A Comparative Assessment of Obfuscated Ransomware Detection Methods

Sergiu SECHEL
Bucharest University of Economic Studies
sergiu.sechel@gmail.com

Ransomware represents a class of malicious applications that encrypts the files of infected system and demands from victims a payment in cryptocurrency in order to receive the decryption key. The mainstream adoption of cryptocurrencies increased the number of ransomware attack. The outbreaks had risen in complexity and received mass-media attention in 2017 when two destructive campaigns crippled companies and institutions around the world. These outbreaks continue at an accelerated pace even though efforts are made to improve the detection and mitigation of ransomware. The purpose of this research is to assess the efficiency of current malware analysis methods and technologies in the detection of ransomware. The experiments presented here were performed using antivirus engines and dynamic malware analysis against live obfuscated ransomware samples.

Keywords: Malware, Ransomware, Detection Techniques, Malware Analysis, Malware Classification, Mutation, Cybersecurity.

1 Introduction

Ransomware represents a class of malware (malicious applications) that encrypts the files of the infected system and demands from affected users a payment in cryptocurrency in order to receive the decryption key. The idea of a crypto-virus has been around for some time, being first mentioned in research papers like "An “Implementation of Cryptoviral Extortion Using Microsoft's Crypto API” (Young, Yung, Moti, 2005)[1].

Ransomware evolved from another type of malware used to block access to the infected devices or systems and display a message to the user impersonating a state authority (local police) while demanding the user to pay a fine because he was caught performing illegal activities like video piracy, pornography or software piracy. The first major campaign of this type was discovered in 2012 using the ransomware family called “Reveton”[2].

An operational risk that stalled the rapid expansion of ransomware campaigns was the lack of anonymous or secretive mechanisms to receive the ransom without being tracked by the authorities and ultimately arrested. This risk was effectively mitigated with the mass adoption of cryptocurrencies, especially bitcoin.

The first ransomware family that used the “modus operandi” that is now considered standard when we are referring to ransomware was “Cryptolocker”[3]. Since then ransomware campaigns had risen in complexity and received mass-media attention in 2017 when two destructive campaigns crippled companies and institutions around the world. The first major outbreak was known as “Wannacry” in May 2017, with estimated infections of 230,000 computers, in a 3 days timespan, affecting companies and institutions in over 150 countries, including 16 hospitals in the UK. The second major outbreak occurred in 27 June 2017, cause by a ransomware called “NotPetya” [4] which in a 2 days timespan produced estimated damages of 10 billion USD, crippling the transport giant Maersk and companies like Fedex TNT, Mondelez and Reckitt Benckiser.[5]

These outbreaks continue at an accelerated pace even though efforts are made to improve the detection and mitigation of this type of malware. The purpose of this research is to assess the effectiveness of current antivirus detection technologies against obfuscated ransomware.
2 Ransomware characteristics and behavior
From an operational perspective ransomware are a family of malicious applications used to encrypt files and data on various computer systems using strong symmetric and asymmetric cryptographic algorithms like RSA [6] and AES [7]. Upon execution the modern ransomware performs the following main activities, with variations, depending on the ransomware family:
1) Connects to a command-and-control server (C2C) and requires the generation of an asymmetric RSA key pair. After the key pair is generated the ransomware downloads the public key (PubKey) from the C2C server;
2) The ransomware generates a symmetric key (SymKey) for the AES encryption algorithm;
3) The ransomware encrypts the files on the target system using the AES encryption algorithm with the previously generated SymKey;
4) The AES SymKey is encrypted with the PubKey that was previously downloaded from the C2C server;
5) The malware deletes or encrypts the backups and disables any recovery mechanisms present on the system;
6) A ransomware note is generated for the user with instructions on how to receive the private key (PrivKey) required to decrypt the SymKey. The decrypted SymKey will be used by the user to recover the encrypted files.

The generic encryption process is presented in Figure 1. Various ransomware families implement different variations of the encryption process depending on the technical knowledge or capabilities of the malicious actor.

