CSCE 5550 Spring 2024

Introduction to Computer Security

Project Ransomware Group 2 Final Report

Varun Mandadapu 11725699

Ponika Mannem 11704153  
 Shreshta Pasham 11649309  
 Ragasree Thatipamula 11724308

Amruthacharya Kothakota 11703396

1. ***ABSTRACT***

This project examines the construction and countermeasures of a simulated ransomware attack within a controlled virtual machine environment. Initially focusing on the development of a ransomware tool, the project progresses through the stages of deployment, encryption activity, and subsequent detection. The simulation aims to mimic real-world ransomware behaviour by encrypting a specific directory structure within the guest operating system, thus offering a practical perspective on the dynamics of ransomware operations and their impact on digital systems.[7]

In response to the simulated threat, the project emphasizes the development and integration of sophisticated monitoring and detection systems that utilize behavioural analysis and heuristic evaluations to identify malicious activities. The subsequent mitigation phase involves implementing automatic responses, including process termination and data backup procedures, designed to minimize damage and facilitate rapid recovery. This thorough approach offers important insights into strengthening cybersecurity defenses against ransomware threats and demonstrates the efficacy of multi-layered security strategies in a practical setting.[3]

Through this effort, a thorough understanding of the workings of ransomware, the possible harm it might cause, and proactive defense strategies against such threats were investigated. The ability of the specially created ransomware tool to successfully encrypt files and directories was thoroughly examined and confirmed. Conversely, the monitoring, detection, and mitigation measures implemented proved capable of identifying the ransomware activities and allowing for safe file recovery. All things considered; this was a useful experience for learning about ransomware.

1. ***INTRODUCTION***

With more and more examples of targeted and widespread assaults that disrupt enterprises by encrypting vital data and demanding a fee to unlock it, ransomware continues to pose a serious threat to global cybersecurity. These hostile assaults have the potential to cause severe harm, including interrupted operations, monetary losses, and tainted data. The need to create efficient countermeasures to guard against these threats is growing as ransomware techniques continue to change and grow more complex.

This project simulates the entire lifetime of a ransomware attack inside a safe, isolated virtual machine environment, directly addressing this urgent cybersecurity issue. This project offers a thorough and useful understanding of ransomware dynamics without the dangers involved with actual attacks by simulating real-world ransomware behavior and researching its deployment, infection techniques, encryption activities, and subsequent impact.[11]

The study takes a multipronged strategy, starting with a thorough investigation of current ransomware tactics, such as encryption schemes, infection vectors, and dissemination tactics. Based on this foundation, a unique ransomware tool was created that could replicate the file encryption method used by real ransomware versions by recursively encrypting a certain directory structure.

A phishing email with a malicious executable was created to authentically mimic an attack. This email was used as the first infection vector to spread the ransomware inside the virtual environment. The ransomware's potential to interfere with and compromise system data was demonstrated when it encrypted the targeted directory after it was triggered.[4]

In response to this simulated threat, the project focused on developing robust monitoring, detection, and mitigation strategies. To detect and log all file system activity, a comprehensive monitoring system was put in place. This made it possible to identify suspicious patterns that are typical of ransomware behavior. Subsequently, a blend of heuristic rules and signature-based methodologies were employed to precisely identify the existence of ransomware in this data.

Following a successful identification, an automated mitigation phase was initiated, which involved stopping malicious activities right away, protecting data that was not impacted, and starting recovery operations. To effectively tackle ransomware threats, this multi-layered approach emphasizes the significance of a comprehensive cybersecurity plan that includes prevention, detection, and quick reaction techniques. The study illustrated the potential devastation caused by ransomware attacks and the effectiveness of proactive security measures in mitigating such threats through an immersive simulation. The endeavor will yield essential insights and practical experiences that will enhance cybersecurity preparedness and resilience against the constantly changing threat landscape of ransomware and other harmful software.

