**Intrusion Detection System**

*A thesis submitted in partial fulfillment of the requirement for the award of the degree*

*Of*

**Bachelor of Computer Application**

***in***

***Faculty of Engineering and Technology***



***Submitted by***

Chinmoy Das

Roll no: ADTU/2022-25/BCASP/014

Abhishek Dutta

Roll no: ADTU/2022-25/BCASP/040

*Under the guidance of*

***Dr. Mala Dutta***

Associate Professor

Faculty of Computer Technology

**Assam down town University**

**Guwahati-26, Assam**

Session: January-June, 2025

**CONTENTS**

|  |  |
| --- | --- |
| * *Certificate of Approval* | i |
| * *Certificate from Guide* | ii |
| * *Certificate from External Examiner* | iii |
| * *Declaration* | iv |
| * *Acknowledgement* | v |
| * *Abstract* | vi |
| * *List of Figures* | vii |

**Chapter Page No**

|  |  |
| --- | --- |
| 1. **Introduction** |  |
| * 1. Overview of the Project | 1-4 |
| * 1. Motivation |  |
| * 1. Scope and Objective |  |
| * 1. Existing System |  |
| * 1. Problem Definition |  |
| * 1. Proposed System |  |
| 1. **Project Analysis** |  |
| * 1. Project Requirement Analysis | 5 |
| * 1. Gantt Chart | 6 |
| * 1. Advantage & Disadvantage | 7-8 |
| * 1. Project Life Cycle   2. Project Feasibility | 9 |
| 1. **Project Design** |  |
| * 1. System Architecture | 10 |
| * 1. Data Flow Diagram(DFD) | 10-12 |
| * 1. NoSQL Diagram | 13 |
| * 1. Use Case Diagram | 14 |
| * 1. Sequence Diagram | 15 |
| * 1. Schema Diagram | 16 |
| 1. **Project Implementation** |  |
| * 1. Description of the Software Used | 17-18 |
| * 1. User Interface | 18-21 |
| 1. **Testing** |  |
| * 1. Types of Testing | 21 |
| * 1. Test Cases | 22 |
| 1. **Conclusion and Future Scope** |  |
| * 1. Conclusion | 23 |
| *References* | 23 |

****

***Faculty of Computer Technology***

***Assam down town University***

Gandhi Nagar, Panikhaiti, Guwahati- 781026, Assam

**CERTIFICATE OF APPROVAL**

This is to certify that the project report entitled ***“Intrusion Detection System”*** submitted by Chinmoy Das bearing Roll No. ADTU/2022-25/BCASP/014 and Abhishek Duttabearing Roll No. ADTU/2022-25/BCASP/040, are hereby accorded our approval as a study carried out and presented in a manner required for acceptance in partial fulfilment for the award of the degree of ***Bachelor of Computer Application*** under Assam Down Town University for approval does not necessary endorse or accept every statement made opinion expressed or conclusion drawn as recorded in the report. It only signifies the acceptance of the project report for a purpose which is submitted.

**Date:**

**Place: Guwahati**

****

***Faculty of Computer Technology***

***Assam Down Town University***

Gandhi Nagar, Panikhaiti, Guwahati- 781026, Assam

**CERTIFICATE FROM GUIDE**

This is to certify that the project report entitled ***“Intrusion Detection System”*** submitted by **Chinmoy Das** bearing Roll No. ADTU/2022-25/BCASP/014 and **Abhishek Dutta** bearing Roll No. ADTU/2022-25/BCASP/040 towards the partial fulfilment of the requirements for the award of the degree of ***Bachelor of Computer Application*** under Assam down town University is a bonafide research work carried out by them under my supervision and guidance. This work has not been submitted previously for any other degree of this or any other University.

I recommend that the thesis may be placed before the examiners for consideration of award of the degree of this University.

|  |  |
| --- | --- |
| **Date:**  **Place: Guwahati** | **Dr. Mala Dutta**  Associate Dean, Faculty of Computer Technology  Assam down town University  Guwahati |
|  |  |

****

***Faculty of Computer Technology***

***Assam down town University***

Gandhi Nagar, Panikhaiti, Guwahati- 781026, Assam

**CERTIFICATE FROM EXTERNAL EXAMINER**

This is to certify that the project report entitled ***“Intrusion Detection System”*** submitted by Chinmoy Das bearing Roll No. ADTU/2022-25/BCASP/014 and Abhishek Dutta bearing Roll No. ADTU/2022-25/BCASP/040 towards the partial fulfilment of the requirements for the award of the degree of ***Bachelor of Computer Application*** under Assam down town University is a bonafide research work carried out by him under the supervision and guidance of ***Dr. Mala Dutta***, Associate Dean, Department of Computer Technology, Assam down town University, Guwahati has been examined by me and found to be satisfactory.

