RansomSOC: A More Effective Security Operations Center to Detect and Respond to Ransomware Attacks

Anthony Cheuk Tung Lai[1,2*], Ping Fan Ke[3], Kelvin Chan[4], Siu Ming Yiu[5*], Dongsun Kim[6], Wai Kin Wong[2], Shuai Wang[2], Joseph Muppala[2], and Alan Ho[1]

[1]VX Research Limited, Langham Place Office Tower, 8 Argyle Street, Suite 2512, Hong Kong
anthonation@gmail.com, alanh0@vxrl.hk

[2]Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
{wkwongal, shuaiw, muppala}@cse.ust.hk

[3]Singapore Management University, 81 Victoria St, Singapore 188065, Singapore
pfke@smu.edu.sg

[4]Microsoft Corporation, One Microsoft Way, Redmond, Washington, 98052-6399, USA
kelvin.chan@microsoft.com

[5]University of Hong Kong, Pok Fu Lam, Hong Kong
smyiu@cs.hku.hk

[6]Kyungpook National University, 80, Daehak-ro, Buk-gu, Daegu, Republic of Korea
darkrsw@knu.ac.kr

Received: May 5, 2022; Accepted: July 27, 2022; Published: August 31, 2022

Abstract

Ransomware remains a major threat for organizations. Despite a lot of research done, existing solutions still have at least two shortcomings. (I) *Slow detection time*: by the time we realize that the system is under ransomware attack, almost all files have been encrypted. (II) *Without a ransomware-aware backup scheme*: Most existing systems, in particular those in SMEs (small and medium enterprises), do not have a proper backup system. Even they have it, either it is not a remote-site backup (i.e., files in the backup system may also be encrypted) or it is not designed for ransomware attacks. In this paper, based on the analysis of four popular ransomware families, we propose the design of a more effective Security Operations Center (SOC) framework specific to ransomware attack detection and response, called RansomSOC. The core ideas behind RansomSOC are the followings. (a) *A novel real-time emergency local data backup scheme*: we exploit a *design flaw* of ransomware and come up with a scheme to enable a real-time emergency data backup of critical files even after the attack starts, to keep the number of encrypted files as few as possible. (b) *Easy-to-detect ransomware honey files*: Based on the change of entropy values, we identified a set of file types to create honey files (in a honeypot), which facilitate our detection module to quickly detect the existence of a ransomware attack. Our experiments show that RansomSOC is able to detect an attack within about 5 - 10 seconds after the attack starts. For a 1GB folder, RansomSOC is able to backup more than 91% of the data even after the attack starts. And over 95% of this data can be restored.

Keywords: Ransomware, Virus, Malware

Journal of Internet Services and Information Security (JISIS), volume: 12, number: 3 (August), pp. 63-75
DOI:*10.22667/JISIS.2022.08.31.063*

[*]Corresponding authors: Anthony Cheuk Tung Lai and Siu Ming Yiu, Tel: +852-28578242, Web: `https://www.cs.hku.hk/index.php/people/academic-staff/smyiu`

1 Introduction

In the recent decade, ransomware attacks are getting more prevalent and severe. The estimated global economic loss due to ransomware is estimated to be 20 billion USD in 2021 and is predicted to be 265 billion in 2031 [1]. The average cost of ransomware, including the ransom and the recovery cost, is estimated to be 1.85 million USD per victim. Although a lot of research has been conducted and there exist quite a number of ransomware prevention and detection products such as anti-virus software and endpoint security solutions, the effectiveness of existing approaches in defending again ransomware attacks is still not satisfactory. There are at least two major issues that we need to tackle to improve the situation.

1. *Slow detection time*: Most systems and networks do not provide sufficient pre-alert indicators and logs to enable a fast detection of a ransomware attack. Adopting a Security Operations Center (SOC) also cannot solve the problem as a typical SOC is not explicitly designed for ransomware (see Section 3 for more details). Thus, most of the files are already encrypted when the defending system realizes that it is under a ransomware attack.

2. *Without a ransomware-aware backup scheme*: Most existing systems, in particular those in SMEs (small and medium enterprises), do not have a proper backup system. Even they have it, either it is not a remote-site backup (i.e., files in the backup system may also be encrypted) or it is not designed for ransomware attacks. We remark that there are existing backup solutions (e.g., ShieldFS [2] and RDS3 [3]) that address ransomware attacks, however, the cost is too high and it is also likely that the ransomware is able to gain access to the backup servers if the network is not segmented in an appropriate way resulting in the encryption of also the backup data. Some ransomware, such as Locky and Crypto, also directly targets backup files [4]].

