An Integrated Approach for Detecting Ransomware Using Static and Dynamic Analysis

Daniel F Netto
Department of Computer Science and Engineering
ER & DCI Institute of Technology
Trivandrum, India
danielfnetto92@gmail.com

Shony K M
Department of Computer Science and Engineering
ER & DCI Institute of Technology
Trivandrum, India
shonykm@gmail.com

Elizabeth Rose Lalson
Department of Computer Science and Engineering
ER & DCI Institute of Technology
Trivandrum, India
elizabeth@cdac.in

Abstract— **Ransomware has been a rising threat in the past few years. By analyzing the number of systems that are infected by ransomware and the amount of ransom paid by the user, the damage done to an IT infrastructure is measured. There are many deployment methods in which the ransomware get into the system. These are similar to malware and ransomware is a form of malware whose central behavior is to prevent legitimate users from accessing the resources when needed. This paper discusses the static and dynamic techniques that can be used to detect and prevent a ransomware attack.**

Keywords—dynamic analysis, honey traps, ransomware, ransomware detection, static analysis

I. Introduction

The past few years we have seen the rise of ransomware on the Internet. Ransomware is a form of malware which exhibits the behavior of preventing legitimate users from accessing the necessary file or system resources when required. This indicates a clear violation of the Availability principle of the CIA triad of the security. Ransomware can thus be defined as, "A type of malware that prevents or limits users from accessing their system, either by locking the system's screen or by locking the users' files unless a ransom is paid." [1].

Ransomware existed decades ago, but the term was coined recently. The first early known ransomware was known as the trojan called AIDS Trojan [2]. Joseph Popp developed this malware and circulated it among AIDS researchers through a company called PC Cyborg. This ransomware encrypted the hard disk contents after a specific number of reboots and demanded the users to pay a license fee to get the decryption key.

With technology like Bitcoin, which favors secret payments, attacker started using it to receive ransoms for their application. With strong cryptographic advancement, attackers started using those in their malicious code, so that security researchers are not able to quickly decrypt the affected files. In the deep web, some malicious actors provide Ransomware as a Service (RaaS), so that a newbie can deploy the malicious code without much effort quickly.

This paper gives a brief idea of how ransomware gets deployed and the standard type of filesystem that gets targeted. It also discusses the static analysis method like honey trap techniques and network detection and also a dynamic technique which includes analyzing the running process periodically to detect the ransomware activity on the system. The paper also highlights the analysis done on the proposed model which was implemented.

II. Literature Survey

The deployment of ransomware is similar to the deployment of a malware. Some of the most commonly noticed methods of deployment [3] are as follows

- Email attachments
- Malvertisements
- Botnets
- Ransomware as a Service

Ransomware like Locky encrypts file with certain extensions [4] like .mid, .wma, .flv, .mkv, .mov, .avi, .asf, .mpeg, .vob, .mpg, .wmv, .fla, .swf, .wav, .qcow2, .vdi, .vmdk, .vmx, .gpg, .aes, .ARC, .PAQ, .tar.bz2, .tbk, .bak, .tar, .tgz, .rar, .zip, .djv, .djvu, .svg, .bmp, .png, .gif, .raw, .cgm, .jpeg, .jpg, .tif, .tiff, .NEF, .psd, .cmd, .bat, .class, .jar, .java, .asp, .brd, .sch, .dch, .dip, .vbs, .asm, .pas, .cpp, .php, .ldf, .mdf, .ibd, .MYI, .MYD, .frm, .odb, .dbf, .mdb, .sql, .SQLITEDB, .SQLITE3, .asc, .lay6, .lay, .ms11 (Security copy), .sldm, .sldx, .ppsm, .ppsx, .ppam, .docb, .mml, .sxm, .otg, .odg, .uop, .potx, .potm, .pptx, .pptm, .std, .sxd, .pot, .pps, .sti, .sxi, .otp, .odp, .wks, .xltx, .xltm, .xlsx, .xlsm, .xlsb, .slk, .xlw, .xlt, .xlm, .xlc, .dif, .stc, .sxc, .ots, .ods, .hwp, .dotm, .dotx, .docm, .docx, .DOT, .max, .xml, .txt, .CSV, .uot, .RTF, .pdf, .XLS, .PPT, .stw, .sxw, .ott, .odt, .DOC, .pem, .csr, .crt, .key, wallet.dat

