An Empirical Analysis of Crypto-Ransomware Behavior

Jasmeen Kaur[1], Fehmi Jaafar[1, 2], Pavol Zavarsky[1]
[1] Department of Information System Security Management, Concordia University of Edmonton, Alberta, Canada
[2] Computer Research Institute of Montreal
Email: jkaur3@student.concordia.ab.ca, {fehmi.jaafar, pavol.zavarsky}@concordia.ab.ca

Abstract— **Crypto-ransomware is a common type of malware that exploits software vulnerabilities of Internet accessible servers, end-user computers, and mobile devices. In this paper, the behavior of crypto-ransomware is empirically analyzed. We performed dynamic analysis of the ransomware in a virtual environment and the behavior of the malware represented using the data flow modeling approach. Modification of registry values and system call functions by the malware were within the scope of the analysis. The outcome of the empirical study provides a number of indicators that can be considered when assessing the effectiveness of solutions designed to prevent and detect crypto-ransomware.**

Keywords— Crypto-ransomware; Malware; Windows Operating System; Security Vulnerability; Execution flow.

I. INTRODUCTION

Ransomware is a malware that restricts the users from using their systems either by encrypting their system or by locking it, and to restore their functionality, attackers ask for ransom in bitcoins. What makes this malware different from traditional malware is its strong encryption. Indeed, crypto-ransomware uses strong encryption (like the Advanced Encryption Standard for 256 bit AES-256) and the decryption key is only provided after the ransom is paid [1]. Unlike other malware, ransomware immediately notifies the victim about the attack and demands ransom in crypto-currency. Ransomware payload is mainly spread by email, exploit kits, drive-by-download, social media, USB sticks, and security exploits in software.

A number of variants can be observed in the past years with different functionalities and features, which are used to exploit a maximum number of users. This study provides in-depth details of crypto-ransomware on the Windows platform, by analyzing registry activity, processes included, and generic flow of information during the attack. Recent variants of crypto-ransomware include CryptoWall, Cryptolocker, Lambda-Locker, and WannaCryptor [2]. Some of the new variants of ransomware target the Master Boot Record (MBR) of the system (a special type of boot sector at the very beginning of partitioned computer mass storage devices that holds the information on how the logical partitions, containing file systems, are organized on that medium). Ransomware is programmed in JavaScript, Hypertext Preprocessor (PHP), and PowerShell or Python. New variants of ransomware use different vulnerabilities to attack the victim like outdated versions of Flash Player. Ransomware uses exploit kits, such as the Angler Exploit Kit, to exploit vulnerabilities [2]. A large-scale ransomware attack took place on May 12, 2017, where the variant WannaCryptor exploited more than 200,000 systems. Ransomware attacks result in huge breaches of security, confidentiality, availability, and integrity of information. Our study shows the behavior of new crypto-ransomware variants by analyzing registry keys and system calls. Our purpose is to acquire a better understanding of the attack process of ransomware on Windows operating systems**.** By giving technical details of ransomware behavior, this study provides in-depth knowledge useful to improve mitigation and prevention methods now.

The objectives of this study are as follows:

- Analyze the behavior of ransomware on Windows Operating systems during an attack with different methods instead of using a traditional sandbox.
- Analyze the modifications made by ransomware during an attack to understand the purpose of each attack channel.
- Track the information flow during the attack.
- Suggest recommendations to improve current security mechanisms against ransomware and mitigate the risk of ransomware.

We present in this paper an empirical study of crypto-ransomware's behavior by using different real-time monitoring tools. To obtain the results, real-time attacks on target machine were performed in a virtual environment and observations were made accordingly. Furthermore, the flow of execution of malware was studied and generalized.

Section II describes related work done on ransomware. Empirical setup and methodology used in performing the crypto-ransomware behavior analysis are discussed in Section III. In Section IV, we explain the behavior of crypto-ransomware and its modification in specific operating system's files while performing an attack on the victim's machine. Dynamic-link library (DLLs) executed by crypto-ransomware are briefly discussed in section V. In Section VI, we show an execution flow of crypto-ransomware in High-Level architecture. From our empirical study, we observed some indicators of compromise. Hence, some recommendations based on results are presented in Section VII, which are expected to improve the current prevention and detection of crypto-ransomware Section VIII provides additional discussion of our findings. Finally, we conclude the paper and present future work in Section IX.

