Ransomware Prevention System Design based on File Symbolic Linking Honeypots

Publisher: IEEE Cite This PDF

Danyil Zhuravchak ; Taras Ustyianovych ; Valery Dudykevych ; Bogdan Venny ; Khrystyna Ruda **All Authors**

Abstract:
The data-driven period produces more and more security-related challenges that even experts can hardly deal with. One of the most complex threats is ransomware, which is very taxing and devastating to detect and mainly prevent. Our research methods showed significant results in identifying ransomware processes using the honeypot concept augmented with symbolic linking to reduce damage made to the file system. The CIA (confidentiality, integrity, availability) metrics have been adhered to. We propose to optimize the malware process termination procedure and introduce an artificial intelligence-human collaboration to enhance ransomware classification and detection.

Published in: 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)

Date of Conference: 22-25 September 2021

DOI: 10.1109/IDAACS53288.2021.9660913

Date Added to IEEE *Xplore*: 05 January 2022

Publisher: IEEE

ISBN Information:

Conference Location: Cracow, Poland

ISSN Information:

SECTION I.
Introduction

The constant development of digital innovations allows the usage of new ways of storing and transmitting information. Data has become one of the essential assets of various institutions and organizations. The fact of the high data value, especially in corporate environments, has made ransomware attacks one of the biggest threats to business, primarily if the latter is guided by the principles of “data-driven decision making”, “business intelligence”, data mining, big data [1], [2]. That is why many institutions get vulnerable to ransomware attacks in the information technology and digital communications environments. Accordingly, both the development and improvement of methods and procedures for their prevention and detection in real-time are becoming critical.

It is challenging to detect and, in addition, to prevent a ransomware attack due to the complexity of the encryption algorithms used. As soon as the device becomes “infected”, other systems and components connected to the network also become vulnerable. However, detection and/or prevention of the threat in the network or endpoint is possible in different ways: anti-virus and anti-protection systems against malware programs; monitoring; using the principle of “honeypot” or “decoy”4]. For example, the Sysmon service is able to collect data on suspicious WinAPI calls and use system log files to report data encryption. However, a solution such as network/file system monitoring may not be effective to detect and prevent damage from a ransomware attack.

In this paper, we propose honeypot usage for ransomware attacks identification and data loss prevention. The research novelty is the application of a custom set of rules for honeypots deployment to make them attractive to intenders and increase malware identification probability. Our method is based on multiple symbolic links creation for a single honeypot file to minimize resource utilization. Instead of controlling many decoy files on incursion, we do the inspection just for one that contains N-number of symbolic links.

Honeypot is a computer system intended as a decoy for cyberattacks and can be weakened intentionally to become more vulnerable to threats than other file system objects [5]. We aim to solve the problem of low ransomware detection efficiency by using honeypots. Also, our research goal is to review various methods for ransomware prevention, present their tradeoffs and possible ways for improvements so that ransomware detection can get improved. While analyzing honeypot tradeoffs, we characterize the optimal number of honeypots to meet system performance requirements. The proposed solutions can help enhance corporate security with distinct types of innovations, making ransomware prevention software more flexible and accurate than currently available options.

A ransomware attack consists of several successive steps during which the threat detection can take a few hours to a month or more. Even security experts are not always able to decrypt files through complex decryption keys [6]. Mostly, that is, the virus is identified based on changes to the main files, which causes more damage than in the case of using “honeypot” (“decoy”). After all, despite the existence of various means of ransomware threat detection, numerous incidents occur, and even the most highly cyber-protected companies are exposed to this threat and have a specific set of vulnerabilities [7], [8]. That is why an effective and integrated system capable of analyzing and protecting reliable control of file/network resources is necessary to counter cyber threats. This problem is especially relevant during Industry 4.0 because the extortion program can potentially cause the failure of several devices, reduce the performance of devices, and obtain confidential information illegally [9], [10].

SECTION II.
Recent Literature Analysis

Ransomware is a type of malware that blocks access to the asset or its data or blocks different critical processes. The goal of this action is to racketeer money from ransomware victims [11]. This type of threat acts by the following algorithm:

Ransomware scans filesystem.

This malware type encrypts sensitive data.

Defense from ransomware is a continuous process that requires research and development of detection and prevention daily because the offensive side are continuously developing initial penetration techniques with antivirus bypass tactics.