1.1 Our solution

In this paper, based on the analysis of four popular ransomware families (BlackMatter, Conti, DarkSide, and REvil [5, 6]), we propose a more effective Security Operations Center (SOC) framework specifically designed for ransomware attack detection and response, called **RansomSOC**. There are two core ideas behind RansomSOC.

1. *A novel real-time emergency local data backup scheme*: we exploit a *design flaw* of ransomware and come up with a scheme to enable a real-time emergency data backup[1] of critical files when the attack starts, to keep the number of encrypted files as few as possible. The key concept is that ransomware may not encrypt system files or directories. That is, if we are able to move and rename our files accordingly, even though the ransomware locates these files, it may not encrypt it. See Section 2 for more details. This solves the problem of ”without a real-time ransomware-aware backup scheme”.

2. *Easy-to-detect ransomware honey files*: The key concept behind is that encrypting a file will change its entropy values. Based on the change of entropy values, we identified a set of file types which facilitate our detection module to quickly detect the existence of a ransomware attack. We create a set of honey files based on these file types. Together with a well-designed honeypot, keeping track of these honey files (or referred as decoy files), RansomSOC can detect a ransomware attack in a faster manner.

[1] Note that this backup is different from the regular backup as regular backup may not be up-to-date and we want to backup the files in an emergency manner when a ransomware is detected.

We remark that RansomSOC does not aim at providing a perfect defensive and no-data loss solution. On the other hand, our main objective is to attempt to save as many important files as possible based on the following assumptions during an incidence of a ransomware attack.

- The attackers have already compromised the system.
- It is difficult to terminate the ransomware process as the anti-virus software may have been disabled and the user credentials may have been changed or compromised by the attackers.
- Some user files are already encrypted due to the time lag between the start time of the ransomware encryption process and and the countermeasure start time of our backup procedure. Thus, there is no guarantee that all files can be saved and recovered (e.g., may be we are not able to backup the whole file resulting with a corrupted version).

We have conducted several experiments to verify the effectiveness of RansomSOC. Our experiments show that RansomSOC is able to detect a ransomware attack withinabout 5 - 10 seconds. For a 1GB folder, RansomROC is able to backup more than 91% of the data and restore over 95% of the backup data even after the attack has started.

1.2 Our contributions

To summarize, the novelty and originality of this research include the followings. (1) We identify folder(s) and file type(s) that are not targets of ransomware. Based on this finding, we develop a real-time emergency local backup scheme to move and hide critical files locally to avoid being encrypted during an attack. To the best of our knowledge, we are the first to propose such a solution. This novel scheme, when compared to existing solutions, has a number of advantages. Although local data backup is also used by some anti-virus software when ransomware is detected, the ransomware usually already stopped the anti-virus software, so the backup cannot be executed. Some security software also does local data backup as an explicit file archive. However, ransomware also targets and encrypts those well-known security software backup file types. On the other hand, we leverage the fact that in those folder(s) with those file type(s), even the ransomware discovers the files, it will not encrypt them due to its design. It is not easy for the ransomware to distinguish the data files from the system files in a short time with limited resources. Compared with remote backup, since we conduct this emergency backup in local stores, we are much faster without being affected by the network delay. (2) Using the property of entropy change, we design a honeypot-based module that can quickly detect if the system is under a ransomware attack. The key is the identification of certain file types, based on observable changes in Shannon entropy before and after encryption, for the creation of easy-to-detect honey files. Together with (1), our solution provides a fast and simple approach to rescue as many files as possible even under a ransomware attack.

The rest of the paper is organized as follows. Section 2 describes the details of our two core ideas: (i) the proposed novel real-time emergency local data backup scheme; and (ii) how to identify easy-to-detect ransomware honey files. Our RansomSOC framework and how it is different from a typical SOC will be discussed in Section 3. Sections 4 provides experimental results to illustrate the effectiveness of RansomSOC. Section 5 concludes the paper.

2 The two core ideas

2.1 A novel real-time emergency local data backup scheme

We propose a novel approach to back up data locally by hiding the critical files into the file types that are not targeted by ransomware. This approach exploits a *design flaw* of ransomware in which system-related files may not be encrypted so that the system can still function to display a ransom note. We analyzed four recent ransomware families (BlackMatter, Conti, DarkSide, and REvil) to identify the untargeted file types. We found that executable (EXE), shortcut (LNK), and library files (DLL, SYS) files are not encrypted by these ransomware families. Similarly, we found that some ransomware will not encrypt the files under system-related folders such as C:\Windows and C:\Windows\System32. Our proposed data backup scheme is used for emergency[2] backup during a ransomware attack.