II. RELATED WORK

M. Choudhary et al. [3] discussed and analyzed the different variants from different families of ransomware and defined the characteristics of ransomware evolution. They analyzed samples on two major platforms, i.e. Windows and Android. The analysis was mainly based on monitoring the file system and registry activities by using tools like Cuckoo Sandbox on Windows and Anubis and Andrubis for Android. We expand their research by analyzing the behavior of new variants and combining the approach of dynamic and static analysis.

Sudhir Kumar Pandey and B. M. Methre [4] discussed three malware detection approaches, i.e. signature-based, anomaly-based, and specification-based. Malware analysis techniques included static malware analysis, string analysis and dynamic malware analysis. Static analysis uses a large database of already known suspicious codes, file signatures, and behavior of malware. In string analysis, the malware analyst tries to look for the malware specimen's name, user dialogue, password for backdoors, URLs, attacker's email address, libraries, different function calls, and processes. For dynamic malware analysis, a run time analysis is performed. In fact, the authors introduced a 12-stage lifecycle to analyze behavior of malware on run time that includes network surveillance and command-and-control (C&C are centralized machines that are able to send commands and receive outputs of machines part of a botnet) servers communication and peer coordination.

Scaife et al. [5] introduced an approach called CryptoDrop which focuses on monitoring user's data instead of monitoring each potential malicious software. This program analyzes user's data for any modifications and assigns threshold points to the process with any modifications. Researchers assigned the threshold to all the processes, and when any process reached a specific level of the assigned threshold, it was considered malicious and would be terminated. CryptoDrop has three primary and two secondary indicators. The three primary indicators are file type changes, similarity measurement, and Shannon entropy. Secondary indicators include deletion and file type funneling [5]. However, while analyzing all the files, this approach's performance is reduced. In addition, ransomware may include some newer files to encrypt to its list, which might not be present in the CryptoDrop database.

III. METHODOLOGY

In this section, we describe the empirical setup and tools used to carry out the empirical study. All empirical study steps were performed in a virtual environment. The objective of this study is to describe the behavior of new ransomware variants such as WannaCryptor Ransomware. Another purpose of this study is to show an attack's execution flow based on registry keys and system function. To gather this information, a virtual environment setup was developed in VMware Workstation 11.1.0 on a host machine Windows 7 Professional, 64bit. In VMware Workstation, we installed Windows 10, 64 bits which acted as the target machine. We configured the network interface for Windows 10 to HOST ONLY in order to avoid malware spreading to the host's operating system. Also, Host System was secured and protected from infection by enabling the firewall and Windows Defender. In order to monitor the infection and malware activities, we used Process Monitor, a real-time system monitoring tool installed on Windows 10. We also used some additional tools like Regshot and Wireshark to monitor the malware activities. We then collected the malicious executable files of ransomware of various families. Most of the ransomware samples were collected from VirusTotal and ViruShare malware repositories. Then, we executed malicious samples and performed a real-time attack on Windows 10. We placed some random document, images and .rar files on Windows 10 and disabled the Windows Defender and Firewall on the targeted machine to successfully execute the empirical study and monitor the behavior of crypto-ransomware.

IV. BEHAVIOR OF RANSOMWARE

This section is dedicated to explaining the behavior of ransomware when attacking the victim's machine. Ransomware encrypts the user's system using encrypting algorithms like AES, Rivest–Shamir–Adleman (RSA), or Rivest Cipher 4 (RC4) [18]. Then, it asks for a ransom in bitcoins. The main families analyzed in this study are the following: Sage Ransomware, WannaCryptor, Lambda-Locker, Hydra-Crypt, CryptoWall, and SamSam. Most of the ransomware samples exhibit similar general behavior when manipulating the Windows operating system with some different aspects. The ransomware has a similar flow of execution when infecting a system: an executable file via different delivery methods is launched into the target system during the execution, and it creates various legitimate sub-processes. Then, it communicates with Command and Control server by sending a POST request. Ransomware modifies various registry keys and executes various DLLs, which serve different purposes accordingly. After encrypting the victim's system, the ransomware deletes its executable file and shows the ransom note asking for a ransom. The ransomware mainly uses spam emails, advertisements, and vulnerability exploitation in network or applications as a delivery method. This malware can hide on the user's system for some time before execution and then start the encryption process at a very fast pace. The functionality of ransomware is explained in four phases which include: 1) the setup phase describes a number of modifications executed by a malware to set itself in the victim's system; 2) the communication phase describes the communication held during the attack to receive the encryption key; in addition, this phase identifies network related artifacts; 3) the encryption phase highlights the information regarding the encryption used in the ransomware attack; and 4) the last phase is the deletion of volume shadow copies of Windows.