After the encryption process is finished the ransomware will display a message to the user with instructions on how to recover the encrypted files. Usually the instructions require the user to make a cryptocurrency payment (bitcoin or similar) to the attacker in order to obtain the decryption key (PrivKey) as presented in a note generated by the WannaCry ransomware presented in the Figure 18 and a note generated by the TeslaCrypt ransomware is presented in Figure 17.
Some ransomwares require the user to make the payment in a certain amount of time. Trying to delay the countdown timer is not usually a successful strategy because the PrivKey is not hosted on the victim system and as such it can be deleted at any given time by the attacker.

3 The difficulty of ransomware detection
Currently there are several methods employed for malware detection and classification. The most common methods deployed in a wide range of antivirus software products are the following:

a) **signature based detection** – the signature of the suspect code is compared against a database of known malicious signatures;

b) **heuristic detection** – suspect code functionalities are compared against a known malicious functionalities database;

c) **machine learning** - using supervised or unsupervised algorithms a model is trained to identify and classify new specimens of malware based on similar characteristics shared with the training set.

Professionals in the antivirus, forensics and cybersecurity industries use the following methods to detect and classify and analyze suspect code:

a) **static analysis** – the suspect code is analyzed using a disassembler with the purpose to understand the code structure and the code functionalities

b) **dynamic analysis** – the suspect code is executed in a controlled environment and its behavior is analyzed using different tools. The code execution in a debugger or in a sandbox are forms of dynamic analysis.

Ransomwares behave differently than other types of malware, mainly because of their destructive nature. The main purpose of a ransomware is to successfully execute the payload (encryption module) which will proceed to encrypt the files and folders on the infected system [8]. From a stealth perspective some ransomwares are employing different techniques to evade detection until the encryption process is finished, but in general ransomwares don’t employ advanced stealth functionalities because the malware is designed to have a short life span. Another reason why ransomwares don’t employ advanced stealth mechanisms is because once the ransomware’s destructive actions are finished the user will be become aware that the system was infected.

4 Evasion techniques used by ransomware
Malware families are constantly seeking new ways to hide their code, thwart replication, and avoid detection. A recent trend for the delivery of ransomware is the use of the Nullsoft Scriptable Install System (NSIS) with an encrypted payload. The list of the most common families using this technique is diverse and includes Cerber, Locky, Teerac, Crysis, CryptoWall, and CTB-Locker.[9]

The antivirus industry published several research papers describing various obfuscated ransomware samples, ranging from the Loky ransomware analysis released by Avast[10], the recent analysis of the Synack ransomware released by Kaspersky Lab [11] or the analysis of the GandCrab ransomware released VMRay [12]. One common evasion method used by ransomware authors involves the use of packers and crypers:

- **Packer** - is a program that takes the executable as input, and it uses compression to obfuscate the executable's content. This obfuscated content is then stored within the structure of a new executable file; the result is a new executable file (packed program) with obfuscated content on the disk. Upon execution of the packed program, it executes a decompression routine, which extracts the original binary in memory during runtime and triggers the execution.

- **Cryper** - is similar to a packer, but instead of using compression, it uses encryption to obfuscate the executable's content, and the encrypted content is stored in the new executable file. Upon execution of the encrypted program, it runs a decryption
routine to extract the original binary in the memory and then triggers the execution. Packed or crypted ransomware is difficult to be analyzed by antivirus engines or by static analysis, because both the antivirus engine and the analyst are presented with only the packed code of the suspect application. The packing and unpacking process of an executable is presented in the Figure 2.

![Fig. 2. The packing and unpacking process of a PE executable](image)

To demonstrate the difficulty to analyze a packed executable the Microsoft Calculator (calc.exe) was packed with the Themida Packer [13]. The sections of the packed executable were inspected using PE Studio [14]. The sections of the packed executable have less available data for analysis because the code will be unpacked directly in memory after execution. In the Figure 3 are presented the sections of the original calc.exe and in Figure 4 are presented the sections of the packed calc.exe.

