# Ransomware - Final Report

**Group – 3**

Anila Arram – 11719939

Durga Shankar Dalayi – 11757864

Sai Keerthana Thummalapalli – 11695371

Mounika Yarram - 11710264

## Abstract

Among the most widespread cybersecurity threats, ransomware, encrypts user data and requires the payment of ransoms to fix. In this project, we wish to design a complete ransomware lifecycle in a controlled virtual environment and examine how to detect and mitigate effective strategies. Our creation was a simple ransomware tool capable of encrypting the total contents of a directory structure with AES encryption. For the sake of infection simulation, we also made a made a PDF exploit based on vulnerabilities in outdated versions of PDF readers (Adobe Acrobat in particular). Reverse shell was launched from the PDF and access was granted once the PDF was opened which then executed the ransomware payload.

We used Python's watchdog library for a real time file system monitoring and OSSEC to build a detection system for monitoring and mitigating effects of the simulated ransomware. As unauthorized changes were detected, files that were renamed to a .enc extension or had hash mismatches were flagged by the detection component. It then automatically would quarantine the infected files, restrict their permissions so they couldn't access anything else, and restore clean backups.

One approach of this project reflects on the necessity of layered security which integrates proactive monitoring along with the automation of mitigation to stave off ransomware threats. Despite the fact that SMTP authentication errors caused problems with the email alert feature, the detection and recovery system was up to par. In context of our study, we emphasize the need to be prepared with backup, monitored behavioral, and aware social engineering tactics to prevent infection. The simulation offered a much broader view of ransomware behavior and made it clear to upgrade your cyber security defenses to reduce system vulnerabilities.

**1. Introduction**

Ransomware is a real threat to people, corporate and public sector institutions, securing key files and demanding payment in exchange for the decryption keys. This project is aimed at both, as an academic exercise on the ransomware threat landscape and as a practical exploration of it. In simulation on a virtual machine (VM), we sought to understand the tools, techniques, and effects of ransomware along with what responses and protections will stop the harm from the ransomware.

Therefore, we subdivided our project into several phases to mirror ransomware's lifecycle, which include research, encryption development, infection method, monitoring system, detection algorithm, and mitigation strategy. All phases were implemented and tested in isolation and combined in order to produce a comprehensive ransomware simulation. This malicious PDF came as an infection vector that included the embedded payload that was exploiting known vulnerabilities of obsolete PDF readers. A detection system was developed to watch for suspicious behavior with regards to file modification and renaming and monitoring tools were used to watch file system changes.

As such, this approach provided valuable perspectives of offensive and defensive cybersecurity practices. When focused on real time monitoring and response layered defenses were shown to be very effective at minimizing the damage done by ransomware. By conducting these hands-on projects, we applied theoretical learning through experimentation in a 'safe' lab environment.

**2. Related Works**

Many studies and several real ransomware cases have helped shape our understanding of ransomware and bring us to develop this project. The first ransomware attack that attracted widespread attention for its use of strong RSA encryption and asking Bitcoin for payment was the CryptoLocker in 2013. This was a change away from more primitive and financially motivated attacks toward more advanced and financially motivated ones. A sparkling example is WannaCry, which hit in 2017 using a vulnerability in the Windows SMB protocol unfixed to that time. The virus infected over 200,000 systems worldwide and stayed within the target, the impact of which remained to paralyze organisations such as the Britain's NHS.

In the research conducted at IEEE in year 2020 on file integrity monitoring, hashing and change detection were viewed as paramount for detecting ransomware type behaviors. We used Python's watchdog library to observe for file system events in conjunction with hash comparisons to find unauthorized file system changes in the way that they had also done. In addition to this, we used the result of ACM's 2022 research on applying machine learning to ransomware detection, this discovery demonstrated how pattern recognition can complement static rules-based techniques.

Moreover, the MITRE ATT&CK framework gave a standardized framework to think about aggressor conduct and comprehend aggressor plans inside the fiery and transversal development periods. In order to keep our method of infection realistic and based on observable threat actor behavior, we aligned our infection method with tactics described by MITRE.

The importance and role of these references helped to give foundational concepts, tools, and evaluation methods that drove our decisions along the project. Also, they assisted us in making sure that we had technical credibility, followed best practices, and placed our work within the broader community of cybersecurity offense and defense research.

## 3. Project Approach

We executed a ransomware lifecycle in a virtual machine to make sure that we would remain isolated and safe from the ransomware. The project consisted of six phases of research, implementation of encryption, infection, monitoring, detection, and mitigation. To begin with, we performed a full literature review to know ransomware behaviors along with informing our design. For the encryption, a Python script was developed to recursively encrypt all the files in the target directory in AES encryption and change their extensions to.enc.

