

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

RANDETER

USING NOVEL STATISTICAL AND PHYSICAL CONTROLS TO DETER RANSOMWARE ATTACKS

A thesis presented in partial fulfillment of the requirements

for the degree of

Master of Information Sciences

In Software Engineering

at Massey University, Auckland, New Zealand

Timothy Raymond McIntosh

2018

Abstract

Crypto-Ransomware are a type of extortion-based malware that encrypt victims' personal files with strong encryption algorithms and blackmail victims to pay ransom to recover their files. The recurrent episodes of high-profile ransomware attacks like WannaCry and Petya, particularly on healthcare, government agencies and big corporates, have highlighted the immediate demand for effective defense mechanisms.

In this paper, RANDETER is introduced as a novel anti-crypto-ransomware solution that deters ransomware activities, using novel statistical and physical controls inspired by the police anti-terrorism practice. Police try to maintain public safety by maintaining a constant presence to patrol key public areas, identifying suspects who exhibit out-of-ordinary characteristics, and restricting access to protected areas. Ransomware are in many ways like terrorists; their attacks are unexpected, malicious and aim for the largest number of victims. It is possible to try to detect and deter crypto-ransomware by maintaining a constant surveillance on the potential victims – MBR and user files especially documents and photos.

RANDETER is implemented as two compatible and complementary modules: PARTITION GUARD and FILE PATROL. PARTITION GUARD blocks modifications to the area of MBR on the booting disk. FILE PATROL checks all file activities of directories protected by RANDETER against a list of Recognized Processed with Multi-Tier Security Rules. Upon detection of violations of such rules, which may have been initiated by crypto-ransomware as judged by FILE PATROL, FILE PATROL will freeze access of the monitored directories, terminate the offending processes, and resume access of those directories.

Our evaluation demonstrated that RANDETER could ensure less and often no irrecoverable file damage by current ransomware families, while imposing less disk performance overheads, compared to existing competitor anti-ransomware implementations like CRYPTOLOCK, SHIELDIFS and REDEMPTION. In addition, RANDETER was shown to be resilient against masquerading attacks and ransomware polymorphism.

Acknowledgements

I would like to express my deep gratitude to my supervisors, Associate Professor Julian Jang-Jaccard of Massey University (New Zealand) and Professor Paul Watters of La Trobe University (Australia). Both are cybersecurity experts and have shared their extensive industry and research experience, which has been very invaluable to the success of the RANDETER project and this thesis.

I would like to thank Dr Andrea Continella (PhD) and his colleagues at the NECST Lab of Politecnico di Milano (Italy), who developed the self-healing, ransomware-aware filesystem SHIELDFS¹. They have kindly shared their IRP logs and machine statistics obtained while developing SHIELDFS, which have provided great research value during the development of RANDETER.

The RANDETER project is based on Microsoft® Minispy File System Minifilter Driver and Storage Class Driver. The Microsoft® version of Minispy sample is a command-line only tool to monitor and log any I/O and transaction activity that occurs in the system. Minispy is implemented as a minifilter and is available open source² on GitHub. The Microsoft® version of disk class driver sample is used for managing disk devices and is also available open source³ on GitHub.

¹ <http://shieldfs.necst.it/>

² <https://github.com/Microsoft/Windows-driver-samples/tree/master/filesys/miniFilter/minispy>

³ <https://github.com/Microsoft/Windows-driver-samples/tree/master/storage/class/disk>

Table of Contents

CHAPTER 1. INTRODUCTION	1
1.1 BRIEF INTRODUCTION TO RANSOMWARE THREATS.....	1
1.2 RESEARCH SCOPE AND OBJECTIVES.....	2
1.3 STRUCTURE OF THE THESIS	2
CHAPTER 2. LITERATURE REVIEW	5
2.1 WHAT IS RANSOMWARE?	5
2.2 CLASSIFICATIONS OF RANSOMWARE	6
2.3 A TYPICAL RANSOMWARE ATTACK.....	7
2.4 HISTORY OF RANSOMWARE AND THE SCALE OF THE PROBLEM	12
2.5 CURRENT DETECTION STRATEGIES AND THEIR LIMITATIONS.....	15
2.6 GENERAL STRATEGIES FOR MITIGATING RANSOMWARE RISKS	17
2.7 RESEARCH APPROACHES ON RANSOMWARE DETECTION AND PREVENTION WITHOUT MONITORING FILE SYSTEM ACTIVITIES.....	18
CHAPTER 3. A CLOSER LOOK INTO CRYPTO-RANSOMWARE ATTACKS.....	21
3.1 CRYPTO-RANSOMWARE MUST ATTACK THE FILE SYSTEMS.....	21
3.2 ENTRY POINTS OF TARGETED ATTACKS ON FILE SYSTEMS	22
3.3 PREFERENCE TO ACCESS NON-SYSTEM FILES AND FOLDERS	23
3.4 AGGRESSIVENESS TOWARDS FILE SYSTEMS	26
3.5 GREEDINESS FOR MODIFYING MORE DIVERSE TYPES OF FILES.....	27
3.6 OTHER STATISTICAL FINDINGS.....	29
CHAPTER 4. IMPLEMENTATIONS MONITORING FILE SYSTEM ACTIVITIES.....	31
4.1 CURRENT RESEARCH ARTICLES	31
4.2 CURRENT PATENTS.....	35
4.3 WINDOWS DEFENDER “CONTROLLED FOLDER ACCESS” AND MASQUERADE ATTACKS.....	37
CHAPTER 5. RANDETER THEORETICAL CONCEPTS	41
5.1 SIMILARITIES BETWEEN RANSOMWARE ATTACKS AND TERRORISM ATTACKS	41
5.2 THE FIVE GUIDING PRINCIPLES OF ANTI-TERRORISM AND THEIR RELEVANCE TO ANTI-RANSOMWARE PRACTICE	42
5.3 THE PATROL-AND-DEFEND APPROACH EXPLAINED.....	43
5.4 RELEVANCE TO AUSTRALIAN SIGNALS DIRECTORATE CYBERSECURITY STRATEGIES	46
5.5 FILE SYSTEM EVENTS.....	47
5.6 RECOGNIZED PROCESSES AND MULTI-TIER SECURITY RULES	51
CHAPTER 6. RANDETER TECHNICAL DETAILS.....	57
6.1 THE ARCHITECTURE OF RANDETER	57

6.2	PARTITION GUARD	59
6.3	FILE PATROL	61
6.3.1	THE FILE SYSTEM MINIFILTER	62
6.3.2	THE SHELL PROGRAM.....	63
 <u>CHAPTER 7. EVALUATION</u>		<u>65</u>
7.1	EXPERIMENT SETUP	65
7.2	SELECTION AND COLLECTION OF RANSOMWARE SAMPLES.....	67
7.3	DETECTION RESULTS	71
7.4	BENCHMARKS OF FILE SYSTEM I/O PERFORMANCE	73
7.5	EFFECTIVENESS AGAINST RANSOMWARE POLYMORPHISM.....	75
 <u>CHAPTER 8. A CASE STUDY OF WANNACRY</u>		<u>79</u>
8.1	WANNACRY INTRODUCTION.....	79
8.2	WANNACRY ATTACK IN DETAILS.....	80
8.3	FILE PATROL IN ACTION AGAINST WANNACRY	87
8.4	FILE PATROL CAN NEUTRALIZE WANNACRY MASQUERADING AS MICROSOFT WORD	88
 <u>CHAPTER 9. A CASE STUDY OF PETYA</u>		<u>91</u>
9.1	PETYA INTRODUCTION.....	91
9.2	PETYA ATTACK IN DETAILS	91
9.3	PARTITION GUARD IN ACTION AGAINST PETYA.....	94
 <u>CHAPTER 10. DISCUSSIONS AND LIMITATIONS OF RANDETER.....</u>		<u>97</u>
10.1	RISKS OF ATTACKS ON THE SYSTEM OF RECOGNIZED PROCESSES	97
10.2	POSSIBILITY OF TOTAL PREVENTION OF IRRECOVERABLE FILE DAMAGE	98
10.3	IMPROVED ADMISSION INTO RECOGNIZED PROCESSES	99
10.4	REDUCING POSSIBILITY OF DENIAL OF SERVICE ATTACKS	99
 <u>CHAPTER 11. CONCLUSION</u>		<u>101</u>
 <u>REFERENCES.....</u>		<u>103</u>
 <u>APPENDIX 1. LIST OF RANSOMWARE SAMPLES TESTED WITH RANDETER</u>		<u>108</u>

List of Figures

FIGURE 1 - RANSOM.PETYA MASQUERADING AS AN ADOBE PDF FILE.....	7
FIGURE 2 - RANSOM.BADRABBIT MASQUERADING AS ADOBE FLASH PLAYER UPDATE.....	8
FIGURE 3 - COMPARISON OF A PURE TEXT FILE BEFORE (LEFT) AND AFTER (RIGHT) RANSOM.TESLACRYPT ATTACK.....	10
FIGURE 4 - RANSOM.TESLACRYPT ATTEMPTING TO DELETE VOLUME SHADOW BACKUP COPIES IN SILENT MODE	11
FIGURE 5 - RANSOM MESSAGE DISPLAYED BY RANSOM.TESLACRYPT AFTER AN ATTACK	12
FIGURE 6 - RANSOM MESSAGE OF RANSOM.WANNACRY IN JAPANESE, WITH MANY LOCALIZATIONS AVAILABLE	15
FIGURE 7 - PERCENTAGE OF IRP ACCESS ON BENIGN MACHINES (DATA SOURCE: SHIELD FS STUDY BY CONTINELLA ET AL, 2016).....	25
FIGURE 8 - PERCENTAGE OF IRP ACCESS ON RANSOMWARE-INFECTED MACHINES (DATA SOURCE: SHIELD FS STUDY BY CONTINELLA ET AL, 2016).....	25
FIGURE 9 - NUMBER OF IRP REQUESTS TO MODIFY FILE CONTENTS OR FILE INFORMATION OF PDF FILES	27
FIGURE 10 - NUMBER OF DIFFERENT FILE TYPES MODIFIED BY THE PROGRAM WHICH MODIFIED MOST FILE TYPES	28
FIGURE 11 - THE TRADITIONAL INFECT-AND-IMMUNE APPROACH AGAINST MALWARE	44
FIGURE 12 - THE NEWLY PROPOSED PATROL-AND-DEFEND APPROACH AGAINST RANSOMWARE	45
FIGURE 13 - THE CHANGE EVENT GENERATED BY USER SAVING A FILE IN NOTEPAD++.....	47
FIGURE 14 - FILE EVENTS GENERATED BY USER MODIFYING A DOCX FILE USING MICROSOFT WORD.....	48
FIGURE 15 - GENERAL ARCHITECTURE OF RANDETER	58
FIGURE 16 - PARTITION GUARD TECHNICAL DETAILS	60
FIGURE 17 - FILE PATROL TECHNICAL DETAILS.....	64
FIGURE 18 - COMMAND LINE COMMAND TO SWITCH ON TEST MODE IN WINDOWS 10	66
FIGURE 19 - THE "TEST MODE" WATERMARK ON WINDOWS 10 DESKTOP.....	66
FIGURE 20 - WEBSITE OF SYMANTEC™ THREAT EXPLORER	68
FIGURE 21 - A SCREENSHOT OF VIRUSSHARE.COM WEBSITE.....	69
FIGURE 22 - A SCREENSHOT OF VIRUSTOTAL.COM WEBSITE.....	70
FIGURE 23 - FILE SYSTEM I/O PERFORMANCE BENCHMARK WITH OR WITHOUT RANDETER	74
FIGURE 24 - ON VIRUSTOTAL.COM, 59 OUT OF 67 ANTI-VIRUS ENGINES RECOGNIZED THE SAMPLE.....	76
FIGURE 25 - REPACKING THE RANSOM.WANNACRY SAMPLE USING UPX AND THE --BRUTE PARAMETER.....	77
FIGURE 26 - ON VIRUSTOTAL.COM, ONLY 40 OUT OF 66 ANTI-VIRUS ENGINES RECOGNIZED THE REPACKED SAMPLE BY UPX AS RANSOM.WANNACRY OR SOMEWHAT MALICIOUS	78
FIGURE 27 - THE ZIP ARCHIVE CONTENTS OF WANNACRY RANSOMWARE	81
FIGURE 28 - FILE SYSTEM ACTIVITIES OF WANNACRY RANSOMWARE DURING AN ATTACK, RECORDED BY FILE PATROL.....	82
FIGURE 29 - WALLPAPER USED BY WANNACRY TO DISPLAY RANSOM MESSAGE ON USER'S DESKTOP BACKGROUND	83
FIGURE 30 - THE RANSOM MESSAGE OF WANNACRY DISPLAYED IN ENGLISH.....	84
FIGURE 31 - COMPARISON OF DIRECTORY FILES BEFORE (LEFT) AND AFTER (RIGHT) WANNACRY ATTACKS	85
FIGURE 32 - COMPARISON OF THE PURE TEXT FILE CONTENT BEFORE (LEFT) AND AFTER (RIGHT) WANNACRY ENCRYPTION	86
FIGURE 33 - WANNACRY WAS NOT ON THE LIST OF RECOGNIZED PROCESSES AND WAS DENIED ACCESS BY FILE PATROL	87
FIGURE 34 - FILE PATROL IN ACTION, DENYING ACCESS OF RANSOMWARE PROCESSES AND TERMINATING THEM	88
FIGURE 35 - FIRST TEST RUN WITH DIFFERENT FILE TYPES: DURING A SIMULATED MASQUERADING ATTACK AS MICROSOFT WORD, RANSOM.WANNACRY FAILS THE "FILE TYPE RULE".....	89
FIGURE 36 - SECOND TEST RUN WITH ONLY DOC AND DOCX FILES: DURING A SIMULATED MASQUERADING ATTACK AS MICROSOFT WORD, RANSOM.WANNACRY FAILS THE "OPERATION RULE".....	90
FIGURE 37 - RANSOMWARE PETYA TRIGGERS THE BLUE SCREEN WITH A 0xc0000350 ERROR TO FORCE A SYSTEM RESTART	92
FIGURE 38 - RANSOMWARE PETYA MASQUERADING AS CHKDSK WHILE ENCRYPTING THE MASTER FILE TABLE.....	93
FIGURE 39 - RANSOM MESSAGE DISPLAYED BY RANSOMWARE PETYA.....	94
FIGURE 40 - FLOWCHART OF RANSOMWARE ATTACK ON MBR (LEFT), AND HOW IT IS DETERRED BY PARTITION GUARD (RIGHT)	95
FIGURE 41 - PARTITION GUARD IN ACTION, DENYING WRITE ACCESS TO MBR	95

List of Tables

TABLE 1 - COMPARISON OF EXISTING IMPLEMENTATIONS MONITORING FILE SYSTEM ACTIVITIES OF RANSOMWARE	32
TABLE 2 - DEFINITION OF FILE SYSTEM EVENTS AND UNDERLYING IRP OPERATIONS	50
TABLE 3 - CLASSIFICATION OF TYPES OF RECOGNIZED PROCESSES	51
TABLE 4 - AN EXAMPLE SECURITY RULE OF "EXPLORER.EXE", A WINDOWS SYSTEM MODULE	52
TABLE 5 - AN EXAMPLE SECURITY RULE OF "SVCHOST.EXE", A WINDOWS SYSTEM MODULE.....	53
TABLE 6 - AN EXAMPLE SECURITY RULE OF 7-ZIP, A UTILITY	53
TABLE 7 - AN EXAMPLE SECURITY RULE OF MICROSOFT WORD, AN EDITOR.....	54
TABLE 8 - DETECTION RESULTS AND COMPARISON WITH OTHER ANTI-CRYPTO-RANSOMWARE SOLUTIONS.....	72
TABLE 9 - FILE SYSTEM I/O PERFORMANCE BENCHMARK WITH OR WITHOUT RANDETER	74
TABLE 10 - ORGANIZATION AFFECTED BY WANNACRY	79
TABLE 11 – SUMMARY OF RISKS OF ATTACKS ON THE SYSTEM OF RECOGNIZED PROCESSES AND MITIGATION STRATEGIES ..	97

Chapter 1. INTRODUCTION

In this chapter, a brief introduction is given about the project and the structure of the thesis.

1.1 Brief Introduction to Ransomware Threats

In recent years, ransomware have developed to become one of the most significant cybersecurity threats on the Internet. They execute on victims' computers, make important user documents and data inaccessible, and demand ransom payments from victims to release the restrictions.

The ever-growing cases of high-profile ransomware attacks on hospitals, universities, government agencies and corporates have led to numerous disruptions in services offered by the affected entities, in addition to financial losses. In response to the rising ransomware threats, users are often advised to regularly backup their data, use security software, and be vigilant while opening files from unknown sources. However, ransomware developers can target unsophisticated users, who often do not fully follow such recommendations, and continue to create new, evolved and more sophisticated attacks to evade detections.

Current defense solutions, which are often based on pure-detection approaches relying on malware signature matching, appear to be no longer sufficient, as modern ransomware implement multiple different techniques including polymorphism to evade detection by common security software and cause frequent outbreaks. There has been substantial amount of illegal profit generated by ransomware campaigns, and the interest of cybercriminals in ransomware schemes continues to grow. A forward-thinking progressive solution to the issue could be to equip operating systems with a generic and practical self-defense system against ransomware intrusions.

1.2 Research Scope and Objectives

In this paper, RANDETER (**R**ansomware **D**eterrent) is introduced as a novel generic real-time anti-crypto-ransomware solution to overcome the limitations of existing detection mechanisms. The following original contributions have been made:

- A generic approach to defend against ransomware attacks was presented. In this approach, modifying files or disk information in protected areas is strictly controlled. Only selected applications on a pre-defined whitelist are permitted, and they are only permitted to perform file system activities in prescribed manners.
- It was demonstrated that performance-efficient anti-crypto-ransomware protection with no data loss is possible, without consuming extra disk storage.
- The possibility of performing “Masquerading Attacks” was hypothesized, when a ransomware sample could replace a legitimate application that has already been added to the trusted list of security software, to carry out ransomware attacks.
- A proof-of-concept prototype implementation of RANDETER was developed and presented for the NTFS file system on Microsoft® Windows platforms and evaluated with recent ransomware samples.
- The detection rate and accuracy of RANDETER was compared with other similar implementations that monitored file system activities of ransomware. RANDETER could deter ransomware attacks, while imposing no perceptible performance sacrifices.
- RANDETER was proved to be able to deter the theoretically possible masquerading attacks described above.

1.3 Structure of the Thesis

Chapter 2 is the literature review of the current knowledge and research outcomes of ransomware and ransomware threats. It seeks to highlight the primary challenges in ransomware detection and monitoring.

Chapter 3 examines the crypto-ransomware attacks in depth.

Chapter 4 reviews current research outcomes, patents and existing implementations on monitoring file system activities, and discusses about their limitations.

Chapter 5 presents the ideas and concepts of RANDETER, a deterrent approach to deter ransomware attacks.

Chapter 6 describes the technical and implementation details of RANDETER.

Chapter 7 evaluates the implementation with crypto-ransomware samples and compares the results with other competitor anti-ransomware implementations.

Chapter 8 and 9 are case studies of RANDETER in action against WannaCry and Petya ransomware respectively.

Chapter 10 discusses about RANDETER and its limitations

Chapter 11 is the conclusion and future work.

Chapter 2. LITERATURE REVIEW

In this chapter, a literature review is conducted to explore the current knowledge of ransomware and the research trend.

2.1 What is Ransomware?

Ransomware is a class of malware designed to extort victims into paying the criminals (Liska & Gallo, 2016). They infect and restrict users' access to computer systems or important files and data, often before users notice the intrusions, and demand "ransom" to be paid to unlock their systems or resume access. They are mainly designed for criminal revenue generation, while victims' computers are effectively taken hostage (Liska & Gallo, 2016). Ransomware campaigns are found to be almost always motivated by economic gains; it is a type of criminal activities (O'Gorman & McDonald, 2012).

The exponentially growing problem of ransomware is caused by a combination of multiple factors (Liska & Gallo, 2016):

- The potential profit is substantial and very lucrative.
- The "business" model is mature.
- The popularity of crypto digital currencies like Bitcoin makes it possible to collect payments anonymously and launder the criminal profit, with little chance of being tracked by law-enforcing authorities.
- The advancement of better hardware enables faster encryption running asynchronously at the background, without noticeable system performance compromise.
- The advancement of more complex encryption schemes enabled more efficient and robust encryption of victims' files.
- The emergence and adoption of Ransomware as a Service (RaaS) allowed skilled hackers to easily offer their services to a wider criminal population to launch ransomware campaigns, targeting a wider range of victims.

2.2 Classifications of Ransomware

Symantec™ classified ransoms into two categories: locker-ransomware and crypto-ransomware (Savage, Coogan & Lau, 2015).

Some studies (Salvi & Kerkar, 2016; O'Gorman & McDonald, 2012) considered *Scareware* to be the earliest form of ransomware, which often take the form of harassing or threatening users to pay the ransom. Most earlier scareware hijack the web browser and dead-lock it with unlimited number of pop-ups or fake law-enforcement screens, until the victim pays the “fine”. Scareware usually rely on coercion, are not sophisticated, and are often easy to remove. Because *Scareware* do not completely lock the entire operating systems or user data from access, they are not considered as ransomware by this thesis.

Locker-ransomware (computer locker) lock the victim’s computer system, denies access until the ransom is paid. Some lock the device user interface only, while others infect important boot record or operating system kernels, so victims can only interact with the ransomware. Locker ransomware usually do not encrypt underlying system and files; Seftad overwrites the MBR with a malicious one and upon reboot, lies to the victims that their files are encrypted unless ransom payments are made. Tech-savvy victims can often fix their systems using various utilities and techniques.

Crypto-ransomware (data locker) scan and encrypt the victims’ important user files and data with strong encryption algorithms, and are often the most damaging and difficult to restore (ESET, 2017). It was found that many computer users do not back up their important files and data; crypto ransomware target this weakness and assume that the files carry too much sentimental or business values to be lost forever (Symantec, 2017).

Most victims regain access to their computers and files after paying the ransom, as criminals who designed the ransomware treasure their “reputation” and would encourage more victims to pay the ransom as an easier way to regain access (Symantec, 2017). However, this is not always guaranteed, and there is no guarantee that ransomware would be removed upon ransom payments (ESET, 2017). Some victims have been reported to suffer permanent loss of files and data, due to the payment method no longer available, or criminals hide to evade police investigation (ESET, 2017).

2.3 A Typical Ransomware Attack

Liska and Gallo (2016) have defined a five-stage anatomy of a typical ransomware attack; crypto-ransomware, the most damaging type of ransomware, must complete the following list of tasks to install itself and encrypt victims' files.

(1) Infection and Installation

A ransomware sample can spread itself via spam emails or malicious websites and can disguise itself as something legitimate like a PDF document (*Ransom.Petya*, in figure 1) or an Adobe Flash update (*Ransom.BadRabbit*, in figure 2). It will try to deceive or coerce the user into executing the file, so it can install itself onto the victim system.

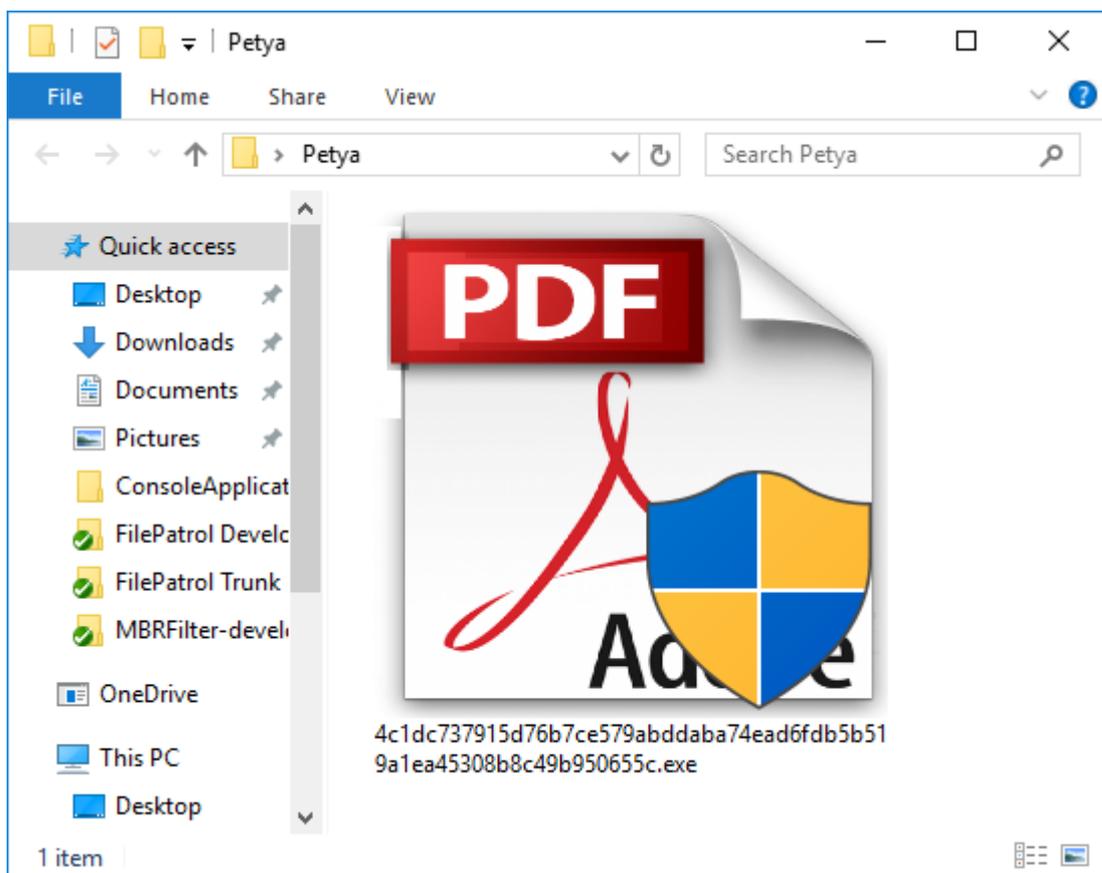


Figure 1 - *Ransom.Petya* Masquerading as an Adobe PDF file

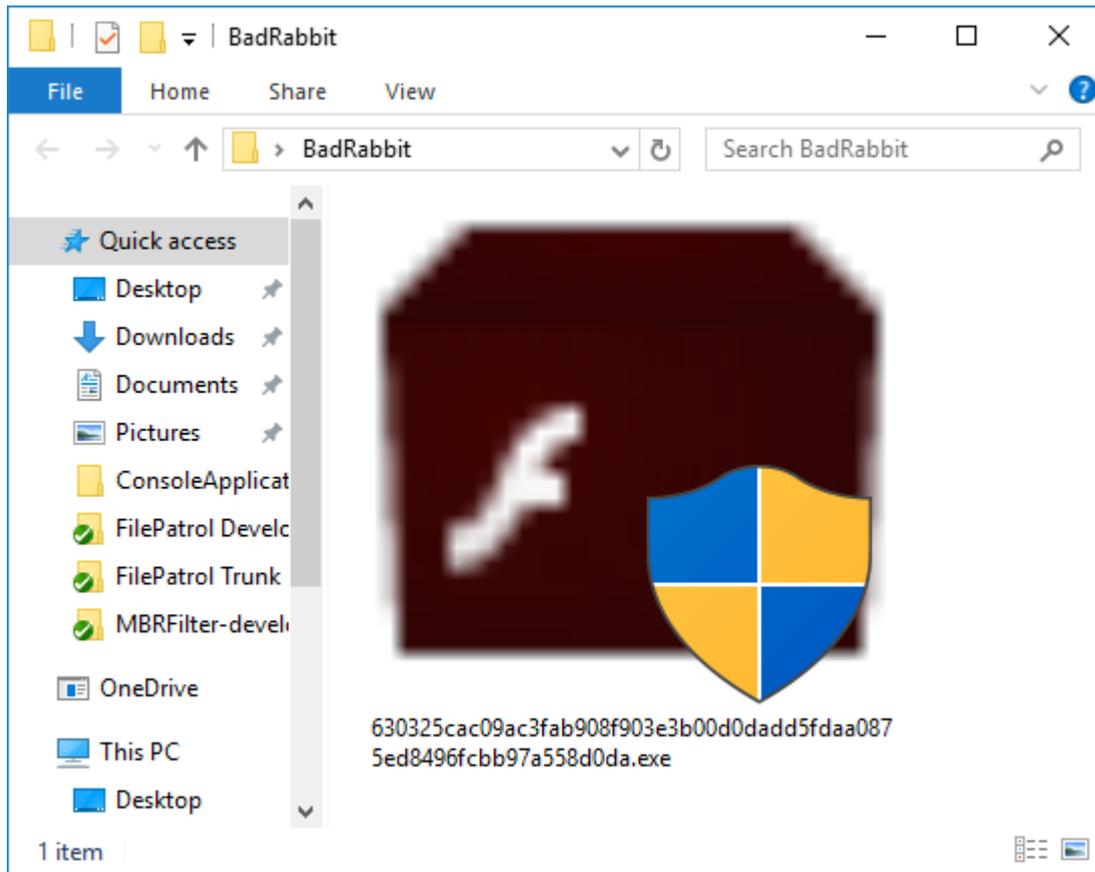


Figure 2 - *Ransom.BadRabbit* Masquerading as Adobe Flash Player update

(2) Command-and-Control

Many ransomware cybercriminals operate remote servers that act as command-and-control centers of ransomware. Once executed, a ransomware would try to establish a connection with its center, to confirm whether itself should be active and should execute, and/or to try to obtain a unique private encryption key that is often specific to the victim machine, to prevent mass decryption when others have paid their ransom. Some ransomware types, like *Ransom.WannaCry*, carry default encryption keys and can still attack without Internet connections. *Ransom.Petya*, however, has no known Command-and-Control mechanism and does not need to communicate with the ransomware developers before attacks.

(3) Selection of File Targets

To take as many important user files as hostage as possible, ransomware need to traverse the file systems using folder-listing operations, to look for user files stored on the victim machine that contain individualized data and are not part of the operating system. Ransomware then selectively create a list of files to encrypt, often based on file types that are more likely to contain valuable user data, like documents and photos.

(4) Encryption

Ransomware would read the content of the selected files, encrypt them using strong encryption algorithm with the encryption key obtained from the command-and-control center. Then the cipher would be written back to the original files, replacing the original files and making them unreadable. The encryption step is often performed rapidly in the background, generating high disk I/O operations.

Below is the comparison of a plain text file before and after an attack by *Ransom.TeslaCrypt* (figure 3). The original plain text includes “The quick brown fox jumps over the lazy dog. 0123456789” and some commonly used symbols. The encrypted text is completely illegible.