![Fig. 3. Unpacked calc.exe PE sections](image)
Comparing the code structure of the packed `calc.exe` with the unpacked `calc.exe` shows the significant differences between the two executables. When the unpacked `calc.exe` is loaded in the Ghidra Disassembler [15] the `Import Table` (10 libraries are imported) and the `Functions` of the application are displayed and can be analyzed, as shown in Figure 5.

Using the decompiling features of the Ghidra the pseudocode for each of the unpacked `calc.exe` functions can be analyzed alongside the assembly instructions as presented in the Figure 6 where the decompiled pseudocode of the `calc.exe entry point` can be inspected.
By comparison the packed calc.exe shows only 2 functions alongside the entry point and the Import Table has only 2 libraries. The decompiled pseudocode of the entry point calls the FUN_140589009, a function used for the unpacking of the code, as shown in Figure 7.

![Packed calc.exe import table, functions and entry point pseudo-code](image1)

The inspection of the FUN_140589009 pseudo code and assembly instructions do not reveal enough information about the purpose of the application, as shown in Figure 8.

![Packed calc.exe unpacking function pseudo-code](image2)

**5 Methodology**

To assess the effectiveness of current antivirus detection technologies against obfuscated ransomware the following experiment was designed involving 11 live ransomware specimens that were analyzed using the VirusTotal [16] platform. The detection rate was recorded for each ransomware sample and is presented in Table 1.

### 5.1 Ransomware sample selection

The 11 live ransomware samples were obtained from the Malware Zoo GitHub repository [17]. Each sample was executed in an isolated environment to validate that it can encrypt the files and folders on the system. The test was performed to gain assurance that each sample was performing as expected and in a malicious way.
Table 1. The hash signatures for the 11 live ransomware samples

| No. | Ransomware sample | SHA-256 Signature (searchable on VirusTotal) | VirusTotal Detection Rate (72 engines) |
|-----|------------------|---------------------------------|----------------------------------|
| 1   | Cerber           | e67834d1e8b38ec5864caf01b140a94ebf8e f1900a6e2696e94c90fcbfe56678 | 84.72%                           |
| 2   | Cryptowall       | 45317968759d3e37282ceb75149f627d64 | 84.72%                           |
| 3   | Locky            | bc98c8b22461a2c2631b2f3ee3992087dc4 | 91.67%                           |
| 4   | Mamba            | 2ecc52517ed52c74ddaaac47ad513450e58c0f16b1f79be5b576164bf63 | 80.56%                           |
| 5   | Matsnu           | 7634433f8fcf4d13fb46d680802e48e2eb16 | 77.78%                           |
| 6   | Petrwrap         | 027cc450ef5f8c5f65329641ec1fed91f69 | 88.89%                           |
| 7   | Petya            | 26b4699a7b79eb16e76305d843d4ab5e9 | 83.33%                           |
| 8   | Satana           | 683a09da219918258c5a71f617cd416a13a7a377cf8b2a3b440baabf9a4a496 | 87.50%                           |
| 9   | TeslaCrypt       | afaba4200552c7032a5c4c6615df374d0 | 79.17%                           |
| 10  | Vipasana         | c0cf40b883d0d66a24b4f4ebcd162e95aa3 | 75.00%                           |
| 11  | WannaCry         | e01cbfbc9eb5bbae545af4d01b5f10716 | 87.50%                           |

5.2 Ransomware samples obfuscation process
The same 11 live ransomware specimens went through an obfuscation process to increase the difficulty of detection and analysis. The mutated specimens were analyzed using the VirusTotal platform and the results and detection ratio were recorded. The VirusTotal platform was chosen for this research because it uses up to 72 antivirus engines for each submitted sample. All of the 11 ransomware samples are targeting Microsoft Windows based operating systems and they use the PE (portable executable) format.

