HORUS - "Ransomware Canary Protection"

TEAM MEMBERS: -

- Vibhu Yadav: - 2315300033
- Dushyant Nagal: -2315300005
- Prem Singh: 2315300018

MENTOR: -

Dr. Arvind Parsad Sir

Acknowledgment

*We would like to express our heartfelt gratitude to **GLA University, Mathura**, for providing us with the opportunity and environment to carry out our major project — **HORUS: Ransomware Canary Protection System**. This project has been a valuable part of our academic journey, allowing us to apply theoretical knowledge to a practical, real-world cybersecurity challenge.*

*We extend our deepest appreciation to our mentor, **Dr. Arvind Prasad**, for his continuous guidance, motivation, and insightful feedback throughout the development process. His expertise in the field of computer science and cybersecurity played a pivotal role in helping us refine our ideas and implement effective technical solutions.*

*We are also grateful to the **Department of Computer Science and Engineering** for their support and for providing the necessary resources and encouragement to complete this project successfully.*

*Working on HORUS has greatly enhanced our understanding of **ransomware defense mechanisms, event-driven security systems, and process-level threat mitigation**. It has been a rewarding and enriching experience that strengthened our technical, analytical, and collaborative skills.*
Finally, we would like to thank our peers and everyone who contributed directly or indirectly to the successful completion of this project.
.

Abstract

HORUS *is a Python-based cybersecurity solution designed to provide* ***real-time ransomware detection and automated response*** *for Windows systems. The project aims to minimize data loss and system downtime by instantly identifying and neutralizing ransomware activity before significant damage occurs.*

The system integrates two key components — ***Windows Security Event Log Monitoring*** *and a* ***Canary File Protection Mechanism****. By leveraging Windows Event ID* ***4663****, HORUS continuously monitors file access patterns to detect unauthorized modification attempts. Specially deployed canary files act as early-warning traps; any attempt to encrypt or alter these files triggers immediate detection.*

Upon confirmation of a ransomware event, HORUS extracts the ***Process ID (PID)*** *of the malicious process, terminates it, and* ***isolates the system network*** *to prevent further spread of infection. The tool also maintains detailed logs for post-incident analysis and provides a* ***user-friendly Tkinter-based GUI****, allowing users to initialize, monitor, and control the protection system effortlessly.*

Developed in Python using libraries such as ***pywin32****,* ***tkinter****, and standard Windows APIs, HORUS represents a* ***lightweight yet proactive defense mechanism*** *against ransomware threats. It bridges the gap between detection and response, delivering a fast, automated, and effective layer of protection for Windows users*

Introduction

In the modern digital environment, ransomware has emerged as one of the most severe and fast-spreading cyber threats, capable of encrypting thousands of files within minutes and demanding payment for data recovery. Traditional antivirus and signature-based tools often detect such attacks too late or fail to identify the exact malicious process responsible for the infection. This creates an urgent need for ***real-time, automated defense mechanisms*** *that can respond instantly before irreversible damage occurs.*

HORUS *is a Python-based cybersecurity project developed under the guidance of* ***Dr. Arvind Prasad*** *at* ***GLA University, Mathura****, with the objective of providing a* ***proactive ransomware detection and protection system*** *for Windows platforms. The tool is designed to identify, terminate, and isolate ransomware activity using system-level event monitoring and canary file traps. HORUS operates through two primary components:*

- ***Event Log Monitoring*** *– continuously tracks Windows Security Event Logs, specifically* ***Event ID 4663****, to detect unauthorized file access attempts in real time.*
- ***Canary File Mechanism*** *– strategically places decoy files across protected directories, which serve as early-warning sensors for ransomware behavior.*

When ransomware attempts to modify a canary file, HORUS immediately identifies the malicious process through its ***Process ID (PID)****, terminates it, and disconnects the system from the network to prevent the spread of infection. The goal of HORUS is to deliver an* ***automated, lightweight, and user-friendly protection system*** *that bridges the gap between detection and response, ensuring maximum data safety and minimal manual intervention during ransomware incidents.*

