Intelligent Network Intrusion Detection Using Machine Learning

by

Student’s Name…

This project is submitted to the Gannon University graduate faculty in

partial fulfillment for the degree Master of Science in Computer and Information Science.

Option: <Degree Option>

Approved:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Deacon Stephen T. Frezza, Ph.D., PSEM |  | Jeremy Cannell, MS |
| Advising Professor in Charge of Research |  | Committee Member |
|  |  |  |
|  |  |  |
| Mei-Huei Tang, PhD |  | Deacon Stephen T. Frezza, Ph.D., PSEM |
| Committee Member |  | Chair, Computer and Information Science Department |
|  |  |  |

Gannon University

Erie, Pennsylvania 16541

December 2020

Acknowledgements

< The student thanks significant stakeholders and supporters of the research effort. Consider thanking mentors, colleagues, individuals or institutions supporting the research, family. If a explicit permission was given to reproduce or to use material in the research, those entities are named and the relevance of their permission identified and credited. >

Table of Contents

[Acknowledgements ii](#_Toc8392258)

[Abstract v](#_Toc8392259)

[List of Figures vi](#_Toc8392260)

[List of Tables vii](#_Toc8392261)

[1. Introduction 1](#_Toc8392262)

[1.1 Overview 1](#_Toc8392263)

[1.2 Curriculum Scope 1](#_Toc8392264)

[1.3 Key Stakeholder Needs 1](#_Toc8392265)

[1.4 Product Perspective 1](#_Toc8392266)

[1.5 Product Position Statement 2](#_Toc8392267)

[1.6 Summary of Capabilities 2](#_Toc8392268)

[1.7 Alternatives and Competition 2](#_Toc8392269)

[1.8 Project Management Plan 3](#_Toc8392270)

[1.9 References 3](#_Toc8392271)

[2. Requirements Management 4](#_Toc8392272)

[2.1 Requirements Development Perspective 4](#_Toc8392273)

[2.2 Use Characteristics 4](#_Toc8392274)

[2.2.1 User Classes and Characteristics 4](#_Toc8392275)

[2.2.2 Use-Case Model Survey 4](#_Toc8392276)

[2.2.3 User Documentation 5](#_Toc8392277)

[2.3 Feature Attributes 5](#_Toc8392278)

[2.3.1 Name\_of\_Attribute-1 6](#_Toc8392279)

[2.3.2 Name\_of\_Attribute-2 6](#_Toc8392280)

[2.3.3 Name\_of\_Attribute-3 6](#_Toc8392281)

[2.4 Key System Features 6](#_Toc8392282)

[2.4.1 Key System Feature 1 6](#_Toc8392283)

[2.4.2 Key System Feature 2 (and so on) 7](#_Toc8392284)

[2.5 Key Design and Implementation Constraints 7](#_Toc8392285)

[2.5.1 Operating Environment 7](#_Toc8392286)

[2.6 Interface Requirements 7](#_Toc8392287)

[2.6.1 User Interfaces 7](#_Toc8392288)

[2.6.2 Hardware Interfaces 8](#_Toc8392289)

[2.6.3 Software Interfaces 8](#_Toc8392290)

[2.6.4 Communications Interfaces 8](#_Toc8392291)

[2.7 Nonfunctional Requirements 8](#_Toc8392292)

[2.7.1 Performance Requirements 8](#_Toc8392293)

[2.7.2 Security Requirements 9](#_Toc8392294)

[2.7.3 Software Quality Attributes 9](#_Toc8392295)

[2.7.4 Safety Requirements 9](#_Toc8392296)

[2.7.5 Other Requirements 9](#_Toc8392297)

[2.8 Assumptions and Dependencies 9](#_Toc8392298)

[3. Design 10](#_Toc8392299)

[3.1 Introduction 10](#_Toc8392301)

[3.1.1 Goals and Objectives of Design 10](#_Toc8392302)