For the obfuscation process the Themida packer was used to modify the ransomware samples. Themida 2.4.6.0, is currently considered the most difficult packer to reverse engineer and it uses anti-debugging and anti-virtualizations techniques to make protected software harder to reverse engineer. It offers features to run the packed executable inside a virtual machine to make the analysis of the packed executable even harder for reverse engineers. The main difference between Themida and other commercial packers is that Themida offers the ability to run different functions of the packed executable in multiple virtual machines making the analysis even more difficult.

The obfuscated ransomware samples were analyzed using the VirusTotal platform and using the Cuckoo Sandbox [18]. The Cuckoo Sandbox is a security mechanism for separating running programs. It is often used to execute untested code, or untrusted programs from unverified third-parties, suppliers, untrusted users and untrusted websites. A sandbox is used to run an unknown and untrusted application or file inside an isolated environment and observe its behavior. Malware sandboxing is a practical application of the dynamical analysis approach: instead of statically analyzing the binary file, the file is executed and monitored in real-time [19]. The Cuckoo sandbox was...
deployed using the concept of nested virtualization as presented in Figure 9.

![Fig. 9. Cuckoo sandbox architecture used for dynamic analysis](image)

The ransomware sample is loaded in the packer’s interface and the following protection mechanisms are configured, as presented in Table 2 and Figure 10.

| No. | Themida Protection Feature          | Feature Configuration                  |
|-----|-------------------------------------|----------------------------------------|
| 1   | Anti-debugging                      | Advanced                               |
| 2   | Advanced API-Wrapping               | Level 2                                |
| 3   | Compression                         | Application, Resources, SecureEngine    |
| 4   | Anti-Dumpers                        | Yes                                    |
| 5   | Anti-Patching                       | File Patching                          |
| 6   | Entry Point Obfuscation             | Enabled                                |
| 7   | Taggant Information                 | Add Taggant                             |
| 8   | Monitor Blockers                    | File, Registry, Sandbox                 |
| 9   | Resource Encryption                 | Enabled                                |
| 10  | Memory Guard                        | Enabled                                |
| 11  | Delphi/BCB Form Protection          | Enabled                                |
| 12  | VMWare/Virtual PC Execution         | Enabled                                |
| 13  | When Debugger is detected           | Exits silently                         |
The ransomware sample is configured to use two virtual machines for execution, as presented in Figure 11.

The packed ransomware sample will be encrypted, will we loaded as a .dll library (DLL plugin) and the packer will use techniques to hide from PE (portable executable) scanners as presented in Figure 12.
Like in the case of the packed calc.exe the code analysis of the packed ransomware samples is difficult. For example, the unpacked wannacry.exe ransomware sample, when disassembled, shows four libraries in the **Import Table** and more than 20 functions that can be analyzed, as presented in Figure 13.

The packed wannacry.exe ransomware sample, when disassembled, shows 2 libraries in the **Import Table** and 6 functions that can be analyzed, as presented in Figure 14.
6 Results
The packed ransomware samples were analyzed using the following methods:
1) Antivirus analysis using VirusTotal engines
2) Dynamic analysis using Cuckoo Sandbox

6.1 Antivirus analysis results using VirusTotal engines
The detection of each ransomware sample is presented in Table 4. The average detection rate was 32.58%. The average detection rate was increased to 44.95% after 24 hours from the samples submission. The spike in detection rate is attributed to the fact that VirusTotal shares submitted samples with all antivirus companies that didn’t detect the sample as malicious. The samples can be independently verified by searching the SHA-256 signature on the VirusTotal website.

