Final Year Project Report to obtain Bachelor's Degree in Fundamental Studies in Mathematics and Computer Science

Design and Implementation of an Intrusion Detection and Prevention Monitoring System

ABOUHANE Zahra
BAZGOUR Yassine

Supervised by

Prof. OUAISSA Mariya

Held On

June 29, 2024

Before

Prof. OUAISSA Mariya
Prof. ELBACHARI Essaid

Acknowledgements

First of all we would like to express our deepest gratitude to Allah, the Almighty, for granting us the strength, patience, and perseverance to complete this project successfully.

We are immensely grateful to our supervisor, Pr. OUAISSA Maria, for their continuous support, insightful advice, and invaluable feedback throughout this project. Their guidance has been instrumental in shaping the direction and outcome of our work.

We would also like to extend our sincere thanks to our examiner, Pr. Elbachari Essaid, for their willingness to evaluate our project. We appreciate their time, effort, and the valuable insights they will provide during the evaluation process.

We would like to extend our sincere thanks to the faculty members of SEMLALIA for their dedication and commitment to excellence in teaching. The knowledge and skills we have gained from them have been crucial to the successful completion of this project.

We are deeply thankful to our families and friends for their patience, encouragement, and understanding during the course of our studies and this project. Their support has been a constant source of strength for us.

We also appreciate the resources and facilities provided by the Faculty of Semlalia.

Thank you all for your support and confidence in our abilities.

Abstract

This project is centered on the study and implementation of an Intrusion Detection and Prevention System (IDPS) to enhance network security. The primary objective was to work with a robust IDPS capable of identifying and mitigating cyber threats in real-time. Our approach involved an in-depth analysis of various cybersecurity threats and the defense mechanisms necessary to counter them, emphasizing the importance of a secure network infrastructure.

We began by researching the foundational aspects of cybersecurity, including common attack vectors and defense strategies. This was followed by the design and implementation of an IDPS using a combination of open-source tools and custom-developed scripts. The system was rigorously tested in a simulated environment to evaluate its effectiveness in detecting and preventing various types of intrusions, such as Distributed Denial of Service (DDoS) attacks, malware infections, and unauthorized access attempts.

The project also explored the implementation of a Demilitarized Zone (DMZ) to provide an additional layer of security for sensitive data and services. Throughout the development process, we faced challenges related to system integration, configuration of detection rules, and fine-tuning the response mechanisms. However, through persistent effort and iterative testing, we were able to create a functional and efficient IDPS.

Our results demonstrate the system's capability to detect and respond to threats in real-time, significantly enhancing network security. The successful completion of this project lays the groundwork for further advancements in intrusion detection and prevention, with the ultimate aim of safeguarding organizational networks against evolving cyber-threats.

ملخص

هذا المشروع يركز على دراسة وتنفيذ نظام اكتشاف ومنع التسلل لتعزيز أمان الشبكات. الهدف الأساسي هو العمل مع نظام قوي يمكنه تحديد وصد التهديدات الإلكترونية في الوقت الفعلي. بدأنا بالبحث في الجوانب الأساسية للأمن السيبراني، بما في ذلك أساليب الهجوم الشائعة واستراتيجيات الدفاع اللازمة لمواجهتها، مع التأكيد على أهمية بنية تحتية آمنة للشبكة. تضمن عملنا تصميم وتنفيذ النظام باستخدام مزيج من الأدوات مفتوحة المصدر والبرامج النصية المطورة خصيصًا. تم اختبار النظام بدقة في بيئة محاكاة لتقييم فعاليته في اكتشاف ومنع أنواع مختلفة من التسللات، مثل هجمات الحرمان من الخدمة الموزعة، وإصابات البرمجيات الخبيثة، ومحاولات الوصول غير المصرح بها.

كما استكشف المشروع تنفيذ منطقة منزوعة السلاح لتوفير طبقة إضافية من الأمان للبيانات والخدمات الحساسة. وخلال عملية التطوير، واجهنا تحديات تتعلق بتكامل النظام وتكوين قواعد الاكتشاف وضبط آليات الاستجابة. لكن من خلال الجهد المستمر والاختبار المتكرر، تمكنا من إنشاء نظام فعال وقابل للتشغيل.

تظهر النتائج قدرة النظام على اكتشاف والرد على التهديدات في الوقت الفعلي، مما يعزز بشكل كبير أمان الشبكة. إن الانتهاء الناجح من هذا المشروع يضع الأساس لمزيد من التطورات في اكتشاف ومنع التسلل، بهدف نهائي هو حماية شبكات المؤسسات من التهديدات السيبرانية المتطورة.

Résumé

Ce projet est centré sur l'étude et la mise en œuvre d'un Système de Détection et de Prévention d'Intrusions (SDPI) pour améliorer la sécurité réseau. L'objectif principal était de travailler avec un SDPI robuste capable d'identifier et de contrer les menaces informatiques en temps réel. Notre approche a impliqué une analyse approfondie de diverses menaces en cybersécurité et des mécanismes de défense nécessaires pour les contrer, en mettant l'accent sur l'importance d'une infrastructure réseau sécurisée.

Nous avons commencé par rechercher les aspects fondamentaux de la cybersécurité, y compris les vecteurs d'attaque courants et les stratégies de défense. Cela a été suivi par la conception et la mise en œuvre d'un SDPI utilisant une combinaison d'outils open-source et de scripts développés sur mesure. Le système a été rigoureusement testé dans un environnement simulé pour évaluer son efficacité à détecter et à prévenir divers types d'intrusions, tels que les attaques par Déni de Service Distribué (DDoS), les infections par logiciels malveillants et les tentatives d'accès non autorisé.

Le projet a également exploré la mise en œuvre d'une Zone Démilitarisée (DMZ) pour fournir une couche de sécurité supplémentaire pour les données et services sensibles. Tout au long du processus de développement, nous avons rencontré des défis liés à l'intégration du système, à la configuration des règles de détection et à l'affinage des mécanismes de réponse. Cependant, grâce à un effort persistant et des tests itératifs, nous avons pu créer un SDPI fonctionnel et efficace.

Nos résultats démontrent la capacité du système à détecter et à répondre aux menaces en temps réel, améliorant ainsi significativement la sécurité du réseau. La réussite de ce projet jette les bases pour de nouvelles avancées en détection et prévention des intrusions, avec pour objectif ultime de protéger les réseaux organisationnels contre les menaces cybernétiques évolutives.

Contents

List of Figures

List of Acronyms

2FA Tow Factor Authentication.

ACL Access Control List.

AI Artificial Intelligence.

ASCII American Standard Code for Information Interchange.

CGI Common Gateway Interface.

CIA Confidentiality Integrity Availability.

DDOS Distributed Denial of Service.

DMZ Demilitarized Zone.

DOS Denial of Service.

FAQ Frequently Asked Questions.

FTP File Transfer Protocol.

HIDS Host Intrusion Prevention System.

ICMP Internet Control Message Protocol.

ID Identification.

IDPS Intrusion Detection and Prevention System.

IDS Intrusion Detection System.

IP internet protocol.

IPS Intrusion Prevention System.

IPV6 Internet Protocol version 6.

IPVAR IP variable.

JS Javascript.

LOIC Low Orbit Ice Common.

MITM Man In The Middle.

NIDS Network Intrusion Detection System.

OS Operating System.

SID Signature ID.

SIEM Security Information and Event Management.

SMB Server Message Block.

SQL Structured Query Language.

SSH Secure Shell.

SYN Synchronize.

TCP Transmission Control Protocol.

UDP User datagram Protocol.

UML Unified Modeling Language.

USB Universal Serial Bus.

VIM Vi Improved.

VPN Virtual Private Network.

XML Extensible Markup Language.

General Introduction

In the current era of digital transformation, the proliferation of networked systems has exponentially increased the scope and complexity of cyber threats. The need for robust security mechanisms has become paramount to protect sensitive data and ensure the integrity of information systems. Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are pivotal in this landscape, serving as the first line of defense against malicious activities. This project, titled "Implementation of a Monitoring System for Intrusion Detection and Prevention," aims to develop a comprehensive solution that not only detects but also prevents unauthorized access and potential attacks. Through a combination of theoretical analysis and practical implementation, this project aspires to contribute significantly to the field of cybersecurity by enhancing the resilience of networked environments.

Chapter 1

General Context

1.1 Introduction

This chapter sets the stage for understanding the structural and strategic foundations of the implemented monitoring system. This chapter outlines the project's scope, objectives, and the methodologies employed to achieve the desired outcomes. It serves as a blueprint, detailing the conceptualization, design, and execution phases of the project. By providing a clear and structured overview, this chapter ensures that readers grasp the comprehensive approach undertaken in developing an effective intrusion detection and prevention system.

1.2 General Project Presentation

1.2.1 Problem Statement

In today's digital age, the increasing prevalence of cyber threats poses significant risks to networked systems. Organizations face the constant challenge of protecting their sensitive data and maintaining the integrity of their information systems against unauthorized access and malicious attacks. Traditional security measures are often insufficient to detect and prevent sophisticated cyber threats. This project aims to address the critical need for an advanced monitoring system that can effectively detect and prevent intrusions, thereby enhancing network security.

1.2.2 Objectives

The primary objectives of this project are:

- To design and implement a comprehensive monitoring system for intrusion detection and prevention (IDPS).

- To enhance the accuracy and efficiency of detecting various types of cyber threats and attacks.
- To integrate real-time monitoring and automated response mechanisms to prevent potential intrusions.
- To evaluate the performance and effectiveness of the implemented system in a controlled environment

1.2.3 Scope

The scope of this project includes:

- Research and analysis of existing IDS and IPS technologies.
- Design and development of a custom monitoring system tailored to specific network environments.
- Implementation of detection algorithms for identifying different types of attacks.
- Integration of preventive measures to block detected threats.
- Testing and evaluation of the system's performance in detecting and preventing intrusions. Exclusions:
- The project will not focus on physical security measures or user training programs.
- The project will not include the development of a commercial-grade product for mass deployment.

1.2.4 Methodology

To achieve the project objectives, the following methodology will be employed:

- Literature Review: Conduct a comprehensive review of existing research and technologies in the field of intrusion detection and prevention.
- System Design: Develop a detailed design for the monitoring system, including the architecture, components, and detection algorithms.
- Implementation: Use programming languages such as Javascript and Python to develop the system components. Tools like Snort and Suricata will be utilized for intrusion detection.

- Testing: Conduct extensive testing using simulated attack scenarios to evaluate the system's effectiveness and performance.

- Evaluation: Analyze the test results to assess the system's accuracy, efficiency, and overall impact on network security.

1.2.5 Expected Outcomes

The expected outcomes of this project include:

- A fully functional monitoring system capable of detecting and preventing a wide range of cyber threats.

- Improved accuracy and speed in identifying malicious activities within the network.

- Enhanced security posture for networked systems through real-time intrusion detection and automated preventive actions.

- Comprehensive documentation of the system's design, implementation, and evaluation results, providing valuable insights for future research and development in the field of cybersecurity.

1.3 Design

1.3.1 System Architecture:

The monitoring system will be based on a layered architecture, consisting of the following components:

- Sensor Layer: Collects network traffic and system logs for analysis.

- Detection Layer: Applies detection algorithms to identify potential threats

- Response Layer: Executes automated responses to mitigate detected threats.

- Management Layer: Provides a user interface for monitoring and managing the system.

