Defenses Against Ransomware

CS658 Research Paper Proposal

Connor Raymond Stewart
David R. Cheriton School of
Computer Science
University of Waterloo
Waterloo, Ontario, Canada
crstewar@uwaterloo.ca

ABSTRACT

Ransomware attacks present new problems in computer security, such as preventing, recovering from, and detecting ransomware attacks. Ransomware is a relatively new and rapidly expanding phenomena in computer security, and designs for defences against ransomware are growing at a rapid pace. Ransomware is a new area of computer security design, and likewise, new forms of ransomware are being created constantly.

In the paper herein, defenses against ransomware attacks – and their implementations – will be discussed. A large range of approaches for ransomware protection are being brought into motion in todays world. As methods for ransomware attacks grow, an arms race between these different methods ensures. A multitude of different defences against ransomware attacks ranging from low level designs such as the hardware organization, up to systems and operating systems level designs, and including higher level designs such as cryptographic applications and web systems are discussed.

ACM Reference format:

Connor Raymond Stewart. 2021. Defenses Against Ransomware: CS658 Research Paper Proposal. *David R. Cheriton School of Computer Science. University of Waterloo, Waterloo, ON, Canada, 1 page.* https://doi.org/XXXXXXX.XXXXXXX

1. INTRODUCTION

Multiple topics related to modern defences against ransomware and approaches for defending against ransomware will be covered in the paper herein. Defences against ransomware include multi-layered defences which combine antimalware software, firewalls, DNS/Web filtering, email security, backups, and staff training [1]. Another defence includes Static and dynamic analysis tools, such as RansomWall, a layered defence system for ransomware attacks [10]. Security authentication approaches focusing on vulnerabilities rather than exploits also help with ransomware defence [4]. User-based training for avoiding ransomware attacks is also helpful [11]. Ransomware can be detected by monitoring API calls and file system activity and can be defended against by using decoy files [2]. Deep learning techniques to identify malicious behaviour accurately allow for ransomware detection [6]. Ransomware defence using stealthily spare space can be used as a ransomware recovery strategy, as hidden backups can recover lost data [5]. Firmware modifications like SSD-insider++ are proposed as defences and recovery mechanisms for ransomware attacks [9]. File monitoring for suspicious activity can be leveraged as a defence [7]. Renaming system tools that handle shadow copies can be used for ransomware recovery [8]. Finally, neural network models can monitor system resources to determine if a ransomware attack occurs [3]. Malware defences for embedded control devices are discussed, along with limitations and optimizations with control system protections [12]. Reinforcement learning methods for testing, debugging, and benchmarking anti-ransomware systems are discussed, along with methods to set up reinforcement learning model optimizations [13]. IoT defences against ransomware mainly focus on end-user defence strategies and behaviours as individual end users can significantly impact security by understanding how to avoid attacks [14].

In the following sections, these methods will be discussed in more detail. Information in the following sections is organized into four categories – high level, systems-level, low level, and workplace/non-technical – related to the type of security being provided.

2. HIGH LEVEL

High-level defences against ransomware attacks run on top of the operating system to monitor, stop, prevent, and mitigate ransomware attacks. Many high-level defences against ransomware attacks include neural network and deep learning methods, multilayered software defences, program analysis tools, and authentication system design.

2.1 NETWORK APPROCHES

Multilayered defences focus on developing a line of defences throughout a network to prevent a ransomware attack [1]. The first focuses are on securing endpoints from malware attacks by using behavioural monitoring and analysis [1]. A second layer uses firewall configurations to block bit-torrent ports, which are used as trapdoors for network access, geolocation blocks, and IP whitelists and blacklists [1]. The third layer of defence includes using a DNS/Web filter to check external DNS requests [1]. Fourthly, data backup and recovery should be implemented with onsite backups maintaining a one-way trust system to prevent corruption from a ransomware attack, and an offsite backup should be maintained as a last resort for recovery after a cyberattack [1]. As a fifth layer, email filters – like blocking specific file extensions or domain names - and security gateways can be implemented to prevent users within the system from being attacked [1]. A final layer of defence includes staff training to recognize problems like phishing attacks or suspicious activity [1]. Multilayered defences follow a typical path of network communication and filter network traffic at all steps of the network path [1]. Scenarios involving phishing email attacks were devised in the report and fleshed out into real-world scenarios [1]. Overall, defences throughout various points of the network stack are protected, and multiple layers of protection and backups are introduced with multilayered defences [1].