The infection method was crafting the malicious PDF using the Metasploit Framework. In this PDF, the payload was embedded as part of a document that exploited known vulnerabilities in legacy PDF readers which would be executed upon opening the document. This meant that the attack vector was a zero-click attack that only required opening the file, as is common in many real-world ransomware situations.

OSSEC was used for monitoring while a custom Python script using the watchdog library was used for monitoring. In this setup, we tracked real time file events that took place in the targeted critical directory namely, creation, modification, deletion etc. File integrity checks, filename analysis, and the usage of. enc extensions were used as behavioral triggers for the detection logic.

Finally, we designed a mitigation strategy to respond to detections involving file access restriction, data isolation and data backup restoration. Combined, these elements formed an entire end to end simulation of a ransomware attack providing both practical insights into the offensive mechanics and defensive strategies of a ransomware attack.

### Step1. Research Phase

Our first step involved an in-depth study of ransomware trends, past attacks, and state-of-the-art defensive mechanisms. Since we studied notorious ransomware like the ones where the ransom was paid in Bitcoin and strong RSA encryption was used (e.g., CryptoLocker) as well as another type, which was WannaCry, where the Windows SMB protocol was involved by using the EternalBlue exploit. From these case studies, we arrived at our technical approach that would inform how hackers using ransomware — through time — learn how to bypass traditional security controls.

Besides, we used peer reviewed research in IEEE and ACM journals as a reference. The sources emphasized the use of file integrity monitoring (FIM), behavioral anomaly detection and a hybrid detection system, which will use both rule and machine learning to determine if the malicious software is present. We consulted MITRE ATT&CK framework in order to align our simulated attack tactics to actual adversary behavior as well as to ensure that our infection method was similar to that of realistic threats.

**Step2. Action (Encryption Development)**

To perform the encryption, there was a custom Python script using the pycryptodome library to apply AES-256 encryption in CBC (Cipher Block Chaining) mode. This is a widely used mode of real ransomware, as it is a balance between security and implementation simplicity. The script was designed to:

- Recursively scan all files in a specified directory
- Encrypt each file's contents securely
- Generate a unique Initialization Vector (IV) for each encryption session
- Rename the encrypted files with a .enc extension

Files were read in binary mode, padded to fit the block size requirement, encrypted, and then rewritten with the IV prepended to support later decryption. We took care to exclude system directories and sensitive OS paths to maintain VM stability.

python

CopyEdit

```
cipher = AES.new(key, AES.MODE_CBC, iv)

ciphertext = cipher.encrypt(pad(data, AES.block_size))

with open(file_path + ".enc", 'wb') as f:

  f.write(iv + ciphertext)
```

**Step3. Infection Method**

To simulate real-world malware distribution, we used the **Metasploit Framework** to generate a **malicious PDF file** embedded with an executable payload. We leveraged the exploit/windows/fileformat/adobe_pdf_embedded_exe module, targeting vulnerabilities in outdated versions of Adobe Reader. Upon opening the PDF, the payload initiated a **reverse shell**, granting remote access to the system and automatically executing the ransomware script.

This infection method reflected typical phishing-based delivery tactics, requiring only a single user action (opening a file) to launch a full attack. It demonstrated the low-effort, high-impact nature of modern ransomware infiltration techniques.

**Step4. Monitoring System**

Our monitoring architecture was composed of two components:

- **OSSEC**, an open-source Host-based Intrusion Detection System (HIDS)
- A custom Python script using the watchdog library

**OSSEC** was configured to monitor file access logs, detect suspicious modifications, and observe system-level alerts. Simultaneously, our **watchdog script** listened for real-time filesystem events such as:

- File creation
- File modification
- File deletion
- Filename/extension changes

This real-time layer of visibility allowed us to detect ransomware behavior patterns like mass renaming and high-volume file changes. Events were logged, and flags were raised if multiple suspicious actions occurred within a short timeframe.

**Step5. Detection Logic**

Our detection logic integrated multiple indicators of compromise (IOCs) and behavioral heuristics. The following triggers were used to detect active ransomware behavior:

- **SHA-256 hash mismatches** between baseline and current versions of files
- **Sudden spikes in file creation/modification events**
- **Files with .enc extensions appearing en masse**

- **Timestamps showing bulk changes in under 30 seconds**

If these thresholds were exceeded, the system generated an internal alert and initiated the mitigation process. Though we integrated Python's smtplib to send email alerts to administrators, **SMTP authentication errors with Gmail** prevented message delivery. This limitation, while not critical to the detection itself, emphasized the importance of testing third-party integrations under real-world constraints.

**Step6. Mitigation Strategy**

As soon as possible, when it detected the breach, our system launched an automated quarantine and restoration routine. First, I identified their infected files, and then I moved them to the quarantine directory and revoking permissions to them using the chmod 000 to prevent further access or spreading.