Technology Stack:-

***HORUS** is developed entirely in **Python**, offering a lightweight, modular, and efficient framework suitable for real-time cybersecurity applications on Windows systems. The following tools and libraries form the technological foundation of the project:*

- ***Python (3.8+)** – Core programming language used for system automation, process management, and GUI development.*
- ***pywin32** – Enables interaction with **Windows APIs** and **Security Event Logs** (used for reading Event ID 4663 and managing process-level operations).*
- ***tkinter** – Provides a **Graphical User Interface (GUI)** that allows users to easily initialize protection, start monitoring, and view alerts in real time.*
- ***os, subprocess, and time** – Handle system commands, process control, and timing operations for automation and logging.*
- ***PowerShell Integration** – Executes administrative commands (e.g., disabling network adapters) during ransomware mitigation.*
- ***Windows Event Viewer** – Used for validating and monitoring security event logs generated during detection.*
- ***Task Manager / Command-line Tools** – Assist in process verification, PID identification, and testing during development.*

*Together, these technologies enable HORUS to deliver a **robust, automated, and real-time ransomware defense system** that functions efficiently with minimal resource usage on Windows platforms.*

Module 1: Event Log Monitoring

(Real-Time Ransomware Detection & Process Identification)

Overview: -

*Event Log Monitoring is the first and core module of the **HORUS** system, responsible for continuously analyzing **Windows Security Event Logs** to detect early signs of ransomware activity. It specifically tracks **Event ID 4663**, which is generated whenever a process attempts to access, modify, or delete a protected file.*

*This module plays a critical role in **real-time threat detection** by linking file access events directly to the **Process ID (PID)** responsible for the operation. Unlike traditional antivirus tools that rely on static signatures, HORUS leverages this event-driven approach to identify **new or unknown ransomware variants** through their behavioral footprint.*

Once an unauthorized file access attempt is detected, the module extracts key event details — including the timestamp, process name, user context, and PID — and flags the process as suspicious. This information is then passed to the mitigation module for automated action.

*By integrating deeply with the Windows security framework, the Event Log Monitoring module ensures that ransomware activity is **identified instantly and precisely**, reducing the window for potential data loss. It serves as the **foundation** of the HORUS protection workflow, enabling proactive defense against file encryption attacks in real time.*

Key Features: -

- ***Real-Time Ransomware Detection:***
  *Continuously monitors **Windows Security Event Logs (Event ID 4663)** to instantly detect unauthorized file access or encryption attempts.*
- ***PID-Based Process Identification:***
  *Accurately identifies the **exact Process ID (PID)** of the ransomware or malicious program responsible for the attack.*
- ***Automated Mitigation:***
  *Immediately **terminates** the malicious process and **disables all network adapters** to prevent lateral movement or data exfiltration.*
- ***Canary File Protection System:***
  *Deploys strategically placed **decoy (canary) files** that act as early-warning traps, triggering alerts the moment ransomware attempts to encrypt or modify them.*
- ***Event Log Integration:***
  *Seamlessly integrates with the **Windows Event Log** infrastructure, ensuring native-level system monitoring and precise event tracking.*
- ***User-Friendly GUI:***
  *Provides an intuitive **Tkinter-based interface** that allows users to initialize the system, start protection, and view live security alerts with ease.*
- ***Incident Logging:***
  Automatically records all detection and mitigation events in structured log files for later analysis, auditing, or forensic reporting.
- ***Lightweight and Automated:***
  Runs efficiently in the background with minimal CPU usage, requiring no manual supervision once initialized.