[3.1.2 Statement of Software Scope 10](#_Toc8392303)

[3.1.3 Software Context 10](#_Toc8392304)

[3.1.4 Major Design Constraints 10](#_Toc8392305)

[3.2 Data Design 10](#_Toc8392306)

[3.2.1 Major Internal Software Data Structure 10](#_Toc8392307)

[3.2.2 Global Data Structure 10](#_Toc8392308)

[3.2.3 Temporary Data Structure 10](#_Toc8392309)

[3.2.4 Database Description 11](#_Toc8392310)

[3.3 Architectural and Component-Level Design 11](#_Toc8392311)

[3.3.1 Program Structure 11](#_Toc8392312)

[3.3.2 Key Software Components 11](#_Toc8392313)

[3.4 User Interface Design 12](#_Toc8392314)

[3.4.1 Description of the User Interface 12](#_Toc8392315)

[3.4.2 Interface Design Rules 12](#_Toc8392316)

[3.4.3 Components Available 12](#_Toc8392317)

[3.4.4 User Interface Design Description 12](#_Toc8392318)

[3.5 Restrictions, Limitations, and Constraints 12](#_Toc8392319)

[4. Verification and Validation 13](#_Toc8392320)

[4.1 Test items 13](#_Toc8392322)

[5. Conclusion 14](#_Toc8392323)

[6. Bibliography 15](#_Toc8392324)

[Appendix A: Glossary 16](#_Toc8392325)

[Appendix B: Use Case Analysis 17](#_Toc8392326)

[Appendix C: Analysis Models 17](#_Toc8392327)

[Appendix D: Design Models 17](#_Toc8392328)

[Appendix E: Testing Log and Summary Status 17](#_Toc8392329)

[Appendix F: Screen Captures 17](#_Toc8392330)

[Appendix G: Project File Repository Definitions 17](#_Toc8392331)

Abstract

< Give a brief summary (100-300 words) of the key information presented in the document. Do not simply describe what each segment of the writing offers, but condense the significant aspect of each section into the key facts and primary end states, decisions made, or conclusions reached in each section. Succinctly state the aspects of what situation was addressed by the research, why was the situation significant to its stakeholder, what solution design was followed, what final development arose, what major difficulties were addressed, how successful was the research in achieving its goals.>

# List of Figures

[Figure 1. Maybe a Context Diagram 1](#_Toc1377697)

[Figure 2. Maybe something else 5](#_Toc1377698)

[Figure 3. This is another Caption 5](#_Toc1377699)

< A listing of the figures or graphs displayed in the document are itemized here, including the page number of the appearance of the graphic. The following format pattern should be used in the listing and should be used as the graphic’s caption. The text of the caption should be boldface, all capital letters, the caption should appear under the graphic, and the numbering should be enabled through the /References/Insert-Table-of-Figures option of MS Word. “N” refers to the page number where the figure appears.

The List of Figures for this template has been generated above.>

# List of Tables

[Table 1: Position Statement 2](#_Toc8392414)

[TABLE 2: Benefits and Supporting Features 2](#_Toc8392415)

< A listing of the tables displayed in the document are itemized here, including the page number of the appearance of the table. The following format pattern should be used in the listing and should be duplicated as the table’s caption. The text of the caption should be boldface, all capital letters, the caption should appear under the graphic, and the numbering should be enabled through the /References/Insert-Table-of-Figures option of MS Word. “N” refers to the page number where the table appears.

The List of Tables for this template has been generated above.>

>

1. Introduction

Overview

This project involves the design and implementation of an AI-powered Intrusion Detection System (IDS). Its purpose is to monitor, analyze, and detect suspicious network activity in real time. The system will employ machine learning and deep learning to identify both known and emerging threats. By learning from network behavior and detecting anomalies, this intelligent IDS will provide superior accuracy, faster response times, and increased adaptability to evolving cyberattacks compared to conventional security systems. This work aims to contribute to smarter, more resilient cybersecurity infrastructure.

