITRE ATT&CK framework. MITRE have indexed offensive techniques and assigned them an ID. Applying standardised definitions is convenient when generating overviews and comparing TTPs.

2) Static Analysis: Static analysis is an umbrella term for techniques that inspect malware without executing it. Static analysis techniques are useful for examining the source code, binary strings, Windows Portable Executable (PE) files, and the execution paths of the malicious software [10]. One of the main benefits of static analysis is that it is relatively safe, as the malicious program is never executed [10]. It can also be reasonably quick, which is convenient during triage. However, if the code is packed or obfuscated, an in-depth analysis using static techniques will be complicated and time-consuming.

3) Dynamic Analysis: Dynamic analysis is performed by executing programs and analysing their behaviour at runtime. Ransomware is commonly run and monitored in a controlled environment, such as a sandbox. One of the main benefits of dynamic analysis is that the researcher can observe how the strain interacts with the system, such as API calls made and behaviour in memory [12]. Many ransomware strains are aware of their environment. If such a strain detects that it is run in a sandbox or debugger, it will likely refrain from executing the malicious payload [12].

4) Forensics Analysis: A digital forensic examination is a process that uses science and technology to assess digital objects to answer questions about events that occurred [13]. Artefacts that exist due to malicious software strains are often concatenated into a list with identifiable traits, or Indicators of Compromise (IOC). IOCs are a subtype artefact; an artefact is a piece of forensic data related to an event, while an IOC is a forensic item directly related to the threat [14]. Collecting, systemising, and implementing IOCs is a part of the threat intelligence cycle.

D. Existing Analysis of Ragnar Locker

Examinations of completed attacks have shown that the Ragnar Locker operators gain access to a targeted entity's network through RDP and MSP exploits. The group acquires and exfiltrates the targets data before manually deploying the Ragnar Locker payload on the victim's domain [3].

Ragnar Locker will terminate itself on systems where the strings in `LCIDLOCALE\_SYSTEM\_DEFAULT` matches that of a language used in the Commonwealth of Independent States (CIS) countries [3]. After checking the language settings, Ragnar Locker starts the encryption process. The ransomware enumerates directories on the victim's system and copies the ransom note named "RGNR_⟨ID⟩.txt" to all directories except for system directories and web browsers, which are ignored [3]. The threat actors still need the operating system (OS) to function so the victim can access a web browser and pay the ransom. The files are encrypted using an RSA 2048-bit

public key, and "_RAGNAR_" is added as a footer within the files [15]. At this stage, the extension of the encrypted files has changed to ".ragnar_⟨ID⟩", where ⟨ID⟩ represent an eight-digit victim ID [15]. Tavares [3] found that the Ragnar Locker strain did not check if a file had already been encrypted, so if invoked again, the file would be encrypted a second time. However, the analysis performed by Blaze Information Security [15] contradicts this, stating that Ragnar Locker will not encrypt a file if the footer "_RAGNAR_" is detected in it. As the algorithm used will be the same in both instances, the only consequence would be that encrypted data must be decrypted an equal number of times.

III. Data Gathering

Due to the economic and operational costs of having data encrypted, it is imperative that ransomware is detected at an early stage. By examining a Ragnar Locker strain using the HMF, this paper aims to uncover traits that can be used to profile the Ragnar Locker ransomware family in a Windows 10 environment. The analysed strain has a SHA-256 value of `9bdd7f965d1c67396afb0a84c78b4d12118ff377db7efdca4a1340933120f376`. The analysis will be conducted on a Windows 10 virtual machine. To account for Ragnar Locker's reaction to a closed environment, network connectivity will be simulated to overcome the strains' evasion techniques. However, none of the existing public analyses of the stain suggests that it communicates over the network. The strain will first be inspected using static analysis techniques, followed by dynamic, network, and forensic analyses.

IV. Results

The strain does not appear to be masquerading as legitimate software, nor claim to be anything other than Ragnar Locker. As it does not hide or attempt to lure a victim into executing it, this may indicate that the operators manually deploy Ragnar Locker on the compromised system. According to the DLS, the operators view themselves as penetration testers and bug-hunters. The group claims to search for weaknesses in corporate networks and are willing to provide penetration reports to victims who pay the ransom. They also offer to help fix the issues. Such a mindset corresponds with the emerging trend of operators manually conducting some of the attack procedures.

A. Static Analysis

The static analysis was performed using single-purpose tools to determine PE information, entropy, and strings. The x32dbg debugger and the IDA Freeware disassembler were used to analyse the strain's structure and functions at the assembly level.