The system then managed to check clean backup versions that were stored in a previously decided backup archive. Prior to infection, hash values of these backups were stored and integrity checks performed to ensure that the backups were generated manually with tar. It had restored the original directory to one of the backup if the hash of the backup matched the hash of the expected snapshot.

This meant that three important recovery steps were guaranteed.

1. Stopping the spread of ransomware
2. Forensic evidence of the attack should be preserved.
3. Recovering the business with clean data

**Conclusion of Approach**

This multi layer approach gave us a holistic view on offensive as well as defensive aspect of cyber security. We've experienced each part of the kill chain from the creation, deployment and real time detection and even recovery of the ransomware. Most importantly, the simulation brought out the need for proactive monitoring, recovery protocols tested, layered defenses, and the battle against advanced threats such as ransomware.

## 4. Results and Analysis

Finally, the simulation that we performed as ransomware against our monitoring, detection, and mitigation systems identified that it worked as expected for a controlled attack scenario. Using the encryption script, it was possible to obfuscate some files which it renamed to .enc. As soon as the malicious PDF was accessed, our monitoring system was able to detect the creation and modification of it. Real time ransomware detection occurred through the mechanism of file hash comparison and filename analysis.

One of the most important successes was to change permissions so that further file access was automated blocked and be able to restore original, clean file from backup. These actions proved that our system was able to limit damage, stop any further infection, and bring the system back to normal from ransomware triggered data loss.

Noted failure in testing was SMTP authentication errors would prevent the email alert system from sending the emails to administrators. However, even though the alert was set up correctly, Gmail's security rules were blocking the connection and yet another reason third party integrations need to be tested within real world constraints. According to him, this is an area for improvement and represents a move for the future to secure internal messaging or alerting systems.

The results were overall very encouraging regarding the layered defense strategy that was used. Real time detection combined with access restriction and backup restoration made a strong defense on the encryption-based attacks. As we simulated, it did validate our technical approach in parallel with clearly asserting how time wins races and no number of technical approaches can work without timely detection and well-rehearsed recovery plans in a real-world cybersecurity operation.

**References**

1. M. Masum, M. J. H. Faruk, H. Shahriar, K. Qian, D. Lo, and M. I. Adnan, "Ransomware Classification and Detection With Machine Learning Algorithms," *arXiv preprint arXiv:2207.00894*, Jul. 2022. https://arxiv.org/abs/2207.00894arXiv+1ResearchGate+1

2. MITRE Corporation, "MITRE ATT&CK®," https://attack.mitre.org/LinkedIn+12MITRE ATT&CK+12Wikipedia+12

3. MITRE Engenuity, "Top ATT&CK Techniques," https://top-attack-techniques.mitre-engenuity.org/LinkedIn+2Top Attack Techniques+2The Security Validation Platform+2

4. P. M. Jenkins, "Device-Centric Ransomware Detection using Machine Learning," in *Proceedings of the 2022 Annual Computer Security Applications Conference (ACSAC)*, Dec. 2022. https://www.acsac.org/2022/workshops/icss/2022-icss-jenkins.pdfACSAC

5. E. Berrueta, D. Morato, E. Magaña, and M. Izal, "Crypto-ransomware Detection Using Machine Learning Models in File-Sharing Network Scenario with Encrypted Traffic," *arXiv preprint arXiv:2202.07583*, Feb. 2022. https://arxiv.org/abs/2202.07583arXiv+1ACM Digital Library+1

6.  N. Rani and S. V. Dhavale, “Leveraging Machine Learning for Ransomware Detection,” *arXiv preprint arXiv:2206.01919*, Jun. 2022. https://arxiv.org/abs/2206.01919arXiv

7. M. Hirano and R. Kobayashi, “Machine Learning-based Ransomware Detection Using Low-level Memory Access Patterns Obtained From Live-forensic Hypervisor,” *arXiv preprint arXiv:2205.13765*, May 2022. https://arxiv.org/abs/2205.13765arXiv

8.  M. Masum, M. J. H. Faruk, H. Shahriar, K. Qian, D. Lo, and M. I. Adnan, “Ransomware Classification and Detection With Machine Learning Algorithms,” in *Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, Jan. 2022, pp. 0316–0322. https://arxiv.org/abs/2207.00894

9.  A. Vehabovic, N. Ghani, E. Bou-Harb, J. Crichigno, and A. Yayimli, “Ransomware Detection and Classification Strategies,” in *Proceedings of the 2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, Sofia, Bulgaria, Jun. 2022, pp. 316–324. https://arxiv.org/abs/2304.04398

10. R. A. Mowri, M. Siddula, and K. Roy, “Application of Explainable Machine Learning in Detecting and Classifying Ransomware,” in *Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, Jan. 2022, pp. 0323–0329. https://arxiv.org/abs/2210.11235