Curriculum Scope

This project is directly related to the principles of machine learning, data science, and cybersecurity taught in the Master of Science in Computer and Information Science program. The learning experiences gained in this research will involve applying theoretical knowledge of classification algorithms to a real-world problem, demonstrating a practical understanding of network security, and developing skills in data preprocessing and model evaluation.

Key Stakeholder Needs

The primary stakeholders are network security professionals who need a more effective way to detect sophisticated cyber threats. Their needs include a system that can:

* Detect new and previously unknown attacks (zero-day attacks).
* Reduce the number of false positives to avoid overwhelming security analysts.
* Provide real-time threat detection and alerting.
* Be adaptable to new types of attacks without requiring frequent manual updates.

Product Perspective

The intelligent IDS is a standalone security tool that integrates with existing network monitoring systems. It is not a complete network security suite but a specialized component focused on real-time threat detection. It will operate in the user’s network environment as a passive listener, analyzing traffic without disrupting normal operations.

Product Position Statement

For network security professionals who need a robust defense against evolving cyber threats, our machine learning-based IDS is an intelligent and adaptable solution that provides superior real-time threat detection. Unlike traditional signature-based systems that cannot detect new attacks, our product uses an innovative machine learning approach to identify and classify both known and unknown threats with high accuracy and minimal false positives.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| For... | Who... | The... | That... | Unlike... |
| Network security professionals | need a robust defense against evolving cyber threats | machine learning-based IDS | provides a superior real-time threat detection | traditional signature-based systems that cannot detect new attacks |

Summary of Capabilities

The primary capabilities of this IDS include:

* Real-time network traffic analysis.
* Anomaly detection for unknown threats.
* High accuracy in threat classification.
* Low false-positive rates.
* Automatic learning from new data.

Alternatives and Competition

Traditional signature-based IDSs are the main alternative. Their strength is their efficiency in detecting known threats, but their weakness is their inability to identify new attacks. Another alternative is heuristic-based IDSs, which can detect some new threats but often have high false-positive rates. Our machine learning-based approach provides a more balanced solution, offering both high accuracy and adaptability.

Project Management Plan

The project will follow an iterative development approach. The initial phase will focus on data collection and preprocessing. The next phase will involve training and evaluating the machine learning models. The final phase will be dedicated to integrating the models into a functional system and preparing the final documentation. The main difficulties faced were data acquisition and the computational resources required for training the models. These were addressed by utilizing publicly available datasets and cloud-based computing resources.

References

\*No references are available at this time.

1. Requirements Management
2. **Do not delete… Bug in document**

Requirements Development Perspective

The product’s core requirement is to accurately and efficiently identify cyber intrusions. This system must be able to integrate with a larger network monitoring infrastructure, receiving network data for analysis. The system's output will be a classification of network activity, which will be used to trigger alerts for security teams.

Use Characteristics

User Classes and Characteristics

The primary user class is network security administrators who have a strong technical background. They need to understand the system's output to make informed decisions and respond to threats.

Use-Case Model Survey

* Detect Intrusion: The system monitors network traffic and identifies malicious activity.
* Classify Threat: The system classifies the detected intrusion based on its type.
* Generate Alert: The system sends an alert to a security administrator when a threat is identified.

User Documentation

The product will include a user manual that details its functionality and provides guidance on interpreting its output.

Feature Attributes

Each feature will be evaluated based on the following attributes:

* **Priority:** High, Medium, Low
* **Status:** Planned, In Progress, Complete
* **Effort:** The estimated development time.

Key System Features

* Real-time Detection Engine: Analyzes network traffic as it occurs.
* Machine Learning Classifier: The core component for identifying threats.
* Alerting Mechanism: Notifies administrators of detected threats.

Key Design and Implementation Constraints

The system’s design is constrained by the need for computational efficiency to allow for real-time analysis. It must also be able to handle large volumes of network data and operate on standard hardware.

