FACULTY OF SCIENCE, ENGINEERING AND COMPUTING

School of *Computer Science & Mathematics*

BSc DEGREE
IN

Cyber Security & Digital Forensics (Top-Up), Kingston University (UK)

PROJECT PROPOSAL

Name: Wanni Arachchige Nayana Anjana Shakthi Monarawila

ID Number: K2558179

Project Title: AI-Powered Ransomware Early Detection System

Project Type: Build

Date: 25-11-2025

Supervisor: Mr. Madhawa Ranawaka

Kingston University London

Did you discuss and agree the viability of your project idea with your supervisor?	Yes
Did you submit a draft of your proposal to your supervisor?	Yes
Did you receive feedback from your supervisor on any submitted draft?	Yes

Abstract

One of the most damaging cyber threats affecting people, companies, and vital infrastructures globally is ransomware. The effectiveness of conventional signature-based antivirus programs against contemporary ransomware variations, such as fileless ransomware, polymorphic malware, and zero-day attacks, is diminishing. This research suggests an AI-Powered Ransomware Early Detection System that can recognize suspicious activity prior to encryption to overcome this difficulty. The system analyzes system-level processes such file manipulations, process creation, registry updates, and resource utilization trends using machine learning methods. Technology can classify behaviors in real time and produce early alarms by training models on datasets of tagged ransomware and benign behavior.

This proactive strategy seeks to improve cybersecurity resilience, minimize data loss, and stop financial harm. The project involves gathering datasets, training models, developing real-time monitoring prototypes, and evaluating performance based on recall, accuracy, precision, and false-positive rates. The result is a lightweight, host-based detection system that adds to the expanding field of AI-driven cyber defense and is appropriate for both academic demonstration and practical applications.

Contents

List of Figures/Tables

Glossary of Terms

1. Introduction & Background

1.1 Introduction

Attacks using ransomware have grown to be one of the biggest cybersecurity risks in the world, impacting people, businesses, government agencies, and hospitals. Ransomware, in contrast to conventional malware, encrypts files and requests a payment to unlock them. Contemporary ransomware variations can spread quickly by taking use of zero-day vulnerabilities, are extremely elusive, and are frequently undetectable by signature-based antivirus programs.

Early detection is now essential due to the growing frequency and intensity of attacks. Conventional techniques are unable to identify unidentified ransomware or novel variations. A proactive solution is provided by artificial intelligence (AI) and machine learning (ML), which can detect suspicious patterns suggestive of ransomware by monitoring system behavior rather than depending on recognized signatures.

The goal of this project is to develop a host-based artificial intelligence system that can track system activity, including file access patterns, process creation, CPU/memory utilization, and registry changes, and determine whether or not these behaviors point to ransomware. By identifying ransomware before it encrypts files, consumers or organizations can lessen the impact of assaults.

1.3 Background and Motivation

Ransomware has become one of the most severe cybersecurity threats globally, targeting individuals, organizations, and critical infrastructure. High-profile attacks like WannaCry (2017) and Ryuk ransomware have caused widespread financial and operational damage, highlighting the need for early detection systems (Europol, 2022). While signature-based antivirus solutions and network monitoring have been used, they often fail against zero-day ransomware and polymorphic variants, leaving systems vulnerable to rapid data loss.

Recent research shows that ransomware exhibits detectable early-stage behaviors, such as abnormal process execution, rapid file changes, registry modifications, and CPU/disk

spikes (Anderson et al., 2020; NIST, 2021). Although AI and machine learning techniques have been applied for malware detection, most solutions are network-dependent or offline, and few are lightweight host-based systems suitable for personal computers or small organizations.

This project focuses on host-based, real-time ransomware detection using machine learning. By monitoring local system behavior with tools like psutil and watchdog, and classifying activity with Python-based ML models (Random Forest, Decision Tree, SVM), the system can provide early alerts before encryption occurs. This approach addresses the research gap, offering a practical, lightweight solution that improves early detection and user protection against ransomware attacks.

References:

- Europol. (2022). *Ransomware Threat Assessment Report*.
- NIST. (2021). *Behavioral Malware Detection Framework*.
- Anderson, H., et al. (2020). *Machine Learning for Malware Detection.* IEEE Security & Privacy.