Malware defences using network security authentication provide the potential for new strategies when dealing with malware attacks. By focusing on vulnerabilities rather than exploits, the system proposed by Antonio and Fulp uses a remote security scanner to check for vulnerabilities and quarantines machines using logical network segmentation [4]. Although the paper does not specify ransomware in particular, its applications can be extended to ransomware since it focuses on all malware attacks. By preventing ransomware attacks from propagating through a mobile network, as is described in the paper, the scope and scale of a ransomware attack can be mitigated to a small grouping of devices on the mobile network rather than an entire network [4]. Mobile networks commonly fail with malware protection of particular since the typical inside-outside paradigm for threat categorization does not work [4]. The proposed system's strategy is to isolate system vulnerabilities before they become infected or attack others [4]. Three elements are used to construct the proposed architecture: a system to detect vulnerabilities, a system to enforce quarantines, and a system to integrate and manage the overall security policy [4]. A proof-of-concept system was developed to test the results with an experimental network setup [4]. The testing results show that the system provided malware defences and could manage, and quarantine machines based on vulnerabilities detected [4]. Future research could focus on better defining security policies employed by managers to confer better network protection [4]. The effects of implementing the design for ransomware protection are significant since it would minimize the number of computers affected by a ransomware attack during a network-based attack.

2.2 AI APPROCHES

RansomWall is a layered defence system for protection against cryptographic ransomware arracks [10]. RansomWall functions by following a hybrid approach combining static and dynamic analysis to generate a novel compact set of features that characterize ransomware behaviour [10]. Defences including a strong trap layer and machine learning-based zero-day intrusion detection systems are included in RansomWall [10]. Suspicious processes modify RansomWall backups files to prevent user data loss until the event is cleared as either actual ransomware or benign [10]. RansomWall showed substantial successes in the test cases – 574 cases specifically – it was used against; with a near-zero false-positive, it has a 98.25% detection rate when using a gradient tree boosting algorithm [10]. RansomWall detected thirty zero-day intrusion samples representing a less than ten percent detection rate [10]. The works developed by RansomWall include identifying a compact set of features that characterized the cryptographic ransomware behaviour, creating a trap layer for early detection, using machine learning for unearthing zero-day intrusions, and developing a backup mechanism for preserving user data [10]. RansomWall was evaluated against 574 samples from 12 cryptographic ransomware families and 442 samples of benign software in real-world test environments [10]. The analysis of cryptographic ransomware attack stages (propagation, infection, communication with command-and-control server, user data encryption, and extortion of end-users) followed by the implementation of a layered defence architecture (static and dynamic analysis for monitoring behaviour before and after execution along with honey files and trap layers to bait ransomware attacks) were used with RansomWall [10]. The machine learning model uses sequential supervised learning with a moving average sliding window [10]. RansomWall will be tested for large-scale systems in the future [10].

