Carnegie Mellon University
Software Engineering Institute

CURRENT RANSOMWARE THREATS

Marisa Midler Kyle O'Meara, and Alexandra Parisi

May 2020

Acknowledgments

The authors wish to acknowledge the contributions of National Cyber-Forensics and Training Alliance (NCFTA) for providing valuable insight to the current ransomware landscape and top 10 trending ransomware families.

Executive Summary

Ransomware continues to be a grave security threat to both organizations and individual users. The increased sophistication in ransomware design provides enhanced accessibility and distribution capabilities that enable attackers of all types to employ this malicious tool. This report discusses ransomware, including an explanation of its design, distribution, execution, and business model. Additionally, the report provides a detailed discussion of encryption methods and runtime activities, as well as indicators that are useful in their detection and mitigation.

Ransomware has evolved into a sophisticated tool that is usable by even non-technical persons and has multiple variants offered as Ransomware as a Service (RaaS). RaaS decreases the risk for ransomware authors, since they do not perform attacks, and reduces the affiliates' cost to mount attacks. Additionally, as of 2019, some ransomware families have started threatening public disclosure of a victim's sensitive data if they do not pay a ransom and are following through with the threat.

Ransomware uses strong encryption, making decryption without a key or implementation flaws practically impossible. The success of initial ransomware infections is primarily attributed to the following:

- failures in email filtering
- users who are unaware and susceptible to opening malicious email attachments
- unpatched systems and applications that are vulnerable to exploits
- operating systems that lack proactive heuristics-based monitoring

This report recommends both proactive and reactive approaches that help avoid having to pay a ransom and minimize the loss of data. The best way to mitigate against ransomware is to sustain frequent offline backups of all data, which minimizes data loss and increases the likelihood of not

paying ransomware operators. Additionally, to mitigate the data breach threat from data exfiltration, organizations should employ data encryption on data at rest.

Ransomware attacks can take down critical systems, and currently these variants are targeting government agencies and the healthcare, education, and transportation industries. Ransomware will continue to be a problem for the unforeseeable future and, with the advent of RaaS, the threat landscape is likely to expand. Detection and mitigation of ransomware is possible by making frequent offline backups, conducting ongoing user awareness training, and applying system and network security enhancements.

Table of Contents

List of Figures

List of Tables

1 Introduction

Ransomware is one of the most profitable cybercrime schemes in use today. Simply stated, ransomware locks access to a victim's data and holds it hostage in return for money. Ransomware attacks occur virtually every day, affecting victims ranging from large organizations to individual computer users. The majority of victims end up paying the ransom in order to recuperate their data. Underlying the success in ransomware schemes are several important security deficiencies:

- Email filtering fails to identify and block incoming malicious emails (e.g., phishing and malicious spam emails).
- Users lack security awareness in detecting, avoiding, and reporting potentially suspicious emails, leading to the opening and execution of malicious email attachments and allowing malware into the system.
- Current host-based malware detection software is inadequate to keep end users from being victimized and permits the success of ransomware despite the presence of various security measures.

The ransomware threat continues to increase, driven by security deficiencies and quick profitability. Many industries are seeking to better understand the attack landscape to mitigate the threat and prepare to respond if necessary. This report addresses these issues by providing up-to-date information on what ransomware is, how it functions, and what users can do to avoid being a victim of it.

1.1 Definition

The defining characteristics of ransomware are the data encryption and extortion components. Since encrypted data can't be recovered without involving the attacker, it facilitates ransomware to demand, and, in most cases, receive various sums of money. All other characteristics of ransomware are present in other types of malicious code and can be generalized as belonging to the class of malware. As it is typically used, the encryption deployed by ransomware is intended to make the ransom payment the most economical way for victims to recover their data. Unless organizations have prepared a means to recover their data, or the encrypted files have no value, they end up paying the demanded amount of money.

It can be argued that requiring money in the form of a ransom is what defines this class of malware. However, there are other types of malicious code, such as scareware (e.g., FakeAV), that also attempt to lure the user into paying money to receive something in return, such as a clean computer or the removal of annoying pop-up windows. Ransomware is formally categorized as part of the cryptovirology field, which focuses on ways in which cryptography can be used as a component of malware. The Department of Homeland Security (DHS) Cybersecurity and Infrastructure Security Agency (CISA) provides an accurate definition based on these characteristics of ransomware:

> *Ransomware is a type of malicious software, or malware, designed to deny access to a computer system or data until a ransom is paid. (Cybersecurity and Infrastructure Security Agency (CISA) 2020)*

1.2 Recent History

In recent years, ransomware has evolved and expanded due to advancements in related technologies and the lure of potential profits. The attack landscape has shifted from targeting individual users to targeting healthcare, government agencies, universities, and corporations. Ransomware encryption mechanisms have advanced from weak, custom implementations to recognized industry standard encryption algorithms. Communication with command-and-control (C2) servers became a common feature with some ransomware using encrypted and anonymous communication channels. Payment methods have also changed, shifting from wire transfers and prepaid cards to the use of cryptocurrency (e.g., Bitcoin).

As ransomware evolves, it also adapts to the types of data it needs to encrypt. Early ransomware samples primarily encrypted user documents and pictures, while newer ransomware variants focus on network storage, databases, and websites in addition to user files. As users increasingly attempted to recover encrypted data from backups, ransomware adapted to locating and encrypting network-accessible backup storage. In 2019, ransomware started exfiltrating company data, with the intent of publishing it if the affected company does not pay the ransom. In addition to the new data exfiltration threat, some ransomware is now charging two fees: (1) to decrypt the encrypted data and (2) to prevent the publishing of exfiltrated data. These adaptations demonstrate the remarkable ability of ransomware authors to rapidly adjust to the changing landscape.

1.3 Business Model

Ransomware is a highly lucrative criminal scheme motivated by profit and proven to be an effective producer of revenue for cyber criminals. Anyone can be victimized by ransomware; the same ransomware variant can attack businesses and individuals. It is target agnostic and has caused significant impact to several sectors of society.

The existence of cryptocurrency is crucial to the success of ransomware. Bitcoin and other cryptocurrency transactions are largely unregulated by legal authorities; this allows cyber criminals to move ransom payments out of otherwise well-regulated financial environments and into jurisdictions less hostile to the criminals' activities.

Additionally, ransomware is primarily a service-based business. As with all profitable service-based business models, sustainment is achieved by delivering the goods once payment is made. In the case of ransomware, the service provided is the delivery of decryption keys, decryption software, and customer support.

The most successful ransomware schemes excel at providing this service. Ransomware tends to be well organized, providing each victim with unique identifiers, which are then used to deliver the correct decryption keys. The decryption software usually works as expected and all encrypted files tend to be fully restored to their original form. The software is also designed to be easy to use, even by a non-technical person, and is often available in multiple languages.

The creators of the ransomware business model have made the following fundamental assumptions that generally hold true and have facilitated its success:

- **Most users do not routinely back up their data.** The financial success of ransomware clearly illustrates that most victims do not adequately back up their data. If they did, payments would potentially be less frequent. In 2019, Backblaze conducted a survey of 1,858 participants, each owning at least one computer. The survey showed that only 9% of users back up their data daily, 20% never back up their data, and 38% only back up data when they remember to do so. (Bauer 2019)

Backblaze has conducted annual surveys of backup frequencies since 2008. The results from 2008-2019 are shown in Table 1.

Over the twelve-year period, only a small fraction of users backed up data weekly; 19-26% of users performed regular yearly backups; and in the largest category, 20-35% reported that they never backed up their data at all. However, the trend toward more frequent backups continues to slowly improve over time.

Table 1: Backblaze Computer Backup Frequency Survey Data 2008-2019

Frequency	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Daily	6%	6%	8%	6%	10%	10%	9%	8%	8%	9%	6%	9%
Weekly	7%	8%	8%	8%	10%	9%	9%	9%	9%	12%	11%	10%
Monthly	14%	12%	14%	13%	16%	17%	16%	19%	17%	16%	17%	20%
Yearly	20%	20%	21%	20%	19%	22%	22%	23%	25%	26%	26%	25%
Year plus	19%	15%	15%	16%	14%	13%	16%	16%	16%	15%	17%	16%
Never	35%	35%	33%	34%	29%	29%	29%	25%	25%	21%	24%	20%

- **Encryption is mathematically designed to be very difficult to break.** It is well known that the mathematical foundation of encryption relies on the use of large numeric sequences. The purpose of these sequences is to require an unreasonable amount of time to test all possibilities to find a match.
- **Users are susceptible to phishing.** The Verizon 2019 Data Breach Investigations Report found in a study that email attachments were the most common point of entry in cyber incidents. The report noted that when malware installation was detected, over 90% of the malware was distributed by email and that ransomware also uses this infection vector. (Verizon 2019) Over the past few years, user click rates on phishing emails have decreased to 3%. However, research conducted by the Avant Research Group, LLC. indicates that users are more susceptible to phishing attacks through mobile devices due to how users interact with mobile hardware and software. Mobile interfaces provide limited methods to verify emails due to small screen sizes which scale down the information portrayed to the user. Finally, the Avant research study also investigated the multitasking aspect of mobile users' behavior and found it to interfere with users' ability to pay attention to details.
- **Organizations have additional priorities.** Ransomware authors realize that unlike regular users, businesses are more sensitive to losing access to their data, even for a short period of time. While organizations are more likely to backup data frequently, they also rely on their services to be available at all times. Even if ransomware only encrypts a portion of the data in the datacenter and restoring from a backup only takes a day, it still requires the business to go offline temporarily. This weakness often causes the business to pay the ransom to be operational faster and with minimal loss of data. Recently, Jackson County,

GA agencies paid a $400,000 ransom for the decryption key after the Ryuk hackers impacted law enforcement, emergency dispatchers, and the county jails. (Novinson, The 10 Biggest Ransomware Attacks of 2019; 9. Jackson County, Ga. 2019) In a similar incident, government leaders in Lake City, FL paid the Ryuk ransomware hackers $460,000 in Bitcoin to decrypt the city's data. (Novinson, The 10 Biggest Ransomware Attacks of 2019; 8. Lake City, Fla. 2019) Often, the amount of the ransom is perceived as the lowest cost response option, especially compared to the potential loss of future profits and liability concerns.

- **End-user machines connect to networks.** In corporate environments, the end-user laptop or desktop is typically connected to a host of network servers housing large amounts of organizational data belonging to multiple people. Once the user is logged on, the connection to these servers is typically automatic and may not require further authentication for remote read and write privileges. It is therefore easy for ransomware to discover, access, and encrypt network server data once it runs on the user's local machine. This weakness can allow for large amounts of data to be encrypted, increasing the likelihood of receiving payment.

The above fundamental assumptions of the ransomware business model exclude assumptions about computers in general, such as software not being regularly updated and the ability to subvert most security products. These more general assumptions aid in the success of all malware and are not unique to ransomware.

2 Current State

2.1 Known Ransomware Families

Sections 2.1.1 through 2.1.10 describe the top ten trending ransomware variants from January 2018 to present as identified by the National Cyber-Forensics and Training Alliance (NCFTA).

2.1.1 FuxSocy Encryptor

Overview: Discovered by MalwareHunterTeam in October 2019, FuxSocy is similar to the now non-operational Cerber ransomware. (New Jersey Cybersecurity & Communications Integration Cell 2019) Once FuxSocy is on a system, it creates several registry entries to gain administrative privileges. (Carballo 2019) During encryption, FuxSocy encrypts files whose path contains particular strings. (New Jersey Cybersecurity & Communications Integration Cell 2019) Once the system is encrypted, it changes the desktop background notifying the victim of the infection and provides instructions to contact them on the ToxChat messaging app. (Woods 2019) What is unique about FuxSocy is that it does not encrypt entire files. It only partially encrypts the files, corrupting them through a combination of RSA and AES-256. (New Jersey Cybersecurity & Communications Integration Cell 2019)

Infection Vector: FuxSocy's preferred method of infection is phishing emails, that drop a malicious payload primarily via the AppData, Local, and LocalLow directories. In addition to phishing emails, ransomware may be delivered with the help of macro-laced documents and untrustworthy downloads to load malicious scripts and/or trojans or worms and plant ransomware on the device directly. (New Jersey Cybersecurity & Communications Integration Cell 2019)

Encryption used: Combination of RSA and AES-256 encryption

Decryptor available: No public decryptor available

Industries targeted: Information unavailable

Countries targeted: Information unavailable

See Appendix 6.1 for more indicators of compromise (IOC).

2.1.2 GlobeImposter/GlobeImposter 2.0

Overview: Also known as "Fake Globe" because of the way the ransomware mimics the Globe ransomware family, GlobeImposter is commonly distributed by malware spam email campaigns. (New Jersey Cybersecurity & Communications Integration Cell 2019) Most commonly, the adversary sends malicious spam to the victim with a Zip archive containing malware written in JavaScript. Most of the time, GlobeImposter remains almost fully unrecognizable and runs silently during encryption. The ransom note contains a personal infection ID that is required for ransom payment; the note directs the victim to use Tor to contact one of the associated email addresses (see Appendix 6.2) for payment and decryption instructions. (Manuel and Salvio 2019) (Malwarebytes Labs 2017)

Infection Vector/Methodology: GlobeImposter infects Windows machines via a malware spam campaign, using exploits, malicious advertising, false updates, and repacked infected installers.

Encryption used: Combination of RSA-4096 and AES-256 (Malwarebytes Labs 2017) (Neumann and Natvig 2019)

Decryptor available: A decryptor is available for some variants on the No More Ransom website and through the NCFTA (The No More Ransom Project 2019)

Industries targeted: Information unavailable

Countries targeted: United States and Europe (NCFTA 2020) (Manuel and Salvio 2019)

See Appendix 6.2 for more IOC.

2.1.3 LockerGoga

Overview: Discovered in January 2019, LockerGoga is believed to be the product of the FIN6 group. There are three stages to LockerGoga's execution which it switches between based on the given parameters. While executing, LockerGoga changes account passwords, drops ransom notes, and disables interfaces. It uses a parent/child model that accelerates the encryption of a victim's files. The parent process finds files to target and then writes the file paths in the shared memory to communicate to the child processes a list of files to encrypt. (Manuel and Salvio 2019) (Lopez 2019) This ransomware uses administrative rights and may be part of a multipronged attack. (Manuel and Salvio 2019) (NCFTA 2020) LockerGoga has common trends also found in attacks with WannaCry, NotPetya, and SamSam.

Infection Vector: LockerGoga utilizes phishing emails that contain malicious attachments with embedded macros. It also utilizes the Server Message Block (SMB) protocol and Active Directory services through scheduled tasks to spread payload across victim networks as well as modifies all user account passwords. (Manuel and Salvio 2019) (NCFTA 2020) (Neumann and Natvig 2019)

Encryption used: Combination of RSA-4096 and AES-256 (Neumann and Natvig 2019)

Decryptor available: No public decryptor available

Industries targeted: Industrial and Engineering (NCFTA 2020)

Countries targeted: United States and France (Manuel and Salvio 2019) (NCFTA 2020)

See Appendix 6.3 for more IOC.

2.1.4 SamSam

Overview: Discovered in December 2015, SamSam was created for targeted attacks. It does not spread automatically and requires human involvement to run. (Malwarebytes Labs 2018) SamSam can only be launched by the author, or someone who knows the author's password. In November 2018, the U.S. Department of the Treasury's Office of Foreign Access Control and U.S. Department of Justice indicted two Iran-based individuals on the charges of exchanging Bitcoin ransom payments from SamSam ransomware and depositing those payments into Iran-based banks. (Coveware 2019)

Infection Vector: SamSam utilizes vulnerabilities to infect specific organizations and institutions. It does this by execution of brute force of weak passwords, attacks on vulnerable JBoss host servers, Remote Desktop Protocol (RDP) systems, Java-based web servers, and File Transfer Protocol (FTP) servers. (Infradata 2019)

Encryption: RSA-2048 (Boyd 2019)

Decryptor: No public decryptor available (Coveware 2019)

Industries Targeted: Government, transportation, healthcare, and education (Infradata 2019)

Countries Targeted: United States and Netherlands (Cybersecurity and Infrastructure Security Agency (CISA) 2018)

See Appendix 6.4 for more IOC.

2.1.5 MedusaLocker

Overview: MedusaLocker was discovered in September 2019 by MalwareHunterTeam. The delivery method is unconfirmed but malicious payloads have been distributed to victims via phishing and spam emails with an attached link that leads to a malicious website. (MalwareHunterTeam 2019) (Walter, How MedusaLocker Ransomware Aggressively Targets Remote Hosts 2019) It then restarts the LanmanWorkstation service, responsible for creating and holding client-network connections to remote servers over SMB protocol. When this is halted or restarted, MedusaLocker forces configuration changes made into effect. It then targets executables and kills generic products used to conduct analysis and reverse engineering.

Infection Vector: A malicious payload is distributed to the victim through phishing and spam email with an attached link to a malicious website.

Encryption used: Combination of RSA-2048 and AES-256

Decryptor available: No public decryptor available

Industries targeted: Information unavailable

Countries targeted: Information unavailable

See Appendix 6.5 for more IOC.

2.1.6 Ryuk

Overview: Discovered in 2018, Ryuk is considered a modified version of the Hermes ransomware commonly attributed to the North Korean Advanced Persistent Threat (APT) Lazarus Group. Ryuk gains access through different social engineering techniques or an unsecure website. Once on the network, it utilizes TrickBot and Emotet to gain direct access via RDP. TrickBot and Emotet spread using PsExec and/or Group Policy to drop Ryuk, steal sensitive information before the encryption process, and leave the victim network more susceptible to further Ryuk attacks. (Malwarebytes Labs 2019) (Oza 2020) This ransomware targets large organizations and government entities that the actors know will pay substantial amounts of money to decrypt their data.

Infection Vector: Ryuk gains access to a network via a phishing email, unsecure website, or a user clicking on a random popup. The use of TrickBot and Emotet allow the adversary direct access into the victim's network via RDP.

Encryption: Combination of RSA-2048 and AES-256 (Walter, How MedusaLocker Ransomware Aggressively Targets Remote Hosts 2019)

Decryptor: No public decryptor available

Industries Targeted: Retail, health, transportation, and government agencies (Oza 2020)

Countries Targeted: Information unavailable

See Appendix 6.6 for more IOC.

2.1.7 Nemty

Overview: Nemty is a ransomware as a service (RaaS) discovered in August 2019. It utilizes RDP to leverage total control. It is distributed by Trik botnet and RIG exploit kit. (Ilascu, Nemty Ransomware Gets Distribution from RIG Exploit Kit 2019) Once on a system, the file extension ".nemty" is added to encrypted files and leads to instructions for data recovery. (Ilascu, Nemty Ransomware Gets Distribution from RIG Exploit Kit 2019) (GoldSparrow, Nemty Ransomware 2020) The actors behind Nemty are believed to be associated with Gand-Crab and Sodinokibi ransomware families. (Ilascu, Nemty Ransomware Gets Distribution from RIG Exploit Kit 2019)

Infection Vector: Nemty is spread through compromised RDP connections that allow attackers to obtain total control over the process. Recent updates of Nemty have found the ransomware being delivered through a fake PayPal website, and phishing emails (GoldSparrow, Nemty Ransomware 2020) (Ilascu, Nemty Ransomware Now Spreads via Trik Botnet 2019) (Paganini 2020)

Encryption Used: Combination of RSA-2048, RSA-8192, AES-128, and AES-256 (GoldSparrow, Nemty Ransomware 2020)

Decryptor Available: Yes, there is a known decryptor for some variants (The No More Ransom Project 2019)

Industries Targeted: Information unavailable

Countries Targeted: China, South Korea, and United States

See Appendix 6.7 for more IOC.

2.1.8 MegaCortex

Overview: Discovered in May 2019, MegaCortex is a targeted ransomware which is installed via network access through trojans, stolen credentials, and/or social engineering. (Abrams, FBI Issues Alert For LockerGoga and MegaCortex Ransomware 2019) The updated variant was discovered by MalwareHunterTeam in late 2019. (Abrams, New Megacortex Ransomware Changes Windows Passwords, Threatens to Publish Data 2019) When a network is compromised, the adversary downloads Cobalt Strike which allows them to deploy beacons, open a Meterpreter reverse shell, perform privilege escalation, or create a new session to develop a listener on the system. (Abrams, FBI Issues Alert For LockerGoga and MegaCortex Ransomware 2019) (Kim 2019) It then proceeds to encrypt victims' files, change their Windows passwords, and threaten to publicize all stolen data if the ransom is not paid. (Barth 2019)

Infection Vector: In the original version of MegaCortex, it is installed via network access through an Emotet trojan and pushes out onto machines by exploit kits or Active Directory controller. In the updated version, the installation process includes exploits, phishing attacks, Structured Query Language (SQL) injections, and/or stolen login credentials.

Encryption Used: AES-128 (Trend Micro 2019)

Decryptor Available: No public decryptor available

Industries Targeted: Large enterprise networks (Abrams, FBI Issues Alert For LockerGoga and MegaCortex Ransomware 2019)

Countries Targeted: United States, Italy, United Kingdom, Norway, Canada, the Netherlands, Ireland, and France (Abrams, FBI Issues Alert For LockerGoga and MegaCortex Ransomware 2019)

See Appendix 6.8 for more IOC.

2.1.9 Maze

Overview: Maze Ransomware, also called ChaCha ransomware after one of its encryption methods, has become remarkably known for its public extortion campaigns. First discovered in May 2019, attacks began to gain aggression in October 2019 with a three-step approach combination of encryption, exfiltration, and extortion as a part of a multipronged cyberattack. (Abrams, Maze Ransomware Says Computer Type Determines Ransom Amount 2019) Maze's known method of infection of a victim is to pose as a legitimate government agency or security vendor through phishing emails and stolen branding or lookalike domains. (Walter, Maze Ransomware Update: Extorting and Exposing Victims 2020) Once on a system, the ransomware scans all folders, assigns the files found randomly generated extensions, and encrypts all files only excluding itself and those with .ini extensions. Shadow copies of files on the machines are deleted, the wallpaper is changed, and the ransomware then proceeds to create a ransom note named "DECRYPT-FILES.html" that includes the author email with further instructions on ransom payment for decryption. (Abrams, Maze Ransomware Says Computer Type Determines Ransom Amount 2019) (Malware Guide 2020). Maze has a reputation of publicly disclosing victim data through "name and shame" websites if the ransom is not paid in the timely manner.