1.4 Problem in brief

Ransomware detection techniques, such network monitoring and signature-based antivirus software, are primarily reactive and unable to stop data loss from novel or unidentified ransomware strains. Many AI-based solutions are either network-dependent, requiring several systems and equipment, or offline, examining logs after an attack, which restricts their usefulness to people or small businesses.

The absence of a lightweight, host-based, real-time system that can keep an eye on a user's local computer, identify ransomware activity early, and send out instant alerts prior to file encryption is the primary issue this project attempts to solve. This project attempts to close this gap and provide a workable solution appropriate for small business settings and home computers by concentrating on host-based monitoring.

2. Aim & Objectives

Write Aim and Objectives of the project under a separate heading as follows.

1.2.1 Aim

The aim of this project is to develop an **AI-powered system for early detection of ransomware** by analyzing system behavior using machine learning techniques.

1.2.2 Objectives

- Conduct a critical review of ransomware detection techniques and attack behaviors.
- Critically study AI and machine learning technologies that can be used to detect ransomware early.
- Design and develop an AI-powered system for early detection of ransomware using system behavior analysis.
- Evaluate the proposed system using performance metrics such as accuracy, precision, recall, F1-score, and false positives.
- Prepare final documentation, including system design, methodology, implementation steps, results, and conclusions.

3. Technologies & Resources

The technologies, software, hardware, and datasets needed to put the AI-powered ransomware detection system into practice are described in this section. It describes the rationale behind each technology's selection, how it will advance the project, and the source of the data.

2.1 Technologies Used

1. Programming Language

- **Python 3.x** - Chosen for its simplicity, readability, and extensive support for data analysis and machine learning. Python allows rapid prototyping of AI models and integration with system monitoring tools.

2. Machine Learning Libraries

- **Scikit-learn** - Provides easy-to-use implementations of machine learning algorithms such as Random Forest, Decision Tree, and Support Vector

Machine (SVM). These algorithms are suitable for classifying system behavior as benign or malicious.

- **Pandas & NumPy** - Essential for preprocessing datasets, cleaning data, handling missing values, and transforming features into formats suitable for machine learning models.
- **Matplotlib & Seaborn** - For visualizing patterns in the data, analyzing anomalies, and presenting model performance metrics such as accuracy, recall, and precision.

3. System Monitoring Tools

- **Psutil** - A Python library is used to collect real-time system-level data such as CPU usage, memory consumption, disk activity, and running processes. These metrics act as features for detecting abnormal behaviors indicative of ransomware.
- **Watchdog** - Monitors file system events such as file creation, deletion, and modification. This helps capture ransomware activities like mass file encryption.

4. Data Sources

- **CIC Ransomware Dataset**
  Publicly available dataset containing labelled ransomware and benign behaviors. Provides reliable data for training and testing ML models.
- **EMBER Dataset**
  Malware behavior dataset with rich features, supporting ML-based detection.
- **Custom Sandbox Logs**
  Generated by running ransomware in a controlled virtual machine environment. These logs include system behavior, file operations, and process activities, enhancing model accuracy with practical, real-world data.

5. Development & Testing Tools

- **VS Code / Jupyter Notebook**
  For developing Python scripts, testing algorithms, and visualizing results.
- **VirtualBox / VMware**
  Safe environments to execute ransomware for dataset generation and system testing without affecting the main system.
- **GitHub**
  For version control, collaborative development, and secure storage of project code.

2.2 Resource Requirement

1. **Hardware Requirements**
   - **Laptop -** ASUS Vivo Book, Intel Core i7 (11th Gen), 16 GB RAM, 1 TB SSD + 1 TB HDD.
   - **Operating System -** Windows 11 Pro.
   - Capable of running Python, ML libraries, system monitoring scripts, and virtual machines simultaneously.

2. **Software & Libraries**
   - Python 3.x and ML libraries (Scikit-learn, Pandas, NumPy, Matplotlib, Seaborn)
   - System monitoring libraries (psutil, watchdog)
   - VirtualBox / VMware for sandbox testing
   - GitHub for version control and backup

3. **Data Requirements**
   - Publicly available datasets (CIC Ransomware Dataset, EMBER)
   - Custom logs generated from controlled sandbox testing for additional training and validation

4. **Other Resources**
   - Internet access for research, dataset downloads, and software installation
   - External storage for backing up datasets and project files.