I recommend the thesis for consideration for the award of the degree of ***Bachelor of Computer Application*** under Assam down town University.

**Date:**

**Place: Guwahati**

**DECLARATION**

We, Chinmoy Das bearing Roll No. ADTU/2022-25/BCASP/014 and Abhishek Dutta bearing Roll No. ADTU/2022-25/BCASP/040 hereby declare that the thesis entitled ***“INTRUSION DETECTION SYSTEM”*** is an original work carried out in the Department of Computer Technology, Assam down town University, Guwahati with exception of guidance and suggestions received from my supervisor, ***Dr. Gunikhan Sonowal***, Assistant Professor, Department of Computer Technology, Assam down town University, Guwahati. The data and the findings discussed in the thesis are the outcome of my research work. This thesis is being submitted to Assam down town University for the degree of ***Bachelor of Computer Application”.***

**ACKNOWLEDGMENT**

We would like to extend our heartfelt appreciation to everyone who contributed to the successful completion of this project. Our sincere thanks go to our project team members for their dedication and collaboration throughout the project. Each member played a significant role in shaping the outcome. Special thanks to our supervisor, Dr. Mala Dutta, for her guidance and valuable feedback, which enriched our work. Lastly, we want to thank our friends for their patience and encouragement during this project. Their believe helped us to stay motivated and to persevere through difficult times.

**ABSTRACT**

In today’s highly interconnected digital environment, the exponential growth of networked systems and internet-connected devices has significantly increased the risk of cyber threats. Organizations, governments, and individuals are constantly exposed to a wide spectrum of malicious activities ranging from data breaches and identity theft to large-scale Distributed Denial of Service (DDoS) attacks. These threats not only compromise sensitive data but can also cripple critical infrastructure, resulting in substantial financial and reputational losses. While traditional defense mechanisms such as firewalls and antivirus programs form the first line of defense, they often fall short in identifying sophisticated or rapidly evolving threats. This shortfall necessitates the implementation of more dynamic and intelligent solutions—chief among them, Intrusion Detection Systems (IDS).

This project presents the design and implementation of a custom-built Intrusion Detection System (IDS) aimed at detecting and mitigating malicious activities through continuous network traffic analysis. The system is engineered to detect a broad array of cyber threats, including unauthorized access attempts, malware infections, port scanning, and suspicious behaviors indicative of zero-day attacks. Emphasis is placed on utilizing both **signature-based** techniques (to identify known attack patterns) and **anomaly-based** techniques (to detect previously unseen behaviors). This dual approach ensures a more robust detection mechanism capable of adapting to dynamic threat landscapes.

One of the distinguishing features of the proposed system is its **user-friendliness and flexibility**, achieved by integrating Python into the system's architecture. Python’s intuitive syntax and extensive libraries allow us to build interfaces and workflows that simplify rule management, logging, and real-time alert visualization. In contrast to traditional systems like Snort, which can be daunting for beginners due to their complex configurations, our approach makes deployment and maintenance accessible to a broader audience, including small organizations and educational institutions.

Furthermore, to improve the effectiveness of anomaly detection, the system is designed to be extensible with **machine learning algorithms** capable of identifying subtle behavioral deviations across network traffic. These include clustering techniques for traffic profiling and supervised models trained on labeled datasets to distinguish benign from malicious packets. Such integration not only enhances the detection of novel attacks but also reduces false positives by enabling context-aware classification.

For real-time packet analysis and low-level networking, we utilize tools and libraries such as **libpcap**, **DAq**, and **dnet**, alongside **Snort** as the primary detection engine. The Python layer acts as a control interface and analytics dashboard, handling data visualization, system status, and logging for security analysts. Log data, alerts, and detection metrics are captured and stored for auditing and forensics, and alert thresholds can be tuned based on user-defined criteria.

The system is also engineered with **scalability and performance** in mind. By leveraging multi-threading and CPU affinity libraries such as **hwloc**, it ensures efficient resource utilization even under high network loads. Additionally, the use of **PCRE2** for efficient regular expression matching, **OpenSSL** for cryptographic functions, and **LuaJIT** for lightweight configuration scripting ensures high performance without sacrificing detection capabilities.

In terms of deployment, the system supports both centralized and distributed architectures. In smaller environments, it can run as a standalone IDS on a gateway or a monitoring device. In enterprise environments, it can be scaled across multiple sensors and connected to a central logging server, facilitating distributed detection and response.

Overall, this project provides a lightweight yet powerful IDS solution that balances usability, performance, and detection effectiveness. By enhancing the traditional Snort-based engine with modern usability features and extending it through Python scripting and machine learning, the system not only protects against known threats but is also well-equipped to handle emerging ones. The result is a customizable, scalable, and real-time IDS that caters to the diverse and evolving security needs of modern networks.