1) PE Structure: According to the file header, the strain was compiled at 16:36:20 Friday 31. January 2020. The executable is a 32-bit PE file, indicating that it is created for 32-bit processors. Although 64-bit processors are more common for the Windows 10 OS, using the 32-bit architecture ensures better backwards combability. 32-bit programs also use fewer memory resources, making them suitable for computers with 4GB of RAM or less. The use of 32-bit indicates that the developers designed Ragnar Locker to infect a wide range of OS versions. The strain is compressed when at rest and unpacks itself in memory when executed. Profile-guided Optimisation (PGO), a C++ compiler from Intel, is likely the compression method used. PGO compresses files in three steps and is suitable for processor-intensive applications with few variations in applied data sets.

2) Windows API Class: The Ragnar Locker strain imports six native Windows libraries, as seen in Table I. The Windows application programming interfaces (APIs) are standard for Windows machines. Windows APIs are used to invoke lower-level functions. Relying on these API Classes help Ragnar Locker blend with the environment and reduce the need to import tools.

TABLE I
DLL libraries used by Ragnar Locker.

Library	DLLs	Description
crypt32.dll	4	Crypto API32. Implements Certificate and Cryptographic functions.
kernel32.dll	57	Windows NT BASE API Client DLL. Handles memory management, I/O operations, and interrupts.
user32.dll	2	Multi-User Windows USER API Client DLL. Handles the user preferences.
advapi32.dl	20	Advanced Windows 32 Base API. Handles restarts and shutting down the system, the Windows registry, user accounts and Windows services.
shell32.dll	1	Windows Shell Common DLL. Is used when opening web pages and files.
shlwapi.dll	3	Shell Light Library. Contains functions for URL paths, Windows registry, and colour settings.

3) Entropy: Entropy analysis can help determine whether an executable file is compressed or obfuscated. When analysing digital files, the entropy is measured by calculating the randomness of bytes. The result is measured on a scale from 0 to 8, where 0 indicates low entropy and 8 high entropy. Ordinary files usually have an entropy of around 5. By examining the Ragnar Locker strain using PeStudio and Detect It Easy, it becomes evident that parts of the strain are obfuscated or compressed. The `.text`-section has an entropy of 6.521, and the `.keys`-section has an entropy of 6.442. The `.text`-section contains the program code, and the high entropy indicates that the code is packed. As for the `.keys-section`, it is expected that the entropy is high, as it contains encryption data. The sections and their entropy is listed in Table II.

4) Termination Criteria: The strain will abort its execution based on the computer locale. Ragnar Locker uses the `GetLocaleInfoW()` function to determine the OS language settings. If it detects that it is in an environment that matches one of the pre-defined countries, it will obtain its own process

TABLE II
ENTROPY OF THE DIFFERENT SECTIONS OF RAGNAR LOCKER.

Binary Entropy		
Section	**Entropy**	**Content**
.text	6.521	Executable code
.rdata	5.354	Imports and exports
.data	n/a	Global data
.keys	6.442	Contains encrypted stings
.rsrc	4.702	Strings, icons, and images
.reloc	4.810	Base relocation table

and terminate itself. The OS language settings that will make Ragnar Locker stop executing are presented in Table III.

TABLE III
OS LANGUAGES THAT WILL CAUSE RAGNAR LOCKER TO TERMINATE.

Language settings that will cause termination		
Azerbaijani	Armenian	Belorussian
Kazakh	Kyrgyz	Moldovian
Tajik	Russian	Turkmen
Uzbek	Ukrainian	

5) File Encryption: The Ragnar Locker strain will iterate through the volumes and drives on the computer. Logical volumes are also mapped and assigned a drive letter. The files that are to be encrypted is added to the memory stack. The `.keys`-section is called several times to decrypt the strings that reference the files exempt encryption. Folders, files, and extensions refrained from encryption is listed in Table IV.

TABLE IV
FOLDERS, FILES, AND EXTENSIONS EXEMPT ENCRYPTION.

Folder exceptions		
Windows	Windows.old	Tor browser
Internet Explorer	Google	Opera
Opera Software	Mozilla	Mozilla Firefox
§Recycle.Bin	ProgramData	
File exceptions		
autorun.inf	boot.ini	bootfont.bin
bootsect.bak	bootmgr	bootmgr.efi
desktop.ini	iconcache.db	ntldr
ntuser.dat	ntuser.dat.log	ntuser.ini
Extension exceptions		
.db	.sys	.dll
.lnk	.msi	.drv

Ragnar Locker will also utilise strings to search for certain running processes. Processes that match these strings will be terminated. This behaviour is likely a security mechanism to avoid detection, as the strings are related to several known MSP, anti-virus (AV), and security solution providers. The strings called are listed in Table V.