But not all ransomware exhibits the same behavior. The files it encrypts and the files it avoids depends on the logic of the malware coders' logic. Another noticed behavior of ransomware is that it tries to contact a Command & Control (C & C) server, to get further instruction on how to execute on the

Authorized licensed use limited to: Modern Education Society's College of Engineering. Downloaded on October 09,2023 at 05:19:34 UTC from IEEE Xplore. Restrictions apply.

system. It may download additional files or transfer the key required to encrypt the file.

Once ransomware starts execution and affects a system, the user gets to know he is affected when a popup comes up saying that the system is locked and you need to pay a ransom to get back the files. Paying the ransom doesn't guarantee that the file is retrieved. Paying the ransom to the attacker only fuels his motivation to launch more attacks. While some ransomware encrypts the file, some just display fake banners that the system is locked. These banners take up the whole screen of the desktop so that the user cannot do anything else.

The industries which are affected by ransomware range from health care, education, finance, etc. According to a blog [5], the most affected sector in 2017 was education. Ransomware attacks where commonly adopted by newbie actors for the fact that there are others who provide their code for a royalty of the ransom being paid. Also, it doesn't require much effort from the attackers since its success depends on the number of unpatched systems on the Internet.

III. Proposed Model

Based on the anti ransomware techniques exhibited by the ransomware coders, a model is built based on the static and dynamic analysis techniques. By creating a hybrid approach, the system ensures that the ransomware evasion techniques are defeated.

1. Static Techniques

Our first approach is to place decoy files in the filesystem. The number of files that are created and placed varies on different systems and user intervention. The anti-ransomware evasion techniques exhibited by the ransomware needs to be defeated; hence the decoy file generation needs to be random. We formulated an algorithm for the creation of decoy files.

```
Step 1: Start
Step 2: Create a set of alphabets, numerals, symbols
Step 3: Create a set of valid extension set
Step 4: Get a random number to determine the filename length and assign to filenameLen
Step 5: Get another random number to determine the number of characters and assign to filecontentLen
Step 6: Get another random number to determine the length of generated extension and assign to fileextLen
Step 7: Generate a random name with filenameLen and alphabet, numerals set
Step 8: Generate a random content with filecontentLen and alphabets, numerals, symbols set
Step 9: Randomly choose between valid extension set or dynamic extension set to append after the filename
Step 10: Stop
```

The user is given an option to manually specify which all areas in the filesystem need to be protected from ransomware attacks. If he doesn't specify any paths, the application automatically enumerates paths from the filesystem and randomly chooses paths to protect. These could range from a minimum of ten to several hundreds of path depending on the filesystem size. After the paths are identified, based on our algorithm the newly created files are placed on these paths and triggers are attached to it for detecting any file access or file modification.

The following block diagrams demonstrate the flow of the process for our proposed model in static analysis technique. It demonstrates the flow during the installation of the model and the flow of operation taking place in the system after the installation of the model.

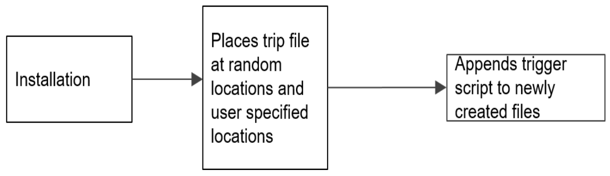

Installation
Places trip file at random locations and user specified locations
Appends trigger script to newly created files

Fig. 1: Process for setting honey trap technique.

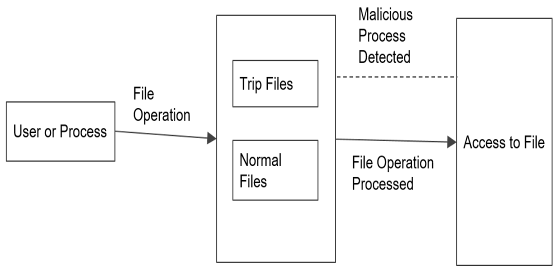


Fig. 2: Shows how process get detected using honey trap technique when deployed in a system.