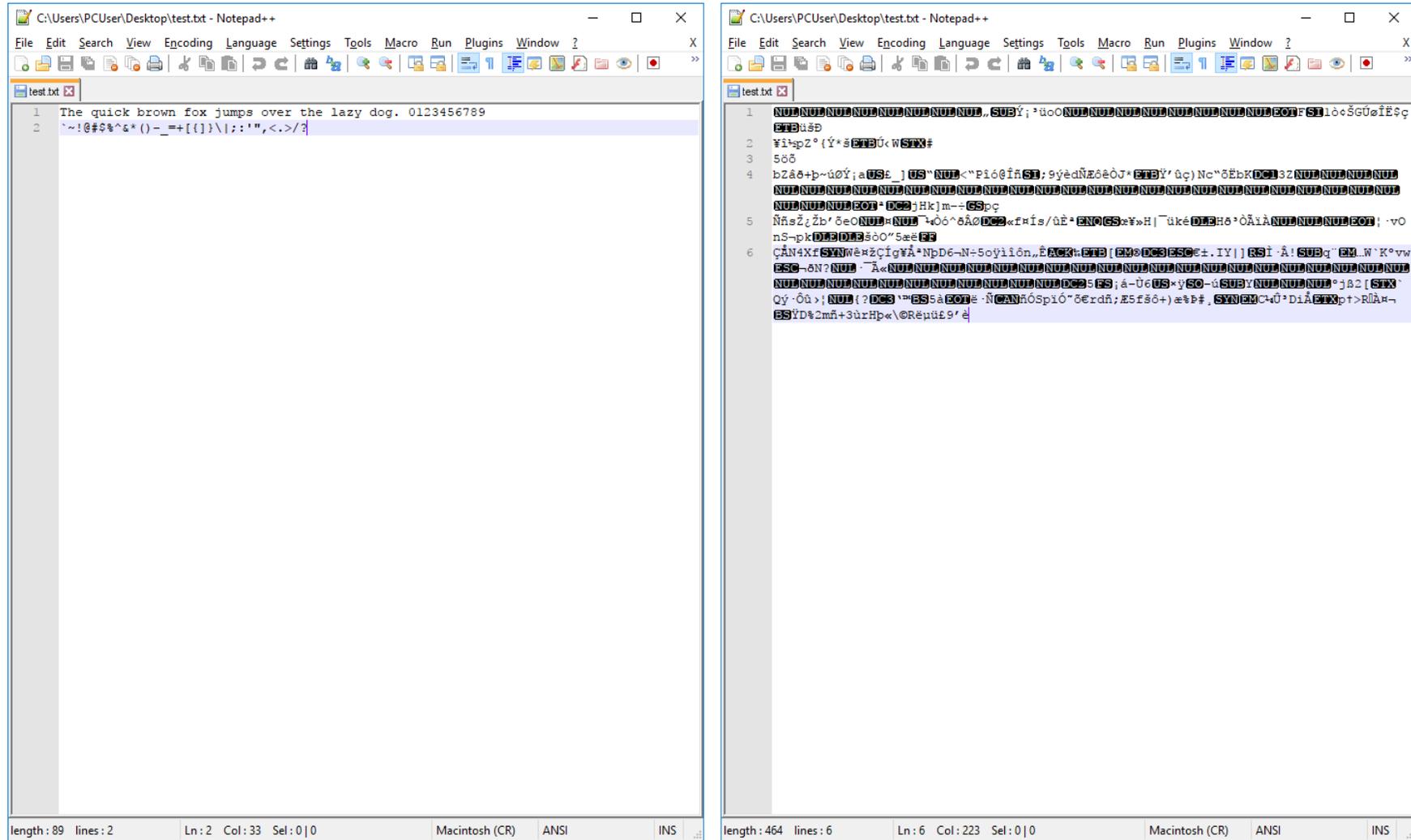


Figure 3 - Comparison of a Pure Text File Before (Left) and After (Right) Ransom.TeslaCrypt Attack

Many ransomware would also destroy backup copies, delete System Restore points, and terminate Volume Shadow Copy service, which provides the backup infrastructure for Windows operating systems (Microsoft, 2009); doing so would make it impossible to restore backup copies of encrypted user files, forcing victims to pay the ransom. Below is a screenshot showing *Ransom.TeslaCrypt* was trying to use command line to delete volume shadow backup copies (figure 4). It was using the “/all” parameter to delete all copies and using “/quiet” parameter to mute all warnings to minimize user suspicion.

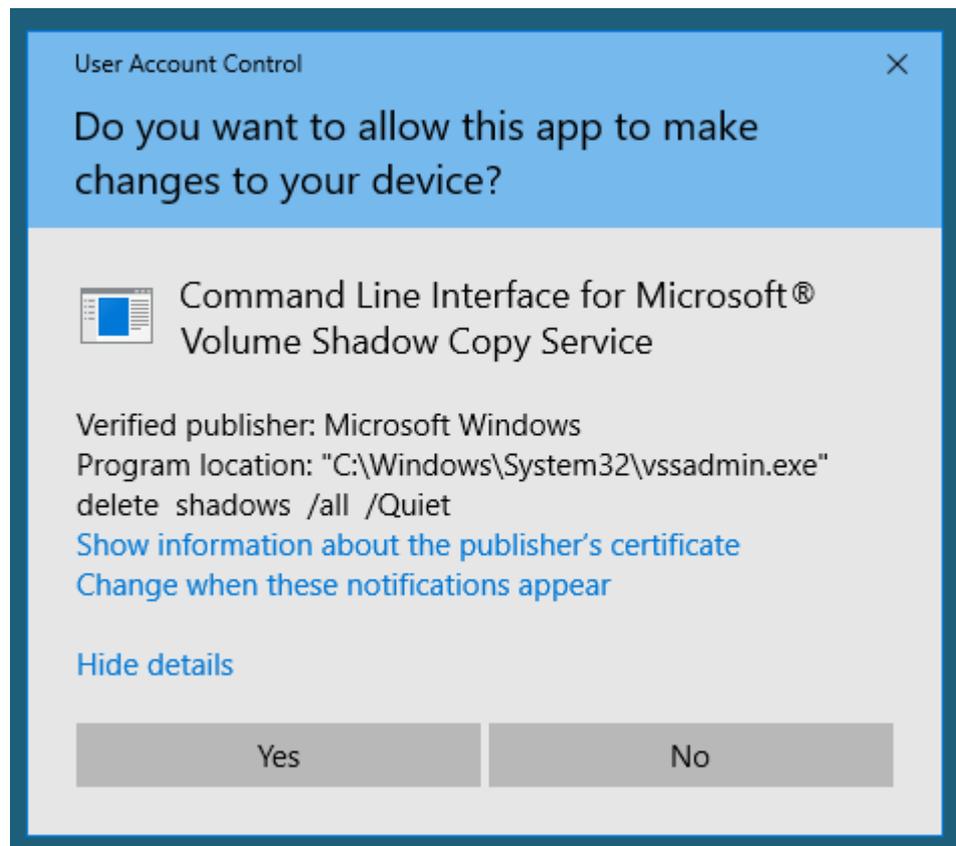


Figure 4 - *Ransom.TeslaCrypt* Attempting to Delete Volume Shadow Backup Copies in Silent Mode

(5) Extortion

Displaying a screen giving payment instructions, imposing a time limit to pay up before the encryption becomes permanent and irreversible. The typical amount of ransom ranges from USD\$300 to \$500, and often must be paid in untraceable bitcoins or other cryptocurrencies. After a ransom payment is made, some “lucky” victims may receive a

decryption key to decrypt their files. Symantec™ found that some ransomware had design flaws, cannot decrypt files properly, and victims lost their files forever (Savage, Coogan & Lau, 2015). Many ransomware include a countdown timer in their ransom message to threaten victims to pay the ransom by certain deadlines, or their files will remain encrypted permanently; this adds further stress on victims and pressures them to pay the ransom urgently.

Below is the ransom demand screen of *Ransom.TeslaCrypt* (figure 5).

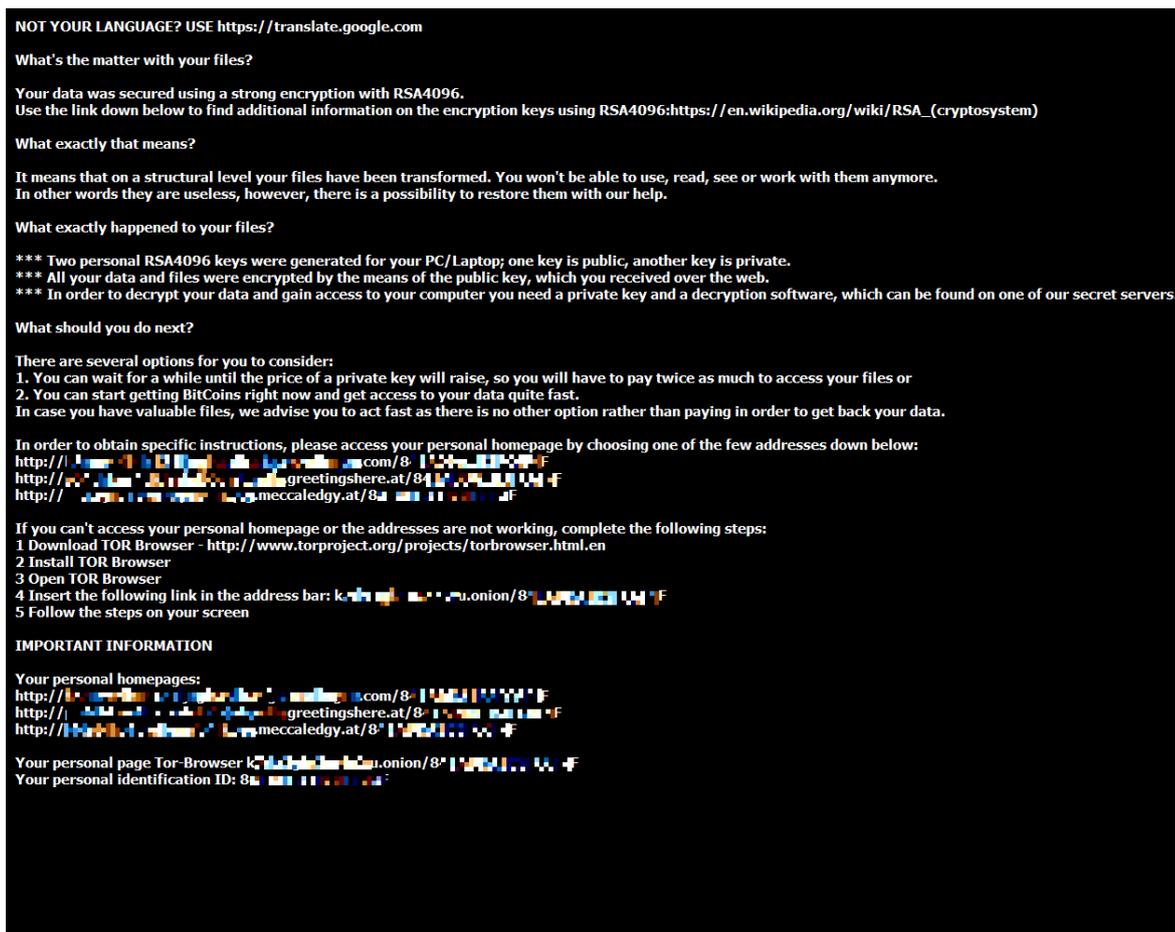


Figure 5 - Ransom Message Displayed by *Ransom.TeslaCrypt* After an Attack

2.4 History of Ransomware and the Scale of the Problem

The first known ransomware was called AIDS, written in 1989; it modified the AUTOEXEC.BAT on the DOS platform to scramble filenames and demanded payments.

Ransomware briefly went out of fashion in the '90s, but returned to prominence since 2005 (Pathak & Nanded, 2016).

Ransomware exert a prominent level of cybersecurity risk. The top countries affected by ransomware are mostly high-income economies; in 2016, the top six are the USA, Japan, UK, Italy, Germany, and Russia (Zavarsky & Lindskog, 2016). Symantec™ estimated the several more damaging ransomwares managed to gain tens of millions of dollars each year, which in turn fed back into criminal activities and encouraged more to follow the examples (O'Gorman & McDonald, 2012). Meeuwisse (2016) noted that a bank found that 50% of customer devices had known forms of malware, and that in 2014, a Symantec™ executive admitted that their software at that time was only able to defeat approximately 45% of cyber-attacks.

The exponentially growing problem of ransomware relies on a combination of multiple factors.

- The potential profit is substantial and very lucrative. the average ransom amount is USD\$300. 7% of the victims are willing to pay the ransom; business operators are more likely to pay (O'Gorman & McDonald, 2012; Simmonds, 2017). The illegal revenue generated often feedback to criminal organizations which can produce more damaging ransomware (O'Gorman & McDonald, 2012).
- The “business” model is mature. Markets to sell high-end mature ransomware as a service allow lower entries for criminals (Liska & Gallo, 2016)
- The popularity of crypto digital currencies like Bitcoin makes it possible to collect payments anonymously and launder the criminal profit, with little chance of being tracked by law-enforcing authorities (Liska & Gallo, 2016).
- The advancement of better hardware allows faster encryption running asynchronously at the background, without noticeable system performance compromise (Savage, Coogan & Lau, 2015).
- The advancement of more complex encryption schemes enabled more efficient and robust encryption of victims' files (Savage, Coogan & Lau, 2015).
- the emergence and adoption of Ransomware as a Service (RaaS) allowed skilled hackers to easily offer their services to a wider criminal population to launch ransomware campaigns, targeting a wider range of victims (Savage, Coogan & Lau, 2015; Liska & Gallo, 2016).

Ransomware affect certain operating systems more than others. Android is the most targeted OS in the mobile market, and Microsoft Windows is the most targeted on PCs, workstations, and servers; these 2 operating systems enjoy high popularity, with market shares close to monopoly, thus having a much larger user base (Savage, Coogan & Lau, 2015). Both are reasonably open with less restrictions and less vendor scrutiny on which applications can be installed and what they could do after installation. Fragmentation of versions is also a challenge; many devices are running older versions of Android or Microsoft® Windows operating systems, and many of them either do not apply regular software updates or patches, or could not be upgraded due to restrictions by outdated hardware (Savage, Coogan & Lau, 2015).

Many ransomware have adopted better localizations. They display ransom messages and payment instructions in a local translation; some may even adjust the payment amount to take into consideration the economic prosperity of the intended markets (Liska & Gallo, 2016). Below is the ransom payment screen of *Ransom.WannaCry* in Japanese, with many other localizations available (figure 6).



Figure 6 - Ransom Message of Ransom.WannaCry in Japanese, with Many Localizations Available

2.5 Current Detection Strategies and Their Limitations

Various vendors of security software have developed different strategies to detect ransomware.

Static Analysis

Static analysis is when malware are analyzed without executing; it is often based on “static” features exhibited by the malware code. Many security software, especially virus scanners, perform scans and searches looking for certain syntactic signatures; those systems have a database of regular expressions describing sequences of bytes or machine instructions that have been studied by security analysts and are considered malicious (Moser, Kruegel & Kirda, 2007). The static features are usually obtained by reverse engineering of malicious code samples captured, and can include portable executable (PE), byte-sequence n-grams, OpCode, and string features (Islam, Tian, Batten, & Versteeg, 2013).

Static analysis is prone to some major drawbacks, and therefore often insufficient to be used alone. The distinctive “signature” features of a malware often require a human expert for analysis; the analysis would take time and would always fall behind the launch of malware attack (Islam, Tian, Batten, & Versteeg, 2013). The static features can be easily bypassed by code obfuscation, which is often used by malware to escape detection (Moser, Kruegel, Kirda, 2007). Static comparison against too many malware signatures can be computationally expensive (Islam, Tian, Batten, & Versteeg, 2013).

Dynamic Analysis

Dynamic analysis is when malware are analyzed during executions; it is often based on “dynamic” features observed during the executions, which can include API calls, file system or Windows Registry activities, and network connections activities (Islam, Tian, Batten, & Versteeg, 2013). Most commercial security software vendors have included their own implementations in their retail products to enhance detection rate, like SONAR by Symantec™, and Heuristic and Behavioral Protection by ESET®, although the implementation details are commercially confidential.

Common utilities used in dynamic analysis include endpoint protection in a real environment and sandboxing in a virtualized environment (Liska & Gallo, 2016). An endpoint protection agent runs in the background of a real environment, tries to identify suspicious activities, and aims to block and report them. This kind of solution can provide more effective protection than what can be provided by traditional signature-based antivirus solutions, and are more likely to catch the ransomware in action. However, a real-time endpoint agent often requires injection into the operating system and maintenance of the highest administrator privileges; it can also consume system resources and lead to performance sacrifice. Sandboxing, on the contrary, creates a virtualized environment to try to trick the ransomware into executing, and analyze the malicious behaviors in isolation to the real system. A virtualized environment is often simpler and is often insufficient to trigger ransomware execution; many ransomware have developed techniques to detect the presence of a sandbox and would refuse to execute.

Dynamic analysis can have several advantages over static analysis. Ransomware follow similar attack patterns, which can be identified during analysis of a zero-day attack

by an unknown type of ransomware (Islam, Tian, Batten, & Versteeg, 2013). It is nevertheless in theory possible to automate the dynamic analysis to extract relevant dynamic features, and apply machine learning algorithms to generate classification results (Tian, Islam, Batten, & Versteeg, 2010; Sgandurra, Muñoz-González, Mohsen, & Lupu, 2016).

Limitations of dynamic analysis have been identified too. It is a time-consuming process; the system must be running, with no guarantee that the ransomware would be activated and exhibit malicious activities (Islam, Tian, Batten, & Versteeg, 2013). The dynamic analysis must be in a controlled environment, often virtual machines, but ransomware may be able to realize the existing monitoring and refuse to behave maliciously. Further research is required to address the relatively high false-positive and false-negative rates of detection; system utilities and compression software can be identified as malicious due to the similarity of their activities to those performed by ransomware (Sgandurra, Muñoz-González, Mohsen, & Lupu, 2016).

2.6 General Strategies for Mitigating Ransomware Risks

Modern security software often cannot detect or deter all ransomware attacks, especially the zero-day attacks abusing recently found security loopholes. For a new outbreak of new ransomware, literature have suggested several common practices and general approaches to defend against zero-day malware attacks.

Harden the system and restrict access

Restricting what users can do based on authentication and authorization remains one of the most commonly deployed method. Because many ransomware spread by macro-enabled Microsoft Office documents or infected PDFs, disabling Macros and regularly patching Adobe Reader proved to be effective. Browser protection and ad-blockers can deter JavaScript-based malware from downloading ransomware into the system.

Patch systems and keep robust backups

Many ransomware make use of operating system vulnerabilities, so one of the best step to boost defense is to patch operating systems proactively and aggressively, including

servers (Brewer, 2016). In an unfortunate event of ransomware attack, businesses with backups can often restore the data and resume normal business activities with least loss or disruption (Mansfield-Devine, 2016).

Educate users to raise awareness on cybersecurity

Companies can make efforts to raise employee awareness on cybersecurity, and education has been proven effective in reducing the likelihood of a ransomware attack. Management are advised to implement robust cybersecurity policies and procedures, and alert employees of new threats if there are any (Luo & Liao, 2007). In companies with effective education programs, employees were found to be less risk-taking and more cautious about their data integrity (Mansfield-Devine, 2016).

2.7 Research Approaches on Ransomware Detection and Prevention without Monitoring File System Activities

Several studies have proposed a few approaches of better detecting new crypto-ransomware and deterring their zero-day attacks, based on certain characteristics of some ransomware and their attack patterns, without monitoring file system activities. The studies on monitoring file system activities will be discussed in chapter 4.

Connection Monitoring and Breaking

Ahmadian, Shahriari, and Ghaffarian (2015) designed a novel system to check the key exchange process during the control-and-command phase. It assumed that if a software communicates with a malicious server, the software itself must be malicious. Their system could detect variants of recently discovered ransomware, but the malicious server addresses must be in their knowledge, for their system to function.

Cryptographic API Hooking

Kolodenker et al (2017) developed PAYBREAK, a system designed to observe the usage of crypto libraries and attempt to intercept encryption keys being used. It assumed that many ransomware variants would use well known cryptographic libraries, like the opensource Crypto++ library or the Windows CryptoAPI. Their implementation demonstrated effectiveness against *Ransom.Cerber* and *Ransom.CryptoLocker*, but it could not detect or prevent attacks by *Ransom.TeslaCrypt* that used neither of the known cryptographic libraries. Their implementation can be easily defeated by using different cryptographic implementations unknown to PAYBREAK.

Honeypot Techniques

Moore (2016) designed a honeypot system with witness files which are constantly monitored for modification or deletion. The study proved a theoretical possibility of such implementation, but Moore (2016) acknowledged that the checking was performed by scripting and may not be computationally efficient.

Chapter 3. A CLOSER LOOK INTO CRYPTO-RANSOMWARE ATTACKS

In this chapter, the characteristics of crypto-ransomware attacks are discussed.

3.1 Crypto-Ransomware Must Attack the File Systems

A long-term study by Kharraz et al (2015) found that although different families of crypto-ransomware carried out attacks with multiple and varied sophistications, they shared similar characteristics from a file system perspective; there is a sudden and significant change in the file system activities involving either the Master Boot Record (MBR), the I/O Request Packets (IRP) or a combination of both. To encrypt user file contents, ransomware must call Windows file system APIs, which will in turn generate IRPs and send them through the I/O stack. Kharraz et al (2015) proposed to protect MBR because the ransomware family of Seftad locked up MBR of victim computers to prevent proper booting into operating systems. Their study was conducted before the discovery of *Ransom.Petya*, which overwrites MBR to gain privileged access to encrypt the Master File Table (\$MFT) of NTFS partitions (Fayi, 2018). \$MFT contains metadata information of files, including how they are stores in different locations of the disk partition. Without the \$MFT, even if MBR is restored, the operating system cannot easily reconstruct the user files (Fayi, 2018).

Crypto-ransomware encrypt user data using strong encryption algorithms and a key obtained from the remote criminal server (Scaife, Carter, Traynor & Butler, 2016, June). Due to the nature and engineering of NTFS file system, which relies on a healthy \$MFT, commonly deployed on most Microsoft® Windows-based operating systems, it is possible and essential to capture such data-centric behaviors of crypto-ransomware develop effective detection and defense mechanisms (Scaife, Carter, Traynor & Butler, 2016, June).

Scareware and Locker ransomware do not actually encrypt user files or attack the underlying file systems, and their damages are often reversible using specialized utilities developed by security software vendors (Symantec, 2017; ESET, 2017).

3.2 Entry Points of Targeted Attacks on File Systems

Selected User Files based on File Types

Many crypto-ransomware, such as *Ransom.TeslaCrypt* and *Ransom.WannaCry*, were found to selectively encrypt files, often based on file types. They quietly search and index victims' files at the background. They appear to select files based on filename extensions, and target documents, photos and presentation. According to Scaife, Carter, Traynor & Butler (2016, June), the *pdf*, *odt*, *docx*, *pptx* and *txt* file types are the most commonly and frequently attacked. Selecting and encrypting certain types of files is more effective, allowing the encryption to complete within the shortest time possible, before the intrusion is noticed. Operating system modules usually do not get encrypted; there is no valuable individual data in them and they can be recovered by reinstalling the operating systems.

Master Boot Record

Since Microsoft Windows Vista, Windows offers two different disk-partitioning options: Master Boot Record and Globally Unique Identifier Partition Table (GPT) (Halsey & Bettany, 2015). The partitioning information describes how the logical partitions that contain file systems are organized on the disks. MBR was introduced earlier and was designed for BIOS-based systems; it also contains a piece of simple executable code at address 0, known as "bootstrap" or "bootloader", which will in turn select and load the actual operating system. MBR must be in sector 0 of the disk, and the maximum possible size of MBR on the disk is 512 bytes. GPT works with UEFI-based systems and contains a protective copy of MBR at address 0. UEFI provides a simple Boot Manager to select an operating system, and each operating system provides its own bootloader. When SecureBoot is enabled in UEFI settings, only properly signed and trusted modules can be loaded during the boot process.

A few variants of ransomware, such as *Ransom.Petya* and *Ransom.Redboot*, would instead attempt to attack the Master Boot Record (MBR) first, in order to gain control

during the next reboot to be able to encrypt the master file table (\$MFT), a database containing information about every file and directory on an NTFS volume. \$MFT cannot be modified directly on Microsoft® Windows operating systems; all file operations must be performed using standard file system APIs defined by the Windows operating system, which makes changes to the \$MFT accordingly. Therefore, encrypting the \$MFT is carried out in a two-staged attack, which involves (1) modifying the MBR and rebooting to gain control and (2) encrypting the \$MFT after the reboot (ESET, 2017). Most unsophisticated users do not need to perform operations that requires writing into MBR of the boot drive, unless they are installing a new operating system.

On a BIOS/MBR-based system, *Ransom.Petya* would write its own bootloader into the MBR, crash the operating system to force a reboot into its own malicious bootloader, and impersonate CHKDSK (the Windows Check Disk command line utility) while encrypting the \$MFT. The MBR could be repaired by running disk utilities. However, because detailed file information can be either stored in \$MFT entries, or external spaces described by \$MFT entries, it would be difficult and sometimes impossible to rebuild \$MFT to regain access to files; the encryption of \$MFT in combination with a compromised bootloader leaves most victims no choice but to pay the ransom. On a UEFI/GPT-based system with SecureBoot enabled, *Ransom.Petya* would still write its own bootloader into the MBR. However, when the system reboots, the Boot Manager inside UEFI cannot find a signed and trusted bootloader, and would fail to boot the operating system. As a result, *Ransom.Petya* does not damage the actual \$MFT on a UEFI/GPT-based system, but would still make it unusable until the GPT is repaired.

The attack patterns of *Ransom.WannaCry* and *Ransom.Petya* are examined in more details as case studies in Chapter 9 and 10.

3.3 Preference to Access Non-System Files and Folders

Barreau and Nardi (1995) found that for electronic file storage and organization on operating systems, computer users were more likely to store their files in certain folders of their choice. They also concluded that users overwhelmingly preferred location-based file search, by going to the most possible folder and performing a file listing and browsing.

There are three special folders on Microsoft Windows platforms: “Windows”, “Program Files”, and “Documents and Settings” (Agrawal, Bolosky, Douceur, & Lorch, 2007). The “C:\Windows” folder contains modules of the Windows operating system, installed drivers, system logs and services essential for running the Windows operating system (Agrawal, Bolosky, Douceur, & Lorch, 2007; Halsey, & Bettany, 2015). The “Program Files” folders, including “C:\Program Files” and “C:\Program Files (x86)” contain installed applications that are not essential to the Windows operating system (Agrawal, Bolosky, Douceur, & Lorch, 2007; Halsey, & Bettany, 2015). The “Documents and Settings” folder on Windows XP, which has later been renamed as “Users” since Windows Vista, contains the folders for each user account on the computer including domain profiles if applicable (Agrawal, Bolosky, Douceur, & Lorch, 2007; Halsey, & Bettany, 2015). A few documents and shell folders are created by Windows inside each user folder, such as “Documents”, “Pictures” and “Desktop”, where most computer users usually store their user files (Halsey, & Bettany, 2015).

Crypto-ransomware appeared to demonstrate strong preference to access non-system files and folders. Continella et al (2016) collected the logs of approximately 1.5 billion IRP requests, on 11 “benign” machines without ransomware infections and on 19 “infected” machines during ransomware attacks. Each IRP request contained information on where the target file was located, but if the files were not located in “Windows” or “Program Files” folders, the folder names in the path were hashed to protect confidentiality of the users. The raw results of their study were re-examined based on the location of the target files, to count what percentage of files accessed were in the “Windows” folder, in “Program Files” folders or in other folders. Results were rounded to the nearest 1%. The statistics showed that the average percentage of IRP access to other folders is 47% on benign machines (the lime green bars), but it rose significantly to the average of 87% on infected machines during ransomware attacks, showing that crypto-ransomware attacks increased the percentage of IRP access to files and folders that were not in in the “Windows” folder or in “Program Files” folders. Given that computer users do not usually store user files in those folders, there appeared to be a strong preference for crypto-ransomware to access user files. The finding led to the decision to let RANDETER only monitor selected user folders, instead of performing system-wide monitoring on all file system activities.

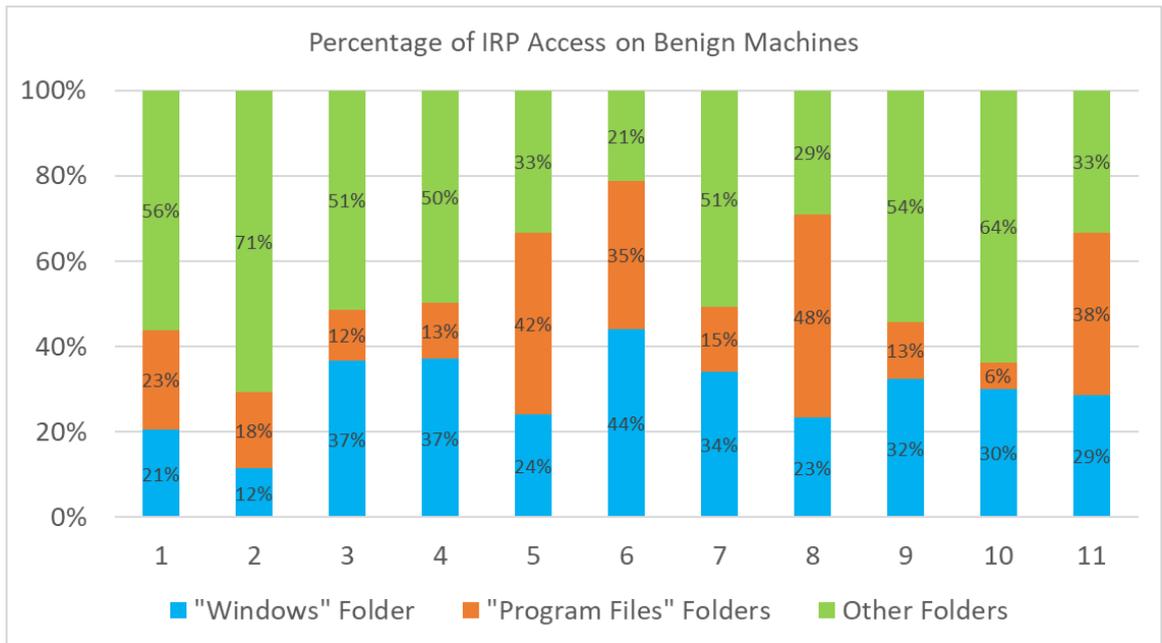


Figure 7 - Percentage of IRP Access on Benign Machines (Data Source: SHIELDIFS Study by Continella et al, 2016)

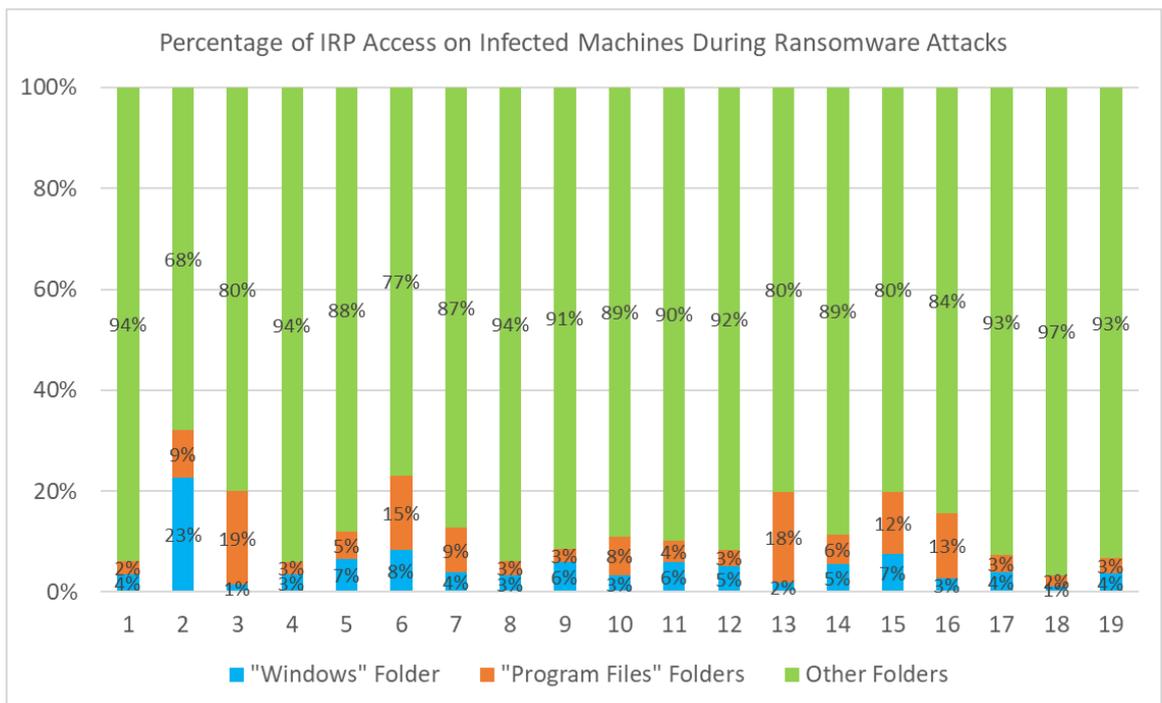


Figure 8 - Percentage of IRP Access on Ransomware-Infected Machines (Data Source: SHIELDIFS Study by Continella et al, 2016)

3.4 Aggressiveness towards File Systems

Kharraz et al (2015) noted the unusual and aggressive file system activities by crypto-ransomware, which generated a large amount of create, read, write and rename operations on many different types of non-system files. Continella et al (2016) discovered that statistically, crypto-ransomware performed more diverse types of file operations (create, read, modify, rename, delete, move etc.), on more diverse types of files (all files within a wider set of file extensions), and much more frequently. Kharraz & Kirda (2017) concluded that “modify” and “move” were potentially more dangerous file operations that were more frequently involved in file encryption performed by crypto-ransomware.

