Altaf et al. LGURJCSIT 2023

ISSN: 2521-0122 (Online)
ISSN: 2519-7991 (Print)

LGU Research Journal of Computer Science & IT

doi: 10.54692/lgurjcsit.2023.074485

Vol (7): Issue (4), October – December 2023

Unraveling Ransomware in the Digital Battlefield: Threat Analysis and Countermeasures

Mahroosha Altaf[1], Zafar Iqbal[1]* , Adeen Shahid[1]
Depatement of Cyber Security Air University, Islamabad, Pakistan.

Email: mahrooshaaltaf987@gmail.com

ABSTRACT:

Ransomware is a notorious form of malware known for causing severe and permanent damage to its targets. Prompt identification of such attacks is crucial to mitigate the devas- tating consequences they can inflict. According to some reports, the number of ransomware attacks has grown significantly since 2016, with a significant increase in the number of attacks targeting businesses and the military. It is widely considered a major cyber threat at both indi- vidual and organizational levels. Organizations can implement and maintain comprehensive ran- somware mitigation strategies, such as backup, network segmentation, HR education, endpoint protection, and advanced threat hunting. It's worth noting that only some techniques are foolproof. Ransomware has been used in the context of the Russia-Ukraine war, primarily by Russian-backed cybercriminal groups. It has been found that Russian groups have targeted Ukrainian infrastructure and businesses with ransomware attacks, encrypting their systems' data and demanding payment in exchange for the decryption key. These attacks have caused significant disruptions and financial losses as their aim was destruction rather than data breach for the targeted organizations. In this paper, we have analyzed the ransomware used in the Russia-Ukraine war and summarized the most prominent malware involved in the war. We have chosen one of the malware, "Hermetic Ransom", which performed its thorough analysis and created the YARA rule for its detection, prevention, and response.

KEYWORDS: Malware Detection, Ransomware, File Wipers, Cyberwarfare, Cyber-attacks, Static Analysis of Malware, Advanced Cyber-attacks.

1. INTRODUCTION

APTs(Advance Persistent Threats) are highly skilled, persistent cyberattacks, organized and car- ried out by organized, well-funded threat actors. Such attacks employ advanced Tactics, Techniques, and Procedures to compromise user's integrity, accountability, and confidentiality by per- forming malicious activity [1, 2]. Moreover, these employ various evasion techniques to dodge security devices [3]. In Russia Ukraine, war, several such attacks were launched to take control of the organizations and destroy the targeted systems [21]. The attackers left some traces behind, which left doubt that Russia was involved in these attacks to destroy Ukraine. The threat actors involved in the attack presented clear ties with Russian special services like the Russian Federal Security Service (FSB) and Russian Federation (GRU). The first at- tack with the objective of destruction was conducted in February 2022, Hermetic Wiper followed by another espionage attack in the same month, grim plant, graph steel [22, 23]. Malicious attachments via emails were used to deliver this malware. One of the most well-known malware attacks in this conflict is the "BlackEnergy" malware which has been used to target the Ukrainian govern- ment and infrastructure organizations [24]. This malware is believed to have been used in several attacks, including the 2015

power outage in Ukraine that affected over 225,000 people. Another malware used in this conflict is the "SandWorm" malware which targeted the Ukrainian government and military and organizations in other countries [25]. This malware is believed to have been developed and used by the Russian military.

Additionally, there have been reports of ransomware and phishing attacks being used to target organizations in both Russia and Ukraine, as well as other countries involved in the conflict. According to Getstra, 1.7 million ransomware attacks happen every day and the average cost of a ransomware attack is 1.85 Million Dollars [26].

The WannaCry ransomware assault caused a 100 million dollars loss for the National Health Service (NHS) [27]. In the last five years, ransomware assaults have increased by 13%, according to Ver- izon's 2022 data breach report. Nearly 236.7 mil- lion ransomware assaults occurred worldwide in the first half of 2022. Static Analysis is meant to be an analysis of malware without running it, we look for strings, imports, sections, file size, and time stamps in this phase. Static analysis is further divided into signature-based and heuristic-based, In signature-based detection is done using hashes like (MD5, SHA-1, and SHA-256), and Heuristic- based detection is based on some rules, algorithms, and heuristics for identifying malware. In dynamic analysis, we run malware samples in a controlled environment and observe their activity like file system activity, system calls and API monitoring, registry activities, network activity, etc. 10% of these breaches are due to ransomware.

