Poster：A New Approach to Detecting Ransomware with Deception

Yun Feng, Chaoge Liu, Baoxu Liu
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
(fengyun, liuchaoge, liubaoxu)@iie.ac.cn

***Abstract*—Ransomware is a harmful threat in cybersecurity now. It seriously affects user's data and property. The increasing amount of ransomware's new variants and Ransomware-as-a-Service make the problem much urgent. In this work, we proposed a new approach to detecting suspicious ransomware in real time based on deception and behavior monitoring. We create decoy files for malicious behaviors judging so that our approach result in no loss before ransomware is detected. We have done a preliminary experiment with locky, which the result shows that our approach is effective with insignificant spatial cost and system resource consumption.**

Keywords—ransomware; malware detection; deception

I. INTRODUCTION

In recent years, ransomware, a specific kind of malware, which attacks victims to extort money, has been a new trend for cybercriminals. Generally, there are two kinds of ransomware, Lock and Crypto [1]. The former one restrict users to accessing their computer normally. The later one, which is more common in present, encrypts files with strong and complicated algorithm to limit users from accessing them. Relatively, Lock can be remedied by wiping the system or removing the disk [2], so Crypto becomes main menace. For this reason, we concentrate on Crypto ransomware in this poster.

Ransomware usually leave an instruction on screen. After victims paying the ransom, cybercriminals will restore files/system or break their promises and disappeared, leading to bigger losses. Large amounts of users are lack of knowledge of cybersecurity and consciousness of backing up important files. Worse, Ransomware-as-a-Service(RaaS) becomes a new trend because it allows nontechnical criminals to make attacks at a very low cost.

Some methods were proposed to against ransomware. The first one is static pattern matching or static code analysis. But this approach turns to be inefficient because there are many new variants and they are usually difficult distinguished from normal application. At present, a more practical and efficient method is dynamic detection based on analysis of behavioral feature. [3] presented a solution to detect ransomware by analyzing I/O requests sequence and difference of screenshots in a virtual environment. However, this solution is only applied to detected environment instead of real situation. [2] summarized and classified the behaviors of ransomware while doing file operation as indicators, and detected ransomware by monitoring and tracking those indicators. [4] presented a distributed client-server architecture, and detected suspicious traits of processes which are doing file operation. [5] used the method that monitor application's connection requires, against ransomwares which would connect C&C server while running. Even though these methods are advanced, they cannot prevent loss completely.

II. DESIGN

In this poster, we presented a new approach based on deception and behavior monitoring to detect crypto ransomware in real time with low cost and no loss.

Crypto ransomware encrypts files in victims' system. Generally, crypto ransomwares first traverse all files and then do read/write operation to encrypt them. Because this reason, we take measures in file traversing. We create decoy files and make the ransomware first find and operate them. Meanwhile, we design a monitor module observing the behavior of suspicious application when it operates decoy files.

We believe this approach is effective. Because ransomwares usually encrypt files according to the search order. If not, our approach would work as well for file traversing is recursive and depth-first. That is, file searching threads traverse all files in one directory before accessing next directory. Through our method, it is equal to placing decoy files in every directory. Therefore, ransomwares will search large quantities of decoy files while traversing files. It is worth noting that there are not many files copied, which avoid excess disk occupation.

Applications implement file operations by calling two Windows APIs, FindFirstFile and FindNextFile. (In fact, FindFirstFileW and FindNextFileW for Unicode, while FindFirstFileA and FindNextFileA for ANSI.) Using FindFirstFile searches a directory for a file or subdirectory with a name that matches a specific name or partial name with wildcards. FindNextFile is called after FindFirstFile to continue a file search. Hence, we focus on every process which calls FindFirstFile or FindNextFile. The way to judge whether a process has called the two APIs is hook. Hook is a kind of technique used to alter or augment the behavior of applications by intercepting function calls or messages or events passed between software components.

The process of our design is showed in Figure 1.