| Ransomware sample packed with Themida | Signature (SHA-256) | VirusTotal Detection Rate (72 engines) after first submission | VirusTotal Detection Rate after 24 hours |
|--------------------------------------|---------------------|-------------------------------------------------------------|----------------------------------------|
| cerber                               | dc9a03120c937119e644fe5dc3 617be5e31d6dd6e146dc4080ad eafb36e631e3c | 34.72%                                       | 55.56%                                  |
| cryptowall                           | 7f4d77a3e8707ab000246522e 28cb0156c7364ac963ff2d8a81 b914c797384e4 | 34.72%                                       | 34.72%                                  |
| locky                                | 6745220a083e7f1a0d69b7ca6 d50c7f9c295e66e66ddec de632437b5844 | 31.94%                                       | 55.56%                                  |
| mamba                                | 252f58b9af5ab2f5e1ce3f2d7c2 780edd03f57d3e11e7f2a0b9 2a374e3f397 | 20.83%                                       | 20.83%                                  |
| matsnu                               | 8687a45fa950a378b0d7a3ada0 6c574705cc12f31748355d5d8d2 99aa1e485c2a6 | 26.39%                                       | 47.22%                                  |
| petrwrap                             | fa678168ef979af5b11829a199 ece6987d3ef07b885cfa292 7978f292a56 | 25.00%                                       | 51.39%                                  |
| petya                                | 4aef0f8ace19b7bb9a63baaf72 7d4d73934220e4523de56367f 8f8caaa5a30e9c | 41.67%                                       | 41.67%                                  |
| satana                               | c121c15e487396189f58b906 5ebd16a3625f524bd3a2ad0fe c5b2566af663e | 34.72%                                       | 34.72%                                  |
| teslacrypt                           | 21e7daea747d6930dca953754 cabdfe841d9b0b43f36b93b5c5 5b405ea711a7c | 29.17%                                       | 29.17%                                  |
| vipasana                             | e1c9bb603b7e6269ada664eb12 9fe668887f2dec52a547d1cd31b de7174b40e0d3 | 34.72%                                       | 55.56%                                  |
| wannacry                             | 36ec96655138b148fc84136ef39 b86037533166f7f49b9f8d395 6664f6f7b47 | 44.44%                                       | 68.06%                                  |

Although the samples were flagged as malicious only an average of 7.20% of the antivirus engines flagged the samples as ransomware. This is an important aspect because as previously stated in the case of ransomware accurate classification is very
important to prevent accidental infection. If a ransomware is obfuscated and distributed in what appears to be an important document or software application for a specific user, if the antivirus alert is ambiguous there are increased chances that the user will create an exception and execute the ransomware. The ransomware classification rate for the 11 samples is presented in Table 4.

**Table 4.** Packed ransomware samples classification rates on VirusTotal

| Ransomware sample packed with Themida | Signature (SHA-256) | VirusTotal Classification Rate as Ransomware (72 engines) after 24 hours |
|--------------------------------------|---------------------|------------------------------------------------------------------------|
| cerber                               | dc9a03120c937119e644fe5dc3617be5e316dd6e146dc4080adeaf36e631e3c | 11.11%                                                                |
| cryptowall                           | 7fd4d77ae38707ab002446522e28cb0156c73c64ac963ff2da81b914c797384e4 | 4.17%                                                                |
| locky                                | 6745220a083e7f1a0d69b7ca6d50e7cf65d25c055e66866dce632437b5844 | 2.78%                                                                |
| mamba                                | 252f58bfad5ab25f1ec3f2d7f2780edd03f57d3e11dea7f2a0b92a374e3f97 | 0.00%                                                                |
| matsu                                | 8687a45fa950a378b0d7a3ada06c574705cc12f31748355d8d29faa1e485c2a6 | 2.78%                                                                |
| petrwrap                             | fa678168ef979af511829a199eece569873ef07b885cf2927987921925a6 | 16.67%                                                               |
| petya                                | 4aef08aee19b799b9a63baba724d739394220e4523ed56367f88c5a530e9c | 0.00%                                                                |
| satana                               | c121c15e48739618958b9065ebd16a3625f524bd34a2ad0f5c52b6af663e | 2.78%                                                                |
| teslacrypt                           | 21e7d3a7d47d6930dca953754ca8dfe841d9b043f36b935c55b405e7a1f74c | 2.78%                                                                |
| vipasana                             | e1c9bb603b7e6269da664cb129fe6888fd2dfec52a547d3d3bde7174b40ed3 | 15.28%                                                               |
| Wannacry                             | 36e29655138b148fe84136ef39b86037533166df7f4b9fc8d39566645f6fb747 | 20.83%                                                               |

### 6.2 Dynamic analysis results using Cuckoo Sandbox

The 11 packed ransomware samples were analyzed in an isolated environment with the Cuckoo Sandbox. Each packed sample was executed in a Windows 7 32bit virtual machine. The sandbox doesn’t use any malware signatures or other heuristic detection methods. The analysis methodology is based on the antivirus industry best practices and methodology for suspect code analysis.