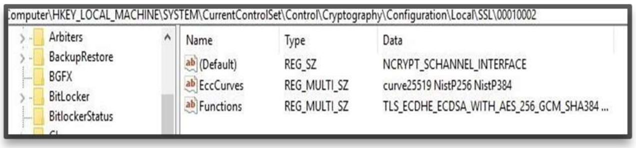
A. Setup Phase

The ransomware creates some persistence keys on the victim's system to bypass a reboot. These files and keys are deleted when encryption is completed. The original executable malware creates a copy of itself and deletes the original file. Initially, the malware creates some files in the prefetch folder, which is usually created when an application runs for the first time from any location in Windows. The prefetch folder contains files used in loading the program. The path of the folder contains the file name and a hash of 8 characters added to give the malware file a unique ID: C:\Windows\Prefetch\filename-Hashvalue.pf. We observed that the malware was executed from this path.

Then, another registry value of Windows is parsed, to verify whether the system isrunning in a mode compatible to the application or not and check whether the basic functions of Terminal Services are enabled or not. Basically, in Windows operating systems, Terminal Services Configuration applications determine which user can perform actions like connect and disconnect to Terminal Service session.

After this step, the ransomware deletes the safeboot option to prevent the user from restarting the system in safe mode by reparsing the HKLM\System\CurrentControl registry key (HKEY_LOCAL_MACHINE, often abbreviated as HKLM, is one of several registry hives that make up the Windows Registry). We notified that in for all analysed ransom-ware analysed in our empirical study, they modify a registry value to change log on (HKLM\SOFTWARE\Microsoft\ WindowsNT\CurrentVersion\WinLogon). In fact, this registry value is responsible for user log on and log off. Malware changes it to malware path and filename so that it is executed at startup.

After making these changes, and before starting the encryption process on the victim's system, the ransomware collects the victim's system information like computer name, operating system, Digital Product Id, and System BIOS data. Then, it creates a hash of this information by reading the registry value (saved on HKLM\System\CurrentControlSet\Control\Computer\Name \ActiveComputerName\Compute_Name). To further the processing of crypto ransomware, a window appears on the victim's system and will keep popping up until the victim selects "yes". It also called the default cryptographic function of Windows, which is HKLM\SYSTEM \CurrentControlSet\Control\Cryptography\Configuration\Local\SSL\0002 and added a new value as shown in figure 1.



omputer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Cryptography\Configuration\Local\SSL\00010002

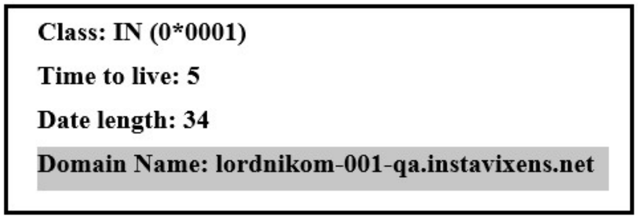
Name	Type	Data
(Default)	REG_SZ	NCRYPT_SCHANNEL_INTERFACE
EccCurves	REG_MULTI_SZ	curve25519 NistP256 NistP384
Functions	REG_MULTI_SZ	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 ...

Figure 1. New value added in the Cryptography Registry key

Then, the crypto ransomware uses the Image File Execution Options (IFEO) to check if there is any active debugger. Malware also verifies whether it can attach itself to another executable like explorer.exe or svchost.exe.

B. *Communication Phase*

The ransomware tries to keep a real-time connection with the malware authors through domain and C&C Server. The malware generates a POST request to send the victim's system information to the attacker and ask to generate the domains. A UDP-based request from the victim's system is sent to thousands of Hosts. Certain packets of traffic during the attack are encrypted with RC4. In case of CryptoWall, the malware sends the POST request to the servers to get an onion address and public key to encrypt the user's system. In Figure 2, we are showing the connection attempt of a ransomware to the domain.



