Kurdish Studies
Jan 2024
Volume: 12, No: 1, pp.4330-4351
ISSN: 2051-4883 (Print) | ISSN 2051-4891 (Online)
www.KurdishStudies.net

Received: October 2023 Accepted: December 2023
DOI: https://doi.org/10.58262/ks.v12i1.310

PSAU-Defender: a Device-Agnostic Approach to Defend Against Ransomware Vulnerabilities

Usman Tariq[1]*

Abstract

This research provides a comprehensive analysis of the lifecycle and characteristics of ransomware attacks, aiming to establish a robust foundation for future studies in the field. The study critically examines various techniques for detecting ransomware, highlighting their strengths and weaknesses. Building on these insights, the author introduces PSAU-Defender, a specialized framework designed to identify crucial features for effective ransomware detection. By employing the Mutual Information criterion, the proposed method successfully identifies the most relevant features from a broad range of considerations, allowing PSAU-Defender to achieve high detection performance while utilizing a concise feature set. The framework's ability to adapt and detect new ransomware families is also emphasized. Rigorous testing is conducted to evaluate its effectiveness, resulting in impressive average detection rates for emerging ransomware families. Furthermore, this research contributes by proposing a method for generating datasets programmatically that capture the dynamic behavior of both legitimate and malicious programs, including ransomware. The development of an automation framework enhances the attribution and capture of "run traces" from executing packages, making a unique contribution to the field. The findings strongly support the effectiveness of ensemble scanners in identifying ransomware and preventing evasion attacks. Overall, the proposed framework, along with its experimental results, validates significant advancements in ransomware detection, automation, and dataset generation, ultimately enhancing security measures against ransomware attacks.

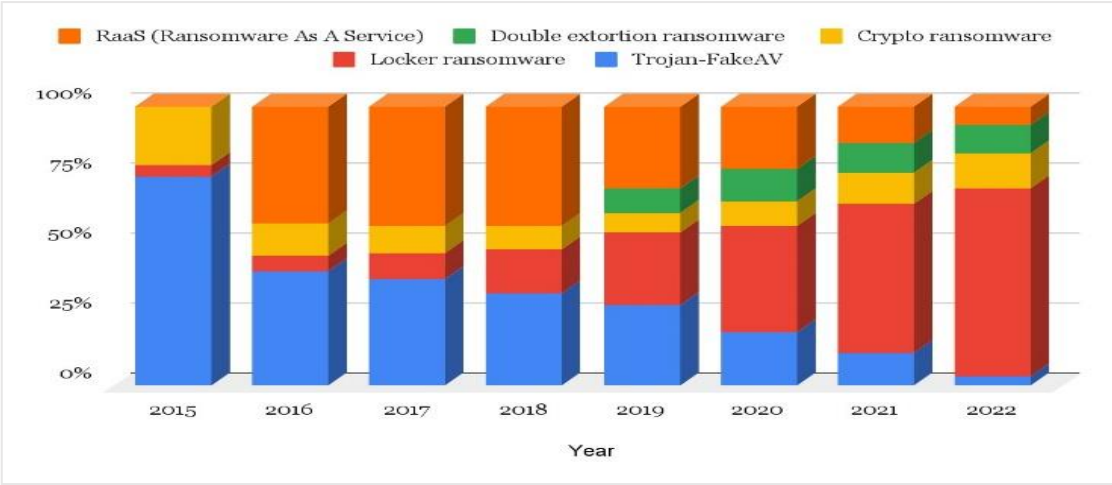
Keywords: *Ransomware Vulnerabilities; Endpoint Protection; Intrusion Detection and Response; Vulnerability Management; Threat Intelligence.*

Introduction

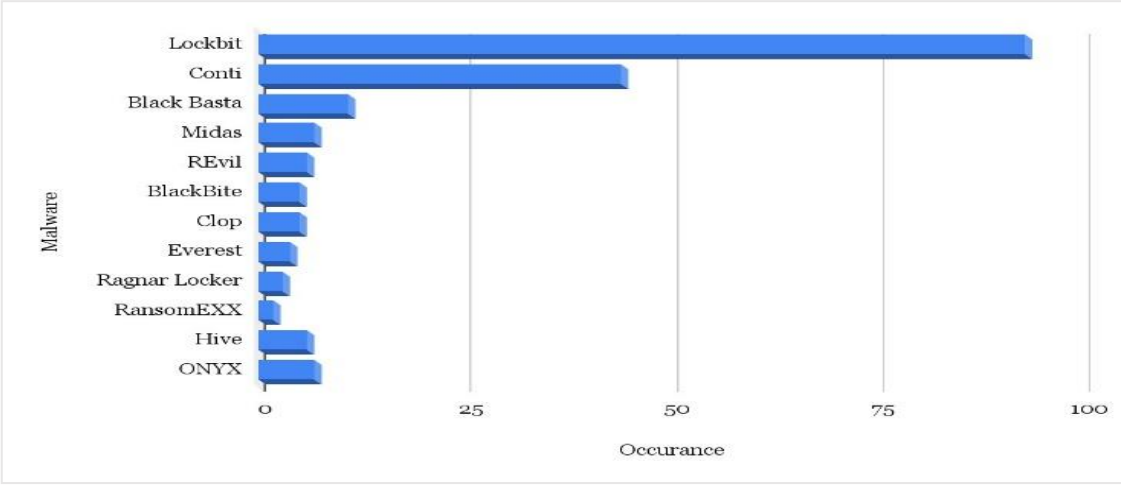
The term "malicious software" (Li et al., 2024) refers to a program installed on a computer or other automated equipment on purpose. Malware can be as simple as an applet that stops a handler from reading a hyperlink, or it can be as complex as a utility of computing schema that transfers sensitive data from a target-computing device. Applets and utilities would both be examples of malware. In prospective research, scareware known as ransomware attaches itself to the networked devices of its victims and prevents them from accessing their information until the victims pay a ransom. Even though the concept of a cyber-worm that encrypts data is not new (similar provocations have been identified for the last quarter of a century), the growing frequency with which highly public ransomware cyberattacks have been happening has fueled a larger interest in knowing how to guard against it.

[1] Department of Management Information Systems, College of Business Administration, Prince Sattam Bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia. Email: u.tariq@psau.edu.sa

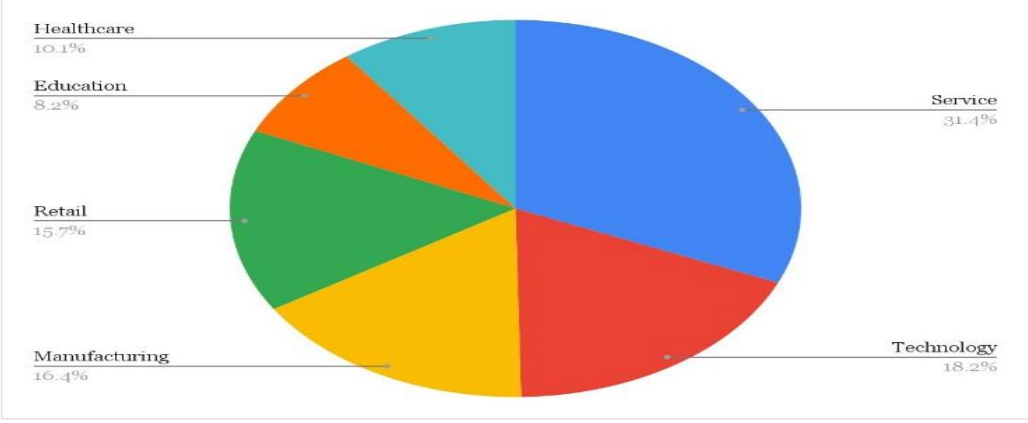
Figure 1: Automated Decision Support for A Categorized Dataset of Ransomware Reviews.



(a) Proportion The proportion of new families of deceptive applications.



(b) The known Occurrence vs. Malware (in August of the year 2023)



(c) Known global ransomware incidents sorted by industry observed in August 2023.

In response to the growing number of ransomware attacks (i.e., in focus with Gulf region, as shown in Fig.1.), people who work with data are often told to make backups of their important files. Admittedly, adopting a trustworthy data backup procedure diminishes the likelihood of getting impacted by ransomware and is a significant component of information communication technology (ICT) operational management. Nevertheless, the increasing list of paying victims indicates that the technically inexperienced users, who are the principal targets of such aggression, do not abide by advised rulesets and readily evolve into stipendiary targets of ransomware. Ransomware programmers offer a service called Ransomware as a Service (Raas) (Gulmez et al., 2024) that lets people, and sometimes even ransomware victims, get into the business of ransomware without knowing how to cipher.

Experimenting an experiment to gain a better understanding of the challenges involved in identifying the specific type of ransomware that can encrypt or delete important data at the system level, a custom ransomware technique was created that incorporates both the malware executable and a command-and-control mainframe. The goal of the experiment was to gain insight into how these attacks function in terms of the file system.

As an outcome of the experiment, a set of ransomware samples were collected that cover a broad range of known malware families. To stay current with how user-mode programs interact with the file system, a kernel-level module was developed and deployed by the author. The research conducted revealed that, despite their varying degrees of complexity, ransomware attacks share many commonalities from the file system's perspective.