* + 1. ***RELATED WORKS***

1. " The Evolution of Ransomware" by Kharraz et al.'s 2015 study offered an in-depth look at how ransomware has evolved over time. By analysing over 15 different ransomware families, they traced the transition from basic screen lockers that disabled system access, to more advanced crypto-ransomware capable of encrypting files and demanding cryptocurrency payments. Their findings highlighted growing trends like targeted enterprise attacks and tactics to bypass detection like stalling encryption.[9]
2. "RansomWall: A Ransomware Mitigation Framework" by Acar et al. (2018) developed Ransom Wall, a novel framework leveraging data mining on system logs to identify ransomware behaviour patterns indicative of malicious file encryption activities. When detected, Ransom Wall could automatically kill the ransomware processes and revert encrypted files using secured backup data. Evaluations showed their approach achieved over 96% detection accuracy with minimal performance overhead.[14]
3. " Encoded Ransomware: How Ransomware Persists with Alternate Data Streams" by Sipe et al. in 2019, employed a unique persistence mechanism called Alternate Data Streams to conceal its payload. By hiding within these streams, the ransomware could covertly encrypt files over time while evading most antivirus detection. The researchers demonstrated how this technique circumvented common defences and proposed kernel-level monitoring to mitigate such threats.
4. " Ransomware Detection Using Machine Learning on Execution Patterns" by Almashhadani et al. (2021) explored using machine learning for ransomware detection based on execution patterns observed during dynamic malware analysis. Their approach extracted distinct system behaviour features which were used to train detection models like Random Forests, XGBoost, and deep neural networks. Evaluations on a new ransomware corpus found XGBoost achieved 99.5% accuracy in flagging both known and unknown ransomware variants.[2]
5. “ Ransom Stopper: A Comprehensive Windows Service Against Ransomware" by Alajlouni et al. (2022) designed Ransom Stopper, an anti-ransomware solution for Windows that monitors process activities and file system changes. It combines signature-based rules with behavioural heuristics to identify ransomware. Upon detection, Ransom Stopper can terminate the malicious process, recover affected files from backups, and restore deleted shadow copies to enable full recovery. Extensive real-world testing validated its effectiveness against modern strains.[1]
6. ***APPROACH***
7. ***Research on ransomware techniques***

In order to comprehend the many processes and evolution of ransomware attacks, a thorough literature review and research were conducted during the project's first phase. Understanding the complexity of ransomware, from its encryption techniques to the different ways it infiltrates systems, required a thorough understanding of this core study. Through a review of both historical and contemporary ransomware cases—which are recorded in academic papers, security reports, and real-world event analyses—the team was able to pinpoint common characteristics and creative strategies used by attackers. This includes researching the various stages of the ransomware lifecycle, including first access, execution, encryption, and ransom communication, in order to create a thorough foundation for creating our simulation. Understanding the encryption methods, including RSA and AES, that are frequently utilized by ransomware and how they are applied to lock victim's data effectively.[6]Top of Form

Building on this understanding, the group investigated the efficacy of several current defence strategies in thwarting ransomware attacks. Modern detection methods, like behaviour-based, signature-based, and anomaly detection techniques, were reviewed as part of this investigation since they may be able to spot ransomware activity before serious harm is done. We also looked at methods for removing ransomware without caving in to requests for payment, such as using decryption software created by cybersecurity experts and community-based projects. The knowledge gathered from this research phase helped us choose the technologies and techniques to create our own ransomware tool and the ensuing detection and mitigation plans. This preparation was critical for setting a realistic simulation environment that mirrors the tactics and techniques of modern ransomware attackers, ensuring that the subsequent phases of action, infection, monitoring, detection, and mitigation were grounded in practical and current cybersecurity knowledge.[8]

1. ***Action***

The action phase of the project focused on the practical implementation of the ransomware tool based on insights gathered from our initial research. Utilizing Python and the cryptographic library PyCrypto, we developed a ransomware script capable of recursively encrypting files within a designated directory structure labelled as "critical". This directory, along with its subdirectories "lab1", "lab2", and "lab3", were populated with test files to simulate a realistic environment for the ransomware to target.

The encryption mechanism chosen was AES (Advanced Encryption Standard), due to its balance of speed and security, making it a common choice in real-world ransomware attacks. The script was designed to be triggered via a simulated phishing email, reflecting a common vector for ransomware distribution. In parallel, we crafted a decryption component, which could theoretically be used to restore the encrypted files once a ransom was paid, mimicking the ransomware's decryption functionality. This part of the project was critical in demonstrating the complete lifecycle of a ransomware attack from encryption to potential recovery, preparing us for the subsequent phases of detection and mitigation.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedTop of Form

A screenshot of a computer

Description automatically generated

1. ***Infection***

In the "Infection" phase of the project, we implemented a realistic simulation of a ransomware infection using a phishing email tactic. This phase involved the creation of a Python script that simulates sending a phishing email containing a malicious executable disguised as a legitimate attachment. Upon execution of this attachment i.e., free\_money.exe by the unsuspecting recipient, the ransomware is activated, initiating the encryption process across the designated 'critical' directory structure. The infection method was meticulously tested in a controlled environment to ensure it mirrored actual attack vectors used by cybercriminals.