Class: IN (0*0001)

Time to live: 5

Date length: 34

Domain Name: lordnikom-001-qa.instavixens.net

Figure 2. Example of crypto ransomware trying to connect to the domain

The ransomware creates the registry key HKLM\SYSTEM\Current_ControlSet\Control\NetworkProvider\HwOrder to check the list of network providers. While some crypto ransomware variants use HTTP protocol, another set of them uses The Onion Router Software, (TOR) which is hard to trace as it is enabling anonymous communication. New strains of ransomware avoid using C&C servers for communication, so blocking the outbound communication does not help stopping the attack.

C. *Encryption Phase*

Encryption is the critical factor, which makes ransomware different from other malware and hard to defeat. After making a successful connection with the victim's system, the encryption phase starts. In some cases, the encryption key is generated on the victim's system. Attackers could send the key through the C&C server, but they do not share the private key. The encryption used in most of the ransomware variants is AES, RSA 2048, SHA 256, RSA-AES, which is a level 2 encryption. The ransomware uses ECDH and the Domain generation algorithms (DGA). Indeed, the Elliptic-curve Diffie–Hellman (ECDH) is an anonymous key agreement protocol that allows two parties, each having an elliptic-curve public–private key pair, to establish a shared secret over an insecure channel [20]. The DGA are algorithms seen in various families of malware that are used to periodically generate a large number of domain names that can be used as rendezvous points with their command and control servers [19]. Most of the ransomware variants add their name as an extension to every file. Ransomware payload contains a list of the files to encrypt but skips some folders like "WINDOWS", "Program Files" and "Temp" to keep the Windows System working in normal conditions. For decryption purposes, the malware keeps information about the files, like file name, size, etc. After encrypting the files, some Helper files on the victim's system are created to notify the user of the attack with instructions to pay the ransom. Most of the crypto-ransomware created files for decrypt instructions usually have the format of HTML and Text file.

D. Deletion of Volume Shadow Copies

The ransomware prevents the user from restoring the volume shadow copies by deleting them. It gains administrator rights and calls the cmd.exe to delete the Windows volume shadow copies by command: vssadmin delete shadow/all/quiet. By using this command, it deletes all the shadow copies taken by Windows without the user's knowledge. Some ransomware executes: wbadmin delete catalog-quiet to delete the shadow copies. To trace the VSS backup, a registry value was queried. Then, the ransomware changes the Windows policies depending on their functionality. For example, we observed that crypto-ransomware takes all the permissions including special permissions. In the case of WannaCryptor, system permissions are modified by using command: icacls./grantEveryone :F/T/C/Q and grant the access to Everyone. The ransomware also changes the VAD MEMORY protections to PAGEEXECUETE|PAGE_NOCACHE instead of PAGE_EXECUTE_WRITECOPY.

V. DLLs CALLED BY RANSOMWARE

A dynamic link library, DLL, is a library that contains code and data that can be used by more than one program at the same time in order to promote code reuse and efficient memory usage. During the attack, the ransomware malware uses a number of DLLs for various purposes. Some of the DLLs are generally used by other executable applications that serve a basic purpose for Windows, like Kernal32.dll, ntdll.dll, user32.dll, KernalBase.dll, python27.dll. Other major DLLs called during the malware execution were msvcrt.dll, 4ernel.appcore.dll, ws2_32.dll, Powrprof.dll, SecRuntime.dll, atl.dll, usermgrcli.dll. DLLs executed by WannaCryptor were sspicli.dll, ucrtbase.dll, rpcrt4.dll, rsaenh.dll, ntmarta.dll, uxtheme.dll, windows.storage.dll, msvcp-win.dll. It used SysWOW64 to call chkdsk.exe to check volume of disk, sector information and display the status of the drive. DLLs related to deleting the volume shadow copies are vssapi.dll and vsstrace.dll and they have some static linked DLLs, which are srcore.dll, spp.dll iasdatastore.dll.

