Self-recovery Service Securing Edge Server in IoT Network against Ransomware Attack

In-San Lei, Su-Kit Tang, Ion-Kun Chao and Rita Tse
School of Applied Sciences, Macao Polytechnic Institute, Macao, China

Keywords: Edge Computing, Ransomware, Recovery.

Abstract: Edge server takes an important role in IoT networks that distributes the computing power in the network and serves as a temporary storage for IoT devices. If there is any ransomware attack to the edge server, its network segment will be paralyzed, raising the data integrity and accuracy issues in the IoT system. In this paper, we propose a self-recovery method, called Self-Recovery Service (SRS), which can detect ransomware signature and recover victim files automatically. No interruption to the operation of edge server would be caused by ransomware. SRS is evaluated in the simulation test and the result shows that SRS takes insignificant system resources for its operation that does not degrade the performance of the edge server.

1 INTRODUCTION

The rapid development of smart cities has driven the deployment of Internet of Things (IoT) system in solving specific problems in different fields, resulting in a dramatical growth in the number of deployed IoT devices. IoT devices are heterogeneous and of low computing capability. An IoT device collects and generates data continuously and data are forwarded to backend servers for processing. If an IoT system is connecting to a large number of IoT devices, a bottleneck problem may be created at the backend. Adding edge servers into the system can ease the bottleneck problem and complicated computing tasks can be simplified at the early stage.

In general architecture, edge servers are interconnected and they serve as a gateway for their IoT local subnet. (Fernández et al, 2018) (Lopez et al, 2015) (Shi et al, 2016) An edge server comes with computing capabilities to pre-process dynamic data generated from IoT devices. If intensive computing tasks on the data are involved, the pre-processing not only reduce the workload of backend servers, but it also simplifies the logic of the data processing. This distributes the power across a network and improves the efficiency of the entire IoT system. For this reason, the market size of edge servers grows steadily and is expected to climb up to 1031 million by 2025 while the number of IoT devices will be projected to reach 75 billion by 2025 (Statista, 2017) (Statista, 2020). In this context, edge server is likely to raise interest from the academia and the industry. It is also very attractive to cyber extortionists, as attacking edge servers rather than IoT devices makes more significant impact to IoT systems. It would raise many challenges related to security and privacy concerns (Shi et al, 2016).

A number of studies (Xiao et al, 2019) (Alrowaily & Lu, 2018) (Hafeez et al, 2016) (Caprolu et al, 2019) (Zhang et al, 2018) have been conducted regarding to the security and privacy of edge server. Edge servers in IoT networks are normally accessible in the public. Any design flaws, misconfigurations and implementation bugs may put them at risk, suffering a number of cyber-attacks including ransomware attack. Recently, ransomware has rapidly become one of the severe network threats for enterprises and individual users, causing billions of dollars in loss globally. The maturity of ransomware has even reached a new height that can attack millions of computers at a time. Files encrypted by ransomware are often unable to be decrypted, unless the decrypting key is obtained. Reasonable defense measures for edge servers to reduce the chance of being attacked is definitely needed. Suppose that a ransomware is injected into an edge server under the malware injection attack, the server would stop accepting data from IoT devices. No processing is done for the IoT system. This results in data lost that prevents the system from making critical decision and thus downgrades the performance of the system.

Lei, I., Tang, S., Chao, I. and Tse, R.
Self-recovery Service Securing Edge Server in IoT Network against Ransomware Attack.
DOI: 10.5220/0009470303990404
In *Proceedings of the 5th International Conference on Internet of Things, Big Data and Security (IoTBDS 2020)*, pages 399-404
ISBN: 978-989-758-426-8

In this paper, we aim at easing the damage caused by ransomware to edge server by proposing a self-recovery method, called Self-Recovery Service (SRS), for edge server. SRS can detect ransomware signature and recover victim files automatically. Its concept is to monitor important files by a system service. If ransomware is detected, SRS recovers infected files by restoring the corresponding backup of raw data. No interruption would be caused to the operation of edge server. The service only takes insignificant system resources for its operation that does not degrade the performance of the server.

The remaining of this paper is organized as follows. In Section 2, we review the ransomware attack. In Section 3, we present the design of SRS. Section 4 will highlight the verification of SRS. Finally, we conclude this paper in Section 5.