The dataset of IRP statistics shared by Continella et al (2016) was examined to calculate the speed of IRP requests to modify file contents or file information of PDF files outside of “Windows” or “Program Files” folders. IRP logs of PDF files were selected to be evaluated, because PDF files were the most abundant file type of documents in user files on their test systems. It was found that ransomware-infected systems displayed much higher numbers of such IRP requests per minute than the benign machine, which suggested that a “Speed Rule” on file system activities could be implemented to detect crypto-ransomware activities on file systems.

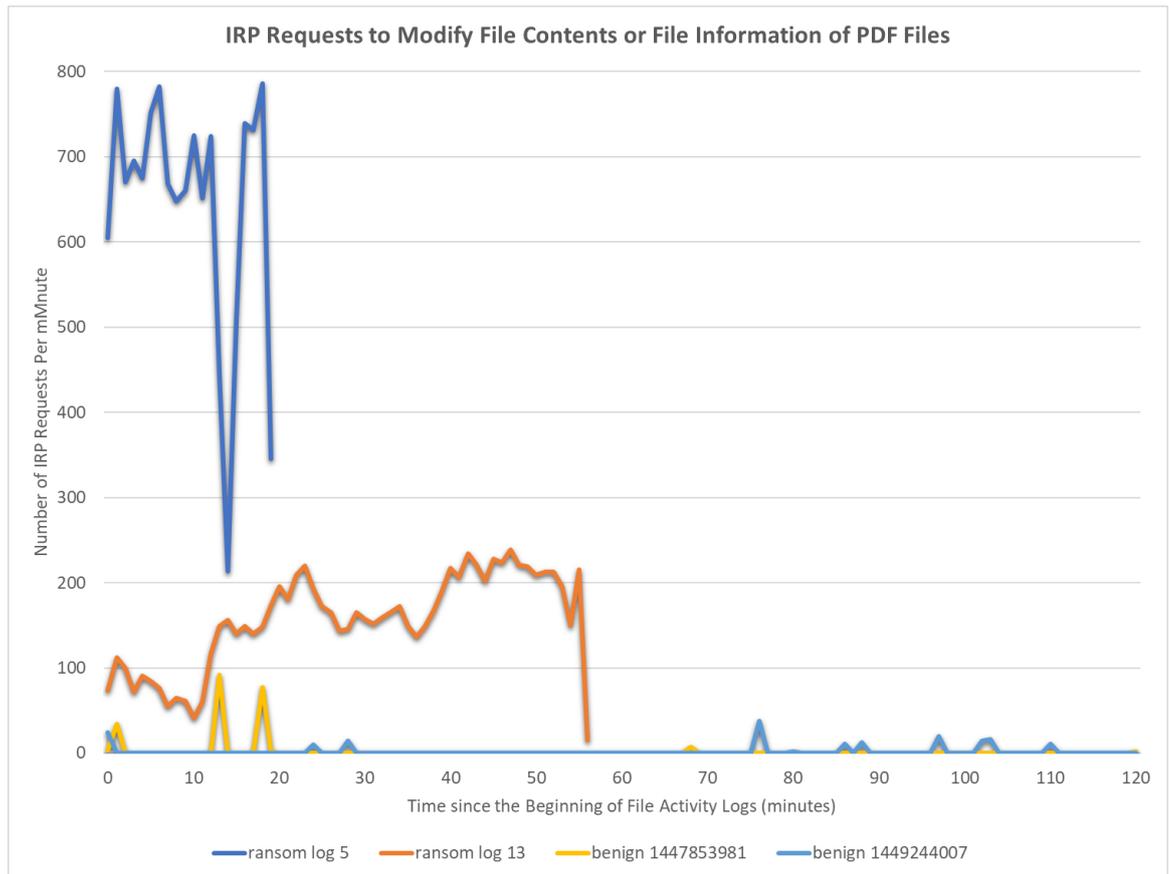


Figure 9 - Number of IRP Requests to Modify File Contents or File Information of PDF Files

(Data Source: SHIELDIFS Study by Continella et al, 2016)

3.5 Greediness for Modifying More Diverse Types of Files

Kharraz et al (2015) and Continella et al (2016) both noted that crypto-ransomware could modify the file contents of more diverse types of files (based on file extension names) than many other benign applications could, because of the need to encrypt as many files as possible and because some types of crypto-ransomware generate random filenames during the encryption process.

The dataset of IRP statistics shared by Continella et al (2016) was examined to count the number of different file types (based on file extension names) modified by the program which modified most file types outside the “Windows” and the “Program Files” directories in each log. It was found that:

- On the 11 benign systems not attacked by crypto-ransomware, the number of different file types that can be modified by a single program was between 5 and 51.
- On the 19 systems during ransomware attacks, all but 2 systems had a single program in each system that was able to modify thousands of different file types. Upon the inspection of the actual log of each log, many different file extension names were found to be associated with those applications.

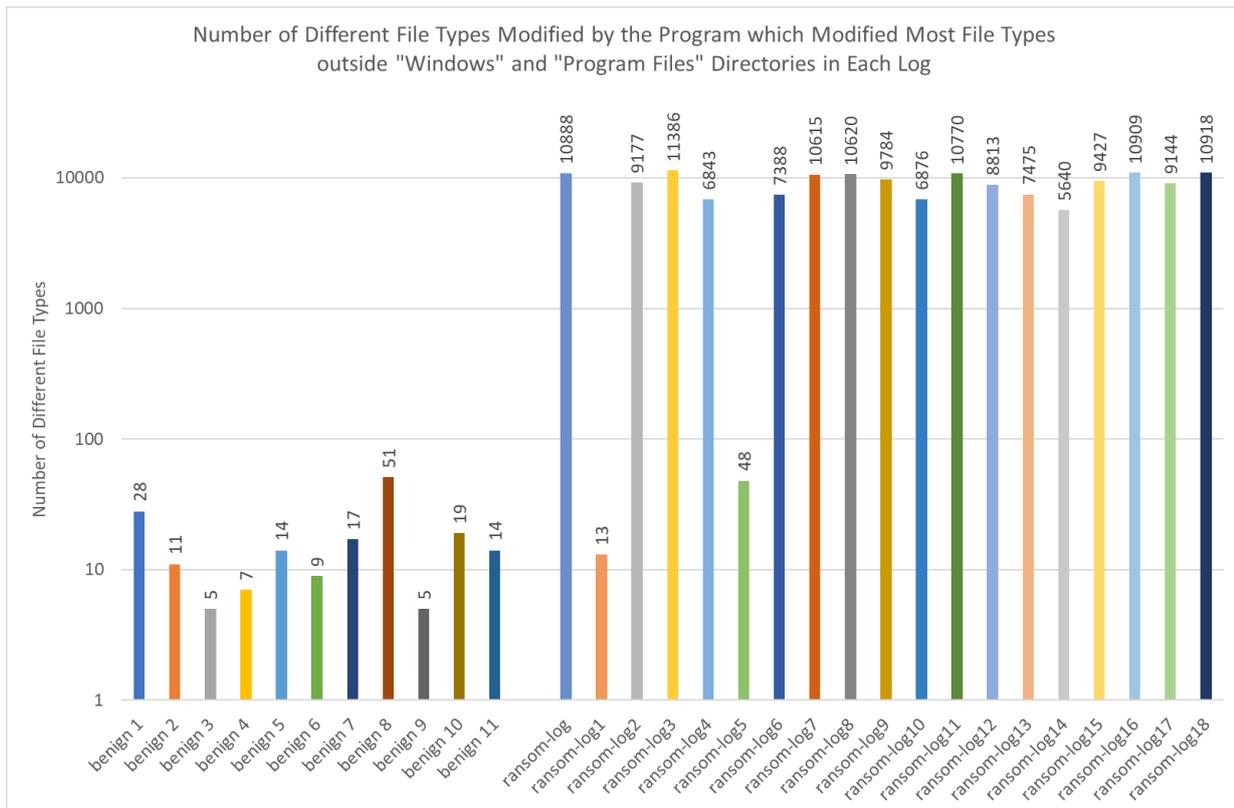


Figure 10 - Number of Different File Types Modified by the Program which Modified Most File Types outside "Windows" and "Program Files" Directories in Each Log (Data Source: SHIELDIFS Study by Continella et al, 2016)

The logs of ransom-log1 and ransom-log5 were re-examined, due to their low number of file types of modified files. It was found that for ransom-log1, "svchost.exe" was responsible for creating or modifying 13 different types of files, including BMP files; "svchost.exe" is not a known viewer editor of BMP files, so it's likely that "svchost.exe" was injected with ransomware processes. In the log of ransom-log5, a program named "lknwy-bc.exe" created or modified files of 48 different file types, including jpg, js, zip,

docx, xlsx; it suggested that “lknwy-bc.exe” was either a sophisticated multi-purpose editor, or a ransomware executable. Searching the Internet for “lknwy-bc.exe” returned no results, suggesting the filename “lknwy-bc.exe” of the program appeared to be randomly generated and may indicate it was a ransomware executable.

Based on the findings of inspecting SHIELDIFS study data, it is believed that:

- On a benign system with only legitimate applications installed, many applications that can modify file contents are designed with pre-determined functional specifications, and only know how to open and decode certain types of files.
- On a system infected by crypto-ransomware, ransomware executables can be involved in modifying file contents of more diverse file types (500 times more on average).
- Most of the crypto-ransomware studied by Continella et al (2016) directly modified file contents, but two samples deleted original files and created new ones.
- Some of the file modifications were initiated by “svchost.exe”, which may suggest that either crypto-ransomware masqueraded as “svchost.exe” during the attacks, or they injected into the actual Windows component of “svchost.exe” to carry out attacks.

Therefore, a “File Type Rule” on file system activities could be implemented to detect file modifications performed by crypto-ransomware attacks.

3.6 Other Statistical Findings

Upon further inspecting the file event log by the FILE PATROL module of RANDETER, and the dataset of IRP statistics shared by Continella et al (2016), it was found that:

- On a system not infected by any ransomware
 - Some Windows system modules, like rundll32.exe and svchost.exe, do not usually modify user files. The Windows File Manager Explorer.exe may perform “delete” and “rename” operations in batches on user files, but only “modify” the content of “zip” files that are placed in user directories.

- Utilities, like Disk Cleanup (cleanmgr.exe), may perform many file operations within short period of time, but the file operations are usually of the same type, for example batch-deleting and batch-renaming. Most utilities do not modify file contents. The utilities used to compress files only “modify” files of archive formats.
- When benign programs create or modify files due to human operations, the speed is usually no more than 1 file operation per second; in case when temporary files are involved, the speed is usually no more than 5 file operations per second.

- On a system infected by ransomware
 - The process names are often random, although some ransomware use fake filenames or descriptions to masquerade as Windows modules.
 - There is often an alternating pattern of file operations. A file gets modified and often renamed or moved, before the next file gets modified and so on.

The aggressiveness of crypto-ransomware towards file systems provided sufficient statistical power to differentiate them from benign programs.

Chapter 4. IMPLEMENTATIONS MONITORING

FILE SYSTEM ACTIVITIES

In this chapter, a few existing implementations that monitor file system activities of ransomware are discussed.

4.1 Current Research Articles

In a long-term ransomware study between 2006 and 2014, Kharraz et al (2015) analyzed 1,359 ransomware samples, and concluded that most real-world ransomware attacked user files in similar patterns on the Microsoft® Windows platform, and the attack patterns were distinguishable from file system activities of benign processes. They found a common theme that all crypto-ransomware samples modified either the files of the file system, or the Master File Table of NTFS file system. They suggested that monitoring I/O requests of file system activities and the Master File Table of NTFS file system could be effective in detecting and preventing most zero-day ransomware attacks. Their study was the first to analyze a significant quantity of ransomware samples, and contributed to the theoretical base of strategically monitoring, analyzing and protecting file systems integrity.

Since then, there have been several studies focusing on analyzing file system I/O operations, some in combination with other indicators. All those studies have completed successful implementations, based on the evidence that file system I/O activities by crypto-ransomware were significantly different to those of benign programs.

Table 1 - Comparison of Existing Implementations Monitoring File System Activities of Ransomware

Authors	Date	Project Name	Detection Approach	File Recovery	Claimed Effectiveness	Overhead
Scaife et al	2016 Jun	CRYPTOLOCK	File type changes, similarity, entropy changes	-	Medium loss of 10 files	Performance overhead 1ms latency
Kharraz et al	2016 Aug	UNVEIL	File system monitor, desktop lock monitor. Virtualized user environment	-	Detection rate 96.3%. False positive 0.0%. User-level ransomware only.	- (not an endpoint solution)
Continella et al	2016 Dec	SHIELDIFS	File system I/O access patterns, cryptographic primitives	Automatic by workflow	True positive 100%. False positive 0.27%	Storage overhead 0.74% Performance overhead 26.2%
Kharraz & Kirda	2017 Sep	REDEMPTION	File content changes and behavior features	Full data recovery	True positive 100%. False positive 0.2%	Storage overhead 5.6% Performance overhead 2.6%

Scaife et al (2016) developed CRYPTOLOCK, which aimed to halt malicious process as early as possible, if they were found to be tampering with large amount of user data. The system checks several pre-defined indicators during runtime, like file type changes, similarity, file deletions, file type funneling. Their system managed to achieve a satisfactory 100% detection rate against 492 real-world ransomware samples. They were able to detect a ransomware attack, after a medium loss of 10 files permanently damaged by ransomware and irrecoverable, although CRYPTOLOCK provided no file recovery mechanism. They claimed 1ms latency on most file operations because of running CRYPTOLOCK. RANDETER aims to achieve a total file damage of no more than 5 files per ransomware attack; the tolerance threshold is configurable depending on use cases and administrative settings. CRYPTOLOCK and RANDETER use similar ransomware indicators, but RANDETER does not compute file similarity changes or file entropy changes. Files that are already compressed, like zip and JPEG files, have almost random and uniform byte distributions in file contents (Karresand & Shahmehri, 2006). The computations can be resource-intensive; the changes of file entropy may not be a good indication of ransomware-initiated encryption on those already compressed file formats (Sencar & Memon, 2009; Taubman & Marcellin, 2012).

Kharraz et al (2016) proposed UNVEIL, a dynamic analysis system that automatically created an artificial but realistic execution environment, and aimed to detect abnormal file system activities and screen locking behaviors. UNVEIL was able to identify previously unknown evasive ransomware, i.e. the new ransomware family of *SilentCrypt*, before most security vendors. However, the consumption of system resources by UNVEIL was high, making it unsuitable to become an endpoint solution to be deployed to real user devices for everyday ransomware protections. There were no data provided on the performance overheads of UNVEIL. RANDETER is an endpoint implementation on Microsoft® Windows platforms that can be deployed to real user environments, and aims to evaluate benign or malicious disk and file access patterns. RANDETER does not create virtualized environments, to try to minimize impact on user experience or system performance.

Continella et al (2016) implemented SHIELDIFS, a self-healing, ransomware-aware file system that checked both low level file system I/O activities and cryptographic primitives in processes in system memory, and updated a set of adaptive profiling models. Their main goal was to create a Windows file system driver that combined automatic

ransomware detection and transparent file-recovery. Their system used copy-on-write mechanism, which created duplicate copies of files to be modified by the file system activities, instead of overwriting the files directly. The file activities were then analyzed and if benign, the file changes would then overwrite the original files; if the file activities were deemed malicious, the file changes would be discarded. SHIELDIFS could roll back most damages committed by ransomware attacks, with a low false positive rate, but is susceptible to targeted evasion (ransomware operating below the alarming threshold that will trigger SHIELDIFS alarms) and denial of service attacks on file systems (ransomware filling up the shadow drive so no spare disk space is available for copy-on-write file protection). While SHIELDIFS appeared to be a significant improvement over pure-detection approaches (sandboxes or pipelines), it made assumptions that the ransomware would use a known cryptographic library, would pre-compute the key schedule in predictable locations in the memory, and the encryption key schedule could be scanned and located easily. Continella et al (2016) acknowledged the estimated average runtime overhead was $0.26\times$, possibly due to employing copy-on-write technique to shadow protected file copies. On inspecting the raw datasets of SHIELDIFS, it was found that the file sizes of files used in their evaluations were relatively small; most were under 1MB. It is possible that the performance overheads could be higher for users like content creators, who often need to create or modify much larger files. The increased performance overheads may be insignificant for a home user, but could impact performances of data-heavy platforms. RANDETER seeks to implement a more generic approach resistant to unknown cryptographic methods, and aims to minimize performance overheads even for files of larger sizes.

Kharraz & Kirda (2017) presented REDEMPTION, a generic real-time monitoring system that monitors file system I/O request patterns on a per-process basis. REDEMPTION performed system-wide monitoring, and used content-based features (entropy, file overwrite, delete operations) and behavior-based features (directory traversal, output file types, access frequency), to calculate a malice score and estimate how malicious a process could be. While the implementation of REDEMPTION achieved a low runtime overhead of $0.026\times$, its decision to calculate a malice score per-process could potentially mark legitimate Windows modules as malicious, when some ransomware inject themselves into legitimate Windows processes like svchost.exe, or when ransomware execute in the form of PowerShell scripts. RANDETER overcomes this issue by evaluating the file system

activities per process and per directory; file activities within the same directory are computed together, so multi-threaded or multi-process ransomware attacks would be detected. In addition, REDEMPTION checked the entropy ratio of data blocks in I/O buffers to determine whether the data to be written into disks have been compromised by ransomware; RANDETER does not check the entropy changes of file operations, because computing entropy changes per file operation can introduce resource overheads, and high entropy changes do not necessarily reflect crypto-ransomware activities, especially for already compressed file types like JPEG (Taubman & Marcellin, 2012).

However, none of those studies implemented monitoring on disk activities attempting to modify MBR. MBR can be modified either via an IRP request of `FileReadWrite`, or via an SCSI device control command of `SCSI_PASS_THROUGH`, practiced by one variant of Petya Ransomware. RANDETER is a dual-module solution that provides additional disk activity protection on MBR against any forms of tampering.

4.2 Current Patents

A few patent applications demonstrated patent inventors' attempts to detect ransomware attacks via monitoring file system activities.

Hunt & Tiernan (2018) proposed *Ransomware Protection For Cloud File Storage* (US patent US20180007069A1), an approach that detected ransomware attacks on an endpoint device (such as a PC), based on that assumption that local changes of files would be synchronized to backup copies of the same files in the cloud file storage, and would trigger the recognition of sequences of file operations that may indicate ransomware activity. The recognition is based on sequences of cloud storage APIs called due to the file activities caused by file changes. Their implementation relies on internet connection and proper file synchronization with the cloud server; in theory, a ransomware attack can still infiltrate their system by terminating the internet connection before starting to attack or attacking when offline, or killing the endpoint synchronization client application to result in synchronization failure. Furthermore, their implementation does not monitor other areas that are part of the file system but do not contain user files, such as MBR, that are not synchronized to or stored on the cloud. RANDETER can function without any

Internet connection, and its FILE PATROL module can monitor the file system activities directly. RANDETER also protects the MBR via the PARTITION GUARD module.

Schmugar et al (2018) designed a system of *Mitigation of ransomware* to perform dynamic detection of ransomware in a real operating system environment on an electronic device (US patent US20180018458A1). They proposed two possible algorithms: (1) if an application attempts to modify a type of files which this application is not allowed to modify, the file access would be rejected. (2) when an application attempts to modify a file, and the entropy change (how different the file content is going to be after the modification) is above a pre-determined threshold, the modification would be rejected. Unlike RANDETER, their system is not restricted to Microsoft Windows platform, and could be deployed to electronic devices and have the checking logic wired into the hardware. However, their way of achieving application authorization is by using “an access control list that includes a list of files that the application is authorized to modify”, which is not flexible because all files to be modified must be added to the access control list, which the patent does not describe how to. It is also subject to “masquerading attack”, when a ransomware disguises itself as the authorized application and attack files on its access control list. RANDETER checks whether an application is recognized and behaves in a prescribed manner to minimize the risk of “masquerading attack”. In addition, using a pre-determined entropy threshold value can be arbitrary and computationally expensive, and can cause issues: a small change such as brightness in some image files can result in huge entropy changes, and that ransomware can in theory only encrypt the dictionary of a large compressed file, which will result in a small entropy change but still make the file inaccessible. RANDETER does not check file entropy changes, but implements a system of enhanced access control with multi-tier security rules that is specific for each recognized application.

4.3 Windows Defender “Controlled Folder Access” and Masquerade Attacks

Since the Windows 10 Fall Creators Update (build 1709), released in October 2017, Microsoft has implemented the feature named “Controlled Folder Access”, designed to help protect files in key system folders from being modified by malicious or suspicious applications, by allowing users to define and control what applications can access certain folders (Microsoft, 2017). Folders where users commonly store their files, like “Documents” and “Pictures”, are by default included in the list of folders to protect. Microsoft programs like “WINWORD.exe” appears to have been whitelisted by Microsoft to access the protected folders, while third party applications like 7-Zip must be manually added by system administrators to the whitelist.

“Controlled Folder Access” is part of the Windows Defender Exploit Guard, which represents a significant effort by Microsoft to combat malicious intrusions. However, the “exclusion list” appears to be purely based on program filename and path. This approach creates three further issues.

- 1) Each executable program must be added individually, which can require advanced knowledge of computer systems and software.

Take the open-source 7-Zip as an example. The File Manager of 7-Zip with GUI interface is by 7zFM.exe, but the right-click menu of 7-Zip is by 7zG.exe. Both must be added to the exclusion list to allow the proper functions of 7-Zip.

- 2) It is a black-and-white approach that grants the excluded programs power to modify anything protected.

Unsuspected users can be tricked into adding malicious programs into the exclusion list. Once a program is added to the exclusion list, there is no further checking to prevent it from behaving in a non-prescribed manner. A ransomware developer can release a free benign utility to be added to the list, and later convert it into a malicious ransomware in a subsequent product update, while retaining its access rights to protected folders.

- 3) The exclusion list is susceptible to “Masquerade Attacks”, when a malware pretends to be a legitimate software, to bypass “Controlled Folder Access” and attack files in protected folders.

Steps to replicate the Masquerade Attacks described in issue 3:

- (1) Use Hyper-V on Windows 10 Pro build 1709 (2017 Fall Creators Update).
- (2) Create a new virtual machine. In the New Virtual Machine Wizard, assign 4096MB dynamic memory. Configure network connection as “Not Connected”. Set virtual hard disk as 60GB. Set “Installation options” as “Install an operating system from a bootable CD/DVD-ROM” and load from the image file of the Windows 10 Pro build 1709 (2017 Fall Creators Update) installation ISO image. Finish the wizard.
- (3) Right click the virtual machine. In “Processor”, set the “Number of virtual processors” to be “4”. Confirm.
- (4) Boot the virtual machine. In the Windows installation wizard, choose “Windows 10 Pro” (last modified on “9/29/2017”). Read and accept the license terms. Complete the installation.
- (5) Install Office 2016 from the official ISO image. Run “Word 2016”, accept the terms and conditions. Use the “Task Manager” to find out where the “Winword.exe” is installed. For example, C:\Program Files (x86)\Microsoft Office\root\Office16
- (6) Open the Windows Defender Security Center. On its “Home” tab, the “Last threat definition update” was on “August 25, 2017”.
- (7) Still in Windows Defender Security Center, on “Virus & threat protection” tab, open “Virus & threat protection settings”, turn on “Controlled folder access”, but do not modify any of its other settings.
- (8) Copy some random docx, xlsx and pdf files into “My Documents”.
- (9) Shut down the virtual machine. Take a checkpoint as “Checkpoint 0”.

Testing with *Ransom.BadRabbit* without impersonations

- (10) From the host system, download a zipped file of the *Ransom.BadRabbit* sample from VirusShare.com (SHA-256 value

630325cac09ac3fab908f903e3b00d0dadd5fdaa0875ed8496fcbb97a558d0da).

This sample was submitted to VirusTotal on October 26, 2017, which was later than the definition update date of the Windows Defender on the virtual machine.

- (11) Copy the zipped file of the *Ransom.BadRabbit* sample into the virtual machine, and unzip the file in the virtual machine (unzip password required). Execute the ransomware.

Result: “Controlled folder access” blocks the program. Attack unsuccessful.

Testing with *Ransom.BadRabbit* impersonating as Microsoft Word

- (12) Copy the zipped file of the *Ransom.BadRabbit* sample into the virtual machine, and unzip the file in the virtual machine (unzip password required).
- (13) Rename the ransomware as “WINWORD.exe”.
- (14) Copy the fake “WINWORD.exe” to the following directory to replace the real “WINWORD.exe”. Administrative privilege is required.
C:\Program Files (x86)\Microsoft Office\root\Office16
- (15) Delete other files including the zip file from the virtual machine. Leave nothing in “My Documents” or on the desktop.
- (16) Run the fake “WINWORD.exe”. Wait for 10 minutes.

Result: the machine would reboot into the malicious bootloader; attack by BadRabbit is successful.

Testing with *Ransom.BadRabbit* impersonating as 7-Zip

- (17) Restore the virtual machine to “checkpoint 0”. Install 7-Zip.
- (18) Add “7zFM.exe” to the exclusion list of “Controlled folder access”.
- (19) Copy the zipped file of the *Ransom.BadRabbit* sample into the virtual machine, and unzip the file in the virtual machine (unzip password required).
- (20) Rename the ransomware as “7zFM.exe”, and replace the installed actual “7zFM.exe” in “Program Files”
- (21) Run the fake “7zFM.exe”. Wait for 10 minutes.

Result: the machine would reboot into the malicious bootloader; attack by BadRabbit is successful.

Chapter 5. RANDETER THEORETICAL

CONCEPTS

In this paper, a new method of detecting and deterring ransomware attacks is proposed, inspired by standard operational doctrine in anti-terrorism as promulgated by the US Army (Department of the Army, 2011), known as the Five AT Principles (assess-detect-defend-warn-recover). The similarities between terrorist attacks and ransomware infections were first reviewed, before a simplified ransomware detection model was provided, which can be characterized as “patrol-and-defend”. This is a subset of the Five AT Principles. The goal of RANDETER is to detect and deter unwanted massive user data encryption as early as possible, while minimizing performance overheads and disk usage. It is achieved by guarding commonly targeted attack entry points on the file system, and neutralize any suspicious targets that are unrecognized or do not behave in prescribed manners.

5.1 Similarities between Ransomware Attacks and Terrorism Attacks

The use of cyber as an attack vector by terrorists is well-known (the Sony Pictures attack is a contemporary example; Ismail, 2017). However, there are broader characteristics which define the attack surface and potential impacts in both cases. These include:

- Both target civilians and innocent victims
- Both aim for the maximum damage possible
- Both are rare events, making predictions difficult; most victims only encounter the attack once
- Both select targets that are easy to attack and that would suffer maximum damage or consequences
- Both seek to explore and employ novel attack patterns (zero-day attacks) to evade existing detection mechanisms

- Victims affected are often unprepared; guidelines of general precautions available, but it is difficult to train the public to take specific precautions

To minimize civilian deaths and injuries, governments typically pursue both counter-terrorism and anti-terrorism strategies. Counter-terrorism tends to be more strategic, creating policy frameworks and regulatory settings that aim to deter terrorists from achieving their goals, while anti-terrorism refers to defensive measures that can reduce the impact of terrorist activity when it is likely to occur. Anti-terrorism operations aim to minimize civilian deaths and injuries without significantly interfering with public normal functions; the main aim of RANDETER is to minimize data loss or compromise, by actively pursuing and monitoring system activity for threats which are likely to occur.

5.2 The Five Guiding Principles of Anti-Terrorism and Their Relevance to Anti-Ransomware Practice

Anti-Terrorism is defined as the defensive measure applied to reduce vulnerability of individuals and properties to terrorist acts. Because of the similarities, the Five Anti-Terrorism principles can also be similarly applied to anti-ransomware practice.

Assess: monitoring and evaluating the current situation. RANDETER constantly monitors the MBR and important files, and their file system activities are evaluated to identify abnormalities at the background, based on statistical analysis on file system activities of benign systems without ransomware infections and systems that have been attacked by ransomware.

Detect: identifying an act of aggression and analyzing its validity. RANDETER has a set of pre-defined rules on what are benign activities and what are suspicious; the rules are defined according to statistics obtained by previous studies, and by artificial interpretation of what is possible. It seeks to differentiate human-initiated file operations from pre-programmed automatic file operations.

Defend: protecting an asset from aggression by delaying or preventing an adversary's movement towards the asset or by shielding the asset from threats. RANDETER does not

aim for no loss of important file or data at the expense of system performance degradation; it defends MBR and key directories to deter ransomware attacks.