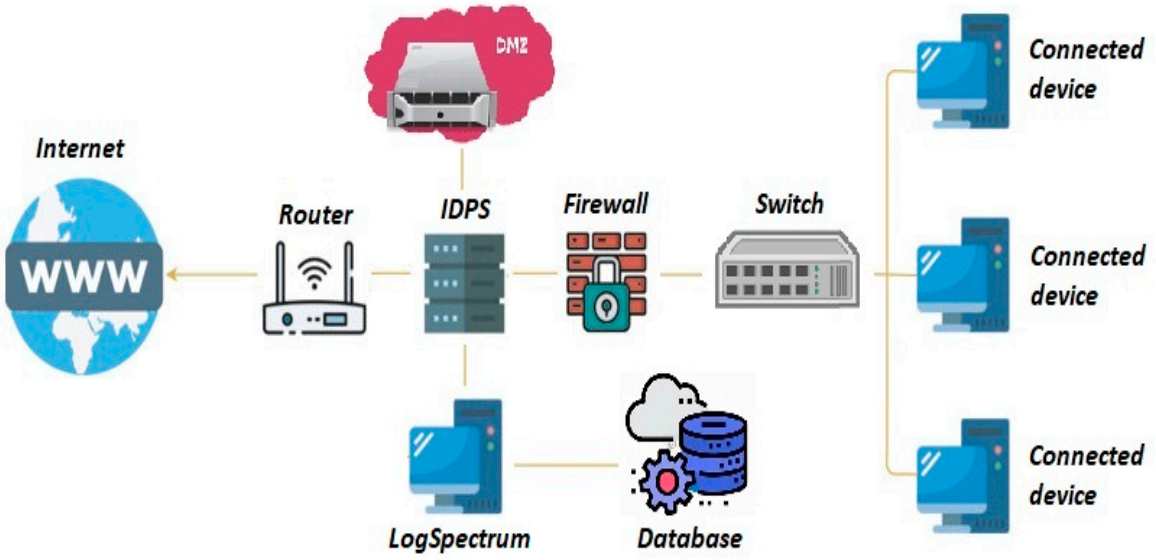

Figure 1.1: Architecture of NIDS

1.3.2 Components Description:

- Network Sensors: Devices or software agents that capture network traffic.

- User Interface: A dashboard for real-time monitoring and configuration management.

1.3.3 Technology Stack:

- Programming Languages: javascript for scripting and Node.Js framework for Back-end.

- Tools: Snort for intrusion detection,

1.4 Project Planning

The provided grant chart offers a comprehensive overview of the project' s timeline. The chart displays a well-structured plan, enabling a clear understanding of the project' s progression. Overall, it is a well-structured and comprehensive chart that provides a solid foundation for successful project execution.

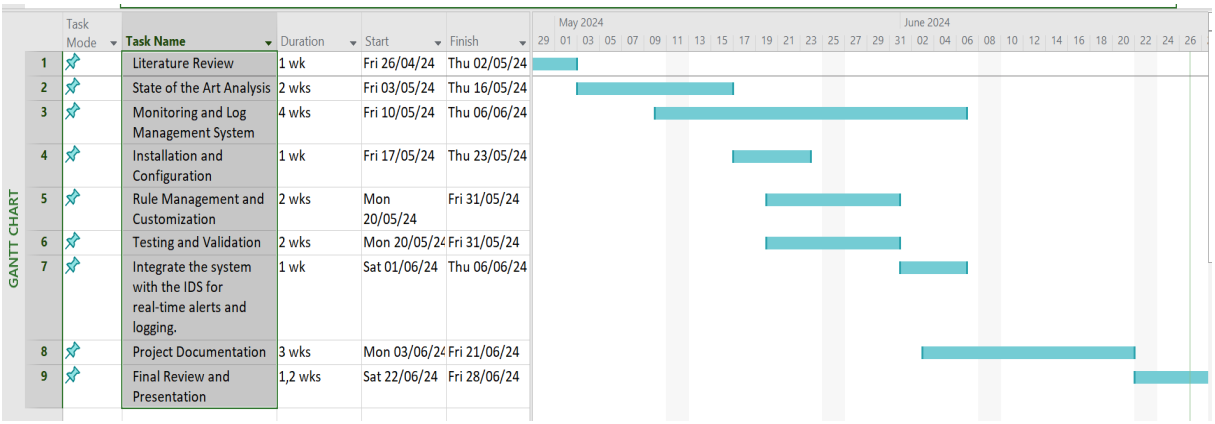

Figure 1.2: Timeline Representation

1.5 Conclusion

In conclusion, this chapter has outlined the comprehensive framework for the "Implementation of a Monitoring System for Intrusion Detection and Prevention" project. By clearly defining the problem statement, objectives, scope, methodology, and expected outcomes, we have established a solid foundation for the project. The detailed design and planning sections provide a roadmap for the successful execution and completion of the project. This framework not only ensures a structured approach to development but also sets the stage for addressing the challenges and achieving the goals of enhancing network security through advanced intrusion detection and prevention mechanisms. Moving forward, the subsequent chapters will delve into the implementation details and the evaluation of the system's effectiveness.

Chapter 2

State of the Art

2.1 Introduction

The ”State of the Art” chapter delves into the current advancements and existing methodologies in the field of cybersecurity, particularly focusing on intrusion detection and prevention. This chapter provides a thorough examination of fundamental security concepts, prevalent threats, and the evolution of IDS and IPS technologies. By comparing various systems and analyzing their effectiveness, this chapter aims to highlight the strengths and limitations of current solutions, thereby setting the context for the innovations introduced in this project. This detailed exploration not only underscores the importance of the research but also situates it within the broader landscape of ongoing cybersecurity efforts.

2.2 Basic Concepts of Cybersecurity

2.2.1 What is cybersecurity?

Cybersecurity is about protecting computers, servers, mobile devices, electronic systems, networks and data from malicious attacks. It is also called computer security or information systems security. You can meet it in many contexts, from corporate IT to mobile terminals. It can be divided into several categories.[1]

- **Network security:** it is about protecting the computer network from intruders, whether they are targeted attacks or opportunistic malware;
- **Application security:** aims to protect software and devices from threats. A corrupted application could open access to the data it is supposed to protect. A reliable security system is recognized from the design stage, well before the deployment of a program or device.

- **Information security:** ensures the integrity and confidentiality of the data, whether it is stored or in transit.
- **Operational safety** understands the processes and decisions related to the processing and protection of data. User permissions for network access and procedures that define data storage and location fall under this type of security.
- **Disaster recovery and business continuity:** specify how a company responds to a cybersecurity incident or any other event causing a loss of operations or data. Disaster recovery policies govern how a company recovers its operations and information to regain the same operational capability as before the event. Business continuity refers to the plan on which a business relies while trying to operate without certain resources.
- **Training of end users:** the most unpredictable factor: people. Anyone can accidentally introduce a virus into a usually secure system by not following good security practices. Teaching users to remove suspicious attachments and not plug in unidentified USB drives is essential for the security of a business.

2.2.2 Cybersecurity Fundamentals

2.2.2.1 The three concepts of cyber security

The foundation of cyber security basics lies in the CIA triad, which stands for confidentiality, integrity and availability. These three concepts of cyber security form the basis of protecting sensitive information and ensuring digital systems can operate securely.[2]

- **Confidentiality**

  Within the CIA triad, 'confidentiality' refers to the assurance that data and information are accessible only to authorised persons. It involves strict security measures like encryption and access controls, which are deployed to prevent unauthorised access and data breaches.

- **Integrity**

  Integrity is all about keeping the data accurate and unaltered during transfers and storage. Any unauthorised modification or tampering with data will compromise its integrity, potentially leading to serious consequences.

- **Availability**

  Availability ensures that systems, data and resources are accessible and usable by autho-

rised individuals around the clock. Protection against denial-of-service (DoS) attacks and system failures is crucial to maintaining availability.

2.2.2.2 the basics of cybersecurity

Aspiring professionals should familiarise themselves with some cyber security fundamentals:

- Common cyber security terminology.
- types of cyber security threats.
- cyber security best practices.

1. **Common cybersecurity terminology**
   Beyond the basic concepts of cyber security, you can' t master the fundamentals of cyber security without first understanding the common parlance. Here are some cyber security terms you should familiarise with:

   - **Firewall:** A security barrier between internal and external networks, designed to filter and block unauthorised traffic.
   - **Encryption:** This is the process of converting data into code that protects it during transfers or storage.
   - **SSH:** Secure Shell is a network protocol that allows secure remote login and other network services to operate over an insecure network. It provides a secure channel over an unsecured network in a client-server architecture, connecting an SSH client application with an SSH server. SSH uses cryptography to authenticate and encrypt connections between devices.
   - **TCP:** Transmission Control Protocol is a standard communication that allows application programs and computing devices to exchange messages over a network. It is designed to send packets over the Internet and ensure the successful transmission of data and messages over networks.
   - **FTP:** File Transfer Protocol is a standard network protocol used for transferring files between a client and a server on a computer network.
   - **ICMP:** Internet Control Message Protocol, generally works with other network protocols such as TCP/IP or UDP (User Datagram Protocol, user datagram protocol). Hosts and routers exchange ICMP messages or packets when certain network events occur.

- **UDP:** User Datagram Protocol is a protocol of substitution communication TCP. It is mainly used to establish low latency and loss tolerance connections between applications on Internet. UDP and TCP run on the IP protocol (Internet Protocol) and are therefore sometimes referred to as UDP/IP and TCP/IP respectively. Both protocols send data in small packets, or datagrams.
- **Vulnerability:** A weakness or flaw in a system that can be exploited.
- **Two-factor authentication (2FA):** This requires users to provide two forms of ID before they can access an account or system.
- **Antivirus:** Antivirus software is designed to detect, prevent and remove malicious software.
- **Penetration testing:** This is a controlled simulation of cyber attacks on a system or network that can help identify vulnerabilities and assess its security posture.
- **Virtual private network (VPN):** A VPN is a secure and encrypted network connection that allows users to access the internet or a private network confidentiality.
- **DMZ:** A demilitarized zone (DMZ) is a subnetwork that is separated from the local network and isolated from it as well as from the Internet (or another network) by a firewall. This subnetwork contains machines that are likely to be accessed from the Internet and do not need access to the local network.

  Services that are likely to be accessed from the Internet are located in the DMZ, and all incoming traffic from the Internet is redirected by default to the DMZ by the firewall. The firewall will therefore block access to the local network from the DMZ to ensure security. In case one of the services in the DMZ is compromised, the attacker will only have access to the machines in the DMZ and not the local network.

2. **Types of Cybersecurity Threats**

   While malicious actors are continuously developing and deploying new types of cyber attacks, there are some common threats aspiring cyber professionals should be aware of, including:

   - **Malware:** Short for malicious software, malware covers a wide range of harmful software, including viruses, worms, Trojans and ransomware. Malware can disrupt systems, steal sensitive data and even extort money from targets.
   - **Phishing:** Phishing is a social engineering technique used by cybercriminals to trick users into revealing sensitive information. This is often in the form of login credentials

or financial details, which are 'phished' using deceptive emails, links, downloads and websites.

- **Data breaches:** Data breaches happen when an unauthorised attacker gains access to sensitive information, which usually leads to personal or confidential data being exposed.
- **Denial-of-service attacks:** In DoS attacks, the attacker floods a system or network with an overwhelming amount of traffic, which causes it to become unavailable to legitimate users.
- **Insider threats:** Insider threats are when someone within an organisation misuses their privileges to access and steal sensitive information or intentionally cause disruption.
- **Man in the middle:** This attack occurs when an attacker intercepts and possibly alters the communication between two parties without their knowledge.
- **Eadvesdropping attack:** Also known as sniffing or snooping, this attack involves intercepting and listening to the communication between two parties over the network.
- **Drive-By Attack:** This attack involves malicious software being automatically downloaded onto a user's device when they visit a compromised or malicious website, without any user interaction.
- **SQL Injection Attack:** This attack involves inserting malicious SQL code into a query input to manipulate the database, allowing the attacker to access, modify, or delete data.

3. **Cybersecurity best practices**

   The best practices described below should be considered cyber security fundamentals, but they aren' t an exhaustive list. Instead, cyber professionals will need to continuously monitor and assess potential risks, stay informed about emerging threats and adapt best practices accordingly.[3]

   - **Strong passwords**

     Arguably the fundamental of cyber security is the use of strong, unique passwords. A strong password should include a combination of upper- and lower-case letters, numbers and special characters. The worst thing you can do is to use passwords with easily guessable information, such as birthdays or common words and numbers. It' s also important to use a different password for each account.

- **Regular software updates**

  Developers regularly release updates to their software that include security patches and address vulnerabilities and bugs. Enabling automatic updates for your operating systems and commonly used programs will protect against known exploits and potential cyber attacks.