TABLE V
PROCESSES TERMINATED BY RAGNAR LOCKER.

Process termination strings		
vss	sql	memtas
mepocs	sophos	veeam
backup	pulseway	logme
logmein	connectwise	splashtop
kaseya		

Once Ragnar Locker finishes enumerating the OS and adds file references to the stack, the encryption process will begin. The strain has a hardcoded public key, and the private key is never present in the system. Ragnar Locker copies the ransom note named "RGNR_ ⟨ID⟩.txt" to all directories except for the ignored folders. The encrypted files get the extension ".ragnar_ ⟨ID⟩". The ID is a hash of the computers NETBIOS name [16]. Once the encryption process is finished, all running processes are terminated, and a notepad window opens to display the ransom note. The ransom note is hardcoded in the strain. The analysed strain appears to be tailor-made for PSE Credit Union. PSE Credit Union is not mentioned on the DLS, nor have the company made a public statement or implied that they have suffered a ransomware attack.

B. Dynamic Analysis

The strain has been run in both a local virtual environment and commercial sandboxes to observe how it behaves when executed. The sandboxes used is VirusTotal, Joe Sandbox, and AnyRun. The findings from the static analysis were validated when the sample was executed locally. One function offered by VirusTotal is to check the detection rate of the strain against a large base of AV software. 64 out of 71 AV engines flagged it as a malicious file the 31.01.2021, giving it a 90.14% detection rate one year after its creation.

1) Initiated processes: The Ragnar Locker strain initiates three processes: `vssadmin.exe`, `wmic.exe`, and `notepad.exe`. The processes are started to perform actions on the system that Ragnar Locker cannot conduct independently.

`Vssadmin.exe` is the main Windows function that administers the volume shadow copies. In `vssadmin.exe`, the command `vssadmin delete shadows /all /quiet` is run. This command deletes all shadow copies on systems running Windows 8.1 and Windows Server 2008 and newer. The volume shadow copies are deleted to prevent a user from recovering information by rolling the system back to a known good version. Ragnar Locker launches `wmic.exe` in addition to `vssadmin.exe`. `Wmic.exe` is a command-line utility used to access the Windows Management Instrumentation and can only be used by the local system administrator. When the strain has obtained access to the command-line, it runs the command `shadowcopy delete`. As some systems have mechanisms in place to secure the shadow copies, this double function is likely a redundancy mechanism for Ragnar Locker. When initiating the same process using two different methods, the threat actors increase their chance to succeed.

Notepad.exe is initiated after the system is encrypted and is used to display the ransom note.

C. Network Analysis

The examined strain does not appear to communicate with external entities. The static analysis found that the strain does not contain or call any web-related APIs. Executing Ragnar Locker whilst monitoring the internet activity confirmed that no attempts to connect to the internet was made. Wireshark was used to listen to the network traffic from the infected machine. To circumvent potential sleep-mechanisms, the network was monitored for three hours. Analyses from the web-based sandboxes further confirmed this finding. The lack of network connectivity further supports the hypothesis of Ragnar Locker being manually deployed by the attackers. The sole purpose of the ransomware is to encrypt the system and inform the victim of the compromise.

D. Forensic Artefacts

When conducting a digital forensics examination of a system, it is essential to note that each interaction with the system will change data on it. As Ragnar Locker is a ransomware strain, it will produce many artefacts, both in the registry and the file system. The number of artefacts will depend on the system.

1) Registry Analysis: Ragnar Locker adds 17 keys to the Windows registry. It adds the `HKLM\SOFTWARE\Microsoft\IdentityCRL\ThrottleCache`-key and appends five accounts to it. The ThrottleCashe is used to store request counts. All the accounts added have the SID S-1-5-18, which originally belongs to the local system's service accounts. Throttles are akin to permissions, so these actions indicate that the strain tries to change the privilege of the accounts it is using. It then adds the provisioning package `UbYfyvbG8E+B6zqO.0` to the `HKLM\SOFTWARE\Microsoft\Provisioning\Sessions`-key. The provisioning packages contains settings used to configure the device. Adding a provisioning package in this location indicates that the strain will change runtime settings on applications, which allow the ransomware to customise its environment. The remaining ten keys are added with the SID S-1-5-21-⟨*domain*⟩-1000, which is a user account on the domain. Five of the keys is associated to the `Microsoft\Speech_OneCore\Recognizers`, and the remaining five values edit the `ApplicationViewManagement\W32`. Three registry keys are deleted, the old `Provisioning\Sessions`-key and two keys related to system updates. The Ragnar Locker strain also adds several values to existing keys. One significant value is stored in the newly created `Provisioning\Sessions`-key. The `NextSessionID` is the same as the key added previously, and the `BeginTime` within that key holds the timestamp of when Ragnar Locker was executed. Knowing when the strain was executed will help build a timeline of events.