The hypothesis is that any file submitted for analysis is unknown and suspicious. The behavior of the suspect sample is analyzed from a threat perspective and all actions that can have a malicious intent are flagged and reported to the analyst. All 11 submitted samples were flagged as malicious by the Cuckoo Sandbox, as presented in Table 5, and upon execution 4 of the samples were identified as ransomware.

**Table 5.** Packed ransomware samples detection rates using Cuckoo Sandbox

| No | Signature | Sample Name | Malicious Score |
|----|-----------|-------------|-----------------|
| 1  | de9a03120c937119e644fe5dc3617be5e316dd6e146dc4080adeaf36e631e3c | cerber.exe | 75.2% |
| 2  | 7fd4d77ae38707ab002446522e28cb0156c73c64ac963ff2da81b914c797384e4 | cryptowall.exe | 57.6% |
Given the fact that the Themida packer uses heavy anti-debugging and anti-analysis techniques not all of the 11 samples completed the encryption process while being analyzed in the sandbox. The 4 packed samples that started the encryption process and generated the ransom note were: Cerber, Satana, TeslaCrypt and WannaCry. The remaining 7 samples were flagged as malicious based on activities ranging from process and code injection, the installation of boot-kits, connection to suspect internet servers without performing DNS checks etc.

The Cerber ransomware note retrieved during analysis is presented in Figure 15.

The Satana ransomware note retrieved during analysis is presented in Figure 16.
The TeslaCrypt ransomware note retrieved during analysis is presented in Figure 17.

The WannaCry ransomware note is presented in Figure 18.

The malware analysis reports and relevant data extracted from the 11 ransomware samples are published on GitHub [20].

7 Conclusions
In a research paper published at DIMVA 2015 conference researchers stated that, by analyzing over 1395 ransomware samples between 2006 and 2014, the number of families with sophisticated destructive capabilities remains quite small. The analysis revealed that in a large number of samples, the malware simply locks the victim’s computer desktop or attempts to encrypt or delete the victim’s files using only superficial techniques. [21] The ransomware threat landscape has changed significantly in the last 5 year and ransomware attacks are currently
representing a serious threat to organizations around the world. From a financial perspective ransomware can cripple business operations, e-business systems and were responsible for the biggest financial losses produced to organizations in a timespan measured in hours. From this perspective the experiments presented in this research follow the current cybersecurity narrative, that malicious actors are increasing their effort to protect the ransomware code against reverse engineering because in depth analysis can uncover the complex command-and-control network used to manage the ransomware infections. The narrative is supported by several reports and articles published by companies such as NTT Data [22] and IBM [23].

As such the results presented show that by using various obfuscation techniques (like packing and encryption) on known ransomware samples can hinder detection and classification by antivirus engines. By packing the ransomware executable with the Themida packer the detection rates dropped significantly as presented in the Table 6.

The detection rates improved after 24 hours but that should not be considered a significant achievement because in the case of large ransomware outbreaks, like WannaCry, most of the damage was produced in less than 24 hours and at a global scale. Another conclusion is that each of the samples used in the experiment is more than 24 months old, and still by performing obfuscation on the executable code (not on the source code) it can evade the heuristic detection mechanisms found in modern antivirus engines.