2 RANSOMWARE ATTACK

Ransomware is a kind of malicious trojan horse program which is secretly injected into victim's devices (computers, smartphones, servers, etc.) and interferes with their operation by encrypting some important files, such as user data files or system files. To rescue the files and even save the devices, the decrypting key is needed which can be obtained by paying for the ransom (O'Gorman & McDonald, 2012). In 1989, the first ransomware, called PC Cyborg, had successfully forced the user to pay a ransom of $189 by encrypting its hard disk (Gazet, 2010).

There are five types of ransomware reported (Johansen, 2018), which are Crypto malware, Lockers, Scareware, Doxware and RaaS. Crypto malware is a well-known ransomware that can spread over thousands of computers and make damages worldwide. One of the noticeable examples is WannaCry; Lockers is another type of ransomware that aims to attack operating system. It locks down a victim's computer. No files or applications can be accessed; Similarly, Scareware would lock down a victim's computer and pop up annoying messages to ask for ransom; Doxware is another type of ransomware that may reveal a victim's sensitive information. It threatens the victim by posting the information online, if ransom is not paid; The last one, called "Ransomware as a Service (RaaS)", contributed greatly to the growth of ransomware attack because it enables anyone to be a cyber attacker. RaaS is deployed as a portal that enables legitimate venders to unintentionally setup malicious services to their customers (victims).

Ransomware has been evolved gradually since 1989. It has expanded their scope of attack on devices from computers to mobile devices, covering individuals, enterprises, governments, medical institutions, banking systems, etc. A number of attack cases have been reported, including Hollywood hospital network system and Muni subway in San Francisco. They targeted the healthcare industry and government organizations because it makes significant impact to the public. In 2016, the first ransomware for mobile devices, called "Gooligan" has been reported, which led to 1 million Android devices being attacked, by maliciously obtaining the root permission of the devices (Adhikari, 2016).

There are a number of methods to inject ransomware into edge servers: 1) by injecting malicious scripts/codes or 2) by compromising an edge server to spoof other edge servers. In 1), an escape character can be used to attach malicious string into a SQL query that can spoof the database to execute the string, loading unexpected file remotely into an edge server (victim) (Anley, 2002). Similarly, XSS in HTML/JavaScript allows loading expected codes remotely into the edge server (victim) as the victim does not verify XSS codes (Martin & Lam, 2008). On the other hand, in 2), as edge servers are interconnected and would work collaboratively with each other, they would exchange data for processing. If an edge server (victim) is spooled to listen to a compromised edge server (attacker), the victim will execute malicious codes from the attacker. XML signature wrapping is commonly used when launching the attack (McIntosh & Austel, 2005).

Nevertheless, ransomware attack is not detectable immediately when it is taking place in an edge server. In case of being attacked, the IoT system would face the data integrity and data accuracy problems, which may not cause significant harm to them. However, the problems for data-oriented IoT systems (Tse et al, 2018) (Tse & Pau, 2016) (Aguiari et al, 2018) may affect the quality of critical decision the systems make. Thus, a self-recovery function against ransomware attack should be enabled in edge servers. In the next section, we present the Self-recovery service (SRS), which secures edge servers against ransomware attack.

3 SELF-RECOVERY SERVICE

In this paper, the Self-recovery service (SRS) is proposed to ease the damage caused to an edge server by ransomware. SRS is a system service that runs at the kernel level, monitoring the activity of the file system in an edge server. It is particularly sensitive to the ransomware attack. If any suspected activity in storage is detected, it checks for the ransomware signature. If the signature is found in any IoT data files, data files should be unexpected encrypted. It is confirmed that the server is under the ransomware attack. To resolve this situation, SRS takes an immediate action to recover the encrypted file, by recalling its backup copy from the backup node. If there is no suspected activity, SRS continues to send IoT data to the backup node.

It is noteworthy that an edge server may be free from ransomware attack if IoT data is not kept in its storage. However, some existing IoT systems (Tse et al, 2018) (Tse & Pau, 2016) (Aguiari et al, 2018) working in the environment without stable Internet connection is required to keep the data in local storage temporarily. In this context, those systems are vulnerable to the ransomware attack during the blackout period. Thus, SRS is highly recommended.

3.1 Architectural Design

SRS provides a seamless integrating solution that does not require complicated modification to existing IoT systems. Figure 1 shows the architectural design of SRS.