E. MITRE ATT&CK

Ragnar Locker uses legitimate user accounts to gain access and maintain persistence on systems. Using legitimate accounts also help the attackers evade defence and detection mechanisms, as the activity is less likely to stand out. To further mask their activity as benign, the attackers use ordinary user accounts and do not try to escalate their privileges to a level that would attract attention. The strain makes use of Windows APIs when executing behaviours. These will largely blend with benign API calls made. Ragnar Locker monitors good API and DLL calls and spawns' processes from them. It may also try to inject arbitrary code in the Windows Explorer process. Process spawning and injection are used to conserve a low profile on the system and escalate privileges. To maintain persistence on the system, Ragnar Locker is equipped with functionalities to infect the boot sector. Such functions are often referred to as bootkits. Bootkits reside in the Master Boot Record, on the layer below the file system, making them hard to discover and remove. The strain is aware of its environment, and queries disk information and compare the user and computer. These actions are common to use when determining whether the system is real or virtual. It also run the `GetProcessHeap` function to check if it is run in a debugger. Ragnar Locker deletes the system shadow copies to hinder the system from being recovered to a known good state. The process of assessing files for encryption is automated, which means that MITRE flags the *Automated Collection* and *Data from Local System*-tags. The strain contains a public RSA-key and calls the Windows Crypto API but shows no sign of attempting to connect to a C2 when run. If Ragnar Locker runs uninterrupted on a system, and its termination criterion is not triggered, Ragnar Locker will encrypt the file system. Table VI presents a high-level overview of the tactics and techniques as classified by MITRE ATT&CK.

V. Conclusion and Future Work

The growth of agile and tailored ransomware is likely to continue, necessitating procedures that will facilitate thorough analysis of detected strains. Openness and sharing of threat intelligence are essential for building technical defences and mitigate the risk before breaches occur.

The Hybrid Malware Framework coalesce existing methods for reverse engineering malicious code into one forensic oriented approach. This paper is an extension of the testing and validation of this framework. This paper details the analysis of an early strain from the Ragnar Locker ransomware family. The threat actor behind Ragnar Locker tailor their attacks to each victim and are actively present when conducting lateral movement and data exfiltration. The number of manual processes applied to each attack is one factor that leads to exorbitant ransom demands. The ransomware is deployed when the attackers decide to end the engagement.

The strain was analysed using static analysis, dynamic analysis, and network analysis techniques. The Ragnar Locker strain appears to be manually executed on the system, supporting the thesis of the attackers being hands-on present during

TABLE VI
MITRE ATT&CK OVERVIEW

Initial Access	Execution	Persistence	Privilege Escalation	Defence Evasion	Discovery	Collection	C2	Impact
Valid Accounts	Native API	Bootkit	Valid Accounts	Bootkit	Security Software Discovery	Automated Collection	Encrypted Channel	Data Encrypted for Impact
		Valid Accounts	Access Token Manipulation	File Deletion	Sandbox Evasion	Data from Local System		
			Process Injection	Valid Accounts	System Service Discovery	Archive Collected Data		
				Sandbox Evasion	System Information Discovery			
				Access Token Manipulation	Process Discovery			
				Process Injection	Account Discovery			
					File and Directory Discovery			
					System Owner/User Discovery			

all stages of the attack. It uses the standard Windows APIs and DLLs and does not initiate any network connections. Several folders, files, and file extensions are exempt from encryption. The reason for this is that the attackers need the computer to function to enable the victim to pay the ransom. If Ragnar Locker detects that the OS uses one of the CIS-countries' language settings, it will terminate itself.

Systematically analysing ransomware strains' key traits enables researchers to compare results, both within one family and against other species. Such an approach will facilitate the construction of solid and effective prevention mechanisms. By uncovering similarities between strains and families, one can possibly predict behaviour characteristics in future strains. Many security solutions rely on signature-based detection, which only enables them to detect known threats.