4. Methodology & Work plan

Agile development will be used for this project since it encourages iterative development, ongoing improvement, and quick feedback integration. Multiple rounds of data preprocessing, model training, evaluation, and improvement are necessary for cybersecurity systems, particularly AI-driven detection algorithms. Agile enables the project to be broken up into brief sprints, each of which produces a quantifiable result such a detection module, model prototype, dataset preparation, or evaluation framework. This guarantees adaptability, lowers risk, and permits ongoing system performance and accuracy validation over the course of the project.

Early detection of mistakes or difficulties, such as data imbalance, overfitting, false positives, or problems with system performance, is also supported by the Agile methodology. Before going on to the next phase, each iteration concentrates on testing and improvement, guaranteeing a dependable and efficient ransomware-detection system.

Main Phases of the Project

1. **Requirement Analysis & Literature Review**
   Understand ransomware behavior, study previous detection techniques, define system specifications.
2. **Data Collection & Preprocessing**
   Gather public ransomware datasets and normal system activity logs, clean and label data for model training.
3. **Model Development**
   Train machine learning models (Random Forest, Decision Tree, or Neural Networks) to detect ransomware patterns.
4. **System Integration & Real-Time Monitoring**
   Implement detection on endpoints; generate alerts and logs.
5. **Testing & Evaluation**
   Simulate ransomware attacks in a controlled virtual environment; evaluate model accuracy, false positives, and detection speed.
6. **Iteration & Optimization**
   Improve model based on testing results

7. **Documentation & Reporting**

Prepare final report, user guide, and project presentation.

Main Phases, Milestones & Deliverables

Table 1 Main Phases, Milestones & Deliverables

Phase	Activities	Milestones	Deliverables	Contingencies
Phase 1 - Requirement Analysis & Literature Review	Study ransomware patterns, AI techniques, and existing solutions	Milestone 1	Research summary, system requirements document	**Insufficient data** - Expand literature sources
Phase 2 - Data Collection & Preprocessing	Gather ransomware & normal activity datasets; clean, normalize, and label data	Milestone 2	Preprocessed dataset ready for training	**Missing data** - Use synthetic dataset generation
Phase 3 - Model Development	Train ML models for anomaly detection; evaluate preliminary results	Milestone 3	Trained AI detection model	**Low accuracy** – Try alternative algorithms (SVM, Ensemble)
Phase 4 - System Integration & Real-Time Monitoring	Implement real-time monitoring on endpoints; generate alerts	Milestone 4	Functional endpoint monitoring system	**Integration errors** -Use virtualization for safe testing
Phase 5: Testing & Evaluation	Simulate ransomware attacks; evaluate model performance metrics	Milestone 5	Performance report with accuracy, detection speed, false positives	**High false positives** - Fine-tune thresholds and retrain model
Phase 6 - Iteration & Optimization	Optimize AI model, improve detection accuracy, reduce false positives	Milestone 6	Optimized AI detection system	**Performance lag** - Optimize code and reduce data dimensionality
Phase 7 - Documentation & Reporting	Prepare final documentation, report, and presentation	Milestone 7	Project report, user guide, presentation	**Delay in report** -Start documentation alongside development