Recent scientific works reflect new methods in ensuring the maximum security of information systems, effective system analysis techniques, and malicious software detection. However, there is room for improvement in this area. Despite the popularity of behavior analysis-based ransomware detection, there is still a lack of distinct solution designs and techniques to identify threats promptly and eliminate a threat with minimal or no loss. [12]– [14]

Kharraz A. and Kirda E developed a novel ransomware prevention system in their study that provides high resilience. Their approach is based on behavior analysis and requires minimum operating system changes. Their systems monitor I/O applications request patterns for each process to find signs of a ransomware-like behavior. The process gets terminated once abnormal behavior is observed. [15]

This paper aims to improve the efficiency of ransomware detection and prevention using honeypots monitoring and accomplish malware identification based on the interactions with file systems. There is a demand for continuous thread hunting for ransomware because it is still a severe problem. It builds a plan that will detect ransomware attacks based on the interaction with file-based systems.

Table 1 contains a comparison of different ransomware detection and prevention systems.

Shannon's entropy & fuzzy hash technique is the least accurate in our comparison. However, it has 0% of false positives [17].

The AIRad tool is fully based on various statistical and machine learning techniques [18]. The detection rate is very high considering this intelligent system.

DIMAQS (database ransomware detection) is a database with known ransomware [19]. Such a solution is the least efficient when dealing with new malware samples and detecting them because they might not exists in the database.

Table I. Ransomware detection and prevention systems comparison overview: cryptolock, shannon's entropy & fuzzy hash technique, airAD, dimaqs.

Ransomware	Data loss	Detection rate	False positive rate
CryptoLock	0.2% (10/5099)	100%	3.8%
Shannon's entropy & fuzzy hash technique	-	95.7%	0%
AIRad	-	99.54%	0.0005%
DIMAQS	-	100%	0%

SECTION III.
Materials and Methods

In this research, we used the file-based honeypot method for the detection and prevention of ransomware attacks. The basic idea of the honeypot concept is the creation of synthetic files and folders to keep track of changes. This approach to detecting extortion programs should have the lowest false positive rate of the first kind and is the goal for most extortion programs because it is based on behavior analysis. The honeypot method was used in C. Moore's study on Windows-based OS. The author admits that honeypots have limited value because there is no method to control the ransomware to access system objects areas containing decoys [20]. Another research shows that honeypots might be used to study new types of malwares through network logs analysis, and therefore, develop precise security prevention mechanisms [21]. Also, in other authors' research, machine learning is applied, which improves the mechanism of threat detection. A feature of this approach is the use of honeypots as traps for suspicious programs and packages, while machine learning algorithms analyze the behavior and classify viruses [22]. Our method differs from the aforementioned in a such way: application of specific set of rules to weaken the honeypots; single honeypot file monitoring with one/multiple symbolic links deployed in different locations; local

Linux-based environment to conduct the experiments. Also, we analyze the impact of honeypot objects on compute resources. We address a problem of a large decoy targets number that significantly increases the load on the system. That is why it is advisable to apply a substantial and well-defined number of honeypots for ransomware detection. However, in our approach, just a single honeypot is used, which significantly minimizes resource utilization.

An experiment to review our method's performance impact was conducted. On a virtual machine with 2 CPU

Encryption keys are sent to the attacker's server.

Advanced ransomware attacks encrypt system's internal files and spread them via the network to other assets. This virus' approach means that other assets in the same internal network are infected, and more sensitive data are encrypted. cores and 4 memory GiB, less than 1% CPU was consumed. The most resource-consuming component was the code execution during honeypots deployment and malware process termination.

Three ransomware samples that enumerate tiles and start the encryption process were used to validate our proposed approach.

The course of the study consists of several successive steps, starting with the environment preparation. The workflow of the ransomware detection and prevention system in a device file system is shown on Fig. 1. Honeypot files need to be created first with a function that takes two arguments 1) the path where the honeypot should be stored and 2) the file size. The function returns the absolute path to the honeypot.