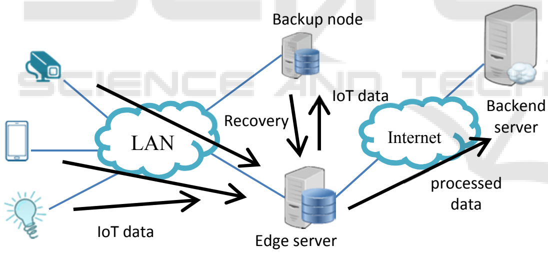


Figure 1: Architectural Design of SRS.

As can be seen in Figure 1, there are a number of IoT devices (i.e., camera, smartphone and lightbulb) connecting to the edge server via Local Area Network (LAN). The edge server collects IoT data from the devices and immediately forwards them to the backup node before doing any pre-processing on the data. The backup node, which is located in the same subnet, acts as a secondary storage for the edge server only. When the recovery process is activated in the edge server, it will recall the data from the backup node. Once the recovery process is done and the data is processed, it is then sent to the backend server through the internet for further processing. It ensures that the entire IoT system operates continuously.

3.2 Backup Node

In SRS architecture, the backup node will receive IoT data periodically from the edge server using a secure network connection, e.g., sftp. Data is received in a form of raw data format. When it is stored in the file system, extra information, such as its originality, will be associated to it. Moreover, SRS maintains the data consistency by Subversion server, tracking its generation date, time and version. The tracking operation also covers system files and folders, in order to recover the edge server if its system is attacked.

To detect the ransomware in the edge server, SRS monitors the activity of the file system. The detection logic will be explained in the next section.

3.3 Ransomware Detection

Ransomware attack typically aims to take control of important files in a computing system by cryptographic encryption. There are a number of file types that ransomware is interested in, which are the user files (e.g., docx, pptx, pdf, etc.) in home folder and system files in the system folders. In addition, other particular files and folders may be their targets depending on their goal. Thus, in SRS, the detection logic is designed to work at the file I/O level. It is rule-based, monitoring the activity of local file system. There are two rules in the logic, as listed below.

Rule 1) File removal for over 30% of data files in local file system; and

Rule 2) Change of file extension in any physical file path.

In Rule 1), if data files are encrypted under ransomware attack, their original file would be deleted. If there are more than 30% of data files deleted in a short period of time, it is assumed that ransomware is taken place. This does not disturb the normal operation of the edge server as the probability of the deletion is very low. In Rule 2), if the file extension of data files is renamed into something unrecognizable in the system, it is also assumed to be under ransomware attack. Figure 2 outlines the decision making of the Detection logic.

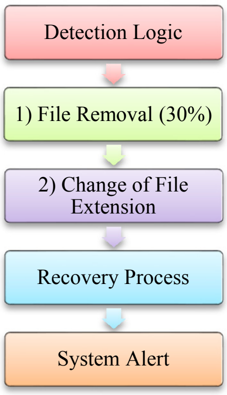


Figure 2: Ransomware Detection Logic.

When SRS finds an edge server under ransomware attack, the recovery process will be activated automatically and the system alert will be sent out for ransomware removal.

3.4 Recovery Process

When ransomware attack is detected, the recovery process will be activated instantly. A secure network connection will be established to the backup node, requesting for the IoT data for a certain period of time. It is noted that the recovery process will not be activated when the Internet connection is not available. When the data is received, it will be sent to the backend server after pre-processing.

4 EVALUATION

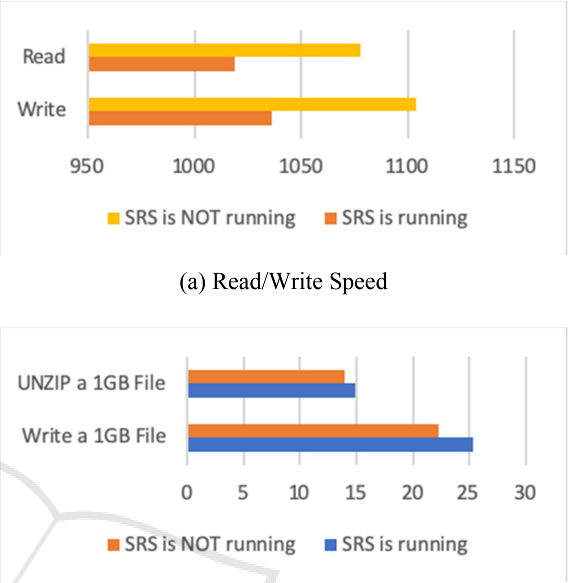
A simulation test has been conducted on SRS for verifying its correctness. In the test, for verification purpose, a standard computer running windows 7 SP2 operating system is used as the test platform. The core hardware component of the computer is listed in Table 1. It is noted that the computing power for an edge server depends on the need of the IoT system. It does not need to be a high-ended computer. A raspberry pie device may be used as an edge server.