A screenshot of a computer screen

Description automatically generated

This testing demonstrated the effectiveness of the infection mechanism and provided crucial insights into how such threats propagate, setting the stage for subsequent monitoring and detection efforts

A screenshot of a computer

Description automatically generated Top of Form

A screenshot of a computer screen

Description automatically generated

After the encryption is completed, it will send the key.bin file i.e., the key that is used to encrypt the entire directory also the key which the victim needs to decrypt his directories will be sent to our mail so that the hackers can demand the money in exchange for the key.

A screenshot of a computer

Description automatically generated

1. ***Monitoring***

The monitoring system was built using the watchdog Python library, which provides a robust framework for watching directories and files for modifications. The watchdog events and logging capabilities enable real-time tracking of all file access and changes, which are essential for detecting patterns typical of ransomware, such as rapid file encryption.

***Setup and Configuration:***

**Environment:** The system was configured to monitor a designated 'critical' directory, which contains various subdirectories and files. These are typical targets in ransomware attacks.

**Tools Used:** Utilized watchdog. Observers for setting up observers and watchdog. Events for defining custom event handlers that respond to specific file system changes.

A screenshot of a computer

Description automatically generated

***Custom Event Handlers:***

**File Modification:** Any modification to files within the monitored directories triggers a log entry that captures the time, file name, and nature of the modification.

**File Creation:** The creation of new files, especially those with suspicious extensions (e.g., .enc), is immediately logged and reported.

**Deletion and Renaming Events:** These are also tracked, as ransomware may attempt to disrupt backup and recovery processes by deleting or renaming critical files.

A screenshot of a computer

Description automatically generated

The monitoring system proved effective in detecting all simulated ransomware activities, with real-time alerts enabling quick response times. The detailed logs generated by the system provided valuable data for the detection algorithms, facilitating rapid and accurate identification of ransomware attacks.[5]

1. ***Detection****:*

The detection system was developed to analyse the logs collected during the monitoring phase, employing both heuristic and signature-based methods to identify potential ransomware activity.[10]

***Analytical Methods:***

**Pattern Recognition:** The system scans log entries for abnormal patterns of file access and modification, such as rapid encryption of multiple files, which are characteristic of ransomware.

**Heuristic Analysis:** Using heuristic rules based on typical ransomware behaviour, such as the creation of .enc files or modifications to system backup files, the system assesses the likelihood of an ongoing attack.

A screenshot of a computer

Description automatically generated

***Integration with Monitoring Data:***

**Data Utilization:** The detection system continuously ingests the real-time data provided by the monitoring system, ensuring that the analysis is based on the most current and relevant information.[15]

**Event Correlation:** By correlating different types of events, such as file creations and modifications with suspicious extensions, the system enhances its accuracy in detecting ransomware.

A screenshot of a computer

Description automatically generated

***Alert System:***

**Real-Time Alerts:** Upon detecting probable ransomware activity, the system generates alerts that are immediately sent to system administrators and security personnel, facilitating swift action.

**Detailed Alert Information:** Each alert includes detailed information about the suspected ransomware activity, such as the affected files, time of detection, and suggested immediate actions.

A screenshot of a computer screen

Description automatically generated

1. ***Mitigation Phase***

The mitigation phase is a critical component of the ransomware response strategy, activated upon the successful detection of a ransomware threat. This phase involves executing predefined countermeasures aimed at limiting the spread of the ransomware, preserving the integrity of unaffected data, and beginning the recovery process.

***Implementation Strategy***

The mitigation strategies are designed to be swift and decisive to counteract the ransomware effectively before it can inflict widespread damage.

**Process Termination:**

**Immediate Action:** Upon detection of a ransomware attack, processes identified as malicious are immediately terminated to halt further encryption.

**Tools Used:** Utilization of psutil for process monitoring and termination, ensuring that any process executing unauthorized encryption is quickly stopped

A screenshot of a computer

Description automatically generated

When we are executing this mitigation file, we are doing it manually. By that time the encryption is being done since the directories are few and their contents are also of less in size.

1. ***RESULTS***

The development and implementation of our ransomware simulation project have provided invaluable insights into the dynamics of cybersecurity threats and the effectiveness of our preparedness strategies. Through this initiative, we've successfully simulated the infection, propagation, detection, and mitigation of a ransomware attack within a controlled environment, thereby achieving a deeper understanding of both the vulnerabilities present in our systems and the robustness of our defensive measures.