Infection Vector: Maze utilizes the Spelevo and Fallout exploit kits, targeting CVE-2018-15982, CVE-2018-8174 and/or CVE-2018-4878, to trigger victims to execute PowerShell scripts and/or to download and deploy the ransomware. (Meskauskas 2019)

Encryption used: Combination of RSA-2048 and ChaCha20 encryption. (Abrams, Maze Ransomware Says Computer Type Determines Ransom Amount 2019)

Decryptor available: No public decryptor available

Industries targeted: Healthcare, manufacturing, businesses, and information technology (IT) services (NCFTA 2019)

Countries targeted: North America and Europe

See Appendix 6.9 for more IOC.

2.1.10 Sodinokibi

Overview: Sodinokibi (also known as REvil) is a ransomware as a service (RaaS) that was discovered in April 2019. Sodinokibi works to encrypt the victim's data and delete all shadow copy backups in order to make recovery increasingly difficult. (Cadieux, et al. 2019) The actors behind the ransomware claim they are forbidden to do business in the Commonwealth of Independent States region, which includes Ukraine, Russia, Belarus, and Moldova. The authors of Sodinokibi have been previously linked as the same authors as GandCrab. Ransom payments have been known to record up to six million USD and actors are now following through with posting non-compliant victims' information on shaming websites. (Hall 2020)

Infection Vector/Methodology: Sodinokibi exploits vulnerabilities in servers and managed service providers (MSPs) to take control of networks via RDP and remotely launch attacks. Newer variants use phishing emails as well as a wide range of trojans and exploit kits, such as the RIG exploit kit and Ostap trojan, to infect systems. (Fakterman 2019)

Encryption: Combination of AES and Salsa20 (Tiwari and Koshelev 2019)

Decryptor: No public decryptor available

Industries Targeted: Healthcare and government agencies (Balaban 2020)

Countries Targeted: Asia and Europe

See Appendix 6.10 for more IOC.

2.2 Notable Ransomware Behaviors

These latest forms of ransomware behave similarly to their predecessors but also adapt to changes in the cybersecurity landscape. The two most notable behavior shifts in recent years are ransomware exfiltrating victim's data and Ransomware as a Service (RaaS).

2.2.1 Publishing Exfiltrated Data

Many organizations are using data backups to restore their systems in the event of a ransomware attack. Data backups allow organizations to avoid paying the ransom fee since they do not need to reach out to the ransomware authors for the decryptor. Since ransomware authors want to be paid, they have started exfiltrating data to threaten the organizations into paying a fee to delete the data and not publish it.

Nemty, MegaCortex, Maze, and Sodinokibi have followed through with the threat of publishing victim's data. This strategy is different from past strategies where encrypting a victim's data was sufficient to receive payment. If publishing stolen data proves to increase payment probability, it could indicate that users place a higher importance on avoiding data becoming public as opposed to losing data that stays private. The first corporate victim of publishing exfiltrated data was a secure staffing firm called Allied Universal whose data Maze ransomware published after they refused to pay 300 Bitcoin which approximately translated to $2.3 million USD. (Abrams, Allied Universal Breached by Maze Ransomware, Stolen Data Leaked 2019)

2.2.2 Ransomware as a Service

Some ransomware families, such as Sodinokibi and Nemty, have started using a Ransomware-as-a-Service (RaaS) business model. (McAfee Labs 2019) (Mundo and Lopez 2020) In RaaS, the ransomware developers sell

the ransomware and an easy-to-use platform to affiliate groups who perform the attack campaigns. RaaS also lowers the risk for ransomware developers since they are only providing a platform to perform ransomware attacks and not performing the attacks themselves. This service also reduces the affiliates' cost to mount attacks since they can use the proven prebuilt ransomware. RaaS also increases the threat landscape for organizations since it is no longer necessary for affiliate groups to build custom ransomware to perform an attack. RaaS can be just as profitable for ransomware developers as direct ransom payments since, in most situations, both developers and affiliates get a percentage of the paid ransoms and the ransomware attacks are more widespread. Figure 1 portrays a workflow for an attack using a RaaS platform. The following steps occur in this workflow:

1. The ransomware author develops custom exploit code which is then licensed to a ransomware affiliate for a fee or share in proceeds from the attack.
2. The affiliate uses the custom exploit code and updates the hosting site with this code.
3. The ransomware affiliate identifies and targets an infection vector and delivers the exploit code to the victim (e.g., via malicious email).
4. The victim clicks the link/goes to the website/etc.
5. The ransomware is downloaded and executed on the victim's computer.
6. The ransomware encrypts the victim's files, identifies additional targets on the network, modifies system configurations to establish persistence, disrupts or destroys data backups, and covers its tracks.
7. Victim is instructed to pay ransom with untraceable funds, typically cryptocurrency.
8. A money launderer will move the money through multiple transformations to obscure the identities of the ransomware affiliate and author.
9. The ransomware affiliate may send a decryptor to victim after successful ransom payment. The affiliate may make additional demands on the victim or do nothing at all and leave the victim with encrypted files.

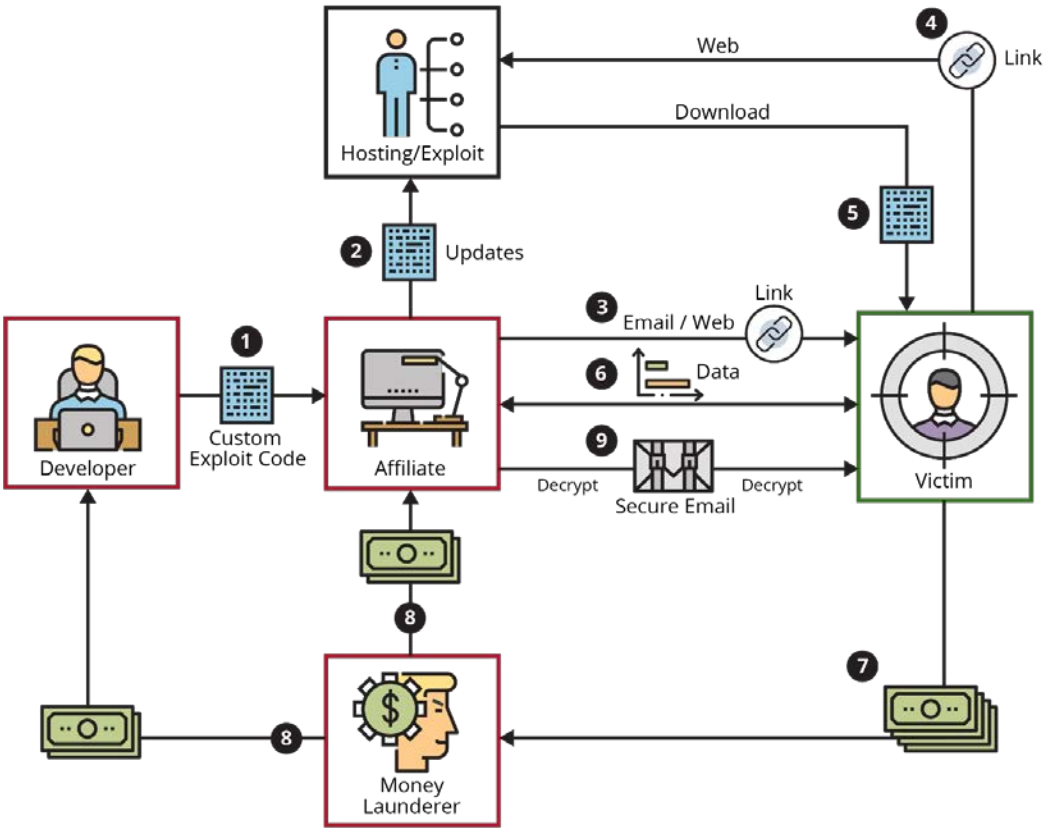

Figure 1: Ransomware as a Service Workflow.[1]

2.3 Ransomware Groups Profitability and Targets

Ransomware typically falls under the category of crimeware. Ransomware campaigns remain largely opportunistic and attempt to attack the easiest targets in the hopes of getting the highest financial reward. However, some ransomware is evolving to be much more targeted. Ransomware has proven to be a highly profitable scheme for its operators. There are two main sources of revenue for ransomware: (1) direct ransom payments and (2) RaaS fees and profit sharing. In the case of direct ransom payments, the ransom amounts can vary from a few hundred dollars to several thousand.

At the RSA Conference 2020, FBI Special Agent Joel DeCapua presented his ransomware research on tracking Bitcoin wallets which found that ransomware victims have paid $140 million in ransoms from 2013-2019. (Spadafora 2020) Currently, Ryuk is the most profitable ransomware family bringing in $61.26 million with its ongoing attack campaigns. (Spadafora 2020)

Ransomware has been targeting healthcare, government, and education and has been successful in obtaining ransom payouts since these organizations tend to have a critical need for their systems. According to 2019 ransomware statistics from Emsisoft, these ransomware attacks affected:

[1] CERT Division researchers and designers created this graphic.

- 764 healthcare providers
- 113 state and municipal governments and agencies
- 89 universities, colleges, and school districts, potentially affecting up to 1,233 individual schools (Emsisoft Malware Lab 2019)

Not only was it a high cost to remedy these attacks, but they also disrupted emergency services, locked police and medical professionals out of internal systems, caused surveillance systems to go offline, and prevented schools from accessing student data regarding medications or allergies. (Emsisoft Malware Lab 2019) These are side effects that can potentially cause physical harm or loss of life.

Mobile devices are another ransomware target to be aware of. While this market is still likely smaller, less studied, and, arguably, less profitable, several samples of ransomware demonstrated their capabilities to encrypt files on mobile devices and demanded a ransom to recover them. This problem will likely persist in the near future, especially on devices that lack full cloud storage backup.

Ransomware is a global problem. MegaCortex alone targets the United States, Italy, United Kingdom, Norway, Canada, the Netherlands, Ireland, and France. (Abrams, FBI Issues Alert For LockerGoga and MegaCortex Ransomware 2019) Nearly half a million ransomware infections were reported globally in 2019 and this number is expected to increase in 2020. (Muncaster 2020) However, ransomware is also regional to areas where ransomware families do not attack. Nemty, Sodinokibi, and Maze ransomware will avoid attacking the Commonwealth of Independent States region.

3 Technical Overview

3.1 Ransomware: Attack Approaches and Techniques

3.1.1 Ransomware Attack Overview

What to Expect. A typical ransomware attack consists of several stages. Knowing what to expect during these stages enables the organization to be better prepared when attacks occur. It is important to remember that ransomware is software code that executes on a compatible compromised computer. Ransomware code manipulates the data it can access on local data storage, over the network, or in the cloud. The code may also use available network access and the Internet to communicate to a command and control (C2) server, which is part of the attacker's infrastructure. Victims of an attack should expect that ransomware will succeed at encrypting the data using a strong encryption algorithm and that the decryption keys will not be available without contacting the ransomware group. In a situation where the ransom is paid, the delivered decryption tool and the decryption keys may not work correctly and some data may still be left encrypted and inaccessible.

Victims should expect that ransomware is able to encrypt data not only on local computer storage, but also throughout the network. Even if access to data seems to be restricted, ransomware may still be able to exploit vulnerabilities and gain access to restricted data. Ransomware may also disable access to critical enterprise data, which is necessary to run customer-facing and back-end services.

3.1.2 Common Ransomware Infection Vectors

Ransomware uses multiple methods to attempt compromising a system. The next sections include a few of the most commonly used infection vectors: emails, compromised websites, and exploits of misconfigured systems on a network.

3.1.2.1 Email

Email is the most frequently used ransomware delivery mechanism. Nearly all the families mentioned in Section 2 utilize email as an infection vector. Cyber criminals acquire mass numbers of email addresses in various ways and use them for phishing campaigns. Ransomware delivered by phishing emails is designed using social engineering. Its focus is on convincing the user that the email is legitimate and its attachments and links should be trusted. In most cases, the email includes a malicious attachment that leads to the ransomware.

Figure 2 is an example of a spear phishing email, which is a targeted phishing email, used by ransomware. (Klein 2015) In this type of phishing attack, the email body content and links are customized to a specific person or organization. This spear phishing email states that a consumer has filed a complaint against Backblaze. It makes sense for this type of email to be sent to the CEO of the accused company. A person could easily accept this email as legitimate and follow the directions to click on the link.

Figure 2: Locker Ransomware Spear Phishing Email from 2015 (Klein 2015)

3.1.2.2 Websites

Ransomware delivered via compromised websites primarily employs malicious advertising (malvertising) in combination with an exploit kit. Malvertising is not a new distribution form and was first seen in 2007. The Sodinokibi ransomware was discovered using malvertising as an infection vector in June 2019 by redirecting the victims to the RIG exploit kit via the PopCash advertising network. (Trend Micro 2019) Nemty also has malvertising campaigns using the RIG exploit kit. (Ilascu, Nemty Ransomware Gets Distribution from RIG Exploit Kit 2019) The advantage of using websites to deliver ransomware is that user interaction may not be required to successfully infect the machine. Simply browsing a webpage causes the malvertising's malicious code to automatically execute and infect.

The use of malvertising is not limited to dodgy websites; malvertising has entered mainstream sites such as those for The New York Times, NFL, MSN, and BBC. The key to malvertising appearing on websites is the legitimate participation by criminals in advertisement networks. A criminal registers to be an advertiser on a network, bids to place advertisements on popular websites, and does so for a while with malware-free ads to gain trust. After some time, they introduce malvertising into these networks, which is displayed on the websites after winning a bid for advertising space.

The ad-placement process is facilitated by the automation of transactions and minimal security checks put in place by advertisement networks. Malvertisers can easily place their ads unnoticed. Once the malvertising is displayed, and in some cases clicked, several redirection approaches can occur. Most commonly a hidden inline frame (iframe) tag within the code starts a series of domain/IP redirects, eventually landing at a malicious server hosting an exploit kit.

Figure 3 shows an example of malicious redirect to the RIG exploit kit using the PopCash advertisement network which downloads and installs Sodinokibi ransomware. (Abrams, Sodinokibi Ransomware Now Pushed by Exploit Kits and Malvertising 2019) Once downloaded and executed, Sodinokibi encrypts the files as shown in Figure 4.

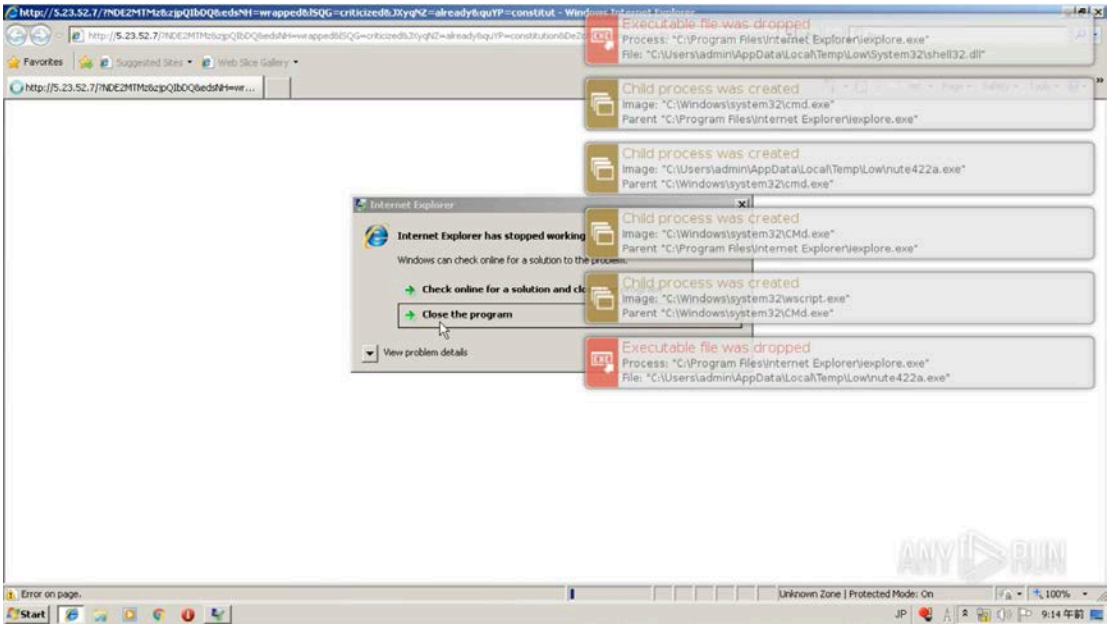

Figure 3: Malvertisement Redirect on an Unpatched Windows PC (Abrams, Sodinokibi Ransomware Now Pushed by Exploit Kits and Malvertising 2019)

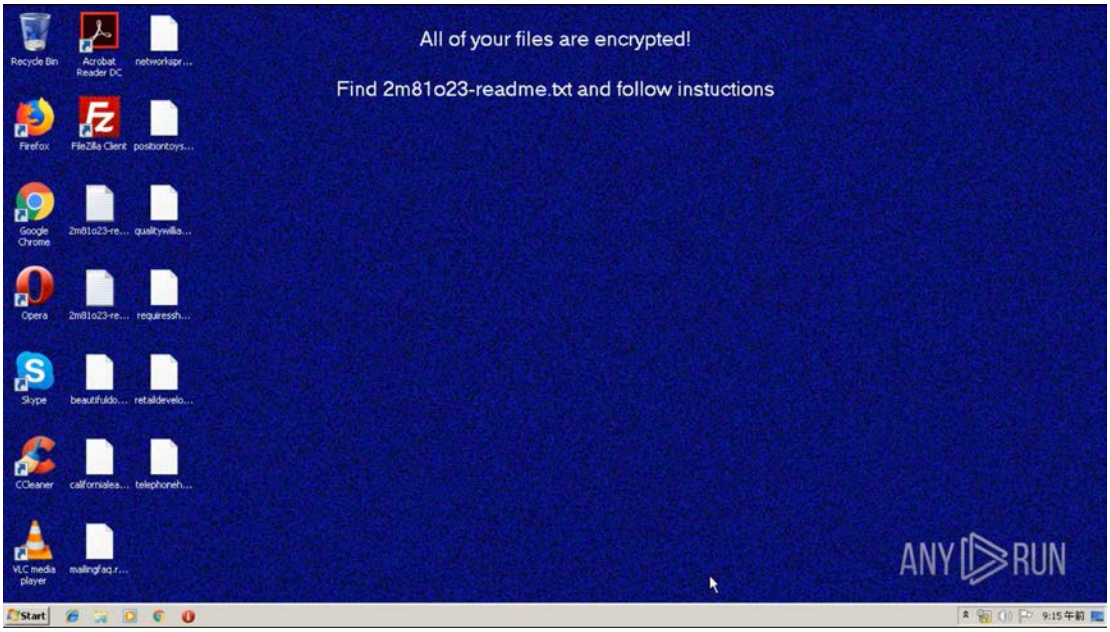

Figure 4: Unpatched Windows PC Encrypted by Sodinokibi Ransomware (Abrams, Sodinokibi Ransomware Now Pushed by Exploit Kits and Malvertising 2019)

3.1.2.3 Exploit Kits

An exploit kit is software that attempts to exploit both known and unknown vulnerabilities in operating systems, browsers, plugins, and other software to compromise a machine. These kits primarily focus on browsers and other software that can be executed automatically by visiting a webpage. Most modern web browsers restrict webpage access to system resources to avoid compromise and require substantial user interaction for the restriction to be lifted.

Once on the system, the malicious code downloads the ransomware from a remote server and executes it on the machine. The infection is usually performed in the background and, if not detected promptly, the data encryption completes and the ransom payment messages appear.

The key to an exploit kit's success is its ability to discover vulnerabilities. Some kits exploit only publicly disclosed vulnerabilities while others stockpile privately discovered vulnerabilities, which give them an edge to raise their price in underground markets. The following are the most often exploited technologies:

- popular software (e.g., Adobe Acrobat Reader, Microsoft Office, WordPress)
- browsers (e.g., Internet Explorer (IE), Firefox, Chrome, Safari)
- plugins (e.g., Adobe Flash)

There are several exploit kits used in the wild. Some ransomware operators may be decreasing their use of these kits in favor of phishing emails. Some of these exploit kits are described below.

RIG. This exploit kit appeared in 2016 and is known to spread at least 35 ransomware families including Nemty and Sodinokibi. (McAfee 2018) The RIG Exploit Kit is maintained and has been updated to abuse over 34

vulnerabilities, including the newer vulnerabilities CVE-2018-4878 and CVE-2018-8174 which target Adobe Flash Player and Microsoft Windows. (McAfee 2018)

Fallout. Discovered in August 2018, this exploit kit is known to spread multiple ransomware families including Maze and Sodinokibi. (McAfee 2019) Fallout uses CVE-2018-4878, CVE-2018-8174, and CVE-2018-15982, which are vulnerabilities in Adobe Flash Player and Microsoft Windows.

Spelevo. Discovered in early 2019 and is known to spread Maze ransomware. (McAfee 2019) The Spelevo Exploit Kit uses CVE-2018-8174 and CVE-2018-15982 to exploit Adobe Flash Player and drop a trojan that creates a persistent scheduled task in Microsoft Windows. (McAfee 2019)

Radio. This exploit kit targets Microsoft Windows and is known to have spread the Nemty ransomware. (Abrams, Exploit Kits Target Windows Users with Ransomware and Trojans 2019) The Radio Exploit Kit is less advanced than other exploit kits and abuses the overused CVE-2016-0189 vulnerability in Internet Explorer which has since been patched. (nao_sec 2019)

Exploit kits come in all flavors. They are able to compromise systems even without user interaction. The best way to avoid falling victim to these kits is to follow strict software update policies. It seems that most kits use exploits for publicly disclosed vulnerabilities with only a smaller number using privately discovered vulnerabilities. The success of exploit kits implies that software is not being regularly patched, allowing kits to continue relying solely on publicly disclosed vulnerabilities.