Ghantt Chart for Project shedule

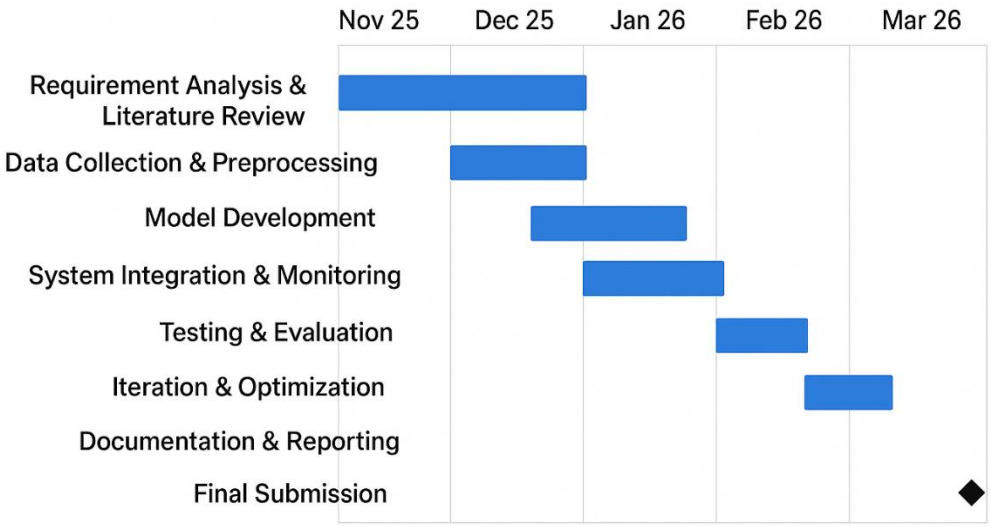

Figure 1 Ghantt Chart for Project schedule

5. Proposed Solution

The proposed solution is an **AI-powered system designed to detect ransomware attacks early** by monitoring system and network behavior. In contrast to conventional signature-based antivirus software, our system identifies ransomware in real-time by focusing on behavioral patterns such odd file alterations, unexpected system calls, and irregular network activity.

The system learns typical system behavior and identifies variations brought on by ransomware using machine learning models, such as Random Forest, Decision Trees, or Neural Networks. When it is detected, it can optionally initiate preventive measures to stop data encryption and generate alerts for administrators. Over time, the AI model's detection accuracy increases as it continuously learns from fresh ransomware samples.

This solution is lightweight and deployable on standard computing environments (e.g., Windows 11 Pro, 16GB RAM laptop), making it practical for both personal and organizational use. By providing proactive detection, the system reduces reliance on

reactive security methods and mitigates potential data and operational losses caused by ransomware attacks.

5.1 Suggested Starting Point

The project will begin with a literature review on ransomware behaviors and AI-based detection techniques to understand existing solutions and identify gaps. Next, relevant datasets containing ransomware and normal system activity will be collected from public sources and prepared for analysis by cleaning, labeling, and normalizing the data. Following this, a preliminary AI detection model will be developed using machine learning algorithms such as Random Forest or Neural Networks and tested in a controlled virtual environment to ensure safe experimentation. Once initial results are obtained, the model will be refined, and the system integration for real-time monitoring and alerts will be implemented on the laptop. This phased approach ensures that the project progresses systematically, starting with research and data preparation, moving to model development, and then practical deployment.

6. Discussion

In addition to addressing important cybersecurity issues, the AI-powered ransomware detection method raises several ethical, legal, sociological, and security issues.

1. **Legal Concerns**

   The project must make sure that any data sets used for testing and training were acquired lawfully and adhere to data protection and copyright laws. To prevent unwanted surveillance, the system must be deployed in real-world settings in accordance with cybersecurity regulations, especially those pertaining to user activity tracking.

2. **Ethical Concerns**

   When keeping an eye on system behavior, ethical concerns include protecting privacy and preventing the improper use of data that has been gathered. Sensitive personal or corporate data should not be compromised by the AI system; it should only access data required for ransomware detection.

3. **Societal Concerns**

By averting monetary loss and operational interruption, the implementation of efficient ransomware detection improves societal faith in digital systems. However, to preserve public trust, poor deployment may result in unforeseen repercussions like false alarms or service outages.

4. **Security Concerns**

The system itself needs to be secure since attackers might target a weak AI detection tool. It is crucial to take precautions like secured data storage, safe coding techniques, and frequent updates. In order to prevent needless inconvenience and guarantee a prompt reaction to actual ransomware threats, the system should also reduce false positives.

In general, the initiative strikes a compromise between the necessity of proactive cybersecurity, ethical monitoring, responsible data handling, and adherence to social and legal norms. The system can offer secure, dependable, and efficient ransomware prevention by taking these factors into account.

7. References / Bibliography

1. Stallings, W. (2021). *Computer Security: Principles and Practice* (5th ed.). Pearson.
2. Kolodenker, E., Koch, W., Stringhini, G., & Egele, M. (2017). Ransomware: Emergence, Detection, and Mitigation. *Journal of Cybersecurity*, 3(2), 1–14. https://doi.org/10.1093/cybsec/tyx003
3. Symantec. (2023). *Internet Security Threat Report*. Symantec Corporation. Retrieved from https://www.broadcom.com/company/newsroom/press-releases
4. Microsoft Malware Classification Challenge. (2015). *Microsoft Malware Dataset*. Kaggle. Retrieved from https://www.kaggle.com/c/malware-classification
5. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
6. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., … Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. *arXiv preprint arXiv:1603.04467*. Retrieved from https://www.tensorflow.org

Appendices