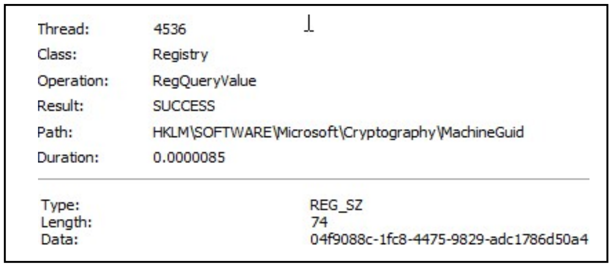
Table I. DLLS CALLED BY CRYPTO-RANSOMWARE

DLL called	Purposed served by DLL called
Kernal32.dll	Used to manage process at kernel level
Mswsock.dll	Called by ransomware to manipulate another program
Urlmon.dll	Used to check the network connection
Ntdll.dll	Used to access the kernel mode from the user mode
Perfo.dll	Used to check the performance of new processes created by the malware

VI. RANSOMWARE EXECUTION FLOW

The ransomware payload is launched into the victim's system in the form of an executable. Crypto-ransomware has the code to check the availability of sandbox, any debugger or any detection technique. There are a number of processes that are called during the ransomware execution. At the execution stage, the ransomware generates a unique victim ID and key and saves it in the ".tmp" folder. When an executable is executed, it creates a legitimate process like svchost.exe or explorer.exe. In most of the variants like CryptoWall and WannaCryptor, a batch file and a copy of the original file are created. Then, the ransomware deletes itself by running a batch script.

Along with deleting the original file, the completion of the installation of a newly created copy is ensured by performing a ping request to the localhost (127.0.0.1). The malware executes conhost.exe to create multiple threads and a process with a ransom string name. Subsequently, a newly created thread under the parent process is scheduled ONLOGON, that is, even if the system restarts or any user logs on the system, this process will be executed at the run level HIGHEST. Along with ensuring its persistence, the ransomware communicates with C&C servers or generates DGA to look for the domain and dynamically change C&C servers to connect with a number of URLs. After making a successful connection, the malware attacker sends back a unique ID for each victim with the encryption key and TOR information which contains the URL for payment. WannaCryptor used TaskData\Tor\Tasksvc.exe to download the TOR information and extract this information into the Taskdata folder. Then, it creates a number of threads to change the file and directory attribute. In addition, the ransomware executed tasksche.exe to copy itself. Some of the variants executed cscript.exe to run a script from a Windows script with command "Cscript.exe//nologo .m.v". Malware collected Windows globally unique identifier (GUID), a 128-bit number used to identify information in computer systems, by querying Windows registry HKLM\SOFTWARE_Microsoft\Cryptography\Machine_Guid. This GUID is used by the malware author to know the Global Unique Identifier of the victim's system. In Figure 3, we show an example of the GUID of a system queried by a crypto-ransomware.



```
Thread:       4536
Class:        Registry
Operation:    RegQueryValue
Result:       SUCCESS
Path:         HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid
Duration:     0.0000085

Type:                     REG_SZ
Length:                   74
Data:                     04f9088c-1fc8-4475-9829-adc1786d50a4
```

Figure 3. Example of GUID of system queried by crypto-ransomware

The ransomware notifies the victim of the attack by showing a ransom note. To manipulate the desktop wallpaper and desktop icons, the ransomware modifies the registry key HKCU\Control Panel\Desktop and performs the operation *SetInfoKey* which results in a change of the desktop wallpaper to the ransomware notification and a ransom note. Ransomware is different from other malware in that it notifies the victim of the attack and demands an amount to free the compromised system. To inform the victim of the attack, Helper Files are created within different folders of the victim's system. In fact, these Helper Files contain instructions regarding the procedure to access the onion links and pay the ransom.

After performing the whole encryption on files and folders, the ransomware deletes all its persistence keys, files and executables by calling cmd.exe, to perform an operation *Delete On Close,* which will delete the copy of the ransomware executable when it is closed.

A. In Case of Worm Functionality

Ransomware like WannaCryptor has the ability to compromise the network by scanning a number of systems in the same network. After compromising a single machine, malware attackers remotely look for other systems in the same network and deploys the payload in the network. Indeed, WannaCryptor ransomware executes its payload, creates sub processes, and runs "attrib.exe" by command attrib +h where it "hides" the directory wherever it is placed. To scan the network, it calls the GetAdapterinfo function to determine the number of IP addresses and subnet masks of the compromised system's network. This variant tries to connect with a number of domains by using API InternetOpenUrlA(). If the connection with any of the domains is successful, then payload will not execute. Otherwise, it will run the malicious payload and search for other vulnerable systems in the network on the domain.