3.1.3 Operating System (OS) Modifications

Ransomware behaves like most malware once it is executed on the victim machine and has a common set of initial execution goals:

- Establish persistence on the compromised machine.
- Avoid detection and subsequent removal on the compromised machine.
- In some instances, establish network connection to a C2 server.

3.1.3.1 Establishing Persistence

Persistence allows ransomware to continue executing and have access to all needed resources on the system long enough to achieve its malicious goals. Ransomware uses some of the strategies below to establish persistence on a compromised system.

- **Self-replication into new files.** Ransomware makes multiple copies of its own executable file into system folders and less common locations such as temporary folders. The purpose of copying this file is to avoid full eradication if all the copies are not detected or disallowed to be deleted in the case of system files. In the Windows OS, typical locations are the system32 and AppData, Local, and temp folders.
- **Self-replication into existing files.** Ransomware writes malicious code, that is either a copy of itself or some other nefarious code, into existing binaries. Ransomware tends to copy into system folders because anti-malware software does not delete these system files or stop them from executing since they are critical to proper OS functionality. Further, files in system folders tend to have administrative privileges on the system.

- **Creation of new binaries.** The ransomware dropped on a system often creates new binaries that are later executed to perform malicious tasks. The purpose of doing this is to separate the two files. If the created file is eradicated, the originally dropped file can recreate the file and reattempt its malicious goals. The binaries are either downloaded from a remote host or copied from embedded code in its own executable file (but not an exact copy); this last case is a form of self-replication.
- **Windows Registry.** Ransomware sometimes uses the Windows Registry to establish persistence. The Windows Registry is a hierarchical database that contains settings for the operating system and applications that opt in. Ransomware attempts to create and alter registry keys for its own use as well as reset or delete keys related to security and anti-malware. Additionally, ransomware attempts to set up auto start of malicious binaries by setting paths to several known start-on-boot registry keys, primarily the "CurrentVersion\Run" key.

3.1.3.2 Avoiding Detection

Ransomware needs to avoid detection and subsequent removal long enough to encrypt the victim's data, which can sometimes take hours. It uses some of the strategies below to remain undetected on a compromised system.

- **Deletion of ancestor files.** Ransomware executes the binary files it downloads or creates. These processes often spawn malicious child processes which help avoid removal from the compromised system. These processes are typically self-replications running a new instance, meaning the child processes perform the same action as the parent process. These processes may be trojan horses performing various malicious tasks or other binaries providing some supporting functionality that is not inherently malicious. To avoid detection, a spawned child process may delete the binary image of an ancestor process from the file system to minimize artifacts and the overall footprint left on the compromised system.
- **Termination of ancestor processes.** As mentioned above, ransomware typically creates and spawns malicious child processes after the initial download or creation of a binary file. After creating the child processes, the parent process may self-terminate. In some cases, a child process terminates a parent process; this is an example of ancestor process deletion. Its purpose is to separate the dropped ransomware process from descendants performing malicious tasks to avoid detection and removal from the compromised system.
- **Disabling of anti-malware software.** Some ransomware variants are aware of processes and configuration options known to belong to anti-malware programs. To avoid detection, the ransomware attempts to disable anti-malware software by terminating processes, causing the anti-malware program to run improperly or stop altogether.
- **Injection of code.** To delegate malicious tasks to non-malicious processes, ransomware often injects code into other currently running processes to avoid detection. Code injection is also useful because the non-malicious processes may have higher access to system resources and can be used by ransomware for data collection or to observe user interactions. Often targeted are system-level OS processes, file managers, and web browsers. In the Windows OS, some of these processes might include winlogon, svchost, iexplore, and explorer.

3.1.3.3 Establishing a Network Connection

Remote C2 server communication. Ransomware variants often attempt to communicate with remote C2 servers. This communication is a critical step in ransomware's core functionality and is used primarily to store encryption keys, to store unique identifiers of victim machines, and, more recently, to exfiltrate data to extort

ransomware victims. Upon initial connection, ransomware issues DNS lookups and reverse DNS lookups, and attempts connections to several IPs until successful. The failed attempts should stand out and be flagged in real-time traffic analysis. To make detection difficult, communication with C2 servers may also be routed through network anonymizers (e.g., Tor).

3.1.4 Finding the Target

Ransomware may target specific data, all files on a system, or even entire disks; most variants of the ransomware families discussed in this report scan the full local filesystem system of a device. Additionally, some ransomware families like Ryuk and MedusaLocker target not only the local hard drive but also any attached or accessible network storage as shown in Table 2.

Table 2: Overview for Ransomware Families [2]

Ransomware	Encryption Used	Locations Encrypted	File Extensions Encrypted	Published Data
FuxSocy (Abrams, New FuxSocy Ransomware Impersonates the Notorious Cerber 2019)	Combination of RSA and AES-256	Most variants attempt full system, except whitelist	Wide range including archive, office (e.g., documents, spreadsheets, presentations), image, media, script/code, database. Does not encrypt entire file, only enough to corrupt the data (NCFTA 2020)	No
GlobeImposter (Kline 2017)	Combination of RSA-4096 and AES-256	Most variants attempt full system, except whitelist; Looks for other hosts on network to encrypt (Cyware Social 2019)	Attempts all files; except whitelist (Cyware Social 2019)	No
LockerGoga (Trend Micro 2019)	Combination of RSA-4096 and AES-256	Predefined list or full system	Wide range including archive, office (e.g., documents, spreadsheets, presentations), image, media, script/code	No
SamSam (Coveware 2019)	RSA-2048	Full system	Attempts all files; except whitelist	No
MedusaLocker (Abrams, MedusaLocker Ransomware Wants Its Share of Your Money 2019)	Combination of RSA-2048 and AES-256	Full system, except whitelist; Also targets mapped network drives (Walter, How MedusaLocker Ransomware Aggressively Targets Remote Hosts 2019)	Wide range including executables, office (e.g., documents, spreadsheets, presentations), image, media, script/code	No
Ryuk (Hanel 2019)	Combination of RSA-2048 and AES-256	Most variants attempt full system, except whitelist; Looks for other network-accessible shares	Attempts all files; except whitelist	No
Nemty (Mundo and Lopez 2020)	Combination of RSA-2048, RSA-8192, AES-128, and	Full system, except whitelist	Attempts all files; except whitelist	Yes

[2] CERT Division researchers created this table.

Ransomware	Encryption Used	Locations Encrypted	File Extensions Encrypted	Published Data
	AES-256 (van den Hurk 2019)			
MegaCortex (Abrams, Elusive MegaCortex Ransomware Found - Here is What We Know 2019)	AES-128	Attempts full system, except whitelist	Attempts all files; except white-list	Yes
Maze (NCFTA 2019)	Combination of RSA-2048 and ChaCha20	Attempts full system, except whitelist	Attempts all files; except white-list	Yes
Sodinokibi (NCFTA 2020)	Combination of AES and Salsa20	Most variants attempt full system, except whitelist; looks for other network-accessible shares (Tiwari and Koshelev 2019)	Wide range including archive, office (e.g., documents, spreadsheets, presentations), image, media, script/code	Yes

Figure 5 is a diagram of the files typically encrypted by ransomware, ranging from those most likely to be encrypted by ransomware, on the outer circles, to the least commonly targeted files, in the smallest circle of the diagram. Typically, the most commonly targeted files are various office-type documents (e.g., .doc, .sxw, .xlsx, and .sxc), image formats (e.g., .tiff, .png, and .bmp), archive files (e.g., .zip and .tar), audio files (e.g., .mp3 and .wav), and video files (e.g., .mp4 and .avi). Some ransomware variants may also include database (e.g., .sqlite and .mdb) and website-related files (e.g., .html and .aspx). Less sophisticated encrypting ransomware may limit its targets to specific directories on a system and highly targeted ransomware may hone in on a specific application (e.g., MongoDB).

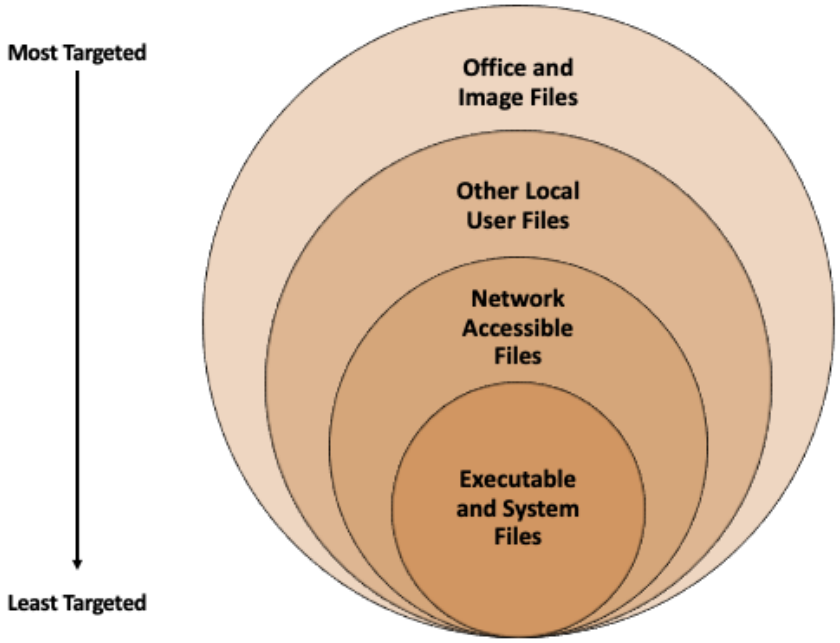

Figure 5: A Diagram of the Files Most Likely to Be Encrypted by Ransomware on the Outer Circles, to the Least Commonly Targeted Files in the Smallest Circle of the Diagram[3]

As shown in Figure 5, many ransomware families avoid encrypting executable or system files because doing so can render the system unstable and may prevent even the ransomware from working. However, this avoidance is changing and many ransomware families are now whitelisting the minimal files needed to boot the computer and encrypting the others. Ransomware also targets data residing on organizations' servers (e.g., databases, websites, and file shares) to leverage the negative impacts of a data breach in order to pressure victims into paying ransoms.

3.2 Encryption

Ransomware uses some form of encryption to restrict access to the data that it holds for ransom. Encryption is a method to protect data by scrambling it with an encryption algorithm that uses a unique encryption key. When used properly, encryption protects the data from being accessed without authorization (i.e., the unique decryption key) and ransomware is designed to not reveal the decryption key unless the ransom is paid. This section will review encryption algorithms used by ransomware.

3.2.1 Algorithms

Most ransomware variants use both symmetric and asymmetric encryption algorithms in their attacks. Symmetric encryption is the simpler and faster of the two and is typically used to encrypt large sizes of data. Symmetric encryption only uses one secret cryptographic key to encrypt and decrypt the data, as shown in Figure 6. Asymmetric encryption is more complex and requires two different cryptographic keys, denoted as a public key and a

[3] CERT Division researchers and designers created this graphic.

private key, as shown in Figure 7. The public key encrypts the data and the private key is needed to decrypt the data encrypted by the public key. Since it is more complex, it is much slower and typically used to encrypt small amounts of data.

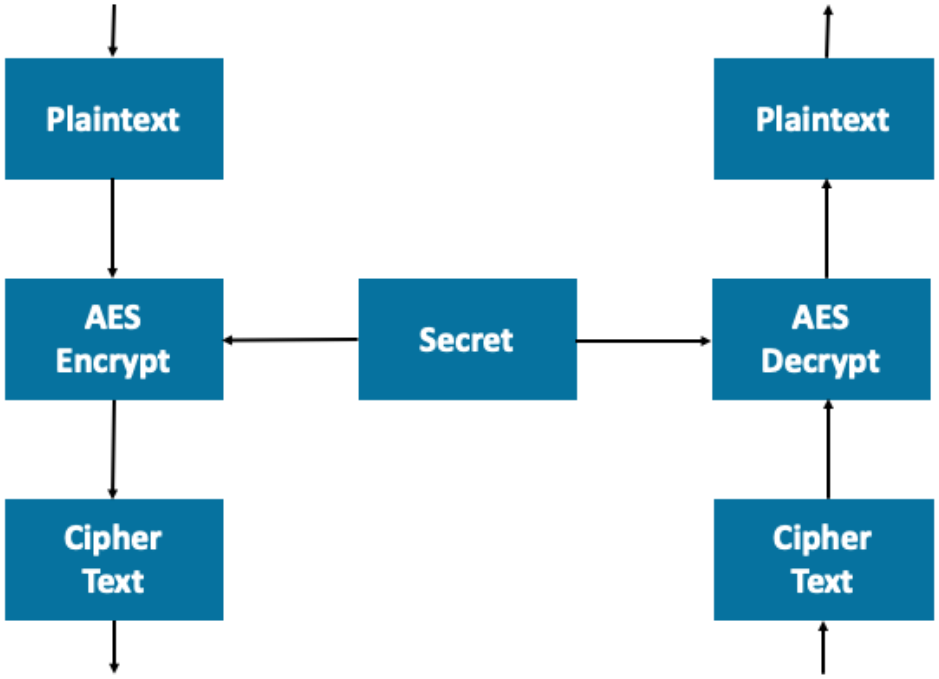

Figure 6: Symmetric Encryption Algorithm

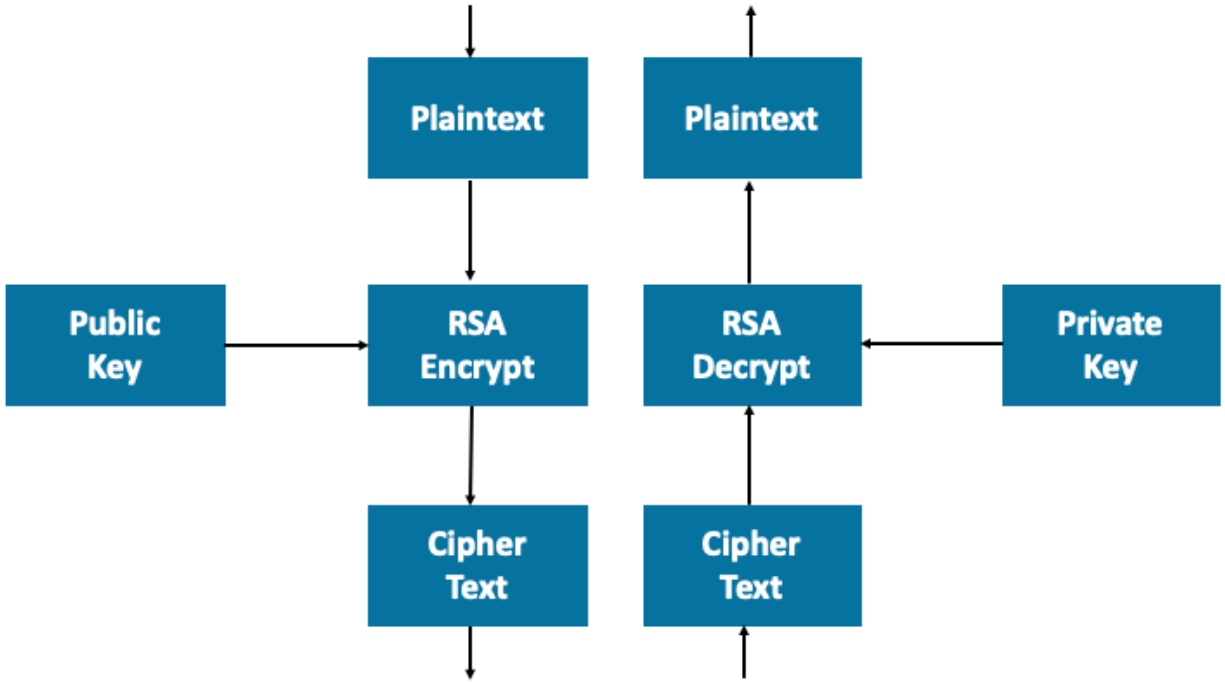

Figure 7: Asymmetric Encryption Algorithm

Consider MedusaLocker as an example. The ransomware encrypts files using an AES-256 symmetric encryption algorithm. Since this is a symmetric encryption algorithm, the same key must be used to encrypt and decrypt the data, as shown in Figure 6. MedusaLocker protects the symmetric key by encrypting it using an RSA-2048 asymmetric encryption algorithm where encryption is done with a public key and decryption is completed with the corresponding private key, as shown in Figure 7. In most cases, the public key is provided to the ransomware from the C2 server, which also holds the associated private key. This limits the encryption methodologies since access to the symmetric encryption key used to encrypt the data was not possible without access to the C2 server. However, MedusaLocker works around this by embedding the public key in the executable, meaning it does not need to connect to a C2 server to utilize both symmetric and asymmetric encryption.

The MedusaLocker example demonstrates the typical complexity of using a combination of encryption algorithms in ransomware. In fact, many new ransomware families try to follow similar best practices for encryption, key generation, and key management making it more difficult to recover data without paying ransom.

A best practice example for data encryption recommends that large amounts of data should be encrypted using a symmetric encryption because symmetric encryption is much faster than asymmetric. As a result, a properly implemented encryption scheme typically involves both a symmetric encryption component (e.g., AES) and an asymmetric encryption component (e.g., RSA), as shown in Figure 8. Vulnerabilities are common in custom encryption algorithms since they are not subjected to the extensive rigor of code review. Ransomware developers limit the potential vulnerability surface by using established encryption algorithms. By limiting the vulnerabilities in the encryption process ransomware uses, developers decrease the chance of a public decryptor being developed to recover victims' data. MedusaLocker follows these best practices by using the AES-256 symmetric

encryption algorithm and the RSA-2048 encryption algorithm, as mentioned above. Currently, there are no publicly available decryptors available for MedusaLocker.

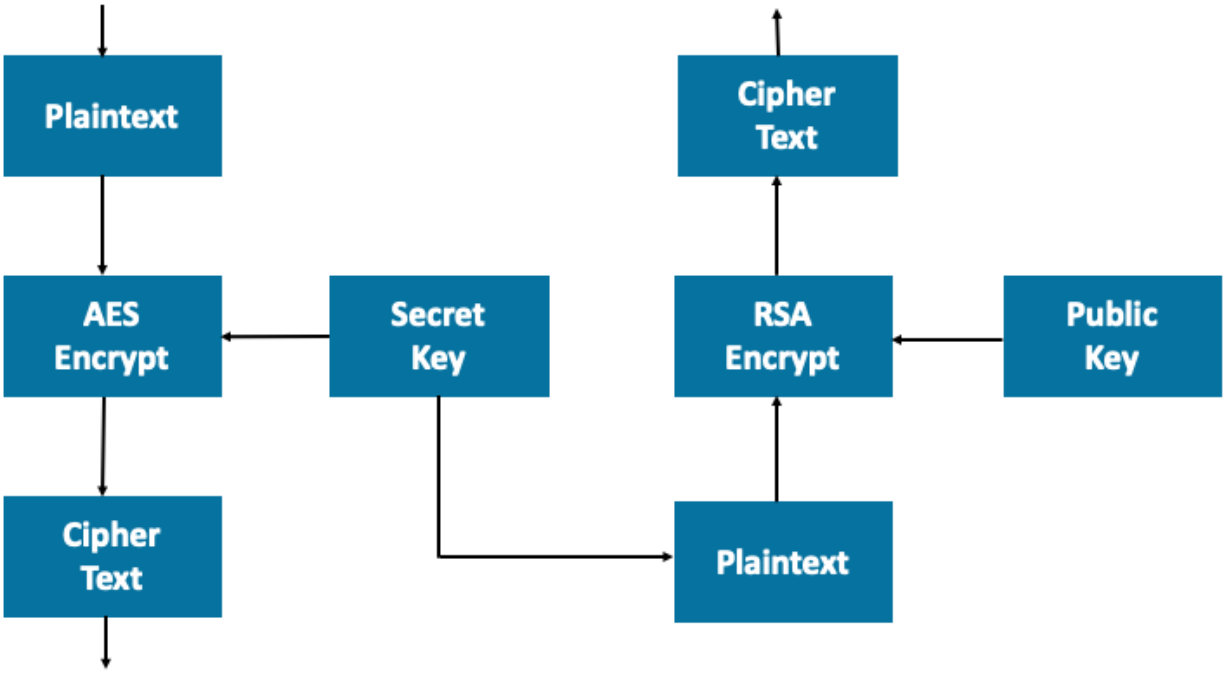

Figure 8: Combining Symmetric and Asymmetric Encryption to Protect a Secret Key

This combination of symmetrical and asymmetrical encryption methods is commonly used among many other ransomware families, such as FuxSocy, GlobeImposter, LockerGoga, Ryuk, and Nemty. Unlike the MedusaLocker example, some of these ransomware families communicate with a C2 server in order to generate the RSA public key that is used to encrypt the AES symmetric key, which makes a connection to a C2 server a requirement. If the ransomware is unable to connect to the C2 server, it will be unable to obtain the public key to encrypt the AES symmetric key, and without a secure symmetric key, the ransomware group will be unable to encrypt and ransom the data.

Several strains of ransomware attempted to solve this problem by encrypting data with a locally generated AES-256 symmetric key. If the key is left in plaintext, it may allow for the victims to decrypt their data without paying a ransom. The ransomware stores the plaintext symmetric key locally unless it is unable to communicate with the C2 server. In that case, the ransomware creates a recovery key from the symmetric key using a hard-coded RSA public key. Once the recovery key is created, the original plaintext AES symmetric key is deleted from the system.

The recovery key can later be submitted to a website controlled by the threat actor, along with the ransom payment, to retrieve the original master AES symmetric key. This symmetric key can then be used by the decryption tool to recover the data.

3.2.2 Data Integrity

Data encryption involves data modification, which may lead to data corruption. If the data is corrupted during the encryption process, it may not be recoverable, even with a properly working decryption tool. Depending on the ransomware implementation, the original plaintext data may either be modified in place or copied to another file container for encryption. Once encrypted, the original plaintext data file is deleted from the system. However, a flaw in the implementation of the ransomware code may lead to an incomplete state of the encrypted data. As with many other applications that involve writing large volumes of data, an abrupt interruption of the data-writing process may lead to data corruption.

Several recent ransomware families modify original data files to include encryption keys and unique identifiers that must be removed before the original file can be recovered by a decryption tool. Finally, the decryption tool may also have implementation flaws, which can cause data corruption.