Works on deep learning and neural network models for ransomware detection are not limited in scope to the works mentioned in the above paragraph. A neural network model proposed by Ketzaki *et al.* shows an effective use of a neural network detection model – the detection procedure is based only on metrics related to system performance, not source code analysis [3]. Two primary focuses for ransomware include static analysis – analyzing the ransomware without the code running – and dynamic analysis – analyzing ransomware with its code running [3]. It is assumed that infections during the malware distribution phase increase the CPU and memory features in the computer and that the file searching phase increases the features related to the number of files and the rate of reading and writing [3]. Implementations occur in three main parts: the first part being the development and the installation of the system monitor service, the second being the feature extraction related to the development of log files generated from the system monitor, and the third part is the development of the neural network model [3]. The neural network model was tested using ten-fold cross-validation and was shown to have an accuracy of 99.98% and a precision of 99.8%, showing the potential neural network model in the testing environment [3]. Neural network models can be expanded to account for deep learning models, which represent different types of ransomware detection platforms. In a model proposed by Al-Hawawreh and Sitnikova, classical auto-encoders and variational auto-encoders are used to perform feature engineering processes [6]. They work in unison to collect data to reduce data dimensions and extract efficient data representation [6]. The encoders generate a new feature vector and are trained to train and test the neural network with a batch normalization classifier to account for the dynamic behaviour associated with ransomware attacks [6]. The deep learning model helps improve the effectiveness of ransomware detection and enhances the generalization capabilities of the decision engine [6]. By reducing data dimension, limitations from some previously discussed approaches, including false positives, high computational complexity, and low generalizability to new ransomware platforms, can be addressed [6]. The model uses a training phase, followed by a hybrid feature engineering phase (for the encoders), finalized with a deep neural network batch normalization phase [6]. Testing phases interplay throughout the hybrid feature and batch normalization phases [6]. The model built in the paper shows promising results with a 95.38% accuracy, a 96.99% precision, a 92.53% detection rate, and a 7.47% false-negative rate [6]. The rates in the deep learning model by Al-Hawawreh and Sitnikova show a more significant margin of error than the previously mentioned AI-based ransomware protection approaches, but the given approach allows for highly generalizable ransomware detection for ransomware within select family groupings.

Reinforcement learning methods for anti-ransomware bug testing have been proposed by Adamov and Carlsson [13]. Ransomware simulations consisting of arbitrary combinations of known tactics and techniques were used to bypass an antimalware defence system to find flaws in its design [13]. An agent was trained with the help of reinforcement learning programs to run the ransomware simulator in a way that can bypass anti-ransomware security software [13]. Ransomware simulators require a set of parameters – which can be optimized using reinforcement learning to operate themselves correctly and bypass antimalware security systems [13]. Q-Learning is used to generate a reinforcement learning model which can maximize a reward function that represents the maximal number of file encryption with a minimal number of steps [13]. The learning model functions by starting with a random exploration policy and then slowly changing the probability from purely random to purely deterministic as the model determines the best possible set of actions to bypass the security system [13]. Analysis shows that the number of successful ransomware attacks increases exponentially with the number of games played by the reinforcement learning agent [13]. Unfortunately, the reinforcement learning agent was only tested on a limited number of ransomware models, but it can easily be extended to other models in the future for a more in-depth look at anti-ransomware vulnerabilities [13]. The reinforcement learning model-based approach could be extended to network penetration optimization in future works made by the authors [13]. Having a software quality testing system of this variety is very important for creating robust anti-ransomware detection systems, and it is vital in the defence against ransomware attacks.

3. SYSTEMS LEVEL

Systems-level protection confers protection against ransomware attacks at the operating system level. Operating systems can be made vulnerable to ransomware attacks to bypass high-level software protection. Various systems-level approaches ranging from backups to renaming of systems tools and dependencies to file monitoring are discussed.

