  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Synopsis

**Created By:**

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

1. Introduction

**Purpose and Overview: -**

* **Purpose:**

To develop MANTIS (Machine learning Anti-ransomware Neural Threat Intelligence System) to neutralize and mitigate ransomware attack

* **Overview:**
  + - Uses canary files, entropy monitoring, and process behavior analysis.
    - Employs a hybrid ML model that combines supervised and unsupervised learning to catch both known and zero-day ransomware.
    - Provides automatic response options such as killing processes, quarantining files, and isolating the network.
    - Includes a dashboard that shows system health, alerts, logs, and ML insights.

**2. Problem Statement**

Ransomware attacks have become increasingly sophisticated, often bypassing signature-based defenses and traditional antivirus solutions. Static detection methods fail to capture the dynamic behavior of ransomware during execution, making systems vulnerable. A machine learning–driven approach that leverages temporal features and system behavior patterns is essential to detect and mitigate ransomware in real time.

**3. Proposed Solution**

The system introduces a real-time ransomware defense mechanism combining canary file monitoring, entropy analysis, and process behavior tracking for early detection. A hybrid machine learning model is employed: supervised learning detects known ransomware families, while unsupervised learning identifies zero-day variants. A fusion layer enhances accuracy by combining both approaches.

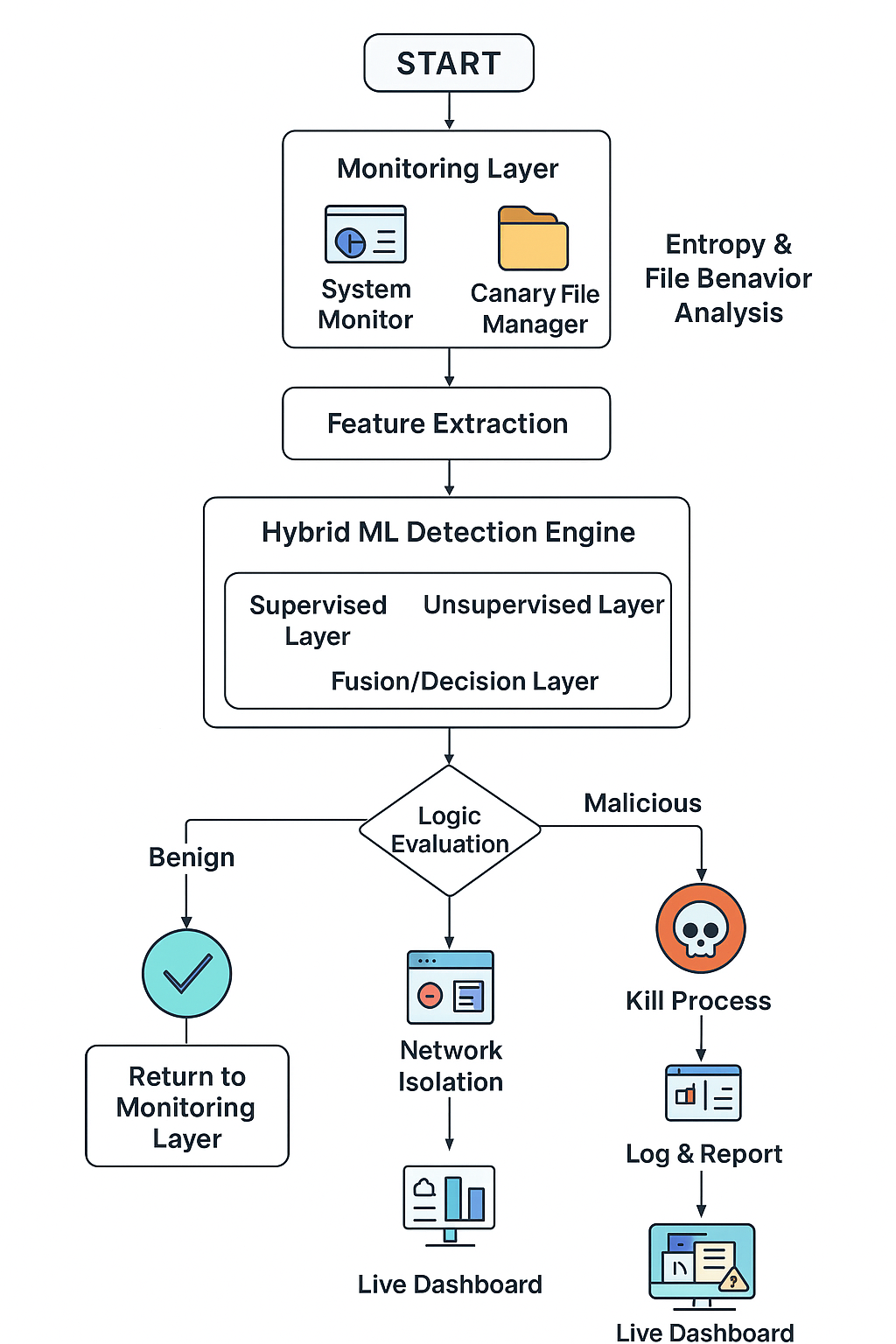
Upon detection, the system automatically responds by terminating malicious processes, quarantining suspicious files, and isolating the network to prevent lateral spread. A dedicated dashboard provides real-time telemetry, entropy graphs, alerts, and forensic logs. All incidents are recorded in a structured format to support investigation and continuous model retraining.