If the original data is deleted but not overwritten in storage, it may be possible to recover deleted files using standard data-recovery tools. If the decryption tool corrupts the data, it may be possible to modify the tool to avoid data corruption. In all cases, it is recommend having a backup copy of the encrypted data before attempting any data recovery.

As with any malicious data modification, it is impossible to guarantee that the data integrity has not been violated without comparing the data to a previously stored copy or data hash values. For this reason, the recovered data should be treated as compromised.

3.3 Payment

3.3.1 Informing the User

Once encryption is complete, ransomware alerts the user of the situation and next steps. Messages are sent in various forms including desktop wallpapers, browser windows, pop-up windows produced by the ransomware executable, and reminders. The ransomware also typically locks or limits the use of the computer and displays messages continuously.

The message content typically consists of four sections:

1. A statement that the files have been encrypted
2. The reasons for the encryption
3. The steps the victim must take to recover their files
4. A final deadline to pay the ransom and the consequences of a victim delaying payment of the ransom; the ransom amounts typically increase over the period of time before the final deadline

These messages often use social engineering to convince the user this is a serious situation and the only answer is to pay. Some messages offer to decrypt a handful of files to convince the victim that paying leads to decrypting the rest.

3.3.2 Currency

The widespread availability of cryptocurrency has enabled the rise of ransomware in recent years. The unregulated nature of cryptocurrency transactions makes it the preferred payment mechanism for ransomware payments. Bitcoin accounted for about 98% of ransom payments made in the first quarter of 2019. (Coveware 2019)

In addition to cryptocurrencies, some ransomware families also accept payments in various forms of prepaid debit cards, gift cards, and more traditional means of wire transfer and money orders. Prepaid cards provide a convenient payment option for a victim who may not be comfortable with the complexity of a cryptocurrency transaction. However, prepaid card payment options may vary based on the location of the user. Different prepaid cards and payment terminals exist in Europe, Asia, North America, and other markets.

3.3.3 Customer Service

Some ransomware operators provide customer service to help victims obtain the required payment currency, make payments, and process decryption. The logic behind this service is that the more help provided to the victims, the more likely the ransom will be paid. Ransomware customer service is usually a common support request form filled out by the user or a chat window. Further communication typically occurs by email. Aside from helping the victim make a payment and decrypt the data, the communication channel may also be used to negotiate the ransom amount or to extend the payment deadline. (Shackelford and Wade 2020)

3.3.4 Payment Deadline Expiration

Ransomware typically sets a deadline for the ransom to be paid before the decryption key is destroyed. In the traditional sense of cryptographic security properties, if the decryption key is gone, the encrypted data is considered destroyed. Ransomware relies on this property to force the victim to make the payment early. The decryption key is typically stored on the C2 server that is under the attacker's control. Therefore, it is feasible that the attacker can implement a key destruction procedure based on the amount of time passed since the original infection.

Prior analysis of ransomware C2 servers revealed several cases where the decryption key was not destroyed past the expiration date. While the key expiration warning was likely used to entice victims to make the payment as early as possible and discourage extending time looking for an alternative solution, it is often impossible to tell without analysis if the key expiration threat is real or not. Therefore, without additional knowledge, it should be taken seriously.

Ransomware may also use the deadline expiration to raise the ransom amount instead of threatening to destroy the decryption key. In such cases, the ransom is typically doubled every 7-10 days until paid. It is not unusual for ransomware customer service to extend the payment deadline or negotiate the price.

3.4 Decryption

3.4.1 What the Actors Provide

After the ransom payment is made, ransomware operators provide decryption keys, software, and support. Sometimes, access to the decryption software is embedded in the ransom payment screen. In these cases, the user only needs to provide the encrypted files. In other cases, the software must be downloaded.

The operators further assure the decryption is correct and all files are returned to their original form. This service is critical to the continued success of the campaign. Knowing that the victim's files will be restored correctly builds a trustworthy path out of the ransom situation. Figure 9 shows example from Alma Locker ransomware.

Figure 9: Alma Locker Built-in Decryption Tool (Cimpanu 2016)

3.4.2 Trustworthiness of Decryption

Some reports suggest that the data held for ransom could not be recovered even after the ransom was paid. Encrypted data can only be successfully decrypted if it has not been corrupted. In Q4 2019, victims who paid for a decryptor still lost 3% of their encrypted data on average. (Coveware 2019) Additionally, a typical decryption tool is responsible for not only decrypting the data, but also for locating the data and identifying the proper decryption keys to use, which can be more complex when a unique key per file is being used.

A more complex identification and decryption process leaves more room for coding and functional errors, which may lead to incomplete or corrupted data decryption. Also, decryption tools are often shipped with a decryption key that is unique to the machine where the encrypted data is stored. The victim assumes that the threat actor provided the correct decryption key from the database of all keys held for ransom. An example of an invalid decryption key involves a poorly generated unique identifier for the compromised system that leads to multiple collisions in the key database. An invalid decryption key leads to either a failed decryption attempt or data corruption.

The integrity of decrypted data can no longer be guaranteed and verified without comparing it to the original data or verifying hash values. Regardless of the outcome of the decryption, or the source of the decryptor tool, the decrypted data should be treated as compromised.

3.4.3 Reliability Concerns

Ransomware is designed to manipulate data on a disk at rest. When encryption occurs, the data goes from a plaintext format to an encrypted format. Properly implemented software that does this kind of data manipulation typically encrypts the data, stores it in a separate new file, and then securely deletes the original plaintext data as shown in Figure 10. This workflow requires careful considerations of file system operations to preserve data integrity.

Figure 10: Ransomware File Encryption Workflow

If the data manipulation process is interrupted or fails, the data may be left in a corrupted state. Ransomware code typically does not go through a rigorous testing phase compared to commercial software products performing similar data manipulations. This untested code may create a larger error surface that leads to data corruption.

On the decryption side, ransomware decrypts the data, stores it in a separate new plaintext file, and then deletes the original encrypted file, as shown in Figure 11. While the decryption process is typically less prone to errors related to the file system and operating system, the tool used for decryption may be less stable than the ransomware encryption component.

Figure 11: Ransomware File Decryption Workflow

Ransomware typically modifies the compromised system or network and may lead to a less stable and secure system. These modifications may involve exploiting various code vulnerabilities to obtain necessary access to perform the attack. For example, ransomware may install a kernel-level driver under elevated privileges. At this level of access, the compromised system may experience instability, performance issues, and system crashes regularly until the ransomware is removed.

These concerns should be considered when making the decision on how to handle the ransomware incident. Backups of the encrypted data should be created after the incident, but before the decryption attempt. Similarly, when the ransomware is removed and the data is recovered, the compromised systems should be completely erased, reinstalled, and reconfigured after the incident. This renewal of the system helps avoid overall system stability and security issues and reduces the potential for future compromise.

3.5 Data Exfiltration

Earlier versions of ransomware were not usually designed to exfiltrate large amounts of data from an infected network. However, it is expected that some potentially sensitive information may be intentionally transferred from the victim to the threat actor. Previous analysis of ransomware families demonstrated that most of them collect some information about the compromised system to create a unique identifier. This information is used by the C2 server to track ransomware attacks and the encryption keys needed to recover the data. The compromised system's computer name, storage device serial number, and operating system version are often used to generate unique identifiers and may be transferred by ransomware to a C2 server controlled by the threat actor.

Some ransomware may also choose to exfiltrate information about the individual files being encrypted, including the filename and path. Obviously, this process poses some risk of data leakage, especially if sensitive information is contained in the filename or path.

Furthermore, the Nemty, MegaCortex, Maze, and Sodinokibi ransomware families have started exfiltrating the full content of the data for the purpose of publicly exposing it unless the ransom is paid. Data exfiltration typically occurs before the data encryption happens. These four ransomware families have followed through on the threat and subjected the victim organizations to a data breach in addition to the ransomware attack. In most cases, the content of documents, databases, and other sensitive digital materials end up in the hands of the attacker. However, there is no guarantee that the data will be deleted securely by the attacker after the ransom is paid, and other threat actors may also gain access to the data.

4 Stopping Ransomware

4.1 Monitoring

4.1.1 System Level

Several indicators of compromise are typically used to detect ransomware at the system level. Antivirus software offers some protection against ransomware, but that protection is incomplete. An antivirus product should be able to detect and block known strains of ransomware at the file and process level before the data is damaged. Additionally, it should be able to scan online downloads and email attachments, the two most common attack vectors used to deliver ransomware. However, antivirus typically relies on updated hash lists, and ransomware can adapt fast enough to avoid detection by an antivirus software.

Removing local administrative rights helps prevent ransomware from running on a local system. The local administrator has the power to modify system files and directories as well as system registry and storage. Those are critical components to any ransomware operation. Removing local administrative privileges reduces the chance that ransomware will persist on the system and propagate throughout the enterprise network. It also reduces the chance that ransomware will have access to critical system resources that may be necessary for destructive file encryption.

Network firewalls can detect ransomware that requires communications with a remote C2 server. Configuring the local system firewall to monitor and block outbound network communications from applications not on the approved list can also help to stop ransomware. Without access to the C2 server, some ransomware variants may not be able to use strong data encryption, may fail to encrypt the data, or may downgrade the encryption mechanism to one with a higher chance of data recovery without paying the ransom.

System vulnerabilities are also leveraged by ransomware. It is important to have a sound strategy of patching the operating system against known security issues. However, third-party applications pose a security risk as well, especially those with higher privileges. Proactively identifying applications that require higher system privileges helps in detecting and preventing ransomware before it can gain access to sensitive data. Maintaining and patching third-party applications regularly, restricting the installation of new applications that are not on the whitelist, monitoring for new process execution, and blocking unapproved applications from running are recommended.

Monitoring the execution of code within the Windows temporary folder and AppData folder also helps reduce the risk of ransomware execution at the point of infection. Many variants of ransomware are delivered through downloads and attachments and must be deployed before the data encryption can begin. Malicious code deployment often requires easy access to a readily available directory to unpack and execute ransomware files. The Temp and AppData folders often meet these requirements. If the system is configured to detect and block code execution within these folders, ransomware may not be able to execute the encryption code after deployment. Assuming the indicators are carried out by a currently running process, monitoring can be achieved using Microsoft-supported mini-filters and callbacks in a Windows system. The runtime events mentioned below should be monitored to increase the likelihood of detecting ransomware execution.

Ransomware typically performs the following actions in a file system:

- modifies autostart files to execute and display ransom messages to the victim
- searches the file system for all files with specific file extensions
- requests high-frequency access to multiple files
- creates new files, possibly with non-standard file type extensions

Ransomware frequently alters the following processes:

- spawns a new process from binaries it created or downloaded
- deletes anti-malware related processes to avoid removal during encryption and ransom demands
- injects part of its own binary image or memory space into the memory space of a previously running process

Ransomware usually makes changes to areas of the Windows Registry:

- sets up the autostart of malicious binaries to control machine access and display ransom messages
- resets or deletes security and anti-malware related keys to avoid removal before the ransom has been paid

4.1.2 Network Level

Some ransomware variants use remote servers to store exfiltrated victim data, encryption information, and other items. When the ransomware executable attempts the initial connection to a C2 server from a compromised system, it must determine which IP address is active. The process of acquiring and connecting to multiple IP addresses until success can produce failed connection attempts and other anomalies. Monitoring for these failed connections and other anomalies can alert the victim to ransomware on their systems.

The following behaviors should be logged and flagged when they occur because they indicate failed connection attempts and other IP address harvesting anomalies:

- DNS queries with no results
- reverse DNS queries with no results
- successful DNS queries and failed connection attempts to the returned IP address
- high frequency of repeated DNS queries to the same or a small number of top-level domains

Proactively logging and searching for these behaviors can be accomplished with a network traffic analyzer. To avoid detection and blocking by ransomware, log and monitor network traffic from within the kernel. Microsoft provides support for doing this using the Windows Filtering Platform, which provides the application programming interface (API) and commands to build kernel-level network monitors in various layers of the network stack.

Additionally, content filtering can be utilized to detect ransomware with data exfiltrating behavior by monitoring known exfiltration vectors such as HTTP, FTP, and email. (Jareth 2020) Some content filters can be customized to identify data patterns that match sensitive organizational data. However, if ransomware obfuscates the exfiltrated data during transfer, the content filters can be circumvented. Digitally watermarking files can also be effective in detecting data exfiltration. (Jareth 2020) A watermark is embedded into a file and can be detected by a deep packet inspection product in real-time when the watermarked file is exfiltrated. By detecting these behaviors, along with others at the host level, it is possible to stop ransomware before encryption begins.

4.2 Policies and Procedures

The most effective prevention mechanism against ransomware attacks is keeping regular, verified data backups. Recent strains of ransomware not only encrypt document files, but they also encrypt Windows OS system restore points and shadow copies that are often used to recover the data after a ransomware attack.

A good backup strategy involves keeping backups on a separate system, not accessible from the network, and helps recover encrypted data without paying the attackers. Also, frequent inspections of data backups help ensure that the data backup strategy is consistent and reliable.

4.2.1 Mitigation

The following are effective mitigation strategies against ransomware:

- **Regularly make and maintain offline backups.** If ransomware can't reach the data over the network, that data has a better chance of staying secure. Existing backup procedures should be checked regularly to verify that data is being backed up properly. Periodic checks of backup data integrity ensure that backup data is valid and available.
- **Educate employees in security best practices through user training.** Ransomware most often attacks the system through email attachments, online downloads, web browsing, and USB drives. Regular employee security training addressing these issues and other best practices helps avoid ransomware infections.
- **Restrict code execution in temporary and data folders.** If ransomware was designed to extract and execute from these folders, it may not be able to proceed with data encryption if it does not have privileges.
- **Maintain regular patching.** Both the operation system and third-party components can be vulnerable to ransomware, updates ensure ransomware can't exploit known vulnerabilities to gain access to the system and data.
- **Restrict administrative and system access.** Ransomware may rely on a system administrator account to perform its operations. Restricting user accounts and eliminating default system administrator accounts creates an extra barrier for ransomware.
- **Maintain and update antivirus software.** Regularly updating antivirus downloads the most recent malware signatures and other identifying data available to maximize the potential for early detection of known ransomware.

4.2.2 Response

The following are effective response strategies to ransomware attacks:

- **Snapshot system memory.** If possible, take a snapshot of the entire system memory while it is still running before shutting down the compromised system. The memory snapshot may assist in finding the attack vector used by the ransomware and help locate cryptographic material that can be used to decrypt the data.
- **Move compromised system offline.** Taking the compromised system offline helps prevent the further spread of the ransomware to prevent further damage to the organization's data.
- **Backup compromised system.** Make backups of all data storage associated with the compromised system to prevent further damage to the data in case data recovery attempts fail.

- **Block network access to any identified C2 servers used by ransomware.** Without access, ransomware would typically not be able to mount a secure encryption scheme. This response makes a data recovery attempt more plausible.
- **Identify the attack vector.** Recall emails suspected of carrying the ransomware attack throughout the enterprise network to prevent further spreading of the ransomware.
- **Restrict write access to network storage.** Restricting write access to network-accessible data storage until the network is clear of ransomware helps to protect the data and prevent the further spread of the ransomware.
- **Notify authorities.** Consider notifying appropriate authorities to help with the investigation.

4.3 System Configuration

While it is unlikely that systems can be completely ransomware-proofed, good system management can reduce the likelihood of a successful attack. The two biggest infection vectors for ransomware are email and websites. The likelihood of infection through either of these vectors can be reduced but not completely eliminated.

For email, the most important system configuration to use for reducing attacks is implementing robust filtering. The fewer spam and malicious email messages that are delivered, the less likely that users will open a malicious attachment or click a malicious link.

Other methods of decreasing the email attack surface include blocking executable attachments and using text-only email. A lot of ransomware is distributed as an executable. Stripping these from emails means that even if a malicious email passes a filter, infection is prevented since the payload was not delivered. Unfortunately, this approach does not help when ransomware is delivered as a Microsoft Office-type file containing macros. Text-only email prevents users from clicking on malicious links or downloading external content that might deliver a ransomware executable, though this does not prevent users from copying the URL into their browsers.

It is more difficult to prevent infection from malicious or compromised but otherwise legitimate websites. At a system level, keeping your web browser updated and using an ad blocker are the best options. Disabling Flash and Java also reduces the attack surface, but many organizations need these for day-to-day business.

More general methods for preventing malware are related to system maintenance and permissions. It is crucial to apply available operating system and software patches. Even if ransomware does not exploit software or operating system vulnerabilities, it may be delivered with an exploit kit that does.

There are many permission-related practices that help prevent or limit the impact of a ransomware attack. First, and most important, is limiting administrative rights on devices. Many exploit kits and some ransomware require permissions to install components or make system changes. As the ransomware starts with the permission of the current user, limiting those privileges can halt infection. Users should not be logging in with administrative rights for day-to-day activities and should revert to standard user permissions as soon as any administrative function is complete.

Other permission-related practices include restricting user write capabilities, preventing execution from user directories, whitelisting applications, and limiting access to network storage or shares. Some ransomware requires write access to specific file paths to install or execute. Limiting the write permission to a small number of directories (e.g., User/Documents and User/Downloads) means those ransomware variants can’t successfully carry out their actions. Removing execution permission within those directories can also prevent ransomware

executables from actually running. Many organizations have a limited set of applications that are used to conduct business. Having a whitelist-only policy for applications applied to systems should prevent any non-whitelisted application, including ransomware, from executing. Overall, permission-related practices are intended mostly for the containment of impact and to help limit spreading. Additionally, with the new data exfiltration behavior ransomware has implemented, organizations should encrypt data at rest to help mitigate the potential data breach threat now associated with some ransomware families.

Many ransomware variants no longer scan only the local drive to find files. They also are not limited to mapped drives but have the capability to find any connected network storage or share. To prevent encryption, these devices should require an explicit login at each access. Furthermore, the number of devices directly accessible through something like File Explorer should be limited. While it may add some burden to users, using an alternate method (e.g., SSH) to access these devices can decrease the likelihood that their contents are encrypted if any user is infected.

Using anti-malware programs is a more active way to protect against ransomware. These programs can be either file-based or behavior-based. File-based anti-malware or antivirus programs can quarantine or delete known ransomware variants delivered via email or website. Behavior-based anti-malware programs monitor application behavior and either (1) halts it if it starts acting as malware or (2) executes the file in a sandbox to check for malicious behavior before making the file fully available to end users. Unfortunately, the first option is usually too late because the ransomware usually encrypts several files before it is identified as ransomware. The second option also has drawbacks; the application may not run long enough in the sandbox to be flagged as malicious or the ransomware may use sandbox-evading techniques so that it does not seem malicious if it detects that it is executing in a sandboxed environment.

A more novel prevention method can help protect files against variants of ransomware that only encrypt files with certain file extensions—file extension mapping. This method involves creating a list of file extensions that are not currently used (check www.file-extensions.org), naming files using only those extensions, and using operating system file extension management to assign those extensions to the appropriate associated program. For example, an organization could use .ourworddocs instead of .docx for all Microsoft Word documents. This approach can help protect important files from encryption when other prevention methods have failed. Unfortunately, there are now several ransomware variants that encrypt all files unless the extension is on a whitelist.

Table 3: Ransomware Prevention Methods[4]

Prevention Method	What It Helps With	Limitations
Use a spam filter	Reduces the likelihood of successful email-based attacks	Is ineffective against ransomware spread unwittingly by legitimate users; filters miss some portion of spam, especially spear phishing attempts
Block executable email attachments	Prevents ransomware executables from being delivered directly in email	Does not prevent infection from malicious URLs embedded in the message

[4] CERT Division researchers created this table.

Prevention Method	What It Helps With	Limitations
Use text-only email	Prevents accidental clicks of malicious URLs	Does not prevent users from copying/pasting malicious URLs
Use an ad-blocker	Prevents infection delivery via web exploits	Only stops infections from blocked ads, not regular webpages or whitelisted ad services
Update the OS and other software	Prevents ransomware from using remedied vulnerabilities	Is ineffective against unpatched vulnerabilities
Limit administrator rights	Limits installation and the ability to make system changes	Only works if the user is not logged in with admin privileges and running services/applications can't be exploited to elevate privilege
Restrict user write permissions	Prevents infections that write to restricted directories and limits encryption	Only works if user is not logged in with administrative privileges and running services/applications can't be exploited to elevate privilege; unrestricted directories still vulnerable
Disallow execution within user directories	Prevents ransomware/exploit kits from running when downloaded as an executable	Only works if the executable file ends up in a protected user directory
Restrict applications to a whitelist	Limits execution	Is ineffective if the ransomware exploited is an allowed application
Map file extensions	Hides files from ransomware that looks for specific file extensions	Is ineffective against ransomware that encrypts hard drives or all files but those in a whitelist
Limit connected storage and shares	Contains infection and limits its encryption	Only helps if the user remains disconnected while the machine is infected; is often impractical
Use file-based anti-malware	Quarantines/removes known ransomware variants	Is ineffective against new and rare variants
Use behavior-based anti-malware	Stops known ransomware behavior	If using endpoint anti-malware, the ransomware is already executing when the behavior is identified as malicious; if using sandbox anti-malware, the behavior may be different than when it is run on the endpoint and consequently the ransomware is not caught
Encrypt data at rest	Protects organization's information from ransomware's data exfiltration behavior	Does not protect against a ransomware attack; ransomware can still encrypt your encrypted data.

4.4 Network Configuration

While much can be done at the system level to stop and prevent ransomware, there is not much that can be done at the network level. For preventing initial infection, good general security practices can help. Firewalls that implement whitelisting or robust blacklisting can lessen the likelihood of successful web-based malware downloads and may prevent ransomware from connecting to C2 servers.

Firewalls should limit or completely block RDP and other remote management services. Vigorous spam lists and other spam-detection techniques can prevent most attack-laden emails from being delivered to users' inboxes. Limiting the file extensions that can be delivered by email can mitigate potential infections that could occur in phishing emails that pass the filtering.

Preventing the spread of ransomware from an infected internal host is more difficult. There are tools to detect malware behaving as a worm, which is behavior exhibited by several ransomware families. However, these tools are not likely to completely stop the spread, since it takes time to identify the behavior. Limiting the networked devices accessible from end-user devices can limit the spread of the infection.