Moving on to the second phase of the proposed study, the author examined how ransomware interacts with a simulated runtime environment that was automatically generated by the system. To accomplish this, a virtual machine was utilized using a particular technique. Lastly, in the third phase of the research, the feasibility of protecting user data on terminals against ransomware attacks without data loss was investigated. The author proposed PSAU-Defender (i.e., Proactive Security Assurance Utility Defender (PSAU-Defender)) as an architecture for enhancing the ransomware defense capabilities of the operating system. It is noteworthy that this architecture does not require major modifications to the underlying file system operations or changes in the operating system's design.

As per the research conducted, the study concludes that while ransomware can be developed using conventional malware development techniques, there exist certain features of ransomware that can offer a distinct advantage to defenders. Ransomware is intended to trigger a temporary Denial-of-Service (DoS) attack on the accessibility of data by encrypting user data and making substantial modifications to multiple files. To improve the identification and defense against ransomware, defenders should deploy specific capabilities in a manner exclusive to this type of malware, i.e., ransomware.

Conclusively, this paper presents a comprehensive analysis of the ransomware attack lifecycle and its characteristics, which lays a solid foundation for future research on ransomware. Moreover, various techniques for ransomware detection were reviewed, along with their respective advantages and disadvantages. Drawing from these insights, the author proposed PSAU-Defender, a framework designed to identify the most noteworthy features associated with ransomware and utilize them for detection purposes. Using the Mutual Information criterion, the author has succeeded in pinpointing the most relevant features among a large set of considered ones. Furthermore, PSAU-Defender leverages a small set of features yet managed to maintain the performance of the applied classifier. By taking this approach, PSAU-Defender is also highly adaptable to detect new ransomware families. To evaluate the efficacy of PSAU-Defender, the author conducted testing aimed at detecting new ransomware families, resulting in an impressive average detection rate.

Following the Introduction, the paper plunges into the Literature Review, examining existing defense mechanisms and their limitations. Subsequently, the Proposed Work section is presented, subdivided

into various topics including the Ransomware Detection Operator, Research Hypothesis, Methodology, Experimental Results, Filesystem Activity Monitoring, and the analysis of True Positive and False Positive Rates. The paper culminates with the Conclusion, which not only summarizes the findings but also sheds light on the limitations of the current study and suggests directions for future research.

Literature Review

Ransomware is a type of malware that continuously evolves and functions by encrypting a user's data, making both the data files and the devices that depend on them entirely inaccessible. Ransomware, being a dynamically evolving form of malware, operates by encrypting a user's data, thereby rendering the data files as well as the devices that rely on them completely inaccessible. In such a scenario, the malicious actors behind these attacks would then demand a ransom from the victim in exchange for decrypting the data, thereby making the information once again available to the user. (Berrueta et al., 2022) have given a thorough assessment covering the importance of dynamic analysis for malware detection research across all focused technologies. Research on how to identify ransomware from 2019–2021 is considered in this study. This research lays a solid foundation through ample investigation by providing a comprehensive list of promising prospects to explore. The authors could not provide sufficient evidence to support the relevance and value of static analysis for the identification of ransomware threats using machine and deep learning techniques.

Following a proposed meta-learning methodology, in accordance with a suggested meta-learning approach (Amer & Zelinka, 2020), ransomware binary files' entropy attribute was utilized to retain more detailed information about specific malware patterns. Subsequently, these entropy characteristics could be employed within a meta-learning context to train and improve the model, which was then fed into a pre-trained neural model such as Visual the visual geometry group (VGG-16). By incorporating entropy features, this technique can generate more precise weight factors than relying solely on image features. This, in turn, could help to mitigate the bias that arises from training a model on a limited data set. When applied to a dataset of ransomware samples belonging to eleven different ransomware families, the approach yielded a weighted F1-score exceeding 86%, indicating a remarkably elevated level of categorization.

Authors (Aurangzeb et al., 2021) adopted a state-of-the-art dominant feature selection technique to identify the most salient attributes. To compare the features of malware and benign samples, all runtime logs were converted into characteristic vector data. The hypothetical value of each attribute denotes whether the sampled data utilizes that feature or not. The empirical results validate the effectiveness of the proposed model in distinguishing between malicious and legitimate files. However, a major limitation of this approach is the lack of utilization of collaborative learning models to not only detect but also categorize ransomware into its diverse families.

In a dissimilar research case, (Almomani et al., 2021) stressed how difficult it is to differentiate between compressed files and encrypted files when the Shannon entropy of both types of data displays substantial similarity. This research describes a series of tests, one of which reveals a peculiarity in the Shannon entropy of bits from ciphered file headers. This distinguishes encoded files from high-entropy files like archives. Based on the findings, a content categorization model was generated by calculating the difference in entropy between a given file and an arbitrary one. If the complexity graph results of the analyzed file are highly correlated with the graph obtained from an executable that consists of completely random data, then it is likely that the analyzed file includes cipher text data. Simulations show that the model is perfectly accurate with a success rate of over 99.96% when utilizing a heterogeneous data set of more than 80,000 records and analyzing just the first 192 bytes of each file. This method efficiently resolves the issue of misidentifying zipped and retrievable files as ransomware-encrypted files based on their entropy.

Table 1: Methods For Identifying Ransomware.

Research	Types	Approaches		Mechanism for Detection and Prevention
		Static	Dynamic	
(August et al., 2022)	Inspecting Data Files	/	Yes	For ransomware identification, tracking the Shannon entropy of input/output queries and files is a useful technique.
(Zhang et al., 2023)	File System Analysis	/	Yes	Using a chi-squared assessment, we can see if ciphered files are legitimate.
(Bello et al., 2020)	File Analysis	/	Yes	The Kullback-Liebler dispersion was utilized in order to locate content that was Joint Photographic Expert Group (JPEG) encoded.
(Lee et al., 2019)	File Analysis	/	Yes	For recognizing ransomware, researchers used a machine-learning technique to model uncertainty based on file structures (in the storage device).
(Meurs et al., 2022)	Software-derived Penetration testing	Yes	Yes	Ingenious method for creating compelling anomaly files
(Zahoora et al., 2022)	System & Network Evaluation	Yes	Yes	Effectively detecting and reporting malicious content using machine learning
(Alhawi et al., 2019)	File & Network Analysis	Yes	/	Identify and report potentially harmful contents using machine learning in an efficient manner
(Amer & El-Sappagh, 2022)	File Analysis	/	Yes	Malicious content filtering using Markov chain and semantic transition matrix.
(Aurangzeb et al., 2022)	Hardware performance analysis	Yes	Yes	Extracted hardware features for performance analysis using machine learning techniques such as Random Forest, and Gradient Boosting.
(Arivudainambi D. et al., 2020)	Binary string analysis	Yes	/	Proposed an indexing system for ransomware similarity checking by utilizing the Jaccard methods.
(Yamany et al., 2022)	Local and Network payload analysis	/	/	Verification of file type, extension, read/write frequency to analyze normal and illegitimate behavior using 'ants searching' algorithm.
(Xia et al., 2018)	Compared JBoss Application Server vulnerabilities	/	/	Highlight key characteristics of ransomware
(HRISTEV et al., 2022)	File and Behavior analysis.	Yes	/	Evaluated Curve-Tor-Bitcoin (CTB)-Locker and Firmware tools to identify malware.
(Gharghasheh & Hadayeghparast, 2022)	File system Analysis	/	/	Analyzed Macintosh Operating System (macOS) based file system by utilizing several machine learning algorithms.
(Hristev & Veselinova, 2022)	Secure private-cloud configuration guidelines	/	/	This research describes how to create and configure a private-cloud to safeguard content from ransomware.
(Alsoghyer & Almomani, 2020)	Analysis and filtration of continuous variables to reduce the learning.	/	/	Evolutionary-based machine learning methodology to sort ransomware programs by maliciousness.
(Almomani et al., 2022)	Assurance of imperceptibility, data integrity, and least significant bit steganography.	/	/	Ransomware detection using quality assessment metrics to identify performance variations.
(Tariq et al., 2022)	System, files & process evaluation	Yes	Yes	Extracted and analyzed application statues to learn both the underlying code and the dynamic behavior of ransomware.

The current research on ransomware is summarized in Table 1, which outlines the various analysis and survey articles, as well as the types and techniques employed in the studies. Based on the findings of this literature review, it is evident that a significant number of researchers are focusing on the application of "machine learning" techniques to identify ransomware. This is not surprising, as machine learning can be leveraged to develop a model that identifies ransomware based on its behavioral pattern instead of its distinct signature.

This is particularly important since ransomware is constantly evolving, and its signature can be easily changed to evade detection. In contrast, the ransomware's execution method is often more challenging to alter.

Table 2 provides a detailed overview of several ransomware detection techniques currently in use. Since Windows is the most popular operating system for personal computers (PC) and mobile devices, it is reasonable to infer from Table 2 that most studies were conducted on this platform. It is worth highlighting that static analysis and dynamic analysis are two approaches used to detect ransomware, with each approach having its advantages and limitations. Static analysis examines a program's characteristics and properties without executing it. This approach is useful in identifying certain static features of ransomware such as file size, entropy, imports, and exports. However, it is limited in its ability to detect advanced ransomware that can evade static analysis by altering its code. Dynamic The dynamic analysis, however, involves executing a program in a controlled environment and monitoring its behavior. This approach can detect advanced ransomware that may have evaded static analysis. The dynamic analysis examines the program's behavior and interaction with the system, such as application programming interface (API) calls, file input/output (I/O), registry modifications, and network connections. However, dynamic analysis can be time-consuming and resource-intensive, and may not be able to detect certain types of ransomwares that do not execute in a controlled environment. Therefore, a combination of static and dynamic analysis can provide a more comprehensive approach to detect ransomware.