Dynamic analysis of the packed ransomware samples, even by using an automated sandbox, proved to me more reliable in detecting the malicious behavior of the samples. The ability to analyze in real time the behavior of the suspect samples can provide all the necessary evidence if the analyzed sample is acting in a malicious way. From 11 packed ransomware samples analyzed in the Cuckoo Sandbox in 4 cases the analysis retrieved the ransom note and the encrypted files from the virtual machine. However, using dynamic analysis and sandboxes to analyze suspect code is not a mainstream activity and it requires both technical resources to deploy the sandbox and skilled personnel with expertise in malware analysis to actually interpret the results.

In March 2019, Norsk Hydro, an aluminum producer was the victim of a ransomware attack which caused more than 40 million USD in losses [24]. The ransomware responsible for the attack is called LockerGoga, as reported by Avira [25]. Although not initially included in the 11 ransomware samples tested in this research, the author obtained a live sample of LockerGoga, from VirusBay [26] and

Table 6. VirusTotal detection rates comparison between the unpacked and packed ransomware samples

| No. | Ransomware sample | VirusTotal Detection Rate (72 engines) - unpacked sample | VirusTotal Detection Rate (72 engines) - packed sample |
|-----|------------------|--------------------------------------------------------|--------------------------------------------------------|
| 1   | Cerber           | 84.72 %                                                | 34.72%                                                |
| 2   | Cryptowall       | 84.72 %                                                | 34.72%                                                |
| 3   | Locky            | 91.67 %                                                | 31.94%                                                |
| 4   | Mamba            | 80.56 %                                                | 20.83%                                                |
| 5   | Matsnu           | 77.78 %                                                | 26.39%                                                |
| 6   | Petwrap          | 88.89 %                                                | 25.00%                                                |
| 7   | Petya            | 83.33 %                                                | 41.67%                                                |
| 8   | Satana           | 87.50 %                                                | 34.72%                                                |
| 9   | TeslaCrypt       | 79.17 %                                                | 29.17%                                                |
| 10  | Vipasana         | 75.00 %                                                | 34.72%                                                |
| 11  | WannaCry         | 87.50 %                                                | 44.44%                                                |
submitted the sample to VirusTotal. The sample identified with the SHA-256 signature presented in Table 7 was detected by 49/72 engines.

### Table 7. Unpacked LockerGoga SHA-256 signature

| SHA-256 signature | Detection engines |
|-------------------|-------------------|
| 2fe3c29913f66c255cb7aa5c34821ab182f889e7f96c25bad31267adc8a19e5b | 49 |

The author packed the LockerGoga sample with the Themida packer and re-submitted the sample to VirusTotal. The sample with the SHA-256 signature. Presented in Table 8 was detected by 20/72 engines and classified as ransomware by two engines.

### Table 8. Packed LockerGoga SHA-256 signature

| SHA-256 signature | Detection engines |
|-------------------|-------------------|
| 974df521074fe3aba941e43e72f16882b9ea268c801ea3eaa01fafa39bad70525 | 20 |

Dynamic analysis of the packed LockerGoga sample revealed that the ransomware executed the encryption process successfully and also generated the ransom note, as presented in the Figure 19.

![Image](https://example.com/image.png)

**Fig. 19.** Packed LockerGoga ransomware note retrieved during dynamic analysis

A general conclusion based on the limited number of samples tested is that signature and heuristic based malware detection algorithms have issues to detect new or obfuscated ransomware. Dynamic analysis and suspect code execution inside a sandbox currently remain the most reliable detection and classification method for ransomware. Ransomwares represents a group of malware applications so destructive that the need accurate detection prior to execution or during the initial stages of execution is crucial in order to mitigate the threat.