Simply "hiding" network devices is not sufficient to prevent their access by ransomware. (G. 2016) If the device can be seen with a scan or something like File Explorer, the ransomware can see it too. If the active user of the compromised system has permission on the device, it is likely that the device will be compromised. If something like Active Directory is set up to not require users to directly log into every device, the ransomware operates under the active user without hindrance. This access means that if the user has write permissions, the ransomware can encrypt, replicate, and exfiltrate data. Even if there is no single sign-on type service, the ransomware or the range of encryption spreads to any device the user is currently logged into, anywhere there is a "Remember me" setting, and any system where the user logs in before realizing the infection.

The most effective way to prevent spreading is to disconnect infected devices (and those suspected of being infected) from the network as soon as possible. These devices include not only wired connections, but Wi-Fi and Bluetooth connections.

5 Conclusion

Ransomware is a major threat. As the attack landscape rapidly expands toward healthcare, government, education, and larger enterprises, the complexity of ransomware technologies also evolves. The new behavior of data exfiltration and public disclosure of victim data demonstrates the adapting nature of ransomware developers' ability to pivot to increase ransom payments. Additionally, the advent of RaaS is an advancement that provides a platform for non-technical threat actors to utilize for their own ransomware campaigns and another avenue for ransomware developers to profit.

Organizations need to minimize their attack surface to the threat of ransomware. The best way to mitigate against ransomware is to sustain frequent offline backups of all data, which minimizes data loss and increases the likelihood of not paying ransomware operators. Additionally, regular active monitoring, patching, and security awareness training helps protect against ransomware infections.

From the perspective of an enterprise-level attack, losing data due to a ransomware attack is not the only reason that this malware seems to be so effective in collecting a ransom. When operations are disrupted, large organizations are pressured to find a solution as quickly as possible. Additionally, even though some organizations can avoid paying the ransom by utilizing data backups to restore their systems, they are now victims of data breaches when the ransomware publishes exfiltrated data due to ransom non-payment.

Despite all of these preventative measures, successful ransomware attacks should be expected. Organizations should consider the steps in how to prepare for an attack and what to do when triaging a ransomware incident. Every reactive step taken has consequences. Isolating the incident, preserving the compromised state, and rapidly restoring normal operations should be the highest priority.

Ransomware is using industry standard encryption algorithms that makes data recovery difficult without the decryption key. While the decision to pay or not pay ransoms is a controversial topic, the victim of compromise has other recovery options. Negotiations with attackers regarding ransom amounts and deadline extensions appear to be a common occurrence. Ultimately, it is up to the individual organization to determine whether or not to pay, but the FBI recommendation is not to pay the ransom to the attackers. (Halpern 2019)

Regulatory authorities and law enforcement are trying to stay on top of the ransomware problem and various guidelines were recently released to help organizations tackle the problem. This report attempted to expand threat awareness from the technical perspective and set the stage for a robust, thorough, and comprehensive response to ransomware.

6 Appendix

6.1 FuxSocy

YARA Rule(s): (Malpedia 2020)

```
rule win_fuxsocy_w0 {
   meta:
      author = "Stephan Simon <stephan.simon@binarydefense.com>"
      date = "2019-10-24"
      description = "A ransomware tweeted about by @malwrhunterteam"
      modified = "2019-10-24"
      reference = "https://twitter.com/malwrhunterteam/status/1187360440734625798"
      tlp = "WHITE"
      malpedia_reference = "https://malpedia.caad.fkie.fraunhofer.de/details/win.fuxsocy"
      malpedia_version = "20191031"
      malpedia_license = "CC BY-SA 4.0"
      malpedia_sharing = "TLP:WHITE"

   strings:
      $n1 = "FuxSocy_Evaluated" wide
      $n2 = "FuxSocy_InstallPlace" wide
      $n3 = "FuxSocy_Instance" wide

      $s1 = "{RAND}" wide
      $s2 = "\\x*x.exe" wide
      $s3 = "%.4d-%.2d-%.2dT%.2d:%.2d:%.2d" wide
      $s4 = "PT1M" wide
      $s5 = "PT0S" wide
      $s6 = "/d /c taskkill /f /pid %d > NUL & ping -n 1 127.0.0.1 > NUL & del \"%s\" > NUL & exit" wide
      $s7 = "/d /c start \"\" \"%s\"" wide
      $s8 = "Win32_ShadowCopy.ID='%s'" wide
      $s9 = "SuperHidden" wide
      $s10 = "ShowSuperHidden" wide
      $s11 = "Shell.IPC.%s" wide
      $s12 = "\\StringFileInfo\\%04x%04x\\%s" wide

   condition:
      filesize <= 100KB and
      (1 of ($n*) or 4 of ($s*))
}
```

Tox contact:
AD049F565435C774D2A7D0A96FC2CC2E4AB5D6B860AEB52F2B1F6A01BB2682104F1361981FDE

Changes file extensions using pattern:

LDLddD

Associated Files: (Woods 2019)

Trojan.Win32.Magniber.4!c

Generic.Ransom.Magniber.8486CAD0

Trojan.Ransom.Filecoder

Trojan-Ransom.Win32.Encoder.fsc

FuxSocy ENCRYPTOR.dll

_readme.txt

readme.txt

Associated Registry Keys: (Carballo 2019)

Run\RunOnce

Hashes [5,6]:

c6866a33a75b9c6c1d90e76729d6879206c7786f323fbbf9d0552c7b037fa87c

6.2 GlobeImposter/GlobeImposter 2.0

YARA Rule(s): (Malpedia 2020)

```
rule win_globeimposter_auto {
   meta:
      author = "Felix Bilstein - yara-signator at cocacoding dot com"
      date = "2019-11-26"
      version = "1"
      description = "autogenerated rule brought to you by yara-signator"
      tool = "yara-signator 0.1a"
      malpedia_reference = "https://malpedia.caad.fkie.fraunhofer.de/details/win.globeimposter"
      malpedia_version = "20190204"
      malpedia_license = "CC BY-NC-SA 4.0"
      malpedia_sharing = "TLP:WHITE"

   /* DISCLAIMER
    * The strings used in this rule have been automatically selected from the
    * disassembly of memory dumps and unpacked files, using yara-signator.
    * The code and documentation / approach will be published in the near future here:
    * https://github.com/fxb-cocacoding/yara-signator
```

```
    * As Malpedia is used as data source, please note that for a given
    * number of families, only single samples are documented.
    * This likely impacts the degree of generalization these rules will offer.
    * Take the described generation method also into consideration when you
    * apply the rules in your use cases and assign them confidence levels.
    */

strings:
    $sequence_0 = { 48 89442418 8b84245c060000 8b4008 8b0488 8b4c2418 }
        // n = 6, score = 400
        //   48                | dec               eax
        //   89442418          | mov               dword ptr [esp + 0x18], eax
        //   8b84245c060000    | mov               eax, dword ptr [esp + 0x65c]
        //   8b4008            | mov               eax, dword ptr [eax + 8]
        //   8b0488            | mov               eax, dword ptr [eax + ecx*4]
        //   8b4c2418          | mov               ecx, dword ptr [esp + 0x18]

    $sequence_1 = { 0ff4e0 0fd4cb 0f6e7618 0ff4f0 }
        // n = 4, score = 400
        //   0ff4e0            | pmuludq           mm4, mm0
        //   0fd4cb            | paddq             mm1, mm3
        //   0f6e7618          | movd              mm6, dword ptr [esi + 0x18]
        //   0ff4f0            | pmuludq           mm6, mm0

    $sequence_2 = { 8bc2 c1e808 23c5 c1e108 }
        // n = 4, score = 400
        //   8bc2              | mov               eax, edx
        //   c1e808            | shr               eax, 8
        //   23c5              | and               eax, ebp
        //   c1e108            | shl               ecx, 8

    $sequence_3 = { 85ff 7452 8bef 8bf0 8b06 8d7604 }
        // n = 6, score = 400
        //   85ff              | test              edi, edi
        //   7452              | je                0x54
        //   8bef              | mov               ebp, edi
        //   8bf0              | mov               esi, eax
        //   8b06              | mov               eax, dword ptr [esi]
        //   8d7604            | lea               esi, [esi + 4]

    $sequence_4 = { 7424 53 8d45fc 50 56 }
        // n = 5, score = 400
        //   7424              | je                0x26
        //   53                | push              ebx
        //   8d45fc            | lea               eax, [ebp - 4]
```

```
        //   50              | push            eax
        //   56              | push            esi

    $sequence_5 = { 3bfb 7504 33c0 eb5f 8b4308 }
        // n = 5, score = 400
        //   3bfb            | cmp             edi, ebx
        //   7504            | jne             6
        //   33c0            | xor             eax, eax
        //   eb5f            | jmp             0x61
        //   8b4308          | mov             eax, dword ptr [ebx + 8]

    $sequence_6 = { 8928 41 8d4014 3b4e6c 7cf5 }
        // n = 5, score = 400
        //   8928            | mov             dword ptr [eax], ebp
        //   41              | inc             ecx
        //   8d4014          | lea             eax, [eax + 0x14]
        //   3b4e6c          | cmp             ecx, dword ptr [esi + 0x6c]
        //   7cf5            | jl              0xfffffff7

    $sequence_7 = { ff9638010000 85c0 7404 6acc }
        // n = 4, score = 400
        //   ff9638010000    | call            dword ptr [esi + 0x138]
        //   85c0            | test            eax, eax
        //   7404            | je              6
        //   6acc            | push            -0x34

    $sequence_8 = { e8???????? ff7608 e8???????? 59 83660400 83660800 c70601000000 }
        // n = 7, score = 400
        //   e8????????      |
        //   ff7608          | push            dword ptr [esi + 8]
        //   e8????????      |
        //   59              | pop             ecx
        //   83660400        | and             dword ptr [esi + 4], 0
        //   83660800        | and             dword ptr [esi + 8], 0
        //   c70601000000    | mov             dword ptr [esi], 1

    $sequence_9 = { ff5014 33c0 eb05 b800afffff }
        // n = 4, score = 400
        //   ff5014          | call            dword ptr [eax + 0x14]
        //   33c0            | xor             eax, eax
        //   eb05            | jmp             7
        //   b800afffff      | mov             eax, 0xffffaf00

condition:
    7 of them
```

```
}

rule win_globeimposter_g0 {
   meta:
      author = "Slavo Greminger, SWITCH-CERT"
      contributions = "Daniel Plohmann"
      malpedia_reference = "https://malpedia.caad.fkie.fraunhofer.de/details/win.globeimposter"
      malpedia_version = "20171130"
      malpedia_license = "CC BY-NC-SA 4.0"
      malpedia_sharing = "TLP:GREEN"

   strings:
      $str_cert   = "CertificatesCheck" wide
      $str_read   = "Read___ME.html" wide
      $str_ID     = "{{IDENTIFIER}}"
      $str_rsa    = "rsa_encrypt"
      $str_cmd1   = "vssadmin.exe Delete Shadows /All /Quiet"
      $str_cmd2   = "reg delete \"HKEY_CURRENT_USER\\Software\\Microsoft\\Terminal Server "
      $str_cmd3   = "for /F \"tokens=*\" %1 in ('wevtutil.exe el') DO wevtutil.exe cl \"%1"

   condition:
      5 of them
}
```

Extensions Appended to Encrypted File Names: (New Jersey Cybersecurity & Communications Integration Cell 2016)

.crypt, .pizdec, .FIX, .keepcalm, .vdul, .2cXpCihgsVxB3, .medal, .paycyka, .wallet, .3ncrypt3d, .skunk, .BRT92, .HAPP, .707, s1crypt, .au1crypt, .p1crypt, .GOTHAM, .crypt, .rose, .ocean, .Mixi, .725, .726, .help, .sea, .mtk118, .492, .astra, .coded, .txt, .ACTUM, .GRAFF, .JEEP, .BONUM, .GRANNY, .LEGO, .D2550A49BF52DFC23F2C013C5, .rumblegoodboy, .zuzya, .UNLIS, .0402, .Trump, .ReaGAN, .C8B089F, .needdecrypt, .write_on_email, .clinTON, .BUSH, .911, .apk .arena, .crypted!, .DREAM, .suddentax, .Nutella, .emilysupp

Associated IP addresses:

185.98.7.180

103.198.0.2

Associated URL:

www[.]zhaksylyk[.]kz

psoeiras[.]net

hxxp://n224ezvhg4sgyamb[.]onion/sup[.]php

topyzscsu5poprxy[.]onion[.]link

promultis[.]it•185.81.1[.]156 HTTP 288 GET/hg65fyJHG??JnqxSiUgE=JnqxSiUgE

trombositting[.]org•91.214.114[.]209 HTTP 295 GET /af/hg65fyJHG?JnqxSiUgE=JnqxSiUgE

hxxps://n224ezvhg4sgyamb.onion[.]link/efwdaq.php

huhighwfn4jihtlz[.]onion/ap

Bitcoin Address:

1MVMkqWS66JySLtkCdkmq9Hg4Y2TibR19N5

Associated Registry Keys: (Zhang 2017)

RunOnce\CertificatesCheck

Delete Shadows /All /Quiet

Associated Emails: (New Jersey Cybersecurity & Communications Integration Cell 2016)

server5[@]mailfence[.]com

keepcalmpls[@]india[.]com

support24[@]india[.]com

support24_02[@]india[.]com

happydaayz[@]aol[.]com

strongman[@]india[.]com

file_free[@]protonmail[.]com

koreajoin69[@]tutanota[.]com

Decoder_master[@]aol[.]com

Decoder_master[@]india[.]com

legosfilos[@]aol[.]com

crazyfoot_granny[@]aol[.]com

Ronald_Reagan[@]derpymail[.]org

Bill_Clinton[@]derpymail[.]org

George_Bush[@]derpymail[.]org

paradisecity[@]cock[.]li

dream_dealer[@]aol[.]com

Hashes: (Zhang 2017) (VirusTotal 2020) (Abrams, New .DOC GlobeImposter Ransomware Variant Malspam Campaign Underway 2017)

e9378336cf81b38bf456a7f1f74580781d6cf423cbad43eb516a8b6707ff2e4c

3a9d5976fbf41daf80f0eb9e6b7aadcece52a82fe9609984ef7f8ea166048547

15e8c986c4602c61a474b51d250e03d5bb178eabc8c5a82a242c1a0fa2227704

10aa60f4757637b6b934c8a4dff16c52a6d1d24297a5fffdf846d32f55155be0

a1cde05bb37cecfecc2ccfb57807d2db66393d73c6e88e129507ffcb70f0ba2e

72ddceebe717992c1486a2d5a5e9e20ad331a98a146d2976c943c983e088f66b

34c57b265c5f3a143ee334b3a651dc76207729b3889ec9959e0224ee640332bc

7bc1c0b67e76b761128ffc478554858a09aa6e5fbb7e57f1f58b3066f6c228fc

6.3 LockerGoga

YARA Rule(s): (Malpedia 2020)

```
rule win_lockergoga_w0 {
   meta:
      description = "Detects LockerGoga ransomware binaries"
      author = "Florian Roth"
      reference = "https://www.nrk.no/norge/skreddersydd-dobbeltangrep-mot-hydro-1.14480202"
      license = "https://creativecommons.org/licenses/by-nc/4.0/"
      date = "2019-03-19"
      hash1 = "c97d9bbc80b573bdeeda3812f4d00e5183493dd0d5805e2508728f65977dda15"
      hash2 = "7bcd69b3085126f7e97406889f78ab74e87230c11812b79406d723a80c08dd26"
      hash3 = "bdf36127817413f625d2625d3133760af724d6ad2410bea7297ddc116abc268f"
      malpedia_reference = "https://malpedia.caad.fkie.fraunhofer.de/details/win.lockergoga"
      malpedia_version = "20190320"
      malpedia_license = "CC BY-NC-SA 4.0"
      malpedia_sharing = "TLP:WHITE"
   strings:
      $x1 = "\\.(doc|dot|wbk|docx|dotx|docb|xlm|xlsx|xltx|xlsb|xlw|ppt|pot|pps|pptx|potx|ppsx|sldx|pdf)" wide
      $x2 = "|[A-Za-z]:\\cl.log" wide
      $x4 = "\\crypto-locker\\" ascii
      $xc1 = { 00 43 00 6F 00 6D 00 70 00 61 00 6E 00 79 00 4E
        00 61 00 6D 00 65 00 00 00 00 00 4D 00 6C 00 63
        00 72 00 6F 00 73 00 6F 00 66 00 74 }
      $xc2 = { 00 2E 00 6C 00 6F 00 63 00 6B 00 65 00 64 00 00
        00 20 46 41 49 4C 45 44 20 00 00 00 00 20 00 00
        00 20 75 6E 6B 6E 6F 77 6E 20 65 78 63 65 70 74
```

```
        69 6F 6E }
      $rn1 = "This may lead to the impossibility of recovery of the certain files." wide
   condition:
      1 of ($x*) or $rn1
}

rule win_lockergoga_auto {

   meta:
      author = "Felix Bilstein - yara-signator at cocacoding dot com"
      date = "2019-11-26"
      version = "1"
      description = "autogenerated rule brought to you by yara-signator"
      tool = "yara-signator 0.1a"
      malpedia_reference = "https://malpedia.caad.fkie.fraunhofer.de/details/win.lockergoga"
      malpedia_version = "20190204"
      malpedia_license = "CC BY-NC-SA 4.0"
      malpedia_sharing = "TLP:WHITE"

   /* DISCLAIMER
    * The strings used in this rule have been automatically selected from the
    * disassembly of memory dumps and unpacked files, using yara-signator.
    * The code and documentation / approach will be published in the near future here:
    * https://github.com/fxb-cocacoding/yara-signator
    * As Malpedia is used as data source, please note that for a given
    * number of families, only single samples are documented.
    * This likely impacts the degree of generalization these rules will offer.
    * Take the described generation method also into consideration when you
    * apply the rules in your use cases and assign them confidence levels.
    */

   strings:
      $sequence_0 = { eb05 e8???????? 83c404 8b95f4feffff 32c0 8bfa c745fc10000000 }
         // n = 7, score = 400
         //   eb05               | jmp               7
         //   e8????????           |
         //   83c404             | add               esp, 4
         //   8b95f4feffff         | mov               edx, dword ptr [ebp - 0x10c]
         //   32c0               | xor               al, al
         //   8bfa               | mov               edi, edx
         //   c745fc10000000       | mov               dword ptr [ebp - 4], 0x10

      $sequence_1 = { e8???????? 8bf0 eb02 33f6 8b3d???????? 85ff 7431 }
         // n = 7, score = 400
         //   e8????????           |
```

```
    //   8bf0              | mov             esi, eax
    //   eb02              | jmp             4
    //   33f6              | xor             esi, esi
    //   8b3d????????      |
    //   85ff              | test            edi, edi
    //   7431              | je              0x33

    $sequence_2 = { 50 e8???????? 8d4da4 c645fc18 e8???????? 8d4dd4 c645fc19 }
    // n = 7, score = 400
    //   50                | push            eax
    //   e8????????        |
    //   8d4da4            | lea             ecx, [ebp - 0x5c]
    //   c645fc18          | mov             byte ptr [ebp - 4], 0x18
    //   e8????????        |
    //   8d4dd4            | lea             ecx, [ebp - 0x2c]
    //   c645fc19          | mov             byte ptr [ebp - 4], 0x19

    $sequence_3 = { ff7518 8b08 ff7514 ff7510 8b9184000000 8bc8 ff750c }
    // n = 7, score = 400
    //   ff7518            | push            dword ptr [ebp + 0x18]
    //   8b08              | mov             ecx, dword ptr [eax]
    //   ff7514            | push            dword ptr [ebp + 0x14]
    //   ff7510            | push            dword ptr [ebp + 0x10]
    //   8b9184000000      | mov             edx, dword ptr [ecx + 0x84]
    //   8bc8              | mov             ecx, eax
    //   ff750c            | push            dword ptr [ebp + 0xc]

    $sequence_4 = { d1e9 0bce 8bf2 c1e61f 894804 83ef01 75e8 }
    // n = 7, score = 400
    //   d1e9              | shr             ecx, 1
    //   0bce              | or              ecx, esi
    //   8bf2              | mov             esi, edx
    //   c1e61f            | shl             esi, 0x1f
    //   894804            | mov             dword ptr [eax + 4], ecx
    //   83ef01            | sub             edi, 1
    //   75e8              | jne             0xffffffea

    $sequence_5 = { 89758c eb09 56 8d4d88 e8???????? 8d856cffffff 50 }
    // n = 7, score = 400
    //   89758c            | mov             dword ptr [ebp - 0x74], esi
    //   eb09              | jmp             0xb
    //   56                | push            esi
    //   8d4d88            | lea             ecx, [ebp - 0x78]
    //   e8????????        |
    //   8d856cffffff      | lea             eax, [ebp - 0x94]
```

```
        //   50                 | push              eax

    $sequence_6 = { e8???????? 8b0e 8d45a0 50 8b7114 8bce ff15???????? }
        // n = 7, score = 400
        //   e8????????           |
        //   8b0e                | mov               ecx, dword ptr [esi]
        //   8d45a0              | lea               eax, [ebp - 0x60]
        //   50                  | push              eax
        //   8b7114              | mov               esi, dword ptr [ecx + 0x14]
        //   8bce                | mov               ecx, esi
        //   ff15????????         |

    $sequence_7 = { 5b e8???????? c9 c23400 55 8bec 83ec30 }
        // n = 7, score = 400
        //   5b                  | pop               ebx
        //   e8????????           |
        //   c9                  | leave
        //   c23400              | ret               0x34
        //   55                  | push              ebp
        //   8bec                | mov               ebp, esp
        //   83ec30              | sub               esp, 0x30

    $sequence_8 = { ffb57cffffff 8d851cffffff 56 52 50 e8???????? 83c414 }
        // n = 7, score = 400
        //   ffb57cffffff        | push              dword ptr [ebp - 0x84]
        //   8d851cffffff        | lea               eax, [ebp - 0xe4]
        //   56                  | push              esi
        //   52                  | push              edx
        //   50                  | push              eax
        //   e8????????           |
        //   83c414              | add               esp, 0x14

    $sequence_9 = { c745ec0f000000 c645d800 897598 897594 e8???????? 8b45a0 0245a4 }
        // n = 7, score = 400
        //   c745ec0f000000      | mov               dword ptr [ebp - 0x14], 0xf
        //   c645d800            | mov               byte ptr [ebp - 0x28], 0
        //   897598              | mov               dword ptr [ebp - 0x68], esi
        //   897594              | mov               dword ptr [ebp - 0x6c], esi
        //   e8????????           |
        //   8b45a0              | mov               eax, dword ptr [ebp - 0x60]
        //   0245a4              | add               al, byte ptr [ebp - 0x5c]

    condition:
        7 of them
}
```