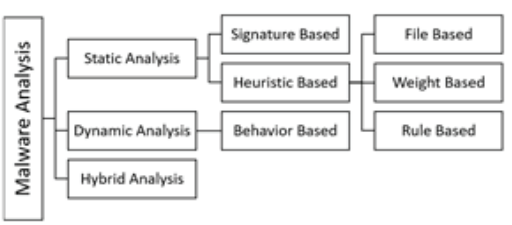

Figure 1: Malware Analysis Techniques

This research identified various artifacts of the Hermetic ransom through static and dyna mic analysis. Moreover, a comprehensive YARA rule is created, which can be used for prevention, and detection of the Hermetic ransom. We have used a virtual environment for an- alyzing the hermetic ransomware, inside the virtual environment we have used various tools to extract static characteristics of malware, and also we have run malware to analyze it dynamically, we have extracted artifacts and Indicators of Compromise which can be used for detection and prevention of malware's in the future. APTs are advanced persistent threats that exploit new vulnerabilities so for this reason, traditional security devices can't detect such attacks, so we need manual analysis for such threats [11].

To combat APTs, cyber threat intelligence (CTI) is considered the most effective tool and static and dynamic analysis is considered the best source of CTI collection [12]. In this research, we performed a detailed analysis of Hermetic Ransomware and extracted CTI shared for cyber threat management, i.e., for cyber threat prevention, detection, and re- sponse.

No previous study has been found on practical manual analysis of war malware, we have performed a detailed analysis of one of the very critical mal- ware of the Russia-Ukraine war. aims and objectives of this paper are:

- To study malware used in the Russia-Ukraine war.
- Manually analyze hermetic ransomware, extract artifacts and IoCs (Indicators of Compromise).
- To write a YARA rule of malware for detection and prevention of such threats in the future.

The rest of the article is organized into the following sections. Section II provides the Literature review. Section III presents the experiment work. Section IV shares the results. Finally, the conclu- sion, along-with future work, is presented in Section V.

2. LITERATURE REVIEW

Ransomware has become a major concern for cybersecurity experts due to the rapid increase in attacks and the emergence of new, advanced variants that can evade traditional security measures such as antivirus and anti-malware software [17]. The emergence of new ransomware variants has seen a significant uptick in recent years, making it crucial to distinguish it from other forms of malware to protect computer systems from ransomware-based attacks. Despite some similarities with other mal- ware, ransomware has distinct features, such as performing a high volume of file operations within a short timeframe to lock or encrypt files on the targeted machine. Traditional detection methods that rely on signatures may struggle to detect previously unseen, zero-day ransomware,

making them an inadequate defense against the potential risks of unknown ransomware. [18].

In researchers combined network traffic analysis and machine learning techniques and solved the problem of pinpointing malicious apps by identifying malicious network behavior [1]. Since only a small traffic portion is malicious and the rest is not harmful, the imbalanced data problem was observed. To cater to that, several imbalanced classification algorithms like SVM were implemented. The accuracy rate observed was up to 99.9%. However, the accuracy of machine learning classifiers declined as the problem of imbalance aggravated. Ransomware has been a prevalent and ongoing threat for the past decade and is a significant concern today. One commonly used method to identify malware is API call-based analysis, which examines the suspicious activities of a program during execution. However, many detection methods need to take into account the importance of identifying key API calls. Emphasizing the significance of key features in API call analysis is crucial for building a robust machine-learning model, as these features serve as the building blocks for such models [19].

In this research, an efficient detection and analysis technique for ransomware was proposed and demonstrated through a case study. The study's re- sults revealed that by utilizing the proposed method, valuable information about the attacker could be obtained by analyzing the behavioral characteristics of the Onion ransomware. This paper also provides a comprehensive overview of the ransomware threat. It offers an examination of the various methods and techniques used for the detection and analysis of ransomware attacks [20]. Researchers proposed two approaches, Mode-A and Mode-B, for malware detection, combining the features of both static and dynamic methods in [2]. In Mode A, for static analysis, permis- sions, native permissions, intent-priority setting, and function calls from the test app were extracted. For dynamic analysis, the test app was uploaded to the sandbox server. The resulting accuracy of 93.4% was observed. To get better results, both methods were combined in the Mode-B system design approach. Vectors from static and dynamic features were merged and extracted. Static features outnumbered dynamic features, so the weights were adjusted for the disparity. Using an SVM algorithm, results show an accuracy of 0.995 and 0.974 for detecting non-split (known) and ten-fold (unknown) malware, respectively. Authors [3] specifically focused on the detection of Android ransomware apps by proposing a smart classification algorithm called SSA-KELM, which is based on a metaheuristic swarm Intelligence algorithm called Salp Swarm Algorithm (SSA) and a machine learning algorithm called Kernel Extreme Learning Machine (KELM). The data set and ransomware detection framework were developed by extracting features like API calls and permissions. An accuracy of 98% was observed with a 2% ratio of false positive rate. The resulting accuracy is higher than the conventional classification algorithms like Naïve Bayes (NB) and C4.5.