- **Data backup**

  Data loss can happen for any number of reasons – ransomware attacks, hardware failures or even disasters like a fire or flood. Both individuals and organisations should have a robust data-backup strategy in place so you can always access critical information – even in the event of a cyber incident or system failure.

- **Network segmentation**

  Network segmentation involves dividing a network into smaller segments – or subnetworks – to limit the potential impact of a cyber attack. By isolating sensitive data and resources from the rest of the network, you can contain and mitigate the effects of any breach.

2.3 Intrusion Detection Systems(IDS)

Intrusion Detection Systems (IDS) are cybersecurity tools designed to monitor network or system activities for suspicious or malicious behavior and alert security personnel or automated systems when potential threats are detected. Here's an overview of IDS and their key aspects:

2.3.1 Types of IDS:

- **Network-Based IDS (NIDS):** NIDS monitors network traffic in real-time, analyzing packets and traffic patterns to detect unauthorized access attempts, malware activities, and suspicious behavior within the network.

- **Host-Based IDS (HIDS):** HIDS resides on individual host systems, monitoring activities such as file system changes, logins, process executions, and configuration modifications to detect intrusions and anomalies at the host level.

2.3.2 Detection Techniques:

- **Signature-Based Detection:** Signature-based IDS compare observed activities against known attack signatures or patterns stored in signature databases. They are effective at detecting known threats but may miss novel or zero-day attacks.

- **Anomaly-Based Detection:** Anomaly-based IDS establish a baseline of normal behavior and flag deviations or anomalies that indicate potential intrusions or unusual activities. They are useful for detecting previously unseen threats but may generate false positives.
- **Hybrid Detection:** Hybrid IDS combine signature-based and anomaly-based detection techniques to leverage the strengths of both approaches, enhancing detection accuracy and reducing false alarms.

2.3.3 Deployment Modes:

- **Inline IDS:** Inline IDS operate in-line with network traffic, actively blocking or allowing traffic based on threat detection results. They provide real-time prevention capabilities but may introduce latency or disrupt network traffic if misconfigured.
- **Passive IDS:** Passive IDS operate in monitoring mode, observing network or host activities without actively interfering with traffic. They generate alerts and reports for analysis and response by security personnel or automated systems.

2.3.4 Key Components:

- **Sensors:** IDS sensors collect and analyze data from network traffic (NIDS) or host activities (HIDS) to detect potential intrusions and anomalies.
- **Analysis Engine:** The analysis engine processes sensor data, applies detection algorithms (signature-based, anomaly-based, or hybrid), and generates alerts or alarms for suspicious activities.
- **Alerting and Reporting:** IDS systems generate alerts, notifications, and reports to inform security personnel about detected threats, providing details such as threat severity, affected systems, and recommended response actions.
- **Integration with SIEM:** IDS can integrate with Security Information and Event Management (SIEM) systems to correlate and analyze IDS alerts alongside other security event data for comprehensive threat visibility and response coordination.

2.3.5 Benefits and Challenges:

- **Benefits:** IDS enhance cybersecurity by providing early detection of intrusions, reducing incident response times, improving threat visibility, and enabling proactive threat mitigation.

- **Challenges:** IDS may generate false positives (false alarms), require fine-tuning and maintenance, consume network resources for monitoring, and face evasion techniques from sophisticated attackers.

Overall, IDS play a crucial role in network security by detecting and alerting organizations to potential cyber threats and intrusions, enabling timely response and mitigation actions to protect critical assets and data.

2.4 Intrusion Prevention Systems(IPS)

Intrusion Prevention Systems (IPS) are cybersecurity solutions that go beyond intrusion detection by actively blocking or preventing malicious activities and intrusions in real-time. Here's an overview of IPS and its key aspects:

2.4.1 Functionality:

- **Real-Time Threat Prevention:** IPS continuously monitors network traffic or host activities, analyzes patterns and behaviors, and takes proactive action to block or mitigate potential threats before they can cause harm.

- **Inline Protection:** Inline IPS operates in-line with network traffic, inspecting packets and applying security policies to allow, deny, or modify traffic based on threat detection results. This inline protection ensures immediate response to detected threats.

- **Signature-Based and Behavior-Based Detection: :** IPS combines signature-based detection (matching against known attack signatures) and behavior-based detection (identifying abnormal patterns or deviations from normal behavior) to identify and block malicious activities.

2.4.2 Deployment Modes:

- **Inline IPS:** Inline IPS actively intercepts and analyzes network traffic, applying security policies to prevent unauthorized access, exploits, malware downloads, and other malicious activities in real-time.

- **Passive IPS:** Passive IPS operates in monitoring mode, observing network activities without direct intervention. It generates alerts and recommendations for action but does not actively block or modify traffic.

2.4.3 Key Components:

- **Sensors:** IPS sensors collect and analyze network traffic or host activities, identifying potential threats, vulnerabilities, and anomalies.
- **Analysis Engine:** The analysis engine in IPS processes sensor data, applies detection algorithms (signature-based, behavior-based), and makes decisions on whether to block, allow, or alert on detected threats.
- **Blocking Mechanisms:** IPS employs blocking mechanisms such as access control lists (ACLs), firewall rules, and packet filtering to block or modify malicious traffic and prevent successful intrusions.
- **Logging and Reporting:** IPS systems log security events, generate alerts, and provide detailed reports on detected threats, blocked activities, and security policy violations for analysis and response.

2.4.4 Benefits and Challenges:

- **Benefits:** IPS provides proactive threat prevention, reduces false positives compared to IDS, enhances network security posture, improves incident response capabilities, and protects critical assets and data from cyber attacks.
- **Challenges:** IPS may introduce latency in network traffic due to real-time inspection, require careful configuration and tuning to minimize false positives and negatives, and face evasion techniques from advanced attackers.

Overall, IPS complements intrusion detection systems (IDS) by adding proactive threat prevention capabilities, helping organizations defend against a wide range of cyber threats, including malware, exploits, network intrusions, and unauthorized access attempts.

2.5 Comparative Study of Intrusion Detection Systems

In this comparative study, we will analyze and compare several Intrusion Detection Systems (IDSs) including Snort, Suricata, OSSEC, Bro IDS (Zeek), and Security Onion. Each IDS will be evaluated based on key criteria to determine the best option for different use cases.

1. **Snort:**
    - *Strengths:*

  - Customizability: Highly customizable with extensive rule sets and plugins.
  - Community Support: Strong community backing with frequent updates and a large repository of predefined rules.
  - Efficiency: Handles moderate to high traffic volumes effectively.
  - Proven Track Record: Trusted by numerous organizations for its reliability and effectiveness.
  - Ease of Integration: Integrates well with other security tools and SIEM systems.
- *Weaknesses:*
  - Complexity: Can be complex to configure and maintain, requiring expertise.
  - False Positives: Tends to generate false positives if not properly tuned.

2. **Suricata:**

- *Strengths:*
  - Performance: Superior performance in multi-core environments due to its multi-threading capability.
  - Protocol Support: Extensive support for various network protocols.
  - Versatility: Can function as both IDS and IPS.
- *Weaknesses:*
  - Resource Intensive: Requires substantial resources, especially in high-traffic scenarios.
  - Configuration Complexity: Needs careful configuration and tuning to operate optimally.
  - Community Support: While growing, the community and rule base may not be as extensive as some other solutions.

3. **OSSEC:**

- *Strengths:*
  - Detailed Host Monitoring: Monitors a wide range of host activities including system logs, file integrity, and process behavior.
  - Cross-Platform: Supports multiple operating systems.
  - Automated Response: Capable of taking automated actions based on detected threats.
- *Weaknesses:*

  - Host-Specific: Limited to monitoring individual hosts, lacking network-wide visibility.
  - Performance Impact: Can impact the performance of monitored hosts.

4. **Bro IDS (Zeek):**

   - *Strengths:*
     - Network Traffic Analysis: Provides detailed insights into network traffic and protocols.
     - Protocol Parsing: Capable of parsing a wide range of network protocols.
     - Scripting Capabilities: Supports scripting for custom analysis.
   - *Weaknesses:*
     - Complexity: Configuring and maintaining Bro IDS may require expertise.
     - Resource Usage: Can be resource-intensive, especially in environments with high network traffic.

5. **Security Onion:**

   - *Strengths:*
     - Integrated Solution: Comprehensive suite of security tools, including IDS/IPS capabilities.
     - Ease of Deployment: Simplifies deployment of multiple security tools through a single platform.
     - Community and Support: Benefits from a broader community and support ecosystem.
   - *Weaknesses:*
     - Resource Requirements: Requires sufficient resources, especially when running multiple security tools simultaneously.
     - Learning Curve: Users may need time to familiarize themselves with the platform.

2.6 Conclusion

After a thorough comparative analysis, the best IDS solution depends on specific requirements:

- *For Network Monitoring:*

  - High Customizability and Community Support: Snort stands out with its extensive rule sets, plugins, and strong community backing.
  - Performance and Versatility: Suricata offers superior performance and versatility, especially in multi-core environments.

- *For Host Monitoring:*
  - Comprehensive Host-Level Security: OSSEC excels in detailed host monitoring, log analysis, and automated response capabilities.

- *For Integrated Security Suite:*
  - Comprehensive Security Suite: Security Onion provides a holistic approach with a suite of security tools, including IDS/IPS capabilities.

Ultimately, the best IDS choice depends on the specific network environment, the level of customization required, resource availability, and the need for integration with other security tools. Snort and Suricata are strong contenders for network monitoring, while OSSEC is ideal for host-level security. Security Onion offers an integrated solution for organizations seeking a comprehensive security suite.

Chapter 3

Implementation of a Snort IDS

3.1 Introduction

In this chapter, we delve into the practical aspects of implementing a Snort Intrusion Detection System (IDS) within a network environment. Snort, as one of the most widely used and versatile IDS solutions, offers robust capabilities for detecting and responding to a myriad of network-based threats.

The implementation of Snort involves several critical steps, ranging from installation and configuration to rule management and performance optimization. This chapter will guide us through each stage of the process, providing detailed instructions and best practices to ensure a successful deployment.

We begin by discussing the prerequisites and environment setup necessary for installing Snort, including the required software dependencies and hardware considerations. Following this, we will cover the installation process on a Unix-based operating system [Ubuntu], which is commonly used in enterprise settings for its stability and security features.

Configuration is a key aspect of any IDS deployment, and Snort's flexibility allows for extensive customization to meet the specific needs of our network. We will explore how to configure Snort to monitor network traffic effectively, including setting up network interfaces, defining home and external networks, and customizing output options for alerts and logs.

An essential component of Snort's functionality is its rule sets, which are used to identify suspicious activities based on predefined patterns. We will provide an overview of Snort's rule syntax and structure, demonstrate how to write and customize rules, and discuss the integration of community-contributed rule sets to enhance detection capabilities.

Performance and optimization are crucial for maintaining the efficiency of Snort in high-traffic environments. This chapter will offer strategies for tuning Snort's performance, managing resource utilization, and minimizing false positives to ensure accurate and timely threat

detection.

By the end of this chapter, we will have a comprehensive understanding of how to implement Snort IDS in our network, from initial setup to advanced configuration and optimization techniques. This knowledge will empower us to deploy an effective intrusion detection solution that enhances our network's security posture and provides valuable insights into potential threats.

3.2 Description

Snort is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plugin architecture. Snort also has a modular real-time alerting capability, incorporating alerting and logging plugins for syslog, a ASCII text files, UNIX sockets or XML.

Snort has three primary uses. It can be used as a straight packet sniffer like the tcpdump, a packet logger (useful for network traffic debugging, etc), or as a full blown network intrusion detection system.

Snort logs packets in the tcpdump binary format or in Snort's decoded ASCII format to a hierarchy of logging directories that are named based on the IP address of the "foreign" host.

3.3 Installation and Configuration

3.3.1 Installation:

To install Snort IDS, we start by updating the package lists on our Ubuntu system with the command **sudo apt update**. Once the package lists are updated, you can install Snort directly from the repository by executing **sudo apt install snort**. This command will automatically handle the installation of Snort and its dependencies as depicted in the image.