Associated Files: (Trend Micro 2019) (Neumann and Natvig 2019)

README_LOCKED.txt

README-NOW.txt

%Program Files%

%ProgramData%

%System Root%\Recycle Bin

%System Root%\Boot

Associated Registry Keys: (Trend Micro 2019)

(HKEY_CURRENT_USER\SOFTWARE\Microsoft\RestartManager\Session00{01-20}]

Associated Emails: (Neumann and Natvig 2019)

AbbsChevis@protonmail.com

IjuqodiSunovib98@o2.pl

CottleAkela@protonmail.com

QyavauZehyco1994@o2.pl

DutyuEnugev89@o2.pl

SchreiberEleonora@protonmail.com

PhanthavongsaNeveyah@protonmail.com

AperywsQaroci@o2.pl

RomanchukEyla@protonmail.com

CouwetIzotofo@o2.pl

SuzuMcpherson@protonmail.com

AsuxidOruraep1999@o2.pl

MayarChenot@protonmail.com

QicifomuEjijika@o2.pl

SayanWalsworth96@protonmail.com

RezawyreEdipi1998@o2.pl

DharmaParrack@protonmail.com

wyattpettigrew8922555@mail.com

Hashes: (Trend Micro 2019) (VirusTotal 2020) (Neumann and Natvig 2019)

c97d9bbc80b573bdeeda3812f4d00e5183493dd0d5805e2508728f65977dda15

88d149f3e47dc337695d76da52b25660e3a454768af0d7e59c913995af496a0f

eda26a1cd80aac1c42cdbba9af813d9c4bc81f6052080bc33435d1e076e75aa0

ba15c27f26265f4b063b65654e9d7c248d0d651919fafb68cb4765d1e057f93f

7bcd69b3085126f7e97406889f78ab74e87230c11812b79406d723a80c08dd26

C3d334cb7f6007c9ebee1a68c4f3f72eac9b3c102461d39f2a0a4b32a053843a

6.4 SamSam

YARA Rule(s): (GitHub 2020)

```
rule SAmSAmRansom2016 {
    meta:
      author = "Christiaan Beek"
      date = "2018-01-25"
    strings:
      $x1 = "Could not list processes locking resource. Failed to get size of result." fullword wide
      $s2 = "Could not list processes locking resource." fullword wide
      $s3 = "samsam.del.exe" fullword ascii
      $s4 = "samsam.exe" fullword wide
      $s5 = "RM_UNIQUE_PROCESS" fullword ascii
      $s6 = "KillProcessWithWait" fullword ascii
      $s7 = "killOpenedProcessTree" fullword ascii
      $s8 = "RM_PROCESS_INFO" fullword ascii
      $s9 = "Exception caught in process: {0}" fullword wide
      $s10 = "Could not begin restart session.  Unable to determine file locker." fullword wide
      $s11 = "samsam.Properties.Resources.resources" fullword ascii
      $s12 = "EncryptStringToBytes" fullword ascii
      $s13 = "recursivegetfiles" fullword ascii
      $s14 = "RSAEncryptBytes" fullword ascii
      $s15 = "encryptFile" fullword ascii
      $s16 = "samsam.Properties.Resources" fullword wide
      $s17 = "TSSessionId" fullword ascii
      $s18 = "Could not register resource." fullword wide
      $s19 = "<recursivegetfiles>b__0" fullword ascii
      $s20 = "create_from_resource" fullword ascii
      $op0 = { 96 00 e0 00 29 00 0b 00 34 23 }
      $op1 = { 96 00 12 04 f9 00 34 00 6c 2c }
      $op2 = { 72 a5 0a 00 70 a2 06 20 94 }
```

```
    condition:
      ( uint16(0) == 0x5a4d and
        filesize < 700KB and
        pe.imphash() == "f34d5f2d4577ed6d9ceec516c1f5a744" and ( 1 of ($x*) and 4 of them ) and all of
($op*)
      ) or ( all of them )
  }

  rule SamSam_Ransomware_Latest
  {
    meta:
      description = "Latest SamSA ransomware samples"
      author = "Christiaan Beek"
      reference = "http://blog.talosintelligence.com/2018/01/samsam-evolution-continues-netting-
over.html"
      date = "2018-01-23"
    strings:
      $s1 = "bedf08175d319a2f879fe720032d11e5" fullword wide
      $s2 = "ksdghksdghkddgdfgdfgfd" fullword ascii
      $s3 = "osieyrgvbsgnhkflkstesadfakdhaksjfgyjqqwgjrwgehjgfdjgdffg" fullword ascii
      $s4 = "5c2d376c976669efaf9cb107f5a83d0c" fullword wide
      $s5 = "B917754BCFE717EB4F7CE04A5B11A6351EEC5015" fullword ascii
      $s6 = "f99e47c1d4ccb2b103f5f730f8eb598a" fullword wide
      $s7 = "d2db284217a6e5596913e2e1a5b2672f" fullword wide
      $s8 = "0bddb8acd38f6da118f47243af48d8af" fullword wide
      $s9 = "f73623dcb4f62b0e5b9b4d83e1ee4323" fullword wide
      $s10 = "916ab48e32e904b8e1b87b7e3ced6d55" fullword wide
      $s11 = "c6e61622dc51e17195e4df6e359218a2" fullword wide
      $s12 = "2a9e8d549af13031f6bf7807242ce27f" fullword wide
      $s13 = "e3208957ad76d2f2e249276410744b29" fullword wide
      $s14 = "b4d28bbd65da97431f494dd7741bee70" fullword wide
      $s15 = "81ee346489c272f456f2b17d96365c34" fullword wide
      $s16 = "94682debc6f156b7e90e0d6dc772734d" fullword wide
      $s17 = "6943e17a989f11af750ea0441a713b89" fullword wide
      $s18 = "b1c7e24b315ff9c73a9a89afac5286be" fullword wide
      $s19 = "90928fd1250435589cc0150849bc0cff" fullword wide
      $s20 = "67da807268764a7badc4904df351932e" fullword wide

      $op0 = { 30 01 00 2b 68 79 33 38 68 34 77 65 36 34 74 72 }
      $op1 = { 01 00 b2 04 00 00 01 00 84 }
      $op2 = { 68 09 00 00 38 66 00 00 23 55 53 00 a0 6f 00 00 }

    condition:
      ( uint16(0) == 0x5a4d and
```

```
        filesize < 100KB and
        pe.imphash() == "f34d5f2d4577ed6d9ceec516c1f5a744" and ( 8 of them ) and all of ($op*)
      ) or ( all of them )
  }
```

Hashes: (Malwarebytes Labs 2018) (VirusTotal 2020)

036071786d7db553e2415ec2e71f3967baf51bdc31d0a640aa4afb87d3ce3050

88e344977bf6451e15fe202d65471a5f75d22370050fe6ba4dfa2c2d0fae7828

ffef0f1c2df157e9c2ee65a12d5b7b0f1301c4da22e7e7f3eac6b03c6487a626

58ef87523184d5df3ed1568397cea65b3f44df06c73eadeb5d90faebe4390e3e

0f2c5c39494f15b7ee637ad5b6b5d00a3e2f407b4f27d140cd5a821ff08acfac

a763ed678a52f77a7b75d55010124a8fccf1628eb4f7a815c6d635034227177e

6.5 MedusaLocker

YARA Rule(s): (ditekshen 2019)

```
rule MedusaLocker {
   meta:
      author = "ditekshen"
      description = "MedusaLocker Ransomware Payload"
      cape_type = "MedusaLocker Payload"
  strings:
                  $s1 = "\\MedusaLockerInfo\\MedusaLockerProject\\MedusaLocker\\Release\\Medu-
saLocker.pdb" ascii
                  $s2 = "SOFTWARE\\Medusa" wide
                  $s3 = "{8761ABBD-7F85-42EE-B272-A76179687C63}" fullword wide
                  $s4 = "{3E5FC7F9-9A51-4367-9063-A120244FBEC7}" fullword wide
                  $s5 = "{6EDD6D74-C007-4E75-B76A-E5740995E24C}" fullword wide
                  $s6 = "vssadmin.exe delete" wide nocase
                  $s7 = "bcdedit.exe /set {default}" wide
                  $s8 = "wbadmin delete systemstatebackup" wide nocase
                  $s9 = ".exe,.dll,.sys,.ini,.lnk,.rdp,.encrypted" fullword ascii
                  $s10 = "[LOCKER] " wide
condition:
                  uint16(0) == 0x5a4d and 6 of them
}
```

Associated URLs: (NCFTA 2019)

zjoxyw5mkacojk5ptn2iprkivg5clow72mjkyk5ttubzxprjjnwapkad[.]onion/pay

hxxps://pbs[.]twimg[.]com/media/Dn4vwaRW0AY-tUu[.]jpg:large

Associated IP Address: (NCFTA 2019)

40.81.94[.]65:123

Associated Files: (Abrams, MedusaLocker Ransomware Wants Its Share of Your Money 2019)

File Path: "C:\Windows\SysWOW64\cmd.exe" /c vssadmin.exe delete shadows /all /quiet & bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no & wbadmin delete catalog -quiet & wmic shadowcopy delete

File Path: cmd.exe /c schtasks.exe /create /sc onstart /tn "NEMTY_<FIELD>_"/tr "C:\Users\user\AdobeUpdate.exe"

%UserProfile%\AppData\Roaming\svchostt.exe

Associated Registry Keys: (Walter, How MedusaLocker Ransomware Aggressively Targets Remote Hosts 2019)

HKCU\SOFTWARE\Medusa

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\ "EnableLinkedConnections" = 1

Associated Emails: (Abrams, MedusaLocker Ransomware Wants Its Share of Your Money 2019) (NCFTA 2019)

elzmflqxj[@]tutanota[.]de

helpdesk_nemty[@]aol[.]com

sambolero[@]tutanoa[.]com

rightcheck[@]cock[.]li

folieloi[@]protonmail[.]com

ctorsenoria[@]tutnota[.]com

mrromber[@]tutnota[.]com

decoder83540[@]protonmail[.]com

decoder83540[@]cock[.]li

Hash: (Walter, How MedusaLocker Ransomware Aggressively Targets Remote Hosts 2019)

dde3c98b6a370fb8d1785f3134a76cb465cd663db20dffe011da57a4de37aa95

72f9d83c7852f2247e24113cb379fff71c06f095910726ea79479f16aac6070f

6.6 Ryuk

YARA Rule(s): (Malpedia 2020)

```
rule win_ryuk_auto {

   meta:
      author = "Felix Bilstein - yara-signator at cocacoding dot com"
      date = "2019-11-26"
      version = "1"
      description = "autogenerated rule brought to you by yara-signator"
      tool = "yara-signator 0.1a"
      malpedia_reference = "https://malpedia.caad.fkie.fraunhofer.de/details/win.ryuk"
      malpedia_version = "20190204"
      malpedia_license = "CC BY-NC-SA 4.0"
      malpedia_sharing = "TLP:WHITE"

   /* DISCLAIMER
    * The strings used in this rule have been automatically selected from the
    * disassembly of memory dumps and unpacked files, using yara-signator.
    * The code and documentation / approach will be published in the near future here:
    * https://github.com/fxb-cocacoding/yara-signator
    * As Malpedia is used as data source, please note that for a given
    * number of families, only single samples are documented.
    * This likely impacts the degree of generalization these rules will offer.
    * Take the described generation method also into consideration when you
    * apply the rules in your use cases and assign them confidence levels.
    */

   strings:
      $sequence_0 = { c3 4053 4883ec20 8bc1 498bd8 33c9 4c8bda }
         // n = 7, score = 600
         //   c3                  | dec             eax
         //   4053                | mov             eax, ebx
         //   4883ec20            | dec             eax
         //   8bc1                | add             esp, 0x30
         //   498bd8              | pop             ebx
         //   33c9                | ret
         //   4c8bda              | dec             eax

      $sequence_1 = { 415e c3 4053 4883ec20 488bd9 33c9 48890b }
         // n = 7, score = 600
         //   415e                | mov             dword ptr [esp + 8], ebx
         //   c3                  | dec             eax
         //   4053                | mov             dword ptr [esp + 0x10], ebp
         //   4883ec20            | ret
         //   488bd9              | inc             eax
         //   33c9                | push            ebx
         //   48890b              | dec             eax
```

```
$sequence_2 = { ff15???????? b803000000 eb05 b805000000 }
   // n = 4, score = 600
   //   ff15????????        |
   //   b803000000          | sub             esp, 0x20
   //   eb05                | mov             eax, ecx
   //   b805000000          | dec             ecx

$sequence_3 = { 4983c8ff e8???????? 488bc3 4883c430 5b c3 48895c2408 }
   // n = 7, score = 600
   //   4983c8ff            | dec             ecx
   //   e8????????          |
   //   488bc3              | or              eax, 0xffffffff
   //   4883c430            | dec             eax
   //   5b                  | mov             eax, ebx
   //   c3                  | dec             eax
   //   48895c2408          | add             esp, 0x30

$sequence_4 = { e8???????? e8???????? b9e8030000 ff15???????? }
   // n = 4, score = 600
   //   e8????????          |
   //   e8????????          |
   //   b9e8030000          | jmp             0xc
   //   ff15????????        |

$sequence_5 = { b809000000 e9???????? 4533c9 4533c0 }
   // n = 4, score = 600
   //   b809000000          | pop             ebx
   //   e9????????          |
   //   4533c9              | ret
   //   4533c0              | dec             eax

$sequence_6 = { 4c8d05???????? 488d15???????? 488d0d???????? ff15???????? 85c0 }
   // n = 5, score = 600
   //   4c8d05????????      |
   //   488d15????????      |
   //   488d0d????????      |
   //   ff15????????        |
   //   85c0                | dec             eax

$sequence_7 = { eb04 4883c020 4883c428 c3 48895c2408 57 4883ec30 }
   // n = 7, score = 600
   //   eb04                | inc             eax
   //   4883c020            | push            ebx
   //   4883c428            | dec             eax
```

```
    //   c3              | sub             esp, 0x20
    //   48895c2408        | mov             eax, ecx
    //   57              | dec             ecx
    //   4883ec30          | mov             ebx, eax

    $sequence_8 = { 50 e8???????? 50 57 6800000030 e8???????? }
    // n = 6, score = 400
    //   50              | push            eax
    //   e8????????        |
    //   50              | push            eax
    //   57              | push            edi
    //   6800000030        | push            0x30000000
    //   e8????????        |

    $sequence_9 = { 6800000030 e8???????? 83c408 85c0 }
    // n = 4, score = 400
    //   6800000030        | push            0x30000000
    //   e8????????        |
    //   83c408            | add             esp, 8
    //   85c0              | test            eax, eax

    $sequence_10 = { b900000030 2bc1 50 51 e8???????? 59 }
    // n = 6, score = 400
    //   b900000030        | mov             ecx, 0x30000000
    //   2bc1              | sub             eax, ecx
    //   50              | push            eax
    //   51              | push            ecx
    //   e8????????        |
    //   59              | pop             ecx

    $sequence_11 = { 8b4508 2d00000030 50 6800000030 e8???????? }
    // n = 5, score = 400
    //   8b4508            | mov             eax, dword ptr [ebp + 8]
    //   2d00000030        | sub             eax, 0x30000000
    //   50              | push            eax
    //   6800000030        | push            0x30000000
    //   e8????????        |

    $sequence_12 = { 8d45f0 64a300000000 8965e8 c745fc00000000 6800000030 }
    // n = 5, score = 400
    //   8d45f0            | lea             eax, [ebp - 0x10]
    //   64a300000000      | mov             dword ptr fs:[0], eax
    //   8965e8            | mov             dword ptr [ebp - 0x18], esp
    //   c745fc00000000    | mov             dword ptr [ebp - 4], 0
    //   6800000030        | push            0x30000000
```

```
$sequence_13 = { c745fc00000000 6800000030 e8???????? 83c404 85c0 7454 8b4508 }
   // n = 7, score = 400
   //   c745fc00000000      | mov             dword ptr [ebp - 4], 0
   //   6800000030          | push            0x30000000
   //   e8????????          |
   //   83c404              | add             esp, 4
   //   85c0                | test            eax, eax
   //   7454                | je              0x56
   //   8b4508              | mov             eax, dword ptr [ebp + 8]

$sequence_14 = { 6823110030 51 e8???????? 8bf0 }
   // n = 4, score = 300
   //   6823110030          | push            0x30001123
   //   51                  | push            ecx
   //   e8????????          |
   //   8bf0                | mov             esi, eax

$sequence_15 = { 754f b90b010000 66398818000030 7541 8b4508 b900000030 2bc1 }
   // n = 7, score = 300
   //   754f                | jne             0x51
   //   b90b010000          | mov             ecx, 0x10b
   //   66398818000030      | cmp             word ptr [eax + 0x30000018], cx
   //   7541                | jne             0x43
   //   8b4508              | mov             eax, dword ptr [ebp + 8]
   //   b900000030          | mov             ecx, 0x30000000
   //   2bc1                | sub             eax, ecx

$sequence_16 = { 8d0c49 c1e103 51 6a00 }
   // n = 4, score = 200
   //   8d0c49              | lea             ecx, [ecx + ecx*2]
   //   c1e103              | shl             ecx, 3
   //   51                  | push            ecx
   //   6a00                | push            0

$sequence_17 = { 6a00 ffd6 6800400000 6a40 }
   // n = 4, score = 200
   //   6a00                | push            0
   //   ffd6                | call            esi
   //   6800400000          | push            0x4000
   //   6a40                | push            0x40

$sequence_18 = { 57 ff15???????? b80c000000 5f }
   // n = 4, score = 200
   //   57                  | push            edi
```

```
    //   ff15????????        |
    //   b80c000000          | mov             eax, 0xc
    //   5f                  | pop             edi

    $sequence_19 = { 2bf0 33c0 66890473 83ffff }
    // n = 4, score = 200
    //   2bf0                | sub             esi, eax
    //   33c0                | xor             eax, eax
    //   66890473            | mov             word ptr [ebx + esi*2], ax
    //   83ffff              | cmp             edi, -1

    $sequence_20 = { 6a00 ff15???????? 6a00 6a02 6a03 6a00 }
    // n = 6, score = 200
    //   6a00                | sub             eax, ecx
    //   ff15????????        |
    //   6a00                | push            eax
    //   6a02                | push            ecx
    //   6a03                | add             esp, 4
    //   6a00                | test            eax, eax

    $sequence_21 = { c20400 8bff 55 8bec b8ffff0000 83ec18 }
    // n = 6, score = 200
    //   c20400              | push            eax
    //   8bff                | push            ecx
    //   55                  | mov             eax, dword ptr [ebp + 8]
    //   8bec                | mov             ecx, 0x30000000
    //   b8ffff0000          | sub             eax, ecx
    //   83ec18              | push            eax

    $sequence_22 = { 6a0a c1e818 51 50 }
    // n = 4, score = 200
    //   6a0a                | push            0xa
    //   c1e818              | shr             eax, 0x18
    //   51                  | push            ecx
    //   50                  | push            eax

    $sequence_23 = { ff15???????? b808000000 5f 5e }
    // n = 4, score = 200
    //   ff15????????        |
    //   b808000000          | mov             eax, 8
    //   5f                  | pop             edi
    //   5e                  | pop             esi

    $sequence_24 = { 7e19 ff734c 4e 0fbe0406 }
    // n = 4, score = 100
```

```
        // 7e19              | jle             0x1b
        // ff734c            | push            dword ptr [ebx + 0x4c]
        // 4e                | dec             esi
        // 0fbe0406          | movsx           eax, byte ptr [esi + eax]

    $sequence_25 = { 0f849e000000 833e00 0f8595000000 6a44 e8???????? }
        // n = 5, score = 100
        // 0f849e000000      | je              0xa4
        // 833e00            | cmp             dword ptr [esi], 0
        // 0f8595000000      | jne             0x9b
        // 6a44              | push            0x44
        // e8????????        |
S
  condition:
    7 of them
}

import "pe"

rule MAL_Ryuk_Ransomware {
  meta:
    description = "Detects strings known from Ryuk Ransomware"
    author = "Florian Roth"
    reference = "https://research.checkpoint.com/ryuk-ransomware-targeted-campaign-break/"
    date = "2018-12-31"
    hash1 = "965884f19026913b2c57b8cd4a86455a61383de01dabb69c557f45bb848f6c26"
    hash2 = "b8fcd4a3902064907fb19e0da3ca7aed72a7e6d1f94d971d1ee7a4d3af6a800d"
  strings:
    $x1 = "/v \"svchos\" /f" fullword wide
    $x2 = "\\Documents and Settings\\Default User\\finish" fullword wide
    $x3 = "\\users\\Public\\finish" fullword wide
    $x4 = "lsaas.exe" fullword wide
    $x5 = "RyukReadMe.txt" fullword wide
  condition:
    uint16(0) == 0x5a4d and filesize < 400KB and (
      pe.imphash() == "4a069c1abe5aca148d5a8fdabc26751e" or
      pe.imphash() == "dc5733c013378fa418d13773f5bfe6f1" or
      1 of them
    )
}
```

Hashes: (VirusTotal 2020) (Cohen and Herzog 2018)