Then the creation of the symbolic link to point to the honeypot is done. This process is automated with a function that takes two arguments as well - 1) the path to the bait and 2) the path where the recursive creation of symbolic links to the honeypot will occur. The name of the bait files must meet the following requirements:

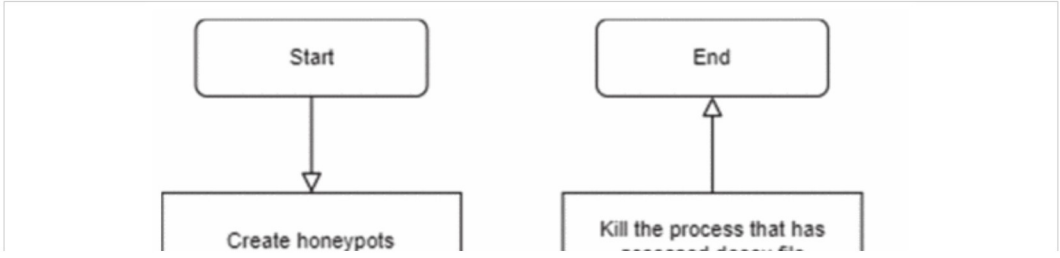
1. to be hidden for users so as not accidental delete action occurs.
2. to have a combination of numbers and special characters so that the file can be located on top in the file system directory.

Once the environment and honeypot files are ready, the monitoring of the individual files is set up. For monitoring, the following options can be used:

1. Watchdogs is a Python library that uses watchdogs to monitor files and directories and generates events. However, it does not log process-related information. So,anothersolution is needed.
2. fuser is a utility on Linux which identifies processes using files or sockets. With “-kw” flags, it will find all writing processes to our honeypot and kill them. The disadvantage of using this tool is a high duration to enumerate all processes and their opened files, so we require to find a more reliable monitoring tool.
3. auditD is the best option for our research monitoring needs. The utility is a system event audit daemon in Linux that creates operations and incidents based on system calls.

Using the auditD tool, we successfully register the write access to honeypot files and automate the monitoring execution. The audit rules are configured to track read/write access to the file system. The next step is to monitor access to honeypot files, which is an ongoing process. The culmination happens when a notification of threat detection is received, and a harmful process that has reached a honeypot is eliminated. The ransomware processes have been successfully destroyed; detection continues.

When the incident is resolved, cybersecurity experts analyze system events and log files to identify the source of the threat. Automation and possibly artificial intelligence applications might be developed in further research to automate this process.



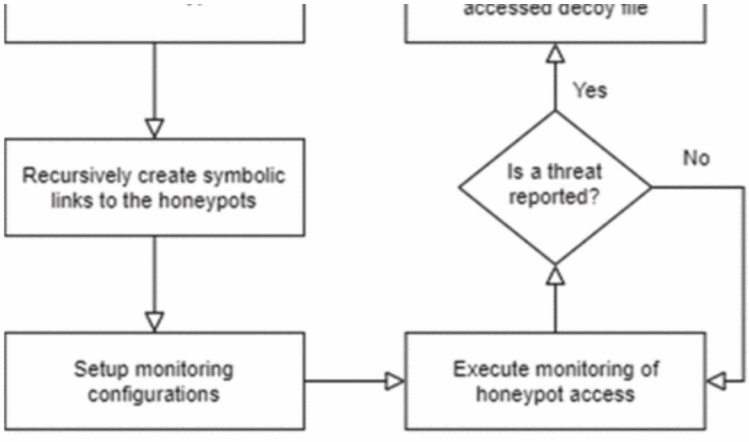


Figure 1.
Ransomware attack's detection and prevention system workflow.

SECTION IV.
Research Results

The standard way of malware sample collection approaches is complex and based on binary extraction from an infected machine or encrypted traffic. These approaches are manual and need people interactions in most cases. Everyday malware vendors are creating new and new samples of ransomware software. Due to that fact, ransomware samples collection and detection approaches described above with human interaction and management are always too late for timely incident response. Based on this assumption, we need a completely automated and programmed malware collection logic to catch these ransomware samples.

Three ransomware samples from the GitHub repository were used to validate the results of the study:

- GonnaCry [23];
- JavaRansomware [24];
- RAASNet [25].