Interface Requirements

User Interfaces

The user interface will be a command-line interface or a simple web dashboard that displays the status of the system and a log of detected intrusions.

Hardware Interfaces

The system will run on a standard server or personal computer with sufficient processing power and memory.

Software Interfaces

The system will interface with network packet capture software (like Wireshark) and may integrate with existing SIEM (Security Information and Event Management) platforms.

Nonfunctional Requirements

Performance Requirements

The system must be able to process network traffic at a rate of at least 1 Gbps to support real-time detection.

Security Requirements

The system must not introduce new vulnerabilities to the network and must handle sensitive data with care. All data logging will be performed locally and encrypted.

Software Quality Attributes

The system will be reliable, accurate, and maintainable

Safety Requirements

The system is designed as a passive monitor and does not take any actions that could harm the network.

Other Requirements

* The system should support both IPv4 and IPv6 traffic.
* The system must be scalable to handle increasing network traffic volumes.

Assumptions and Dependencies

It is assumed that the network data will be provided in a format that the system can process. The project is dependent on the availability of public datasets for training and testing.

1. Design

Introduction

The system’s design follows a modular approach. It is composed of a data collection module, a preprocessing module, and the core machine learning model.

Data Design

The system will handle structured data from network packets, including features such as duration, protocol type, and service type. This data will be stored in a file or database for model training and analysis.

Architectural and Component-Level Design

The system architecture will consist of a **Data Acquisition Unit**, a **Preprocessing Engine**, and a **Classification Module**. The Data Acquisition Unit captures raw network packets. The Preprocessing Engine prepares the data for the model. The Classification Module, which contains the trained model, analyzes the data and provides a classification.

User Interface Design

The user interface will be a simple, text-based console that displays system status, alerts, and a log of network activity. A future iteration may include a web-based dashboard for more comprehensive visualization.

Restrictions, Limitations, and Constraints

The main limitation is that the model's accuracy is dependent on the quality and diversity of the training data. The system may also require a significant amount of memory to handle real-time traffic analysis.

1. Verification and Validation

The system will be tested using a separate, unseen dataset to evaluate its performance. The testing log will include detailed test cases and the results for each test. The success criteria will be based on achieving high accuracy (above 95%) and a low false-positive rate.

1. Conclusion

This research project was successful in achieving its primary objective of developing a machine learning-based IDS that is capable of detecting and classifying network intrusions. The models demonstrated high accuracy and proved to be more effective than traditional signature-based systems in identifying both known and unknown threats.

The final state of the work closely aligns with our original intent. We successfully implemented and evaluated the proposed models, and the system performed as expected. Future extensions of this project could include integrating more advanced deep learning models, such as recurrent neural networks, to analyze temporal patterns in network traffic.

The ethical elements of this product are a top priority. The system is designed to only monitor network traffic for security purposes and does not store any personally identifiable information. All data is handled with a focus on privacy and is only used for intrusion detection.

The benefits realized by this project are significant. It provides a more robust and adaptable cybersecurity tool that can protect organizations from a wider range of threats. The intelligent IDS can reduce the workload on security teams by providing more accurate alerts and helping them focus on genuine threats.

1. Bibliography

No bibliography is available at this time.

Appendix A: Glossary

* IDS: Intrusion Detection System
* NSL-KDD: A publicly available network intrusion detection dataset.
* CICIDS2017: A publicly available dataset for network intrusion detection.
* AI: Artificial Intelligence
* Machine Learning: A subfield of AI that enables systems to learn from data without being explicitly programmed.

Appendix B: Use Case Analysis

No use case analysis is available at this time.

Appendix C: Analysis Models

No analysis models are available at this time.

Appendix D: Design Models

No design models are available at this time.

Appendix E: Testing Log and Summary Status

No testing log is available at this time.

Appendix F: Screen Captures

No screen captures are available at this time.

Appendix G: Project File Repository Definitions

No project file repository definitions are available at this time.