8d3f68b16f0710f858d8c1d2c699260e6f43161a5510abb0e7ba567bd72c965b

3012f472969327d5f8c9dac63b8ea9c5cb0de002d16c120a6bba4685120f58b4

b8e463789a076b16a90d1aae73cea9d3880ac0ead1fd16587b8cd79e37a1a3d8

9b86a50b36aea5cc4cb60573a3660cf799a9ec1f69a3d4572d3dc277361a0ad2

113af75f13547be184822f1268f984b79f35965a1b1f963d23b50a09741b0aec

1455091954ecf9ccd6fe60cb8e982d9cfb4b3dc8414443ccfdfc444079829d56

c51024bb119211c335f95e731cfa9a744fcdb645a57d35fb379d01b7dbdd098e

23f8aa94ffb3c08a62735fe7fee5799880a8f322ce1d55ec49a13a3f85312db2

0300a6367a69ac95bd0dd2e8572a0aa44540898f9d12ba7bfbd25a0ed51cfe2c

113af75f13547be184822f1268f984b79f35965a1b1f963d23b50a09741b0aec

1455091954ecf9ccd6fe60cb8e982d9cfb4b3dc8414443ccfdfc444079829d56

23f8aa94ffb3c08a62735fe7fee5799880a8f322ce1d55ec49a13a3f85312db2

3012f472969327d5f8c9dac63b8ea9c5cb0de002d16c120a6bba4685120f58b4

8b0a5fb13309623c3518473551cb1f55d38d8450129d4a3c16b476f7b2867d7d

8d3f68b16f0710f858d8c1d2c699260e6f43161a5510abb0e7ba567bd72c965b

9b86a50b36aea5cc4cb60573a3660cf799a9ec1f69a3d4572d3dc277361a0ad2

b8e463789a076b16a90d1aae73cea9d3880ac0ead1fd16587b8cd79e37a1a3d8

c51024bb119211c335f95e731cfa9a744fcdb645a57d35fb379d01b7dbdd098e

d083ecc1195602c45d9cb75a08c395ad7d2b0bf73d7e70e2fc76101c780dd38f

dd0691992d947366f1b9caf2acc1fec951f761a39ca3863e81bc2c3fb5efd415

de708f2807f96938e5eb0295d5ebfee870b34dd0cb70708607d4e1adf767ce7b

df3d947eb72a7b10f90222ae5a0aab0aade66f0bc1d3812c1b0366e6e8456591

e75622957decf1594c2cbe726ff0aaba4a509dab7b77721d3db16977f224ae4a

fe55650d8b1b78d5cdb4ad94c0d7ba7052351630be9e8c273cc135ad3fa81a75

fe909d18cf0fde089594689f9a69fbc6d57b69291a09f3b9df1e9b1fb724222b

6.7 Nemty

YARA Rule(s): (Malpedia 2020)

```
rule win_nemty_auto {
   meta:
      author = "Felix Bilstein - yara-signator at cocacoding dot com"
```

```
        date = "2019-11-26"
        version = "1"
        description = "autogenerated rule brought to you by yara-signator"
        tool = "yara-signator 0.1a"
        malpedia_reference = "https://malpedia.caad.fkie.fraunhofer.de/details/win.nemty"
        malpedia_version = "20190204"
        malpedia_license = "CC BY-NC-SA 4.0"
        malpedia_sharing = "TLP:WHITE"

    /* DISCLAIMER
     * The strings used in this rule have been automatically selected from the
     * disassembly of memory dumps and unpacked files, using yara-signator.
     * The code and documentation / approach will be published in the near future here:
     * https://github.com/fxb-cocacoding/yara-signator
     * As Malpedia is used as data source, please note that for a given
     * number of families, only single samples are documented.
     * This likely impacts the degree of generalization these rules will offer.
     * Take the described generation method also into consideration when you
     * apply the rules in your use cases and assign them confidence levels.
     */

    strings:
        $sequence_0 = { 0302 8945c8 8b4dfc 8b511c }
            // n = 4, score = 100
            //   0302               | add             eax, dword ptr [edx]
            //   8945c8             | mov             dword ptr [ebp - 0x38], eax
            //   8b4dfc             | mov             ecx, dword ptr [ebp - 4]
            //   8b511c             | mov             edx, dword ptr [ecx + 0x1c]

        $sequence_1 = { 8b11 6a30 8b4dc8 8b4220 ffd0 }
            // n = 5, score = 100
            //   8b11               | mov             edx, dword ptr [ecx]
            //   6a30               | push            0x30
            //   8b4dc8             | mov             ecx, dword ptr [ebp - 0x38]
            //   8b4220             | mov             eax, dword ptr [edx + 0x20]
            //   ffd0               | call            eax

        $sequence_2 = { 8b4df0 c7410400000000 8b55f0 c70228234200 8b45f0 c70038234200 }
            // n = 6, score = 100
            //   8b4df0             | mov             ecx, dword ptr [ebp - 0x10]
            //   c7410400000000     | mov             dword ptr [ecx + 4], 0
            //   8b55f0             | mov             edx, dword ptr [ebp - 0x10]
            //   c70228234200       | mov             dword ptr [edx], 0x422328
            //   8b45f0             | mov             eax, dword ptr [ebp - 0x10]
            //   c70038234200       | mov             dword ptr [eax], 0x422338
```

```
$sequence_3 = { 8908 8b55fc 8b421c c70000000000 }
   // n = 4, score = 100
   //   8908              | mov             dword ptr [eax], ecx
   //   8b55fc            | mov             edx, dword ptr [ebp - 4]
   //   8b421c            | mov             eax, dword ptr [edx + 0x1c]
   //   c70000000000      | mov             dword ptr [eax], 0

$sequence_4 = { c645fc01 8d8d3cffffff e8???????? 8d8dc0feffff 51 8b4dd4 e8???????? }
   // n = 7, score = 100
   //   c645fc01          | mov             byte ptr [ebp - 4], 1
   //   8d8d3cffffff      | lea             ecx, [ebp - 0xc4]
   //   e8????????        |
   //   8d8dc0feffff      | lea             ecx, [ebp - 0x140]
   //   51                | push            ecx
   //   8b4dd4            | mov             ecx, dword ptr [ebp - 0x2c]
   //   e8????????        |

$sequence_5 = { 52 8b451c 50 8b4df8 51 6834264200 8d55f0 }
   // n = 7, score = 100
   //   52                | push            edx
   //   8b451c            | mov             eax, dword ptr [ebp + 0x1c]
   //   50                | push            eax
   //   8b4df8            | mov             ecx, dword ptr [ebp - 8]
   //   51                | push            ecx
   //   6834264200        | push            0x422634
   //   8d55f0            | lea             edx, [ebp - 0x10]

$sequence_6 = { 7516 83c8ff e9???????? c745e4d80e8302 a1???????? eb5e ff775c }
   // n = 7, score = 100
   //   7516              | jne             0x18
   //   83c8ff            | or              eax, 0xffffffff
   //   e9????????        |
   //   c745e4d80e8302    | mov             dword ptr [ebp - 0x1c], 0x2830ed8
   //   a1????????        |
   //   eb5e              | jmp             0x60
   //   ff775c            | push            dword ptr [edi + 0x5c]

$sequence_7 = { eb92 c745e800000000 eb09 8b55e8 83c201 8955e8 817de88fd44500 }
   // n = 7, score = 100
   //   eb92              | jmp             0xffffff94
   //   c745e800000000    | mov             dword ptr [ebp - 0x18], 0
   //   eb09              | jmp             0xb
   //   8b55e8            | mov             edx, dword ptr [ebp - 0x18]
   //   83c201            | add             edx, 1
```

```
        //  8955e8          | mov             dword ptr [ebp - 0x18], edx
        //  817de88fd44500  | cmp             dword ptr [ebp - 0x18], 0x45d48f

    condition:
        1 of them
}
```

Registry Keys: (Trend Micro 2019)
[HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\RunOnce\wextract_cleanup0]

Email Addresses: (GoldSparrow, Nemty Ransomware 2020)

elzmflqxi[@]tutanota[.]de

helpdesk.nemty[@]aol[.]com

Associated URLs: (GoldSparrow, Nemty Ransomware 2020) (Acronis 2019)

hxxp://api-db-ip[.]com/v2/free/{IP address}/countryName

zjoxyw5mkacojk5ptn2iprkivg5clow72mjkyk5ttubzxprjjnwapkad.onion/pay

hxxp://www.reg.com

hxxp://autohotkey.com

hxxps://dist.torproject.org/torbrowser/8.5.4/tor-win32-0.4.0.5.zip

Hashes: (Ilascu, Nemty Ransomware Now Spreads via Trik Botnet 2019) (VirusTotal 2020) (Acronis 2019)

a127323192abed93aed53648d03ca84de3b5b006b641033eb46a520b7a3c16fc

2c41b93add9ac5080a12bf93966470f8ab3bde003001492a10f63758867f2a88

f3e743c919c1deaf5108d361c4ff610187606f450fabda0bea3786d4063511b1

3d852ca618763ced2e280f0c0079e804935b70dcd4adc3912c2e2b3965e196c4

6.8 MegaCortex

YARA Rule(s): (Ilascu, Nemty Ransomware Gets Distribution from RIG Exploit Kit 2019)

```
rule win_megacortex_auto {

    meta:
        author = "Felix Bilstein - yara-signator at cocacoding dot com"
        date = "2019-11-26"
        version = "1"
        description = "autogenerated rule brought to you by yara-signator"
        tool = "yara-signator 0.1a"
        malpedia_reference = "https://malpedia.caad.fkie.fraunhofer.de/details/win.megacortex"
```

```
    malpedia_version = "20190204"
    malpedia_license = "CC BY-NC-SA 4.0"
    malpedia_sharing = "TLP:WHITE"

/* DISCLAIMER
 * The strings used in this rule have been automatically selected from the
 * disassembly of memory dumps and unpacked files, using yara-signator.
 * The code and documentation / approach will be published in the near future here:
 * https://github.com/fxb-cocacoding/yara-signator
 * As Malpedia is used as data source, please note that for a given
 * number of families, only single samples are documented.
 * This likely impacts the degree of generalization these rules will offer.
 * Take the described generation method also into consideration when you
 * apply the rules in your use cases and assign them confidence levels.
 */

strings:
    $sequence_0 = { 8b10 2bc1 8d4aff f7d9 1bc9 23c1 8b4da8 }
        // n = 7, score = 200
        //   8b10               | mov             edx, dword ptr [eax]
        //   2bc1               | sub             eax, ecx
        //   8d4aff             | lea             ecx, [edx - 1]
        //   f7d9               | neg             ecx
        //   1bc9               | sbb             ecx, ecx
        //   23c1               | and             eax, ecx
        //   8b4da8             | mov             ecx, dword ptr [ebp - 0x58]

    $sequence_1 = { 56 50 e8???????? 83c418 8bf0 8d855cfeffff 50 }
        // n = 7, score = 200
        //   56                 | push            esi
        //   50                 | push            eax
        //   e8????????         |
        //   83c418             | add             esp, 0x18
        //   8bf0               | mov             esi, eax
        //   8d855cfeffff       | lea             eax, [ebp - 0x1a4]
        //   50                 | push            eax

    $sequence_2 = { 8d4dcc 85c0 b801000000 0f44d0 8955cc eb19 8b5708 }
        // n = 7, score = 200
        //   8d4dcc             | lea             ecx, [ebp - 0x34]
        //   85c0               | test            eax, eax
        //   b801000000         | mov             eax, 1
        //   0f44d0             | cmove           edx, eax
        //   8955cc             | mov             dword ptr [ebp - 0x34], edx
        //   eb19               | jmp             0x1b
```

```
        //   8b5708              | mov               edx, dword ptr [edi + 8]

    $sequence_3 = { c7430800000000 c745fc00000000 8db520ffffff c7851cffffff01000000 8bfe 8bce
898d14ffffff }
        // n = 7, score = 200
        //   c7430800000000      | mov               dword ptr [ebx + 8], 0
        //   c745fc00000000      | mov               dword ptr [ebp - 4], 0
        //   8db520ffffff        | lea               esi, [ebp - 0xe0]
        //   c7851cffffff01000000    | mov    dword ptr [ebp - 0xe4], 1
        //   8bfe                | mov               edi, esi
        //   8bce                | mov               ecx, esi
        //   898d14ffffff        | mov               dword ptr [ebp - 0xec], ecx

    $sequence_4 = { d1c1 894df4 8bc8 8bc6 c1c105 034df4 33c7 }
        // n = 7, score = 200
        //   d1c1                | rol               ecx, 1
        //   894df4              | mov               dword ptr [ebp - 0xc], ecx
        //   8bc8                | mov               ecx, eax
        //   8bc6                | mov               eax, esi
        //   c1c105              | rol               ecx, 5
        //   034df4              | add               ecx, dword ptr [ebp - 0xc]
        //   33c7                | xor               eax, edi

    $sequence_5 = { c745fc01000000 e8???????? 59 e8???????? c3 55 8bec }
        // n = 7, score = 200
        //   c745fc01000000      | mov               dword ptr [ebp - 4], 1
        //   e8????????          |
        //   59                  | pop               ecx
        //   e8????????          |
        //   c3                  | ret
        //   55                  | push              ebp
        //   8bec                | mov               ebp, esp

    $sequence_6 = { 8b4308 8d0488 83c0fc 0f1f8000000000 833800 7508 83e804 }
        // n = 7, score = 200
        //   8b4308              | mov               eax, dword ptr [ebx + 8]
        //   8d0488              | lea               eax, [eax + ecx*4]
        //   83c0fc              | add               eax, -4
        //   0f1f8000000000      | nop               dword ptr [eax]
        //   833800              | cmp               dword ptr [eax], 0
        //   7508                | jne               0xa
        //   83e804              | sub               eax, 4

    $sequence_7 = { 0bf0 c783c400000000000000 0fb645f1 0bc8 89b304010000 0fb645f2 c1e108 }
        // n = 7, score = 200
```

```
        //   0bf0              | or              esi, eax
        //   c783c400000000000000    | mov    dword ptr [ebx + 0xc4], 0
        //   0fb645f1           | movzx           eax, byte ptr [ebp - 0xf]
        //   0bc8              | or              ecx, eax
        //   89b304010000        | mov             dword ptr [ebx + 0x104], esi
        //   0fb645f2           | movzx           eax, byte ptr [ebp - 0xe]
        //   c1e108             | shl             ecx, 8

    $sequence_8 = { d1e8 f6410401 8945fc 740a b909000000 8d3409 eb13 }
        // n = 7, score = 200
        //   d1e8              | shr             eax, 1
        //   f6410401           | test            byte ptr [ecx + 4], 1
        //   8945fc             | mov             dword ptr [ebp - 4], eax
        //   740a              | je              0xc
        //   b909000000         | mov             ecx, 9
        //   8d3409             | lea             esi, [ecx + ecx]
        //   eb13              | jmp             0x15

    $sequence_9 = { 1bd2 2bf8 23ca b8abaaaa2a f7ef 03ce 890b }
        // n = 7, score = 200
        //   1bd2              | sbb             edx, edx
        //   2bf8              | sub             edi, eax
        //   23ca              | and             ecx, edx
        //   b8abaaaa2a         | mov             eax, 0x2aaaaaab
        //   f7ef              | imul            edi
        //   03ce              | add             ecx, esi
        //   890b              | mov             dword ptr [ebx], ecx

    condition:
        7 of them
}
```

Contact emails: (Abrams, New Megacortex Ransomware Changes Windows Passwords, Threatens to Publish Data 2019)

redacted@redacted.com

Registry Keys: (GoldSparrow, MegaCortex Ransomware 2019)

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System

IP Addresses:

89.105.198.28

Email Addresses: (Trend Micro 2019) (GoldSparrow, MegaCortex Ransomware 2019)

shawhart154295[@]mail.com

anderssperry6654818[@]mail.com

borismcfallen[@]mail.com

kirkmansion[@]mail.com

Hashes: (MalwareHunterTeam 2019) (VirusTotal 2020)

77ee63e36a52b5810d3a31e619ec2b8f5794450b563e95e4b446d5d3db4453b2

cafe2f5999d945d907eacd5b9d79d5a3bdfa1777d6f3a32f4564daeeaa1f8477

873aa376573288fcf56711b5689f9d2cf457b76bbc93d4e40ef9d7a27b7be466

6.9 Maze

Yara Rule(s): (bartblaze 2019)

```
rule Maze
{
meta:
description = "Identifies Maze ransomware in memory or unpacked."
        author = "@bartblaze"
        date = "2019-11"
        tlp = "White"

strings:
        $ = "Enc: %s" ascii wide
        $ = "Encrypting whole system" ascii wide
        $ = "Encrypting specified folder in --path parameter..." ascii wide
        $ = "!Finished in %d ms!" ascii wide
        $ = "--logging" ascii wide
        $ = "--nomutex" ascii wide
        $ = "--noshares" ascii wide
        $ = "--path" ascii wide
        $ = "Logging enabled | Maze" ascii wide
        $ = "NO SHARES | " ascii wide
        $ = "NO MUTEX | " ascii wide
        $ = "Encrypting:" ascii wide
        $ = "You need to buy decryptor in order to restore the files." ascii wide
$ = "Dear %s, your files have been encrypted by RSA-2048 and ChaCha algorithms" ascii wide
$ = "%s! Alert! %s! Alert! Dear %s Your files have been encrypted by %s! Attention! %s" ascii wide
        $ = "DECRYPT-FILES.txt" ascii wide fullword

condition:
        5 of them
}
```

Associated URLs: (NCFTA 2019)

hxxp://92.63.8[.]47/content/news/snjw.jsp?a=c6&l=qf8g8tqly

hxxp://92.63.32[.]2/bf.aspx?kwh=j4438k31h

hxxp://92.63.37[.]100/qn.do?juw=aj

hxxp://92.63.194[.]20/jcwgydcca.jspx?mwhi=0v3v3h&ojr=i03rqp4h7&dm=7ghn2a8a1

hxxp://92.63.17[.]245/check/she.do

hxxp://92.63.32[.]55/content/view/ov.jsp?pb=p8&afvb=6cfp5w&i=7f76t2v&au=54p5

hxxp://92.63.11[.]151/webaccess/webauth/pqcadpkov.asp?cf=p4&h=ej&kchk=416

hxxp://92.63.194[.]3/hlbrhcrogk.php?t=k3&i=snti74x&qbfw=wu043276&vjy=0

hxxp://92.63.15[.]8/tracker/xgmuy.jspx?v=s76snl&j=4su31f2i&e=26f2110r&yh=w1a21f3p

hxxp://92.63.29[.]137/doqlgpv.php?rgbf=23tkkklxs

hxxp://92.63.32[.]57/view/b.phtml?tlqv=r088&agf=uj

hxxp://92.63.15[.]56/yrawf.asp?i=0ry&kml=8usd357b&dhj=m7x&lyps=uew8h86

hxxp://92.63.11[.]151/checkout/create/jvjmed.phtml?a=4j

hxxp://92.63.32[.]52/ticket/ciqwisje.jspx?cdgn=005205lg1

hxxp://92.63.15[.]6/withdrawal/ccuc.jsp?l=1pe41u76p7

hxxp://104.168.198[.]208/wordupd.tmp

hxxp://104.168.215[.]54/wordupd.tmp

hxxp://104.168.174[.]32/wordupd_3.0.1.tmp

hxxp://192.119.68[.]225/wordupd1.tmp

hxxp://91.218.114[.]38/ticket/qrerqapv.jsp

hxxp://91.218.114[.]25/check/evjgdec.jspx

hxxp://91.218.114[.]79/archive/eqsuxsjii.aspx

hxxp://91.218.114[.]32/yelwkdt.shtml

hxxp://91.218.114[.]4/webaccess/sepa/aos.action

hxxp://91.218.114[.]79/archive/eqsuxsjii.aspx

hxxp://91.218.114[.]77/withdrawal/wywbiubddg.php

hxxp://91.218.114[.]11/kgtlakrun.phtml

hxxp://91.218.114[.]37/messages/sepa/udopsfgpy.action

hxxp://91.218.114[.]37/messages/sepa/udopsfgpy.action

hxxp://91.218.114[.]31/edit/irtfdlapkb.jsp

hxxp://91.218.114[.]26/post/update/u.asp

hxxps://mazedecrypt[.]top/1dcb0b851e857d00

hxxp://aoacugmutagkwctu[.]onion/1dcb0b851e857d00

hxxp://www.mazenews[.]top

Associated IP Addresses: (Abrams, Maze Ransomware Says Computer Type Determines Ransom Amount 2019)

92.63.32[.]2

92.63.37[.]100

92.63.194[.]20

92.63.17[.]245

92.63.32[.]55

92.63.11[.]151

92.63.194[.]3

92.63.15[.]8

92.63.29[.]137

92.63.32[.]57

92.63.15[.]56

92.63.32[.]52

92.63.8[.]47

92.63.15[.]6

49.51.171[.]58

8.208.76[.]132

146.0.72[.]85

Associated Files: (Abrams, Maze Ransomware Says Computer Type Determines Ransom Amount 2019) (Meskauskas 2019)

File Path: %ProgramData%\foo.dat

File Path: C:\f\ojo\..\..\system32\som\vnt\..\..\wbem\q\wdun\jnw\..\..\..\wmic.exe" shadowcopy delete

DECRYPT-FILES[.]html

%ProgramData%\foo.dat

Associated Emails: (Abrams, Maze Ransomware Says Computer Type Determines Ransom Amount 2019)

filedecryptor[@]nuke[.]Africa

koradec[@]tutanota[.]com

yourrealdecrypt[@]airmail[.]cc

gladkoff1991[@]yandex[.]ru

Hashes: (Abrams, Maze Ransomware Says Computer Type Determines Ransom Amount 2019) (Walter, Maze Ransomware Update: Extorting and Exposing Victims 2020) (VirusTotal 2020)