Table 2 describes validations results. The key factor that impacted detection and response rate is the realization of the cryptography algorithm and its encryption speed. For example, the data loss for JavaRansomware is 30 files, 27 of which were our custom honeypots. The kill signal for JavaRansomware had been sent during the encryption process before we were capable to delete the malicious file. The optimal and required detection and response process for our system is the following: the ransomware detection and elimination at the same time when it accesses the file system objects. In the case of JavaRansomware, this goal was not achieved. Accordingly, it is advisable to work on system optimization for rapid detection and elimination of encryption processes. GonnaCry and RAASNet have a low data loss because these ransomware samples are slower in execution than JavaRansomware. It was also easier to terminate these malware processes.

Table II. Results by using detection and prevention systems against ransomware: gonnacry, javaransomware, raasnet

Ransomware	Data loss	Detection rate	False positive rate
GonnaCry	2	100%	0.98%
JavaRansomware	30	100%	1.98%
RAASNet	5	100%	1.19%

In the table above, we tracked the detection rate that is 100% for all experiments and the false-positive rate. After conducting 15 measurable attempts out of 50 total, it turned out that the highest false-positive rate has JavaRansomware. So, the ransomware sample with the highest execution speed is more likely to have a higher false-positive rate than other slow samples.

SECTION V.
Conclusion

The purpose of the study is met by analyzing various ransomware attacks, their consequences, and creating a preventive method based on file-based honeypots using symbolic links. We conclude that ransomware can be effectively detected using our approach of application file symbolic links along with auditD script in a very brief amount of time. However, there are still opportunities to enhance this solution and make it enterprise ready. Specifically, some improvements in the mean time to detect/terminate a suspicious process would be beneficial. We suppose that creating a sufficient number of honeypots based on various calculations will considerably help to improve and/or maintain a good infrastructure condition and protect the system. We are considering using Log management and SIEM systems as an addition to our method and creating an OS daemon and adding more rules for auditD policy to improve ransomware identification precision.

Authors

Figures

References

Download PDFs | Export | References & Cited By

Select All

1. N. Shakhovska, S. Fedushko, N. Melnykova, I. Shvorob, Y. Syerov, “Big data analysis in development of personalized medical system,” *Procedia Computer Science*, vol. 160, pp. 229–234, 2019.
Show in Context CrossRef Google Scholar

2. S. Fedushko, T. Ustyianovych, M. Gregus, “Real-time high-load infrastructure transaction status output prediction using operational intelligence and big data technologies,” *Electronics*, vol. 9, issue 4, 668, 2020.
Show in Context CrossRef Google Scholar

3. Z. A. Genç, G. Lenzini, D. Sgandurra, “On deception-based protection against cryptographic ransomware,” *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, Cham, 2019, pp. 219–239.
CrossRef Google Scholar

4. M. L. Bringer, C. A. Chelmecki, and H. Fujinoki, “A survey: Recent advances and future trends in honeypot research,” *I. J. Computer Network and Information Security*, pp. 63–75, 2012.
CrossRef Google Scholar

5. D. Fraunholz, M. Zimmermann, H. D. Schotten, “An adaptive honeypot configuration, deployment, and maintenance strategy,” *Proceedings of the 19th International Conference on Advanced Communication Technology (ICACT)*, 2017, pp. 53–57, DOI: 10.23919/ICACT.2017.7890056.
Show in Context View Article Google Scholar

6. J. P. Tailor, and A. D. Patel, “A comprehensive survey: ransomware attacks prevention, monitoring and damage control,” *Int. J. Res. Sci. Innov*, vol. 4, no. 15, pp. 116–121, 2017.

Show in Context Google Scholar

7. C. Kwan, *Acer reportedly targeted with \$50 million ransomware attack*, March 22, 2021. [Online]. Available at: https://www.zdnet.com/
Show in Context Google Scholar

8. S. Adler, *Incident of the Week: Garmin Pays \$10 Million To Ransomware Hackers who Rendered Systems Useless*, August 14, 2020. [Online[. Available at: https://www.cshub.com/
Show in Context Google Scholar

9. S. Sibi Chakkaravarthy, D. Sangeetha, M. V. Cruz, V. Vaidehi, and B. Raman, "Design of intrusion detection honeypot using social leopard algorithm to detect IoT ransomware attacks, *IEEE Access*, vol. 8, pp. 169944–169956, 2020.
Show in Context View Article Google Scholar