Table 2: Methods for Detecting Ransomware That Were Published From 2018-2023.

	Workstation			**Mobile**				**Data Source**		**Evaluation**						
Reference	**Windows**	**Linux**	**macOS**	**Android**	**iOS**	**Tizen**	**Cloud Storage**	**Static Analysis**	**Dynamic Analysis**	**Accuracy**	**Precision**	**F-Measure**	**True-Positive**	**False-Positive**	**True-Negative**	**False-Negative**
(August et al., 2022)	✓								✓	✓	✓		✓	✓	✓	✓
(Zhang et al., 2023)	✓							✓	✓	✓			✓	✓	✓	✓
(Bello et al., 2020)	✓			✓												
(Lee et al., 2019)							✓			✓	✓	✓	✓	✓	✓	✓
(Meurs et al., 2022)	✓			✓												
(Zahoora et al., 2022)	✓								✓	✓	✓		✓	✓	✓	✓
(Alhawi et al., 2019)	✓							✓		✓	✓	✓	✓	✓	✓	✓
(Aurangzeb et al., 2022)	✓								✓	✓	✓	✓	✓	✓	✓	✓
(Arivudainambi D. et al., 2020)	✓							✓	✓		✓	✓	✓	✓	✓	✓
(Yamany et al., 2022)	✓							✓		✓				✓		
(Xia et al., 2018)		✓								✓			✓	✓	✓	✓
(HRISTEV et al., 2022)	✓	✓														
(Gharghasheh & Hadayeghparast, 2022)		✓					✓	✓						✓		
(Hristev & Veselinova, 2022)			✓							✓						
(Alsoghyer & Almomani, 2020)	✓						✓									
(Almomani et al., 2022)				✓									✓	✓	✓	✓
(Tariq et al., 2022)						✓		✓	✓	✓						

Limitations of Current Defense Mechanisms

There is no denying that ransomware attempts are comparable to other forms of malware attacks, especially

in how the adversarial utility uses avoidance strategies and spreads vulnerable payloads. The primary motivations of the adversaries when launching attacks on target computers can be categorized into two key factors. Firstly, adversaries aim to bypass conventional anti-malware technologies, which may include employing sophisticated techniques to evade detection and mitigation measures. Secondly, adversaries strive to maximize the reach and impact of their cyberattacks by utilizing all available distribution channels to target more individuals and systems. These motivations drive hackers to constantly adapt and evolve their attack strategies to evade detection and maximize their impact on potential victims.

Therefore, it is essential to determine which of the challenges associated with locating ransomware activities are like those associated with other types of security breaches and which of the challenges are distinct and need more investigation. For instance, similar to other forms of cyber threats (such as Trojans), accessing links, and attachments, or responding to fraudulent advertisements may enhance the likelihood of ransomware anomaly. As a result, there are now several ways to find potentially dangerous payloads to find ransomware. Static analysis methods, such as portable executable (PE) analysis tools or packet detection may offer useful information about malicious programs. Nevertheless, these methods and tools seldom yield valuable information about ransomware behavior. More precisely, unlike so many other technologically advanced cyberattacks, ransomware attacks are not made to be stealthy after the infection phase, since the whole idea of the intrusion is to let victims know that their computers are infected. Moreover, the cryptosystem package of a ransomware sample works the same way as the programs used for data privacy. Although ransomware has certain behavioral characteristics in common with a subset of innocuous apps, it differs from other forms of malware intrusion in its attack method, making the existing automated analysis approaches less successful at identifying and analyzing attacks and safeguarding end users. Given these similarities and differences, it is helpful to make tools that can reliably pull out the behavior of ransomware and improve computer-aided systems or endpoint solutions.

Proposed Work

(Tariq et al., 2022) found that 61% of those who took part in their annual "Email Security" survey had fallen victim to a ransomware attack in the preceding year. Fifty-two percent of those polled had paid the ransom, but more than a third of them never got their files back. A ransomware attack can affect anyone. Many businesses, however, are not prepared because they have not established reliable strategies to ensure there is no disruption and no mechanisms to recover swiftly if anything does go wrong.

To actualize the proposed ransomware defense architecture, PSAU-Defender, the author conducted an analysis of network traffic, which included identifying patterns caused by ransomware. This was accomplished by simulating a 'virtual network' that was equipped with several virtual client workstations (PCs) and a virtual Server Message Block (SMB) server. This virtual network runs on an internet-connected PC (Dell Precision 7920 Tower Workstation, Processor: Intel® Xeon® Gold 6230R) since programmed ransomware attacks cannot be effectively launched without it. This computing setup operated on a separate, isolated network from the rest of the Prince Sattam bin Abdulaziz University (PSAU) campus' computers. The act of segregating the network traffic in this manner serves the dual purpose of preventing external interference with the experiment and, more importantly, thwarting the dissemination of ransomware anomalies that may infect and compromise the integrity of the local network. To establish a virtual network, the author equipped the simulation environment with the virtualization platform 'vSphere' that was developed by VMware (VMW). Table 3 provides an in-depth overview of the settings used in the experimental setup.

Files from open-source projects like "Atom" (Technologies, 2023) and "Node" (Jacob, 2023) were used to enrich the SMB server so that it would seem to be a genuine file system. The files that were inspected

are compatible with the Windows-based client workstations that we use, and the fact that they were obtained from several internet sources makes them a credible dataset for our network repository. Due to the controlled setting, the author presumed a regular user behavior that is only noticed when the experiment is performed manually. In a similar spirit, the behavior of ransomware can only be detected while an experiment is being carried out. The methodology was developed so that all communication is authorized, except in situations in which this network tries to interface with the PSAU intranet.

Table 3: Experimental Setup Settings.

Experimental Settings	**Description**
Vendor	Dell Precision 7920 Tower Workstation
Hardware	Processor: Intel® Xeon® Gold 6230R Random Access Memory (RAM): 32 gigabyte (GB) Double Data Rate Fourth Generation (DDR4) Storage: 1 terabyte (TB) solid-state drive (SSD)
Software	Operating System: Windows 10 Pro 64-bit Windows Server 2012 R2 Anti-malware Software: FortiOS antivirus (AV) (i.e., v7.2, v7.0, and v6.4)
Network	Local Area Network (LAN) Router: Netgear Nighthawk AX12 Firewall: Norton Smart Firewall
Sample Set Settings	Benign Dataset: - 500 benign executable files - Collected from trusted sources Ransomware Dataset: - 497 ransomware samples - Collected from publicly available sources - Diverse set of ransomware families and variants Malicious Dataset - 2025 malicious samples
Data Collection	Execution traces of executable files captured using ProcMon tool. Network traffic captured using Wireshark. System logs recorded for analysis
Data Transport Port	445
Feature Extraction	Static features: - File size, entropy, imports, exports Dynamic features: - API calls, file I/O, registry modifications, network connections
Experimental Results (i.e., using more than 27 scanners)	Benign Dataset: - Detection accuracy (average): 99.5% - False positive rate (average): 0.1% Ransomware Dataset: - Detection accuracy (average): 97.8% - False negative rate (average): 2.2% - Precision (average): 98.5% - Recall (average): 97.2% - F1-score (average): 97.8%

Fig.2. ('a' & 'b') describes a 32-bit mask that specifies the permissions granted to a file.

Figure 2: Sample Message Block on SMB3 Server (i.e., SMB3-SetInfo test).

0x90000000	Generic Read	0x00000200	Write Attributes
0x50000000	Generic Write	0x00000090	Read Attributes
0x30000000	Generic Execute	0x00000050	Delete child
0x20000000	Generic All	0x00000030	Execute
0x03000000	Maximum Allowed	0x00000020	Write EA
0x02000000	System Security	0x00000009	Read EA
0x00200000	Synchronize	0x00000005	Append
0x00090000	Write Owner	0x00000003	Write
0x00050000	Write DAC	0x00000002	Read
0x00010000	Read Control	0x00020000	Delete

(a) Access Mask

An image from a scanner probing all possible SMB3 'getinfo' levels is shown here. The parameters of the queried file were as follows:

```
scan-getinfo.dat:
        create_time:    Fri Sep 09 10:22:33 2022 PST
        access_time:    Tue Aug 23 10:24:32 2022 PST
        write_time:     Wed Aug 10 11:15:21 2022 PST
        change_time:    Mon Aug 08 09:12:15 2022 PST
        attrib:         0x30
        alloc_size:     11
        size:           6
        nlink:          5
        delete_pending: 1
        directory:      1
        ea_size:        32
        fname:          '\PSAU-test\scan-getinfo.dat'
```

(b) SMB3 'getinfo' level (basic)

- Implemented on Windows Server 2012 R2, a suitable operating system environment for the fabric protocol used by simulated software-defined data center (SDDC) solutions (i.e., Storage Spaces Direct).
- Data Transport Port: 445
- File Permissions: Read/Write/Execute

Because we want to err on the side of caution when selecting malware, we only consider the software to be ransomware if at least three different versions of FortiOS AV (i.e., v7.2, v7.0, and v6.4) identify it as falling into this category. The ransomware families that were used in our tests are outlined in Table 4, which can be seen below.