91514e6be3f581a77daa79e2a4905dcbdf6bdcc32ee0f713599a94d453a26fc1

e8a091a84dd2ea7ee429135ff48e9f48f7787637ccb79f6c3eb42f34588bc684

0b9c99276ed36110afc58b3fb59ada135146180189c25d99618ca5897537ee21

34d05620f964945a59d1da2c76e45c1d214affd42094b2de31d38739a9241970

5badaf28bde6dcf77448b919e2290f95cd8d4e709ef2d699aae21f7bae68a76c

7c03b49d24c948f838b737fb476d57849a1fd6b205f94214bf2a5a3b7a36f17a

8d995102e28d96bb93016e3abbb38b4597b9e497668fa7a0e10cf28db105516b

ecd04ebbb3df053ce4efa2b73912fd4d086d1720f9b410235ee9c1e529ea52a2

6.10 Sodinokibi

YARA Rule(s): (VirusTotal 2020)

```
rule ransomware_sodinokibi {
    meta:
      description = "Using a recently disclosed vulnerability in Oracle WebLogic, criminals use it to install a new variant of ransomware called “Sodinokibi"
      author = "Christiaan Beek | McAfee ATR team"
      date = "2019-05-13"
      hash1 = "95ac3903127b74f8e4d73d987f5e3736f5bdd909ba756260e187b6bf53fb1a05"
      hash2 = "34dffdb04ca07b014cdaee857690f86e490050335291ccc84c94994fa91e0160"
```

```
        hash3 = "0fa207940ea53e2b54a2b769d8ab033a6b2c5e08c78bf4d7dade79849960b54d"
        hash4 = "9b62f917afa1c1a61e3be0978c8692dac797dd67ce0e5fd2305cc7c6b5fef392"
    strings:
        $x1 = "sodinokibi.exe" fullword wide

        $y0 = { 8d 85 6c ff ff ff 50 53 50 e8 62 82 00 00 83 c4 }
        $y1 = { e8 24 ea ff ff ff 75 08 8b ce e8 61 fc ff ff 8b }
        $y2 = { e8 01 64 ff ff ff b6 b0 }
    condition:
        ( uint16(0) == 0x5a4d and filesize < 900KB and pe.imphash() ==
"672b84df309666b9d7d2bc8cc058e4c2" and ( 8 of them ) and all of ($y*)
        ) or ( all of them )
}

rule Sodinokobi
{
    /*
        This rule detects Sodinokobi Ransomware in memory in old samples and perhaps future.
    */
    meta:
        author      = "McAfee ATR team"
        version     = "1.0"
        description = "This rule detect Sodinokobi Ransomware in memory in old samples and perhaps fu-
ture."
    strings:
        $a = { 40 0F B6 C8 89 4D FC 8A 94 0D FC FE FF FF 0F B6 C2 03 C6 0F B6 F0 8A 84 35 FC FE
FF FF 88 84 0D FC FE FF FF 88 94 35 FC FE FF FF 0F B6 8C 0D FC FE FF FF }
        $b = { 0F B6 C2 03 C8 8B 45 14 0F B6 C9 8A 8C 0D FC FE FF FF 32 0C 07 88 08 40 89 45 14 8B
45 FC 83 EB 01 75 AA }
    condition:
        all of them
}
```

Registry Key:

Software\Classes\mscfile\shell\open\command\

IP Addresses: (Cadieux, et al. 2019)

188.166.74[.]218

45.55.211[.]79

130.61.54[.]136

URLs: (Cadieux, et al. 2019)

arg0s-co[.]uk

projectstore[.]guru

Hashes: (Cadieux, et al. 2019) (VirusTotal 2020) (Malwarebytes Labs 2019)

963e31fef7c8db9e002c56ee30fd3cd4b240db466bc23687979e2f161ba5606e

0fa207940ea53e2b54a2b769d8ab033a6b2c5e08c78bf4d7dade79849960b54d

95ac3903127b74f8e4d73d987f5e3736f5bdd909ba756260e187b6bf53fb1a05

34dffdb04ca07b014cdaee857690f86e490050335291ccc84c94994fa91e0160

9b62f917afa1c1a61e3be0978c8692dac797dd67ce0e5fd2305cc7c6b5fef392

References/Bibliography

*URLs are valid as of the publication date of this document.*The No More Ransom Project. 2019. *The No More Ransom Project; DECRYPTION TOOLS.* https://www.nomoreransom.org/en/decryption-tools.html.

Abrams, Lawrence. 2019. *Allied Universal Breached by Maze Ransomware, Stolen Data Leaked.* BleepingComputer. November 21. https://www.bleepingcomputer.com/news/security/allied-universal-breached-by-maze-ransomware-stolen-data-leaked/.

—. 2019. *Elusive MegaCortex Ransomware Found - Here is What We Know.* BleepingComputer. July 19. https://www.bleepingcomputer.com/news/security/elusive-megacortex-ransomware-found-here-is-what-we-know/.

—. 2019. *Exploit Kits Target Windows Users with Ransomware and Trojans.* BleepingComputer. September 9. https://www.bleepingcomputer.com/news/security/exploit-kits-target-windows-users-with-ransomware-and-trojans/.

—. 2019. *FBI Issues Alert For LockerGoga and MegaCortex Ransomware.* BleepingComputer. December 23. https://www.bleepingcomputer.com/news/security/fbi-issues-alert-for-lockergoga-and-megacortex-ransomware/.

—. 2019. *Maze Ransomware Says Computer Type Determines Ransom Amount.* BleepingComputer. May 31. https://www.bleepingcomputer.com/news/security/maze-ransomware-says-computer-type-determines-ransom-amount/.

—. 2019. *MedusaLocker Ransomware Wants Its Share of Your Money.* BleepingComputer. October 22. https://www.bleepingcomputer.com/news/security/medusalocker-ransomware-wants-its-share-of-your-money/.

—. 2017. *New .DOC GlobeImposter Ransomware Variant Malspam Campaign Underway.* BleepingComputer. December 22. https://www.bleepingcomputer.com/news/security/new-doc-globeimposter-ransomware-variant-malspam-campaign-underway/.

—. 2019. *New FuxSocy Ransomware Impersonates the Notorious Cerber.* BleepingComputer. October 25. https://www.bleepingcomputer.com/news/security/new-fuxsocy-ransomware-impersonates-the-notorious-cerber/.

—. 2019. *New Megacortex Ransomware Changes Windows Passwords, Threatens to Publish Data.* BleepingComputer. November 5. https://www.bleepingcomputer.com/news/security/new-megacortex-ransomware-changes-windows-passwords-threatens-to-publish-data/.

—. 2019. *Sodinokibi Ransomware Now Pushed by Exploit Kits and Malvertising.* BleepingComputer. June 24. https://www.bleepingcomputer.com/news/security/sodinokibi-ransomware-now-pushed-by-exploit-kits-and-malvertising/.

Acronis. 2019. *Threat Analysis: Nemty Ransomware and the Fake PayPal Site.* Acronis. October 7. https://www.acronis.com/en-us/blog/posts/threat-analysis-nemty-ransomware-and-fake-paypal-site.

Balaban, David. 2020. *Why CISOs Should Worry About Sodinokibi Ransomware.* CISO MAG. February 6. https://www.cisomag.com/sodinokibi-ransomware/.

bartblaze. 2019. *bartblaze.* GitHub. November 24. https://github.com/Yara-Rules/rules/pull/360/files.

Barth, Bradley. 2019. *MegaCortex ransomware variant threatens data breach, alters credentials.* SC Media. November 8. https://www.scmagazine.com/home/security-news/ransomware/megacortex-ransomware-variant-threatens-data-breach-alters-credentials/.

Bauer, Roderick. 2019. *More People Than Ever Backing Up According to Our Survey.* July 9. https://www.backblaze.com/blog/more-people-than-ever-backing-up-according-to-our-survey/.

Boyd, Christopher. 2019. *SamSam ransomware: what you need to know.* Malwarebytes Labs. November 17. https://blog.malwarebytes.com/cybercrime/2018/05/samsam-ransomware-need-know/.

Cadieux, Pierre, Colin Grady, Jaeson Schultz, and Matt Valites. 2019. *Sodinokibi ransomware exploits WebLogic Server vulnerability.* Cisco Talos. April 30. https://blog.talosintelligence.com/2019/04/sodinokibi-ransomware-exploits-weblogic.html.

Carballo, Anthony. 2019. *Threat Report - FuxSocy Ransomware.* knogin. October 29. https://blog.knogin.com/threat-report-fuxsocy-ransomware.

Cimpanu, Catalin. 2016. *Security Firm Releases Decrypter for Alma Locker Ransomware.* Softpedia. August 25. http://news.softpedia.com/news/security-firm-releases-free-decrypter-for-alma-locker-ransomware-507613.shtml.

Cohen, Itay, and Ben Herzog. 2018. *Ryuk Ransomware: A Targeted Campaign Break-Down.* Check Point Research. August 20. https://research.checkpoint.com/2018/ryuk-ransomware-targeted-campaign-break/.

Coveware. 2019. *Ransom amounts rise 90% in Q1 as Ryuk increases.* Coveware. https://www.coveware.com/blog/2019/4/15/ransom-amounts-rise-90-in-q1-as-ryuk-ransomware-increases?utm_source=emsisoft.

—. 2019. *Ransomware Costs Double in Q4 as Ryuk, Sodinokibi Proliferate.* Coveware. https://www.coveware.com/blog/2020/1/22/ransomware-costs-double-in-q4-as-ryuk-sodinokibi-proliferate.

—. 2019. *What Is SamSam Ransomware and How to Recover and Remove It [Guide].* Coveware. January 14. https://www.coveware.com/blog/samsam-ransomware-recovery-removal-guide.

Cybersecurity and Infrastructure Security Agency (CISA). 2020. *Ransomware.* Cybersecurity and Infrastructure Security Agency (CISA). https://www.us-cert.gov/Ransomware.

—. 2018. *SamSam Ransomware.* Cybersecurity and Infrastructure Security Agency (CISA). December 3. https://www.us-cert.gov/ncas/alerts/AA18-337A.

Cyware Social. 2019. *A Glance At The Ever-evolving Globeimposter Ransomware.* Cyware . October 29. https://cyware.com/news/a-glance-at-the-ever-evolving-globeimposter-ransomware-1ef3c773.

ditekshen. 2019. *CAPE/MedusaLocker.yar at master · ctxis/CAPE · GitHub.* GitHub. October 26. https://github.com/ctxis/CAPE/blob/master/data/yara/CAPE/MedusaLocker.yar.

Emsisoft Malware Lab. 2019. *The State of Ransomware in the US: Report and Statistics 2019.* Emsisoft. December 12. https://blog.emsisoft.com/en/34822/the-state-of-ransomware-in-the-us-report-and-statistics-2019/.

Fakterman, Tom. 2019. *Sodinokibi: The Crown Prince of Ransomware.* Cybereason. August 5. https://www.cybereason.com/blog/the-sodinokibi-ransomware-attack.

G., Steven. 2016. *How does ransomware find unmapped network shares?* Spiceworks. May 4. https://community.spiceworks.com/topic/1594288-how-does-ransomware-find-unmapped-network-shares.

GitHub. 2020. *The world's leading software development platform · GitHub.* GitHub. https://github.com.

GoldSparrow. 2019. *MegaCortex Ransomware.* EnigmaSoft. May 2. https://www.enigmasoftware.com/megacortexransomware-removal/.

—. 2020. *Nemty Ransomware.* EnigmaSoft. February 23. https://www.enigmasoftware.com/nemtyransomware-removal/.

Hall, Gabriel E. 2020. *Sodinokibi creators leak and sell data stolen from organizations.* 2SPYWARE. 3 21. https://www.2-spyware.com/sodinokibi-creators-leak-and-sell-data-stolen-from-organizations.

Halpern, Mollie. 2019. *FBI, This Week: Advocating Against Ransomware Payment Demands.* FBI. August 22. https://www.fbi.gov/audio-repository/ftw-podcast-ransomware-082219.mp3/view.

Hanel, Alexander. 2019. *Big Game Hunting with Ryuk: Another Lucrative Targeted Ransomware.* CrowdStrike. January 10. https://www.crowdstrike.com/blog/big-game-hunting-with-ryuk-another-lucrative-targeted-ransomware/.

Hope, Alicia. 2020. *Ransomware Costs in 2019.* CPO Magazine. January 15. https://www.cpomagazine.com/cyber-security/ransomware-costs-in-2019/.

Ilascu, Ionut. 2019. *Nemty Ransomware Gets Distribution from RIG Exploit Kit.* BleepingComputer. September 3. https://www.bleepingcomputer.com/news/security/nemty-ransomware-gets-distribution-from-rig-exploit-kit/.

—. 2019. *Nemty Ransomware Now Spreads via Trik Botnet.* BleepingComputer. November 4. https://www.bleepingcomputer.com/news/security/nemty-ransomware-now-spreads-via-trik-botnet/.

Infradata. 2019. *What is SamSam Ransomware?* Infradata. https://www.infradata.com/resources/what-is-samsam-ransomware/.

Jareth. 2020. *Ransomware data exfiltration detection and mitigation strategies.* Emsisoft. January 23. https://blog.emsisoft.com/en/35235/ransomware-data-exfiltration-detection-and-mitigation-strategies/.

Kim, Christopher. 2019. *MegaCortex Ransomware.* Infoblox. https://www.infoblox.com/wp-content/uploads/threat-intelligence-report-megacortex-ransomware.pdf.

Klein, Andy. 2015. *Locker: Cryptolocker Progeny Awakens.* Backblaze. June 12. https://www.backblaze.com/blog/locker-cryptolocker-progeny-awakens/.

Kline, Amanda. 2017. *Globe Imposter Ransomware Makes a New Run.* PhishLabs. August 17. https://info.phishlabs.com/blog/globe-imposter-ransomware-makes-a-new-run.

Lopez, Marc Rivero. 2019. *LockerGoga Ransomware Family Used in Targeted Attacks.* McAfee. April 29. https://www.mcafee.com/blogs/other-blogs/mcafee-labs/lockergoga-ransomware-family-used-in-targeted-attacks/.

Malpedia. 2020. *Malpedia.* Fraunhofer FKIE. https://malpedia.caad.fkie.fraunhofer.de.

Malware Guide. 2020. *How To Remove Maze Ransomware And Restore Infected Data.* Malware Guide. https://malware-guide.com/blog/how-to-remove-maze-ransomware-and-restore-infected-data.

Malwarebytes Labs. 2017. *Ransom.GlobeImposter.* Malwarebytes Labs. August 9. https://blog.malwarebytes.com/detections/ransom-globeimposter/.

—. 2019. *Ransom.Ryuk.* January 3. https://blog.malwarebytes.com/detections/ransom-ryuk/.

—. 2019. *Ransom.Sodinokibi.* Malwarebytes. July 18. https://blog.malwarebytes.com/detections/ransom-sodinokibi/.

—. 2018. *SamSam ransomware: controlled distribution for an elusive malware.* Malware Labs. June 19. https://blog.malwarebytes.com/threat-analysis/2018/06/samsam-ransomware-controlled-distribution/.

MalwareHunterTeam. 2019. *MalwareHunterTeam on Twitter.* Twitter. October 16. https://twitter.com/malwrhunterteam/status/1184459976506576896.

Manuel, Jasper, and Joie Salvio. 2019. *LockerGoga: Ransomware Targeting Critical Infrastructure.* Fortinet. April 11. https://www.fortinet.com/blog/threat-research/lockergoga-ransomeware-targeting-critical-infrastructure.html.

McAfee Labs. 2019. *McAfee ATR Analyzes Sodinokibi aka REvil Ransomware-as-a-Service – What The Code Tells Us.* McAfee. October 2. https://www.mcafee.com/blogs/other-blogs/mcafee-labs/mcafee-atr-analyzes-sodinokibi-aka-revil-ransomware-as-a-service-what-the-code-tells-us/.

McAfee. 2019. *Threat Landscape Dashboard Fallout Exploit Kit.* McAfee. https://www.mcafee.com/enterprise/en-us/threat-center/threat-landscape-dashboard/exploit-kits-details.fallout-exploit-kit.html.

—. 2018. *Threat Landscape Dashboard RIG Exploit Kit.* McAfee. https://www.mcafee.com/enterprise/en-us/threat-center/threat-landscape-dashboard/exploit-kits-details.rig-exploit-kit.html.

—. 2019. *Threat Landscape Dashboard Spelevo Exploit Kit.* McAfee. https://www.mcafee.com/enterprise/en-us/threat-center/threat-landscape-dashboard/exploit-kits-details.spelevo-exploit-kit.html.

Meskauskas, Tomas. 2019. *Maze Ransomware Exploiting Exploit Kits.* Security Boulevard. November 8. https://securityboulevard.com/2019/11/maze-ransomware-exploiting-exploit-kits/.

Muncaster, Phil. 2020. *Ransomware Costs May Have Hit $170bn in 2019.* Infosecurity. February 13. https://www.infosecurity-magazine.com/news/ransomware-costs-may-have-hit-170/.

Mundo, Alexandre, and Marc Rivero Lopez . 2020. *Nemty Ransomware – Learning by Doing.* McAfee. April 2. https://www.mcafee.com/blogs/other-blogs/mcafee-labs/nemty-ransomware-learning-by-doing/.

nao_sec. 2019. *Weak Drive-by Download attack with "Radio Exploit Kit".* nao_sec. July 15. https://nao-sec.org/2019/07/weak-dbd-attack-with-radioek.html.

NCFTA. 2020. "FuxSocy."

NCFTA. 2019. "Maze Ransomware."

NCFTA. 2019. "MedusaLocker."

NCFTA. 2020. "Sodinokibi."

NCFTA. 2020. "Top Ten Ransomware Variants."

Neumann, Robert, and Kurt Natvig. 2019. *LockerGoga ransomware - how it works.* Forcepoint. March 22. https://www.forcepoint.com/blog/x-labs/lockergoga-ransomware-how-it-works.

New Jersey Cybersecurity & Communications Integration Cell. 2019. *FuxSocy.* New Jersey Cybersecurity & Communications Integration Cell. https://www.cyber.nj.gov/threat-profiles/ransomware-variants/fuxsocy.

—. 2019. *GlobeImposter.* New Jersey Cybersecurity & Communications Integration Cell. https://www.cyber.nj.gov/threat-profiles/ransomware-variants/globeimposter.

—. 2016. *NJCCIC Threat Profile GlobeImposter.* New Jersey Cybersecurity & Communications Integration Cell. December 28. https://www.cyber.nj.gov/threat-profiles/ransomware-variants/globeimposter.

Norton. 2016. *What is Ransomware?* Norton. July 15. https://www.nortonsecurityonline.com/faq/art/what-is-ransomware/.

Novinson, Michael. 2019. *The 10 Biggest Ransomware Attacks of 2019; 8. Lake City, Fla.* CRN. December 27. https://www.crn.com/slide-shows/security/the-10-biggest-ransomware-attacks-of-2019/4.

—. 2019. *The 10 Biggest Ransomware Attacks of 2019; 9. Jackson County, Ga.* CRN. December 27. https://www.crn.com/slide-shows/security/the-10-biggest-ransomware-attacks-of-2019/3.

Oza, Shyam. 2020. *Ryuk Ransomware — Malware of the Month, January 2020.* Security Boulevard. January 24. https://securityboulevard.com/2020/01/ryuk-ransomware-malware-of-the-month-january-2020/.

Paganini, Pierluigi. 2020. *Security experts uncovered an ongoing campaign delivering Nemty Ransomware via emails disguised as messages from secret lovers.* SecurityAffairs. March 2. https://securityaffairs.co/wordpress/98755/malware/nemty-ransomware-malspam.html.

Shackelford, Scott, and Megan Wade. 2020. *Deal with ransomware the way police deal with hostage situations.* GCN. March 20. https://gcn.com/articles/2020/03/30/ransomware-negotiation.aspx.

Sheridan, Kelly. 2019. *LockerGoga, MegaCortex Ransomware Share Unlikely Traits.* Dark Reading. May 13. https://www.darkreading.com/perimeter/lockergoga-megacortex-ransomware-share-unlikely-traits/d/d-id/1334696.

Spadafora, Anthony. 2020. *FBI: Over $140 million handed over to ransomware attackers.* TechRadar. February 28. https://www.techradar.com/news/fbi-over-dollar140-million-handed-over-to-ransomware-attackers.

The No More Ransom Project. 2019. *The No More Ransom Project.* https://www.nomoreransom.org.

Tiwari, Ravikant, and Alexander Koshelev. 2019. *Taking Deep Dive into Sodinokibi Ransomware.* Acronis. https://www.acronis.com/en-us/articles/sodinokibi-ransomware/.

Trend Micro. 2019. *Ransom.Win32.MEGACORTEX.AD.* Trend Micro. November 9. https://www.trendmicro.com/vinfo/my/threat-encyclopedia/malware/ransom.win32.megacortex.ad.

—. 2019. *Sodinokibi Ransomware Group Adds Malvertising as Delivery Technique.* Trend Micro. June 25. https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/-sodinokibi-ransomware-group-adds-malvertising-as-delivery-technique.

—. 2019. *What You Need to Know About the LockerGoga Ransomware.* Trend Micro. March 20. https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/what-you-need-to-know-about-the-lockergoga-ransomware.

van den Hurk, Frank. 2019. *Nemty 2.2 and 2.3: analysis of their cryptography, and a decryptor for some file types.* Tesorion. 12 20. https://www.tesorion.nl/nemty-2-2-and-2-3-analysis-of-their-cryptography-and-a-decryptor-for-some-file-types/.

Verizon. 2019. *2019 Data Breach Investigations Report.* https://enterprise.verizon.com/resources/reports/dbir/.

VirusTotal. 2020. *VirusTotal.* VirusTotal. https://www.virustotal.com.

Walter, Jim. 2019. *How MedusaLocker Ransomware Aggressively Targets Remote Hosts.* SentinelOne. November 28. https://www.sentinelone.com/blog/how-medusalocker-ransomware-aggressively-targets-remote-hosts/.

—. 2020. *Maze Ransomware Update: Extorting and Exposing Victims.* SentinelOne. April 17. https://labs.sentinelone.com/maze-ransomware-update-extorting-and-exposing-victims/.

Woods, Alice. 2019. *Remove FuxSocy ransomware (Free Guide) - Decryption Methods Included.* 2SPYWARE. October 29. https://www.2-spyware.com/remove-fuxsocy-ransomware.html.

Zhang, Xiaopeng. 2017. *Analysis of New GlobeImposter Ransomware Variant.* Fortinet. August 5. https://www.fortinet.com/blog/threat-research/analysis-of-new-globeimposter-ransomware-variant.html.

Carnegie Mellon University
Software Engineering Institute

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479
Web: www.sei.cmu.edu
Email: info@sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM20-0436