One advantage of this approach is the fast execution time. Other backup solutions, especially those requiring a network connection, may not be feasible when ransomware strikes. At the same time, the data survival time could be extended when the data is hidden in the untargeted file types.Compared to a kill-switch solution, which shuts down the computer or stops all processes when ransomware is detected, our approach will not cause data corruption due to unexpected process termination. One limitation is that some ransomware may behave differently and target some system-related files. To increase the chance of data survival, instead of using a naive circumvention technique by simply putting the critical files into an untargeted folder (e.g., C:\Windows\System32) or changing the extension of critical files to an untargeted one (e.g., DLL), we propose the following algorithm to enable stealthy local data hiding backup. It leverages both untargeted file type and untargeted folder and injects the encrypted data in a steganographic technique to avoid possible detection by ransomware.

Algorithm to hide and restore the data locally from ransomware encryption:

1. Hiding (for every important file in the folder)
   (a) Encrypt the file
   (b) Equally split the file into *n* partitions
   (c) Append *n* splits of the file to the non-ransomware targeted file header.
   (d) Save to non-ransomware targeted folder as backup
2. Restoration (for every protected file split)
   (a) Remove the file header from the protected file split
   (b) Combine *n* splits of the file
   (c) Decrypt and output the restored file

We tested the above algorithm by running the ransomware against the backup files. We placed these specially crafted backup files in various non-ransomware targeted folders, including C:\Windows, C:\Windows\System32, and C:\Program Files, using one of the following non-ransomware targeted file types: DLL, EXE, LNK, SYS. We found that these backup files could circumvent ransomware encryption, and we could successfully restore those data from the ransomware attack. See the experiments in Section 4 too.

2.2 Easy-to-detect ransomware honey files

We first illustrate how a ransomware-specific honey file should be built. Although there are many ways to make honey files (e.g., passwords.txt [7]), for our proposed RansomSOC, the honey files should be based on the types of files that exhibit an obvious change in entropy after being encrypted by ransomware, especially when different ransomware may have different behavior in manipulating the folders and files.

[2] This scheme can be used for regular backup as well, however, if the system already has a regular backup system, it may not need to use this approach to do regular backup. The main objective of this scheme is to move away important files as fast as possible to avoid them being encrypted by the ransomware since the regular backup version may be outdated.

To obtain a useful ransomware-specific honey file, we tried to identified these file types by infecting 1,000 randomly crawled files from the Internet with different file types, including CAB, DLL, DOCX, EXE, JPG, GIF, LNK, MP3, MP4, PDF, PNG, PPTX, SYS, TXT, and XSLX using the four ransomware families we mentioned earlier (BlackMatter, Conti, DarkSide, and REvil), in a sandbox. We then compare the Shannon entropy of the honey files before and after the ransomware infection. We found that PNG and RTF files with small file sizes from our sample will exhibit a significant change in entropy before and after the ransomware infection. For examples, in our experiments, RTF and PNG files with size around 0.003MB will have a Shannon Entropy of about 4-5 before encrypted by ransomware and a Shannon Entropy close to 8 after encrypted by ransomware, while other file types have a much smaller difference (e.g. for JPG file, the Shannon Entropy before encryption (about 7.6) and after encryption (close to 8). Other file types also exhibit a similar small difference on the Shannon Entropy before and after encryption). Thus, we select these two types of files (RTF and PNG) as honey files and put them in the hidden honey folders for ransomware attack detection.

3 RansomSOC framework

3.1 Overview of the framework

To justify the need for a specially designed SOC for ransomware, we first compare the differences between a typical SOC and RansomSOC that we are going to propose in Table 1. One key difference is that a typical SOC will not deploy honeypot for ransomware detection. It is difficult for a typical SOC to detect quickly the existence of a ransomware attack. To achieve the desired outcomes of RansomSOC shown in Table 1, we propose a systematic framework to perform ransomware attack data collection, detection, and incident response with interdependent modules. A RansomSOC has a pipeline to first detect abnormal file activities and encryption performed by ransomware, followed by backing up the critical data, executing defense instructions to lower the probability of ransomware attacks against other not-yet infected systems, and notifying the administrators as soon as possible once a ransomware attack is detected. The data backup modules should include the above mentioned real-time emergency local data backup (Section 2.1) scheme.