3.1 RECOVERY SYSTEMS

Recovery from ransomware attacks provides a safety net if a ransomware attack bypasses other forms of defence within a system. Paying ransom money after a ransomware attack may not provide a victim with a genuine recovery key to decrypt their data, and it has ethical implications since it provides money to criminals and incentivizes future ransomware attacks. Recovery components involve creating backup systems which cannot be targeted by ransomware, like cloud services or external device drives [5]. The RDS3 system consists of two main components, a ransomware detector component and a ransomware recovery component [5]. The detection component monitors the computing system for a ransomware attack, attempts to block the ransomware attack, and sends a notification to the recovery system [5]. Preventing the ransomware from having access to the data despite having access to system privileges can be addressed by using a separate high-security lightweight OS on a virtual machine monitor, which manages the backup volumes in the system drives [5]. Since a system cannot duplicate its memory entirely, delta encoding can be used to incrementally backup user data based on a secure list of essential files to be backed up [5]. The recovery system consists of a backup system that periodically communicates with the user operating system to extract data backed up and a recovery system that helps the user operating system read data from the recovery blocks to restore data after the ransomware attack [5]. Algorithms for marking data for backup and backing up the data securely are produced such that ransomware cannot determine the identity of essential data or corrupt the backup datasets when the implemented program uses those algorithms [5]. A limitation of the approach is that it cannot determine when a ransomware attack will occur, meaning data that has not had the chance to get backed up will be lost if a ransomware attack encrypts new data [5].

3.2 OPERATING SYSTEM

Operating system and file system techniques can defend against ransomware attacks. A tactic for ransomware detection includes file monitoring for suspicious modifications and recovery from backups [7]. May and Laron propose two techniques to improve protection: consideration of the file lifecycle and content analysis [7]. File lifecycle using complex events allows for reconstructing the user's mental model, and content analysis using Apache Tika detects attacks by monitoring suspicious content changes [7]. Determining what characterizes ransomware from regular system activity, the files that need to be backed up, and whether a file modification represents malicious ransomware activity becomes a primary focus [7]. Behavioural monitoring of the file system and operating system can determine what characterizes ransomware since many ransomware tools encrypt a multitude of files simultaneously [7]. A surge of file read/write operations occurs during a ransomware attack and can be used to identify ransomware activity [7]. Keeping shadow backups or backing up all files about to be written can be used to backup systems for a ransomware attack [7]. Lastly, monitoring the file entropy of files in the filesystem or file extension changes can be used to determine if file modifications are suspicious [7]. File lifecycle analysis involves studying how users use files in a system and how files are created, modified, then deleted [7]. By focusing on significant file versions and ignoring irrelevant files, file lifecycle analysis can help ransomware detection [7]. Content analysis tools filter emails and files to determine a file type based on a set of predefined rules and can be used to compare a file's content type before and after modification [7]. Suspicious data structure changes indicate a file could have been modified maliciously rather than accessed commonly by a user; they can be used to detect malicious changes to high entropy binary file formats that undergo complex modification steps [7]. Event modelling and handling of API modules can be used to determine filesystem events occurring on the system, as was implemented using a pure Java software utility by May and Laron [7]. The software utility successfully detected all encryption attacks by a popular ransomware software known as $ucyLocker and protected the filesystem without allowing any false positives [7].

Another system protection can be conferred by renaming system tools that handle shadow copies [8]. Analyzing the four most common crypto ransomware reveals that all infections rely on tools available on the target system to be able to prevent a simple recovery after an attack is detected [8]. The proposed system allows for the recovery of files after a ransomware attack by using the window's native function, making shadow copies, and modifying it to act as a backup system for ransomware attacks [8]. Existing literature does not mention internal backup systems of this type, and most authors suggest that if a system is infected, an external backup is the only feasible means of recovery [8]. The recreation of files from the internal recovery system is elaborated on within the paper [8]. Testing reveals that if the script from the paper is executed before an attack occurs, there is a significant probability that the files will be restored [8].

Monitoring the MFT filesystem table can allow a system to detect deletion, creation, and encryption of files [2]. The use of decoy files in the filesystem can be used as a buffer to slow the progress of a ransomware attack until a detection is made [2].

LOW LEVEL

Low-level defences against ransomware attacks prevent pre-operating system-level program code hijacking during potential ransomware attacks. Bypassing operating system protection allows a hacker to bypass systems-level protections and make ransomware applications hidden from any high-level protections. Methods such as the addition of firmware protections within SSD drives or the protection of API calls can be implemented to prevent possible attacks at this level.