**List of Figures**

|  |  |
| --- | --- |
| **Sl no.** | **Name of the figure/chart** |
| **1** | **Gantt Chart** |
| **2** | **System Architecture** |
| **3** | **Data Flow Diagram** |
| **4** | **Context diagram** |
| **5** | **Nosql diagram** |
| **6** | **Use case Diagram** |
| **7** | **Sequence diagram** |
| **8** | **Schema Diagram** |
| **9** | **Wireframes/Ui** |

**1. INTRODUCTION**

**1.1 Overview of the project**

This project aims to develop an **Intrusion Detection System (IDS)** that monitors network or system activity to detect and alert against suspicious or unauthorized behavior. As cyber threats continue to grow, an IDS adds an important layer of security by identifying potential attacks in real time.

This project involves the development of an Intrusion Detection System (IDS) that uses Nmap to scan and monitor network activity for known vulnerabilities and threats. By relying on signature-based detection, the system can identify suspicious patterns and behaviors that match a predefined database of attack signatures. The focus is on detecting known intrusions efficiently and alerting users to take necessary action. It is designed for small to medium-scale networks where basic security monitoring is required.

**1.2 Motivation**

With the rise in cyber threats targeting network vulnerabilities, there is a strong need for simple and effective tools that help monitor and protect systems. Many organizations, especially smaller ones, lack advanced security infrastructure. This project is motivated by the need to provide a basic yet reliable Intrusion Detection System that uses Nmap to identify known threats by scanning for open ports, services, and known vulnerabilities. By focusing on signature-based detection, the system offers a practical solution for early threat detection and strengthens basic network security.

**1.3 Scope and Objective**

This project focuses on developing a lightweight and effective Intrusion Detection System (IDS) capable of monitoring network or system activity to detect potential security breaches.

The main objective of this project is to develop an Intrusion Detection System (IDS) that can accurately detect and alert on potential security threats in real time. It uses Nmap to scan network activity and detect known types of attacks. It focuses on signature-based detection, making it effective for recognizing predefined threats and alerting administrators for timely response.

**1.4 Existing system**

Several intrusion detection systems (IDS) are currently available, with Snort, Suricata, and OSSEC being among the most widely used. These systems provide robust features such as real-time traffic analysis, signature and anomaly-based threat detection, and integration with external threat intelligence feeds. Snort, in particular, is a highly reliable open-source IDS known for its flexible rule-based detection engine and wide community support.

However, while Snort offers powerful detection capabilities, its default configuration can be complex for users with limited experience in network security. It typically requires manual rule writing, command-line configuration, and integration with other tools to achieve full functionality. Moreover, Snort in its default form may not be tailored for specific use cases like focused detection of vulnerabilities scanned by tools such as Nmap.

Therefore, this project builds on Snort but customizes and optimizes it for more targeted detection of network scans and known vulnerabilities, offering a lightweight and user-focused IDS environment. By leveraging Snort’s engine while simplifying its deployment and tuning it for specific scanning behavior, the project addresses limitations of general-purpose configurations and makes IDS more accessible for smaller-scale or educational use cases.

**1.5 Problem Definition**

In today’s increasingly connected world, ensuring robust network security is essential. Cyberattacks such as data breaches, unauthorized access, and vulnerability exploitation threaten the confidentiality, integrity, and availability of digital systems. While traditional defenses like firewalls and antivirus software provide basic protection, they often fail to detect sophisticated or targeted intrusions.

Intrusion Detection Systems (IDS) help address this gap by analyzing network traffic for suspicious patterns. However, many existing solutions are limited by complex configuration, high false positive rates, and resource-intensive performance. They are often too generalized or cumbersome for targeted use cases like detecting reconnaissance activity from tools such as Nmap.

This project aims to enhance and tune Snort, an open-source IDS, to deliver precise, rule-based detection for specific network threats. By focusing on identifying known vulnerabilities and scanning behaviors, the system provides real-time alerts with improved usability and adaptability—making it suitable for both educational purposes and lightweight security setups.

**1.6 Proposed System**

The proposed Intrusion Detection System (IDS) integrates the powerful detection capabilities of **Snort** with an intuitive, Python-based graphical interface to improve both usability and efficiency. While Snort performs deep packet inspection and rule-based detection of threats, the Python layer simplifies system interaction by automating configuration, monitoring, and alert visualization.

This combination makes the IDS more accessible for beginners and non-expert users who may find traditional command-line IDS tools complex. The system continuously analyzes network traffic in real-time, effectively identifying known vulnerabilities and common scanning behaviors—such as those generated by tools like **Nmap**. It also minimizes false positives and ensures stable performance across networks of varying size, making it suitable for both educational and operational environments.