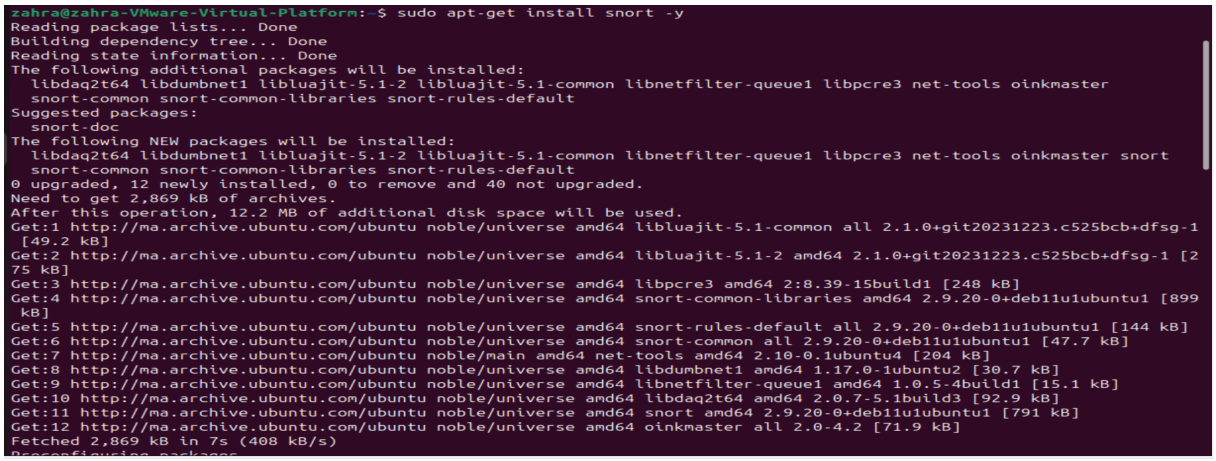

Figure 3.1: Installation of Snort

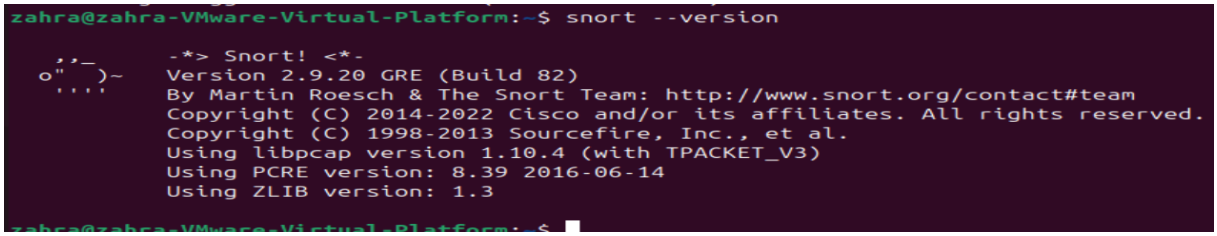

Figure 3.2: Snort Version

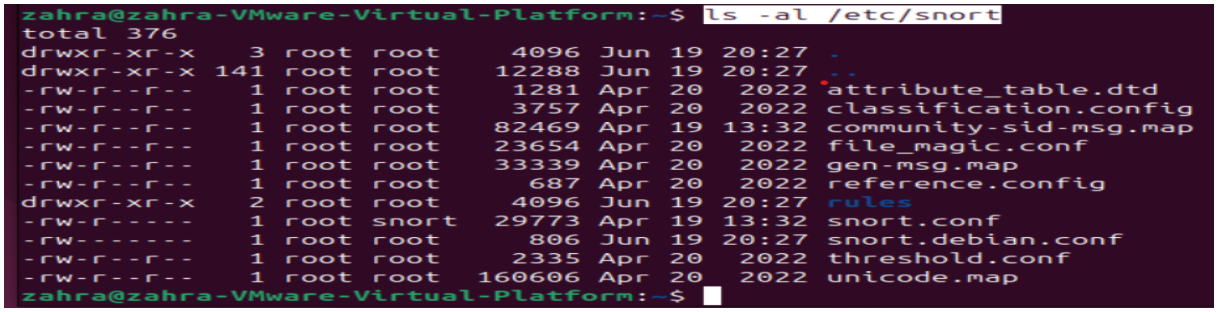

Figure 3.3: Snort Folder Files

3.3.2 Configuration:

After the installation is complete, We can configure Snort by editing the **snort.conf** file, which is typically located in **/etc/snort/**. In this configuration file, we need to update the network variables to match the network environment and specify the paths for the rules we will be using. Ensure that all necessary variables and rule paths are correctly defined.

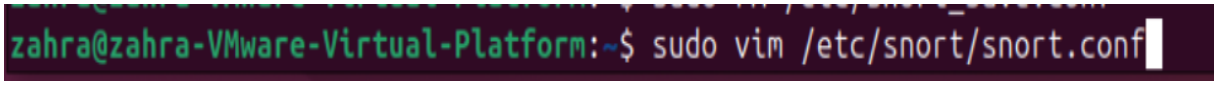

Figure 3.4: Vim Snort Config

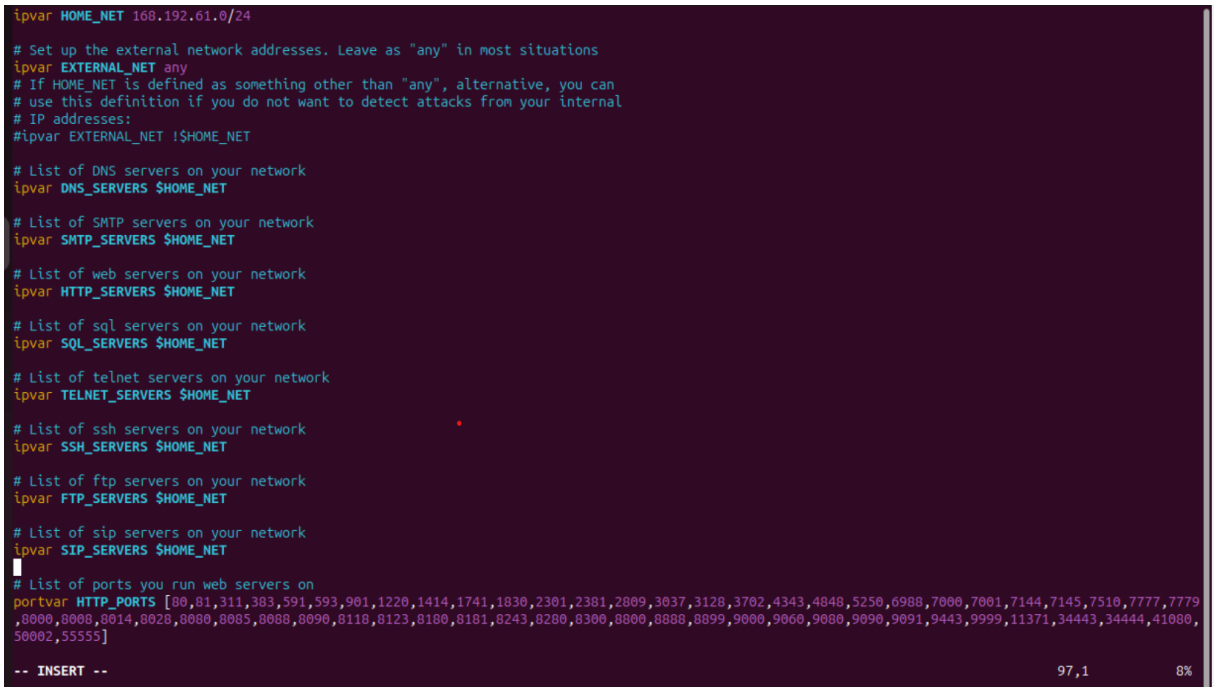

Figure 3.5: Ipvar Config

Figure 3.6: Include Rule Files

3.3.3 Run in Test Mode:

Running Snort in test mode allows as to check for any configuration errors without starting the actual intrusion detection. With the command sudo **snort T -c /etc/snort/snort.conf -i ens33**. This command will parse the configuration file and load all specified rules, providing feedback on any issues or errors encountered.

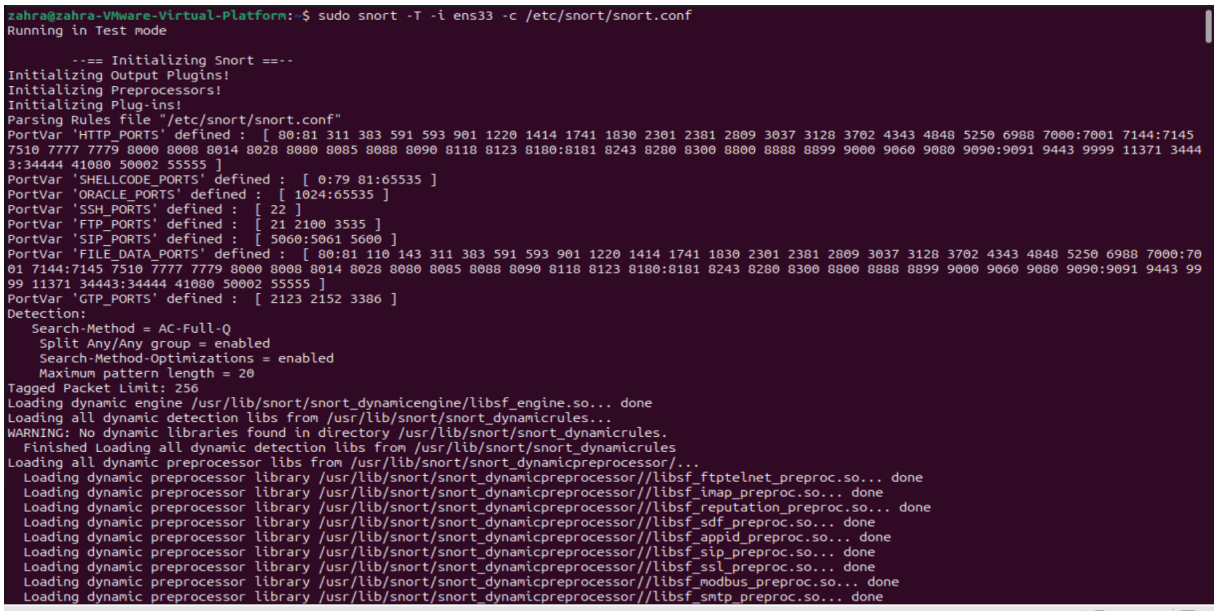

Figure 3.7: Running Snort Mode Test

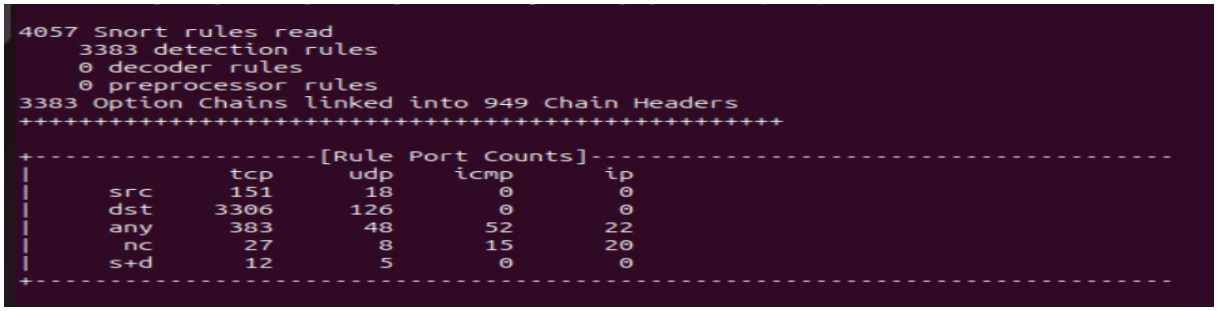

Figure 3.8: Read Snort Rules

3.4 Rule Management

define the behavior and characteristics of potential intrusions or attacks. Rule management includes tasks such as writing new rules to detect emerging threats, modifying existing rules to improve detection accuracy, categorizing rules based on severity levels or attack types, and periodically updating rules to address evolving security challenges. An efficient rule management strategy ensures that the IDS remains effective in identifying and mitigating security incidents while minimizing false positives and optimizing system performance.

```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
alert icmp any any -> $HOME_NET any (msg:"ICMP Ping Detector"; sid:100001; rev:1;)
alert tcp any any -> $HOME_NET 22 (msg:"SSH Authentication Attempt"; sid:100002; rev:1;)
alert tcp any any -> 192.168.61.135 21 (msg:"FTP Authentication Attempt On MS2" sid:100003; rev:1;)
alert tcp any any -> $HOME_NET 445 (msg:"OS-WINDOWS Microsoft Windows SMB remote code execution attempt"; flow:to_se
rver,established; content:"|FF|SMB|00 00 00 00|"; depth:9; offset:4; byte_extract:2,26,TotalDataCount,relative,litt
le; byte_test:2,>,TotalDataCount,20,relative,little; metadata:policy balanced-ips drop, policy connectivity-ips drop
, policy max-detect-ips drop, policy security-ips drop, ruleset community, service netbios-ssn; reference:cve,2017-0
144; reference:cve,2017-0146; reference:url,blog.talosintelligence.com/2017/05/wannacry.html; reference:url,isc.sans
.edu/forums/diary/ETERNALBLUE+Possible+Window+SMB+Buffer+Overflow+0Day/22304/; reference:url,technet.microsoft.com/e
n-us/security/bulletin/MS17-010; classtype:attempted-admin; sid:100004; rev:5;)
```