10. W. Zhang, B. Zhang, Y. Zhou, H. He, and Z. Ding, "An IoT honeynet based on multi-port honeypots for capturing IoT attacks," *IEEE Internet of Things Journal*, 2019.
Show in Context Google Scholar

11. B. Ross, "Ransomware attacks: detection, prevention and cure," *Network Security*, no. 9, pp. 5–9, 2016.
Show in Context Google Scholar

12. V. Dudykevych, I. Prokopyshyn, V. Chekurin, I. Opirskyy, Y. Lakh, T. Kret, Y. Ivanchenko, I. Ivanchenko, "A multicriterial analysis of the efficiency of conservative information security systems," *Eastern-European Journal of Enterprise Technologies. Information and controlling System*, vol. 3, no. 9 (99), pp. 6–13, 2019.
Show in Context CrossRef Google Scholar

13. S. Vasylyshyn, I. Opirskyy, V. Susukailo, "Analysis of the use of software baits as a means of ensuring information security," *Proceedings of the 2020 IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2020*, 2020, vol. 2, pp. 242–245.
Show in Context View Article Google Scholar

14. Z. Hu, Y. Khokhlachova, V. Sydorenko, I. Opirskyy, "Method for optimization of information security systems behavior under conditions of influences," *International Journal of Intelligent Systems and Applications (IJISA)*, vol. 9, no. 12, pp. 46–58, 2017.
Show in Context CrossRef Google Scholar

15. A. Kharraz, E. Kirda, "Redemption: Real-time protection against ransomware at end-hosts," In: Dacier M., Bailey M., Polychronakis M., Antonakakis M. (eds) *Research in Attacks, Intrusions, and Defenses. RAID 2017. Lecture Notes in Computer Science*, Springer, Cham, vol 10453, 2017. https://doi.org/10.1007/978-3-319-66332-6_5
Show in Context CrossRef Google Scholar

16. N. Scaife, H. Carter, P. Traynor, K. R. Butler, "Cryptolock (and drop it): stopping ransomware attacks on user data," *Proceedings of the 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, 2016, pp. 303–312.
View Article Google Scholar

17. Y.S. Joshi, H. Mahajan, S.N. Joshi,, "Signature-less ransomware detection and mitigation," *J Comput Virol Hack Tech*, 2021. https://doi.org/10.1007/s11416-021-00384-0
Show in Context CrossRef Google Scholar

18. S. Poudyal and D. Dasgupta, "AI-powered ransomware detection framework," *Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 1154–1161. doi: 10.1109/SSCI47803.2020.9308387.

Show in Context View Article Google Scholar

19. C. Hagen, A. Dmitrienko, L. Iffländer, M. Jobst, S. Kounev, “Efficient and effective ransomware detection in databases,” *Proceedings of the Annu. Comput. Secur. Appl. Conf(ACSAC)*. 2018.
Show in Context Google Scholar

20. C. Moore, “Detecting ransomware with honeypot techniques,” *Proceedings of the 2016 Cybersecurity and Cyberforensics Conference (CCC)*, 2016, pp. 77–81. doi: 10.1109/CCC.2016.14.
Show in Context View Article Google Scholar

21. V. Sethia and A. Jeyasekar, “Malware capturing and analysis using dionaea honeypot,” *Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST)*, Chennai, India, 2019, pp. 1–4. doi: 10.1109/CCST.2019.8888409.
Show in Context View Article Google Scholar

22. I. M. M. Matin and B. Rahardjo, “Malware detection using honeypot and machine learning,” *Proceedings of the 2019 7th International Conference on Cyber and IT Service Management (CITSM)*, Jakarta, Indonesia, 2019, pp. 1–4. doi: 10.1109/CITSM47753.2019.8965419.
Show in Context View Article Google Scholar

23. *Tarcisio-Marinho*, 2020. GonnaCry Rasomware [Source code]. Available at: https://github.com/tarcisio-marinho/GonnaCry
Show in Context Google Scholar

24. P. Drakatos, 2017. *JavaRansomware [Source code].* Available at: https://github.com/PanagiotisDrakatos/JavaRansomware
Show in Context Google Scholar

25. L. Voerman (leonv024), 2020. *RAASNet [Source code]*. Available at: https://github.com/leonv024/RAASNet
Show in Context Google Scholar

Citations

Keywords

Metrics

Back to Results