Our second approach for detection is to analyze the incoming and outgoing packets in the users' system. A dataset is prepared based on the following parameters host address, destination and source ports, used filenames. Any packets that are sent or received is analyzed with the dataset. If any of the parameters match with the dataset, the packet is flagged, and the IP is blacklisted so that no future requests are received or sent from the system. The following block diagram demonstrates how network packets are analyzed for the detection of ransomware.

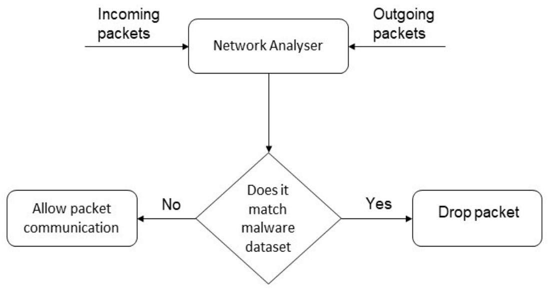


Fig. 3: Block diagram of how network analyzer module works

Authorized licensed use limited to: Modern Education Society's College of Engineering. Downloaded on October 09,2023 at 05:19:34 UTC from IEEE Xplore. Restrictions apply.

2. Dynamic Techniques

In this approach, the network packet details are analyzed in depth. We build a dataset with known hexadecimal for properties like pe_imports, reg_access, reg_read, file_write, file_read, file_drop, etc. If any properties are matched, the network packet is flagged, and IP added to the firewall to prevent further damage.

When our application is first executed on the system, a scheduled task is created to run daily to compute a threshold value for the application to detect the anomalous behavior of the process. Based on few properties of the process that is running on the system, it is checked with our threshold value. If the value is higher than the threshold value, the process is flagged and blocked from executing else continues with the execution. Another scheduled task is executed to run at fixed intervals, to record the behavior of the system activity, so that it can be later used to compute the threshold value for the start of the day. The flagged process is attempted to be removed from the system so that it cannot later cause damage again to the system.

The following block diagram shows how a process analyzer module of the dynamic analysis part works in our proposed model.

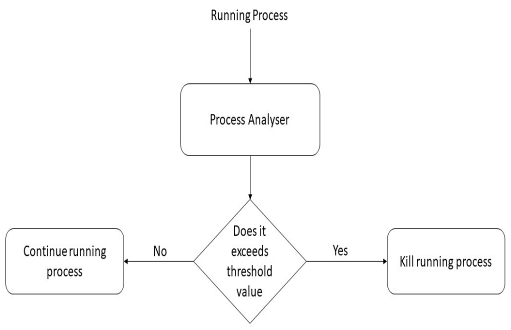


Fig. 4: Block diagram of a process analyzer block used in dynamic analysis

IV. Analysis

The proposed model was coded in C# and was tested on a Windows 10 operating system. When a user doesn't specify any file path, the system path are enumerated, and a set of a path is randomly chosen. In all these chosen paths, the generated file is stored, and triggers are attached to it. In user-centric approach, the user is given an option to choose multiple locations to place the generated files. Once this process is completed, it waits for a ransomware activity before triggering the alert and informing the user.

On an extensive file system, it was observed that the automated process of enumerating the system directory takes up much time to build the list of a randomly chosen path. In a file system of 1 Terabyte in size, the process of building the path set takes several hours before creating and placing generated files. Also, there is a probability that the randomly chosen path doesn't contain the path where the user wants his data to be safe.

In user-centric approach, the user has full control over what all path to place the protection files. Since this is a manual process, it is very much faster when compared to the automated approach. It also ensures that the user has marked all areas of the file system which are vital to him. A user can protect a single path or choose from multiple paths.