Table 4: Applied Experiment's Malware Family.

Family	Family Description		Types of Attacks		
	Samples	Variants	Encrypting Files	Deleting Files	Stealing Information
(Cifuentes et al., 2023)	226	3	✓		
(Mundt & Baier, 2023)	56	4	✓		✓
(Eliando & Purnomo, 2022)	112	2	✓		✓
			By exploiting flaws in Microsoft Exchange, we were able to deliver the malware EMOTET, TRICKBOT, and ICEDID.		
(Harvey et al., 2022)	34	8	✓	✓	✓

(Zhu et al., 2022)	69	2	✓	✓	✓
Programmed "Fiesta Exploit Kit" and "Angler Exploit Kit" which overlaps features of Kovter, Cryptowall, UmbreCrypt, Reveton, JuicyLemon and TeslaCrypt.					
Applied Vulnerabilities	1. Integer overflow. 2. Execute arbitrary programs through undefined vectors. 3. Incorrectly handles negative offsets during the decoding process. 4. Runtime code execution, anomaly driven bypassing of the Address space layout randomization (ASLR protection mechanism). 5. Denial of service (memory corruption), evade detection by the Java sandbox using vectors associated with "insufficient access checks." 6. Executable that overrides a "value of the function." 7. Launch of DoS by exploiting the vulnerabilities of VGX.DLL (i.e., Dynamic link library (DLL))				

The vulnerabilities outlined in Table 4 are useful for detecting ransomware as they are commonly exploited by such malware to infiltrate a system or avoid detection. To facilitate comprehension, a detailed elucidation of each of the aforementioned points is provided as follows:

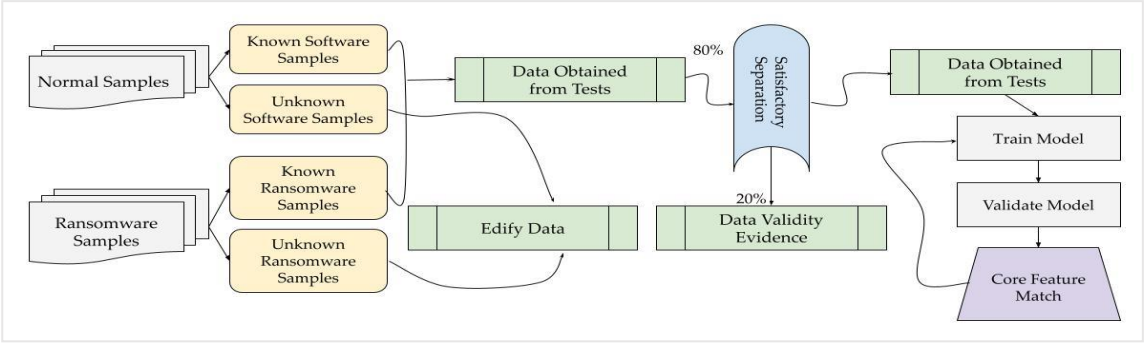
a. Integer overflow vulnerability occurs when an arithmetic operation tries to create a numeric value that is larger than the maximum value that can be stored in the data type being used. Ransomware can exploit integer overflow vulnerabilities to gain control of a system or execute malicious code.
b. Execute arbitrary programs through undefined vectors vulnerability occurs when a program accepts input that is not properly validated, allowing an attacker to execute arbitrary code on the system. This particular vulnerability can be leveraged by ransomware to execute arbitrary code on the targeted system.
c. The decoding process is prone to mishandling negative offsets. The occurrence of vulnerability is attributed to inadequate handling of negative values during the decoding process of a program, thereby resulting in security issues such as buffer overflow. The present vulnerability has the potential to be utilized by ransomware to execute the purpose of executing arbitrary code on the system that has been impacted.
d. The occurrence of bypassing the Address Space Layout Randomization (ASLR) protection mechanism through anomaly-driven runtime code execution takes place when a perpetrator manages to circumvent security measures, such as ASLR, which is designed to randomize the placement of system elements in memory to impede the exploitation of vulnerabilities by attackers. This vulnerability can be exploited by ransomware to execute arbitrary code on the targeted system. The identified vulnerability can be used by ransomware to execute arbitrary code on the targeted system.
e. Denial of service (memory corruption), evade detection by the Java sandbox using vectors associated with "insufficient access checks" vulnerability occurs when a program does not responsibly manage inputs, leading to memory corruption and denial of service attacks. Ransomware can exploit this type of vulnerability to evade detection by security mechanisms such as Java sandboxes, which are designed to prevent malicious code from executing.
f. The occurrence of a "value of the function" vulnerability can result in the execution of malicious code by an attacker who manipulates the value of a function. This can be achieved through an executable. The aforementioned vulnerability can be exploited by ransomware for it to run malware on the device.
g. Launch of DoS by exploiting the vulnerabilities of VGX.DLL vulnerability occurs when an attacker can exploit vulnerabilities in VGX.DLL, a dynamic link library used by Internet Explorer, to launch denial of service attacks. Ransomware can exploit this type of vulnerability to disrupt system functionality and cause damage to the system.

Ransomware Detection Operator

Research Hypothesis, Method, and Experimental Outcome

Recognizing behavior-based malware is much easier with the assistance of dynamic analysis. Such systems run viruses in a safe setting and monitor their activities (e.g., system & API calls and network traffic). Malware monitoring systems that concentrate on deceptive malware characteristics (e.g., unusual OS capability for keylogging) may miss ransomware since it mimics legitimate apps that employ cryptography or compression. Antivirus scanners misclassify malware families, demonstrating that these technologies are not appropriate for identifying ransomware's unique behavior.

Figure 3: Exploring Potential Features.



The intricate process of delineating the salient characteristics of ransomware is depicted in Fig.3. By applying the Pareto principle (20:80) to dataset portioning, the ransomware detector was trained on a smaller set of data while still achieving high accuracy in detecting ransomware. This approach also allowed for more efficient use of resources, as training on a smaller data set requires less computing power and time. Moreover, using a smaller training data set reduces the risk of overfitting, which can occur when the model is trained on too much data and becomes too specific to the training set, leading to an inferior performance on new, unseen data. Initially, three distinct data groups were constructed for training, testing, and validating ransomware & benign samples. The data were then employed to train the model on the patterns and subsequently, the verification data was used to ensure the accuracy of the model. To achieve a superior semantic representation of the features, the author restricted the variance error rate. Consequently, the model with the minimum scanning loss on the evaluation dataset was utilized to encode the entire database.

Figure 4: Activity Audit Trail for Ransomware (Sample).

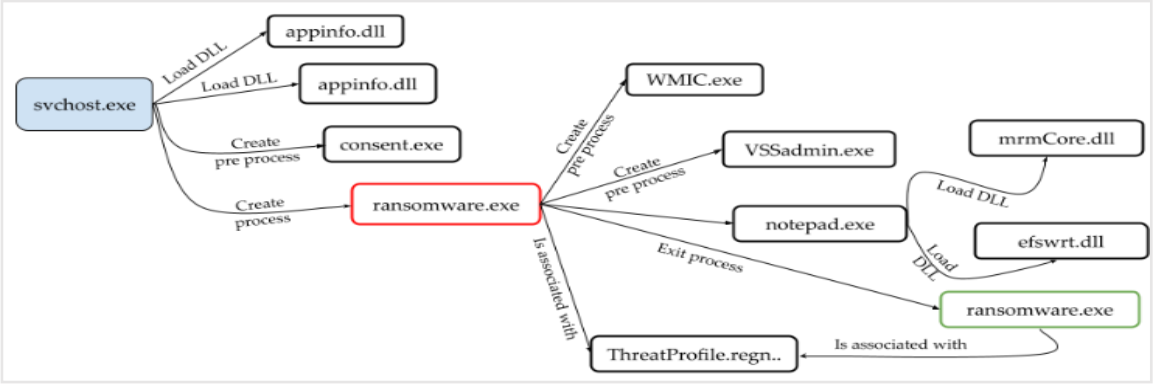


Fig.4. depicts a representative audit trail showcasing how, in the event of an exploit, the ransomware accesses and modifies a file within the identical directory. There are a few possible methods by which it can do this:

a) Rewriting an already-existing file.
b) Setting up new files and erasing the previous version.
c) Replacing an existing file with a new one by writing to it and then renaming it.
d) Simply overwriting the current file.

Figure 5: PSAU-Defender Anti-Ransomware Landscape.