Warn: the knowledge and communication of a broad range of danger, from general to specific and imminent threats. The anti-terrorism response team can warn the affected victims to take actions, and inform the control center to propagate the intelligence to more endpoints.

Recover: dealing with the need to recover operations as the response to a terrorist incident occurs. If possible, attempts should be made to revert the damages done by terrorist attacks.

The first three stages of anti-terrorism doctrine were adapted and characterized as Patrol-and-Defend, where the reactive nature of assessing and detecting was replicated, with a strong bias towards suspicion about all system activities, such as writing types of data, all of which can be associated with activity that could indicate ransomware activity.

5.3 The Patrol-and-Defend Approach Explained

The traditional approach to detect ransomware has mostly been the same approach that has been used to detect virus and malware, which is in short infect-and-immune, like the way human immune system functions (figure 11). This approach relies on three critical steps in order: malware infection -> malware captured and analyzed -> malware definition updates pushed to uninfected computers. It does not protect “unimmunized” victims against zero-day new attacks. Nevertheless, all three steps can fail due to different factors. For example:

- Malware infection may not happen on computers accessible by security vendors.
- Malware may self-destroy or wipe out traces in the compromised systems, leaving only the damages, before they can be captures or analyzed.
- Malware definition updates require internet connection and the execution of update routines; computers that cannot connect to the Internet or cannot run the capable security software cannot be protected.

Therefore, the infect-and-immune approach may not be sufficient to detect and deter ransomware, one of the most aggressive and destructive forms of malware.

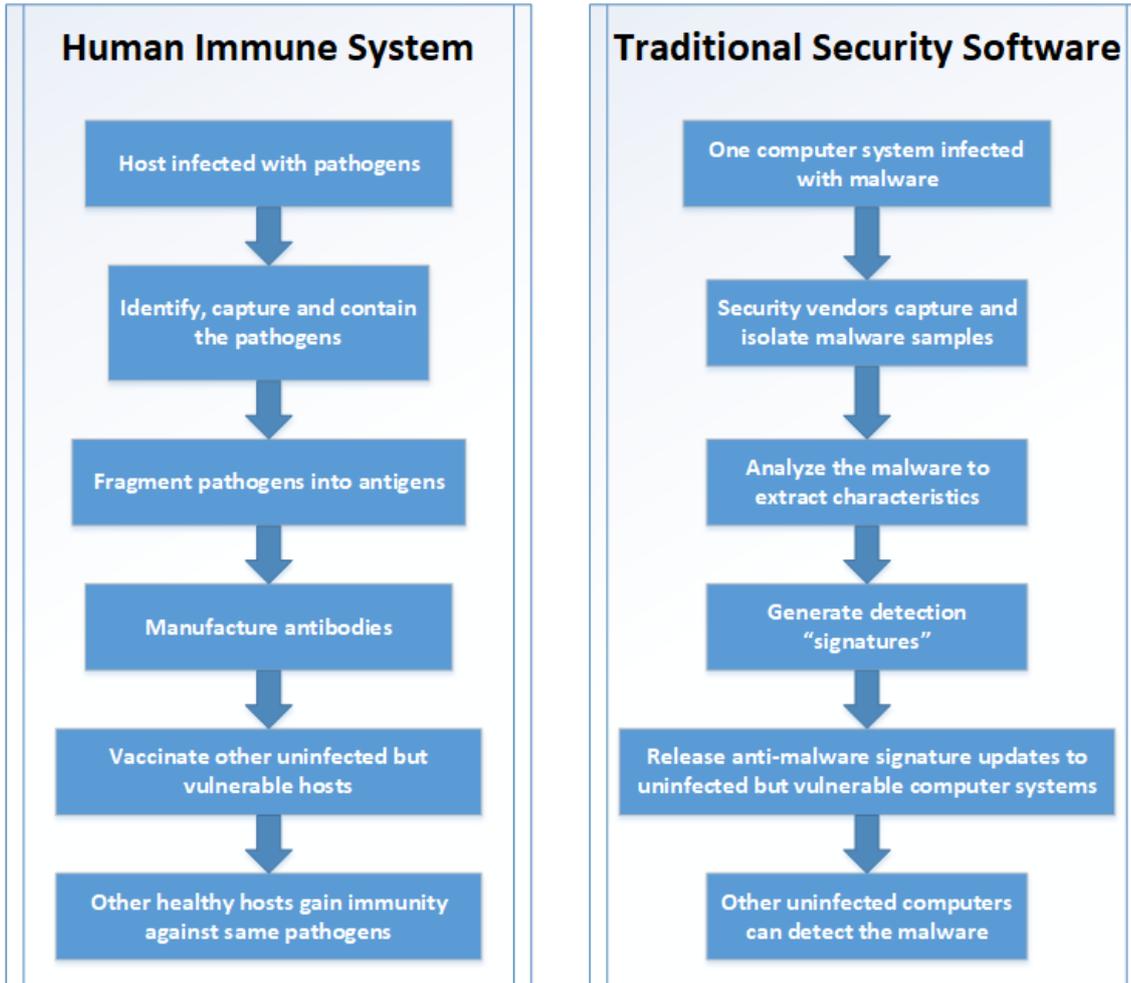


Figure 11 - the Traditional Infect-and-Immune approach against Malware

The RANDETER system employs the Patrol-and-Defend approach that is similar to Police anti-terrorism practices, by protecting the two attack targets identified by previous studies, which are the MBR and the non-system client files (figure 12). RANDETER does not intentionally seek to differentiate the file system IRP patterns. Instead, it aims to maintain “law and order” and deter out-of-ordinary levels of activities.

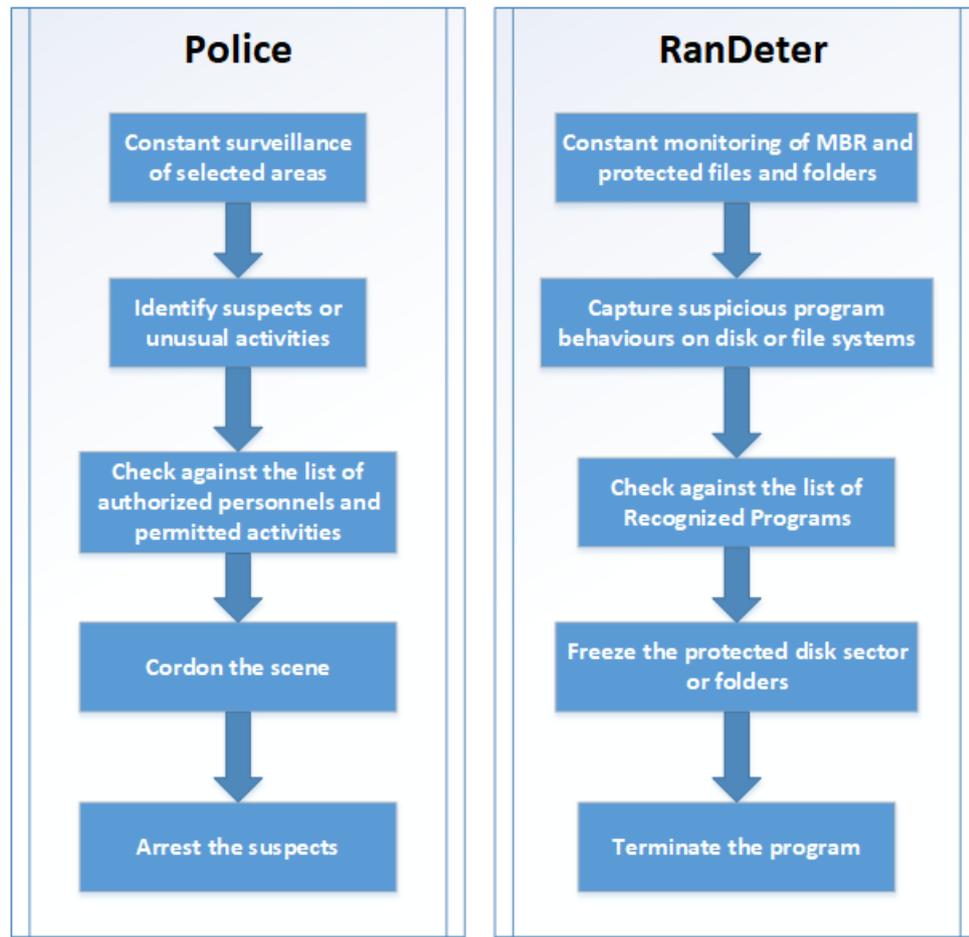


Figure 12 - the Newly Proposed Patrol-and-Defend Approach against Ransomware

- 1) Regular Monitoring: RANDETER drivers are loaded into Windows OS kernel at boot time. RANDETER shell program (FILE PATROL shell) is run at start-up in the administrative mode. Only directories like “Documents”, “Pictures” and other user-specified folders are considered for monitoring.
- 2) Suspicious Behaviors: the definition of suspicious behaviors can be arbitrary, but would include usage patterns unlikely performed by a non-robot human user. Such suspicious behaviors can include accessing too many different types of user files by one program, performing diverse types of file system activities alternately, and modifying many files within short durations.
- 3) Enhanced Application Whitelist: An enhanced application whitelist of programs with their expected behaviors is used. It is normal for Microsoft® Word to create and modify *.docx files, but not *.mp3 files. It is normal for 7-Zip to create and delete files, but it does not usually rename files or modify file contents.

- 4) **Damage Control:** if files in the directory protected by RANDETER may be under ransomware attack, RANDETER can freeze the directory to deny all access by rejecting all I/O requests of the suspicious program, to prevent further file damage by ransomware.
- 5) **Risk Elimination:** the suspicious program will be killed by RANDETER, and the access to the protected directory would resume afterwards.

5.4 Relevance to Australian Signals Directorate Cybersecurity Strategies

The design principles of RANDETER also draws inspiration from the “Strategies to Mitigate Cyber Security Incidents” proposed by Australian Signals Directorate (Australian Signals Directorate, 2017).

Application whitelisting of approved/trusted programs can help to prevent unwanted or unauthorized execution of malicious programs. RANDETER implements an enhanced version of application whitelisting that not only checks the application name, but whether the behaviors of the application at the file system level are expected. This approach may draw potential user resistance and require ongoing maintenance of an accurate and updated whitelist, the relative security effectiveness is high and recommended (Australian Signals Directorate, 2017).

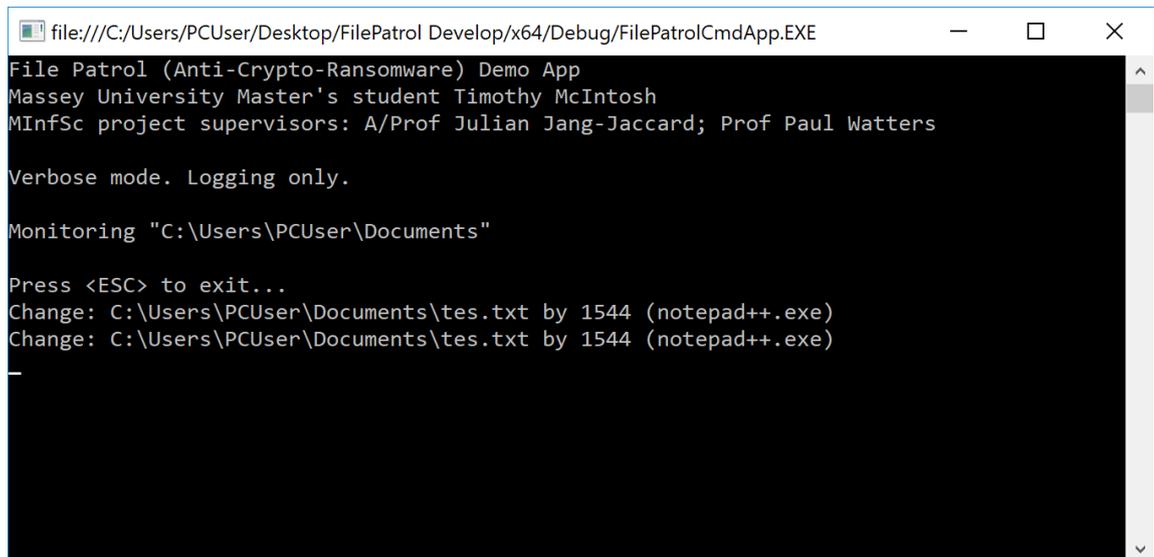
Although application whitelisting of known benign applications is a candidate solution for anti-crypto-ransomware solutions, it must be able to address several issues: Ransomware can inject themselves into whitelisted applications, or use whitelisted interpreters like wscript or command prompt, or masquerade as whitelisted applications (ESET 2017). All those issues are dealt with by RANDETER implementations.

Operating system hardening can prevent anomalous programs from abusing operating system vulnerabilities. RANDETER denies any modification of MBR, which can be modified and replaced with a malicious bootloader by simple programs with Administrative privileges on unprotected systems. This approach requires low ongoing maintenance and has “very good” security effectiveness (Australian Signals Directorate, 2017).

5.5 File System Events

An *event* on Windows operating system is defined by Microsoft to be any significant occurrence in a program or in the operating system that requires user notification or log entry (Troelsen, & Japikse, 2017). A *file event* is the occurrence of an operation that is performed to the entity of a file or folder at the file system level, which can include *Create*, *Delete*, *Change* (also known as *Modify*), *Move*, *Close*, *Rename*, and *Undelete* (Troelsen, & Japikse, 2017). Of the file events, *Create*, *Delete*, *Change*, *Close* and *Undelete* are considered unary events, because they only involve one full file name including the path; *Move* and *Rename* are binary events that involve two different full file names including the path.

User manipulations of files via file managers or application writing data into disks would generate different combinations of basic file events. When a user cuts and pastes a file in Windows File Manager, it creates a *Move* event. When a user copies a folder containing several files to a different location, it creates one *Create* event for the folder and for each file moved. When a user modifies a file and saves it in Notepad++, the “Save” operation generates a *Change* event (figure 13).



```

file:///C:/Users/PCUser/Desktop/FilePatrol Develop/x64/Debug/FilePatrolCmdApp.EXE
File Patrol (Anti-Crypto-Ransomware) Demo App
Massey University Master's student Timothy McIntosh
MInfSc project supervisors: A/Prof Julian Jang-Jaccard; Prof Paul Watters

Verbose mode. Logging only.

Monitoring "C:\Users\PCUser\Documents"

Press <ESC> to exit...
Change: C:\Users\PCUser\Documents\tes.txt by 1544 (notepad++.exe)
Change: C:\Users\PCUser\Documents\tes.txt by 1544 (notepad++.exe)

```

Figure 13 - the Change Event Generated by User Saving a File in Notepad++

When the user modifies and saves a “docx” file using Microsoft Word, the following series of file events occur (figure 14):

- 1) When the user opens the document (Basic Resume.docx) in Microsoft Word, Winword.exe *Creates* an initial temporary file (~\$sic Resume.docx), maybe for document recovery purposes.
- 2) When the user modifies and saves the document, Microsoft Word
 - i. *Creates* a first temporary file (~WRD0000) with the content changes
 - ii. *Renames* the original “Basic Resume.docx” into a second temporary file (~WRD0001)
 - iii. *Renames* the first temporary file (~WRD0000) with the content changes into “Basic Resume.docx”
 - iv. *Deletes* the second temporary file (~WRD0001)
- 3) When the user closes Microsoft Word, it *Deletes* the initial temporary file (~\$sic Resume.docx)

```

file:///C:/Users/PCUser/Desktop/FilePatrol Develop/x64/Debug/FilePatrolCmdApp.EXE
File Patrol (Anti-Crypto-Ransomware) Demo App
Massey University Master's student Timothy McIntosh
MInfSc project supervisors: A/Prof Julian Jang-Jaccard; Prof Paul Watters

Verbose mode. Logging only.

Monitoring "C:\Users\PCUser\Documents"

Press <ESC> to exit...
Create: C:\Users\PCUser\Documents\~$sic Resume.docx by 6528 (WINWORD.EXE)
Create: C:\Users\PCUser\Documents\~WRD0000.tmp by 6528 (WINWORD.EXE)
Rename: C:\Users\PCUser\Documents\Basic Resume.docx -> C:\Users\PCUser\Documents\~WRL0001.tmp by 6528 (WINWORD.EXE)
Rename: C:\Users\PCUser\Documents\~WRD0000.tmp -> C:\Users\PCUser\Documents\Basic Resume.docx by 6528 (WINWORD.EXE)
Delete: C:\Users\PCUser\Documents\~WRL0001.tmp by 6528 (WINWORD.EXE)
Delete: C:\Users\PCUser\Documents\~$sic Resume.docx by 6528 (WINWORD.EXE)

```

Figure 14 - File Events Generated by User Modifying a DOCX File using Microsoft Word

At the I/O stack level, each file system event is caused by one or several I/O requests with different parameters (table 2). Each I/O request contains a major function code

(named in the form of IRP_MJ_*XXX), which tells the driver what I/O operation should perform to satisfy the I/O request (Microsoft Hardware Dev Center, 2017a).

Table 2 - Definition of File System Events and Underlying IRP Operations

Event Type	Code	Definition	IRP Major Functions	Other conditions
Unknown	0	A file event that does not belong to any of the others	-	-
Create	1	A new file is created.	IRP_MJ_CREATE	IO Status is FILE_CREATED
Delete	2	A file has been removed from its current folder.	IRP_MJ_SET_INFORMATION	FileInformationClass is FileDispositionInformation
Change	3	The content of the file has been modified.	IRP_MJ_CREATE	IO Status is FILE_OVERWRITTEN
			IRP_MJ_WRITE	FO_FILE_MODIFIED
Move	4	The file has been moved to a different location that is not the Recycle Bin.	IRP_MJ_SET_INFORMATION	FileInformationClass is FileRenameInformation Different directory names but same file name
Close	5	The file handle has been closed and the resource is released.	IRP_MJ_CLOSE	-
Rename	6	The name of the file is changed, but remains in the same folder.	IRP_MJ_SET_INFORMATION	FileInformationClass is FileRenameInformation Same directory name but different file names
Undelete	7	The file has been restored from the Recycle Bin to its original location before deletion	IRP_MJ_SET_INFORMATION	FileInformationClass is FileRenameInformation Same file name. Source directory is the Recycle Bin.

5.6 Recognized Processes and Multi-Tier Security Rules

A key original contribution of RANDETER is the introduction of Recognized Processes and Multi-Tier Security Rules; it is a method of enhanced application whitelisting that will improve the concept of “Controlled Folder Access” and prevent “masquerading attacks”. RANDETER is implemented as a custom classifier that would deter crypto-ransomware activities, by detecting and rejecting non-conforming disk or file system activities.

Recognized Processes are processes which have clearly defined expectations of how they shall behave at the file system level. Only processes that are listed as the recognized processes by RANDETER can modify files in the protected directory.

In the current implementation of RANDETER, processes which attempt to modify files in protected directories are classified into four categories: Windows system modules, utilities, editors or unclassified.

Table 3 - Classification of Types of Recognized Processes

Windows System Modules
Programs that are part of the Microsoft Windows operating system and carry out core functionalities.
Examples: explorer.exe; svchost.exe; rundll32.exe; PowerShell.exe
Utilities
Programs that are developed to perform system maintenance tasks or file management.
Examples: 7-Zip, CCleaner, Defragment, DiskPart
Editors
Programs that are only used to view or edit specific file types.
Examples: Microsoft Word, Adobe Reader, Notepad++
Unclassified

Programs that are not in the known list of recognized processes, and must be manually evaluated and added to the list by system administrators if applicable. If using the default RANDETER setting, unclassified programs cannot modify files in protected directories.

Examples: software recently downloaded and installed by the users, but have not been added to the list of Recognized Processes

Multi-Tier Security Rules govern what file types the processes can operate on (file type rule), what file operations they can perform (operation rule) and how frequently they can perform file operations (speed rule). Each Recognized Process must operate within a set of predefined rules; the rules are different for different processes. File Patrol employs the principle of “implicit deny”: any operations not permitted in the Multi-Tier Security Rules are considered violations and are denied.

Table 4 - an Example Security Rule of “Explorer.exe”, a Windows System Module

Program	Microsoft® Windows Explorer	
Description	File explorer and manager	
Classification	<i>Windows System Module</i>	
Recognized name	explorer.exe	
Security Rules		
File Type Rule	Operation Rule	Speed Rule
All permitted	Create, Rename	5 per second
	Delete, Move, Undelete	Unlimited only for the same type of operations; no more than 1 type of operation per second

Table 5 - an Example Security Rule of "svchost.exe", a Windows System Module

Program	Microsoft® Windows Service Host	
Description	Host Process for Windows Services	
Classification	<i>Windows System Module</i>	
Recognized name	svchost.exe	
Security Rules		
File Type Rule	Operation Rule	Speed Rule
None permitted	None permitted	(N/A)

Table 6 - an Example Security Rule of 7-Zip, a Utility

Program	7-Zip	
Description	Open source file compressor	
Classification	<i>Utility</i>	
Recognized name	7zFM.exe and 7zG.exe	
Security Rules		
File Type Rule	Operation Rule	Speed Rule
7z, XZ, BZIP2, GZIP, TAR, ZIP, WIM	Create, Delete, Move, Rename, Undelete	Unlimited only for the same type of operations
	Change	1 per second

All other file types	Create, Delete, Move, Rename, Undelete	Unlimited
	Change (prohibited)	(N/A)

Table 7 - an Example Security Rule of Microsoft Word, an Editor

Program	Microsoft® Word 2016		
Description	Document editor		
Classification	Editor		
Recognized name	Winword.exe		
Security Rules			
File Type Rule	Operation Rule	Speed Rule	Note
doc; dot; docx; docm; dotx; dotm; docb; rtf; txt; htm; html; mht; mhtml; odt; wps; xml	Create	1 per second	
	Rename	5 per second, and only from or to “tmp” files	
pdf; xps	Create	1 per second	Word can save the documents as PDF or XPS files, but cannot open them
png; jpg; jpeg; gif; tiff; bmp	Create	1 per second	“Save as Picture” right-click menu

The security rules for common programs (Windows modules, Microsoft Office suites and popular utilities) are determined based on the following sources of information:

- The role and designed functionality of each program, and their supported file types. Information can be obtained from software vendors.
- SHIELDIFS experiment results and datasets of benign and ransomware IRP logs, generously provided by Andrea Continella, primary author of Continella et al (2016)
- File system events logged by RANDETER FILE PATROL monitor in verbose mode, when experimenting with benign software and ransomware samples
- Artificial interpretation by the developer of RANDETER on what is possible by a human user and what is possible to be done by a bot.

Process- and Directory-centric models. A crypto-ransomware could perform its encryption across multiple processes or multiple threads, for lower accountability and higher efficiency (Continella et al, n.d.). As a result, RANDETER adopted two models. The process-centric model checks all IRP requests from each process, and group them by process. The directory-centric model checks IRP requests directed to the monitored directory. The aim to combine both models is to deter multi-process ransomware, while providing the fastest response possible.

Chapter 6. RANDETER TECHNICAL DETAILS

In this chapter, the technical details of RANDETER, including its implementation, are discussed.

6.1 The Architecture of RANDETER

The RANDETER Anti-Crypto-Ransomware System consists of two independent but compatible modules: PARTITION GUARD and FILE PATROL. PARTITION GUARD prohibits unauthorized write access to sector 0 of the boot disk, where MBR is usually stored. It works at the disk I/O level, checks all disk I/O requests and rejects “write” access to sector 0 of the disk. FILE PATROL works at the file system level, patrols specified directories and important file types (documents, photos), and terminates unrecognized access patterns to all files in the protected directory.

The way PARTITION GUARD and FILE PATROL integrate into and interact with the I/O call stack is illustrated in figure 15. The call stack on the left side describes the usual process of a data write request by an application. When a user-mode application needs to write data into the disk subsystem, it calls the API provided by Windows I/O Manager within the Windows operating system kernel. An I/O request is generated as a result, and passed down through various components within the call stack, until it reaches the disk subsystem. On the right side, the FILE PATROL minifilters and the PARTITION GUARD driver exist as optional windows operating system kernel drivers that can be installed into the kernel. An I/O request passing through the call stack would be intercepted by FILE PATROL and PARTITION GUARD, which would have opportunities of examining the requests and rejecting them.

FILE PATROL modules exist in the file system part of the IRP call stack, and interact with the Filter Manager, so it can have visibility on what files have been involved in the IRPs. PARTITION GUARD exists in the disk system part of the IRP call stack, because it needs to determine the raw location of destination on the physical disk, either as byte offset from byte 0, or as the first sector of the disk (sector 0).

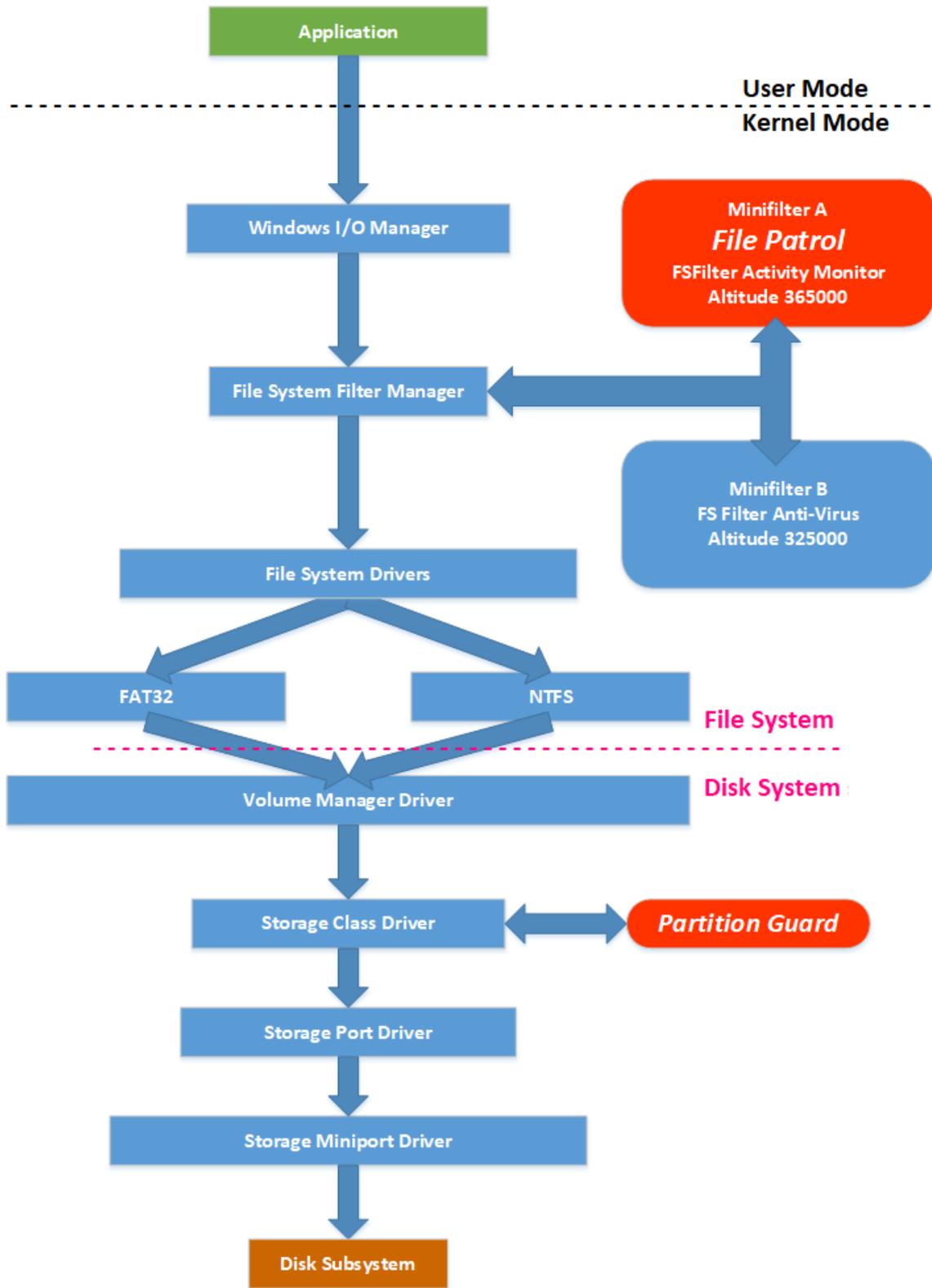


Figure 15 - General Architecture of RANDETER

6.2 PARTITION GUARD

PARTITION GUARD is a Disk Class Driver based on the Microsoft Disk Class Driver sample, and runs in kernel mode. A disk class driver, also known as storage class driver, uses the SCSI class/port interface to control a mass storage device; it handles I/O requests from higher in the storage stack, builds SCSI Request Blocks containing SCSI command descriptor blocks, and issues them to the next lower-level driver. The architectural details of PARTITION GUARD are illustrated in figure 16.

Upon the incoming of an I/O request, PARTITION GUARD checks the type of I/O command, and are only interested in two types of I/O commands that may be used to modify MBR:

- (1) The type of I/O command is **FileReadWrite**, the major function of the IRP is **IRP_MJ_WRITE** and the calculated **ByteOffset** of the command is less than 512
- (2) The type of I/O command is **DeviceControl**, the control code is **SCSI_PASS_THROUGH**, and the command is for sector 0.

The first 512 bytes of the boot disk, located in sector 0, contains the MBR for MBR/BIOS-based systems, or the protective MBR for GPT/UEFI-based systems. Since unsophisticated users usually do not need to modify MBR, PARTITION GUARD would block “write” methods from modifying MBR of the boot drive. All other disk access requests through PARTITION GUARD are permitted without further checking.