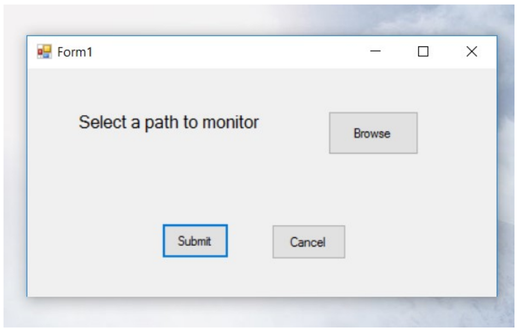
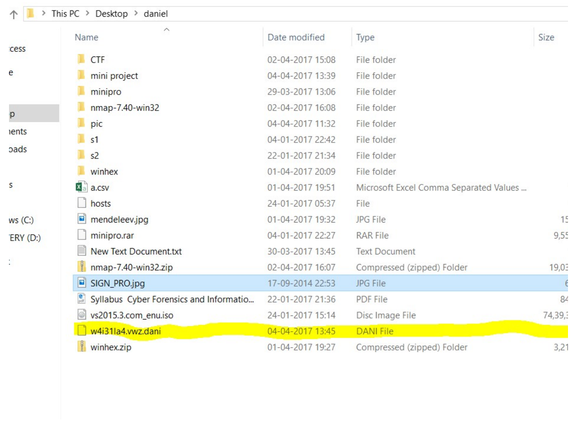


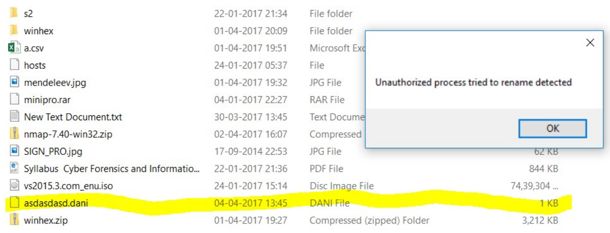
Fig. 5: Modal window is shown for a user for a single path approach



Name	Date modified	Type	Size
CTF	02-04-2017 15:08	File folder	
mini project	04-04-2017 13:39	File folder	
minipro	29-03-2017 13:06	File folder	
nmap-7.40-win32	02-04-2017 16:08	File folder	
pic	04-04-2017 11:32	File folder	
s1	04-01-2017 22:42	File folder	
s2	22-01-2017 21:34	File folder	
winhex	01-04-2017 20:09	File folder	
a.csv	01-04-2017 19:51	Microsoft Excel Comma Separated Values ...	
hosts	24-01-2017 05:37	File	
mendeleev.jpg	01-04-2017 19:32	JPG File	15
minipro.rar	04-01-2017 22:27	RAR File	9,55
New Text Document.txt	30-03-2017 13:45	Text Document	
nmap-7.40-win32.zip	02-04-2017 16:07	Compressed (zipped) Folder	19,03
SIGN_PRO.jpg	17-09-2014 22:53	JPG File	6
Syllabus Cyber Forensics and Informatio...	22-01-2017 21:36	PDF File	84
vs2015.3.com_enu.iso	24-01-2017 15:14	Disc Image File	74,39,3
w4i31la4.vwz.dani	04-04-2017 13:45	DANI File	
winhex.zip	01-04-2017 19:27	Compressed (zipped) Folder	3,21

Fig. 6: Highlighted portion shows the newly generated trigger file

Once all initial setup is completed, the application runs in monitoring mode. The system triggers an alert confirming the presence of ransomware activity if ransomware accesses the generated files. The sole application purpose is detection, and therefore there is no way to prevent the attack from completing its task. The application also doesn't support the recovery of encrypted files.