Table 1: Configuration of Simulation Test Platform.

Component	Type
CPU	i7 – 6700 HQ (8-core)
RAM	8 GB
Storage	SSD

As a system service running in the background, SRS takes system resources when it works. To ensure that it does not degrade the system performance, the test aims to verify the disk performance (I/O), CPU performance, and the disk average transfer rate of the system.

4.1 SRS Performance



(a) Read/Write Speed

(b) Writing Big File

Figure 3: Disk Performance.

In the simulation test, as can be seen in Figure 3(a), the disk read/write speed is about 1077.6MB/1104.03MB per second when SRS is not running. When it is running, the read/write speed drops down to 1019MB/1036.01MB per second, which is about 5.44%/6.16%. In the meantime, the access (read/write) time increases, while SRS is running, from 0.5ms/0.289ms to 0.519ms/0.296ms, which is about 3.8%/2.42%.

To further test the disk performance, a binary file of 1 GB in size was used in unzipping and writing operations. (See Figure 3(b)) When SRS is not running, it takes about 13.93 seconds in unzipping (for testing the read speed). When SRS is running, it takes about 14.92 seconds. There is around 1 second time putting on the unzip operation by SRS which is about 7%. On the other hand, it takes about 22.25 seconds in the writing operation when SRS is not running. When it is running, it takes about 25.32 seconds. It is about 3 seconds (13.8%) of time putting on the operation by SRS. The performance in disk writing is slightly degraded.

Moreover, it revealed that the CPU performance is slightly degraded. It takes around 0.2% (1.195% - 0.917%) of CPU time when SRS is running. The test

Table 2: Summary of the Test.

	SRS is NOT running	SRS is running	Result
Disk Read/Write Speed (MB per second)	1077.6MB/s, 1104.03MB/s	1019MB/s, 1036.01MBs	5.44%/6.16% DROP
Disk Access Time (Millisecond)	0.5ms/0.289ms	0.519ms/0.296ms	3.8%/2.42% INCREASE
UNZIP a 1GB File (Second)	13.93s	14.92s	7% INCREASE
Write a 1GB File (Second)	22.25s	25.32s	13.8% INCREASE
CPU Usage (%)	0.917%	1.195%	0.2% INCREASE
Disk Average Transfer Rate (Second)	1.051s	1.253s	0.2s INCREASE

also showed that the disk average transfer rate is about 1.051 seconds when SRS is not running. When it is running, the transfer rate is about 1.253 seconds, which is about 0.2 seconds increase in the transfer operation. Table 2 summarized the result of the test.

The result shows that SRS take insignificant system resources from the system in terms of the disk performance, CPU performance and disk average transfer rate. However, it ensures the continuity of the operation of edge server after ransomware attack.

5 CONCLUSIONS

Ransomware is arbitrary and destructive. Its harm to IoT systems may create disastrous consequences. To avoid the disaster happens, preventive measures are commonly employed. SRS is one of the solutions that eases the damages caused to edge servers by ransomware. It ensures the continuity of the operation of edge server after ransomware attack. It also ensures that the data integrity and accuracy issues would not be raised in IoT systems. In addition, SRS takes insignificant system resources for its operation that does not degrade the performance of the edge server.

REFERENCES

Fernández, C. M., Rodríguez, M. D., & Muñoz, B. R. (2018). An Edge Computing Architecture in the Internet of Things. In *ISORC, 21st International Symposium on Real-Time Distributed Computing, IEEE, Singapore, 99-102.* https://dx.doi.org/10.1109/ISORC.2018.00021

Lopez, P. G., Montresor, A., Epema, D., Datta, A., Higashino, T., Iamnitchi, A., Barcellos, M., Felber, P. & Riviere, E. (2015). Edge-Centric Computing: Vision and Challenges. *SIGCOMM Computer Communication Review, 45(5), 37–42.* https://dx.doi.org/10.1145/2831347.2831354

Shi, W., Cao, J., Zhang, Q., Li, Y. & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet Things Journal, 3(5), 637–646.* https://dx.doi.org/10.1109/JIOT.2016.2579198