The paper under analysis, proposed a Blockchain-based framework for detecting malware detection techniques [4]. The framework has a private blockchain (internal and external) and a consortium blockchain for a final decision. To protect smartphones from these malicious Apps, different antimalware solutions have been proposed to detect the existing malware and the new generation of malware which are zero-day attacks. A process of recording transactions and tracking assets in a network of databases across different entities is Blockchain. Three types of Blockchains are public, private, and consortium. Consortium and private blockchains are combined to produce a framework for detecting the application which spreads malware on Mobiles. Private Blockchain consists of internal and external blockchains. An internal blockchain performs static and dynamic analysis and stores extracted features. In contrast, an external private blockchain is used for storing the results after detection for the current version of applications in blocks. Based on detection results, the consortium blockchain provides the mechanism of the final decision. Previously, different schemes and methods had been used to detect malicious Apps like local final analysis, static and dynamic analysis, and detection of anomalies in power usage batteries and operating systems. For all apps, one dedicated internal private Blockchain was used to reduce the complexity due to the huge number of applications in the market. Artificial intelligence-based DE and signature-based DE are used for different purposes in detecting malicious applications. Thus, using B2MDF for detecting malicious applications before downloading reduces the false positive rate. It also provides useful features for third parties to make antimalware solutions for detecting

malicious applications in Appstore because B2MDF uses a general detection engine applicable to many machine learning algorithms. The authors held a forensic research study on spyware found in Android devices. Android Operating System is a frequently used OS worldwide; almost 76.1% of people use it globally. This makes it easier for the attacker to target people's Personal Identifiable Information (PII). Although Google has launched a machine learning security application called Play Protect Service (PPS), the threat of being compromised by such spyware remains. This study aims to probe and draw hypotheses on the findings of the method which enabled this spyware to penetrate a device and what the intent is. As the PPS service uses machine learning to detect the malicious application, at first, when the app was installed on a device, it was unable to detect its hostility of it. However, when the app was installed on another device after a significant duration, it flagged it as suspicious; following the third time, PPS did not install the application [6].
Authors shared a review on the use of deep learning in Android Malware Detection in [7]. This study aimed to thoroughly review the use of deep learning in Android Malware Analysis concerning analysis type. It organizes a detailed Android malware analy- sis review using deep learning with static, dynamic, and hybrid analysis. Concept drift is an open issue, as many obsolete datasets are still being used, such as DREBIN. Deep Refiner approached the problem and suggested continual up-gradation of the model [9]. This research produced 97.74% malware detection accuracy. Moreover, deep learning security challenges are a major concern as deep learning needs training datasets to learn behavior; it is vulnerable to the injection of a malignant code or dataset, which could corrupt the entire dataset, lead- ing to erroneous results. Distillation and retraining are two techniques that need to be traversed to de- vise an effective model against adversarial attacks. Static android analysis dominated the existing work.Alazab et al. proposed a system for detecting malware applications vastly available on the Google Play Store, App China, and Anzhi. In this study, the samples from the real world were collected, and then the application programming interface (API) calls of those apps were studied to track malicious behavior. This study aims to devise arobust system to automate the malware detection process and increase the detection rate by conduct- ing an empirical study of ten supervised machine- learning algorithms. The study of the proposed system achieved a detection accuracy of 98.1% with a classification time of 1.22 s when using the Chi-Square and Simple Logistics algorithms. Mobile malware detection is a complicated task regarding the imposition of mining techniques. The study has achieved remarkable results using power- ful machine-learning algorithms [8].
Authors in created a multi-layer Android malware detection tool called Deep Refiner by applying deep neural networks [25]. Deep refiner was compared with a single classifier detection system known as Stormdroid, with multiple anti-virus scanner that was signature-based. The experiment concluded that the deep refiner outperformed the stormdroid and scanners. The paper also tested the tool against obfuscation techniques, and it was shown that the deep tool refiner also detected ob- fuscated applications that are malicious for Android systems. The deep refiner has an accuracy of 97% with TPR equal to 97% and FPR equal to 2.5% (which is low compared to Stormdriod, which is 7.5%. Deep Refiner approached the problem and suggested continual up-gradation of the model. This research produced 97.74% malware detection accu- racy. Moreover, deep learning security challenges are a major concern, as deep learning needs training datasets to learn behavior. It is vulnerable to the injection of a malignant code or dataset, which could corrupt the entire dataset, leading to erroneous results. In [10], the main aim of this research was to check the efficacy of a tool on Android devices in termsof different evasion techniques [3]. So a systematic framework was established for Droid Chameleon with different transformation techniques. Then pos- sible solutions were also produced to improve the current state of malware detection techniques in mobile phones. Ten different anti-malware tools were evaluated for Android systems to check their resilience for the transformation of malware; for this purpose, Droid Chameleon was created, which has multiple transformation techniques. It was found that anti-malware product is vulnerable to transfor- mations. Secondly, 43% of signatures detected were not based on source code-level artifacts. Thirdly, 90% of the signatures did not require static anal- ysis of byte code, and less than ten anti-malware used static analysis. Lastly, anti-malware tools have now shifted to content-based detection rather than