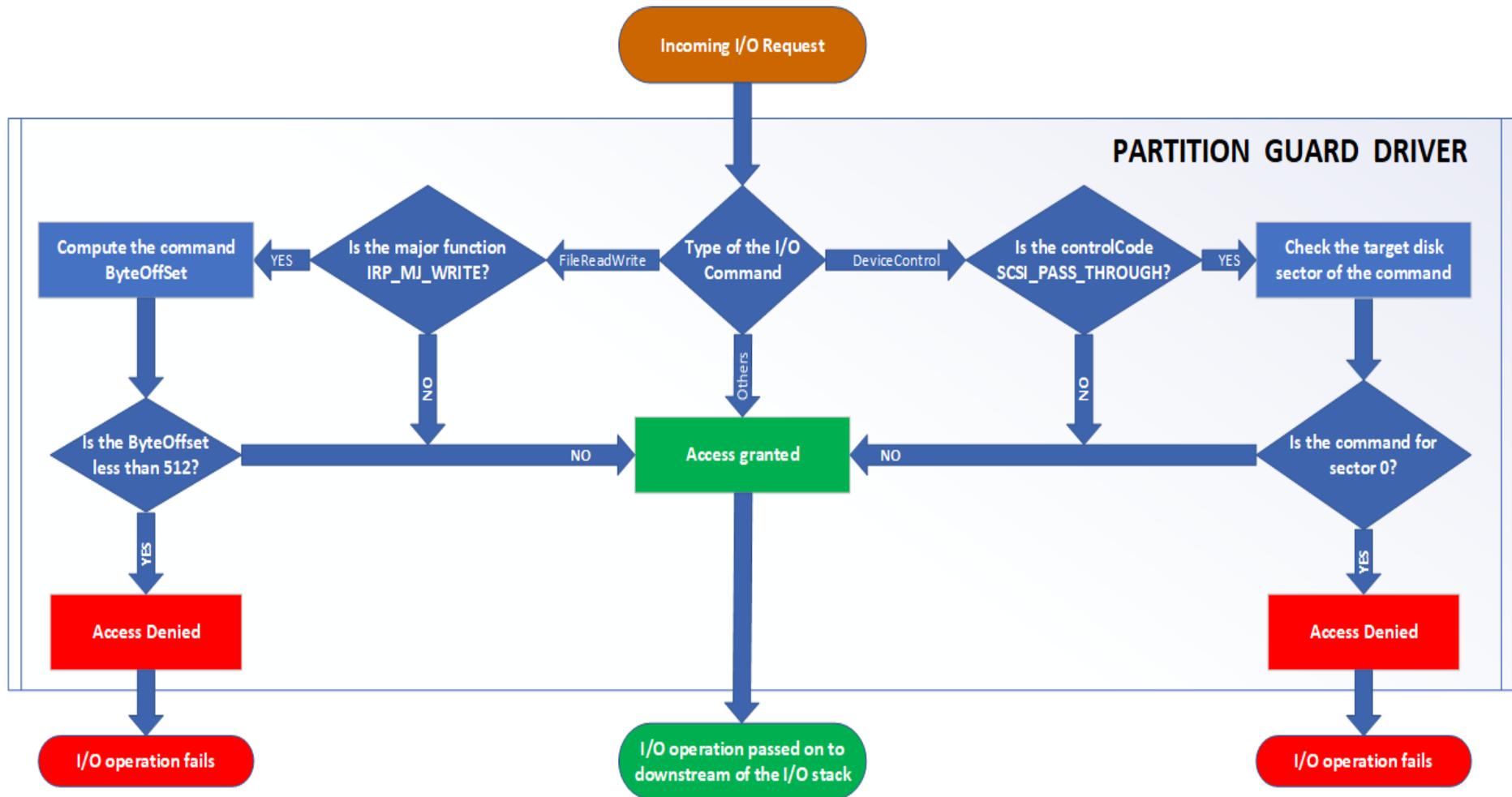


Figure 16 - PARTITION GUARD Technical Details

6.3 FILE PATROL

FILE PATROL comprises of a file system minifilter running in kernel mode, responsible for capturing file system activities and generating file system events, and a shell program running in user mode, responsible for handling file system events. FILE PATROL driver was implemented as the Minifilter and the Filter Manager Model, instead of the legacy file system driver, mainly because the new model has the following advantages over the legacy drivers (Microsoft Hardware Dev Center, 2017b):

- ✓ **Ability to only process necessary operations.** The minifilter can choose to respond to or ignore certain types of operations, and register pre-operation callback or post-operation callback functions.
- ✓ **More efficient usage of kernel stack.** The Filter Manager is optimized to reduce the complexity of kernel call stacks, which would reduce the performance impact of File Patrol on the system.
- ✓ **Better support for multiple Windows platforms.** The minifilters can be installed on any Windows platforms that support the Filter Manager.

The architectural details of FILE PATROL are illustrated in figure 17. Upon system startup, the minifilters driver receives information from the shell program on which directory has been selected by the user to be protected by FILE PATROL. When the Windows I/O Manager passed on I/O requests, the Filter Manager:

- intercepts the IRP requests,
- sends the IRP requests to the mounted File System Minifilters for evaluations.
 - If more than one minifilters, the minifilters with a higher “altitude” values gets the first chance to evaluate the file event.
- and expects an I/O permission flag value returned for each single file event from the minifilters on whether this file event should proceed further down along the call stack.

The File System Minifilter checks whether the file event involves files in the directory selected by the user to be monitored, and whether the access to that directory is permitted. The minifilter will then set the I/O permission flag as “permitted” or “denied” accordingly, and inform the Filter Manager of the I/O permission flag. If a file event does not involve any files in the directory monitored, the file event will be ignored by the

minifilters and permitted to proceed. The Filter Manager will then permit or deny the I/O operation based on the I/O permission flag. If the operation is permitted, the Filter Manager will pass the operation further down the I/O call stack.

Upon completion of a file system event analysis, the minifilters outputs the log to the shell program in user mode. The shell program then checks whether the process which initiated the file event is a Recognized Process, and whether the file event complies with the Multi-Tier Security Rules. If any violations are found, the shell program will send a command to the minifilters to lock down the protected directory and decline all further access to the directory. The shell program will try to terminate the offending process that created the file event of violation, and upon completion, instruct the minifilters to unlock the directory.

6.3.1 The File System Minifilter

The file system minifilter is based on the open-source Microsoft® implementation of MiniSpy infilter, with few modifications. It is a Windows Kernel Driver, written in C and installed into Windows kernel at the file system level. It is designed to intercept I/O Request Packets (IRP) that have been passing itself, and interpret different combinations of IRP operations into eight types of file operations, and make the information available through the specified port “\\FilePatrolPort”.

A file minifilter driver does not act directly on the Windows file system; instead, it registers with the filter manager for the I/O operations on the file system and gets loaded. There can be several minifilters; the order in which they are loaded and attached is the *altitude*. The filter manager would use the altitudes of different minifilters to call the minifilter drivers to handle I/O requests.

Each file system event log entry captured by FILE PATROL contains the essential information:

- Event Type: one of the eight types of file events.
- Destination File Name: the full file name of the target file, including the path.
- Originating Time: the system time at which the file system event takes place.

- Previous Filename (only applicable to *Move* and *Rename*): the full file name of the source file, including the path, before the file operation.
- PID: the process ID of the application program that initiated the file operation. The PID is a long integer value assigned by the operating system to identify the application process.

6.3.2 The Shell Program

The shell program, written in C#, is based on Microsoft.NET technology and runs on a Microsoft® Windows platform. It instructs the minifilters which directory to watch for, collects the information from the kernel driver via a specified port, interprets the information and presents it in a more human-readable manner.

The Shell program imports the native library "FltLib.dll", and communicates with the minifilter via the designated port "\\FilePatrolPort". Once the communication is established, the Shell program sends packages of commands in inbound buffers, and receives the response from the minifilter in outbound buffers.

The Shell program loads and applies the local settings of security rules, and receives security rule updates from the cloud server to apply locally. Shell program would combine the local settings with cloud settings to form a single set of rules, and check file system activities against the rules. When the rules are clearly violated, or the overall risk score is surpassed, further file system I/O requests by the same process would be declined. When applicable, suspicious programs would also be submitted by the Shell to the central cloud server for analysis.

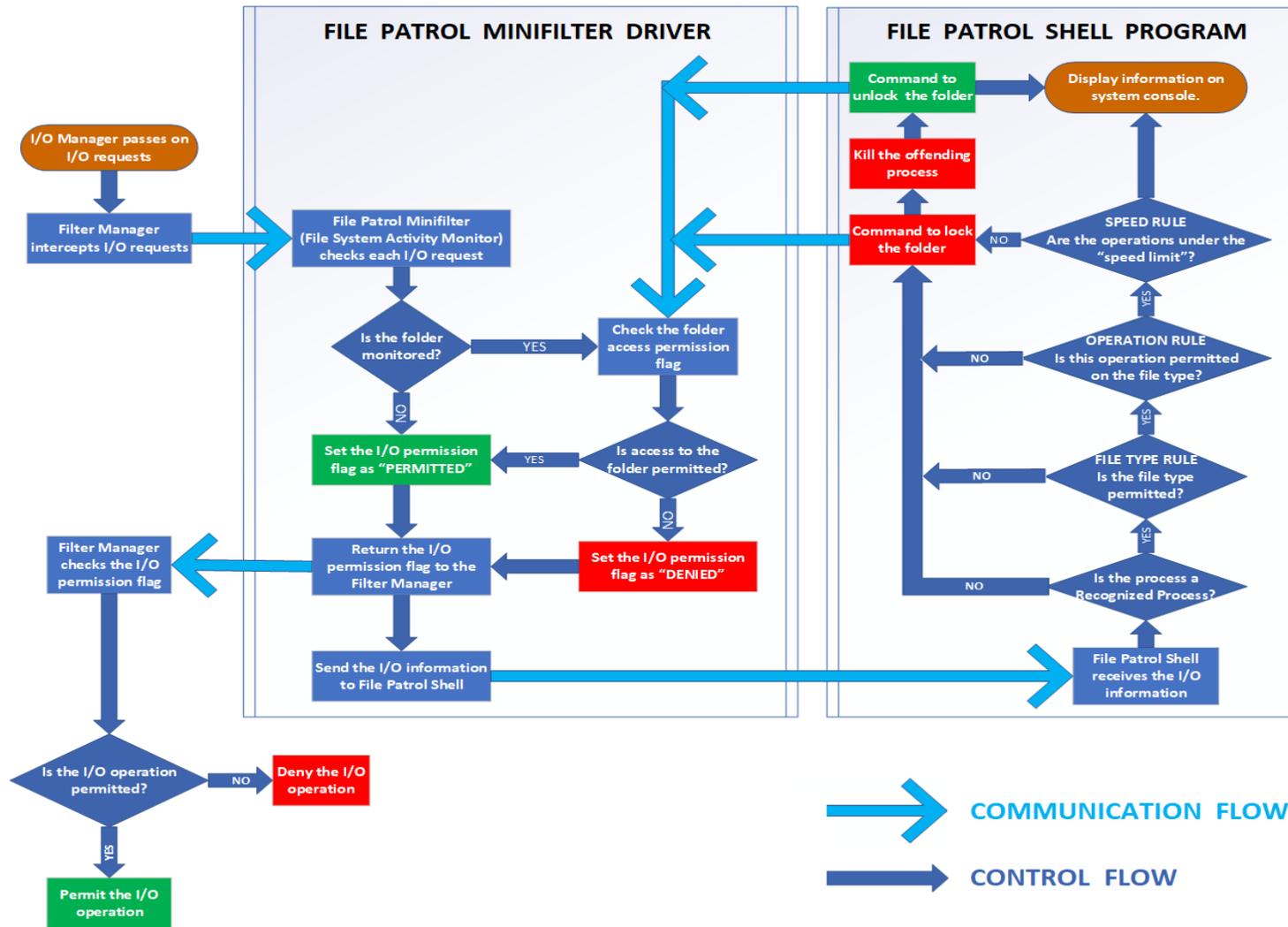


Figure 17 - FILE PATROL Technical Details

Chapter 7. EVALUATION

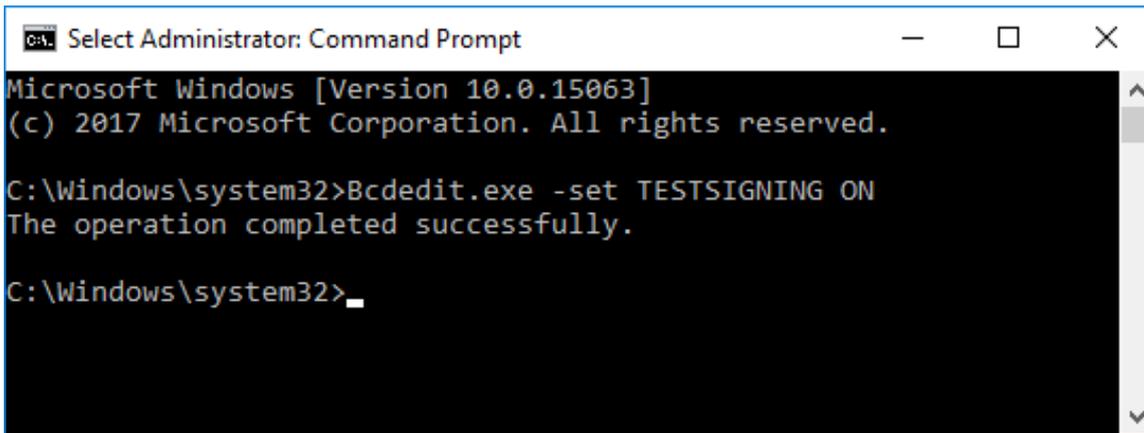
The prototype of RANDETER is developed based on the Microsoft Windows Kernel-Mode Driver Architecture, without modifying any operating system modules or underlying file system semantics. It supports Windows NT-based operating systems.

7.1 Experiment Setup

In our experiment, a Hyper-V virtual machine was set up with 6GB of RAM, which is the average RAM size of PCs in 2016 (Schuknecht, 2016). The virtual machine was created using “Generation 1” profile, to allow creation of MBR disks using the standard BIOS partition table, because “Generation 1” virtual machines boot in BIOS/MBR mode without SecureBoot (Microsoft Windows IT Pro Center, 2016); the condition required to test *Ransom.Petya* infections of MBR. Windows 10 Pro (x64, build 1703) was installed with Microsoft .Net Framework 4.5.2 and Microsoft Visual C++ Redistributable for Visual Studio 2015. The operating system was clean installed using the Microsoft Windows 10 ISO image. After installation, the Windows Defender, Windows Update services and Windows Firewall were disabled, so they would not block known ransomware from executing. Google Chrome web browser was installed, but the “Protect you and your device from dangerous sites” option was switched off, so ransomware samples could be downloaded.

Because both PARTITION GUARD and FILE PATROL drivers were developed using self-signed certificates, the testing Windows 10 system must be configured as “Test Mode”. The TESTSIGNING boot configuration option can be enabled (figure 18), by running the following command in Command Line as a System Administrator:

```
Bcdedit.exe -set TESTSIGNING ON
```



```

C:\Windows\system32>bcdedit.exe -set TESTSIGNING ON
The operation completed successfully.

C:\Windows\system32>

```

Figure 18 - Command Line Command to Switch on Test Mode in Windows 10

After reboot, the bottom right corner of the desktop should display the watermark “Test Mode” with additional operating system build information (figure 19).

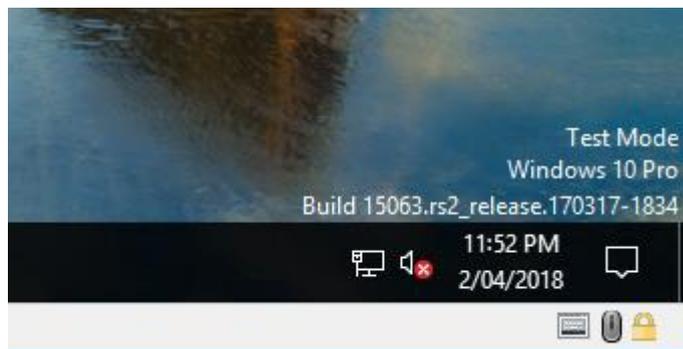


Figure 19 - the "Test Mode" Watermark on Windows 10 Desktop

Two checkpoints were created on the virtual machine, one before the installation of RANDETER modules (first checkpoint), and the other one after the installation and execution of RANDETER (second checkpoint). A checkpoint is a snapshot of the state of the virtual hard disk at the time the checkpoint is taken, so it can be restored later. When a checkpoint is restored, any changes to the virtual machine after the creation of the checkpoint is discarded and lost. Each ransomware sample was run on the virtual machine at the first checkpoint without RANDETER, to demonstrate that the ransomware sample was active and could attack the files. Then the virtual machine was restored with the second checkpoint with RANDETER running. The ransomware sample was run again, to evaluate whether RANDETER was able to deter the attack. After that, the virtual machine was restored to the first checkpoint for the next sample test.

7.2 Selection and Collection of Ransomware Samples

The prototype of RANDETER has been tested using live samples of selected well-known crypto-ransomware that have caused damages worldwide, and crypto-ransomware that were discussed in other studies. A ransomware attack was considered to have taken place in this experiment, when there was at least one file encrypted or damaged by ransomware without the installation and presence of RANDETER. Ransomware that were not crypto-ransomware or did not attack the file systems were excluded from this study. According to Symantec™ Internet Security Threat Report 2017, WannaCry and Petya are two of the most damaging ransomware in 2017; their outbreak and evolvement caught many unprepared individuals and organizations by surprise (Symantec, 2017).

Symantec™ Threat Explorer⁴ has a list of up-to-date information on the ransomware threats detected by them; figure 20 is a screenshot of the website.

⁴ <https://www.symantec.com/security-center/a-z/R>

The screenshot shows the Symantec Threat Explorer website. The page title is "A-Z Listing of Threats & Risks". Below the title, there is a navigation menu with "A-Z" selected. The main content area features a dark blue header with the title "A-Z Listing of Threats & Risks". Below this, a paragraph states: "The Threat Explorer is a comprehensive resource for daily, accurate and up-to-date information on the latest threats, risks and vulnerabilities." A navigation bar includes links for "A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|0-9 Special Characters". Below the navigation bar, the text "A-Z Listing of Threats & Risks (335)" is displayed, along with a "SHOW 10 ENTRIES" dropdown and a feed icon. A table with three columns: "Name", "Type", and "Discovered" is shown. The table contains the following data:

Name	Type	Discovered
Ransom.Zenis	Trojan	03/27/2018
Ransom.GandCrablg1	Trojan	03/13/2018
Ransom.HermesIgen1	Trojan	03/13/2018
Ransom.Rapid	Trojan	03/12/2018

Figure 20 - Website of Symantec™ Threat Explorer

The ransomware samples were downloaded from VirusShare.com website (registration required), using the name of the ransomware as the search keyword. Figure 21 illustrate the screen shot obtained as a result of searching for “WannaCry”. Of the search results, only those with the file type as “executable for MS Windows” were downloaded.

The screenshot shows the VirusShare.com website interface. The browser address bar displays 'https://virusshare.com/search.4n6'. The page title is 'VirusShare.com - Because Sharing is Caring'. The account 'Tim McIntosh' is logged in. Navigation links include Home, About, Hashes, Torrents, Errata, Research, and Support the Project. The main content area displays 'Displaying latest 20 results for string search "WannaCry" from the latest 2.5M malware detections.'

	MD5	9d3c57bdc6649b6640d97212f646f42a
	SHA1	914e398d393d570363574835689403ad360ce17b
	SHA256	94cce9146c9a203b351105cfa453166e7ef7548116e0756d9ff314f55201fd55
SSDeep	3072:Rmrhd5U1eigWcR+uiUg6p4FLIG4tL8z+mmCeHFZjoHEo3m:REd5+IZiZhLIG4AimmCo	
Size	245,760 bytes	
File Type	PE32 executable (GUI) Intel 80386, for MS Windows	
Detections	Ad-Aware = Trojan.Ransom.WannaCryptor.L AegisLab = Troj.Ransom.W32.Wanna.toN1 AhnLab-V3 = Trojan/Win32.WannaCryptor.R200589 ALYac = Trojan.Ransom.WannaCryptor Arcabit = Trojan.Ransom.WannaCryptor.L Avast = Win32:WanaCry-A [Trj] AVG = Win32:WanaCry-A [Trj] Avira = TR/FileCoder.724645 AVware = Trojan.Win32.Generic!BT BitDefender = Trojan.Ransom.WannaCryptor.L CAT-QuickHeal = Ransom.WannaCry.S962568 ClamAV = Win.Ransomware.WannaCry-6313787-0 CrowdStrike = malicious_confidence_100% (W) Cybereason = malicious.dc6649 Cyren = W32/Trojan.FMLA-6191 DrWeb = Trojan.Encoder.11432 Emsisoft = Trojan.Ransom.WannaCryptor.L (B) Endgame = malicious (high confidence) ESET-NOD32 = Win32/Filecoder.WannaCryptor.D F-Prot = W32/WannaCrypt.A Fortinet = W32/WannaCryptor.D!tr.ransom GData = Win32.Trojan-Ransom.WannaCry.E Ikarus = Trojan-Ransom.WannaCry Jiangmin = Trojan.WannaCry.a K7AntiVirus = Trojan (0001140e1) K7GW = Trojan (0001140e1) Kaspersky = Trojan-Ransom.Win32.Wanna.c Malwarebytes = Ransom.WannaCrypt MAX = malware (ai score=95) McAfee-GW-Edition = BehavesLike.Win32.RansomWannaCry.dh McAfee = Ransom-WannaCry!9D3C57BDC664 Microsoft = Ransom/Win32/WannaCrypt.C MicroWorld-eScan = Trojan.Ransom.WannaCryptor.L	

Figure 21 - A Screenshot of VirusShare.com Website

Once the ransomware sample was downloaded, if the file extension was not “.exe”, it would be renamed into the “.exe” file. The file would then be uploaded to VirusTotal.com website for cross-checking of its classification. The cross-checking was necessary, because sometimes different security software vendors classify ransomware different, or some security software cannot yet detect the sample and report it as “clean”. The sample was used for testing either if Symantec™ reported it as the ransomware type of interest, or if at least three security software vendors classified it as the ransomware type of interest.

53 engines detected this file

SHA-256 94cce9146c9a203b351105cfa453166e7ef7548116e0756d9ff314f55201f...

File name malware9.bin

File size 240 KB

Last analysis 2018-03-11 03:11:00 UTC

53 / 66

Detection	Details	Behavior	Community
Ad-Aware			Trojan.Ransom.WannaCryptor.L
AegisLab			Troj.Ransom.W32.Wanna.toNI
AhnLab-V3			Trojan/Win32.WannaCryptor.R200589
ALYac			Trojan.Ransom.WannaCryptor
Arcabit			Trojan.Ransom.WannaCryptor.L
Avast			Win32:WanaCry-A [Trj]
AVG			Win32:WanaCry-A [Trj]
Avira			TR/FileCoder.724645
AVware			Trojan.Win32.Generic!BT
BitDefender			Trojan.Ransom.WannaCryptor.L

Figure 22 - A Screenshot of VirusTotal.com Website

If the sample was correctly identified by VirusTotal.com, it would be executed on the virtual machine at the first checkpoint without RANDETER, and left for at least 15 minutes. If no file damage in “Documents” was found, the ransomware sample was considered inactive and discarded; the virtual machine would be restored to the first checkpoint to test the next sample. Some variants of ransomware samples that appeared no longer active were excluded in the testing, possibly because the shutdown of their command-and-control centers, or they could detect the presence of a virtual machine. The inactive ransomware samples were discarded to avoid false negative results, which would not be useful in proving the effectiveness of RanDeter.

7.3 Detection Results

RanDeter was able to deter the attacks of all ransomware samples tested, achieving a true positive rate of 100% on the 73 unique samples from 7 ransomware families (table 8). The detection results of RanDeter against several groups of crypto-ransomware were summarized in table 8, and were compared with three existing anti-ransomware implementations that were already published and were suitable as end-user products: REDEMPTION by Kharraz & Kirda (2017), CRYPTOLOCK by Scaife et al (2016), and SHIELDIFS by Continella et al (2016). The implementation UNVEIL by Kharraz et al (2016) was excluded from the comparison, because it used virtualized desktop environment and was not a suitable end-user product.

The comparison considered the number of different ransomware samples tested (subject to availability of live ransomware samples at the time of testing), the number of file damage (also known as file loss), defined as files encrypted by ransomware and could not be recovered. File damage however, did not apply to SHIELDIFS, which had inbuilt file recovery mechanism to rollback ransomware damages. The selection of ransomware families for comparison in the table was based on four factors:

- Are there still enough live samples to test with RANDETER?
- What ransomware families were tested by other studies?
- What are the most notorious and damaging ransomware at the time of writing?
- What are some of the latest ransomware samples released to the wild by criminals?

Ransom.CryptoLocker was tested by RANDETER and all other three implementations. Only RANDETER and SHIELDIFS were able to ensure no file damage caused by *Ransom.CryptoLocker*. RANDETER and REDEMPTION share the most number of same ransomware families tested, possibly due to the proximity in time of the two studies. RANDETER appeared to be able to achieve a lower number of file damage per sample on average than REDEMPTION, and was able to completely block the attacks of *Ransom.Petya*. RANDETER was also able to deter *Ransom.BadRabbit* and *Ransom.GandCrab*, developed by criminals after the completion of those three studies.

Table 8 - Detection Results and Comparison with Other Anti-Crypto-Ransomware Solutions

Family First Reported	Attacks MBR?	RANDETER Samples/File Damaged	REDEMPTION Samples/File Damaged	CRYPTOLOCK Samples/File Damaged	SHIELDFS Samples
BadRabbit 2017-10	√	4/0	-	-	-
Cerber 2017-05	×	10/3	30/6	-	-
CryptoLocker 2014-01	×	13/0	29/4	31/10	20
GandCrab 2018-03	×	21/0	-	-	-
Jigsaw 2016-04	×	6/5	12/4	-	-
Petya 2016-03	√	8/0	32/5	-	-
WannaCry 2017-05	×	11/0	7/5	-	-
Total Samples		73	110	31	20
Total Families Compared		7	5	1	1
File Damage Per Sample On Average		0.11	0.22	0.32	-

The experiment proved that RANDETER was effective against a wide range of ransomware attacks on the test system. It was able to detect various ransomware attacks, including *Ransom.GandCrab* that was new and still in the wild, with low rate of irrecoverable file damage.

7.4 Benchmarks of File System I/O Performance

The disk I/O and file system performance of RANDETER have been evaluated using the automatic and standard test profiles provided CrystalDiskMark, a well-known disk benchmark tool for Windows platforms. It was tested on solid state drive to eliminate the performance bottleneck of mechanical rotational hard drives which can be much slower. First, 50MB files were generated by CrystalDiskMark in the protected directory to test the throughput of sequential read and write (32 queues 1 thread), and 4K random read and write (32 queues 1 thread); the tests were run 5 times to obtain an average value. Then $5 \times 500\text{MB}$ files were generated to run the same test 5 times. The disk performance before and after the installation of RANDETER were compared to evaluate the performance overheads. The results are summarized in Table 9 and Figure 23. It appeared that there was some performance degradation on sequential read and write operations, but for 4K random read and write operations, the differences in disk read/write speeds were negligible. The experiments show that RANDETER performs well when handling heavy read and write requests, while imposing almost no system overheads within margins of error.

Table 9 - File System I/O Performance Benchmark with or without RANDETER

Block Size	Disk Operation	Original Disk Performance	Disk Performance with RANDETER installation	Performance Overhead (%)
50MB	Sequential read	1752.8 MB/s	1664.4 MB/s	5.0 %
	Sequential write	751.2 MB/s	726.8 MB/s	3.3%
	4K random read	284.5 MB/s	284.9 MB/s	-0.1% *
	4K random write	161.9 MB/s	161.0 MB/s	0.6%
500MB	Sequential read	1711.8 MB/s	1670.6 MB/s	2.4%
	Sequential write	786.8 MB/s	788.3 MB/s	-0.2% *
	4K random read	310.1 MB/s	311.7 MB/s	-0.5% *
	4K random write	164.5 MB/s	164.0 MB/s	0.3%

* The small negative performance overheads may be due to background system activities like file indexing, which are beyond the control of the benchmarking utility.

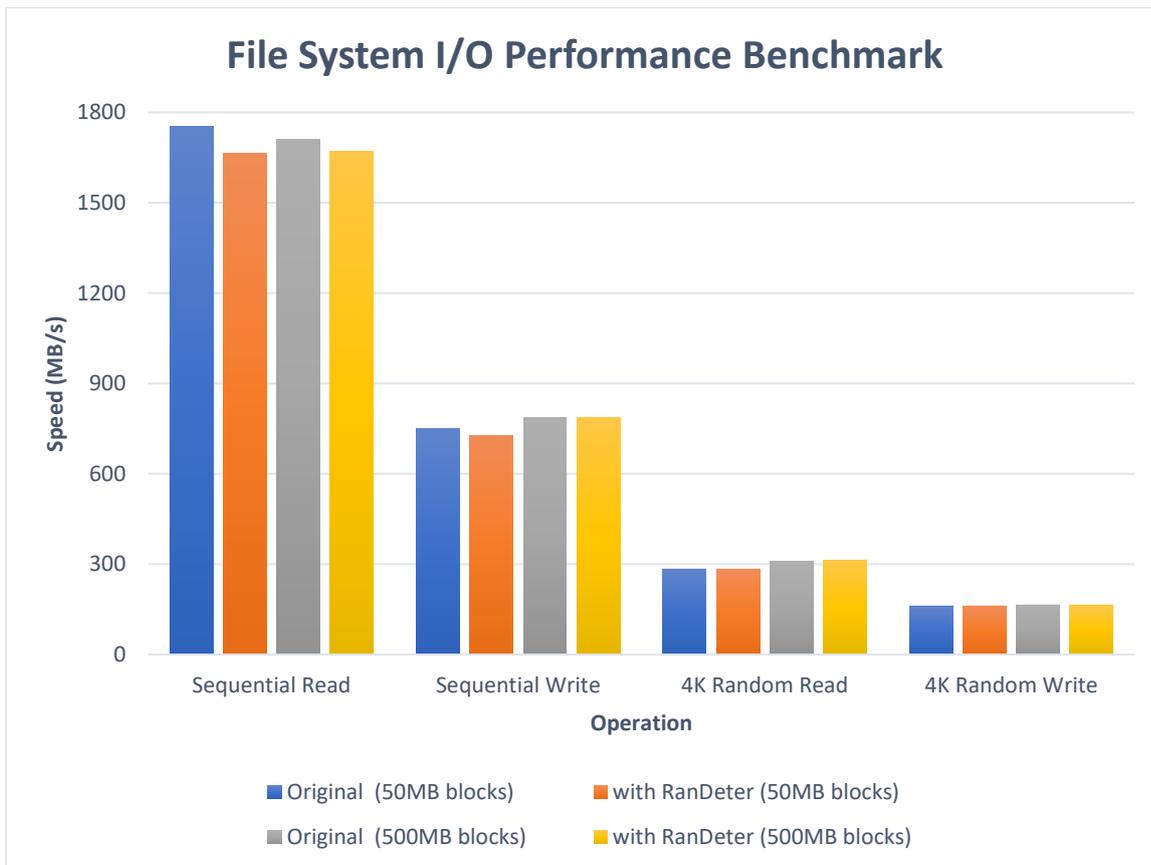


Figure 23 - File System I/O Performance Benchmark with or without RANDETER

7.5 Effectiveness against Ransomware Polymorphism

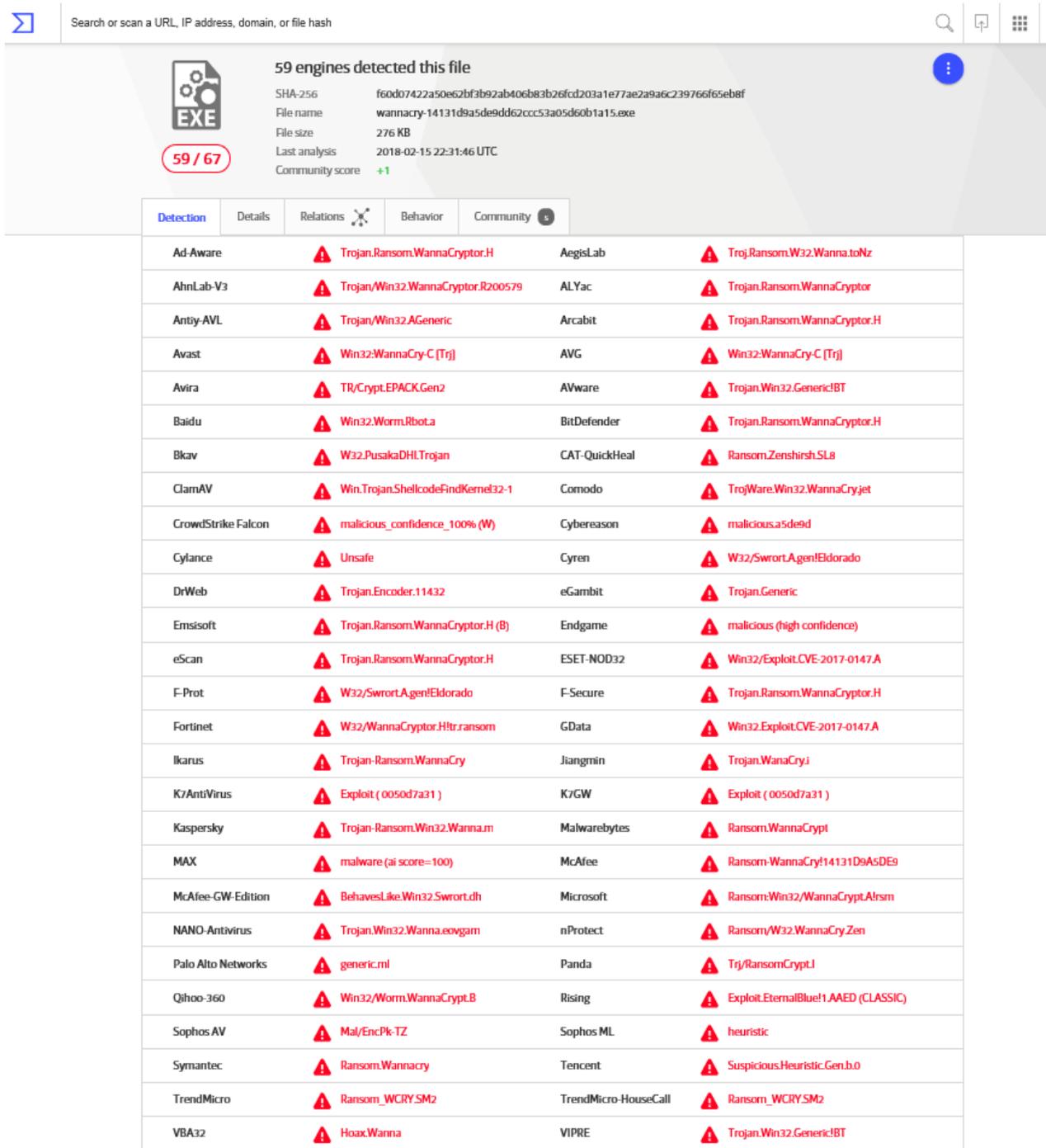
Ransomware Polymorphism is when ransomware developers constantly change the identifiable features of ransomware samples to evade detection by security software or security analyzers. Because signature matching is an important functionality of most anti-virus software, polymorphic techniques try to make analysis of virus harder, by constantly modifying the appearance of the code constantly, in such a way that no permanent common string remain among variants of the same malware that can be detected by any antivirus scanner engine.