Figure 3.9: Rules configuration

3.5 Running Attacks

3.5.1 Testing Protocols Network:

3.5.1.1 ICMP Ping

We sent a ping to the victim machine, and effectively snort had detected the icmp ping and displayed the configured message and the SID and its priority, and specified the protocol network as it identified the IP address of the attacking machine.

```
┌──(kaliza㉿kaliza)-[~]
└─$ ping 192.168.61.128
PING 192.168.61.128 (192.168.61.128) 56(84) bytes of data.
64 bytes from 192.168.61.128: icmp_seq=1 ttl=64 time=3.84 ms
64 bytes from 192.168.61.128: icmp_seq=2 ttl=64 time=1.41 ms
64 bytes from 192.168.61.128: icmp_seq=3 ttl=64 time=5.15 ms
64 bytes from 192.168.61.128: icmp_seq=4 ttl=64 time=1.27 ms
64 bytes from 192.168.61.128: icmp_seq=5 ttl=64 time=0.954 ms
64 bytes from 192.168.61.128: icmp_seq=6 ttl=64 time=2.44 ms
64 bytes from 192.168.61.128: icmp_seq=7 ttl=64 time=3.91 ms
64 bytes from 192.168.61.128: icmp_seq=8 ttl=64 time=2.85 ms
64 bytes from 192.168.61.128: icmp_seq=9 ttl=64 time=2.45 ms
64 bytes from 192.168.61.128: icmp_seq=10 ttl=64 time=3.09 ms
64 bytes from 192.168.61.128: icmp_seq=11 ttl=64 time=2.97 ms
64 bytes from 192.168.61.128: icmp_seq=12 ttl=64 time=2.46 ms
64 bytes from 192.168.61.128: icmp_seq=13 ttl=64 time=1.63 ms
64 bytes from 192.168.61.128: icmp_seq=14 ttl=64 time=1.18 ms
64 bytes from 192.168.61.128: icmp_seq=15 ttl=64 time=2.02 ms
64 bytes from 192.168.61.128: icmp_seq=16 ttl=64 time=1.52 ms
```

Figure 3.10: ICMP Ping

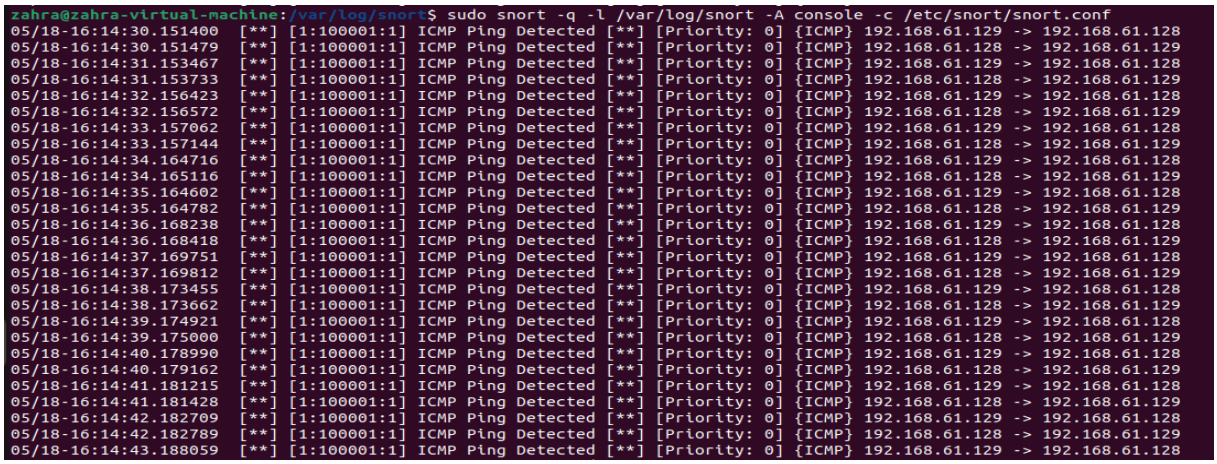

Figure 3.11: Alerting ICMP Ping

3.5.1.2 SSH

We tried a SSH log into the victim machine, and snort had detected it and displayed the configured message about the "SSH authentication attempt", the SID and the priority, and specified that is a TCP protocol network as it identified the IP address of the attacking machine.

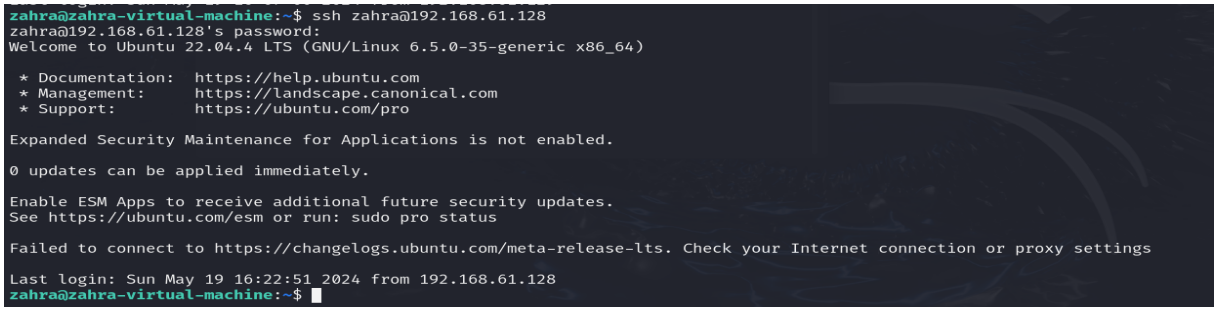

Figure 3.12: SSH Log

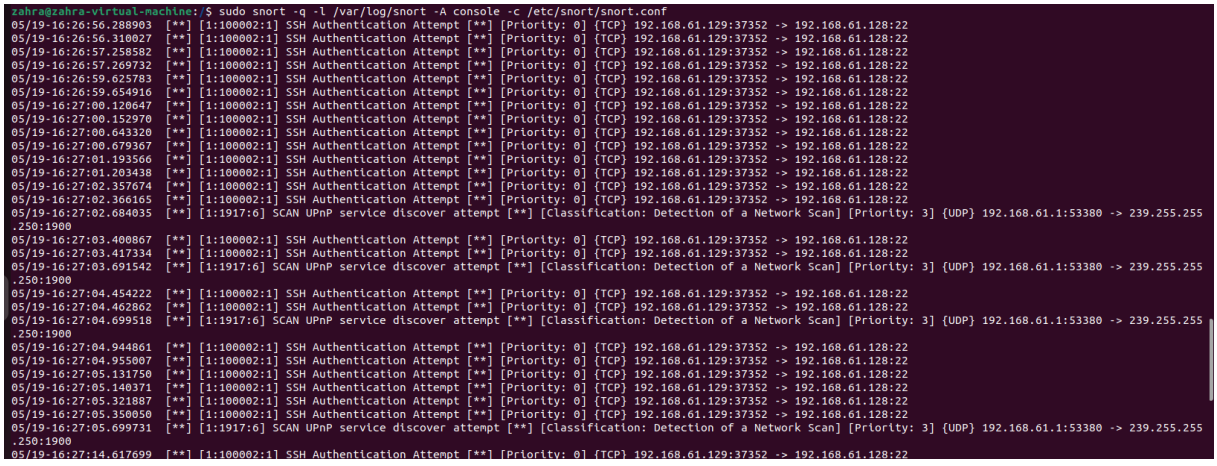

Figure 3.13: Alerting SSH Log

3.5.1.3 FTP

we tried a FTP connection with the victim machine, and snort had detected it and displayed the configured message and the SID and its priority, and specified that is a TCP protocol network as it identified the IP address of the attacking machine.

```
zahra@zahra-virtual-machine:~$ ftp 192.168.61.128 21
Connected to 192.168.61.128.
220 (vsFTPd 3.0.5)
Name (192.168.61.128:zahra): zahra
331 Please specify the password.
Password:
500 OOPS: vsftpd: refusing to run with writable root inside chroot()
ftp: Login failed
ftp>
```

Figure 3.14: FTP Connection

```
zahra@zahra-virtual-machine:/etc/snort$ sudo snort -q -l /var/log/snort -A console -c /etc/snort/snort.conf
05/26-19:23:29.912390  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:51454 -> 192.168.61.128:21
05/26-19:23:29.912678  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:51454 -> 192.168.61.128:21
05/26-19:23:31.687506  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:39340 -> 192.168.61.128:21
05/26-19:23:31.689421  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:39340 -> 192.168.61.128:21
05/26-19:23:31.703485  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:39340 -> 192.168.61.128:21
05/26-19:23:35.009273  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:39340 -> 192.168.61.128:21
05/26-19:23:35.011176  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:39340 -> 192.168.61.128:21
05/26-19:23:40.008490  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:39340 -> 192.168.61.128:21
05/26-19:23:40.197712  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:39340 -> 192.168.61.128:21
05/26-19:23:40.199591  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:39340 -> 192.168.61.128:21
05/26-19:23:40.201739  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:39340 -> 192.168.61.128:21
05/26-19:23:40.204039  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:39340 -> 192.168.61.128:21
05/26-19:23:40.247022  [**] [1:100003:1] FTP Authentication Attempt On MS2 [**] [Priority: 0] {TCP} 192.168.61.129:39340 -> 192.168.61.128:21
05/26-19:24:13.664158  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {IPV6-ICMP} fe80::c1ef:7984:76cc:e13 -> ff02::1:ff9e:6e90
```

Figure 3.15: Alerting FTP Connection

3.5.2 Flooding

Flooding is a type of network attack where an attacker sends a large amount of traffic to a target to overwhelm its resources, causing a denial of service. Common types include SYN flooding, UDP flooding, and ICMP flooding

3.5.2.1 Ping DoS Attack (ICMP Flood (Ping Flood))

We sent a hping3 packet to the victim machine, and effectively snort had detected the hping3 request and displayed the configured message, the SID and its priority, and specified the protocol network IPV6-ICMP as it identified the IP address of the attacking machine.