Many ransomware variants use Windows API to lock a victim's desktop [2]. Two popular ransomware libraries – CryptoLocker and CryptoWall – rely on Microsoft CryptoAPI, and intercepting session keys related to the API made detection possible [2]. Deleting the non-encrypted files after a ransomware attack makes encrypted copies of files on the system can be performed using Windows API function calls [2]. More interceptions of the Windows API function calls can prevent the deletion of files after crypto ransomware makes encrypted copies of everything on the computer. Intercepting the calls makes detection of the ransomware and recovery after an attack. As described earlier in the report, dynamic analysis techniques can be used to check for ransomware activity during code execution [3]. Running captured ransomware in a controlled environment and recording system calls, API calls, and network traffic calls can allow for the creation of anti-malware software to detect ransomware by training those signatures during the runtime of a computer [2]. Problematically, capturing low-level signatures cannot always help with ransomware detection since a hacker can program the ransomware system call signals to resemble regular benign software like standard compression applications [2].

Another ransomware prevention proposal is SSD-Insider++, which is embedded into an SSD controller as a form of firmware [9]. A primary benefit of a firmware approach is that the anti-malware software is not vulnerable to evasion techniques since it is separate from the host machine [9]. Two features exist in the SSD-Insider++ system, allowing ransomware detection and data recovery. Ransomware detection observes input and output patterns in a host system to determine if a host system is being attacked by ransomware [9]. After an encryption attack is detected, a recovery algorithm is activated to restore the original files by using a delayed deletion feature of an SSD [9]. Overhead for the system is measured to be negligible in the report, and high accuracy at detecting ransomware attacks and no false positives and no false negatives are recorded [9]. Update after reading and trim after read activity are used to detect ransomware attacks by the firmware system, as they are used to erase old data after modification [9]. The SSD-Insider++ system can roll back SSD changes after an attack is detected by setting a rollback time, at which point the mapping table of SSD changes reverses the queue of instructions that the user operating system initially has set [9]. If the primary detection system fails, then a backup detection system in the firmware works by checking if new files have high entropy values [9]. Since entropy values are high in encrypted files, entropy checkers can be used to determine if a ransomware attack is occurring [9]. The firmware on the SSD drive can roll back the changes after a ransomware attack is detected by reloading the modified files with their corresponding original data [9]. An in-memory data structure-based counting table is made to detect ransomware attacks by comparing entries corresponding to SSD memory reads and writes and is compared to the output of a ransomware detection algorithm to determine if an attack is occurring [9]. Invariant features of ransomware like input/output footprints and out-of-place updates associated with ransomware are used as markers for detection [9]. The firmware makes some assumptions that may not hold universally true for all ransomware platforms, like the assumption that fsync is immediately invoked to trim commands to deleted files, which would bypass the primary protection systems used by SSD-Insider++ [9].

Traditional antimalware systems rely on a blacklist-based approach where software known to be malicious is blocked by the system [12]. Modern malware is growing at a pace much faster than it originally grew, and a blacklist approach is showing limitations in a ransomware market where new approaches are determined constantly [12]. Modern whitelist approaches - which check if code is allowed to execute if verified as safe - lower system overhead and are much more consistent in long-term security settings [12]. Control systems for embedded devices present a scenario where a whitelist approach that blocks non-trusted code performs better than traditional antimalware approaches [12]. Architecture-based approaches like mandatory access control can be introduced for embedded systems since it allows segregating application into separate execution domains with specific permission granted to those domains [12]. Rootkit prevention is a new and largely unexplored area of malware protection, and it works by ensuring drivers and kernel modules come from trusted sources [12]. Rootkit prevention systems can also verify that system calls were not modified or interfered with, as is the case with hook attacks [12]. Whitelist-based antimalware systems have a significant drawback in that they have a high administrative overhead if new pieces of code need to be added to the whitelist regularly [12]. A whitelist-based system cannot accept new code without the code being approved and whitelisted, limiting the system's usability to only trusted codesets [12]. A MAC system helps by constraining the reach of a piece of software running in the computer system to contain a ransomware attack [12]. Unfortunately, MAC systems are challenging to set up and test; and cannot detect malware that modifies the integrity of the process being placed into a domain (i.e. modifying the filesystem) [12]. Finally, rootkit prevention systems have the benefit that they can help monitor the system for attacks that would otherwise bypass all other security layers by hiding themselves from the kernel, but they also have the drawback that it is difficult since it requires the kernel to monitor itself from any possible attack [12]. Embedded systems by design require a different set of ransomware protections from regular systems, which, as was exemplified above, can be beneficial since custom and niche antimalware systems can be designed around the operational constraints of the embedded system. Not all antimalware systems require the system to be useable in that a typical PC is useable.