### 8. References

[1] Adam L.; Yung, Moti. "An Implementation of Cryptoviral Extortion Using Microsoft's Crypto API" Young. Available at https://www.cryptovirology.com/cryptovf

[2] The Register, "Fake cop Trojan 'detects offensive materials' on PCs, demands money". Available at https://www.theregister.co.uk/2012/04/05/police_themed_ransomware/

[3] Kelion, Leo, "Cryptolocker ransomware has 'infected about 250,000 PCs". Available at https://www.bbc.com/news/technology-25506020

[4] Tripwire, “NotPetya - Timeline of a ransomworm”. Available at https://www.tripwire.com/state-of-security/security-data-protection/cyber-security/notpetya-timeline-of-a-ransomworm/

[5] Wired, “The Untold Story of NotPetya, The Most Devastating Cyberattack In History”. Available at
https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/

[6] Rivest, Shamir, Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. Communications of the ACM, 21 (2), pp. 120-126, February 1978

[7] NIST, “The Advanced Encryption Standard FIPS – 197”. Available at https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf

[8] Subedi, Kul & Budhathoki, Daya Ram & Dasgupta, Dipankar, “Forensic Analysis of Ransomware Families Using Static and Dynamic Analysis”, 10.1109/SPW.2018.00033, 2018

[9] Charles Crofford, Douglas McKee, “Ransomware Families Use NSIS Installers to Avoid Detection, Analysis”. Available at https://securingtomorrow.mcafee.com/other-blogs/mcafee-labs/ransomware-families-use-nsis-installers-to-avoid-detection-analysis/

[10] Avast, „A closer look at the Locky ransomware”. Available at https://blog.avast.com/a-closer-look-at-the-locky-ransomware

[11] Kaspersky Lab, “SynAck targeted ransomware uses the Doppelgänging technique”. Available at https://securelist.com/synack-targeted-ransomware-uses-the-doppelganging-technique/85431/

[12] VMRay, „Gandcrab ransomware evolution analysis”. Available at https://www.vmray.com/cyber-security-blog/gandcrab-ransomware-evolution-analysis/#packer_gandcrav4

[13] Oreans Technologies, “Themida – Advanced Windows Software Protection System”. Available at https://www.oreans.com/themida.php

[14] Winitor, “PE Studio”. Available at https://www.winitor.com/

[15] National Security Agency, “Ghidra Disassembler”. Available at https://ghidra-sre.org/

[16] Google VirusTotal. Available at https://www.virustotal.com

[17] Malware Zoo. Available at github.com/ytisf/theZoo/tree/master/mawares/Binaries

[18] Cuckoo Sandbox. Available at https://cuckoosandbox.org/

[19] Cuckoo Sandbox Manual. Available at https://cuckoo.sh/docs/introduction/sandboxing.html

[20] Sechel Sergiu, “Ransomware dynamic analysis using sandboxes”. Available at https://github.com/tornwire/Ransomware-Dynamic-Analysis-Cuckoo-

[21] Amin Kharraz, William Robertson, Davide Balzarotti, Leyla Bilge, Engin Kirda, “Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks”, Proceedings of the 12th International Conference, DIMVA 2015 Milan, Italy, July 9–10, 2015

[22] NTT Data, “Quarterly Report on Global Security Trends”. Available at https://www.nttdata.com/global/en-/media/nttdataglobal/1_files/media/securetyreport/2018/fy2018_2q_sr_eng.pdf

[23] IBM, “GandCrab partners with NTCrypt for code obfuscation”. Available at https://securityintelligence.com/news/gandcrab-partners-with-ntcrypt-for-code-obfuscation/

[24] ZDNET, “Norsk Hydro ransomware incident losses reach $40 million after one week”. Available at https://www.zdnet.com/article/norsk-hydro-ransomware-incident-losses-reach-40-million-after-one-week/

[25] Avira, “Ransomware causes operation meltdown at Norsk Hydro”. Available at https://blog.avira.com/ransomware-causes-operation-meltdown-at-norsk-hydro/

[26] Virus Bay, “Malware Samples”. Available at https://beta.virusbay.io

DOI: 10.12948/issn14531305/23.2.2019.05
Sergiu SECHEL has graduated the Faculty of Automation and Applied Computer Sciences in 2009. He is a Ph.D. Candidate in Economy Informatics at the Bucharest University of Economic Sciences and an Advisory Manager at EY (Ernst & Young). His areas of research are cybersecurity, audit, risk management and malware research.