signature-based. The authors introduce an approach for defining Android malware dependent on API (in all request permissions and packages) in [11]. The proposed scheme will discuss the similarities among different families of malware that can then be classified to check the risk factor in mobile devices. Since the Android app uses many APIs, three different grouping techniques were proposed to select the most valuable API to help identify Android device malware applications. The fastest algorithm was K-NN and a random tree; both took almost 0.2 sec and about 0.7 sec for the training and testing phase, but overall, the system is reliable. A malware detection technique for An- droid is proposed based on weight measurement and feature selection (FDS) in [12],. It will measure the impor- tance of each feature, then calculate the weight for each and establish the optimal weight for improving the accuracy. This will reduce the computation of irrelevant features. The author's main motivation was to propose malware detection for Android based on weight measurement and feature selection. This paper focuses on the shortcomings of previous schemes. The weight measurement will improve the accuracy of the results. This will reduce the computation of irrelevant features and provide an accuracy of up to 90%. In machine learning and model identification, data pre-processing is essential for dealing with categorization problems. The processing of large data sets reduces the classification model precision while complicating computer processing in terms of time and space. Therefore, it is essential to provide a suitable mechanism for choosing qualities. In the paper, genetic algorithm selection and mutation operators are protected using a machine-learning ap- proach described in this article to address functional selection. The suggested approach uses population estimates in the training set and is adaptable. This research study demonstrates the benefits of an adap- tive hybrid approach to the evolutionary algorithm for resolving the theoretical selection of challenges. The efficacy of their overall proposed framework has been calculated in terms of accuracy in mean and standard deviation values. The algorithm is developed and studied by adjusting the population size and mutation rate to find the best variable rate for dealing with this type of problem[13]. The majority of Android malware detection now used to combat the rising volume of malware is server-side. Powerful computational resources pro- vide more thorough protection for app marketplaces than by relying solely on user detection. Aside from the apps offered by the official market (the Google Play Store), end users are constantly facing major security risks from apps from unauthorized marketplaces and third-party resources. Network transmission has a lot of overhead. Thus download- ing the app first, then uploading it to the server side for detection takes a lot of time. Security vulnera- bilities posed by attackers also harm the uploading procedure. Therefore, the necessity for a last line of protection on mobile devices is great. Using spe- cialized deep neural networks to provide a real-time and responsive detection environment on mobile devices, we present an efficient Android malware detection system, MobiTive, in this research. A pre-installed option is called MobiTive, instead of employing an app monitoring and scanning engine after installation, which is more secure and useful. MobiTive can offer dependable detection accuracy and quick response [14].

3. EXPERIMENTS

We have selected a well-known ransomware from Russia Ukraine war and performed its detailed analysis. For a thorough ransomware analysis, we employed static and dynamic analysis.