A possible way to achieve polymorphism is to use an executable packer to compress and encrypt the code of the executable file. In this experiment, UPX⁵ (version 3.9.4 released on 12 May 2018), an open-source free executable packer, was used with the "--brute" parameter to try all available compression methods and filters. Below is an example of the *Ransom.WannaCry* sample used for analysis (SHA-256 f60d07422a50e62bf3b92ab406b83b26fcd203a1e77ae2a9a6c239766f65eb8f). This sample was uploaded to VirusTotal.com, and was correctly identified by 59 out of 67 (88%) antivirus engines as *Ransom.WannaCry* or malicious (figure 24). After packing it using UPX with the "--brute" parameter (figure 25), its SHA-256 value became bbe95b4a9c2e4e4ab8bf7991361b9d6e24026416889392f133cc9416b890b7c6. The repacked sample exhibited the same ransomware behavior on the test virtual machine, yet obtained a different SHA-256 value through repacking, because the binary opcode had been compressed and / or rearranged without changing its functionality or program flow (Martignoni, Christodorescu, & Jha, 2007). The repacked sample was uploaded to VirusTotal.com again, but only 40 out of 66 (61%) antivirus engines recognized the repacked sample as *Ransom.WannaCry* or somewhat malicious; ClamAV and McAfee reported it as clean (figure 26). The repacked sample was executed and exhibited the same ransom behavior as the original ransomware executable.

All *Ransom.WannaCry*, *Ransom.BadRabbit* and *Ransom.Petya* samples were repacked using UPX with different parameters, and used to challenge the defense by RANDETER. Each were treated as unknown ransomware samples, and were run on the

⁵ <https://upx.github.io/>

virtual machines with or without RANDETER. When no RANDETER protection was present, the repacked ransomware samples exhibited identical behaviors (selection of file targets and display of ransom messages), and caused the same damages as the original source ransomware. All ransomware attacks by the repacked executables were successfully deterred by RANDETER in the same manner: the attacks were thwarted at the same step of ransomware intrusion to the system as that of the source ransomware.



Search or scan a URL, IP address, domain, or file hash

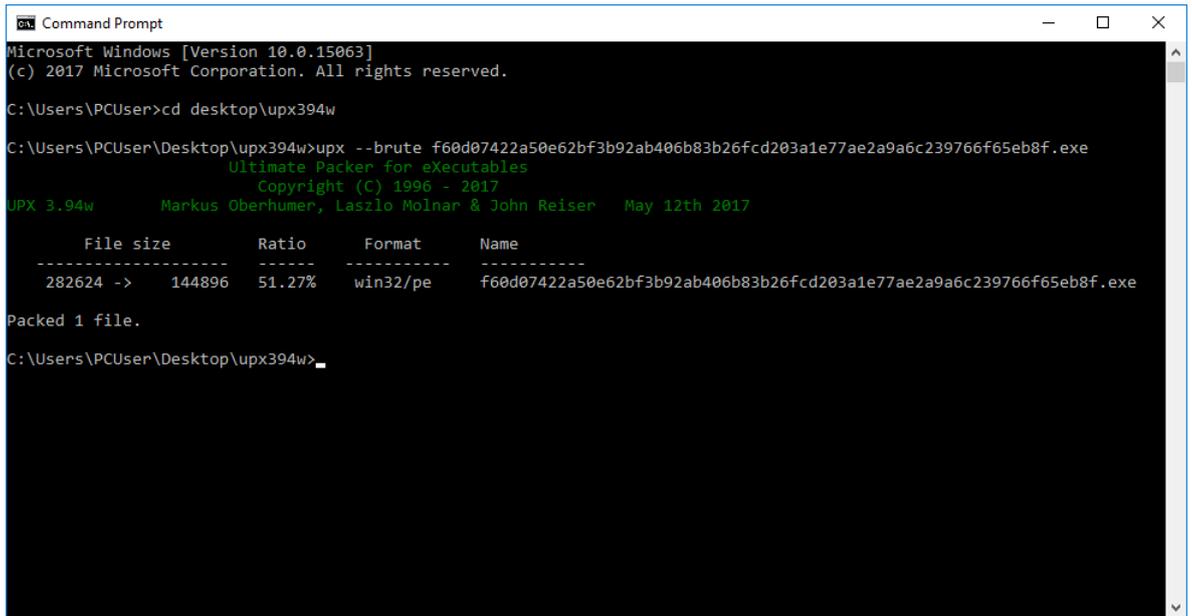
59 / 67

59 engines detected this file

SHA-256: f60d07422a50e62bf3b92ab406b8326fcd203a1e77ae2a9a6c239766f65eb8f
 File name: wannacry-14131d9a5de9dd62ccc53a05d60b1a15.exe
 File size: 276 KB
 Last analysis: 2018-02-15 22:31:46 UTC
 Community score: +1

Detection	Details	Relations	Behavior	Community
Ad-Aware	Trojan.Ransom.WannaCryptor.H	AegisLab	Troj.Ransom.W32.Wanna.toNz	
AhnLab-V3	Trojan/Win32.WannaCryptor.R200579	ALYac	Trojan.Ransom.WannaCryptor	
Antiy-AVL	Trojan/Win32.AGeneric	Arcabit	Trojan.Ransom.WannaCryptor.H	
Avast	Win32:WannaCry-C [Trj]	AVG	Win32:WannaCry-C [Trj]	
Avira	TR/Crypt.EPACK.Gen2	AVware	Trojan.Win32.Generic!BT	
Baidu	Win32.Worm.Rbot.a	BitDefender	Trojan.Ransom.WannaCryptor.H	
Bkav	W32.PusakaDHL.Trojan	CAT-QuickHeal	Ransom.Zenshirsh.SL8	
ClamAV	Win.Trojan.ShellcodeFindKernel32-1	Comodo	TrojWare.Win32.WannaCryjet	
CrowdStrike Falcon	malicious_confidence_100% (W)	Cybereason	malicious.a5de9d	
Cylance	Unsafe	Cyren	W32/Swrotr.A.gen!Eldorado	
DrWeb	Trojan.Encoder.11432	eGambit	Trojan.Generic	
Emsisoft	Trojan.Ransom.WannaCryptor.H (B)	Endgame	malicious (high confidence)	
eScan	Trojan.Ransom.WannaCryptor.H	ESET-NOD32	Win32/Exploit.CVE-2017-0147.A	
F-Prot	W32/Swrotr.A.gen!Eldorado	F-Secure	Trojan.Ransom.WannaCryptor.H	
Fortinet	W32/WannaCryptor.H!tr.ransom	GData	Win32.Exploit.CVE-2017-0147.A	
Ikarus	Trojan-Ransom.WannaCry	Jiangmin	Trojan.WannaCry.j	
K7AntiVirus	Exploit (0050d7a31)	K7GW	Exploit (0050d7a31)	
Kaspersky	Trojan-Ransom.Win32.Wanna.m	Malwarebytes	Ransom.WannaCrypt	
MAX	malware (ai score=100)	McAfee	Ransom-WannaCry!14131D9A5DE9	
McAfee-GW-Edition	BehavesLike.Win32.Swrotr.dh	Microsoft	Ransom:Win32/WannaCrypt!A!rsm	
NANO-Antivirus	Trojan.Win32.Wanna.eovgam	nProtect	Ransom/W32.WannaCry.Zen	
Palo Alto Networks	generic.ml	Panda	Trj/RansomCrypt.L	
Qihoo-360	Win32/Worm.WannaCrypt.B	Rising	Exploit.EternalBlue!1.AAED (CLASSIC)	
Sophos AV	Mal/EncPk.TZ	Sophos ML	heuristic	
Symantec	Ransom.Wannacry	Tencent	Suspicious.Heuristic.Gen.b.0	
TrendMicro	Ransom_WCRY.SM2	TrendMicro-HouseCall	Ransom_WCRY.SM2	
VBA32	Hoax.Wanna	VIPRE	Trojan.Win32.Generic!BT	

Figure 24 – on VirusTotal.com, 59 out of 67 AntiVirus Engines Recognized the Sample

as Ransom.WannaCry or Somewhat Malicious

```
Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\PCUser>cd desktop\upx394w

C:\Users\PCUser\Desktop\upx394w>upx --brute f60d07422a50e62bf3b92ab406b83b26fcd203a1e77ae2a9a6c239766f65eb8f.exe
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2017
UPX 3.94w Markus Oberhumer, Laszlo Molnar & John Reiser May 12th 2017

-----
File size      Ratio      Format      Name
-----
282624 -> 144896  51.27%    win32/pe    f60d07422a50e62bf3b92ab406b83b26fcd203a1e77ae2a9a6c239766f65eb8f.exe

Packed 1 file.

C:\Users\PCUser\Desktop\upx394w>
```

Figure 25 - Repacking the Ransom.WannaCry Sample Using UPX and the --brute Parameter

Search or scan a URL, IP address, domain, or file hash

40 engines detected this file

SHA-256: bbe95b4a9c2e4e4ab8bf7991361b9d6e24026416889392f133cc9416b890b7c6
 File name: f60d07422a50e2bf3b92ab406b83b26fcd203a1e77ae2a9a6c239766f65eb8f.exe
 File size: 141.5 KB
 Last analysis: 2018-05-04 11:13:30 UTC

40 / 66

Ad-Aware, Arcabit, AVG, Babable, BitDefender, Comodo, Cylance, eGambit, Endgame, ESET-NOD32, Fortinet, Ikarus, K7GW, MAX, Microsoft, nProtect, Rising, Sophos ML, TheHacker, Yandex, AegisLab, Antiy-AVL, AVware, ClamAV, Cyren, Jiangmin, Malwarebytes, ALYac, Avast, Avira, Baidu, CAT-QuickHeal, CrowdStrike Falcon, DrWeb, Emsisoft, eScan, F-Secure, GData, K7AntiVirus, Kaspersky, McAfee-GW-Edition, NANO-Antivirus, Qihoo-360, SentinelOne, Symantec, VBA32, ZoneAlarm, AhnLab-V3, Avast Mobile Security, Bkav, CMC, F-Prot, Kingsoft, McAfee

Ad-Aware	Generic.Malware.FWX.79ABB9EB	ALYac	Generic.Malware.FWX.79ABB9EB
Arcabit	Generic.Malware.FWX.79ABB9EB	Avast	Win32:WannaCry-C [Trj]
AVG	Win32:WannaCry-C [Trj]	Avira	TR/Downloader.Gen
Babable	Malware.HighConfidence	Baidu	Win32.Worm.Rbot.a
BitDefender	Generic.Malware.FWX.79ABB9EB	CAT-QuickHeal	Ransom.Zenshirsh.SL8
Comodo	TrojWare.Win32.WannaCry.jet	CrowdStrike Falcon	malicious_confidence_100% (D)
Cylance	Unsafe	DrWeb	Trojan.Encoder.11432
eGambit	Trojan.Generic	Emsisoft	Generic.Malware.FWX.79ABB9EB (B)
Endgame	malicious (moderate confidence)	eScan	Generic.Malware.FWX.79ABB9EB
ESET-NOD32	a variant of Win32/Exploit.CVE-2017-0147.A	F-Secure	Generic.Malware.FWX.79ABB9EB
Fortinet	W32/WannaCryptor.Hltr.ransom	GData	Win32.Exploit.CVE-2017-0147.A
Ikarus	Trojan-Ransom.WannaCry	K7AntiVirus	Exploit (0050d7a31)
K7GW	Exploit (0050d7a31)	Kaspersky	HEUR:Trojan.Win32.Generic
MAX	malware (ai score=84)	McAfee-GW-Edition	Behaves.Like.Win32.Fake.cc
Microsoft	Ransom:Win32/WannaCrypt.A!rsm	NANO-Antivirus	Virus.Win32.Gen-Crypt.ccncc
nProtect	Ransom/W32.WannaCry.Zen	Qihoo-360	HEUR/QVM18.1.EFB3.Malware.Gen
Rising	Trojan.Win32.WanaCryptH (CLASSIC)	SentinelOne	static engine - malicious
Sophos ML	heuristic	Symantec	Ransom.Wannacry
TheHacker	Possible_Worm32	VBA32	TrojanRansom.Wanna
Yandex	Trojan.Rosena.Gen.1	ZoneAlarm	HEUR:Trojan.Win32.Generic
AegisLab	Clean	AhnLab-V3	Clean
Antiy-AVL	Clean	Avast Mobile Security	Clean
AVware	Clean	Bkav	Clean
ClamAV	Clean	CMC	Clean
Cyren	Clean	F-Prot	Clean
Jiangmin	Clean	Kingsoft	Clean
Malwarebytes	Clean	McAfee	Clean

Figure 26 - on VirusTotal.com, only 40 out of 66 AntiVirus Engines Recognized the Repacked Sample by UPX as Ransom.WannaCry or Somewhat Malicious

Chapter 8. A CASE STUDY OF WANNACRY

In this chapter, *Ransom.WannaCry* is examined to demonstrate how its attacks could be deterred by FILE PATROL module of RANDETER.

8.1 WannaCry Introduction

WannaCry, also known as WannaCrypt, was first discovered on May 12, 2017. It spread by exploiting the NSA-leaked Microsoft Windows security loophole “EternalBlue”, and caused havoc in airports, banks, large companies, government departments, hospitals and universities worldwide (Symantec Security Center, 2017). Because it uses the strong RSA-2048 encryption, it is virtually non-decryptable by brute force methods (CERT-EU, 2017). According to Europol, the WannaCry ransomware campaign was unprecedented in scale. The indirect losses caused by WannaCry attacks would reach USD\$4 billion (CERT-EU, 2017). Below is a list of some big organizations confirmed to have been affected by *Ransom.WannaCry* (CERT-EU, 2017; Wikipedia, n.d.).

Table 10 - Organization Affected by WannaCry

Organization Type	Names
Corporates	Automobile Dacia (Romania), Boeing (USA), Deutsche Bahn (Germany), FedEx (USA), LATAM Airlines (Brazil, Chile), Honda (Japan), Hitachi (Japan), O2 (Germany), PetroChina (P. R. China), Portugal Telecom (Portugal), Renault (France), Russian Railways (Russia), Saudi Telecom Company (Saudi Arabia), Telefónica (Spain), Telenor Hungary (Hungary), Telkom (South Africa)
Government	Andhra Pradesh Police (India), Chinese Public Security Bureau (P. R. China), Ministry of Internal Affairs (Russia), Ministry of

	Foreign Affairs (Romania), São Paulo Court of Justice (Brazil), State Governments of India (India)
Hospitals	Dharmais Hospital (Indonesia), National Health Service (England, Scotland)
Universities	Aristotle University of Thessaloniki (Greece), Dalian Maritime University (P. R. China), Shandong University (P. R. China), University of Montreal (Canada)

8.2 WannaCry Attack in Details

The *Ransom.WannaCry* sample used for analysis is one variant that has a SHA-256 value of `ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa`. It is a password-protected self-extracting zip-format exe executable file (figure 27), and contains the following items:

- `\msg` folder has 28 localizations of ransom messages in RTF formats, to explain to victims in plain languages that their files are encrypted and how to pay the ransom.
- `b.wnry` is the bitmap with ransom message used to replace the victim's desktop wallpaper.
- `c.wncy` appears to contain website addresses used for Tor (anonymity network) communication.
- `r.wncy` is a Q&A file with 3 sets of frequently asked questions and their answers.
- `s.wncy` is the Tor client used to connect to the anonymity network.
- `t.wncy` has the default public and private keys used for ransom file encryption.
- `taskdl.exe` and `taskse.exe` appear to be involved in decryption of victim files.
- `u.wncy` is the duplicate copy of `@WannaDecryptor@.exe` file, the UI that displays the ransom message and verifies the ransom payment with the command-and-control center.

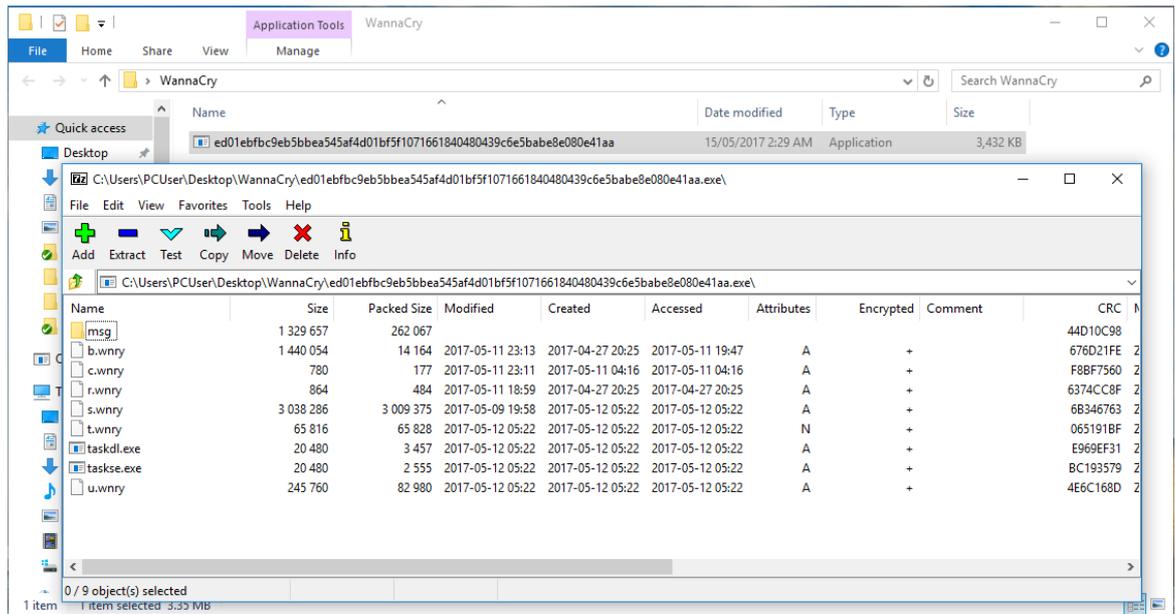


Figure 27 - the Zip Archive Contents of WannaCry Ransomware

FILE PATROL module was switched to “Verbose Mode” to only log the file system activities by a ransomware attack (figure 28). The *Ransom.WannaCry* sample was found to be modifying many user files and renaming them with the WNCRYT file extension name. Based on the verbose log, it appeared that the *Ransom.WannaCry* sample:

- Created an initial temporary file maybe used to log the progress
- Created a file with the same filename but with the extension of WNCRTY to store some information related to the original file
- Renamed the file extension from WNCRTY to WNCRY
- Modified (Changed) the original file with encrypted content
- Moved on to attack the next file

```

C:\Users\PCUser\Desktop\FilePatrol Develop\FilePatrolCmdApp\bin\x64\Release\FilePatrolCmdApp.exe
File Patrol (Anti-Crypto-Ransomware) Demo App
Massey University Master's student Timothy McIntosh
MInfSc project supervisors: A/Prof Julian Jang-Jaccard; Prof Paul Watters
Verbose Mode. Logging file system activities only.
Monitoring "C:\Users\PCUser\Documents"

Press <ESC> to exit...
Create: C:\Users\PCUser\Documents\~SDD1D8.tmp by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Delete: C:\Users\PCUser\Documents\~SDD1D8.tmp by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Create: C:\Users\PCUser\Documents\5mbg.pdf.WNCRYT by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Rename: C:\Users\PCUser\Documents\5mbg.pdf.WNCRYT -> C:\Users\PCUser\Documents\5mbg.pdf.WNCRY by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Change: C:\Users\PCUser\Documents\5mbg.pdf by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Create: C:\Users\PCUser\Documents\Anatomy of the Somatosensory System.pdf.WNCRYT by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Rename: C:\Users\PCUser\Documents\Anatomy of the Somatosensory System.pdf.WNCRYT -> C:\Users\PCUser\Documents\Anatomy of the Somatosensory System.pdf.WNCRY by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Change: C:\Users\PCUser\Documents\Anatomy of the Somatosensory System.pdf by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Create: C:\Users\PCUser\Documents\Basic Resume.docx.WNCRYT by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Rename: C:\Users\PCUser\Documents\Basic Resume.docx.WNCRYT -> C:\Users\PCUser\Documents\Basic Resume.docx.WNCRY by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Change: C:\Users\PCUser\Documents\Basic Resume.docx by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Create: C:\Users\PCUser\Documents\Bggreook1.xlsx.WNCRYT by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Rename: C:\Users\PCUser\Documents\Bggreook1.xlsx.WNCRYT -> C:\Users\PCUser\Documents\Bggreook1.xlsx.WNCRY by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Change: C:\Users\PCUser\Documents\Bggreook1.xlsx by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Create: C:\Users\PCUser\Documents\bluesky printing.pdf.WNCRYT by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Rename: C:\Users\PCUser\Documents\bluesky printing.pdf.WNCRYT -> C:\Users\PCUser\Documents\bluesky printing.pdf.WNCRY by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)
Change: C:\Users\PCUser\Documents\bluesky printing.pdf by 6836 (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe.exe)

```

Figure 28 - File System Activities of WannaCry Ransomware during an Attack, Recorded by FILE PATROL

After the completion of encrypting all files on its list, WannaCry changes the Windows Wallpaper to its own wallpaper containing warning information (figure 29), and displays the ransom message (figure 30).

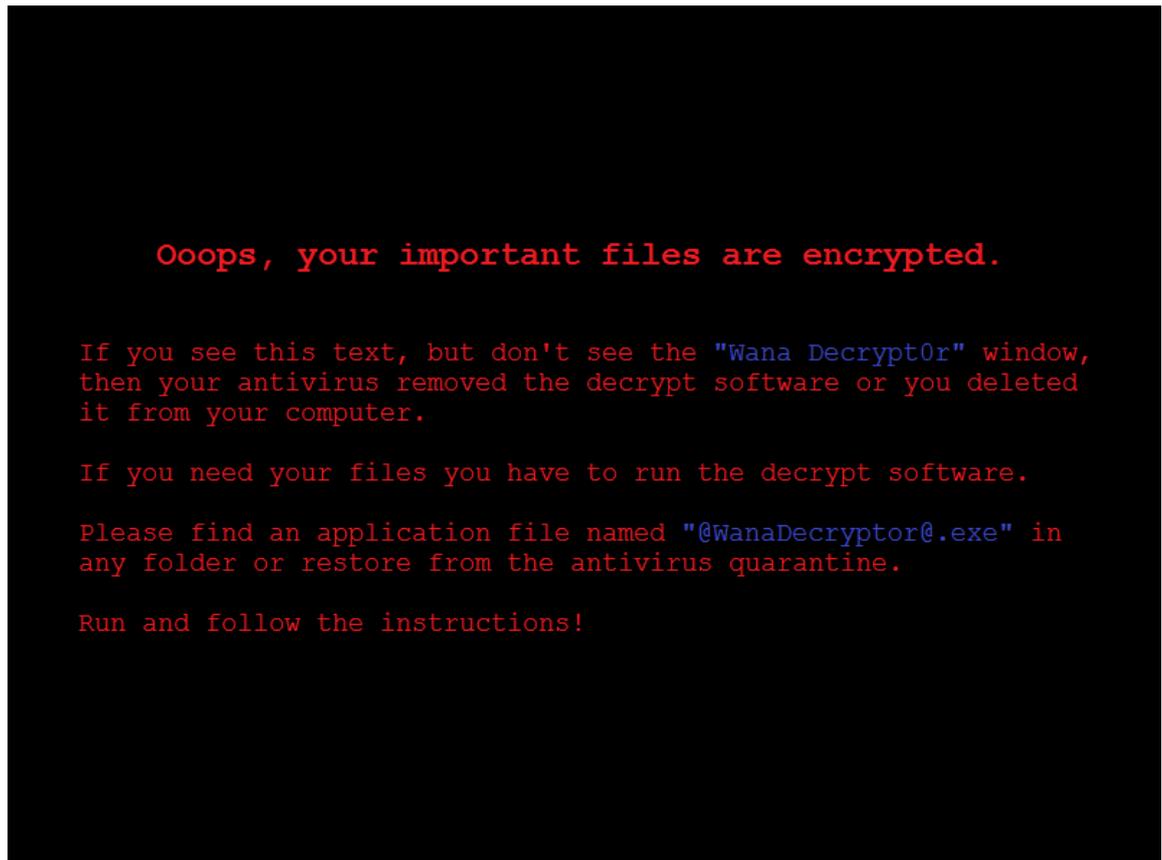


Figure 29 - Wallpaper Used by WannaCry to Display Ransom Message on User's Desktop Background



Figure 30 - the Ransom Message of WannaCry Displayed in English

Upon inspection of the “Documents” folder (figure 31), it was found that most of the documents have been renamed by WannaCry to carry the extension of WNCRY, and they were no longer able to be opened or viewed properly by their usual editors. There are two new files placed in the directory: @Please_Read_Me@.txt remains unencrypted and contains the FAQ about the WannaCry attack, and @WanaDecryptor@.exe would launch the decrypting UI interface to display ransom message and check ransom payments. Both files have names beginning with “@”, possibly because the ransomware authors want the two files to be ranked before filenames beginning with numbers or letters, to remain obviously visible to victims.

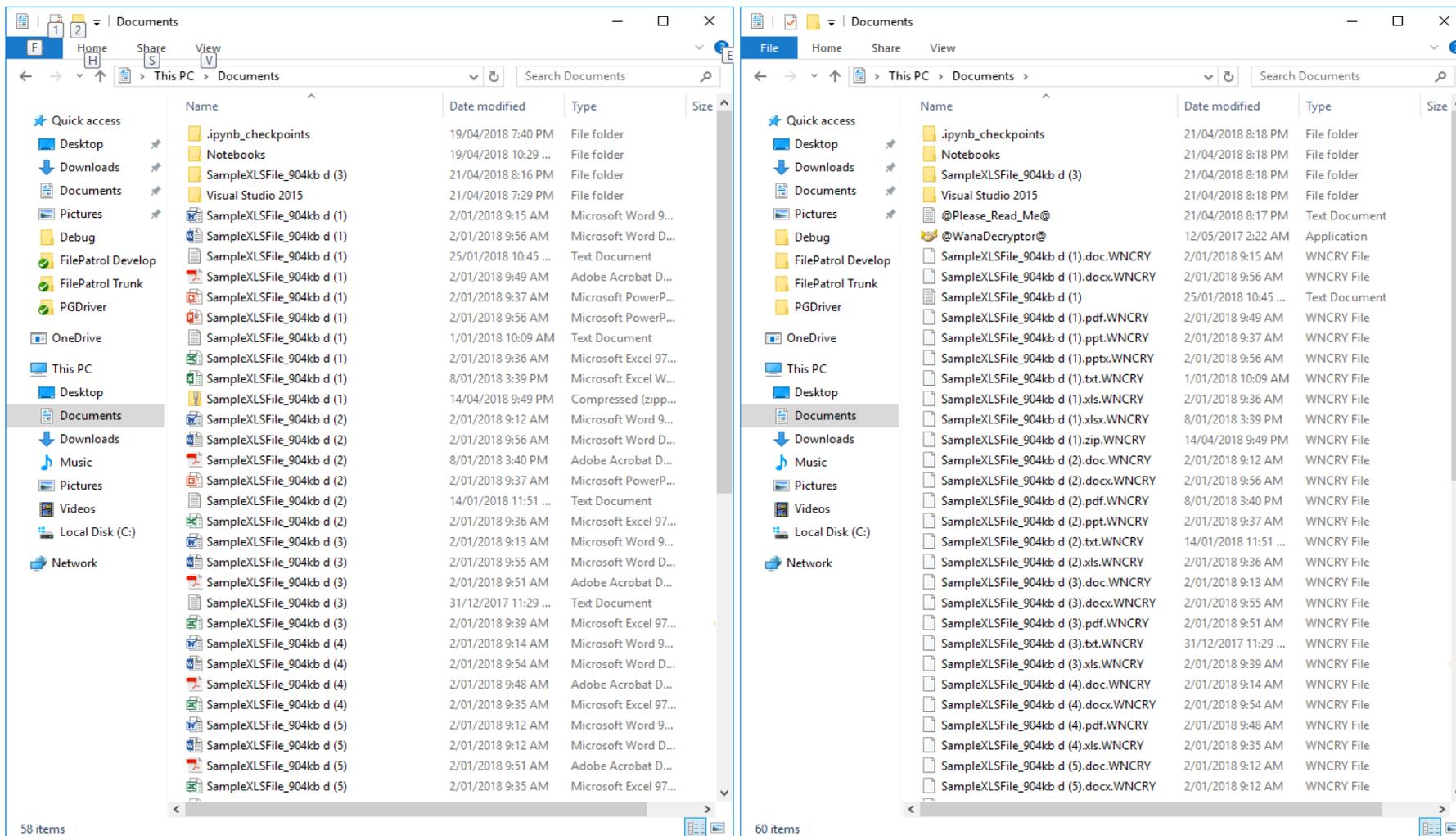


Figure 31 - Comparison of Directory Files before (Left) and after (right) WannaCry Attacks

A test file containing the plain test “The quick brown fox jumps over the lazy dog. 0123456789”, was encrypted by WannaCry into something completely illegible (figure 32).