Table 1: Differences between a typical SOC and RansomSOC

Action	Typical SOC	RansomSOC
Source & collection (data/logs)	Typical system event logs, server logs, access control logs, and threat intelligence only to provide alert and response, without deploying decoy folders and files (known as honey folders and files) for detection as practice.	Specific to ransomware. Hidden decoy folders and files in the server, workstations, and honeypots are deployed strategically.The file or folder change logs are tracked continuously.
Detection	Depend on rules to flag the threat in the SIEM [8] System.	Unlike a typical SOC, the logs of file/folder modification are gathered, and file content entropy and extension are compared.
Reponses	SOC can detect incidents that happen only after encryption or blocking ransomware. However, if the victim machine is compromised, an attacker can execute whatever they want with administrative privilege. Important files will eventually be encrypted as a result.	If RansomSOC detects a ransomware attack, logs will still be exported to the SOC. However, RansomSOC will notify all shortlisted servers and workstations and automatically execute commands to remedy and request file backup in local and remote storage in a separate network with secure authentication and transfer.

In this framework, the RansomSOC should be deployed in a separate network. It mainly comprises eight modules: Ransomware Sandbox, Logs Analyzer, Logs Collector, File Content Entropy and Extension Comparison, File Protector Definition Generator, Administrator Notification Center, Data Backup Orchestra Center, and Defense Command Orchestra Center, as shown in Figure 1. In the system network, servers and workstations are denoted as System(i), accompanied with some honeypot hosts Honeypot(i, j). The details of the honeypot deployment will be discussed in Section 3.2.

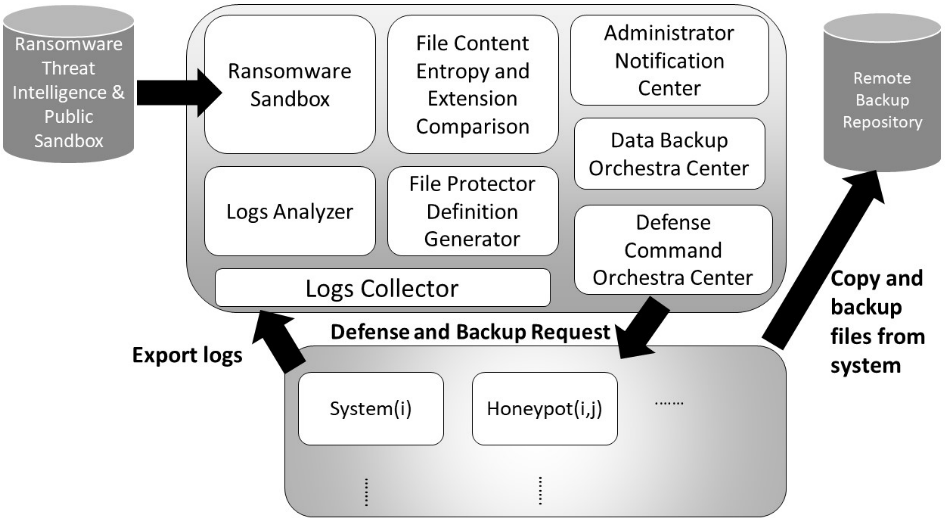

Ransomware Threat Intelligence & Public Sandbox
Ransomware Sandbox
File Content Entropy and Extension Comparison
Administrator Notification Center
Data Backup Orchestra Center
Logs Analyzer
File Protector Definition Generator
Defense Command Orchestra Center
Logs Collector
Remote Backup Repository
Copy and backup files from system
Export logs
Defense and Backup Request
System(i)
Honeypot(i,j)
....

Figure 1: The framework of RansomSOC

To construct the Ransomware Sandbox, we retrieve the latest ransomware sample for analysis from open-source intelligence repositories [9] and VirusTotal [10], a public malware sandbox. The ransomware will be analyzed under our sandbox, which helps to understand what file types will not be encrypted by the ransomware. The sandbox will then append and create a “file protector definition file” through the File Protector Definition Generator. The definition file contains file types and folders that are not encrypted by a particular type of ransomware family. The definition file will be exported to the Data Backup Orchestra Center that instructs the System to hide a pre-defined list of document files to the “file protector” (i.e., the file types that will not be encrypted) under different non-ransomware targeted folders regularly.

The Logs Collector collects the folder and file activity logs of our hidden honey folders and files. It then computes the Shannon entropy value of the honey files. The Log Analyzer is responsible for importing the logs and analyzing whether the log indicates any suspicious file modification, creation, and deletion activities. It helps to identify suspicious attacks in different workstations and servers. The comparison of entropy values (i.e., detect whether the file is encrypted or not) and blacklisted ransomware file extension checks are done in the File Content Entropy and Extension Comparison module. Whenever a positive result is identified, it will communicate with the Administrator Notification Center to notify the administrator, initiating commands to request shortlisted workstations and servers to back up the data and execute defense actions via the corresponding orchestra centers.