3.1. Ransomware Selection

We have selected ransomware from Russia Ukraine war and performed its detailed analysis [41]. For a thorough ransomware analysis, we em- ployed static and dynamic analysis. Cuckoo sandbox is used for automated analysis, we employed various tools like PE studio, PEview, and for code analysis, we employed IDA Pro [28, 29,30]. Ransomware is a common type of malware used against Ukraine by Russia in war. We selected Hermetic Ransom for analysis [42]. This ransomware is employed to enumerate available drives, collecting a list of directories and files except for the Windows and Program Files folders. Hermetic ransomware is a type of ransomware that uses an advanced form of encryption to make it difficult or impossible for victims to recover their files without paying the ransom. One of the key features of hermetic ran- somware is that it uses multiple layers of encryption, making it much harder to decrypt the files with- out the decryption key. Hermetic ransomware also often includes anti-debugging

and anti-tampering mechanisms that make it more difficult for security researchers and law enforcement to analyze and understand the malware. We have used a Windows 10 VMware machine for the analysis of the sample, we have used Pestudio, Peview, PeID, exeinfope, DetectItEasy, regshot, and others. For dynamic anal- ysis, we have used Sysinternals tools like Process Explorer, Process Hacker, and Process Monitor. For automated analysis, we have used Cuckoo sandbox.

3.2. Behaviour

Go programming language is used to write Hermetic Ransom [31]. It enumerates accessible discs, gathering a list of directories and files, except the Windows and Program Files folders, as shown by CRO wd strike's investigation.

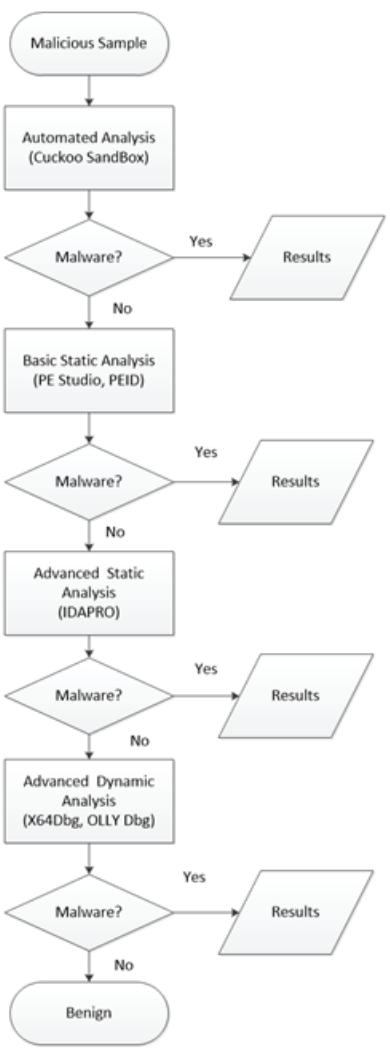

Figure 2: Flow Chart

The ransomware operator's email address and the encrypted JB exension are used to rename specific file types, and the file contents are subsequently encrypted with the AES algorithm. In the Desktop folder, the ransomware also produces a readme.html file con- taining a ransom letter containing the perpetrator's contact information. Files that have been encrypted can be recovered since the encryption method is relatively laborious and has implementation flaws. This issue shows that Hermetic Ransom was most likely utilized as a diversion rather than a true ransomware extortion effort, along with a political message discovered within and distribution timing comparable with Hermetic Wiper.

A. Attack Reported

Hermetic ransom Attack was reported on Feb 23, 2022. It was used in cyberattacks against Ukraine. This research analyzed the ransomware with both static and dynamic techniques; details are provided in the ensuing paragraphs.

3.3. Static Analysis

We The file has a total of 68 indicators showing that it is malicious. The size of the file is suspicious, which is 0 bytes. Strings have some blacklisted flags in the file; the number of flags is 23. Extensions like ransomware and wiper are found in files during static analysis. Many suspicious subsections are present in malware, i.e. /4, /18, /30, /43, /59, /75, /94, /106, and. symtab. The size of the header is suspicious; many strings are more than 1KB, but the size of the file is 0 bytes, which is suspicious. The compiler stamp of the file (Thu Jan 01 00:00:00 1970 — UTC) is suspicious. The checksum is also invalid. The count of libraries (3) and imports (32) indicates that this file is malware. The file targets the OS version Windows NT 4.0. Address Space Layout Randomization is also off here in the file, which means that this file is malicious, as it is stopping the security technique of OS. Code integrity, stack-buffer-overrun-detection (GS), and control flow guard of the file are also off, which gives us the idea that the file is attempting to turn off all security mechanisms that can detect it. Cryptographic functions are also used here in the file. Hashes of the file are:

- MD5: d5d2c4ac6c724cd63b69ca054713e278
- SHA-1: f32d791ec9e6385a91b45942c230f52a ff1626df
- SHA-256: 4dc13bb83a16d4ff9865a51b3e4d24112327c526c

1392e14d56f20d6f4eaf382.