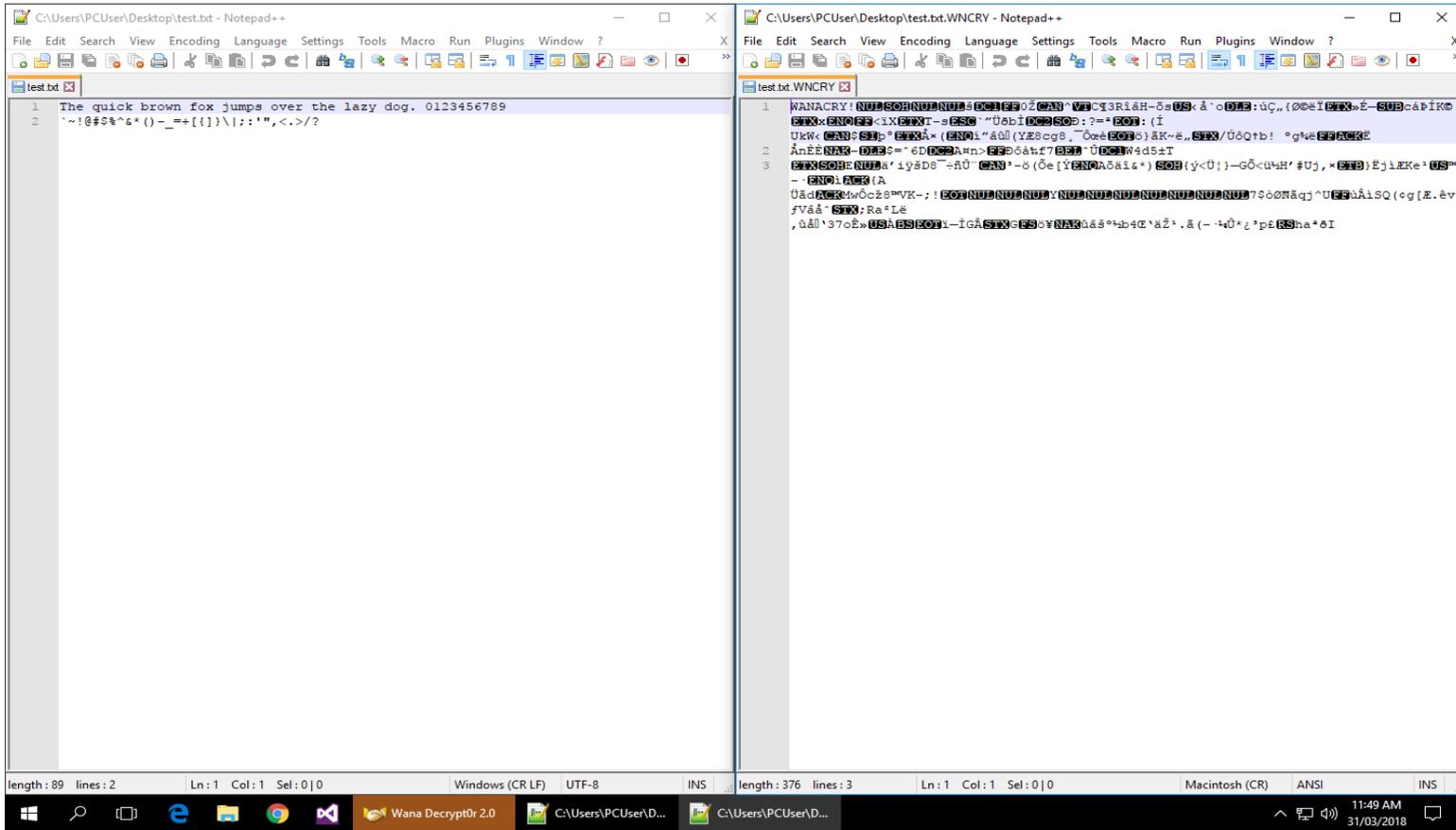


Figure 32 - Comparison of the Pure Text File Content before (Left) and after (Right) WannaCry Encryption

8.3 File Patrol in Action against WannaCry

When FILE PATROL was installed and running, it detected the first I/O request of WannaCry to attempt to commit changes. Because the WannaCry sample attempted to modify a PDF file but was not on the list of Recognized Processes, FILE PATROL rejected the I/O request by WannaCry, locked down the protected directory to prohibit all access, and terminated the offending process of WannaCry (figure 33 and figure 34). After the termination of WannaCry process, FILE PATROL released the lock on the protected directory and resumed all access. The ransomware attack was therefore thwarted.

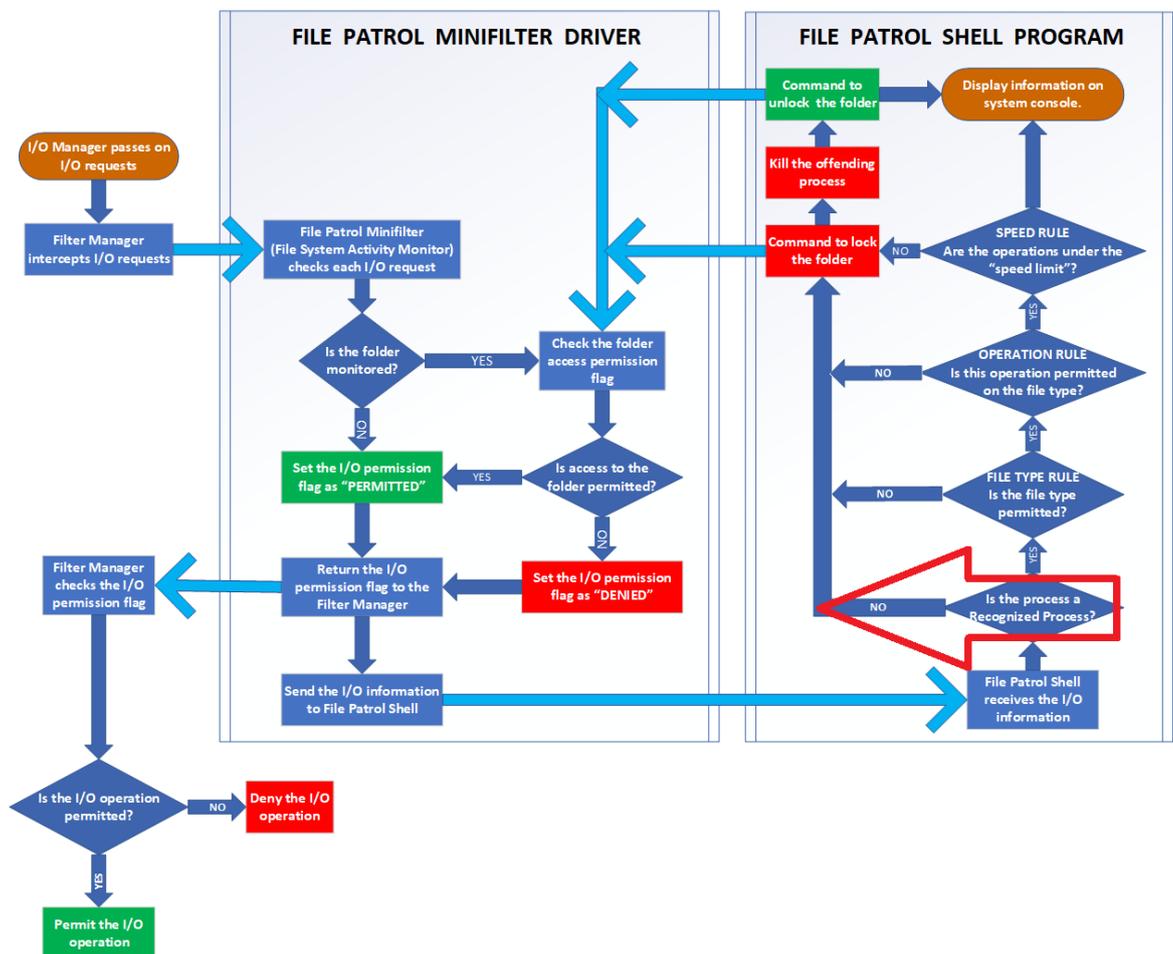


Figure 33 – WannaCry Was Not on The List of Recognized Processes and Was Denied Access by FILE PATROL

```

C:\Users\PCUser\Desktop\FilePatrol Develop\FilePatrolCmdApp\bin\x64\Release\FilePatrolCmdApp.exe
File Patrol (Anti-Crypto-Ransomware) Demo App
Massey University Master's student Timothy McIntosh
MInfSc project supervisors: A/Prof Julian Jang-Jaccard; Prof Paul Watters
Monitoring "C:\Users\PCUser\Documents"

Press <ESC> to exit...
Unrecognized process (PID 5072). Access denied.
Process (PID 5072) has been terminated.
Unrecognized process (PID 5072). Access denied.
Error: Access is denied
Unrecognized process (PID 5072). Access denied.
Error: Process with an Id of 5072 is not running.

```

Figure 34 - FILE PATROL in Action, Denying Access of Ransomware Processes and Terminating Them

8.4 FILE PATROL Can Neutralize WannaCry Masquerading as Microsoft Word

In section 4.3, a theoretical possibility was demonstrated when a ransomware could masquerade as one of the legitimate programs that had been whitelisted by the security software.

The same *Ransom.WannaCry* sample was renamed into “WINWORD.exe”, placed into the installation directory of Microsoft Office, and executed from there. The fake “WINWORD.exe” was executed using Administrator privileges, to enable *Ransom.WannaCry* to generate temporary files in the same directory. Two test runs were performed:

- i. 1st test run: using the same test files of different file types in “Documents” directory.
- ii. 2nd test run: using only “*.doc” and “*.docx” files in “Document” directory.

The second test run was designed to evaluate whether RANDETER could detect *Ransom.WannaCry* attacks, when *Ransom.WannaCry* had no chance of accessing files that were usually not edited using Microsoft Word.

In both test runs, by having the correct application file name in the correct program installation directory, *Ransom.WannaCry* (as the fake “WINWORD.exe”) was able to be

identified as the Recognized Process of Microsoft Word, but failed to pass the Multi-Layered Security Rules of RANDETER. During the first test run (figure 35), *Ransom.WannaCry* was trying to access a PDF file; a PDF file can only be created by Microsoft Word, not modified by it, so the fake “WINWORD.exe” was denied access and killed. During the second test run (figure 36), *Ransom.WannaCry* was trying to modify (*Change*) a DOCX file, and it passed the “File Type Rule”, but failed the “Operation Rule”; Microsoft Word only *Renames* the original file into a temporary file, and *Renames* a temporary file with modified file content into the target DOCX file. As a result, the fake “WINWORD.exe” was denied access and killed again.

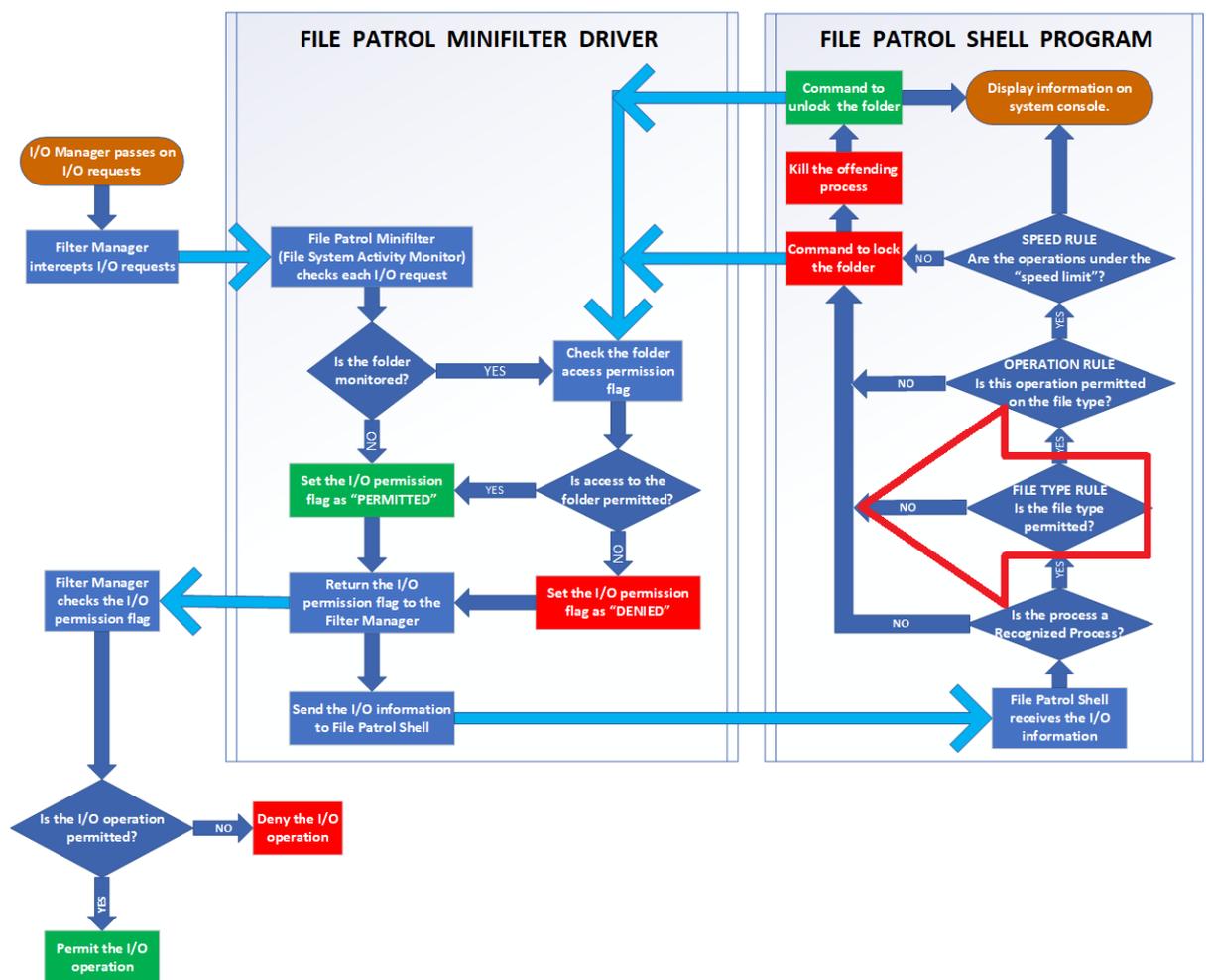


Figure 35 - First Test Run with Different File Types: During A Simulated Masquerading Attack as Microsoft Word, Ransom.WannaCry Fails the "File Type Rule"

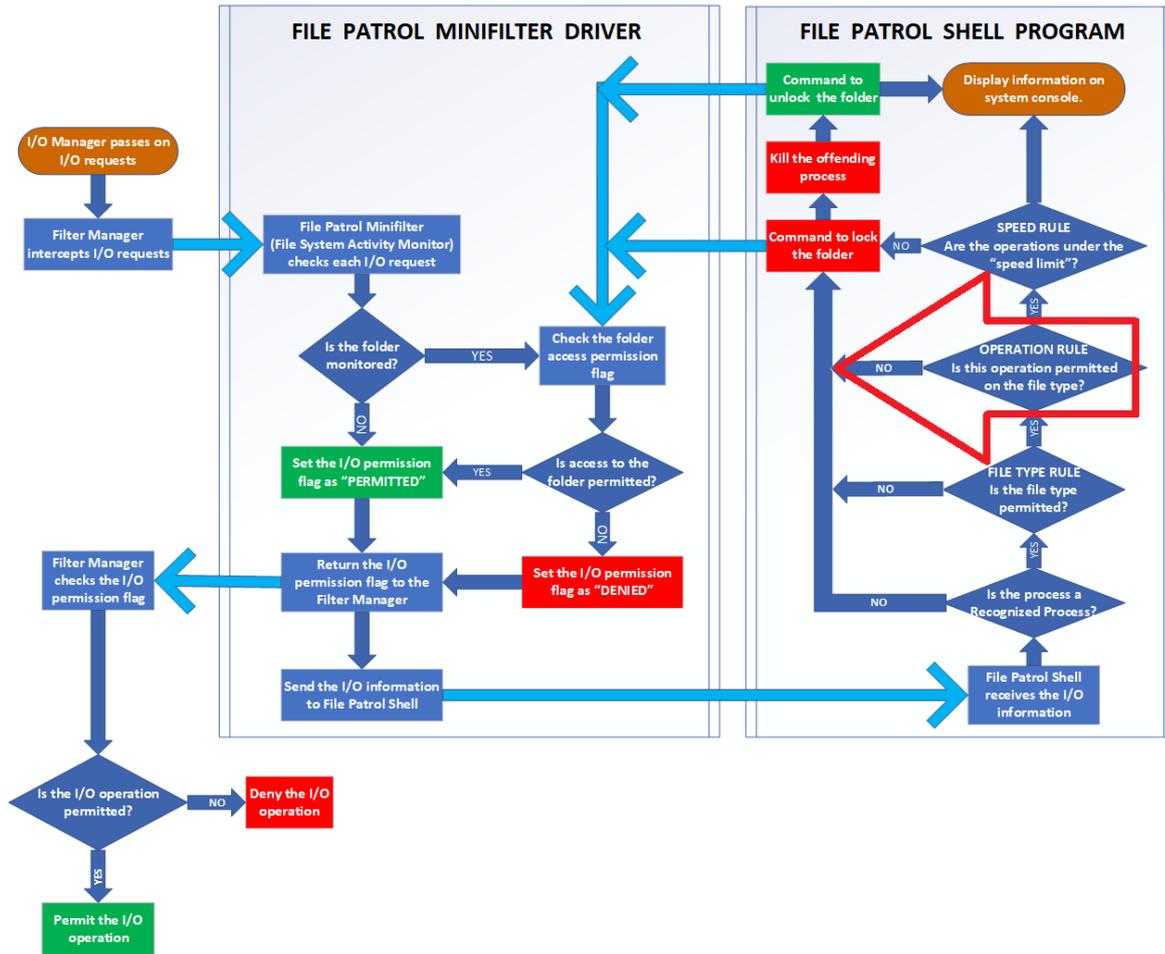


Figure 36 - Second Test Run with Only DOC and DOCX Files: During a Simulated Masquerading Attack as Microsoft Word, Ransom.WannaCry Fails the "Operation Rule"

Chapter 9. A CASE STUDY OF PETYA

In this chapter, *Ransom.Petya* is examined to demonstrate how its attacks could be deterred by PARTITION GUARD module of RANDETER.

9.1 Petya Introduction

Petya, first discovered on March 29, 2016, infected the MBR to execute a payload that encrypted the \$MFT, and prevented Windows from booting properly (Symantec Security Center, 2017 July). A later semi-clone variant of Petya in 2017, also known as NotPetya or DiskCoder, and evolved to include self-propagation. The 2017 new variant of Petya was extremely powerful and infected mainly Ukrainian and Russian entities, including banks, government agencies, media companies and utility companies. Petya used the same Windows security exploit EternalBlue as WannaCry, but also scanned for vulnerable peer computers within organizations using SMB protocols, and spread to them like a worm (Symantec, 2017).

Upon further investigation, Symantec (2017) classified Petya as a “disk wiper”, because the randomly generated Salsa20 key, used to encrypt the disk, had no actual relationship with the “installation key”, which meant victims could not recover their files even if they paid the ransom. In combination with the evidence that Petya was mainly targeting Ukraine, Symantec (2017) believed that the development of Petya might have been politically motivated.

9.2 Petya Attack in Details

Upon execution, Petya would infect the booting disk’s MBR, replace the Windows bootloader with its own malicious bootloader, and trigger the blue screen with a 0xc0000350 error to force a restart.

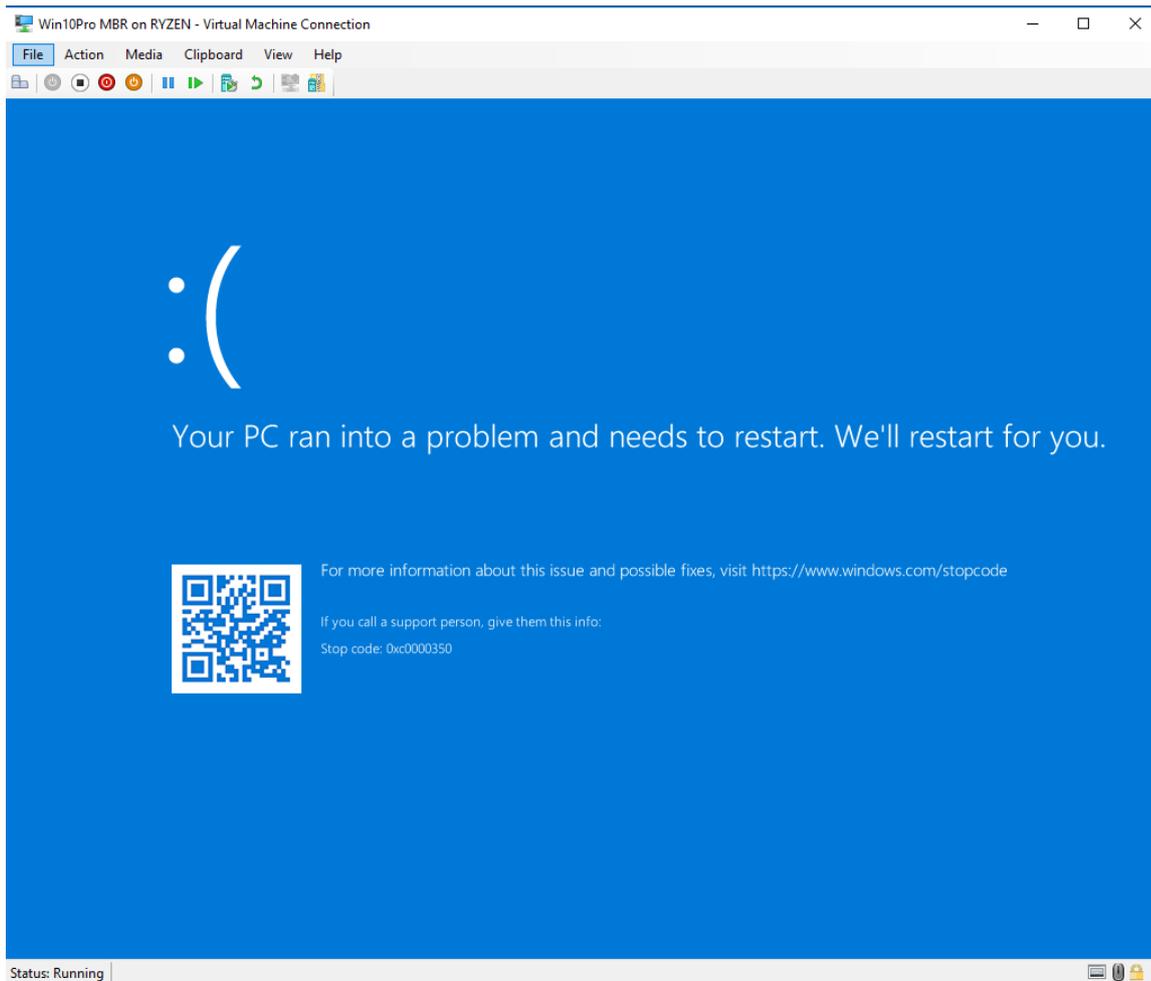


Figure 37 - Ransomware Petya Triggers the Blue Screen with a 0xc0000350 Error to Force a System Restart

After system restarts, the payload of Petya is executed and would output texts similar to the output of CHKDSK, the Windows inbuilt file system scanner, to try to convince the user that a hard disk repair is in progress, while it is encrypting the \$MFT (figure 38).

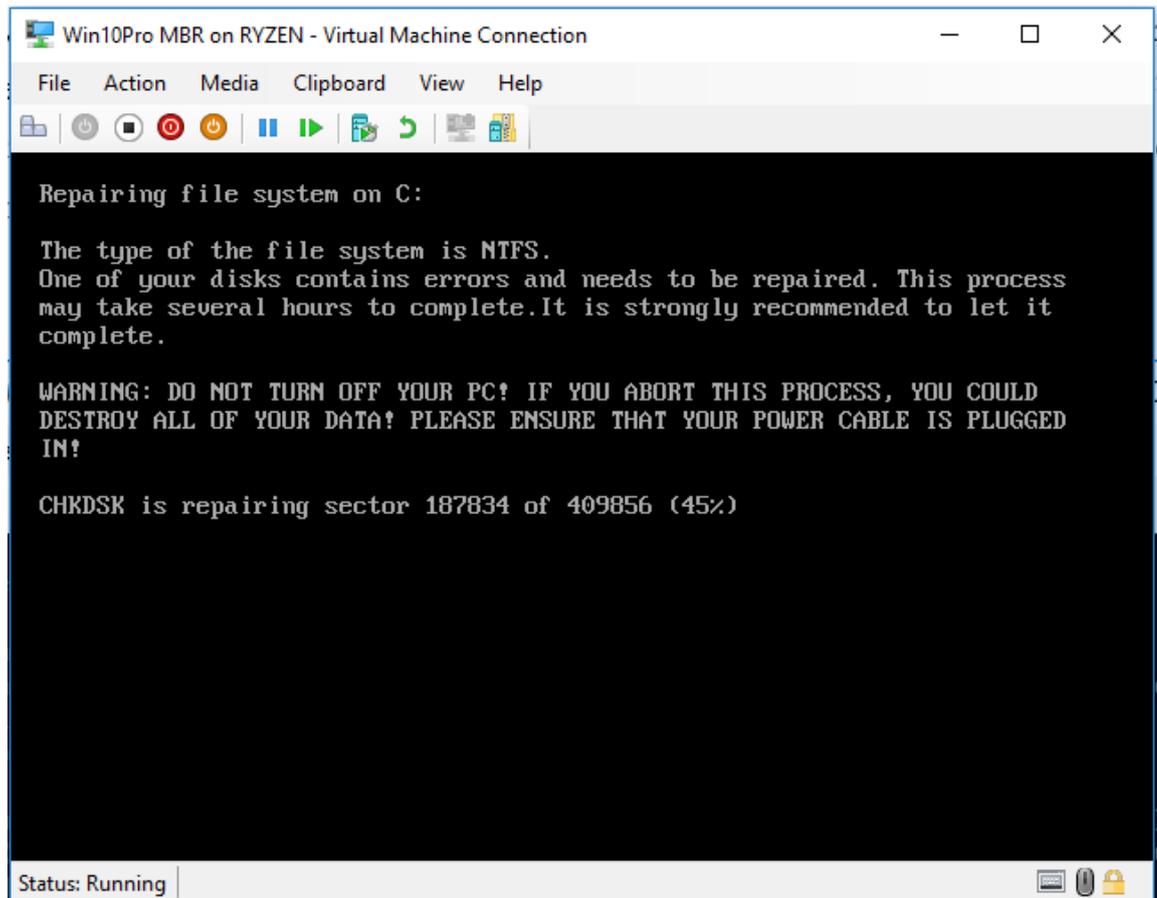


Figure 38 - Ransomware Petya Masquerading as CHKDSK while Encrypting the Master File Table

After the encryption is completed, the ransom message is displayed (figure 39). It is no longer possible to boot into Microsoft® Windows, as the Windows bootloader has been replaced by Petya.

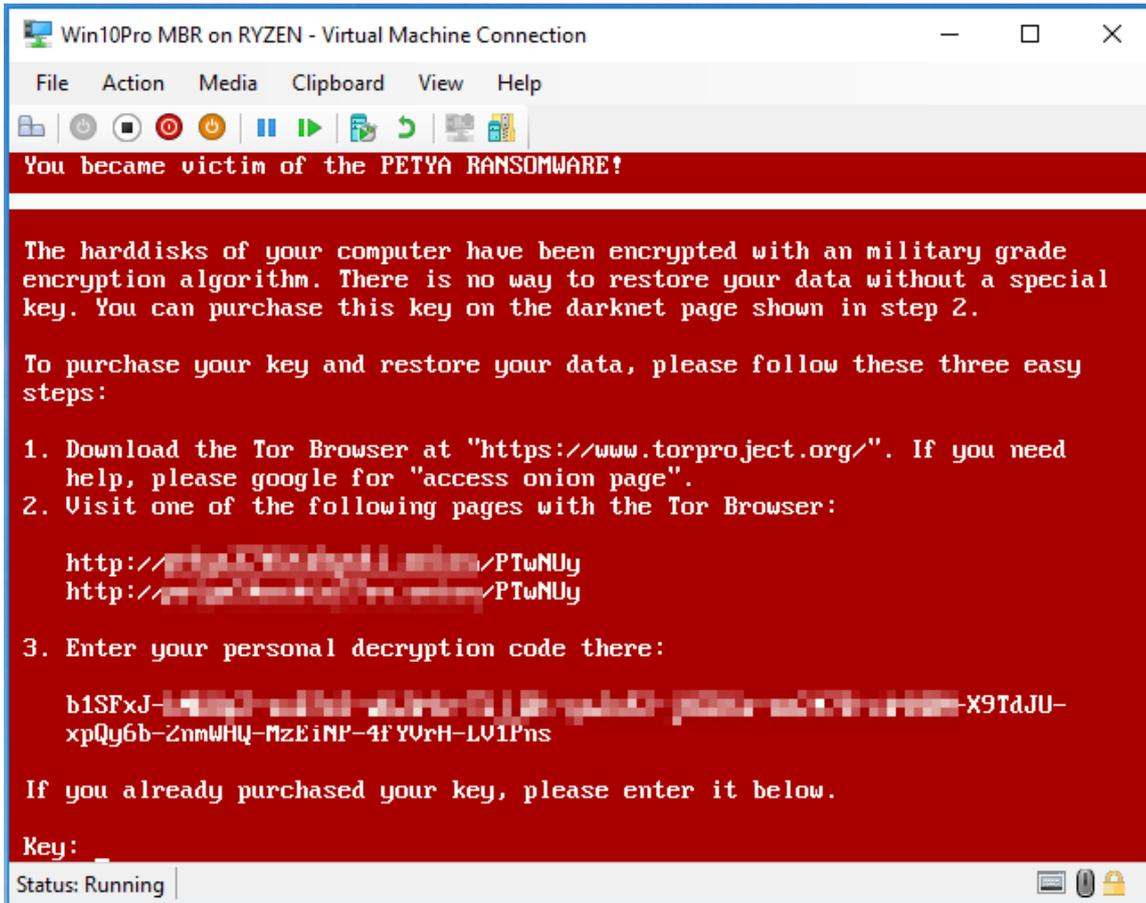


Figure 39 - Ransom Message Displayed by Ransomware Petya

9.3 PARTITION GUARD in Action against Petya

Ransom.Petya was trying to write its malicious bootloader into sector 0 of the boot drive via SCSI pass-through. When PARTITION GUARD driver is installed, the malicious “write” access by Petya is denied. Petya would error and quit (figure 40 and figure 41).