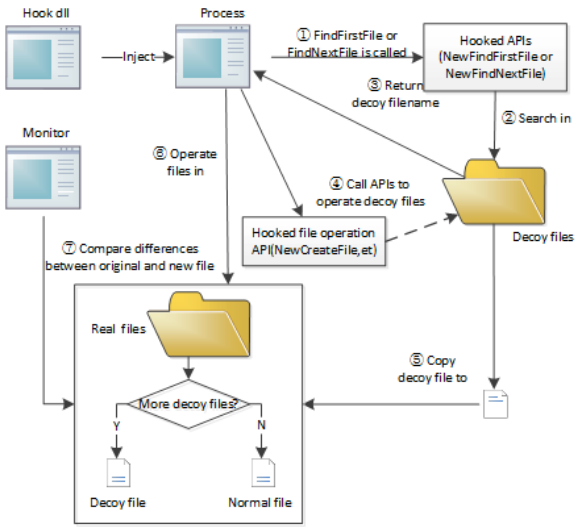

Fig.1. Process of design

Creating decoy files. We create certain number of decoy files in a specific path. Those decoy files cover types which ransomware usually targeted at like txt, doc, pdf, etc.

Injecting all processes. There are several ways to monitor API's calling of a process. For example, modifying IAT or writing Windows driver. In Consideration of efficiency and cost, we decide to adopt the following method and do experiments to test and verify feasibility. We write a demo DLL file which involves the function of hook to monitor all applications running on the system. Once we run the DLL file, every process, as well as a new one, would be injected and then be under monitoring.

NewFindFirstFile API. We alter the function of FindFirstFile API and define it as NewFindFirstFile. By using hook technique, NewFindFirstFile is called first when FindFirstFile API is found calling. Instead of returning a filename the process is really finding, NewFindFirstFile return a name of a decoy file we have already created before. The file name is marked with a special string, which is the symbol of a decoy file so that the file monitor module can keep watch on its change. Thus, it can ensure that every process will first operate decoy files in any condition.

NewFindNextFile API. Similarly, NewFindNextFile is called when FindNextFile API is found calling. It searches files in decoy directory. When there is no more 'next file' in decoy directory, it will call original FindFirstFile API and return the real file handle. Then the process search files in the directory it aims at originally.

Copying decoy files. The NewFindFirstFile and NewFindNextFile make process get a unique filename. For ransomware, to encrypt files, it will do operation such as open, create and delete by calling FileOpen, CreateFile, DeleteFile API. When those APIs are found calling to operate the decoy file, the decoy file will be copied to the path where the original FindFirstFile API has claimed and renamed as the name with the special string. So that the process is able to operate these decoy files correctly, which is helpful for monitor module to observe behaviors of the process.

Monitoring decoy files. Decoy file monitor is used to check whether the file has been encrypted by comparing Shannon entropy between original file and the one has been changed by an application. There are also many effective method to detect, such as modification of file type, similarity comparison with sdhash [6] or a combination of methods mentioned above.

If a process shows any malicious trait, it will be shut down and a notice will be displayed on the screen to let users know.

Our new approach will result in no loss because processes have to operate decoy files first and will be allowed to operate real files only after they pass the detection. The cost of CPU, memory and disk is low. And we are able to detect all of the existing and even new variants due to the mechanism of our approach. We did a preliminary experiment with locky. Our approach worked within tolerable time delays and found the encrypting operation and stoped it timely, which shows that the approach is effective.

III. FUTURE WORK

In the current stage of the work we have implemented the preliminary approach and have done an experiment with locky. In the future, we will do more experiments with different kinds of ransomwares. Besides, adopting more useful and efficient indicators to improve accuracy and efficiency. Moreover, we will test the coexistence of our approach and unmalicious applications to avoid bad influence on user's normal usage.

ACKNOWLEDGEMENT

This work is supported by Key Laboratory of Network Assessment Technology, Chinese Academy of Sciences and Beijing Key Laboratory of Network security and Protection Technology.

REFERENCES

[1] Sgandurra D, Muñozgonzález L, Mohsen R, et al. Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection[J]. 2016.

[2] Scaife N, Carter H, Traynor P, et al. CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data[C]// IEEE, International Conference on Distributed Computing Systems. IEEE, 2016:303-312.

[3] Engin Kirda. UNVEIL: A large-scale, automated approach to detecting ransomware (keynote)[C]// 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE Computer Society, 2017:1-1.

[4] Shukla M, Mondal S, Lodha S. POSTER: Locally Virtualized Environment for Mitigating Ransomware Threat[C]// ACM Sigsac Conference on Computer and Communications Security. ACM, 2016:1784-1786.

[5] Ahmadian M M, Shahriari H R, Ghaffarian S M. Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares[C]// International Iranian Society of Cryptology Conference on Information Security and Cryptology. IEEE, 2015:79-84.

[6] V. Roussev. Data fingerprinting with similarity digests. In *Advances in Digital Forensics VI*, IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, 2010.