Infection Phase Insights: The infection phase, designed to mimic the entry and initial execution of ransomware via a seemingly benign email attachment, underscored the critical need for rigorous email screening and user education. It highlighted how easily attackers could exploit human factors and existing system vulnerabilities.

Monitoring and Detection Efficacy: The monitoring and detection phases demonstrated the capabilities of our current cybersecurity infrastructure to promptly identify and respond to unauthorized activities. By employing advanced monitoring tools and developing bespoke detection algorithms, we were able to detect the simulation in its early stages, preventing widespread damage.[13]

Commitment to Continuous Improvement: In conclusion, while our simulation has proven largely successful, the landscape of cyber threats continues to evolve rapidly. It is imperative that we remain vigilant and committed to continuous improvement. This project serves as a benchmark for our current capabilities and a foundation for future advancements in our cybersecurity efforts.

By addressing the challenges uncovered during this simulation and implementing the recommended improvements, we can better safeguard our digital assets and prepare ourselves for real-world cyber threats. The lessons learned here will guide our ongoing efforts to fortify our systems and protect our stakeholders from the potentially devastating impacts of ransomware and other cyber-attacks.

1. ***REFRENCES***
2. 4 ways hackers use phishing to launch ransomware attacks. (n.d.). Vade | Simple, Sophisticated Email Security Powered by AI. <https://www.vadesecure.com/en/blog/3-ways-hackers-use-phishing-to-launch-ransomware-attacks>
3. Almashhadani, A. O., Kaiiali, M., Sezer, S., & O'Kane, P. (2019). A multi-classifier network-based crypto ransomware detection system: A case study of Locky ransomware. IEEE Access, 7, 47053-47067. <https://doi.org/10.1109/access.2019.2907485>
4. Chrisda. (n.d.). Simulate a phishing attack with attack simulation training. Microsoft Learn: Build skills that open doors in your career. <https://learn.microsoft.com/en-us/microsoft-365/security/office-365-security/attack-simulation-training-simulations>
5. Exploring the potential of the innate immune system for computers network security. (2017). International Journal of Computer Science Issues, 14(2), 24-29. <https://doi.org/10.20943/01201702.2429>
6. Fadheli, A. (2022, December 29). How to make a ransomware in Python. The Python Code - Python Programming Tutorials, Programming Tools & Recipes. <https://thepythoncode.com/article/make-a-ransomware-in-python>
7. Fernet (Symmetric encryption) using a cryptography module in Python. (n.d.). Online Tutorials, Courses, and eBooks Library | Tutorialspoint. <https://www.tutorialspoint.com/fernet-symmetric-encryption-using-a-cryptography-module-in-python>
8. Fernet (symmetric encryption) — Cryptography 43.0.0.dev1 documentation. (n.d.). Welcome to pyca/cryptography — Cryptography 43.0.0.dev1 documentation. <https://cryptography.io/en/latest/fernet/>
9. How does phishing lead to ransomware attacks? (2023, January 6). Egress Intelligent Email Security: Anti-Phishing, Data Loss Prevention & Encryption. <https://www.egress.com/blog/phishing/phishing-leads-ransomware-attacks>
10. Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., & Kirda, E. (2015). Cutting the Gordian knot: A look under the hood of ransomware attacks. Detection of Intrusions and Malware, and Vulnerability Assessment, 3-24. <https://doi.org/10.1007/978-3-319-20550-2_1>
11. Optimal component substitution and multi-resolution analysis Pansharpening methods using a Convolutional neural network. (n.d.). IEEE Xplore. <https://ieeexplore.ieee.org/document/8899299>
12. Python ransomware: An emerging new threat. (2024, February 16).
13. LinkedIn. <https://www.linkedin.com/pulse/python-ransomware-emerging-new-threat-cybernext-ai-nhzqe/?trk=public_post_main-feed-card_feed-article-content>
14. Ransomware archives. (n.d.). Facilities Management Advisor. <https://facilitiesmanagementadvisor.blr.com/tag/ransomware/>
15. Rastogi, S. (2022, August 20). Ransomware with Python. Medium. <https://rsajal.medium.com/ransomware-with-python-b8b34436c12e>
16. Sok: Cryptojacking malware. (n.d.). arXiv.org. <https://arxiv.org/abs/2103.03851>