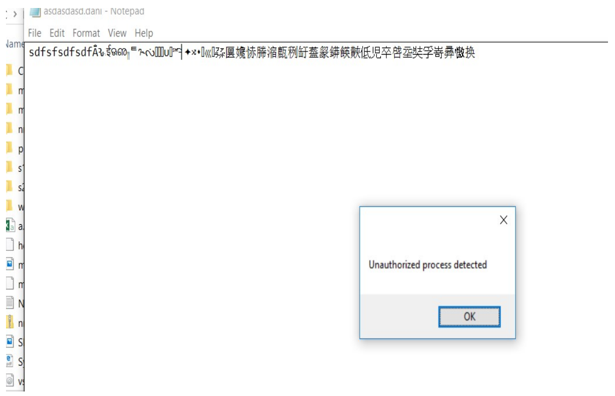


s2	22-01-2017 21:34	File folder	
winhex	01-04-2017 20:09	File folder	
a.csv	01-04-2017 19:51	Microsoft Exc	
hosts	24-01-2017 05:37	File	
mendeleev.jpg	01-04-2017 19:32	JPG File	
minipro.rar	04-01-2017 22:27	RAR File	
New Text Document.txt	30-03-2017 13:45	Text Docume	
nmap-7.40-win32.zip	02-04-2017 16:07	Compressed	
SIGN_PRO.jpg	17-09-2014 22:53	JPG File	62 KB
Syllabus Cyber Forensics and Informatio...	22-01-2017 21:36	PDF File	844 KB
vs2015.3.com_enu.iso	24-01-2017 15:14	Disc Image File	74,39,304 ...
asdasdasd.dani	04-04-2017 13:45	DANI File	1 KB
winhex.zip	01-04-2017 19:27	Compressed (zipped) Folder	3,212 KB

Unauthorized process tried to rename detected

Fig. 7: Show trigger being activated when the highlighted file was renamed

Authorized licensed use limited to: Modern Education Society's College of Engineering. Downloaded on October 09,2023 at 05:19:34 UTC from IEEE Xplore. Restrictions apply.



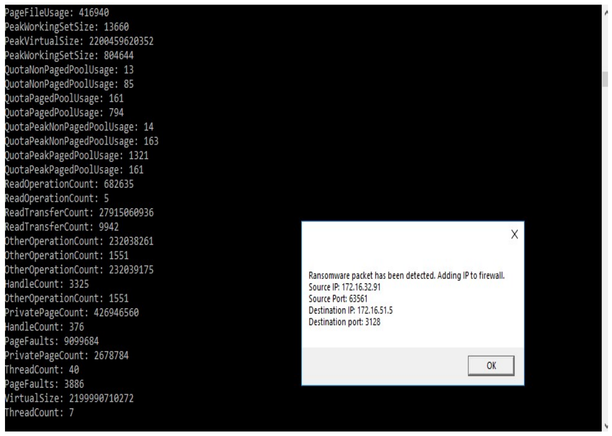
Unauthorized process detected

Fig. 8: Trigger being activated when the file was accessed

Detecting the ransomware during monitoring phase is a challenge to our proposed model. The randomly generated files are placed scattered across the file system. In some path chosen, these files are placed at the beginning, while in some they are placed at the end. In some folders, they are placed in between the files. The problem is different ransomware behaves differently, and they need not start encrypting files by triggering the decoy files first. By the time it starts its activity on decoy files, some of the legitimate files may have been encrypted.

A possible workaround for this problem is by having 3 types of decoy files in each path chosen, such that there is one at the beginning, one at the end and one in between the files. Another challenge to our proposed model is the ransomware code trying to avoid the generated files by detecting the file name, the file extension attached to it, file size, reading the contents, etc. The problem with ransomware trying to read file size or file content is quickly solved by the trigger functionality implemented with the system. However, avoiding file name approach is both a tradeoff for both attackers and us. We have made sure that the name generated is random and try not to fall into generic name categories.

The techniques formulated for network packet analysis was able to detect the ransomware strains when tested in a virtualized environment. The IP of the detected packets where added to firewall rule of the Windows Operating System, and it was observed that no future communication took place. Detection of the ransomware packet is based on the dataset of ransomware. For any ransomware packets, which may avoid detection are handled by the process analyzer module of the application.