**2. PROJECT ANALYSIS**

**2.1 Project Requirement Analysis**

The Intrusion Detection System (IDS) is designed to enhance overall network security by fulfilling both functional and non-functional requirements. On the functional side, the system must be capable of continuously monitoring network traffic in real time, identifying potential threats through both signature-based detection (which matches known attack patterns) and anomaly-based detection (which flags unusual behaviours). It must also generate immediate alerts upon detection, log all relevant incident data for auditing or forensic analysis, and support flexible rule configuration to adapt to different network environments and threat models.

In terms of non-functional requirements, the IDS must maintain high performance and low latency to ensure it does not disrupt network operations, even under heavy traffic. Scalability is key—it should operate efficiently in both small-scale and enterprise-level networks. The system must also exhibit reliability, remaining active and accurate over long periods with minimal downtime. Usability is another critical aspect; administrators should be able to manage and configure the system without extensive training. Finally, the IDS must be secure against tampering, and its maintenance—including updates to detection signatures and anomaly profiles—should be straightforward to ensure continued effectiveness against evolving cyber threats.

**2.2 Gantt Chart**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **March** | **April** | **May** |
| Information Gathering |  |  |  |
| Analysis |  |  |  |
| Design |  |  |  |
| Coding |  |  |  |
| Testing |  |  |  |
| Analysis |  |  |  |

**2.3 Advantage and Disadvantage**

**2.31 Advantages**

1. **Real-Time Threat Detection**  
   The IDS provides continuous surveillance of network traffic, enabling **immediate identification and response** to suspicious or malicious activities. This reduces the window of vulnerability.
2. **Automated Threat Identification**  
   Through a combination of **signature-based** and **anomaly-based** detection techniques, the system is capable of recognizing **known malware patterns** as well as **unusual behaviors** that may indicate zero-day attacks or insider threats.
3. **Enhanced Security Oversight**  
   The system **generates detailed logs and alerts**, providing actionable insights to system administrators. This enhances the organization's ability to **audit, track, and mitigate** security events effectively.
4. **Scalability**  
   Designed to adapt to varying network sizes, the IDS can **scale horizontally** to support increased data throughput and traffic volumes, making it suitable for **enterprise-grade as well as small-scale deployments**.
5. **Customizability and Flexibility**  
   Detection rules, thresholds, and behaviors can be **custom-configured** by administrators, allowing the IDS to **align with specific security policies and network architectures**.

**2.32 Disadvantages**

1. **False Positives and Alert Fatigue**  
   Anomaly-based detection may **flag legitimate traffic as malicious**, leading to **false positives**. This can overwhelm administrators and dilute focus from actual threats.
2. **High Resource Consumption**  
   Real-time traffic inspection and deep packet analysis demand **significant CPU and memory usage**, which could **degrade performance** on lower-end systems if not properly optimized.
3. **Complex Setup and Configuration**  
   Effective deployment requires **technical expertise** in networking, threat modeling, and system integration. Misconfiguration could either reduce detection efficacy or increase false alerts.
4. **Ongoing Maintenance and Updates**  
   To remain effective, the IDS must be **routinely updated** with the latest threat signatures and tuned for evolving attack vectors. This introduces **long-term maintenance overhead**.

**2.4 Project Lifecycle**

The development of the Intrusion Detection System (IDS) adhered to a structured project lifecycle to ensure a smooth transition from concept to deployment and maintenance. The key phases of this lifecycle are outlined below:

**i) Initiation:**  
The project began with identifying the need for a robust IDS solution to enhance network security. This phase involved setting clear objectives, analyzing the existing security landscape, and gathering initial requirements from stakeholders.

**ii) Planning:**  
In this phase, the project scope was defined in detail. A comprehensive plan was formulated covering task allocation, resource requirements, timelines, and risk management strategies. The system architecture and technology stack were also designed.

**iii) Development:**  
Core functionalities were implemented, including network traffic monitoring, packet analysis, rule-based detection using Snort, and Python-based automation for log handling and alerts. Custom rules were also developed to detect specific intrusion patterns.

**iv) Testing & Quality Assurance:**  
Extensive testing was conducted to ensure the system's stability, detection accuracy, performance, and resilience against known threats. Test cases simulated various attack vectors, including scans with tools like Nmap, to validate real-time detection.

**v) Deployment:**  
The IDS was deployed within the target network environment. Network interfaces were configured appropriately, and integration with existing security systems was ensured. The user interface was finalized for streamlined control (e.g., Start/Stop functions).

**vi) Monitoring & Maintenance:**  
Post-deployment, the system was continuously monitored for performance and reliability. Updates to threat signatures, rule refinement, and routine maintenance tasks were carried out to keep the system effective against emerging threats.