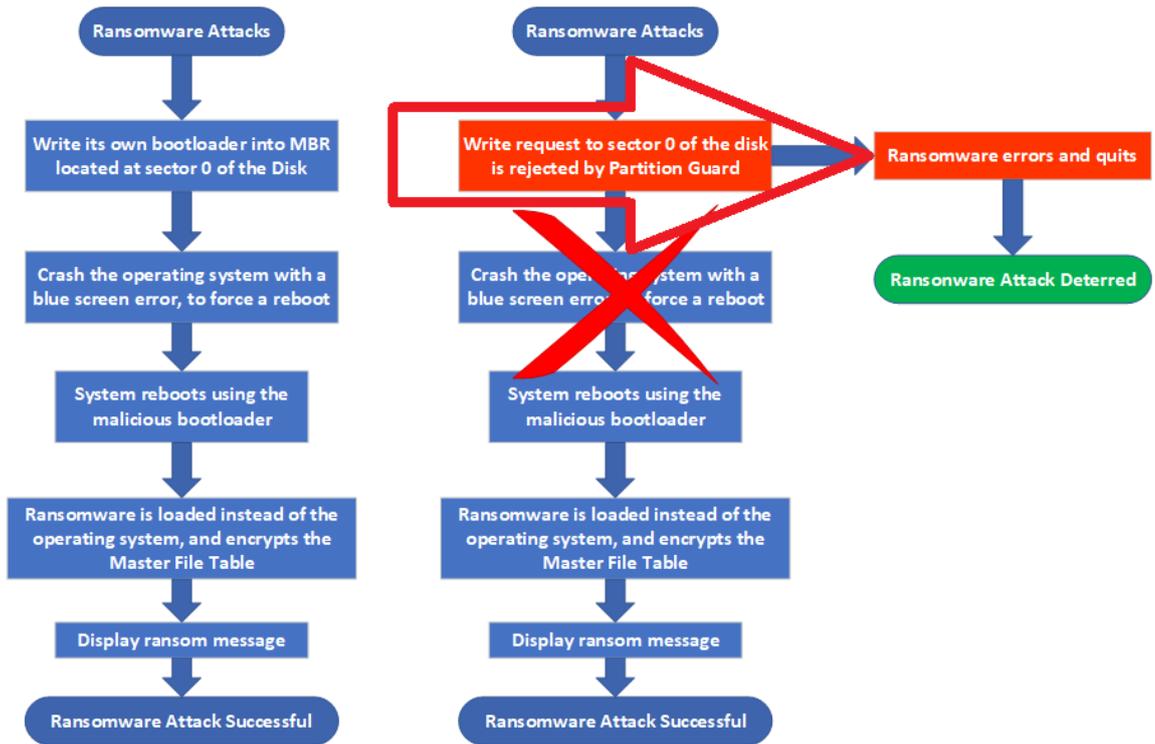


Figure 40 - Flowchart of Ransomware Attack on MBR (Left), and How It Is Deterred by PARTITION GUARD (Right)

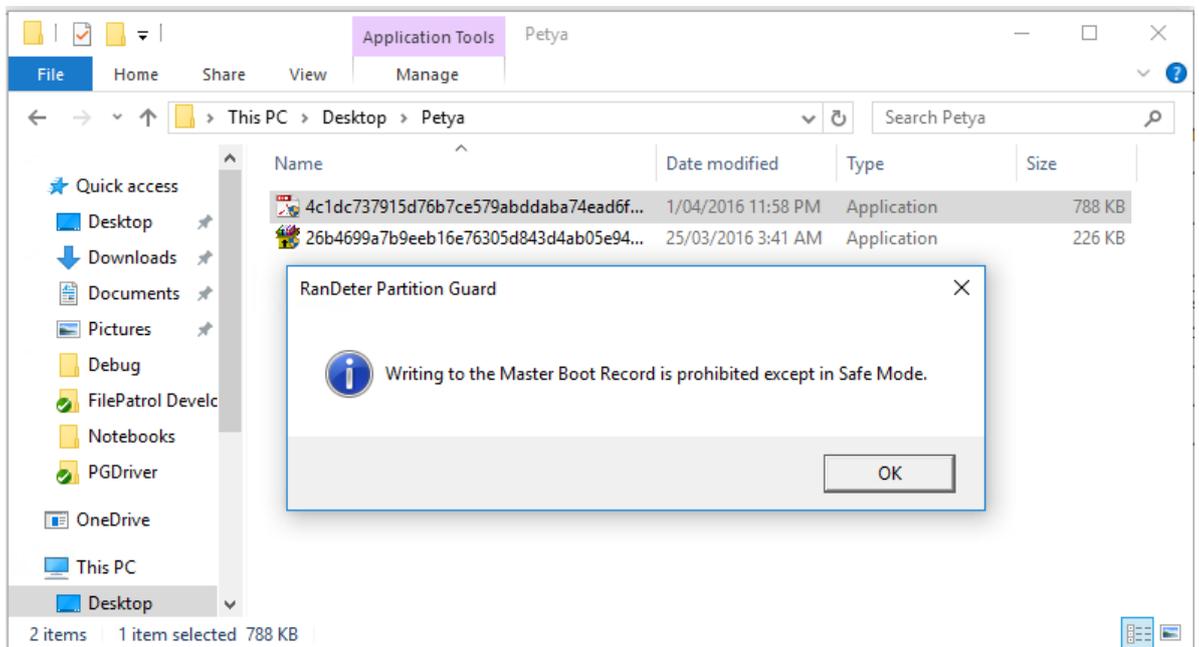


Figure 41 - PARTITION GUARD in Action, Denying Write Access to MBR

Chapter 10. DISCUSSIONS AND LIMITATIONS OF RANDETER

Ransomware research and detection is like an arms race; ransomware developers would exploit every possible security loophole and human weakness to attempt to infiltrate victims' systems and bypass ransomware detection systems. Below is a list of limitations and possible enhancements of RANDETER, in decreasing order of significance.

10.1 Risks of Attacks on the System of Recognized Processes

FILE PATROL relies on the list of Recognized Processes with Multi-Tier Security Rules to check whether applications are recognized and are behaving as prescribed. Theoretically, attacks on the System of Recognized Processes is possible, but would be difficult in practice; table 10 lists some possible ways of compromising the System of Recognized Processes, and their mitigation strategies.

Table 11 – Summary of Risks of Attacks on the System of Recognized Processes and Mitigation Strategies

Risk of Compromising the System of Recognized Processes	Mitigation Strategies
Ransomware need to know what benign applications are on the list of Recognized Processes by FILE PATROL, and the installation paths of those benign applications.	Keep the list of Recognized Processed confidential.
Ransomware need to replace the benign applications in their installation directories, in order to carry out masquerading attacks on file types that could usually be edited by such applications.	(1) Monitor the installation paths of benign applications (2) Ask user to authorize application version updates

	(3) When benign applications are replaced by ransomware, the usual functionality of the software could no longer be fulfilled, and the “malfunction” will be noticed.
Ransomware need to know the threshold of the pre-defined multi-tier security rules, and carefully operate only on selected file types permitted to the benign application, perform only permitted file operations and within the “speed limit”	<p>(1) Keep the multi-layered security rules confidential.</p> <p>(2) Enhance “speed limit” to introduce limits on frequency of file system activities of applications</p> <p>(3) Use statistics from the user and big data to establish baseline of the security rules.</p>

10.2 Possibility of Total Prevention of Irrecoverable File Damage

In the evaluations of RanDeter, some files were encrypted and damaged by test ransomware samples and became irrecoverable. However, total prevention of file damage by crypto-ransomware is possible, but the architecture of FILE PATROL will need to be modified. The FILE PATROL minifilters driver is based on Microsoft® Minispy File System Minifilter Driver, with minimal modifications. The minifilter only logs file system events and sends the information to the Shell program, and it can aggregate events, sending multiple events in one communication roundtrip. The Shell program checks the IRP events against a set of rules, and when required, sends the command to freeze the protected folder. The file damage observed during the evaluation (table 8) could be because the ransomware already started damaging files, before the minifilters receives the command from the Shell program to freeze the protected folder. It is possible to move

the checking logic into the minifilter driver, so any unrecognized program can cause an immediate freeze on the protected folder, before the IRP operation completes.

10.3 Improved Admission into Recognized Processes

Admission into Recognized Processes can be improved by using statistics on a big group of users. Maintaining an accurate and up-to-date list of Recognized Processes is key to the success of FILE PATROL module. If the list is compromised, there is potential that ransomware can infiltrate this access control and cause damage. If admission is too strict or too difficult, it could inconvenience users who wish to use 3rd party programs.

The current proof-of-concept implementation of RANDETER is only distributed with a default list of Recognized Processes. Further improvements are required to improve the list and fine-tune the security rules to adapt to user environment. It is possible to draw inspirations from SHIELDIFS (Continella et al, 2016) and REDEMPTION (Kharraz & Kirda, 2017), by using machine-learning to automate the process. It is also possible to take a cloud-security approach to gather usage data on file activities from a large cohort, by implementing something similar to Norton® Community Watch and ESET LiveGrid®.

10.4 Reducing Possibility of Denial of Service Attacks

It is in theory possible to launch Denial of Service Attack on RANDETER modules. Both PARTITION GUARD and FILE PATROL drivers run in the privileged kernel mode, but the shell program of FILE PATROL runs in the administrative user mode and must connect to the \\FilePatrolPort to communicate with the FILE PATROL minifilter driver. A ransomware could attempt to connect to this port before FILE PATROL shell program and mark itself as legitimate, or unfroze the protected directory. This risk can be mitigated by adding a two-way handshake authentication similar to the CHAP protocol (Simpson, 1997), between the FILE PATROL minifilter and the FILE PATROL shell program; the authentication could be based on shared secrets to avoid key exchanges, and the minifilters will reject communication with unauthenticated shell programs.

Chapter 11. CONCLUSION

Ransomware are a major cybersecurity threat, funded by the quick profit and mature criminal model and continue to evolve to evade detection. Crypto-ransomware are the most damaging form of ransomware that encrypt user data and demand ransom; they can make user data irrecoverable and lead to financial and data losses. Traditional static analysis based on signature is no longer always effective. Dynamic analysis is possible, but most focus on the data-centric behavior of ransomware. Various researchers have proposed different ways of detecting and deterring crypto-ransomware attacks, and those on monitoring file system activities appeared most promising. The correct indicators must be selected and used during the monitoring, to maximize detection rates and minimize false positives.

The approach to detect and deter ransomware can take inspirations from the Police anti-terrorism practices, because there are similarities between ransomware damage and terrorism casualties. While previous studies on monitoring file system activities have provided useful insights on the nature of ransomware attacks, monitoring file system activities of the whole system is probably not necessary, because crypto-ransomware have a preference to access and modify user files. An effective deterrent approach would be to only guard common attack targets (user files and MBR) and neutralize any suspicious threats. It has been observed that ransomware perform file system activities that are different to those of benign applications, in that for activities by ransomware, there are generally more file types and more file operations involved at a faster speed. The concept of the list of Recognized Programs with Multi-Tier Security Rules is proposed as an enhanced implementation of application whitelisting, and believe the implementation can effectively differentiate benign activities and crypto-ransomware like activities, and even prevent “masquerading attacks”, to which “Controlled Folder Access” is susceptible.

RANDETER is implemented as a dual-module implementation on Microsoft Windows operating systems to defend against crypto-ransomware on end-hosts: PARTITION GUARD protects MBR while FILE PATROL carefully examines access patterns to user files in selected directories that need protection. RANDETER was able to successfully deter most recent crypto-ransomware attacks.

RANDETER is foreseen to be an effective, lightweight and evolving anti-crypto-ransomware solution that could provide automatic deterrent and termination of ransomware-like activities, while minimizing usage of system resources. By augment service to the Windows operating systems, it is possible to completely deter ransomware attacks on end-user systems. In addition, RANDETER does not require any other environmental prerequisites or any explicit application compatibility to provide active protection against unknown ransomware. RANDETER may be able to be improved by adding more layers of security rules, should new attack patterns of ransomware be discovered.

REFERENCES

1. Agrawal, N., Bolosky, W. J., Douceur, J. R., & Lorch, J. R. (2007). A five-year study of file-system metadata. *ACM Transactions on Storage (TOS)*, 3(3), 9.
2. Ahmadian, M. M., Shahriari, H. R., & Ghaffarian, S. M. (2015, September). Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares. In *Information Security and Cryptology (ISCISC), 2015 12th International Iranian Society of Cryptology Conference on* (pp. 79-84). IEEE.
3. Australian Signals Directorate. (2017). Strategies to Mitigate Cyber Security Incidents – Mitigation Details.
4. Barreau, D., & Nardi, B. A. (1995). Finding and reminding: file organization from the desktop. *ACM SigChi Bulletin*, 27(3), 39-43.
5. Brewer, R. (2016). Ransomware attacks: detection, prevention and cure. *Network Security*, 2016(9), 5-9.
6. CERT-EU. (2017). “Wannacry ransomware campaign exploiting SMB vulnerability.”
<https://cert.europa.eu/static/SecurityAdvisories/2017/CERT-EU-SA2017-012.pdf>
[Accessed 7 Mar 2018].
7. Continella, A., Guagnelli, A., Zingaro, G., De Pasquale, G., Barengi, A., Zanero, S., & Maggi, F. (2016, December). SHIELDIFS: a self-healing, ransomware-aware filesystem. In *Proceedings of the 32nd Annual Conference on Computer Security Applications* (pp. 336-347). ACM.
8. Continella, A., Guagnelli, A., Zingaro, G., De Pasquale, Barengi, A., Zanero, S., & Maggi, F. (n.d.) SHIELDIFS: The Last Word in Ransomware Resilient Filesystems.
9. Department of the Army (2011). Anti-terrorism. Field Manual No. 3-37.2 . Available at:
<https://fas.org/irp/doddir/army/fm3-37-2.pdf> [Accessed 19 Mar 2018].
10. ESET. (2017). ESET vs Crypto-Ransomware; What, how and why. Available at:
https://cdn1.esetstatic.com/ESET/US/resources/white-papers/WhitePaper_ESET-vs-Crypto-Ransomware.pdf [Accessed 7 Mar 2018].

11. Fayi, S. Y. A. (2018). What Petya/NotPetya Ransomware Is and What Its Remediations Are. In *Information Technology-New Generations* (pp. 93-100). Springer, Cham.
12. Halsey, M., & Bettany, A. (2015). Understanding Windows File Systems. In *Windows File System Troubleshooting* (pp. 13-30). Apress, Berkeley, CA.
13. Hunt, S., & Tiernan, S. (2018). Ransomware Protection For Cloud File Storage. *U.S. Patent Application No. 15/201,007. U.S. Patent Publication No. US20180007069A1*.
14. Islam, R., Tian, R., Batten, L. M., & Versteeg, S. (2013). Classification of malware based on integrated static and dynamic features. *Journal of Network and Computer Applications*, 36(2), 646-656.
15. Ismail, M. (2017). Sony Pictures and the US Federal Government: A Case Study Analysis of the Sony Pictures Entertainment Hack Crisis Using Normal Accidents Theory. Masters thesis submitted to the University of Southern Mississippi.
16. Karresand, M., & Shahmehri, N. (2006, June). File type identification of data fragments by their binary structure. In *Proceedings of the IEEE Information Assurance Workshop*(pp. 140-147).
17. Kharraz, A., Arshad, S., Mulliner, C., Robertson, W. K., & Kirda, E. (2016, August). UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware. In *USENIX Security Symposium* (pp. 757-772). Mansfield-Devine, S. (2016). Ransomware: taking businesses hostage. *Network Security*, 2016(10), 8-17.
18. Kharraz, A., & Kirda, E. (2017, September). REDEMPTION: Real-Time Protection Against Ransomware at End-Hosts. In *International Symposium on Research in Attacks, Intrusions, and Defenses* (pp. 98-119). Springer, Cham.
19. Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., & Kirda, E. (2015, July). Cutting the gordian knot: A look under the hood of ransomware attacks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (pp. 3-24). Springer, Cham.
20. Kolodenker, E., Koch, W., Stringhini, G., & Egele, M. (2017, April). PayBreak: defense against cryptographic ransomware. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (pp. 599-611). ACM.
21. Liska, A., & Gallo, T. (2016). *Ransomware: Defending Against Digital Extortion*. Sebastopol, USA: O'Reilly Media.
22. Luo, X., & Liao, Q. (2007). Awareness education as the key to ransomware prevention. *Information Systems Security*, 16(4), 195-202.

23. Martignoni, L., Christodorescu, M., & Jha, S. (2007, December). Omniunpack: Fast, generic, and safe unpacking of malware. In *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual* (pp. 431-441). IEEE.
24. Meeuwisse, R. (2016). *Cybersecurity: Home and Small Business*. Hythe, United Kingdom: Cyber Simplicity Ltd.
25. Microsoft. (2009). Volume Shadow Copy Service Technical Reference. Available at: [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc738819\(v%3dws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc738819(v%3dws.10)) [Accessed 30 Jan 2018]
26. Microsoft. (2017). Protect important folders with Controlled folder access. Available at: <https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-exploit-guard/controlled-folders-exploit-guard> [Accessed 11 Mar 2018]
27. Microsoft Hardware Dev Center. (2017a). Major Function Codes. Available at: <https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/irp-major-function-codes> [Accessed 11 Feb 2018]
28. Microsoft Hardware Dev Center. (2017b). Advantages of the Filter Manager Model. Available at: <https://docs.microsoft.com/en-us/windows-hardware/drivers/ifs/advantages-of-the-filter-manager-model> [Accessed 11 Feb 2018]
29. Microsoft Windows IT Pro Center. (2016). Hyper-V feature compatibility by generation and guest. Available at: <https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/hyper-v-feature-compatibility-by-generation-and-guest>
30. Moore, C. (2016, August). Detecting Ransomware with Honeypot Techniques. In *Cybersecurity and Cyberforensics Conference (CCC), 2016* (pp. 77-81). IEEE.
31. O'Gorman, G., & McDonald, G. (2012). *Ransomware: A growing menace*. Symantec Corporation.
32. Moser, A., Kruegel, C., & Kirda, E. (2007, December). Limits of static analysis for malware detection. In *Computer security applications conference, 2007. ACSAC 2007. Twenty-third annual*(pp. 421-430). IEEE.
33. Pathak, D. P., & Nanded, Y. M. (2016). A dangerous trend of cybercrime: ransomware growing challenge. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume, 5*.

34. Salvi, M. H. U., & Kerkar, M. R. V. (2016). Ransomware: A cyber extortion. *Asian Journal of Convergence in Technology*.
35. Savage, K., Coogan, P., & Lau, H. (2015). The evolution of ransomware. *Symantec, Mountain View*.
36. Scaife, N., Carter, H., Traynor, P., & Butler, K. R. (2016, June). Cryptolock (and drop it): stopping ransomware attacks on user data. In *Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on* (pp. 303-312). IEEE.
37. Schmugar, C. D., Cochin, C., Furtak, A., Carrivick, A. J., Bulygin, Y., Loucaides, J. J., ... & Heitzmann, G. M. (2018). *U.S. Patent Application No. 15/210,165. U.S. Patent Publication No. US20180018458A1*.
38. Schuknecht, L. (2016). Average PC Memory (RAM) Continues to Climb. Available from:
<https://techtalk.pcpitstop.com/2016/10/05/average-pc-memory-ram-continues-climb/>
[Accessed 8 Mar 2018].
39. Sencar, H. T., & Memon, N. (2009). Identification and recovery of JPEG files with missing fragments. *Digital Investigation*, 6, S88-S98.
40. Sgandurra, D., Muñoz-González, L., Mohsen, R., & Lupu, E. C. (2016). Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection. *arXiv preprint arXiv:1609.03020*.
41. Simmonds, M. (2017). How businesses can navigate the growing tide of ransomware attacks. *Computer Fraud & Security*, 2017(3), 9-12.
42. Simpson, W. A. (1996). PPP challenge handshake authentication protocol (CHAP).
43. Symantec (2017). Internet Security Threat Report - April 2017. ISTR - Volume 22. [online] Symantec. Available at:
<https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf>
[Accessed 27 Jan 2018].
44. Symantec Security Center (2017, May). Ransom.WannaCry. Available at:
<https://www.symantec.com/security-center/writeup/2017-051310-3522-99>
[Accessed 29 Jan 2018]
45. Symantec Security Center (2017, July). Ransom.Petya. Available at:
<https://www.symantec.com/security-center/writeup/2016-032913-4222-99>
[Accessed 29 Jan 2018]

46. Taubman, D., & Marcellin, M. (2012). *JPEG2000 image compression fundamentals, standards and practice: image compression fundamentals, standards and practice* (Vol. 642). Springer Science & Business Media.
47. Tian, R., Islam, R., Batten, L., & Versteeg, S. (2010, October). Differentiating malware from cleanware using behavioural analysis. In *Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on* (pp. 23-30). IEEE.
48. Troelsen, A., & Japikse, P. (2017). *Pro C# 7: With .NET and .NET Core* (8th ed.). New York City, NY: Apress.
49. Wikipedia. (n.d.). WannaCry ransomware attack.
https://en.wikipedia.org/wiki/WannaCry_ransomware_attack [Accessed 29 Jan 2018]
50. Zavarisky, P., & Lindskog, D. (2016). Experimental Analysis of Ransomware on Windows and Android Platforms: Evolution and Characterization. *Procedia Computer Science*, 94, 465-472.

APPENDIX 1. LIST OF RANSOMWARE SAMPLES

TESTED WITH RANDETER

Below is a list of ransomware samples being tested with RANDETER, and their SHA-256 hash values.

Classification	First Reported	SHA-256 Value
BadRabbit	2017-10-24	630325cac09ac3fab908f903e3b00d0dadd5fdaa0875ed8496fcb97a558d0da
	2018-04-29	4116f64adb31593455592e2d5a1be0717a51a3d83952cd8a82123b7b7d9c7406
	2018-05-19	57f5dcee852b7f0be74ec238b743a0b15b9988ca8363958b083350f30b5d2349
	2017-10-24	8ebc97e05c8e1073bda2efb6f4d00ad7e789260afa2c276f0c72740b838a0a93
	2018-05-19	6f3298aa140841be87fa4fb9eaa1576fc4a109f3550b8b09abb6e3231e6c1662
	2018-05-19	ec0a4d8f34141e9575e3cf073cf9d7324ee9a1b494ab36716a38536bb14badfd
	2018-05-19	c12781ba57c9663430fa2f5d5a21f2ddcfba30dc916eddc760d32a5f36324968
Cerber	2017-05-23	e67834d1e8b38ec5864cfa101b140aeaba8f1900a6e269e6a94c90fcbfe56678
	2018-04-04	26f5a4b79ee5a6cc0acacd5d285a10907ff9eb2d32af5bddef3ad81f663a5b14
	2018-04-20	1dc0cf9b5362b52a0754e278f30c2cdf80716271bb58bd5cecc3f5352d46f3cff
	2018-04-13	9a039267871ea2264773dc87a10a82aecb8408f068c21d72c7501301bd6a934b
	2018-04-13	63a2892fc9190fd41e4e73522f39d41e7c6737bce492f9689638e370a0a8d878
	2017-05-01	68daf44d57a4d13701eb66b637a00cc6931fb913515a7c95dec3a318c0365968
	2018-04-06	ce16c87d4d5e5c87a3ad718496a571223abdb7963a5b5e29fec4010a5c4a6369
	2017-03-07	d9dd61a68fc13a903235dab6f73beb063c2dd442a14d8024782fb103783a7fff
	2018-03-21	7d9593302a22fac4a1a604c5a31999b90b68c2204e4cb2c60c6616065a0b75ad
	2018-03-20	5ef5525e1329c69e566bf61238ebc330c978d2bcb3c893ee0b6cf61c57827a60
CryptoLocker	2014-01-22	5291232b297dfcb56f88b020ec7b896728f139b98cef7ab33d4f84c85a06d553
	2014-01-22	8cf50ae247445de2e570f19705236ed4b1e19f75ca15345e5f00857243bc0e9b
	2016-02-17	68ae6cf101448a889010de25d3f766f5dfcdbe94918a68b864b7ae2ca7005d89
	2018-04-18	13b9fe4cbb4186e02809bef0ced44e69ccff4cafc3d501a3230795541c6bbd1c
	2018-04-13	2e44947eee627e6320a177091e4196b88dace542586cc423ade12daab51de89
	2018-03-30	c3de29d2590768cc1839a11780c4e82bdeac6cc3eddf42e4fcf749a5e3875f38
	2018-03-27	777e00ff27b488add560c508fb641714a7c9827d84824d5898274a542f023d7

	2018-03-27	7aea2170574c8184cbc0be44aac12d0d69369e69d7adc357d607b9505417b535
	2018-03-26	ef3186ab35abe909348200edfb1604a3782e2290ef49e3491fbd6e70991f5734
	2018-03-28	aca6af7ee8287c2eb9bd867507944b3fcea13b32a9ec9db6ea6d4b057086991a
	2018-03-20	3079b8da2d1c20755072ac815abcf701cd8756e6daaa279a10ae92eb89873a2e
	2018-03-16	4c1edc113d4ecc6c8208b7d553d31a26b11dfb3c701a790a5b915ffa373f0718
GandCrab	2018-01-30	c7f1df930d6764022aa35602321a82b48518d007326184d1911450f7315a46c2
	2018-03-31	1641bf31059677e5d796d7662921905062a3e1d6365c4e389097d02557bc804f
	2018-04-01	dd8382b26e7cf0599f2991017c1db58b0d2e20fb266e3ee24546bd7e647b77e8
	2018-03-31	89db48ad6a52300012680f9777d70f27b7e08fcaeb079052e56834cdab956a3a
	2018-03-31	34c0969b0bc55fb17e71622c5a3c2f76c72f3bfdc538e81378d7283366d8820d
	2018-03-31	36eae40beb944b63474b3bd2f64267fe07b922852d17a2a20e4cc9f69bc39867
	2018-03-29	fc907e358077e6a1ed66e30258972b8bf470501e5fac28dfc8b0647f4d3abf5a
	2018-03-29	682223355d783ba4ef762e42d687a0ac8e3f1d8d941db1525ca77c3ea02bf5f3
	2018-03-30	93add6b50434284c05b7a7f851fef88f532f8dc6d5b3873fe9d5c3f3d16368f9
	2018-03-30	b8031076255877534cf4aae0a2e24ec13c0b125cf7211891597e7f5794a3c0a5
	2018-03-30	47fcb5dc49597748b75d1948ef1c69747c5727e37f86a8608255f7fd493c6ca7
	2018-03-28	2ffde9807f48f04a2938ab2227e99cd0fb4c7a3420e0b06af4ff9fcdfb1d8df
	2018-03-05	a6b4d155478f96d76f65641031956e3bf4e9247f36aeefba770e0ed6ac82551b
	2018-03-20	f3a8997d63f581353747ec646d318c7121f2faca691568e80080db1b40a495ca
	2018-03-20	5f17f8e0c7aa953943bcd3bcf03e2f370175f6c9001edf24c15530c38c56cea9
	2018-03-14	fc258e4465235bcf7653f30975d6d55e8f0e598503cb5db6be825fec67c2e60c
	2018-03-02	9bce9a86b02c0d913266619d37ea4ec9c924c561546541bc2633a6cc5dfa9106
	2018-03-22	f88b802a18afe12abcd3fe17ef5bf3c07411be81c3e607fe9d73dd9749970b32
	2018-03-04	88d3ce872b9c783ba8fe017224cdddd18fe5acd8bba2a7bfe2e40c232b4eb3e97
	2018-03-02	f3cf55ef2157a63fe876232a088b9578e21e5d8fb7af640e159da1b7f1e35c08
2018-03-01	04b7adfe7a8c6d077c85842119e8e3db0055fbf92f9f6f8b439319fdef270c94	
Jigsaw	2016-04-11	3ae96f73d805e1d3995253db4d910300d8442ea603737a1428b613061e7f61e7
	2017-12-05	8f05a287d371a8004da1465978845270c4db0e6886ef039608d82a3634690f1b
	2017-05-11	2ffd0e766c1c9aeae1b3d5d6219285da413b8c66f47789e69f776ccce7a803e0
	2017-06-01	b04e467cbe144933a3499c4d95a67dfe84562f1d6a384424fbb30bc5b9c6aa22
	2018-02-17	d55901d34165364b3f244ae2bdc040b06c4ca374a23ea890d111cea7c705880b
	2017-04-08	319124b70acd0d324712edbfadb4f571b66b98a8382e54baa96d81f63b61669f
Petya	2016-03-23	26b4699a7b9eeb16e76305d843d4ab05e94d43f3201436927e13b3ebafa90739

	2018-04-29	32260615480157b55c535958672f8e7f08039b3fbee12b6b0cf8d22cd1ec660e
	2016-04-01	4c1dc737915d76b7ce579abddaba74ead6fdb5b519a1ea45308b8c49b950655c
	2018-04-29	2137507a74c13ed1b1187a884998c2f6e7ecb99b51afea7517deab8c734e521f
	2018-04-29	4c739cd5d2b28f22b46ec0c7b979b7bda24f311652dd284dcf3b07f1a9ac6018
	2017-10-25	afac6bacc04b39e13e4667dac8ebdd82328cb735d14902afc84c5a366a8d6f2
	2017-06-27	eae9771e2eeb7ea3c6059485da39e77b8c0c369232f01334954fbac1c186c998
	2017-06-27	02ef73bd2458627ed7b397ec26ee2de2e92c71a0e7588f78734761d8edbdcd9f
WannaCry	2017-05-12	ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
	2017-12-13	f60d07422a50e62bf3b92ab406b83b26fcd203a1e77ae2a9a6c239766f65eb8f
	2018-04-29	e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
	2017-11-14	55504677f82981962d85495231695d3a92aa0b31ec35a957bd9cbbef618658e3
	2018-04-29	4299a8c01063bae721295dc8f5594c0333146fbc3cc7a992ea28398360c03ad
	2018-04-13	3829d01bbb8a0187ca4d01271b2d61aa294780590e049e5175c182a0ed4f788
	2018-03-10	94cce9146c9a203b351105cfa453166e7ef7548116e0756d9ff314f55201fd55
	2017-07-25	8f57eccd5fe87253f8b4ef67c713137aa2d80bc5d73c45fe3c8734328eab432d
	2017-07-14	d9e931bdc6282dab76d6794993e5a473a360e473730a2cedf63d770db03ebdb0
	2017-07-24	b9df76897340af196874f7ec7e43592909c82046068bc84bf76bd3cad2c2f602