Data Backup Orchestra Center will initiate two types of backups: *remote* backup and *real-time emergency local backup*. Remote backup transfers the data from the server to remote backup repositories, including cloud and delegated servers. The real-time emergency local backup is the one we proposed in Section 2.1 which tries to hide the files in local directories during the attack.

3.2 Honeypot deployment

RansomSOC has two major detection mechanisms: honeypot hosts and honey folders and files. While RansomSOC can still work with only honey folders and files, it is recommended to deploy honeypot hosts to increase detection capability. The honeypot host's system configuration and software installation should be close to the production workstation and server to attract the attacker to launch the attack. Meanwhile, the honeypot host can quickly be restored with a hardware reborn card [11]. For each system and honeypot host in the network, RansomSOC will deploy hidden honey folders and files to attract ransomware to access them. Typically, these honey folders and files will be hidden in the document directory (e.g., My Documents). The Log Collector will then monitor the activities of these honey folders and files, like file modifications and file extension changes.

Whenever the honey files exhibit a significant change in Shannon entropy, immediate defense instructions (e.g., password reset, disabling Remote Desktop Connection, and unmourned and disconnect all network drives) will be sent to the target servers and workstations through Defense Command Orchestra Center. In addition, Data Backup Orchestra Center will request target servers and workstations to initiate and execute secure copy command to copy all valuable data files to the remote backup server and locally.

4 Experiments

We will conduct two sets of experiments. The first set of experiments is to check the time taken for RansomSOC to realize that a system is under a ransomware attack and the detection successful rate. The second set of experiments to evaluate the effectiveness of our backup scheme.

4.1 Detection performance

Recall that (explained in Section 2.2) we mainly use PNG and RTF files as our honey files. We select ransomware samples from five high-profile and high-impact ransomware families which are downloadable from a public virus repository [10]. These ransomware samples are the newest and latest, already known by the public and anti-virus detectors in the public virus samples repository.

For this set of the experiments, first, we prepare around 100 MB of files (with documents, images, videos, zipped files, and audio files). Then, we deploy them to each target folder listed in Table 2. Second, we run a ransomware sample in the corresponding ransomware family ten times in the listed folders in a freshly built workstation with Microsoft Windows 10 Professional Edition (the latest security patches installed), where anti-virus and firewall are all disabled. Third, we record the detection time (in seconds) and obtain the average time after ten runs.

In Table 2, it shows that the honey files deployed in common user directories were located and encrypted by all ransomware samples. RansomSOC are able to detect all of them, so the successful detection rate for all cases is 100%. On the other hand, honey files in "Program Files", "System32" and "Windows" folders (system related folders) are not encrypted by ransomware, which verifies our claim in Section 2.1. The detection time on average in seconds among all cases is under 10.2 seconds (ranging from about 5 to 10 seconds). Once the detection is triggered, RansomSOC will send a notification to other machines in the network to carry out further responses.

We have carried out another experiment in an Ethernet network with 100 Mbps speed, a typical (slower) speed in a home network (or a small office network). We have prepared two workstations (same as the above setup), and one receives the ransomware detection notification from the workstation with ransomware encryption. We repeat the same experimental procedure for local ransomware detection by honey files. On average, about an addition of 2.83 seconds are required in such a slower network giving an upper bound of about 13 seconds for all cases. These results show that the performance of RansomSOC is satisfactory.

Table 2: Detection performance of RansomSOC: detection time (in seconds): The detection time is the time required by RansomSOC to detect the attack once a honey file is being encrypted by the ransomware. "NA" means that no files in that directory are encrypted by the ransomware.

Malware families	**The folders where honey files are deployed**							
	Group **A (user data directories)**					Group **B (system directories)**		
	A1	A2	A3	A4	A5	B1	B2	B3
BlackMatter	8.53	6.25	6.41	8.24	8.53	NA	NA	NA
Conti	7.17	6.80	5.15	10.2	7.60	NA	NA	NA
DarkSide	6.76	7.79	5.78	9.54	8.22	NA	NA	NA
Jigsaw	8.41	4.81	4.19	6.14	5.30	NA	NA	NA
REvil	6.32	8.94	7.10	9.23	8.84	NA	NA	NA

A1= C:\<Users>\My Documents B1=C:\Program Files
A2= C:\<Users>\My Downloads B2=C:\Windows\System32
A3= C:\<Users>\My Desktop B3=C:\Windows
A4= C:\<Users>\My Music
A5= C:\<Users>\My Pictures