**vii) Evaluation & Optimization:**  
The IDS was periodically evaluated based on feedback and real-world performance. Analytical reviews led to optimizations in detection logic, user interaction, and system efficiency to reduce false positives and improve accuracy.

**viii) Closure:**  
The final phase involved wrapping up project documentation, compiling performance reports, and transitioning the IDS to operational use. Knowledge transfer and long-term support plans were also outlined.

**2.5 Project feasibility**

The feasibility analysis of the Intrusion Detection System (IDS) project assesses its practicality and potential for successful implementation by examining several critical dimensions:

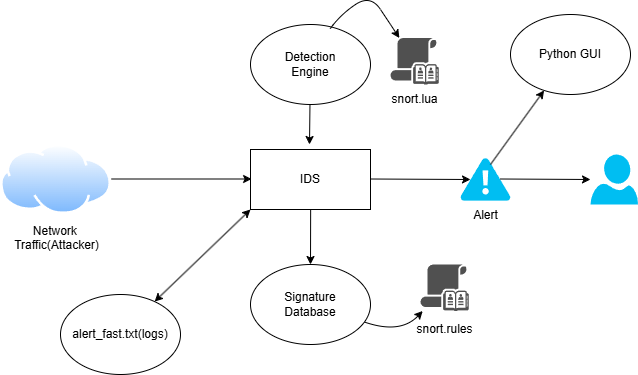
**I. Technical Feasibility:**  
This dimension evaluates the capability to develop and deploy the IDS using the selected technologies, including machine learning algorithms, real-time monitoring tools, and network analysis frameworks. It further considers the system’s compatibility with the existing IT infrastructure and its scalability to accommodate future network expansion.

**II. Market Feasibility:**  
Market feasibility assesses the demand for the IDS within the targeted user base, primarily enterprises and organizations seeking enhanced network security solutions. It includes an analysis of the competitive landscape and identifies possible challenges related to user adoption and market penetration.

**III. Financial Feasibility:**  
This aspect involves estimating the total costs related to system development, implementation, ongoing maintenance, and updates. These costs are compared against the expected financial benefits, such as reduction in cybersecurity risks and potential savings from averting data breaches, to determine the economic viability of the project.

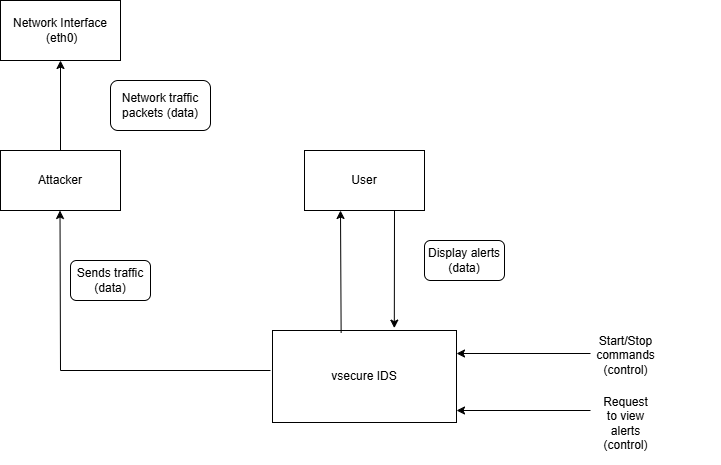
**IV. Operational Feasibility:**  
Operational feasibility examines the practicality of integrating the IDS into the current network environment without interrupting ongoing business processes. It also evaluates the resource requirements for deployment, monitoring, and maintenance, including necessary personnel and training.