B. New Trends of Ransomware

Many new variants of ransomware have been observed recently [6]-[10]. One of the variants offers decryption only if the victim infects two more victims and pays the ransom. Another variant performs the chkdsk, encrypts the hard disk and asks for ransom; if the victim pays, it reboots the system and encrypts the file and the victim has to pay the ransom twice. In the case of another variant called Spora, once it attacks the victim, it offers immunity from further attack. New strains of ransomware like WannaCryptor are capable of compromising numerous systems over a network, servers and databases.

Another new variant called SOREBRECT with a fileless feature is also capable of compromising a whole network [8]. This variant deletes the system's logs to delete all traces. Malicious Code utilizes Microsoft's SysInternal PsExec command-line utility for encryption, which allows the attacker to run malicious activities remotely and eliminate the need of an interactive login session or manual delivery of the malware payload into the remote machine [8]. It was also observed that the detection of this ransomware variant was only possible by analyzing abnormal behavior in the network, the RAM, and the Registry of the system.

VII. RECOMMENDATIONS

During the empirical study, most crypto-ransomware variants caused the same changes. Based on our empirical stdy and previous related work [11]-[14], we propose the following recommendations to mitigate ransomware risks:

1. It is recommended to set up a periodic and continuing strategy for data backup.
2. User awareness can reduce the risks of these attacks as not examining URLs and file extensions before opening a file is one of the common reasons of information system infections [16].
3. It was observed during the empirical study that most of crypto-ransomware variants tend to delete the Windows Volume shadow copies and any Windows backup. Therefore, any executable should not be allowed to access shadow copies of Windows and any process that tries to delete these shadow copies should be terminated.
4. Most crypto-ransomware like WannaCryptor pop up User Account Control (UAC) windows until YES is selected to gain access. Any executable with such behavior should be terminated. This signature does not apply to all the variants as some of the variants bypass this step.
5. Most of the crypto-ransomware variants require administrative privilege to make changes in the system which are not allowed to regular users. No application should be allowed to gain full admin access; if it gains full administrative access and performs more write operation than regular threshold, it should be terminated. We observed during our empirical study more than 4000 writes and 800 reads from a single ransomware sample. Thus, write permissions and other administrative privileges for regular users should be limited.
6. There was a set of specific locations in Windows like .temp folder and tasks in C:\Windows\SysWow64 and C:Windows\System32, where ransomware tried to modify the contents. Thus, these locations in Windows should be placed under scrutiny to detect any malicious executable or malicious code.
7. There was a set of specific registry keys modified by ransomware. For example, a ransomware tends to alter SAFEBOOT option, to prevent booting the system in SAFE-MODE and avoid its deletion. A verification regarding alteration to SAFE-MODE by an executable should be made and if found, that executable should be terminated.
8. To prevent ransomware from spreading into a network and causing further damage, a system which is compromised by malware should be dis-connected immediately from the network. The victim should never pay the ransom, as it encourages the attacker to target more victims.

VIII. DISCUSSION

This section provides the post-experiment outcomes and related discussion. Indeed, crypto-ransomware has diverse impacts on the system registry and registry key values. A number of registries were modified and operated by the malware to ensure its persistence in the system. The ransomware deleted all backup and restoration points. There was a number of processes which were spawned during the attack process. Crypto-ransomware spreads to several legitimate processes of Windows to successfully invade the infected system. With the study of this malware, we were able to suggest a number of corresponding prevention recommendations as attackers always find a way to bypass defense mechanisms [15]. Unpatched vulnerabilities are the most common and easiest way for an attacker to deliver the payload, but a naive user is the best method to compromise the system. Crypto-ransomware disguises itself in any Windows folder-like “Program File” and “Temp”, which should be constantly monitored. Thus, on the one hand, the user should not open any suspicious link. On the other hand, all vulnerabilities should be patched in order to prevent damage by crypto-ransomware.

IX. CONCLUSION

In this paper, we examined a set of crypto-ransomware in order to analyze the common behavior between them. We observed activities of the ransomware on a Windows system by performing a real-time attack in a virtual environment. Our results show different aspects of ransomware infecting the system. All the variants of crypto-ransomware affected the same registry values and deleted existing files. It was also observed that once the attack was completed, no encryption was performed on newly added files. Crypto-ransomware tends to perform more write operations when compared to a system’s normal behavior. Based on our observations, we propose a set of recommendations to improve the current detection and prevention methods. Ransomware uses very strong encryption to attack, which is in general impossible to crack. Therefore, it is always recommended to protect the system in the first place. All variants of crypto-ransomware showed some typical signatures. In future work, we are proposing to use such signatures to enhance a new detection and prevention ransomware approach.