4.2 Evaluation of our backup scheme

To evaluate the performance of our backup scheme (and data recovery), we executed ransomware from the four ransomware families we mentioned earlier (BlackMatter, Conti, DarkSide, and REvil) in our servers that are set up in Windows 10 Professional Edition, with 256GB hard disk and 4GB RAM in virtualized machines. We also include a ransomware family called Jigsaw in this experiment as it is evaluated in a related study [12]. We set up a target folder with 200MB, 500MB, and 1GB of files in a test bed with three target servers. Two honeypots in the same network accompany each server. These three servers have shared drives accessible to each other, and remote desktop service is enabled. For each attack scenario, we simulate the attacker executing the ransomware at a randomly selected compromised honeypot or server, which will initiate an alert and response by RansomSOC. Each scenario will be executed ten times for each ransomware and size of files. After the ransomware is executed, the ransom note will be shown in 6 to 7 seconds on average, except REvil, which will show the ransom note in 22 seconds. Table 3 exhibits the aggregated data backup and restoration rates of the backed-up data. We also compare our *local* emergency backup with a *remote* backup to verify the advantage of using our real-time emergency local backup.

From Table 3, the results show that a high percentage of data can be hidden locally during the attack using our real-time emergency backup. Even in the worst case (1GB data attacked by BlackMatter), the backup rate is still as high as 91.15%. There are cases with the backup rate close to 99%. Since some date may be corrupted during the backup, it is reasonable to have the data restoration rate less than 100%. Among all the cases, the restoration rate ranges from 95.11% to 99.70%. If we look at the overall percentage of how much data our scheme can rescue (i.e., backup rate × restoration rate), among all scenarios, the case with the worst performance is infected by DarkSide on 1GB dataset, which still has 86.88% (91.35% × 95.11%). On the other hand, for remote backup, both the backup rate and the restoration rate are substantially worse than our real-time emergency local backup. This is due to the network delays (96Mbps upload speed). Thus, the approach of using local directories which are not targets of ransomware is a promising approach to backup critical files when the system is being attacked by a ransomware.

Table 3: Performance of our real-time emergency local backup scheme

Local emergency backup	**200MB**		**500MB**		**1GB**	
	Backup rate	**Restoration rate**	**Backup rate**	**Restoration rate**	**Backup rate**	**Restoration rate**
BlackMatter	93.05%	97.07%	92.25%	97.07%	91.15%	95.44%
Conti	94.76%	97.68%	93.34%	97.68%	92.42%	96.62%
DarkSide	93.35%	96.89%	92.53%	96.89%	91.35%	95.11%
Jigsaw	95.78%	97.52%	93.48%	97.52%	95.78%	96.17%
REvil	98.98%	99.70%	97.58%	99.10%	98.98%	98.40%
Remote backup	**200MB**		**500MB**		**1GB**	
	Backup rate	**Restoration rate**	**Backup rate**	**Restoration rate**	**Backup rate**	**Restoration rate**
BlackMatter	53.31%	79.01%	47.88%	71.03%	40.97%	66.94%
Conti	58.45%	86.19%	53.31%	79.95%	45.79%	70.61%
DarkSide	53.89%	78.76%	48.16%	72.29%	41.16%	65.28%
Jigsaw	57.71%	85.29%	52.34%	79.33%	43.11%	71.64%
REvil	78.22%	94.72%	74.12%	86.31%	65.02%	80.66%

5 Conclusions

5.1 Related works

There is research about hiding the data file in Alternate Data Streams (ADS) in a file [12]. We attempt to reproduce their experiment, but we found that the backup generated with their method will still be encrypted by the Jigsaw ransomware they used for evaluation when the files are placed in common user directories (e.g., My Documents, Downloads). In contrast, the files will not be encrypted when placed in C:\Windows, the directory used by their experiment. This also verifies our claim in Section 2.1 that a better hiding place would be a system-related directory. Another related research suggests a new file structure called Protected ADS User File (PAUF) [13] which backup files in ADS, yet ransomware may still scan any ADS of files to uncover the backup.

Regarding file-related attack detection, research has deployed honey files to detect ransomware when the deceptive file is changed [7]. But, they did not have an effective method to see whether the content is encrypted or not. The content can be accidentally changed, but it is not a ransomware infection, and therefore we suggest using Shannon entropy change as the detection method. However, a study indicated that the change in Shannon entropy approach is unreliable by showing a counterexample: a non-encrypted and an encrypted file with the same entropy values are generated with the manipulation of base64 encoding and distributed non-selective partial encryption [14]. Nevertheless, in the real world, there is no ransomware implemented with this method yet. If such an approach exist, entropy-based detection could be combined with other event log analysis to provide an accurate ransomware detection.