The analysis presented in this paper present some of Ragnar Locker's traits. The uncovered technical indicators can be used to generate YARA rules. Future work should analyse newer strains of Ragnar Locker using the same framework. This would enable the security community to detail the evolution of the Ragnar Locker ransomware family.

REFERENCES

[1] Purplesec, *2020 Ransomware Statistics, Data, & Trends*, 2021.

[2] V. Schmitt, "A comparative study of CERBER, MAKTUB and LOCKY," Ph.D. dissertation, Grahamstown, Jan. 2019, pp. 1–167.

[3] P. Tavares, *Ragnar Locker malware: what it is, how it works and how to prevent it*, Jun. 2020.

[4] CrowdStrike, "Global Threat Report 2021," *CrowdStrike*, p. 75, 2021.

[5] N. Lord, *What is Advanced Malware?* Sep. 2018.

[6] A. L. Young and M. Yung, "On Ransomware and Envisioning the Enemy of Tomorrow," *Computer*, vol. 50, no. 11, pp. 82–85, Nov. 2017.

[7] C. B. Simmons, S. G. Shiva, H. Bedi, and D. Dasgupta, in *9th Annual Symposium on Information Assurance*, University of Memphis, New York, 2014.

[8] A. Zimba and M. Chishimba, "Understanding the Evolution of Ransomware: Paradigm Shifts in Attack Structures," *Computer Network and Information Security*, vol. 1, pp. 26–39, 2019.

[9] T. Steffens, *Attribution of Advanced Persistent Threats*. Berlin, Heidelberg, 2020.

[10] H. S. Galal, Y. B. Mahdy, and M. A. Atiea, "Behavior-based features model for malware detection," vol. 12, no. 2, pp. 59–67, May 2016.

[11] M. Eskandari, Z. Khorshidpour, and S. Hashemi, "HDM-Analyser: A hybrid analysis approach based on data mining techniques for malware detection," vol. 9, no. 2, pp. 77–93, May 2013.

[12] U. Bayer, A. Moser, C. Kruegel, and E. Kirda, "Dynamic analysis of malicious code," 1, vol. 2, Aug. 2006, pp. 67–77.

[13] M. Abulaish and N. A. H. Haldar, *Digital Forensics and Forensic Investigations*. Apr. 2020.

[14] D. Greten, *Indicators of Compromise (IOCs) and Artifacts: What's the Difference?* Jun. 2020.

[15] Blaze Information Security, *Dissecting Ragnar Locker: The Case Of EDP*, Jul. 2020.

[16] FBI, "Indicators of Compromise Associated with Ragnar Locker Ransomware," Tech. Rep., 2020.

D

Appendix D

D.1 Artefacts and Resources

D.1.1 The Complete Data Collection

A complete collection of analysis data can be found at GitHub:
https://github.com/Powsnow/Ragnar_Locker

The GitHub repository includes:

- Strain 1
- Strain 2
- Extracted strings
- PCAP files
- Process Monitor files
- Memory dumps
- Ransom notes
- A YARA rule
- RegShot comparison files
- Encryption comparison files
- Links to external resources
- MITRE ATT&CK Navigator files
- Mini Conference presentation
- The Short Paper
- A LaTeX package with the Final Paper

The password for the folders holding the ransomware strains is *infected*. The ransom note of strain 2 has been modified for publication so that it does not list the URLs for the Ragnar Locker DLS.

As a backup, the project is available at Google Drive until July 2021.

D.1.2 Ragnar Locker Download Sources

The links have been shortened to fit on one line.

Strain 1: https://bit.ly/3tEdm6N

Strain 2: https://bit.ly/3uJDoXz

D.1.3 Analysis Sources

Strain 1

VirusTotal: https://bit.ly/2QdnplI

ANY.RUN: https://bit.ly/3ocGn8u

JoeSandbox Cloud: https://bit.ly/3fh7bAE

Strain 2

VirusTotal: https://bit.ly/3odCMqL

ANY.RUN: https://bit.ly/3obXTdb

JoeSandbox Cloud: https://bit.ly/3hjNM4E

Word count metrics

NUC Final Degree Project Word Count:

Total Sum count: 17636 Words in text: 17049 Words in headers: 236 Words outside text (captions, etc.): 342 Number of headers: 71 Number of floats/tables/figures: 41 Number of math inlines: 9 Number of math displayed: 0
(errors:21) NOTE: References are excluded.