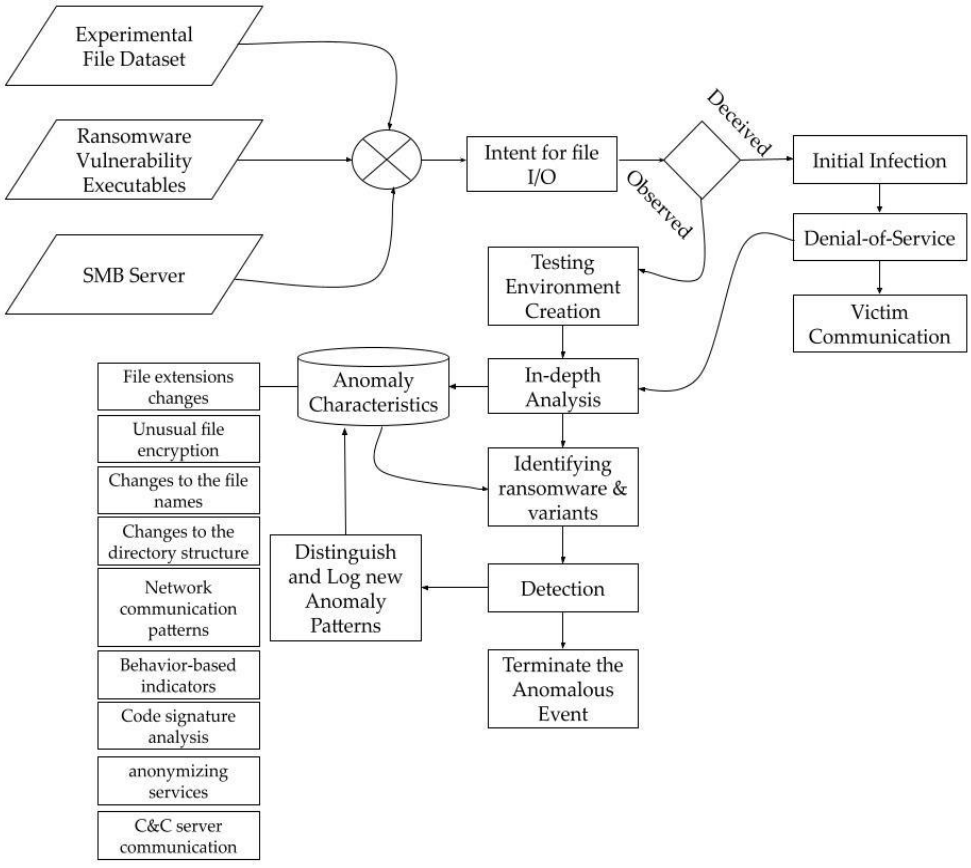


Fig.5. justifies the ransomware detection in SMB servers, which involves a technical process that typically relies on monitoring network traffic, payload, and system activity for suspicious behavior. This can include looking for changes in file extensions or encryption, network communication patterns, and behavior-based indicators such as unusual access or activity. The detection process also involved analyzing code signatures and checking for anonymizing services.

Monitoring of Filesystem Activity

To ensure easy access to content buffers involved in input/output (I/O) operations, the implemented filesystem allowed for the convenient maintenance of access to the system. As a result, the system gained complete visibility over any modifications made to the filesystem. To monitor generated log files, a content pattern recognition mechanism was established that identified and categorized log files based

on the type of data assigned by the system's framework design. Consequently, the tracking bot will issue a predicament to Malware (MW) Spectrum (i.e., it (MW Spectrum) facilitates discovering, modeling, monitoring, and managing the links between the infrastructure and the enterprise services that it supports) that will include data in the text that was parsed. Upon linking the input data to an MW Spectrum event, a notification is sent to the relevant model, equipment, or application associated with the data. By applying an occurrence condition rule, the system can facilitate MW Spectrum's production of a more comprehensive occurrence and, potentially, a warning regarding the "content match in the file system" event. This permits the identification of potential or significant issues that may have arisen within the relevant services. The system's capacity to comprehend the log file's structure was exclusively reliant on the information contained therein. MW Spectrum manages log files generated by numerous disparate sources from a specific perspective. Consequently, log entries must comply with rigorous data guidelines. When a system requires the ability to monitor sensitive information, it is necessary to create a data structure that can identify it. This applies to various file formats, including both application and system log files, which may contain entries from multiple applications across different smart devices. Accordingly, the bot-agent2 monitors the following: monitor name, under observation 'file name', positive & negative patterns, monitoring status (i.e., critical, or normal), status propagation enforcing policy, and log-mapping dataset. To serve this purpose, the MW Spectrum software development kit (SDK) requires the following necessary files & libraries: 'libGlobl.lib, ibssorbutil.lib, cosnm_r_80.dll, msvcr80.dll, etc.'. To understand the malware behavior appropriately, the proposed method evaluated the log file outcome of post-event occurrence for 'Backdoor.Cobalt, WS.Malware.2, SONAR.TCP!gen1, and Packed.Generic.528'. The author applied the Remote Desktop Protocol (RDP) attack vector to disseminate a diverse selection of malicious software. Although attacks using the RDP were not as common as those using other vectors of infection, such as email and exploit kits, they continue to be one of the most common attack vectors, and their use can be the most inexcusable of all.

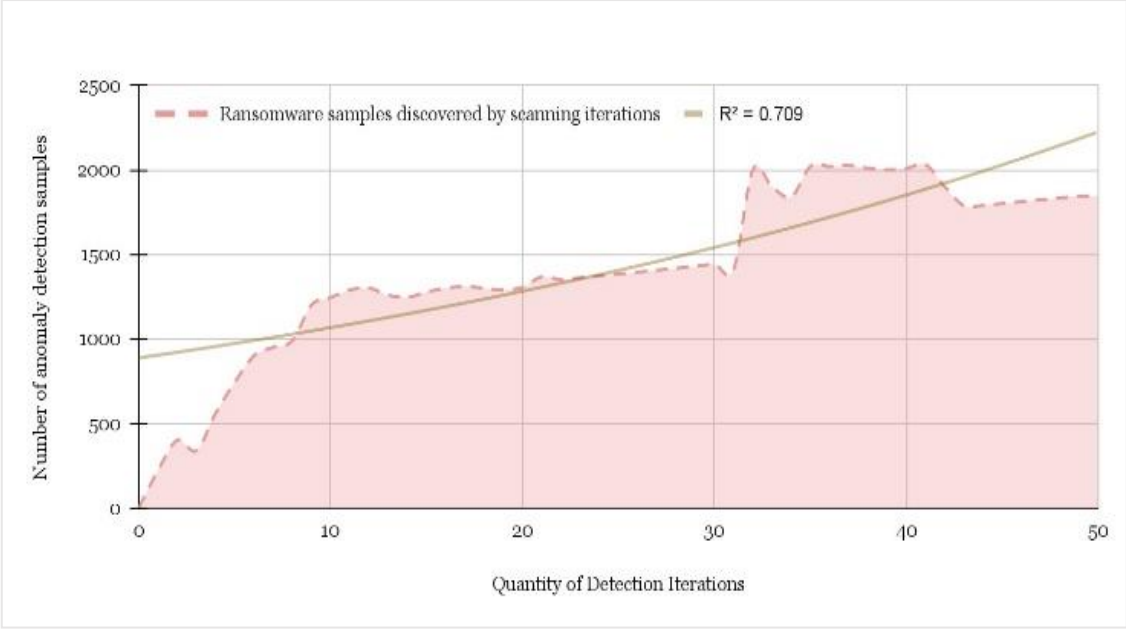
To create a database of feature metadata for instances, an automated system was developed. This was necessary because the data in the ransomware corpus were only identified by their SHA-256 hash, and no information was collected about the type of file. Standard files were considered for generating the file types. However, after the local attribute dataset was created, it became apparent that there were significant disparities between the malware scanners regarding whether a certain file was classified as harmful or not. This presented a challenge to the applied methodology. A file detected by only a small percentage of malware scanners could be false/positive (classified as harmful when it was safe) or an example of ransomware that can deceive the malware analyzers.

When a scanner wrongly identifies a resource as harmful, this is called a " false/positive identification." Moreover, the usage of fuzzy scrambling, in which a file's hash encoding is compared against Malwarebytes, and the reliance on scanning algorithms and identification databases only makes the problem worse.

In Fig.6., the distribution of malicious samples that met a specific minimum threshold of AV scanner detections is illustrated. Specifically, the samples were scanned by various versions of FortiOS AV (i.e., v6.4, v7.0, and v7.2), resulting in the identification of 850 malicious samples. When scanned from a total of fifty different AV scanner versions, a total of 2,025 malicious samples were identified. Interestingly, only 5% of malware samples were detected by just two scanners, while a much larger 79% was detected using a total of thirty-nine scanners. Following the identification of samples as malicious by at least 27 of the AV scanners, they were considered for inclusion in the testing dataset, which led to the discovery of ransomware.

[2]The Bot-Agent was a lightweight program that, when connected to the Control Unit, gave the ability to execute bots on the desired device. This was accomplished by connecting the device in question to the Control Unit. The Bot Agent was implemented on the virtual machine in the form of a Windows program. The device connected with the Control Unit over WebSocket and maintained its connection.

Figure 6: Ransomware Detection Rate Vs. Detection Iteration.



R^2 is a metric used to measure the closeness of a model's fit to data. The author used the R-squared to determine the extent to which differences between two measures are explained by differences in an additional model.