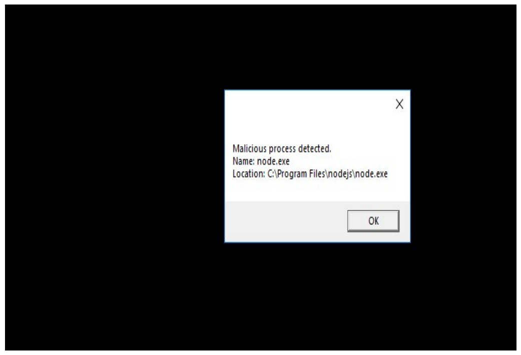
```
PageFileUsage: 416940
PeakWorkingSetSize: 13660
PeakVirtualSize: 2200459620352
PeakWorkingSetSize: 804644
QuotaNonPagedPoolUsage: 13
QuotaNonPagedPoolUsage: 85
QuotaPagedPoolUsage: 161
QuotaPagedPoolUsage: 794
QuotaPeakNonPagedPoolUsage: 14
QuotaPeakNonPagedPoolUsage: 163
QuotaPeakPagedPoolUsage: 1321
QuotaPeakPagedPoolUsage: 161
ReadOperationCount: 682635
ReadOperationCount: 5
ReadTransferCount: 27915060936
ReadTransferCount: 9942
OtherOperationCount: 232038261
OtherOperationCount: 1551
OtherOperationCount: 232039175
HandleCount: 3325
OtherOperationCount: 1551
PrivatePageCount: 426946560
HandleCount: 376
PageFaults: 9099684
PrivatePageCount: 2678784
ThreadCount: 40
PageFaults: 3886
VirtualSize: 2199990710272
ThreadCount: 7
```

Ransomware packet has been detected. Adding IP to firewall.
Source IP: 172.16.32.91
Source Port: 63561
Destination IP: 172.16.51.5
Destination port: 3128

Fig. 9: Alerting the user of a detected ransomware packet

In process detection technique, the ransomware process was detected based on the threshold value set for the day. The identified process was terminated from executing, and the file responsible for it was deleted.

A program written using NodeJS was used to encrypt the filesystem in the test system to determine the working of the analyzer module. An AES encryption was used to encrypt the files. The program was scheduled to run after an hour the system was switched on. Once the test program started executing, the application detected the malicious nature and aborted the process.



Malicious process detected.
Name: node.exe
Location: C:\Program Files\nodejs\node.exe

Fig. 10: Alerting the user of ransomware exhibiting behavior process

V. FUTURE WORKS

A balance needs to be maintained to reduce the computation time when creating decoy files. In larger file system it is observed that it takes several hours when paths are enumerated. The initial setting up of the threshold may not be suitable for every system. It depends on the purpose of the system being used. In a highly critical system, threshold values need to be higher to prevent false positive results. In home systems, the threshold value can be little lower than critical systems.

Authorized licensed use limited to: Modern Education Society's College of Engineering. Downloaded on October 09,2023 at 05:19:34 UTC from IEEE Xplore. Restrictions apply.

VI. Conclusion

Although our system can detect the ransomware activity on instant access, it lacks the feature of recovering the affected files. In detection by process method, there is a chance that a few files are encrypted before it is blocked. Overall the system is much effective against preventing a wide range of ransomware attacks before much damage is caused.

Acknowledgment

The authors wish to thank Satheesh Kumar, Cyber Security Group, CDAC Trivandrum for the guidance and support provided. The authors also wish to thank ER & DCI Institute of Technology for providing an opportunity to research ransomware.

References

[1] Trendmicro.com. (2018). Ransomware - Definition - Trend Micro USA. [online] Available at: https://www.trendmicro.com/vinfo/us/security/definition/ransomware [Accessed 15 Feb. 2018].

[2] Harley, D. (2018). A Trojan Anniversary. [online] WeLiveSecurity. Available at: https://www.welivesecurity.com/2009/12/18/a-trojan-anniversary/ [Accessed 15 Feb. 2018].

[3] Daniel F Netto, Elizabeth Rose Lalson, and Shony K M, "A study on ransomware," National Conference on Recent Advancements in Computing and Information Science 2018, vol. 1, pp. 69-72, December 2017.

[4] Abrams, L. (2018). The Locky Ransomware Encrypts Local Files and Unmapped Network Shares. [online] BleepingComputer. Available at: https://www.bleepingcomputer.com/news/security/the-locky-ransomware-encrypts-local-files-and-unmapped-network-shares/ [Accessed 15 Feb. 2018].

[5] Crowe, J. (2018). Must-Know Ransomware Statistics 2017. [online] Blog.barkly.com. Available at: https://blog.barkly.com/ransomware-statistics-2017 [Accessed 15 Feb. 2018].

Authorized licensed use limited to: Modern Education Society's College of Engineering. Downloaded on October 09,2023 at 05:19:34 UTC from IEEE Xplore. Restrictions apply.