Technology Used: -

- ***Language:*** *Python 3.8+ (for core logic, event monitoring, and GUI development)*
- ***Event Monitoring:*** *Windows* ***Security Event Logs*** *(specifically Event ID 4663) accessed via the* ***pywin32*** *library*
- ***GUI Framework: Tkinter****, providing a lightweight graphical interface for system initialization and real-time protection monitoring*
- ***System Operations:*** *Built-in Python libraries —* ***os****,* ***subprocess****, and* ***time*** *— for process handling, privilege checking, and automation*
- ***Network Isolation:*** *Executed through integrated* ***PowerShell*** *commands to disable active network adapters instantly during an attack*
- ***Audit Policy Configuration:*** *Utilizes* ***auditpol*** *and* ***SACL (System Access Control List)*** *settings for automated Windows auditing setup*
- ***Execution Environment:*** *Optimized for* ***Windows 10/11*** *systems with* ***Administrator privileges****, tested within virtual and sandboxed environments for security validation*

Working:-

Step 1: System Initialization

Purpose:
Enable security auditing and prepare the system for real-time ransomware detection.
Method:

- ✓ *Verifies that HORUS is running with **Administrator privileges** (relaunches itself with elevation if required).*
- ✓ *Enables **Windows File System Auditing** using the command:*
  auditpol /set /subcategory:"File System" /success:enable /failure:enable
- ✓ *Deploys multiple **canary files** (decoy files) across protected directories such as Documents, Downloads, and Desktop.*
- ✓ *Applies **SACL (System Access Control List)** rules to each canary file, ensuring Windows generates **Event ID 4663** whenever any process attempts to modify or delete these files.*

Output:
System configured for event monitoring.
Canary files placed and protected in predefined directories.

Step 2: Real-Time Monitoring
Purpose:
Continuously monitor Windows Security Event Logs to detect unauthorized file access or ransomware activity.
Method:

- *The monitoring thread listens to **Event ID 4663** entries in the Windows Security Log.*
- *When such an event occurs, the log is parsed to extract key details such as:*
  Process ID (PID), Process Name, User Account, Timestamp, and Target File Path.
- *The extracted process is verified against a list of **trusted processes** (e.g., explorer.exe, notepad.exe).*
- *If the process is **not trusted**, the detection is marked as a potential ransomware event.*

Output:
Live alerts displayed in the HORUS GUI.
Event details logged in structured text or CSV format for later analysis.

Step 3: Mitigation and Response

Purpose:

Take instant automated action to stop the ransomware and secure the system.