3.4. Network Behaviour

Cuckoo sandbox, is used for dynamic analysis [33]. Details of our findings are shared in the following subsections.

A. Contacted IPs

It's important to know the con- tacted IPs of malware because they can provide valuable information for identifying and mitigat- ing the malware, such as Command and Control (C&C) server identification, Attribution, Indicator of Compromise (IOC), Compliance, and network behavioral analysis.

Contacted IPs	Contacted IPs
13.107.39.203:80 (TCP)	a83f:8110:0:0:0:6002:0:0:53 (UDP)
13.107.4.50:80 (TCP)	a83f:8110:0:0:1400:1400:2800:3800:53 (UDP)
142.250.195.164 (ICMP)	a83f:8110:0:0:9902:0:0:0:53 (UDP)
172.217.14.195:443 (TCP)	a83f:8110:1800:0:0:0:0:53 (UDP)
192.168.0.10:137 (UDP)	a83f:8110:2002:0:0:0:0:53 (UDP)
192.168.0.1:137 (UDP)	a83f:8110:2800:1800:4000:1800:1800:100:53 (UDP)
192.168.0.22:137 (UDP)	a83f:8110:508:10ff:70a:12ff:70a:12ff:53 (UDP)
192.168.0.66:137 (UDP)	a83f:8110:7000:7000:7200:6500:7000:2e00:53 (UDP)
20.62.24.77:443 (TCP)	
20.80.129.13:443 (TCP)	
20.99.132.105:443 (TCP)	
20.99.133.109:443 (TCP)	
20.99.184.37:443 (TCP)	
23.216.147.64:443 (TCP)	
23.216.147.76:443 (TCP)	
23.40.197.137:443 (TCP)	
23.40.197.184:443 (TCP)	
8.8.8.8:53 (UDP)	

Figure 3: Contacted IPs

B. Contacted Domains

Contacted domains are an important aspect of malware analysis, as they can provide insight into the command and control infrastructure used by the malware. This information can be used to track the malware's spread and identify other infected systems. Additionally, identifying the contacted domains can be used to block or sinkhole the domain, which can disrupt the malware's ability to communicate with its command and control servers and limit its ability to spread.

Contacted Domains
240.143.123.92.in-addr.arpa
254.55.250.8.in-addr.arpa
8.143.101.95.in-addr.arpa
82.250.63.168.in-addr.arpa
arc.msn.com
prda.aadg.msidentity.com
prod.ingestion-edge.prod.dataops.mozgcp.net
telemetry-incoming.r53-2.services.mozilla.com
update.googleapis.com
windowsupdatebg.s.llnwi.net

Figure 4: Contacted Domains

3.5. YARA Rule

A Yara rule is designed to detect the existence of a Hermetic Ransom on a system, as can be seen below. It is worth mentioning that the provided rule is a simplified example of a Yara, but for real-world ransomware identification would require a more extensive rule. The String section provides the domain name and ransomware name strings noted during the static analysis of the Hermetic. The code in the Condition section will take the above strings and trigger an alarm if the above strings are detected in a file/ executable.

Rule HermeticRansomDetection {
Meta:
Description = "Yara rule for HermeticRansom"
Author = "Mahroosha"
strings:
$networkTraffic =
"windowsupdatebg.s.llnwi.net"
$hMarker = "HermeticRansom" wide ascii
condition:
networkTraffic, hMarker
}

3.6. Detection and Prevention

- Keep systems and software updated and patched.
- Use endpoint security techniques such as behavior-based detection to identify and block hermetic ransomware.
- Backup all your data securely and outsource it, which can be recovered after a ransomware attack.
- Deploy Intrusion detection and prevention systems.
- Scan your systems daily for vulnerabilities and patch them immediately.
- The organization should develop an incident response plan also ensure that your employees are trained accordingly.
- Machine learning models can also be used for malware detection, i.e. Random Forest, Gradient Boosting Machines, Support Vector Machine additionally ensemble learning can also be used for more accurate results.

4. FUTURE WORK

Looking ahead, our future work includes the development of a machine learning model and Artificial Intelligence algorithm for the detection of ransomware on a network before it reaches the host. We will work on malware detection over encrypted traffic over the network. Additionally,

different methods to enhance the efficiency and scalability for detection of ransomware techniques shall be explored i.e. bypassing checks like isDe- buggerPresent, to ensure effectiveness in rapidly changing thread-landscapes.