**Addressing the Problem:**

* **Early Detection**: Uses canary files, entropy tracking, and behavior analysis to identify ransomware at its initial stage.
* **Hybrid ML Model**: Detects both known and unknown ransomware through supervised and unsupervised learning with fusion logic.
* **Automated Response**: Neutralizes threats instantly by killing processes, quarantining files, and isolating networks.
* **Forensic Logging**: Maintains tamper-proof logs for analysis and retraining.
* **Dashboard Visibility**: Offers real-time system monitoring, alerts, and forensic data.

**Key Goals:**

* Real-time ransomware detection.
* Resilient hybrid ML-based defense.
* Automated, immediate threat response.
* Structured forensic readiness.
* User-friendly dashboard visibility.
* System stability and scalability for future enhancements.

**Flowchart :-**

**4. Key and Innovative Functionality:** **Hybrid Ransomware Detection System**

* **Canary Files:** : Fake "bait" files serve as early warnings if ransomware attempts to encrypt them.
* **Entropy Check:** Spot sudden randomness in files, which clearly indicates encryption.
* **Hybrid ML Model:** Combines trained models with anomaly detection to identify both known and new ransomware.
* **Auto Response:** Can terminate harmful processes, quarantine files, and block the network immediately.
* **Process Monitoring:** Monitors unusual process behavior and suspicious system calls.
* **Dashboard & Logs:** User-friendly dashboard displays alerts, graphs, and saves logs for further investigation.
* **Self-Learning:** The system continues to improve by automatically learning from new attacks.

**5. Tools and Technology Required**

* **Programming Language:** Python (core development, machine learning, and monitoring scripts)
* **Libraries/Frameworks:**
* Scapy (network packet monitoring and analysis)
* Psutil (system resource and process tracking)
* Scikit-learn / TensorFlow (machine learning model training and evaluation)
* **Database:** SQLite / MySQL (log storage and training data management)
* **Environment:** Virtual Machines, Docker, or Sandboxed setups (safe execution and testing of ransomware behavior)
* **Operating System Support:**
* **Linux –** primary environment for model development, safe testing, and monitoring**.**
* **Windows –** compatibility testing, since ransomware predominantly targets Windows systems.
* **Monitoring/Validation Tools:** Wireshark, Sysmon (traffic capture, process tracing, and validation of ransomware activity).

**6. Conclusion**

The project proposes an intelligent ransomware detection and response system that monitors file entropy, canary files, and process behavior for early threat identification. A hybrid machine learning model enhances accuracy by detecting both known and unknown ransomware variants.

The system further strengthens security through automated responses such as process termination, file quarantine, and network isolation, while providing real-time dashboards and forensic logs for administrators. With its self-learning capability, the solution remains adaptive, reliable, and future-ready against evolving ransomware threats..