**3. PROJECT DESIGN**

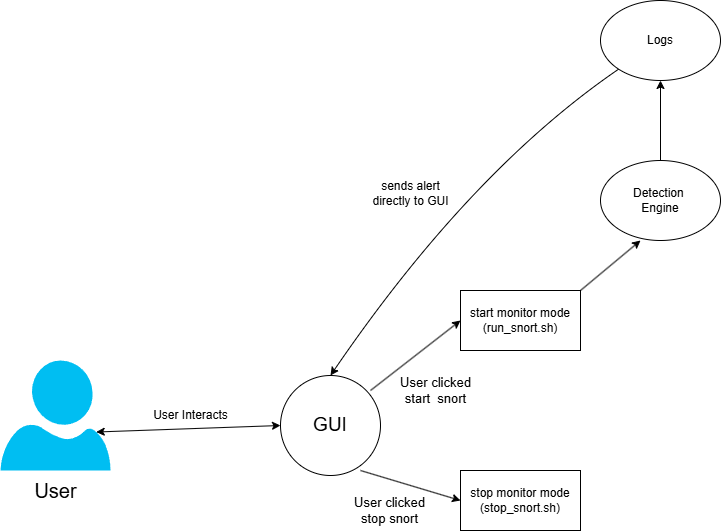
**3.1 System Architecture**

**3.2 Data Flow Diagram**

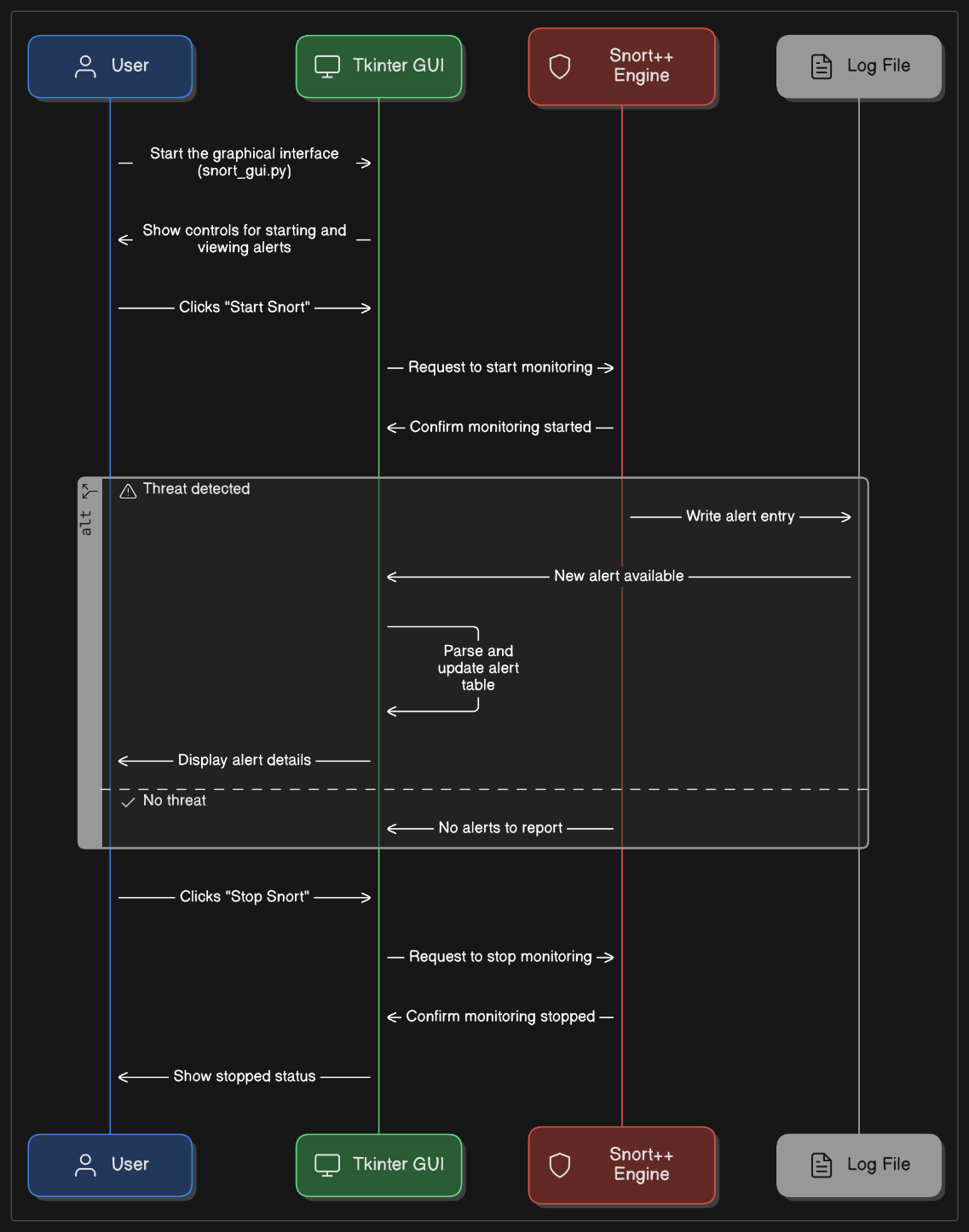
**3.2.1 Context Diagram**

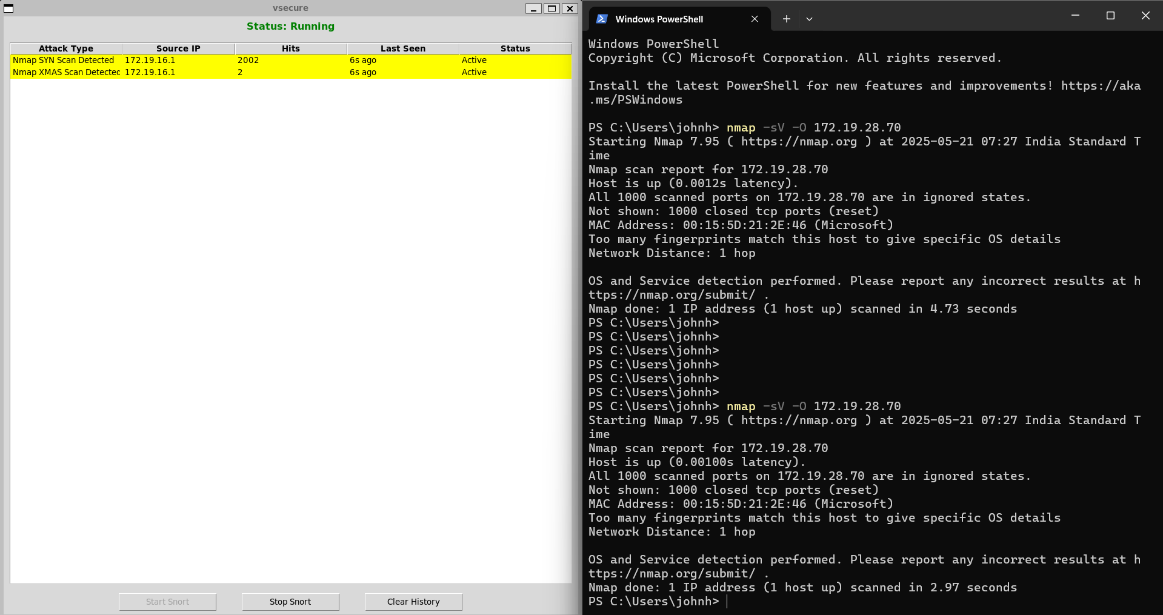
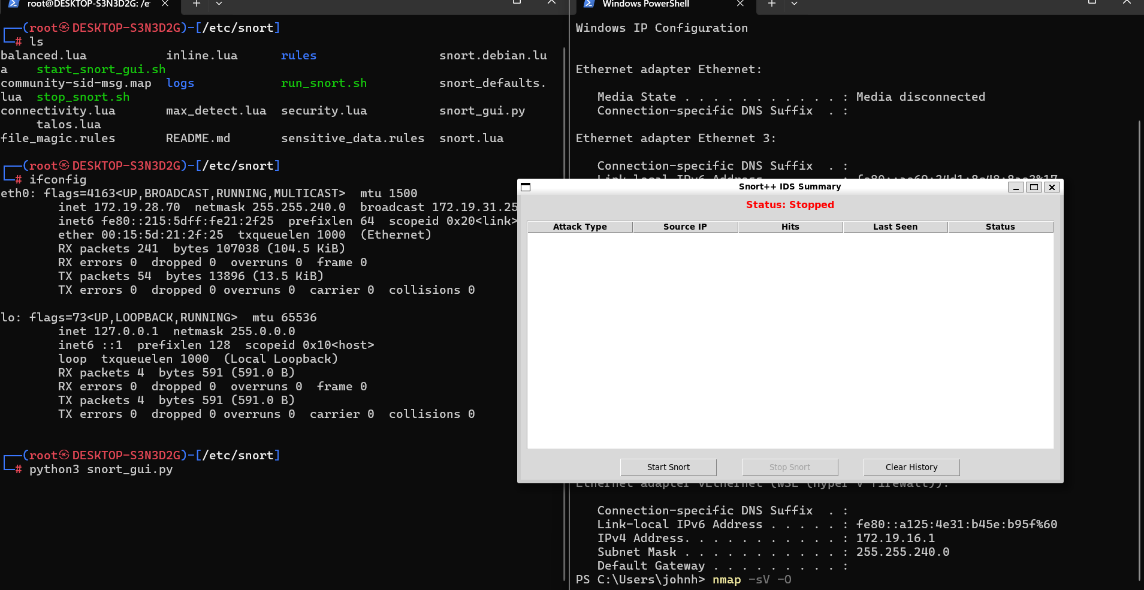


**3.3 Use case Diagram**



**3.5 Sequence Diagram**





**4. PROJECT IMPLEMENTATION**

**4.1 Description of the Software Used**

For the implementation of the Intrusion Detection System (IDS) focusing on detecting Nmap scans and ping attempts, we used the following software components and tools:

**i) Snort IDS:**

* An open-source network intrusion detection and prevention system (NIDS/NIPS).
* Capable of real-time traffic analysis and packet logging on IP networks.
* Utilizes a rule-based language to define patterns and signatures of known threats and suspicious activities.
* Our custom IDS rules were developed to detect specific Nmap scanning techniques (NULL, FIN, XMAS, SYN scans) and ICMP ping traffic.