There are also other backup solutions to avoid damage from ransomware attacks. For example, [15] suggests a device-level backup because backup copied from typical backup solutions may be destroyed by ransomware. Another alternative is to back up data in a secure space locally or remotely (e.g. ShiedFS [2], RDS3 [3]) These alternatives support local secure space backup, but the current system needs to be modified and the cost is high. Organizations, especially SMEs, are reluctant to deploy these approaches. For secure remote spaces, solutions such as CLDSafe [16] need to be engaged with external resources.

Our real-time emergency local backup scheme offers a much simpler way for data backup, which has a high recovery rate even under ransomware attacks.

5.2 Remarks and future work

In this paper, based on our experiments, our proposed RansomSOC performs quite satisfactory in terms of the detection time/rate of a ransomware attack and the percentages of backup and restoration rates. The two interesting ideas behind our RansomSOC are based on (i) the use of non-ransomware targeted file types and folders as a technique for emergency backup; and (ii) the selection of specific file types to create honey files for ease detection using entropy changes. As we mentioned, RansomSOC is not a perfect solution yet although the results show that the direction is promising.

Regarding further work, it is always desirable to have a faster detector. The faster we are able to detect the attack, the more critical files can be protected. In our solution, we mainly use the entropy change of honey files to detect the attack, we believe that there are other indicators that can be leveraged. A more dynamic detector with mixed indicators could also be explored. To hide critical files, a more randomized approach can be developed to confuse the ransomware by putting critical files in multiple folders with different file names and types that change every time when under an attack. We can also implement the scheme in kernel filter driver level. Also, to cover more ranosmware families and how in practice we can integrate RansomSOC with a typical SOC are also important topics.

Acknowledgement

This project is partially supported by the theme-based research project (T35-710/20-R) funded by the RGC of the Government of HKSAR and the HKU-SCF FinTech Academy.

References

[1] D. Bisson. Ransomware costs expected to reach $265 billion by 2031, August 2021. `https://securityintelligence.com/news/ransomware-costs-expected-265-billion-2031` [Online; Accessed on July 23, 2022].

[2] A. Continella, A. Guagnelli, G. Zingaro, G. De Pasquale, A. Barenghi, S. Zanero, and F. Maggi. Shieldfs: a self-healing, ransomware-aware filesystem. In *Proc. of the 16th Annual Conference on Computer Security Applications (ACSAC'16), California, Los Angeles, USA*, pages 336–347. ACM, December 2016.

[3] K.P. Subedi, D.R. Budhathoki, B. Chen, and D. Dasgupta. Rds3: Ransomware defense strategy using stealthily spare space. In *Proc. of the 17th IEEE Symposium Series on Computational Intelligence (SSCI'17), Honolulu, Hawaii, USA*, pages 1–8. IEEE, December 2017.

[4] S. Pritchard. The limits and risks of backup as ransomware protection, May 2022. `https://www.computerweekly.com/feature/The-limits-and-risks-of-backup-as-ransomware-protection` [Online; Accessed on July 23, 2022].

[5] T. Micro. Examining Erratic Modern Ransomware Activities: Ransomware in Q3, Novemeber 2021. `https://www.trendmicro.com/vinfo/us/security/news/ransomware-by-the-numbers/examining-erratic-modern-ransomware-activities-ransomware-in-q3-2021` [Online; Accessed on July 23, 2022].

[6] A. Mundo and M. Elias. BlackMatter Ransomware Analysis; The Dark Side Returns, September 2021. `https://www.trellix.com/en-us/about/newsroom/stories/threat-labs/blackmatter-ransomware-analysis-the-dark-side-returns.html` [Online; Accessed on July 23, 2022].

[7] J. Yuill, M. Zappe, D. Denning, and F. Feer. Honeyfiles: deceptive files for intrusion detection. In *Proc. of the 5th Annual IEEE SMC Information Assurance Workshop (IAW'04), Linz, Austria*, pages 116–122. IEEE, June 2004.

[8] M. Cinque, D. Cotroneo, and A. Pecchia. Challenges and directions in security information and event management (siem). In *Proceedings of the 18th IEEE International Symposium on Software Reli-ability Engineering Workshops (ISSREW'18), Memphis, Tennessee, USA*, pages 95–99. IEEE, October 2018.

[9] A. Sharma, J. Breeden, and J. Fruhlinger. What is OSINT? 15 top open source intelligence tools, June 2021. `https://www.csoonline.com/article/3445357/what-is-osint-top-open-source-intelligence-tools.html` [Online; Accessed on July 23, 2022].