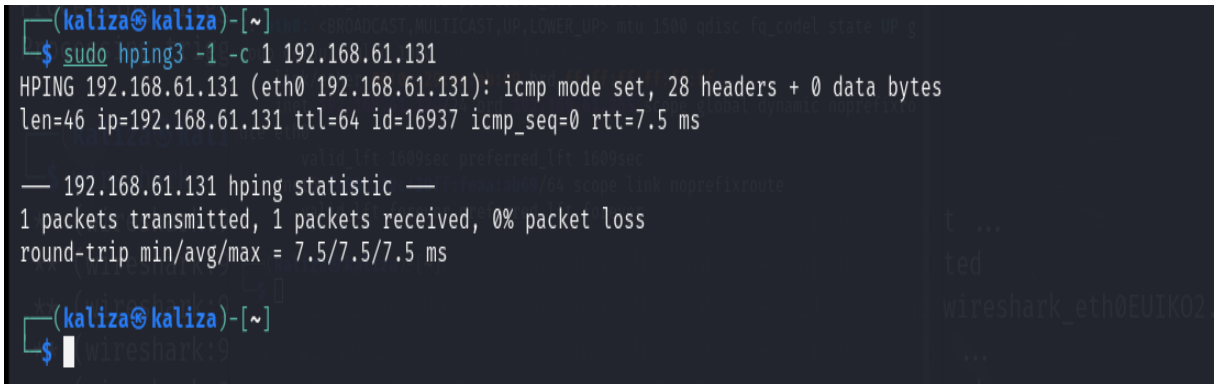

Figure 3.16: ICMP Flood Ping

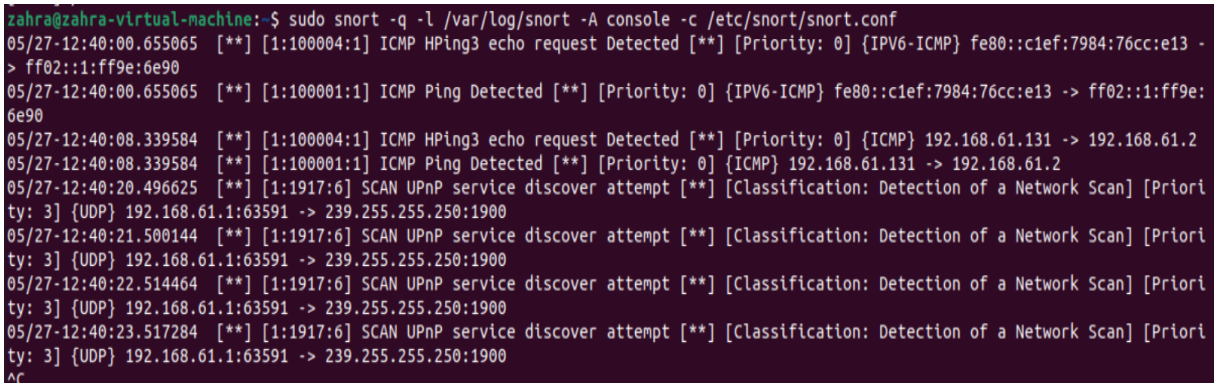

Figure 3.17: Alerting ICMP Flood Ping

3.5.2.2 SYN Flood DoS attack

We applied a SYN flood DOS Attack using hping3.

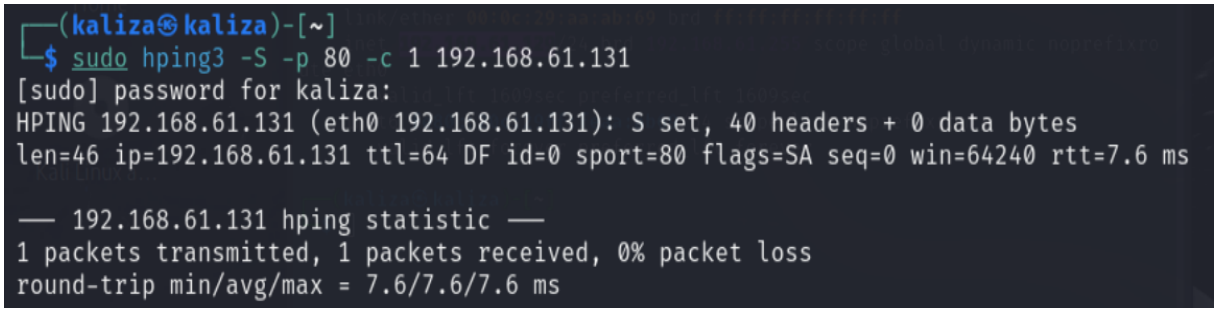

Figure 3.18: SYN Flood DOS Attack

```
┌──(kaliza㉿kaliza)-[~]
└─$ sudo hping3 -S --flood -p 80 192.168.61.131
[sudo] password for kaliza:
HPING 192.168.61.131 (eth0 192.168.61.131): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.61.131 hping statistic ---
746286 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Figure 3.19: SYN Flood DOS Attack with –flood

Here Ubuntu got blocked for more than 20 sec, then snort' s console display severe lines like these ones, as exactly were configured in snort rules, and that is by TCP protocol.

```
.168.61.131:80
05/27-18:38:42.382829  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48725 -> 192
.168.61.131:80
05/27-18:38:42.382830  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48726 -> 192
.168.61.131:80
05/27-18:38:42.382831  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48727 -> 192
.168.61.131:80
05/27-18:38:42.382831  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48728 -> 192
.168.61.131:80
05/27-18:38:42.382831  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48729 -> 192
.168.61.131:80
05/27-18:38:42.382832  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48730 -> 192
.168.61.131:80
05/27-18:38:42.382832  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48731 -> 192
.168.61.131:80
05/27-18:38:42.382833  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48732 -> 192
.168.61.131:80
05/27-18:38:42.383406  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48733 -> 192
.168.61.131:80
05/27-18:38:42.383407  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48734 -> 192
.168.61.131:80
05/27-18:38:42.383409  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48735 -> 192
.168.61.131:80
05/27-18:38:42.383409  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48736 -> 192
.168.61.131:80
05/27-18:38:42.383410  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48737 -> 192
.168.61.131:80
05/27-18:38:42.383410  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:48738 -> 192
.168.61.131:80
05/27-18:38:42.383411  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:49209 -> 192
.168.61.131:80
05/27-18:38:42.383412  [**] [1:100007:0] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.61.129:49210 -> 192
```

Figure 3.20: Alerting SYN Flood DOS Attack

3.5.2.3 UDP Flood DoS Attack

We applied a UDP flood DOS attack with LOIC(low orbit ice common), and snort' s console displayed the configured message and the other informations, and that is by UDP protocol.

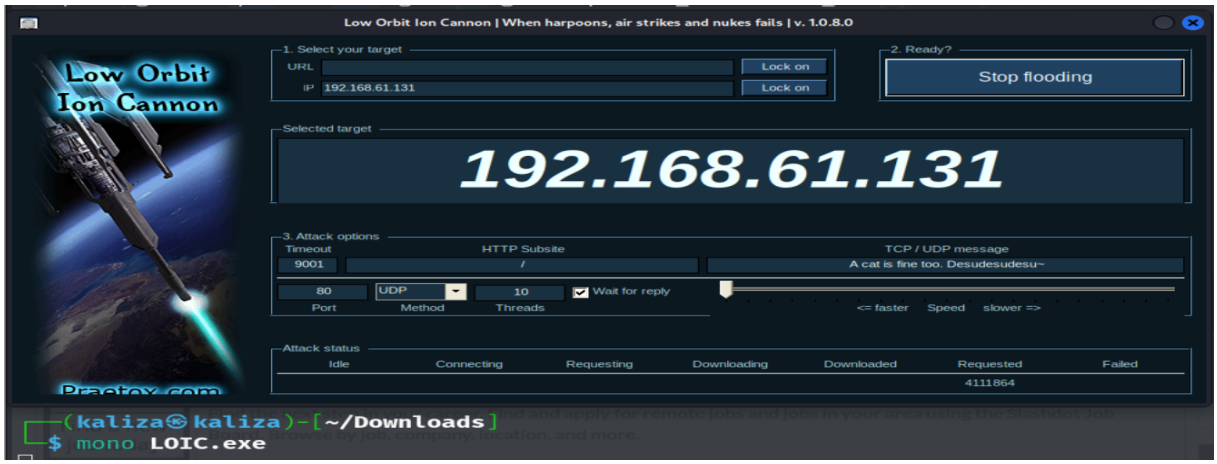

Figure 3.21: UDP Flood DoS attack

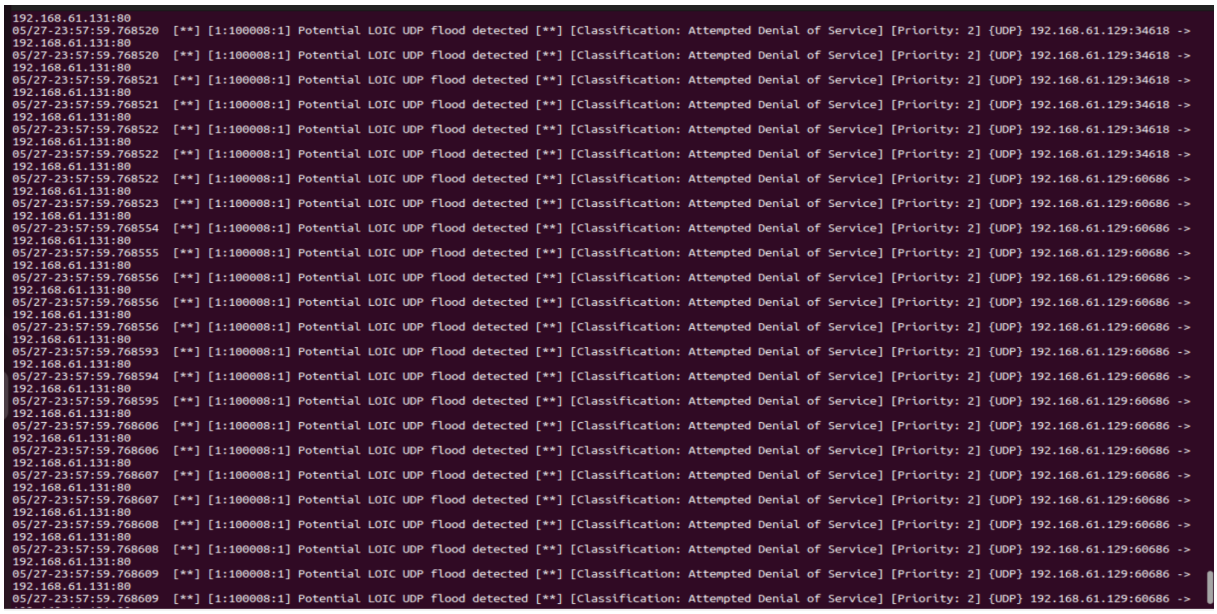

Figure 3.22: Alerting UDP Flood DoS attack

3.5.3 Spoofing

Spoofing is a technique where an attacker disguises themselves as a legitimate entity by falsifying data.

3.5.3.1 IP Spoofing (Using hping3 Tool)

This is the real IP address of the attacking machine.

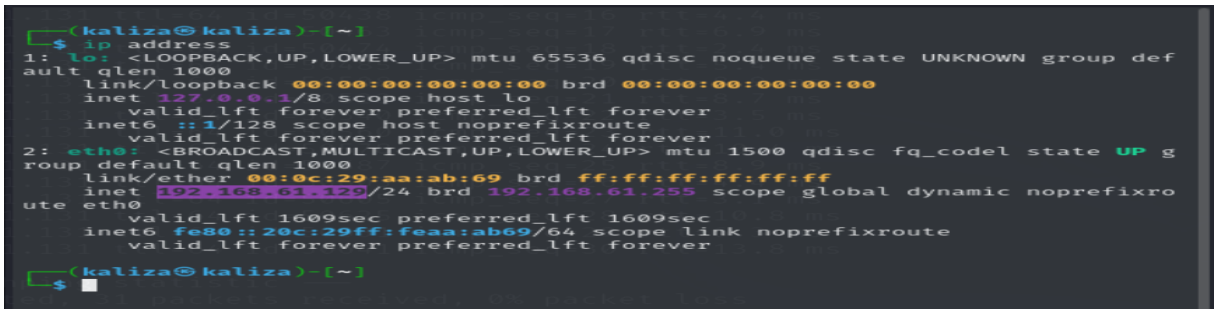

Figure 3.23: Real IP Address of the Attacking Machine

We applied an hping3 attack by spoofing the ip address of the attacking machine, and snort revealed the alert message as configured.

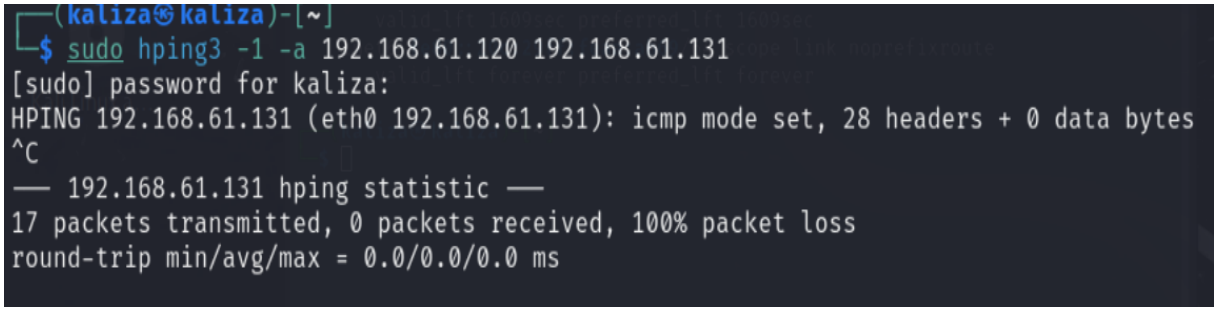

Figure 3.24: IP Spoofing hping3 Attack

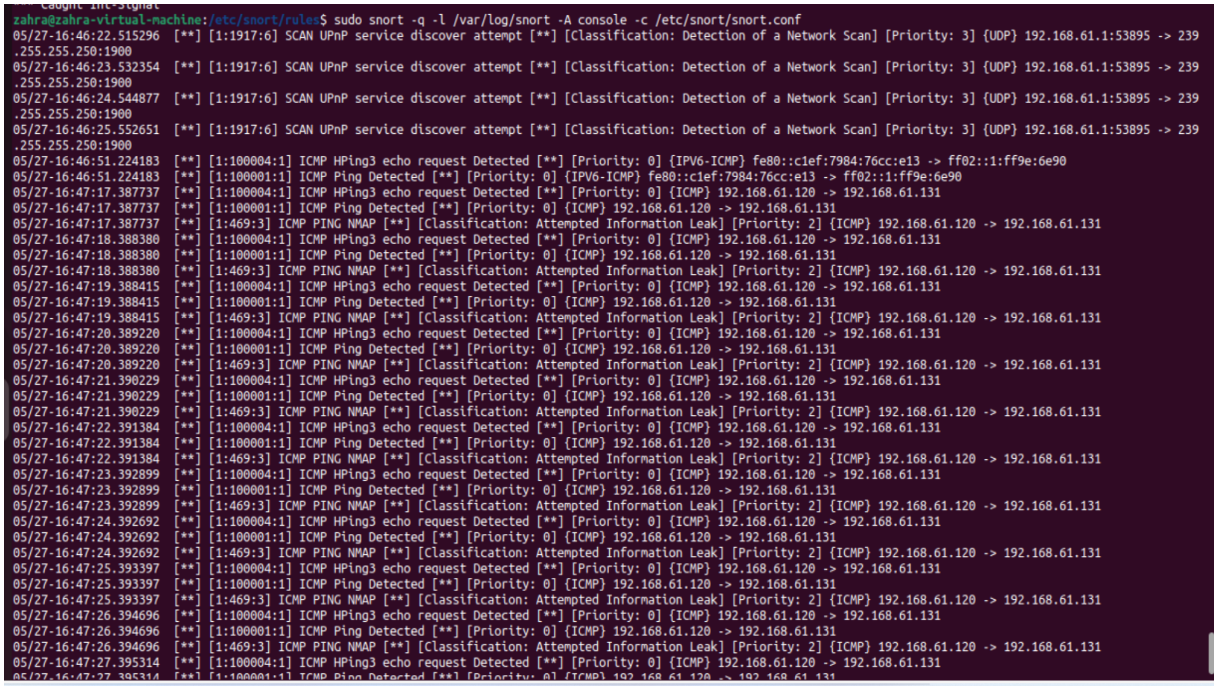

Figure 3.25: Alerting IP Spoofing hping3 Attack

We applied an hping3 attack by spoofing the ip address with a random source, and snort revealed the alert message as configured.