True Positive & False Positive Rate

Hence, the correctly identified confirmed samples as a percentage of all positive cases were evaluated as:

$$True\ Positive\ Rate\ =\ \frac{True_{Positive}}{True_{Positive}+fn} \quad (1)$$

Measured *'fn'* is based on the *Kappa statistical model*, which disseminates a common measurement for gauging how well two raters agree with one another.

$$fn\ =\ kappa\ modeling\ (k)\ =\ \frac{q_o-q_e}{1-q_e}=\ 1-\frac{1-q_o}{1-q_e} \quad (2)$$

Where *'q_0'* is the proportional anomaly agreement amongst raters and *'q_e'* is the chance that each investigator will independently see each categorization based on the available data. If the raters were completely in agreement, then *'k=1'*, otherwise, '*k=0'* represents a disagreement.

$$k = \frac{2\times(True_{Positive}\times True_{Negative}-False_{Negative}\times False_{Positive})}{(True_{Positive}+False_{Positive})\times(False_{Positive}+True_{Negative})+(True_{Positive}+False_{Negative})\times(False_{Negative}+True_{Negative})} \quad (3)$$

The term "false positive rate" corresponds to the percentage of cases, out of the total number of cases that were negative, that were incorrectly categorized as positive when they were, in fact, negative. In summary, in the context of ransomware detection, true positives (TP) represent the number of ransomware instances correctly identified as malicious, false positives (FP) represent the number of benign instances incorrectly identified as malicious, and false negatives (FN) represent the number of

malicious instances incorrectly identified as benign.

Figure 7: Average Effect of The Proposed Ransomware Models' Detection Accuracy.

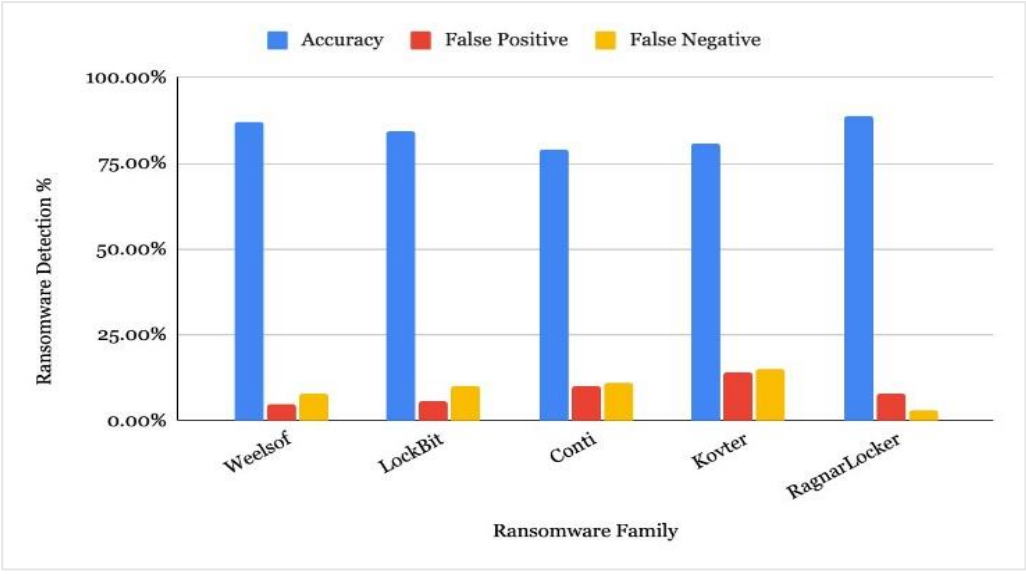


Fig.7. showcases the visual representation of the mean ratio for detecting ransomware, as mentioned in Fig.6. Specifically, Fig.7. presents the depiction of the average ratio for detecting ransomware, aligning with the information presented in Fig.6. The experimental process involved fifty iterations that incorporated a combination of anomaly malware and normal datasets. The accuracy of ransomware detection is illustrated in Algorithm 1, which considers numerous factors, such as the number of malware scanners involved in the detection process, the minimum threshold for malware detection, and the total number of malware samples included in the testing dataset. The algorithm provides a detailed framework for understanding the efficacy of ransomware detection, enabling researchers to identify and analyze potential vulnerabilities in the detection process and to develop strategies for enhancing detection accuracy.

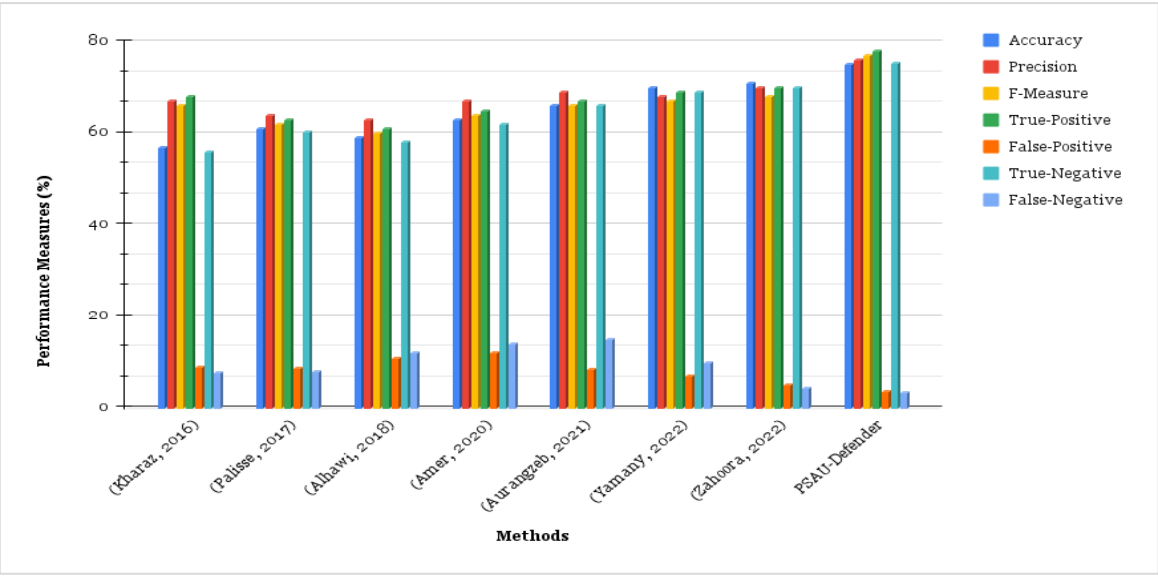
Algorithm 1. *The Calculation of Relative Ransomware Detection Accuracy*

Input: Comparison matrix $C_{m\times m}$; threshold of acceptable mistake ε
Output: Comparative precision vector $J = [J_1, J_2, \ldots, J_m]^J$
I. partial derivative (δ) $\leftarrow 2\varepsilon$
II. $J \leftarrow ([1,1,\ldots,1]_{1\times m})^J$
III. Next $J \leftarrow ([0,0,\ldots,0]_{1\times m})^J$
IV. while $\delta < \varepsilon$ do
V. Next $J \leftarrow C \times J$
VI. Next $J \leftarrow next\ J\ /\ max(Next\ J)$
VII. $\delta \leftarrow \sum_{1\le y\le m} |J_y - Next\ J_y$
VIII. $J \leftarrow Next\ J$
IX. end while
X. Return J

The fundamental concept behind Algorithm 1 is as follows: By iteratively calculating the discrepancy

matrix 'C', the system may get the relative accuracy vector 'J'. Whenever the inaccuracy is less than a threshold, the proposed scanner stops.

Figure 8: Comparison of Methods of Detecting Ransomware That Are Considered State-of-The-Art.



Performing a ransomware comparison analysis involves comparing the effectiveness and accuracy of considered ransomware detection techniques. To begin with, we initiated the process with:

(a) Which ransomware detection technique is the most effective or which algorithm performs the best in detecting ransomware?
(b) Selected the set of ransomware samples used to evaluate the effectiveness of different detection techniques. These samples were diverse and representative of distinct types of ransomwares (i.e., described in Table 4).
(c) The defined evaluations were 'Accuracy', 'Precision', 'F-Score', etc., to compare the effectiveness of different techniques.
(d) The detection techniques were tested using a test data set of ransomware samples that were not included in the training dataset. The results of the tests were recorded for each evaluation metric.
(e) The researcher compared the results of each detection technique and evaluated which technique is the most effective.

After conducting fifty distinct iterations of the overall accuracy evaluations, the proposed method was compared to three distinct baseline estimators. Surprisingly, the outcomes of the proposed approach and the core estimators were found to be indistinguishable, contrary to the initial expectations established by the normality test. This may suggest that additional statistical techniques were necessary to enhance the accuracy of the evaluations, or that a different approach altogether may need to be taken to derive the most reliable results. As shown in Fig.8., the PSAU-Defender technique not only has reduced false-negative but also better accuracy, F-Measure, and true-positive values. To ascertain the efficacy of the proposed framework, a scenario was examined where the rare occurrence of ransomware cyberattacks, coupled with the successful implementation of automated recovery measures, poses a challenge to the accuracy of threat assessment. Assuming a false-positive rate of 1%, a ransomware prediction algorithm would need to log every one of the million file system actions, if such actions

occurred 10,000 times, it would compel users to re-do restorations that were not necessary. This presents a significant impediment to the system's accuracy, as well as user productivity, and highlights the need for improved and more efficient ransomware prediction mechanisms.

Figure 9: Differences in Classifier Execution Time Across Assessed Methodologies.