ACKNOWLEDGMENT

This work has been partly funded by a Public Safety Canada grant, and by the Computer Research Institute of Montréal (CRIM). We gratefully thank the Ministère de l’Économie, Science et Innovation du Québec (MESI) for its generous support.

REFERENCES

[1] F. Cuppens, N. Cuppens, J. Lanet, and L. Legay, “Risks and Security of Internet and Systems”, 11th International Conference, CRiSIS 2016, Roscoff, France, pp. 5-7, 2017.

[2] N. Scaife, P. Traynor, and K. Butler, “Making Sense of the Ransomware Mess (and Planning a Sensible Path Forward)”, IEEE Potentials, 36(6), pp. 28-31, 2017.

[3] M. Choudhary, P. Zavarsky, and D. Lindskog, “Experminetal Analysis of Ransomware on Windows and Android Platform: Evolution and Characterization”, Procedia Computer Science, vol. 94, no. ISSN 1877- 0509, pp. 456-472, 2016.

[4] S. K. Pandey and B. M. Methre, "A Lifecycle Based Approach for Malware Analysis", Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on. IEEE, 2014.

[5] N. Scaife, H. Carter, P. Traynor, and K. R. Butler, "CryptoLock (and Drop it): Stopping ransomware attack on user data", in Distributd Computing systems (ICDCS), 36th International Conference on. IEEE, 2016.

[6] A. Zahra and M.A. Shah, “IoT based ransomware growth rate evaluation and detection using command and control blacklisting”, 23rd International Conference on Automation and Computing (ICAC), pp. 1-6, 2017.

[7] D. Maiorca, F. Mercaldo, G. Giacinto, C. A. Visaggio, and F. Martinelli, “R-PackDroid: API package-based characterization and detection of mobile ransomware”, In Proceedings of the Symposium on Applied Computing, pp. 1718-1723, 2017.

[8] A. L. Young and M. Yung, “Cryptovirology: The birth, neglect, and explosion of ransomware”, Communications of the ACM, 60 (7), pp. 24-26, 2017.

[9] C. Chung and N. Tan, "The Evolution of Ransomware From CryptoWall To CTB LOCKER", in Virus Bullentin Conference, Fortinet Techonlogy Inc., pp. 46-56, 2015.

[10] D. Bazdarevic and M. Dubell “Building ransomware for fun and profit academic research purposes”, 2016.

[11] FireEye, "Ransomware Response Strategies From Identifying attack mechanisms to Detecting and blocking threats", FireEye.Inc, California, 2016.

[12] A. Ajjan and James Wyke, "The Current State of ransomware”, A Sophos Labs Technical paper, pp. 1-61, 2016.

[13]A. Ali, “Ransomware: a research and a personal case study of dealing with this nasty malware”, Issues in Informing Science and Information Technology, 14, pp. 87-99, 2017.

[14] A. Sanatinia and G. Noubir, "OnionBots: Subverting Privacy Infrastructure for Cyber Attacks", in 45th Annual IEEE\IFIP International Conference in Dependable System and Networks, pp. 69-80, 2015.

[15] E. Kirda, “UNVEIL: A large-scale, automated approach to detecting ransomware”, In The IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 1-2, 2017.

[16] X. Luo, and Q. Liao, “Awareness education as the key to ransomware prevention”, Information Systems Security, 16 (4), pp. 195-202, 2007.

[17] C. Everett, “Ransomware: to pay or not to pay?”, Computer Fraud & Security, (4), pp.8-12, 2017.

[18] R. Rivest, ‘”SA Security response to weaknesses in key scheduling algorithm of RC4”. Technical note, RSA Data Security, Inc., pp.1-3, 2001.

[19] D. Gonzalez and T. Hayajneh, “Detection and prevention of crypto-ransomware”, In Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), IEEE 8th Annual, pp. 472-478, 2017.

[20] N. Koblitz, “Elliptic curve cryptosystems”. Mathematics of computation, 48(177), pp. 203-209, 1987.