WORKPLACE AND NON-TECHNICAL

Non-technical defences against ransomware attacks are as important as technical protections. In many cases, users can act as backdoors that an attacker can leverage to gain access to a system. Due to the fact that an attacker can use a human to gain back door entry into a computer system, a computer system cannot be made stronger than the humans that use the system. Therefore, user training becomes very relevant for protection against ransomware attacks, especially in the case of multi-user systems with non-technical and non-security personal.

Information on strategies for ransomware removal and prevention can be found in a paper by Saxena and Soni. Ensuring user workstations have operating systems that are up to date and ensuring antiviruses are up to date can help prevent ransomware attacks [11]. Employees should be instructed to keep software up to date on the schedule to prevent ransomware attacks from propagating through a network. Email gateways should have trusted emails signed appropriately, and all employees should recognize trusted emails [11]. Employees should be educated to survey email sources to prevent a successful ransomware trojan-based attack from infiltrating the entire network [11]. Intermediaries like proxies, external databases, and activity logs can be maintained and used by users with proper security authorization and training [11]. Active monitoring of system security by users can shield the network from an attack since users can detect and stop the unusual activity. Users of intrusion prevention systems should be constantly refreshed on any relevant workplace devices [11]. Applying updates to browsers, programming language compilers, and runtime environments should be maintained by all users in the network [11]. Computer applications should be kept up to date to prevent security vulnerabilities related to their software [11]. Users should be trained on ransomware threats and given examples of previous ransomware attacks – like GoldenEye or WannaCry – for context [11]. Users should be informed not to pay ransomware attackers in the event of a network attack and should also be instructed on network administrators to contact to deal with the situation [11]. Early mitigation protocols like system shutdowns could be added to a list of options for users to engage in since the system can be used to prevent the ransomware from propagating throughout the network [11].

Ransomware awareness in the IoT industry and information on how to defeat ransomware attacks are important training aspects for workers' knowledge [14]. Mobile users can instruct not to install apps outside of official application stores like the Apple App Store or the Google Play Store, never grant system permissions to untrustworthy sources, use passwords, avoid untrustworthy internet websites, and avoid untrustworthy internet websites to keep device IDs secure, etcetera [14]. Knowledge of risks like fake apps that are used as backdoors for ransomware attacks and the various forms of ransomware attacks like *Locker Pin* (resets the pin for the phone) and *Lucky Ransomware* (which encrypts files through email scripts) so that users are prepared for the more esoteric attack angles [14]. Other methods to protect mobile devices include containers, encryption, and device compliance checks [14].

METHODS

Many different methods and runtime environments for program code can be used as a vector for a malware attack. Considering that malware and ransomware attackers can circumnavigate a single layer of defence by attacking different program stacks, developing a full-stack ransomware protection system is imperative. Hackers will attempt to find the vulnerable point in a system's defences, and if a single program stack is locked down completely, attackers will find ways to slip around it. For example, network protections and firewalls are not reasonable if an attacker can install a rootkit on a computer since the computer is now vulnerable to attack regardless of how well designed the firewall is. Furthermore, an attacker can bypass antimalware software if they bypass operating system protections since traditional antimalware software runs on top of the operating system stack. Ultimately, a program security designer would need to look at how ransomware can slip into a system, be it a network, operating system, or otherwise, and devise strategies to protect against the attack.