Statista. (2017). Edge Computing Market Size Forecast in the United States From 2017 to 2025, by Segment (in Million U.S. Dollars). Retrieved from https://www.statista.com/statistics/909308/united- states-edge-computing-market-size-segment/

Statista. (2020). Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025. Retrieved from https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

Xiao, Y., Jia, Y., Liu, C., Cheng, X., Yu, J. & Lv, W. (2019). Edge Computing Security: State of the Art and Challenges. In Proceedings of the IEEE, 107(8), 1608-1631. https://dx.doi.org/10.1109/JPROC.2019.2918437

Alrowaily, M. & Lu, Z. (2018). Secure Edge Computing in IoT Systems: Review and Case Studies. In *SEC IEEE/ACM Symposium on Edge Computing, Seattle, WA, 440-444.* https://dx.doi.org/10.1109/SEC.2018.00060

Hafeez, I., Ding, A. Y., Suomalainen, L., & Tarkoma, S. (2016). Demo Abstract: Securebox — A Platform to Safeguard Network Edge. In *SEC IEEE/ACM Symposium on Edge Computing, Washington, DC*, 117-118. https://dx.doi.org/10.1109/SEC.2016.12

Caprolu, M., Pietro, R. D., Lombardi, F. & Raponi, S. (2019). Edge Computing Perspectives: Architectures, Technologies, and Open Security Issues. In *EDGE IEEE International Conference on Edge Computing, Milan, Italy, 116-123*. https://dx.doi.org/10.1109/EDGE.2019.00035

Zhang, J., Chen, B., Zhao, Y., Cheng, X. & Hu, F. (2018). Data Security and Privacy-Preserving in *Edge Computing Paradigm: Survey and Open Issues. IEEE Access, 6*, 18209-18237. https://dx.doi.org/10.1109/ACCESS.2018.2820162

O'Gorman, G., & McDonald, G. (2012). Ransomware: A Growing Menace. *Symantec Corporation*. Retrieved from https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/ransomware-a-growing-menace.pdf

Gazet, A. (2010). Comparative Analysis of Various Ransomware Virii. *Journal in Computer Virology, 6(1), 77-90*.

Johansen, A. G. (2018). What is Ransomware and How to Help Prevent Ransomware Attacks. *Symantec*. Retrieved from https://us.norton.com/internetsecurity-malware-ransomware-5-dos-and-donts.html

Adhikari, R. (2016). Gooligan Ransacks More than 1M Android Accounts. *TechNewsWorld*. Retrieved from https://www.technewsworld.com/story/84132.html

Anley, C. (2002). Advanced SQL Injection in SQL Server Applications. *CGI Security*. Retrieved from https://www.cgisecurity.com/lib/advanced_sql_injection.pdf

Martin, M. & Lam, M. S. (2008). Automatic Generation of XSS and SQL Injection Attacks with Goal-Directed Model Checking. In *17th Conference on Security Symposium, Berkeley, CA, USA, 31-43*.

McIntosh M., & Austel, P. (2005). XML Signature Element Wrapping Attacks and Countermeasures. In Workshop Secure Web Services, 20–27. https://dx.doi.org/10.1145/1103022.1103026

Tse, R., Aguiari, D., Chou, K.-S., Giusto, D., Tang, S.-K., & Pau, G. (2018). Monitoring Cultural Heritage Buildings via Low-Cost Edge Computing/Sensing Platforms: The Biblioteca Joanina de Coimbra Case Study. In *GOODTECHS, 4th EAI International Conference on Smart Objects and Technologies for Social Good. ACM Press. New York, NY, 148-152*. https://dx.doi.org/10.1145/3284869.3284876

Tse, R., & Pau, G. (2016). Enabling Street-Level Pollution and Exposure Measures. In *MobiHealth, 6th ACM International Workshop on Pervasive Wireless Healthcare. ACM Press. New York, NY, 1–4*. https://dx.doi.org/10.1145/2944921.2944925

Aguiari, D., Delnevo, G., Monti, L., Ghini, V., Mirri, S., Salomoni, P., Pau, G., Im, M., Tse, R., Ekpanyapong, M., & Battistini, R. (2018). Canarin II: Designing a Smart e-Bike Eco-System. In *CCNC, 15th IEEE Annual Consumer Communications Networking Conference. IEEE*. https://dx.doi.org/10.1109/CCNC.2018.8319221