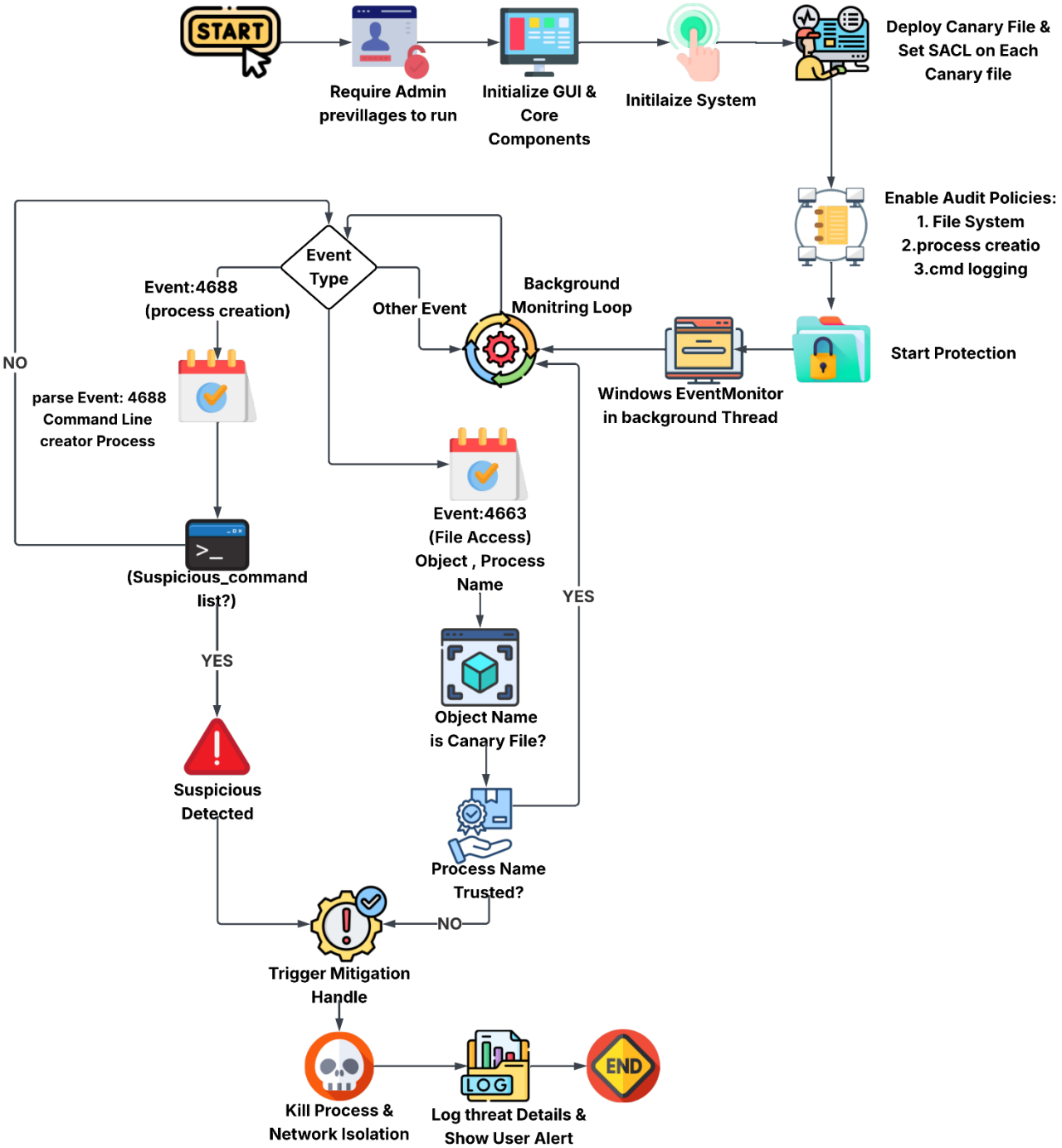
Method:

- *Executes the command:*
  taskkill /F /PID <malicious_PID>
  *to **terminate the malicious process**.*
- *Runs a **PowerShell command** to disable all network adapters, preventing ransomware from spreading to other devices.*
- *Displays a **"Ransomware Detected – System Isolated"** alert to notify the user.*

Output:

Malicious process terminated.
Network connections disabled.
Security log updated with full incident details (PID, process name, and timestamp).

Overall-Flowchart: -



Module 2: Mitigation and Isolation Engine

(Automated Threat Containment & System Recovery)

Overview: -

*The **Mitigation and Isolation Engine** is the second key module of the **HORUS** protection system, responsible for taking **immediate defensive action** once ransomware activity has been detected. While the Event Log Monitoring module handles detection, this component ensures that the identified threat is swiftly neutralized and prevented from causing further damage.*

*Upon confirmation of a ransomware event, the Mitigation Engine retrieves the **Process ID (PID)** of the malicious program from the Windows Security Log and executes a termination command using elevated privileges. This ensures that the ransomware process is **forcibly stopped within seconds**, halting ongoing encryption or modification of system files.*

*In addition to process termination, the module triggers **network isolation** to contain the infection. Using integrated PowerShell commands, it disables all active network adapters, effectively cutting off the compromised system from both local and remote networks. This prevents the ransomware from propagating laterally or exfiltrating sensitive data.*
The module also logs each mitigation event — including the PID, process name, timestamp, and user context — for further forensic analysis. An alert is displayed to inform the user that ransomware activity was detected and neutralized successfully.