Another point of interest is that there are many different ways to defend against ransomware. Since ransomware attackers work by attempting to bypass system securities, having multiple layers of security with different implementations and designs is a good strategy for ransomware defence. A ransomware attacker may be able to bypass a single security approach. However, combining many ransomware detection utilities, including multiple neural networks with differing models, blacklist detection systems, system call retrievals, API monitoring, network monitoring, and more, can be used to provide layers of redundancy in case a single method of antimalware protection fails.

Testing anti-ransomware systems using debugging platforms or ransomware attacks is one method to learn how to defend against ransomware attacks. Getting many different ransomware programs and using reinforcement learning to optimize their runtime parameters is a method to learn what does and does not work with anti-ransomware software. Understanding how to test if anti-ransomware software is effective, robust, and non-bypass-able is as essential as understanding ransomware defence approaches. We can remark that it would be challenging to understand what approach to use or why it works if one does not understand what good anti-ransomware software looks like during a live attack.

The report is based on the understanding that combining, synthesizing, and unifying these approaches is instrumental in designing antimalware software capable of preventing a ransomware attack. Using multiple layers of defence increases the chances of detecting ransomware attacks; since ransomware requires a period to pass undetected to encrypt vital data, detection becomes the primary step for its prevention.

DISCUSSION

Many different approaches for protection against ransomware are discussed, ranging from low-level hardware design to high-level network and neural network models. Future research into ways to combine the various antimalware program stacks to prevent ransomware from slipping into cracks between the varying programs can be done. Educating the general population on these defence methods and ways to install these antimalware programs would be a good step to prevent ransomware attacks. The creation of commercial software combining these approaches for protection would be a significant step in the security industry.

In general, many of the methods described have a drawback: if the attacker understands the security method, they can easily find a way to bypass it. Many security systems revolve around the fact that the attacker can always break into a system, but the security system makes it, so it is not worth it for an attacker to do so. Combining many different methods would make a system that can still be bypassed like the others mentioned above, but a ransomware attacker would not want to spend their time developing such a system. If software combining multiple approaches were to become too popular, it would be likely that a ransomware developer would develop malware capable of bypassing all the approaches described above. Successive additions of new paradigms to the antimalware software would need to be made in order to keep it operational in the long term.

Many different testing platforms for anti-ransomware systems can be introduced in the future to create standardized testing platforms for anti-ransomware systems. A standard battery of tests that measure anti-ransomware systems' general rigour and robustness has not been invented. It would be worthwhile for future researchers to create a generalized examination of anti-ransomware defence systems to measure their performance for comparison. Security errors and optimizations in anti-ransomware designs could be inspected and fixed, and future directions and dead ends in developing anti-ransomware detection systems could be avoided. Reinforcement learning testing agents, as was mentioned by Adamov and Carlsson, seem like strong candidates for the future of ransomware testing since AI development is making significant progress [13].

CONCLUSION

The article provides an overlay of multiple modern anti-ransomware systems for review and discusses their drawbacks and benefits. The article discusses the useability, cost, difficulty of implementation, and overhead for the aforementioned antimalware systems. Possible future directions for software implementation in ransomware prevention are discussed in detail along with the antimalware methods and are summarized at the end of the article in the discussion section. The methodology behind the paper's organization and the relevance of viewing malware protection under this lens is described in detail in the methods section. Testing techniques for anti-ransomware detection programs like reinforcement learning optimization for ransomware programs are also covered. A concise summarization of modern antimalware approaches is outlined in the report herein for review and research.