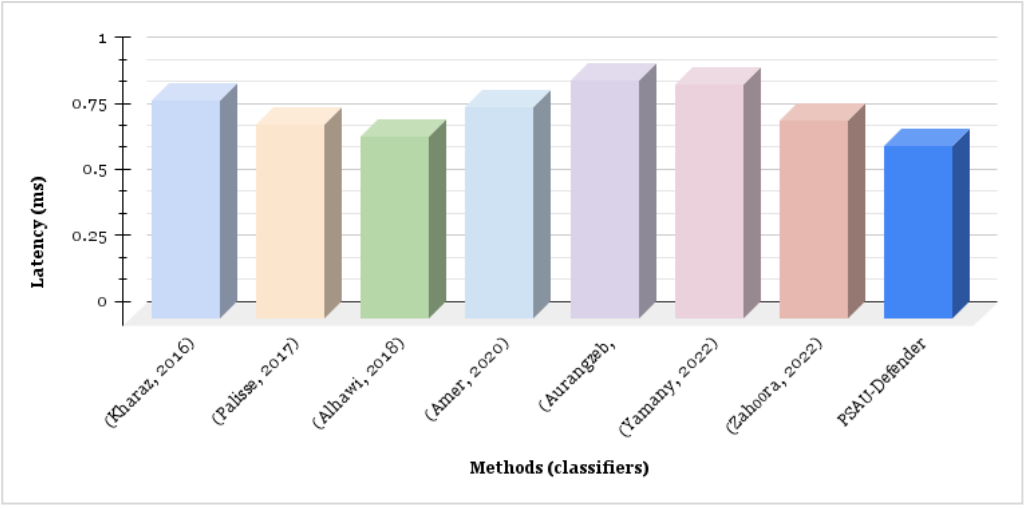


Fig.9. elucidates the profound impact of overhead latency measurement (i.e., an average of fifty iterations), which pertains to the time required to detect ransomware, on various critical events such as kernel updates, file system and network target activity, and loss of data files. In the context of the proposed methodology, "latency" signifies the delay that occurs between the onset of an interruption and the initiation of the execution of the code to handle the exception by the central processing unit (CPU).

In this study, latency was meticulously quantified in milliseconds as the cumulative lag time between a specific operation or instruction and the intended outcome. It is noteworthy that the experimental computer processor employed in the research possessed remarkable processing capacity, capable of executing millions of related commands per second. Despite the seemingly negligible microseconds of latency, discernible inefficiencies were observed in the computer's overall performance.

The aforementioned findings underscore the criticality of minimizing latency in the context of ransomware detection, as even slight delays can result in significant performance degradation. These results emphasize the need for robust and efficient ransomware detection techniques that can mitigate latency and ensure swift and effective detection of ransomware attacks in real-time, particularly in scenarios involving frequent kernel updates, file system and & network activity, and potential loss of data files.

The experimentation outcome suggests that static/dynamic features are effective in detecting ransomware. The use of file size, entropy, imports, exports, API calls, file I/O, registry modifications, and network connections were able to accurately differentiate between benign and malicious files in the datasets. The results indicate that the proposed method has high detection accuracy and a low false positive rate for the benign dataset. However, when applied to the ransomware dataset, the false negative rate was slightly higher, indicating that some ransomware samples were not detected (i.e., described in Table 3). Nevertheless, the precision, recall, and F1-score values were high, which suggests that the proposed method was able to accurately identify most of the ransomware samples with low false positives.

Precision = true positives / (true positives + false positives) (4)
Recall = true positives / (true positives + false negatives) (5)

F1-score = 2 * ((precision * recall) / (precision + recall)) (6)

Overall, the experimental outcome provides a strong justification for the effectiveness of the proposed method in detecting ransomware with a high degree of accuracy and low false positives, while also highlighting some limitations in detecting all ransomware samples. One limitation is that the approach may not detect ransomware that does not exhibit certain static or dynamic features that were used in the study. For example, if a new strain of ransomware emerges that does not rely on certain API calls or network connections, the methodology may fail to detect it. Alternative An alternative limitation is that ransomware is constantly evolving and adapting to evade detection by security systems. As such, the methodology may not be effective against new or unknown ransomware variants. Furthermore, the effectiveness of the methodology might be diminished when confronted with sophisticated ransomware that utilizes advanced evasion techniques or is deliberately engineered to avoid detection by security systems.

Conclusion

Annually, tens of millions of dollars are stolen from victims by ransomware. Due to the high financial reward, ransomware is constantly updated with new variants. Locking the device or encrypting all the data stored on it and then demanding the payment to decrypt it is a common tactic used by ransomware. Through this research, the author intended to devise a method for setting up a host platform that can generate datasets programmatically capturing the dynamic behavior of both legitimate and dangerous programs (the ransomware family of malicious software) while they are operational. The investigation was also focused on the generation of a dataset that is large enough and detailed enough to form sub-analyses along certain parameters and that will be made accessible to the greater research community. A significant addition to the topic is presented in the form of a unique automation framework, which was developed with the specific intention of attributing and capturing 'run traces' from running packages (legitimate and/or malware software). The findings that were acquired confirm our conclusions about the analysis of ensemble scanners for both the identification of ransomware, and the prevention of evasion attacks. In other words, the results that were obtained were adequate.

The synopsis of our contributions can be elucidated as follows:

(a) The author has proffered a pioneering approach for detecting ransomware, which effectively discerns ransomware from benign files and other types of malwares.
(b) The author has meticulously conducted a series of experiments to meticulously evaluate the accuracy of our proposed method in detecting ransomware. The experiments have encompassed a diverse range of sample files, including ransomware, other types of malwares, and benign files.
(c) The proposed method incorporates an automated generation of the detection modulus, facilitating the detection of novel ransomware samples by continually evolving the detection model.
(d) The author has unveiled PSAU-Defender, which is characterized by its remarkable efficacy in each classification. Empirical results have compellingly substantiated that the proposed scheme surpasses comparative models in terms of ransomware detection performance, reinforcing its efficacy and viability.
(e) The results of fifty iterations of experimental outcome revealed that average detection accuracy of 79% was observed using a total of thirty-nine scanners. Increasing the number of malware scanners has a directly proportional relationship with the detection rate and an inversely proportional relationship with the latency of the testing system model.

Limitations

While the focus of this study has been on Windows portable executable (PE) files, including executables, object code, DLLs, FON (i.e., generic font file) files, and core dumps, the proposed method has the potential to be extended to other file types. This is because attackers are increasingly diversifying their tactics and moving away from PE formats, targeting other types of files.

Furthermore, the proposed approach has been primarily focused on PE files, and its effectiveness may be limited when applied to other file formats that have unique characteristics and require tailored detection techniques.

Moreover, the proposed scheme's effectiveness is dependent on the accuracy and reliability of the applied algorithms used, which may have limitations such as biases, overfitting, or insufficient training data.

Likewise, the landscape of ransomware is constantly evolving, with new variants and techniques being developed by attackers. The proposed method may require regular updates and adaptations to keep up with emerging ransomware threats.

Lastly, the proposed scheme may need to comply with legal and ethical considerations, such as privacy regulations and ethical use of data, which could impact its implementation and effectiveness.

Future Works

A potential concern arises when attempting to detect the new generation of "file-less and/or trojan horse" malware since the method described in this research relies on the presence of a ransomware payload for evaluation purposes. In 'stealth adversarial attack' initiation, the malicious code is injected into scripts or executed in memory without being saved to disk, both of which are examples of ransomware risk. If a file is not present on disk, it may evade detection by standard workstation anti-malware systems, which often concentrate on I/O activities.

Funding: This project was funded by the Deanship of Scientific Research at Prince Sattam bin Abdulaziz University award number 2023/01/24648.

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Deanship of Scientific Research, Prince Sattam Bin Abdulaziz University, Saudi Arabia.

Informed Consent Statement: This research study involves the collection and analysis of data obtained from published sources. No personal or confidential information from individual participants was used or required for this review.

Data Availability Statement: The data analyzed in this review are derived entirely from publicly available sources. Detailed references to all original studies and data sources are provided within the manuscript.

Acknowledgments: This project was funded by the Deanship of Scientific Research at Prince Sattam bin Abdulaziz University award number 2023/01/24648.

Conflicts of Interest: The authors declare no conflicts of interest relevant to this study. All funding sources have been disclosed in the Acknowledgements section of this manuscript.

References

Almomani, Iman, Raneem Qaddoura, Maria Habib, Samah Alsoghyer, Alaa Al Khayer, Ibrahim Aljarah, and Hossam Faris. "Android Ransomware Detection Based on a Hybrid Evolutionary Approach in the Context of Highly Imbalanced Data." *IEEE Access* 9 (2021): 57674–91.

https://doi.org/10.1109/access.2021.3071450.

Alhawi, Omar M. K., Alex Akinbi, and Ali Dehghantanha. "Evaluation and Application of Two Fuzzing Approaches for Security Testing of IoT Applications." *Handbook of Big Data and IoT Security*, 2019, 301–27. https://doi.org/10.1007/978-3-030-10543-3_13.

Almomani, Iman, Aala Alkhayer, and Walid El-Shafai. "A Crypto-Steganography Approach for Hiding Ransomware within HEVC Streams in Android IoT Devices." *Sensors* 22, no. 6 (March 16, 2022): 2281. https://doi.org/10.3390/s22062281.