*By combining **automated process control**, **network isolation**, and **incident logging**, the Mitigation and Isolation Engine ensures that HORUS not only detects ransomware threats but also **responds to them instantly**, safeguarding data integrity and system availability.*

Key Features:

- ***PID-Based Detection:***
  *Precisely identifies the **Process ID (PID)** of the ransomware or malicious process responsible for unauthorized file access.*
- ***Canary File Monitoring:***
  *Deploys multiple **decoy (canary) files** across sensitive directories that act as early-warning traps for ransomware behavior.*
- ***Event Log Integration:***
  *Continuously monitors **Windows Security Event Logs (Event ID 4663)** to detect suspicious file operations in real time.*
- ***Automated Mitigation:***
  *Instantly executes defense actions such as **process termination** and **network isolation** upon detection of a threat.*
- ***System Isolation:***
  *Uses **PowerShell-based commands** to disable network adapters, preventing ransomware from spreading to other systems.*
- ***User Interface:***
  *A clean and responsive **Tkinter-based GUI** that allows users to initialize protection, start monitoring, and view live security alerts.*
- ***Comprehensive Logging:***
  Records all detection events, PIDs, timestamps, and mitigation details in structured logs for later analysis and forensic review.
- ***Lightweight Performance:***
  Designed for continuous background operation with minimal CPU and memory usage, ensuring no disruption to system performance.

Technologies Used:

- ***Programming Language:** Python 3.8+ (core implementation and automation)*
- ***Python Libraries:***
  - ***pywin32:** To access Windows APIs and Security Event Logs.*
  - ***tkinter:** For building the graphical user interface.*
  - ***os, subprocess, time:** For system operations, process handling, and timing control.*
- ***System Utilities:***
  - ***auditpol:** Enables Windows auditing policies for file system monitoring.*
  - ***PowerShell:** Executes network isolation and process termination commands.*
- ***Event Tracking:** Windows **Security Event Viewer** for Event ID 4663 monitoring and validation.*
- ***Execution Environment:** Optimized for **Windows 10/11** with **Administrator privileges**.*

Working: -

*The **HORUS** system operates through an automated, multi-stage workflow designed to detect and neutralize ransomware in real time with minimal user intervention. It integrates Windows Event Log monitoring with canary file traps to provide proactive and immediate system protection*

1. Initialization Phase

*HORUS begins by verifying **Administrator privileges** to ensure full access to system resources. Once elevated, it enables **Windows File System Auditing** using the auditpol command and deploys **canary files** across protected directories such as Documents, Downloads, and Desktop.*
*Each canary file is assigned a **System Access Control List (SACL)** rule that instructs Windows to log **Event ID 4663** whenever any process attempts to modify or delete the file. This setup ensures that all suspicious activities are immediately logged and traceable to the responsible process.*

---

2. Real-Time Monitoring

*After initialization, HORUS enters continuous monitoring mode. A background thread listens to the **Windows Security Event Log** for any **Event ID 4663** entries triggered by canary file access.*
Upon detection, the system extracts and analyzes essential event data such as:

- *Process ID (PID)*
- *Process Name*
- *Access Type (Write/Delete)*
- *Target File Path*
- *User Context*
- *Timestamp*

*The extracted details are then compared against a predefined **trusted process list** to distinguish between legitimate and malicious activity. Any unrecognized or suspicious process is immediately flagged as a ransomware threat.*

---

3. Threat Verification and Response

Once a ransomware event is confirmed, HORUS activates its automated mitigation pipeline:

- *Executes the command:*
  taskkill /F /PID <malicious_PID>
  *to **forcefully terminate** the detected process.*
- *Launches a **PowerShell command** to **disable all active network adapters**, preventing ransomware from spreading across connected systems.*
- *Displays an on-screen **alert message** notifying the user that ransomware activity was detected and the threat has been contained.*

---

4. Logging and Reporting

*After mitigation, HORUS automatically records all relevant details — including PID, process name, event timestamp, and mitigation status — in structured **log files (TXT/CSV format)**.*

*These logs can later be used for **forensic analysis**, **incident review**, and **system recovery validation**.*

*The entire detection-to-response process typically executes within **seconds**, ensuring that ransomware activity is neutralized before any significant encryption or data loss can occur.*

Mitigation-*FlowChart: -*