**Name**: Swarnima Mishra

**Project Title**: Ransomware Analysis and Reverse Engineering

**Institution Name**: United Institute of Technology

**Course Name**: Cyber Security

**Date**: 31/07/2025

**Mentor’s Name**: Ayush Kumar

**Abstract**

This project focuses on the analysis and reverse engineering of the Akira ransomware, a malicious software that encrypts files and demands a ransom for decryption. The primary objective was to understand its behavior, functionality, and impact on affected systems. The analysis was conducted using static and dynamic methods, employing tools such as Ghidra for reverse engineering and Any.Run for behavioral analysis. Key findings revealed the ransomware's encryption mechanisms, its method of displaying ransom notes, and its evasion techniques against detection. This project contributes valuable insights to the cybersecurity community, aiding in the development of countermeasures against similar threats.

**Table of Contents**

* Title Page
* Abstract
* Table of Contents
* Introduction
* Methodology/Approach
* Output Screenshots
* Conclusion
* References

**Introduction**

The ransomware poses a significant threat to individuals and organizations by encrypting files and demanding ransom payments for decryption. This project was chosen to address the growing concern of ransomware attacks and to enhance understanding of their inner workings. The analysis aims to uncover the techniques used by the ransomware, providing insights that can help in developing effective countermeasures. Tools such as Ghidra and Any.

Run were utilized to conduct static and dynamic analyses, respectively.

Ransomware has become a prevalent form of cybercrime, with various strains employing different techniques for encryption and evasion. Previous research has highlighted the importance of understanding ransomware behavior to develop effective detection and prevention strategies. The ransomware, in particular, has been noted for its unique encryption methods and ransom demands, making it a relevant subject for analysis.

**Tools & Methodology**

**Tools**

Ghidra --- Static analysis, decompilation, and string analysis

Hybrid Analysis --- Initial behavioral and static assessment

AnyRun Sandbox --- Dynamic execution and behavioral monitoring

Linux CLI Tools --- Utilities like strings, strace for deep inspection

**Methodology**

1. Static Analysis using Ghidra and CLI tools.

2. Dynamic Analysis using AnyRun Sandbox.

3. Extraction of IOCs, ransom notes, and YARA signatures.

4. Cross-verification of behaviors from static/dynamic output.

**Analysis & Results**

* **Static Analysis**

File Type: 64-bit ELF executable

Architecture: x86-64

Size: ~2.68 MB

Key Observations:

Embedded ransom note detected using string analysis.

CPUID-based Anti-VM behavior spotted.

Presence of unique victim ID for tracking.

* **Dynamic Analysis**

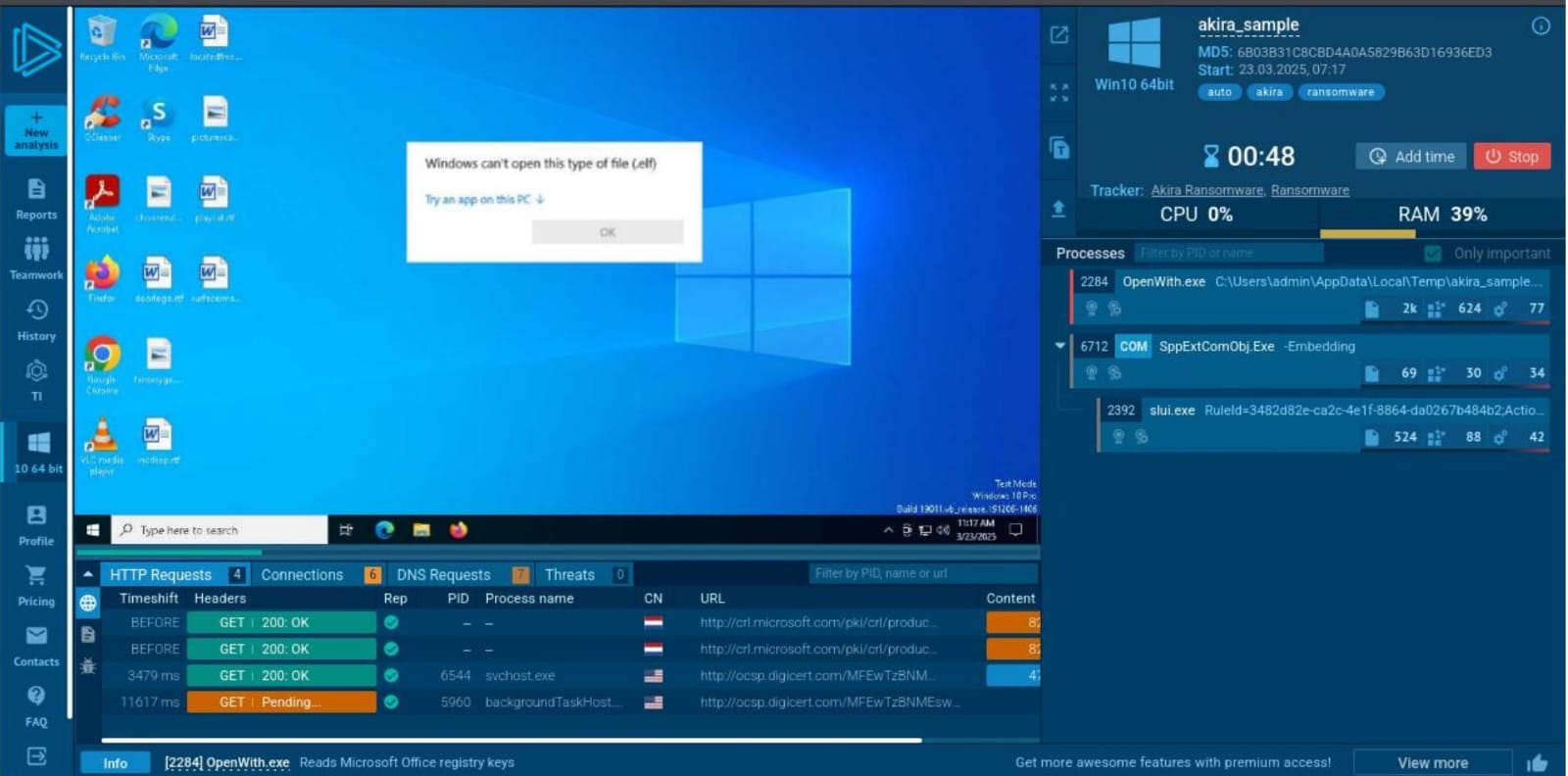
Execution Behavior:

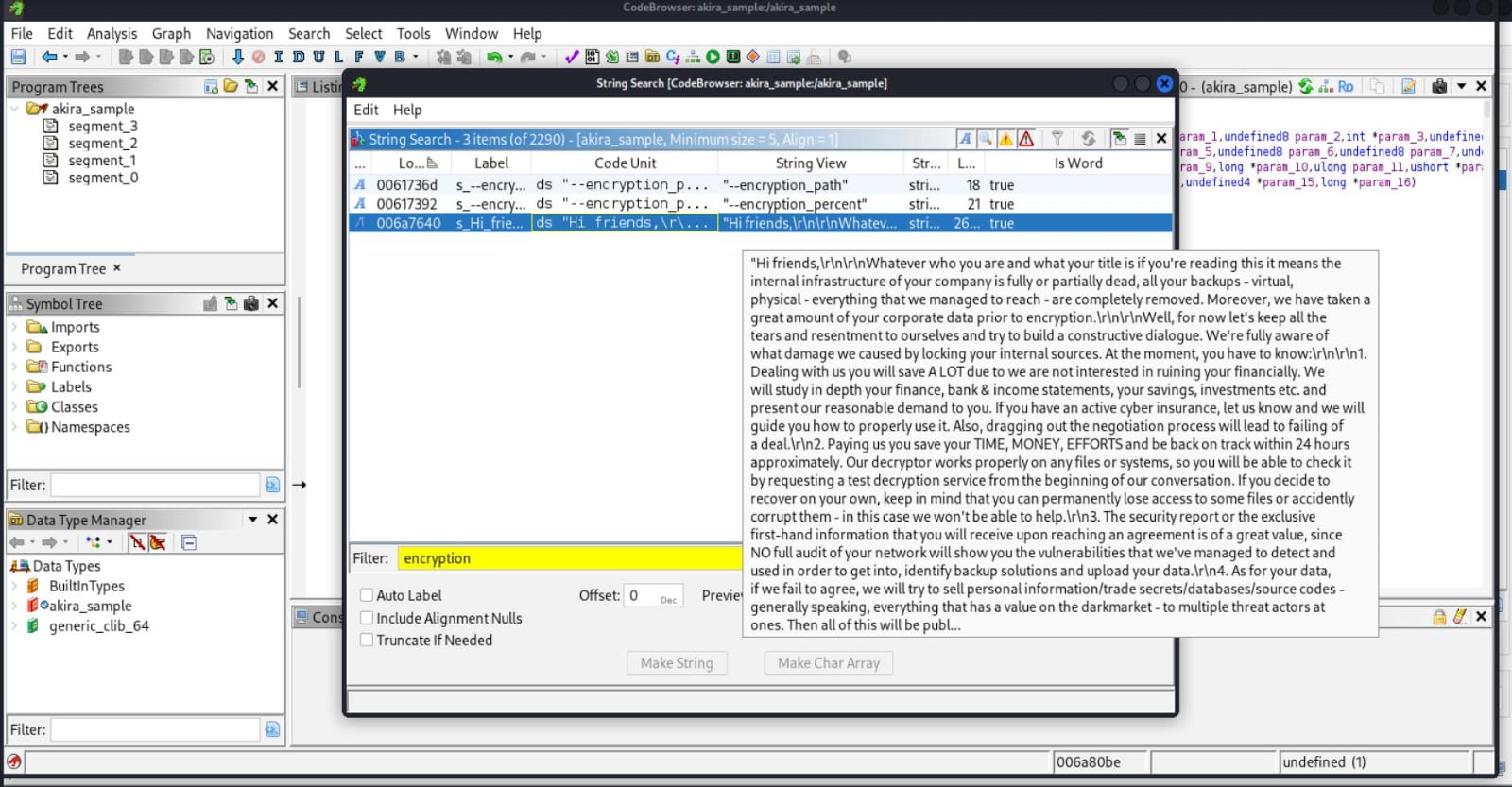
Sample exits with message: "No path to encrypt" if expected directory missing.

System calls logged using strace.

AnyRun shows behavior consistent with anti-analysis mechanisms

**Output & Screenshots**





**Conclusion**

The ransomware sample was effectively analyzed through a combination of reverse engineering techniques. The sample showcased advanced features like anti-VM detection, time-based seeding for encryption, and an embedded ransom note system.

This project strengthened hands-on skills in:

Malware disassembly and inspection,

Behavioral analysis in sandbox environments,

Extracting threat intelligence (IOCs).

It reflects foundational competence in malware research, reverse engineering, and threat analysis.

**References**

AnyRun Sandbox: https://any.run

Hybrid Analysis: https://hybrid-analysis.com

Ghidra Reverse Engineering Tool: https://ghidra-sre.org