**ii) LibDAQ (Data Acquisition Library):**

* A critical library used by Snort for packet acquisition and network traffic capturing.
* Provides abstraction for capturing packets from network interfaces.
* Ensures efficient and reliable packet collection from different network types, supporting the IDS’s core packet processing.

**iii) CMake:**

* A cross-platform open-source tool for managing the build process of software using compiler-independent configuration files.
* Used to configure and build Snort and its dependencies on the development system.
* Ensures reproducible and manageable compilation and installation processes.

**iv) Nmap:**

* A powerful open-source network scanning tool used to test the effectiveness of IDS rules.
* Generates various types of network probes such as SYN, NULL, FIN, XMAS scans, and OS fingerprinting to simulate attacker behavior.
* Provides a practical way to validate and fine-tune IDS detection capabilities against known reconnaissance techniques.

**v) Linux Operating System:**

* The IDS and testing environment is based on Linux (Ubuntu/Debian) for stability and network tool compatibility.
* Provides native support for networking, packet capturing tools, and open-source IDS software.

**5. Testing / Result Analysis**

**5.1 Types of Testing**

To ensure the Snort IDS rules work effectively for detecting various Nmap scan attempts and ping, the following types of testing should be performed:

i) **Signature Testing**:

* Test each IDS rule against the specific Nmap scan or traffic pattern it is designed to detect.
* Ensures that the rule triggers alerts only when the relevant suspicious traffic is present.

ii) **False Positive Testing**:

* Validate that normal, legitimate network traffic does not trigger alerts falsely.
* Helps in tuning the rules to reduce noise.

iii) **Cross-Network Testing**:

* Simulate attacks from different networks or hosts to ensure detection is network independent.
* Confirms IDS rules can detect remote scans and pings accurately.

iv) **Performance Testing**:

* Assess the resource usage of Snort when processing network traffic with these rules enabled.
* Ensures that the IDS does not degrade network performance significantly.

**5.3 Test Cases**

i) **Signature Test - Nmap OS Fingerprinting:**

* **Test Case:** Run an Nmap OS fingerprint scan (e.g., nmap -O <target\_IP>) from an attacker machine.
* **Expected Result:** Snort triggers an alert with the message "Nmap OS Fingerprint Attempt."

ii) **Signature Test - Nmap NULL Scan:**

* **Test Case:** Run an Nmap NULL scan (e.g., nmap -sN <target\_IP>) from a remote host.
* **Expected Result:** Snort triggers an alert with the message "Nmap NULL Scan Detected."

iii) **Signature Test - Nmap FIN Scan:**

* **Test Case:** Run an Nmap FIN scan (e.g., nmap -sF <target\_IP>).
* **Expected Result:** Snort triggers an alert with the message "Nmap FIN Scan Detected"

iv) **Signature Test - Nmap XMAS Scan:**

* **Test Case:** Run an Nmap XMAS scan (e.g., nmap -sX <target\_IP>).
* **Expected Result:** Snort triggers an alert with the message "Nmap XMAS Scan Detected."

v) **Signature Test - Nmap SYN Scan:**

* **Test Case:** Run an Nmap SYN scan (e.g., nmap -sS <target\_IP>).
* **Expected Result:** Snort triggers an alert with the message "Nmap SYN Scan Detected.”

vi) **False Positive Test - Normal Traffic:**

* **Test Case:** Generate normal TCP, UDP, and ICMP traffic without scanning or pinging.
* **Expected Result:** No Snort alerts are generated.

vii) **False Positive Test - Normal Traffic:**

* **Test Case:** Generate normal TCP, UDP, and ICMP traffic without scanning or pinging.
* **Expected Result:** No Snort alerts are generated.

viii) **Cross-Network Test - Remote Scan Detection:**

* **Test Case:** Perform each of the above Nmap scans and ping from a host on a different network than the Snort sensor.
* **Expected Result:** Snort detects and alerts on all scan and ping attempts, confirming remote detection capability.

**6.1 Conclusion**

This Software Requirements Specification (SRS) document outlines the detailed requirements for the design and development of an Intrusion Detection System (IDS). The primary goal of the system is to monitor network traffic, identify suspicious activities, and alert administrators to potential security threats in real time.

Through this document, we have clearly defined the functional and non-functional requirements, detailed the system features, and identified the stakeholder expectations. The IDS is designed to be scalable, efficient, and capable of adapting to evolving security threats. Emphasis has been placed on performance, usability, and security to ensure that the system meets organizational standards and industry best practices.

**6.2 References:**

<https://www.youtube.com/watch?v=nIrpSxMUAsk>

<https://commercejs.com/>

<https://dashboard.stripe.com/>

<https://www.plantuml.com/plantuml/uml/SyfFKj2rKt3CoKnELR1Io4ZDoSa70000>

<https://aws.amazon.com/lambda/>