5. CONCLUSION

In conclusion, our paper provides a comprehensive overview of important Ransomware detection techniques for the early detection of malicious apps in the App Store. Different techniques were explored in this paper including machine learning methods, Blockchain-based frameworks, forensic analysis, artificial immunity approaches, and meta-heuristic intelligence algorithms. Malware from the Russia-Ukraine war was employed to conduct static and automated analysis. Valuable insights were collected for prevention, detection, and response to Hermetic ransomware. We have studied various malware used in the Russia-Ukraine war. We have selected hermetic ransomware for manual analysis we have extracted artifacts and IoCs. We have selected Windows 10 Virtual Machine for analysis and used various tools for static and dynamic analysis, contacted IPs and domains have been found and listed in Figure3 and Figure4 and the YARA rule has been written. Detection, prevention, and responsive measures have been given.

6. REFERENCES

[1] Z. Iqbal et al., "STIXGEN-a novel framework for au- tomatic generation of structured cyber threat information," In *2018 International Conference on Frontiers of Information Technology (FIT),* (pp. 241-246), IEEE, (2018, De- cember).

[2] Z. Iqbal and Z. Anwar, "Ontology Genera- tion of Advanced Persistence Threats and their Automated Analysis," Vol. 9, No. 2, pp.68-75, 2016.

[3] S. Mirza et al., "A malware evasion technique for au- diting android anti-malware solutions," In *2021 IEEE 30th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE),* (pp. 125-130), IEEE, (2021, October).

[4] M. Connell and S. Vogler, "Russia's approach to cyber warfare," *Arlington, VA: CNA,* (pp. 1-29), 2017.

[5] T. Kellermann and R. Murphy, "Modern bank heists 3.0," Annual *"Modern Bank Heists," VMware Carbon Black.*

[6] A. Khorram-Manesh et al., "Social and healthcare impacts of the Russian-led hybrid war in Ukraine–a con- flict with unique global consequences," *Disaster medicine and public health preparedness*, 17, pp. e432, 2023.

[7] A. Cherepanov and R. Lipovsky, "Blacken- ergy–what we really know about the notorious cyber attacks," *Virus Bulletin October,* 541, 2016.

[8] A. Unwala and S. Ghori, "Brandishing the cybered bear: Information war and the Russia-Ukraine conflict," *Military Cyber Affairs*, Vol.1, no.1, pp. 7, 2016.

[9] N. James, "How many cyber attacks per day:The latest stats and impacts in 2023," *Astra Security Blog*, Available athttps://www.getastra.com/blog/security-audit/how-many-cyber-attacks-per-day/ (Accessed: 01 October 2023).

[10] S. Mohurle and M. Patil, "A brief study of wannacry threat: Ransomware attack 2017," *International journal of advanced research in computer science*, Vol. 8, no. 5, pp. 1938-1940, 2017.

[11] S. Ali et al., "Comparative Eval- uation of AI-Based Techniques for Zero-Day Attacks Detection," *Electronics,* Vol. 11, no. 23, pp. 3934, 2022.

[12] Z. Iqbal and Z. Anwar, "SCERM—A novel framework for automated management of cyber threat response activities," *Future Generation Computer Systems,* 108, pp. 687-708, 2020.

[13] S. Kok et al., "Ransomware, threat and detection techniques: A review," *Int. J. Comput. Sci. Netw. Secur,* Vol.19, no. 2, pp. 136, 2019.

[14] M. Al Duhayyim et al., "Artificial Algae Optimization with Deep Belief Network Enabled Ransomware Detection in IoT Environment,"

Computer Systems Science Engineering, 46(2), 2023. in Cyber Security Domain," *Bitlis Eren U¨ niversitesi Fen Bilimleri Dergisi,* 11(2), pp. 508-518, 2022.

[16] S. Gulmez et al., "Analysis of the Dynamic Fea- tures on Ransomware Detection Using Deep Learning-based Methods," In *2023 11th Inter- national Symposium on Digital Forensics and Security (ISDFS),* (pp. 1-6), IEEE, (May, 2023).

[17] Z. Chen et al., "Machine learning based mobile malware detection using highly imbalanced network traffic," *Information Sciences*, 433, pp. 346-364, 2018.

[18] P. M. Anand et al., "A comprehensive API call analysis for detecting Windows-based ransomware," In *2022 IEEE International Conference on Cyber Security and Resilience (CSR),* (pp. 337-344), IEEE, (2022, July).