ACKNOWLEDGMENTS

Written for CS 658 – Computer Security and Privacy as it was offered in the Winter term of 2022 and presented by instructors Meng Xu and Hans Liljestrand at the University of Waterloo. The paper presented is written to fulfill the requirements of a course project for the writing of a research paper on topics related to computer security and privacy.

REFERENCES

[1] Alexander Pagan and Khaled Elleithy. 2021. A multi-layered defense approach to safeguard against ransomware. *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*: 942–947. http://doi.org/10.1109/ccwc51732.2021.9375988

[2] Daniel Gonzalez and Thaier Hayajneh. 2017. Detection and prevention of crypto-ransomware. *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*: 472– 478. http://doi.org/10.1109/uemcon.2017.8249052

[3] Eleni Ketzaki, Petros Toupas, Konstantinos M. Giannoutakis, Anastasios Drosou, and Dimitrios Tzovaras. 2020. A behaviour based ransomware detection using neural network models. *2020 10th International Conference on Advanced Computer Information Technologies (ACIT)*: 747–750. http://doi.org/10.1109/acit49673.2020.9208974

[4] J.V. Antrosio and E.W. Fulp. 2005. Malware defense using network security authentication. *Third IEEE International Workshop on Information Assurance (IWIA'05)*: 43–54. http://doi.org/10.1109/iwia.2005.11

[5] Kul Prasad Subedi, Daya Ram Budhathoki, Bo Chen, and Dipankar Dasgupta. 2017. RDS3: Ransomware defense strategy by using Stealthily spare space. *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*: 1–8. http://doi.org/10.1109/ssci.2017.8280842

[6] Muna Al-Hawawreh and Elena Sitnikova. 2019. Leveraging deep learning models for ransomware detection in the industrial internet of things environment. *2019 Military Communications and Information Systems Conference (MilCIS)*: 1–6. http://doi.org/10.1109/milcis.2019.8930732

[7] Michael J. May and Etamar Laron. 2019. Combating ransomware using content analysis and complex file events. *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*: 1–5. http://doi.org/10.1109/ntms.2019.8763851

[8] Mattias Wecksten, Jan Frick, Andreas Sjostrom, and Eric Jarpe. 2016. A novel method for recovery from crypto ransomware infections. *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*: 1354–1358. http://doi.org/10.1109/compcomm.2016.7924925

[9] Sungha Baek, Youngdon Jung, Aziz Mohaisen, Sungjin Lee, and Daehun Nyang. 2021. SSD-assisted ransomware detection and Data Recovery Techniques. *IEEE Transactions on Computers* 70, 10: 1762–1776. http://doi.org/10.1109/tc.2020.3011214

[10] Saiyed Kashif Shaukat and Vinay J. Ribeiro. 2018. Ransomwall: A layered defense system against cryptographic ransomware attacks using machine learning. *2018 10th International Conference on Communication Systems &amp; Networks (COMSNETS)*: 356–363. http://doi.org/10.1109/comsnets.2018.8328219

[11] Smruti Saxena and Hemant Kumar Soni. 2018. Strategies for ransomware removal and prevention. *2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*: 1–4. http://doi.org/10.1109/aeeicb.2018.8480941

[12] Josh Powers, Rhett Smith, Zafer Korkmaz, and Husam Ahmed. 2015. Whitelist malware defense for embedded control system devices. *2015 Saudi Arabia Smart Grid* (SASG): 1–6. http://doi.org/10.1109/sasg.2015.7449271

[13] Alexander Adamov and Anders Carlsson. 2020. Reinforcement learning for anti-ransomware testing. *2020 IEEE East-West Design & Test Symposium (EWDTS)*: 1–5. http://doi.org/10.1109/ewdts50664.2020.9225141

[14] Soobia Saeed, N.Z. Jhanjhi, Mehmood Naqvi, Mamoona Humayun, and Shakeel Ahmed. 2020. Ransomware: A framework for security challenges in internet of things. *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*: 1–6. http://doi.org/10.1109/iccis49240.2020.9257660