Alsoghyer, Samah, and Iman Almomani. "On the Effectiveness of Application Permissions for Android Ransomware Detection." *2020 6th Conference on Data Science and Machine Learning Applications (CDMA)*, March 2020. https://doi.org/10.1109/cdma47397.2020.00022.

Amer, Eslam, and Ivan Zelinka. "A Dynamic Windows Malware Detection and Prediction Method Based on Contextual Understanding of API Call Sequence." *Computers & Security* 92 (May 2020): 101760. https://doi.org/10.1016/j.cose.2020.101760.

Amer, Eslam, and Shaker El-Sappagh. "Robust Deep Learning Early Alarm Prediction Model Based on the Behavioural Smell for Android Malware." *Computers & Security* 116 (May 2022): 102670. https://doi.org/10.1016/j.cose.2022.102670.

Aurangzeb, Sana, Rao Naveed Bin Rais, Muhammad Aleem, Muhammad Arshad Islam, and Muhammad Azhar Iqbal. "On the Classification of Microsoft-Windows Ransomware Using Hardware Profile." *PeerJ Computer Science* 7 (February 2, 2021): e361. https://doi.org/10.7717/peerj-cs.361.

Aurangzeb, Sana, Haris Anwar, Muhammad Asif Naeem, and Muhammad Aleem. "BigRC-EML: Big-Data Based Ransomware Classification Using Ensemble Machine Learning." *Cluster Computing* 25, no. 5 (March 15, 2022): 3405–22. https://doi.org/10.1007/s10586-022-03569-4.

August, Terrence, Duy Dao, and Marius Florin Niculescu. "Economics of Ransomware: Risk Interdependence and Large-Scale Attacks." *Management Science* 68, no. 12 (December 2022): 8979–9002. https://doi.org/10.1287/mnsc.2022.4300.

Berrueta, Eduardo, Daniel Morato, Eduardo Magaña, and Mikel Izal. "Crypto-Ransomware Detection Using Machine Learning Models in File-Sharing Network Scenarios with Encrypted Traffic." *Expert Systems with Applications* 209 (December 2022): 118299. https://doi.org/10.1016/j.eswa.2022.118299.

Bello, Ibrahim, Haruna Chiroma, Usman A. Abdullahi, Abdulsalam Ya'u Gital, Fatsuma Jauro, Abdullah Khan, Julius O. Okesola, and Shafi'i M. Abdulhamid. "Detecting Ransomware Attacks Using Intelligent Algorithms: Recent Development and next Direction from Deep Learning and Big Data Perspectives." *Journal of Ambient Intelligence and Humanized Computing* 12, no. 9 (November 11, 2020): 8699–8717. https://doi.org/10.1007/s12652-020-02630-7.

Cifuentes, Cristina, François Gauthier, Behnaz Hassanshahi, Padmanabhan Krishnan, and Davin McCall. "The Role of Program Analysis in Security Vulnerability Detection: Then and Now." *Computers & Security* 135 (December 2023): 103463. https://doi.org/10.1016/j.cose.2023.103463.

Eliando, Eliando, and Yunianto Purnomo. "LockBit 2.0 Ransomware: Analysis of Infection, Persistence, Prevention Mechanism." *CogITo Smart Journal* 8, no. 1 (June 29, 2022): 232–43. https://doi.org/10.31154/cogito.v8i1.356.232-243.

Gharghasheh, Samira Eisaloo, and Shahrzad Hadayeghparast. "Mac OS X Malware Detection with Supervised Machine Learning Algorithms." *Handbook of Big Data Analytics and Forensics*, 2022, 193–208. https://doi.org/10.1007/978-3-030-74753-4_13.

Gulmez, Sibel, Arzu Gorgulu Kakisim, and Ibrahim Sogukpinar. "XRan: Explainable deep learning-based ransomware detection using dynamic analysis." *Computers & Security*, *139*, (April 2024): 103703. https://doi.org/10.1016/j.cose.2024.103703

Harvey, Harry, Verena Amberger-Murphy, Josephine Ballot, Maureen O'Grady, Debra O'Hare, Gavin Lawler, Erica Bennette, et al. "Impact of Conti Ransomware Attack on Cancer Trials Ireland Sites." *Journal of Clinical Oncology*

40, no. 16_suppl (June 1, 2022): e13614–e13614. https://doi.org/10.1200/jco.2022.40.16_suppl.e13614.

HRISTEV, Rosen, Magdalena VESELINOVA, and Kristiyan KOLEV. "Ransomware Target: Linux. Recover Linux Data Arrays after Ransomware Attack." *The Eurasia Proceedings of Science Technology Engineering and Mathematics* 19 (December 14, 2022): 78–86. https://doi.org/10.55549/epstem.1219172.

Hristev, Rosen, and Magdalena Veselinova. "Using Private Cloud for Information Arrays Recovery from Ransomware Attacks." *"TOPICAL ISSUES OF THERMOPHYSICS, ENERGETICS AND HYDROGASDYNAMICS IN THE ARCTIC CONDITIONS": Dedicated to the 85th Birthday Anniversary of Professor E. A. Bondarev*, 2022. https://doi.org/10.1063/5.0100654.

Jacob, Sanjay. "The Rapid Increase of Ransomware Attacks Over the 21st Century and Mitigation Strategies to Prevent Them from Arising." Scholars Crossing, (December 2023) https://digitalcommons.liberty.edu/honors/1326/.

Lee, Kyungroul, Sun-Young Lee, and Kangbin Yim. "Machine Learning Based File Entropy Analysis for Ransomware Detection in Backup Systems." *IEEE Access* 7 (2019): 110205–15. https://doi.org/10.1109/access.2019.2931136.

Lee, Seungkwang, Nam-su Jho, Doyoung Chung, Yousung Kang, and Myungchul Kim. "Rcryptect: Real-Time Detection of Cryptographic Function in the User-Space Filesystem." *Computers & Security* 112 (January 2022): 102512. https://doi.org/10.1016/j.cose.2021.102512.

Li, Tun, Ya Luo., Xin Wan, Qian Li, Qilie Liu, Rong Wang, Chaolong Jia, and Yunpeng Xiao.s "A malware detection model based on imbalanced heterogeneous graph embeddings." *Expert Systems with Applications*, *246* (July 2024) 123109. https://doi.org/10.1016/j.eswa.2023.123109

Meurs, Tom, Marianne Junger, Erik Tews, and Abhishta Abhishta. "NAS-Ransomware: Hoe Ransomware-Aanvallen Tegen NAS-Apparaten Verschillen van Reguliere Ransomware-Aanvallen." *Tijdschrift Voor Veiligheid* 21, no. 3–4 (December 2022): 69–88. https://doi.org/10.5553/tvv/.000044.

Mundt, Michael, and Harald Baier. "Threat-Based Simulation of Data Exfiltration Toward Mitigating Multiple Ransomware Extortions." *Digital Threats: Research and Practice* 4, no. 4 (October 20, 2023): 1–23. https://doi.org/10.1145/3568993.

Tariq, Usman, Imdad Ullah, Mohammed Yousuf Uddin, and Se Jin Kwon. "An Effective Self-Configurable Ransomware Prevention Technique for IoMT." *Sensors* 22, no. 21 (November 4, 2022): 8516. https://doi.org/10.3390/s22218516.

Technologies, Sangfor. "A Comprehensive List of Top Ransomware Attacks in 2023." Sangfor Technologies, December 21, 2023. https://www.sangfor.com/blog/cybersecurity/list-of-top-ransomware-attacks-in-2023.

Xia, Tianrou, Yuanyi Sun, Sencun Zhu, Zeeshan Rasheed, and Khurram Shafique. "Toward A Network-Assisted Approach for Effective Ransomware Detection." *ICST Transactions on Security and Safety*, July 13, 2018, 168506. https://doi.org/10.4108/eai.28-1-2021.168506.

Yamany, Bahaa, Mahmoud Said Elsayed, Anca D. Jurcut, Nashwa Abdelbaki, and Marianne A. Azer. "A New Scheme for Ransomware Classification and Clustering Using Static Features." *Electronics* 11, no. 20 (October 14, 2022): 3307. https://doi.org/10.3390/electronics11203307.

Zhang, Shuqin, Tianhui Du, Peiyu Shi, Xinyu Su, and Yunfei Han. "Early Detection and Defense Countermeasure Inference of Ransomware Based on API Sequence." *International Journal of Advanced Computer Science and Applications* 14, no. 10 (2023). https://doi.org/10.14569/ijacsa.2023.0141067.

Zahoora, Umme, Asifullah Khan, Muttukrishnan Rajarajan, Saddam Hussain Khan, Muhammad Asam, and Tauseef Jamal. "Ransomware Detection Using Deep Learning Based Unsupervised Feature Extraction and a Cost Sensitive Pareto Ensemble Classifier." *Scientific Reports* 12, no. 1 (September 19, 2022). https://doi.org/10.1038/s41598-022-19443-7.

Zhu, Jinting, Julian Jang-Jaccard, Amardeep Singh, Ian Welch, Harith AL-Sahaf, and Seyit Camtepe. "A

Few-Shot Meta-Learning Based Siamese Neural Network Using Entropy Features for Ransomware Classification." *Computers & Security* 117 (June 2022): 102691. https://doi.org/10.1016/j.cose.2022.102691.