[19] I. Kara and M. Aydos, "The rise of ransomware: Forensic analysis for windows based ransomware attacks," *Expert Systems with Applications,* 190, pp. 116198, 2022.

[20] M. Y. Su et al., "Android Malware Detection Approaches in Combination with Static and Dynamic Fea- tures," *Int. J. Netw. Secur.,* 21(6), pp. 1031-1041, 2019.

[21] H. Faris et al., "Optimizing extreme learning machines using chains of salps for efficient An- droid ransomware detection," *Applied Sciences,* 10(11), pp. 3706, 2020.

[22] S. Homayoun et al., "A blockchain-based framework for detecting malicious mobile applications in app stores," In *2019 IEEE Canadian conference of electrical and computer engineering (CCECE),* (pp. 1-4), IEEE, (May, 2019).

[23] S. Hutchinson and U. Karabiyik, "Forensic analysis of spy applications in android devices," 2019.

[24] A. Naway and Y. Li, "A review on the useof deep learning in android malware detection," *arXiv preprint arXiv*:1812.10360, 2018.

[25] K. Xu et al., "Deeprefiner: Multi-layer android malware detection system applying deep neural networks," In *2018 IEEE European Symposium on Security and Privacy (EuroSP),* (pp. 473-487), IEEE, (April, 2018).

[26] M. Alazab, "Automated malware detection in mobile app stores based on robust feature generation," *Electronics,* 9(3), pp. 435, 2020.

[27] V. Rastogi et al., "Catch me if you can: Evaluating android anti-malware against transformation attacks," *IEEE Transac- tions on Information Forensics and Security,* 9(1), pp. 99-108, 2013.

[28] M. Alazab et al., "Intelligent mobile malware detection using permission re- quests and API calls," *Future Generation Com- puter Systems*, 107, pp. 509-521, 2020.

[29] H. Sun et al., "Android Malware Detection Based on Feature Selection and Weight Measurement," *Intelligent Automation Soft Computing,* 33(1), 2022.

[30] B. Vivekanandam, "Design an adaptive hybrid approach for genetic algorithm to detect effective malware detection in android division," *Journal of ubiquitous computing and communication technologies*, 3(2), pp. 135-149, 2021.

[31] R. Feng et al., "A performance-sensitive malware detection system using deep learning on mobile devices," I*EEE Transactions on Information Forensics and Security,* 16, pp.1563-1578, https://www.malwarebytes.com/blog/news/2014/05/five-pe-analysistools-worth-looking-at, [Accessed 12-08-2023], 2020.

[32] S. Jamalpur et al., "Dynamic mal- ware analysis using cuckoo sandbox," In *2018 Second international conference on inventive communication and computational technologies (ICICCT),* (pp. 1056-1060), IEEE, (April, 2018).

[33] N. James, "How many cyber attacks per day:The latest stats and impacts in 2023," Astra Security Blog.Available at: https://www.getastra.com/blog/security-audit/how-many-cyber-attacks-per-day/ (Accessed: 01 October 2023), 2023.

[34] C. Eagle, "The IDA pro book. no starch press," *Go (programming language),* Wikipedia. Available at: https://en.wikipedia.org/wiki/Go(programming language) (Accessed: 01 October 2023), 2011.

[35] W. J. Holstein and M. McLaughlin, "Battlefield Cyber: How China and Russia are Undermining Our Democracy and National Security," *Rowman Littlefield*, 2023.

[36] F. Sufi, "Social Media Analytics on Russia–Ukraine Cyber War with Natural Language Processing: Perspectives and Challenges," *Information,* 14(9), pp. 485, 2023.

[37] A. F. Brantly, "Ukraine War OSINT Analysis: A Collaborative Student Report," 2023.

[38] M. S. Dhelie et al., "Methods Used in Cyberattacks in the War Between Russia Ukraine," *(Bachelor's thesis, NTNU),* 2023.

[39] U. Urooj et al., "Ran- somware detection using the dynamic analysis and machine learning: A survey and research directions," *Applied Sciences,* 12(1), pp. 172, 2021.

[40] SentinelOne, Hermetic Wiper: Ukraine Under Attack. SentinelOneLabs. URL: https : //www.sentinelone.com/labs/hermetic-wiper-ukraine-under-attack/, 2022.

[41] Abuse.ch. (n.d.). Sample Hermetic Ransomware. MalwareBazaar. https: //bazaar.abuse.ch/sample/4dc13bb83a16d4ff9865a51b3e4d24112327c526c1392e14d56f20d6f4eaf382/