```
┌──(kaliza㉿kaliza)-[~]
└─$ sudo hping3 -1 --rand-source -c 1 192.168.61.131
HPING 192.168.61.131 (eth0 192.168.61.131): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.61.131 ttl=64 id=3679 icmp_seq=0 rtt=3.5 ms

--- 192.168.61.131 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 3.5/3.5/3.5 ms
```

Figure 3.26: Random IP Spoofing hping3 Attack

```
05/27-16:48:32.265067  [**] [1:100004:1] ICMP HPing3 echo request Detected [**] [Priority: 0] {ICMP} 37.42.207.134 -> 192.168.61.131
05/27-16:48:32.265067  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 37.42.207.134 -> 192.168.61.131
05/27-16:48:32.265067  [**] [1:469:3] ICMP PING NMAP [**] [Classification: Attempted Information Leak] [Priority: 2] {ICMP} 37.42.207.134 -> 192.168.61.131
05/27-16:48:32.265222  [**] [1:100004:1] ICMP HPing3 echo request Detected [**] [Priority: 0] {ICMP} 192.168.61.131 -> 37.42.207.134
05/27-16:48:32.265222  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.61.131 -> 37.42.207.134
05/27-16:48:41.062073  [**] [1:100004:1] ICMP HPing3 echo request Detected [**] [Priority: 0] {ICMP} 192.168.61.131 -> 192.168.61.2
05/27-16:48:41.062073  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.61.131 -> 192.168.61.2
05/27-16:48:53.229226  [**] [1:100004:1] ICMP HPing3 echo request Detected [**] [Priority: 0] {IPV6-ICMP} fe80::c1ef:7984:76cc:e13 -> ff02::1:ff9e:6e90
05/27-16:48:53.229226  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {IPV6-ICMP} fe80::c1ef:7984:76cc:e13 -> ff02::1:ff9e:6e90
05/27-16:49:56.728727  [**] [1:100004:1] ICMP HPing3 echo request Detected [**] [Priority: 0] {IPV6-ICMP} fe80::c1ef:7984:76cc:e13 -> ff02::1:ff9e:6e90
05/27-16:49:56.728727  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {IPV6-ICMP} fe80::c1ef:7984:76cc:e13 -> ff02::1:ff9e:6e90
```

Figure 3.27: Alerting the Random IP Spoofing hping3 Attack

3.6 Conclusion

This chapter has provided a detailed guide on implementing Snort IDS in a network environment. We began with the prerequisites and environment setup, focusing on necessary software dependencies and hardware considerations. The installation process on Ubuntu was covered, emphasizing its stability and security.

Configuration was a key focus, with detailed steps on setting up network interfaces, defining home and external networks, and customizing alert and log outputs. We also explored Snort' s rule sets, including writing, customizing, and integrating community-contributed rules for enhanced detection.

Performance optimization strategies were discussed to ensure Snort operates efficiently in high-traffic environments, managing resource utilization, and minimizing false positives.

By the end of this chapter, you should have the essential skills to implement Snort IDS, enhancing your network's security and gaining valuable insights into potential threats.

Chapter 4

Implementation of a Monitoring and Log Management System

4.1 Introduction