[10] VirusTotal. VirusTotal. `https:/www.virustotal.com` [Online; Accessed on July 23, 2022].

[11] Lenten. Recovery (Reborn) Card User's and Reference Manual, November 2018. `http://testere-adaptoare.sorinescu.net/drivere/card-protectie/varianta-A/Recovery%20%28Reborn%29%20Card.pdf` [Online; Accessed on July 23, 2022].

[12] J. Y. Paik, G. Kim, S. Kang, R. Jin, and E.S. Cho. Data protection based on hidden space in windows against ransomware. In *Proc. of the 6th International Congress on Information and Communication Technology (ICICT'21), Goldwins, Coimbatore, India*, volume 235 of *Lecture Notes in Networks and Systems*, pages 629–637. Springer-Verlag, September 2021.

[13] J. Morris, D. Lin, and M. Smith. Fight Virus Like a Virus: A New Defense Method Against File-Encrypting Ransomware, March 2021. `https://jglobal.jst.go.jp/en/detail?JGLOBAL_ID=202202215360241522` [Online; Accessed on July 23, 2022].

[14] T. McIntosh T, J. Jang-Jaccard, P. Watters, and T. Susnjak. The inadequacy of entropy-based ransomware detection. In *Proc. of the 19th International Conference on Neural Information Processing (ICONIP'19), Vancouver, Canada*, pages 181–189. Springer International Publishing, December 2019.

[15] D. Min, Y. Ko, R. Walker, J. Lee, and Y. Kim. A content-based ransomware detection and backup solid-state drive for ransomware defense. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41:2038–2051, July 2022.

[16] J. Yun, J. Hur, Y. Shin, and D. Koo. Cldsafe: an efficient file backup system in cloud storage against ransomware. *The Institute of Electronics, Information and Communication Engineers Transactions on Information and Systems*, E100-D(9):2228–2231, September 2017.

Author Biography

Anthony Cheuk Tung Lai focuses on threat analysis, incident response, and red/blue team testing. He likes hunting bugs and vulnerabilities. Currently, he is the director of VX Research Limited and is a Ph.D. candidate in Computer Science at HKUST. For community work, he found VXCON in 2010 and is an overseas mentor of the Best of the Best (BoB) program in South Korea. He is the CFP reviewer of Black Hat Asia and Hack In The Box.

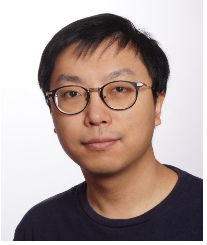
Ping Fan Ke is an assistant professor in the School of Computing and Information Systems at Singapore Management University. His research interests include web application security, blockchain and smart contract security, and economics of security. Dr. Ke received his Ph.D. degree from the Hong Kong University of Science and Technology (HKUST) in 2018.

Kelvin Chan is a Windows Kernel Engineer in Microsoft responsible for researching and enabling new generation technology underneath Windows Operating System; He was a Security Researcher in both Microsoft and Tencent, protecting firmware and PC game software, respectively.

Siu Ming Yiu is currently a professor in the Department of Computer Science of the University of Hong Kong. He is the deputy executive director of HKU-SCF FinTech Academy and the director of the FinTech and Blockchain Laboratory. His research areas include cyber security, cryptography, and FinTech.

Dongsun Kim is an assistant professor in the School of Computer Science and Engineering at Kyungpook National University. His research interest includes software engineering, program analysis, and software testing.

Wai Kin Wong is a Ph.D. student at the Hong Kong University of Science and Technology. He is interested in software security, binary analysis, and machine learning. He has presented his research in HITCON and VXCON.



Shuai Wang is an Assistant Professor at CSE, HKUST since 2019. Before that, he was a Postdoctoral Scholar in the AST lab at ETH Zurich. He received his Ph.D. from Penn State University and B.S. from Peking University.

Joseph Muppala is currently an associate professor in the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. He served as the program director for the Master of Science in Information Technology (MSc(IT)) program from 2008-2016. He also served as the Undergraduate Studies Director in the department from Aug. 1999 to Aug. 2001 and the Chair of Facilities Committee in the department from Sep. 2005 to Aug. 2007.

Alan Ho has over 15 years of experience in the information security field. He is the co-founder of VX Research Limited, as well as the red-blue team lab architect. He focuses on penetration testing, incident response, training, and security operation planning for different clients. He is certified as an OSCP, also a SANS GCIH, GWAPT Holder, and published the SANS Gold Paper - “Website Security For Mobile”.