The implementation of a Monitoring and Log Management System is a fundamental component in maintaining the security and operational efficiency of a network. This chapter delves into the deployment and capabilities of our comprehensive monitoring and log management interface called **LogSpectrum** designed for the Snort Intrusion Detection System (IDS). Our system not only enables real-time monitoring of network activities and immediate threat detection but also ensures efficient log collection, storage, and analysis. We will explore the interface design, the integration of log files, data visualization techniques, and the application of artificial intelligence (AI) for advanced threat analysis. Additionally, this chapter covers the implementation of a robust user authentication system to ensure secure access to the monitoring platform.

4.2 Technoligies Used

The interface utilizes a modern web technology stack. The frontend is built using HTML, CSS, and JavaScript, providing a responsive and interactive user interface. For the backend, the Node.js framework is employed, offering a robust and scalable server-side environment. This combination allows for efficient data processing, seamless API integration, and real-time communication between the client and server, enhancing the overall performance and user experience of the network security application.

4.3 Related UML Diagrams

4.3.1 Use Case

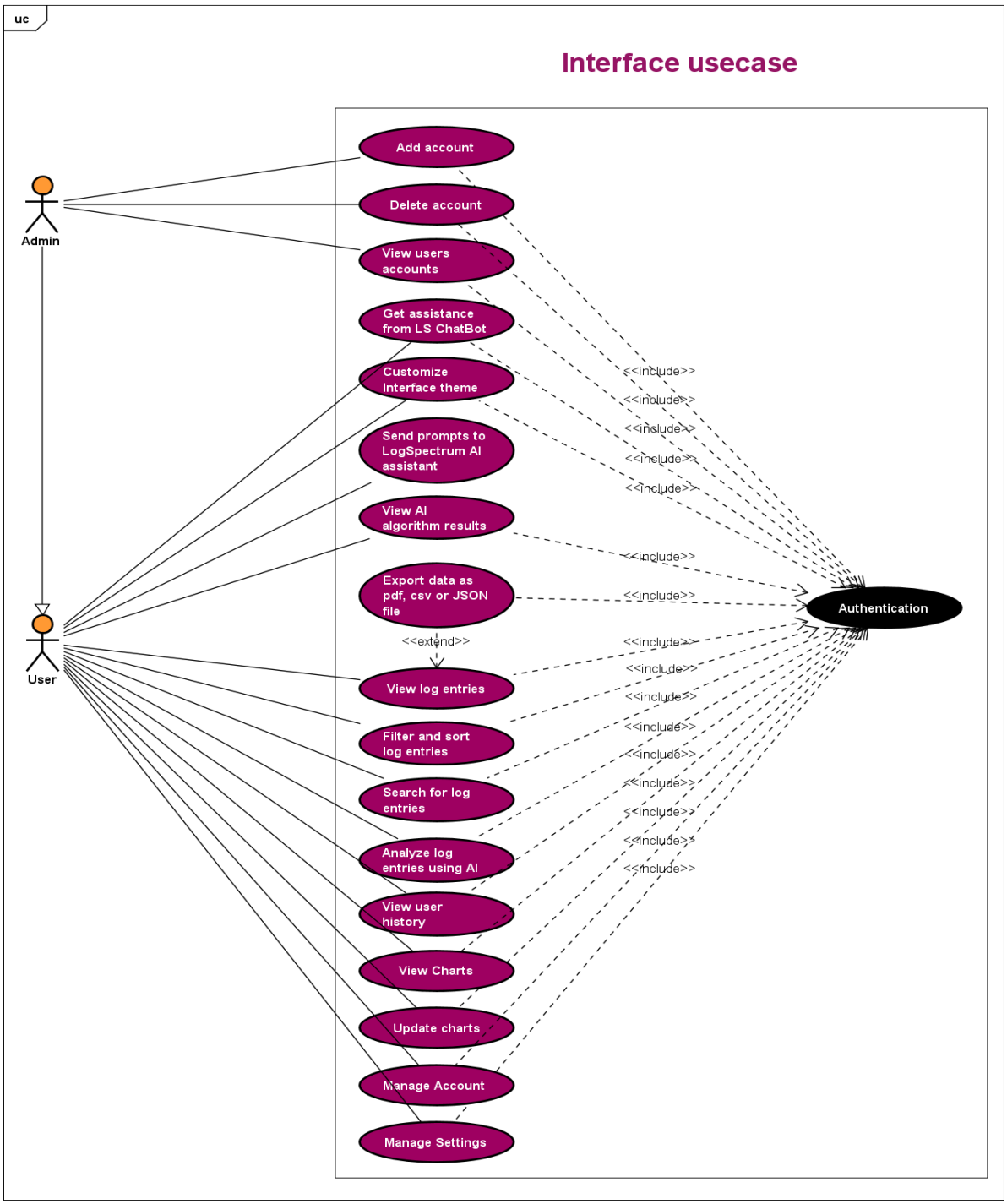

Figure 4.1: Use Case Diagram

4.3.2 Deployment Case

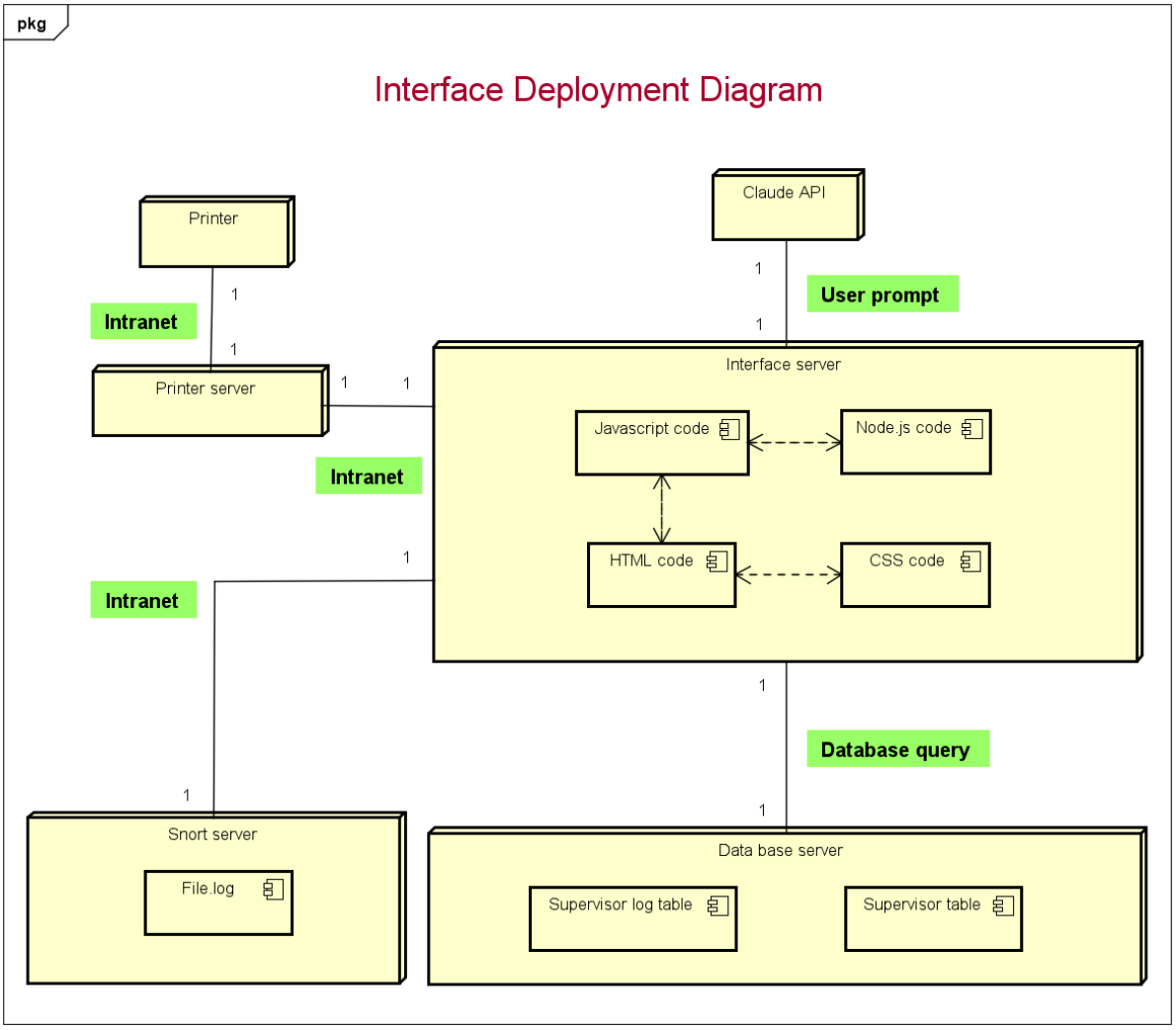

Figure 4.2: Deployment Diagram

4.4 Interface Design

4.4.1 User Interface Overview

The LogSpectrum monitoring interface features a dark theme with a striking purple and blue color scheme, centered around a unique flower-like logo. This sleek design enhances visibility and reduces eye strain during extended use. The interface comprises four main pages: Dashboard, Log, Report, and Help. The Dashboard presents nine visual charts for data representation. The Log page offers detailed entry information with control features, while the Report page displays AI algorithm results and logs with unusual structures. The Help page provides AI-driven assistance for network security queries. This intuitive layout ensures easy navigation for users of all experience levels, effectively balancing functionality with aesthetic appeal.

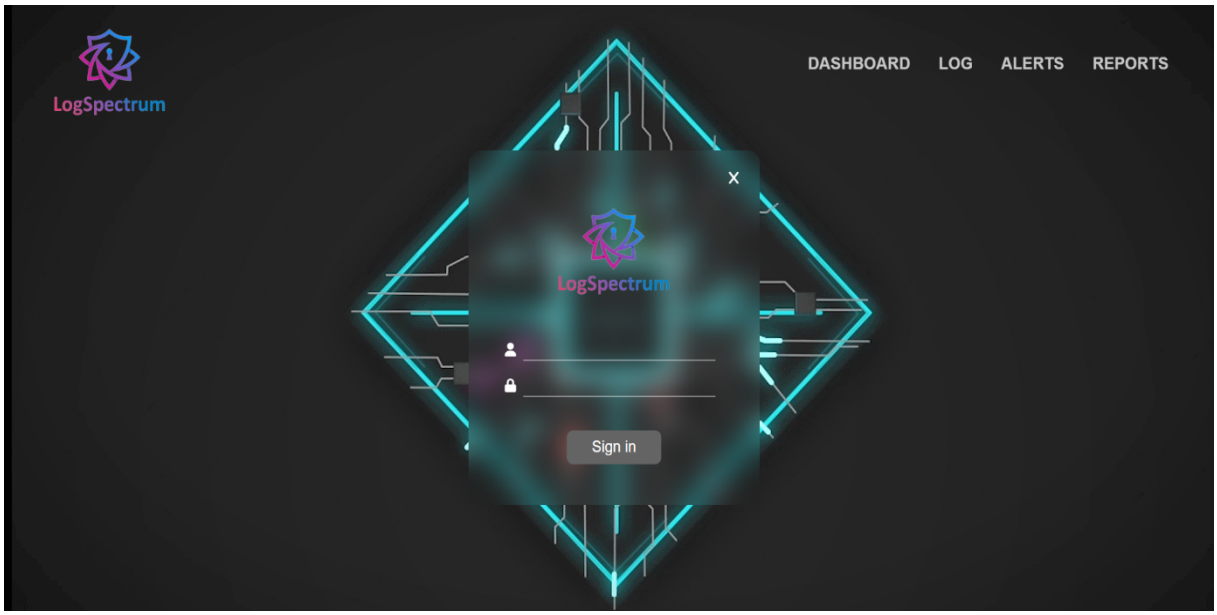

Figure 4.3: Login Page

4.4.1.1 Dashboard Page

The Dashboard page serves as the central hub for monitoring and visualizing the IDS' s activity. The key features of the Dashboard page include Graphical Representations that provides real-time visualizations of various network intrusions detected by the system. The charts display the severity, the type, protocols, number of intrusions over time, ports, ip addresses and classifacation, allowing for quick assessment of network security status.

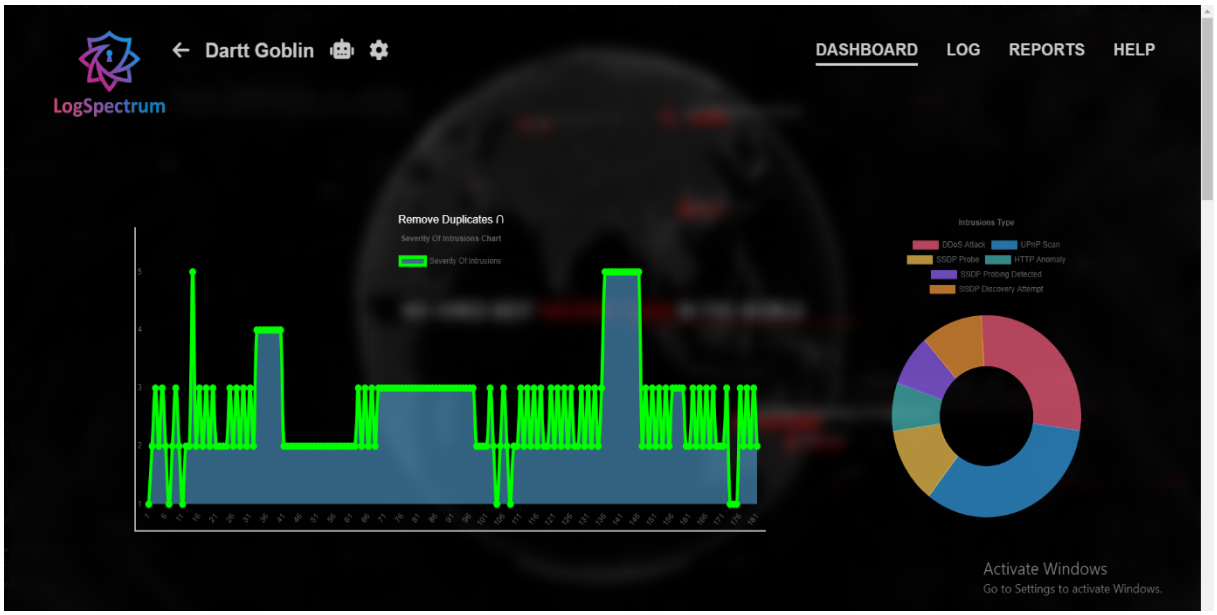

Figure 4.4: Severity and Intrusion Type Charts

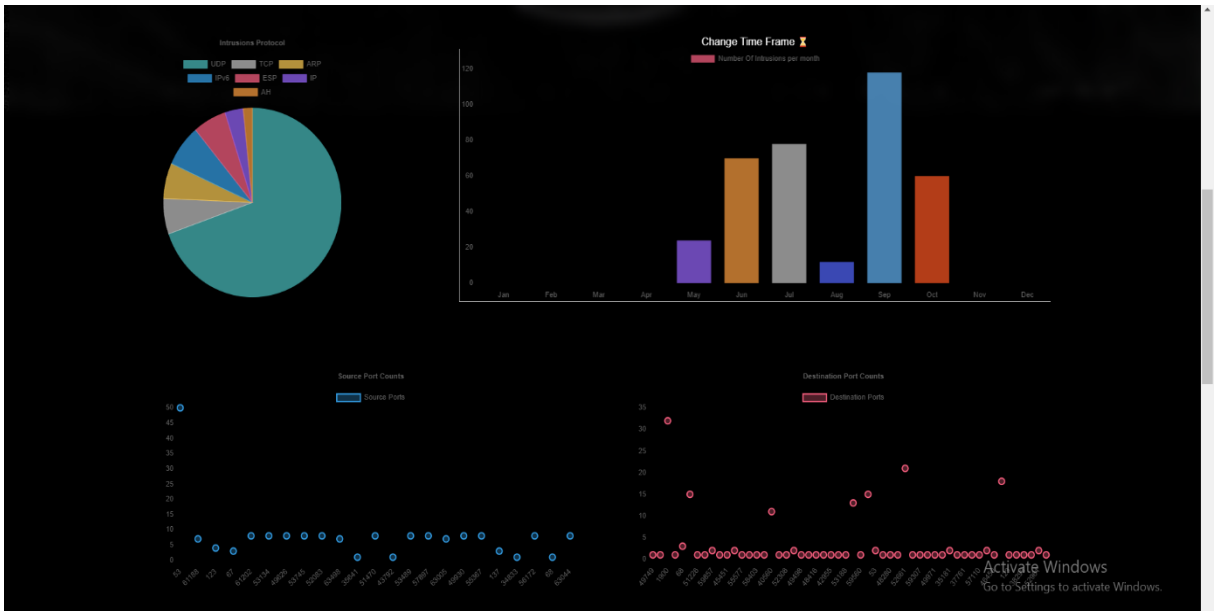

Figure 4.5: Protocol, Number of Intrusions, Source and Destination Ports Charts.

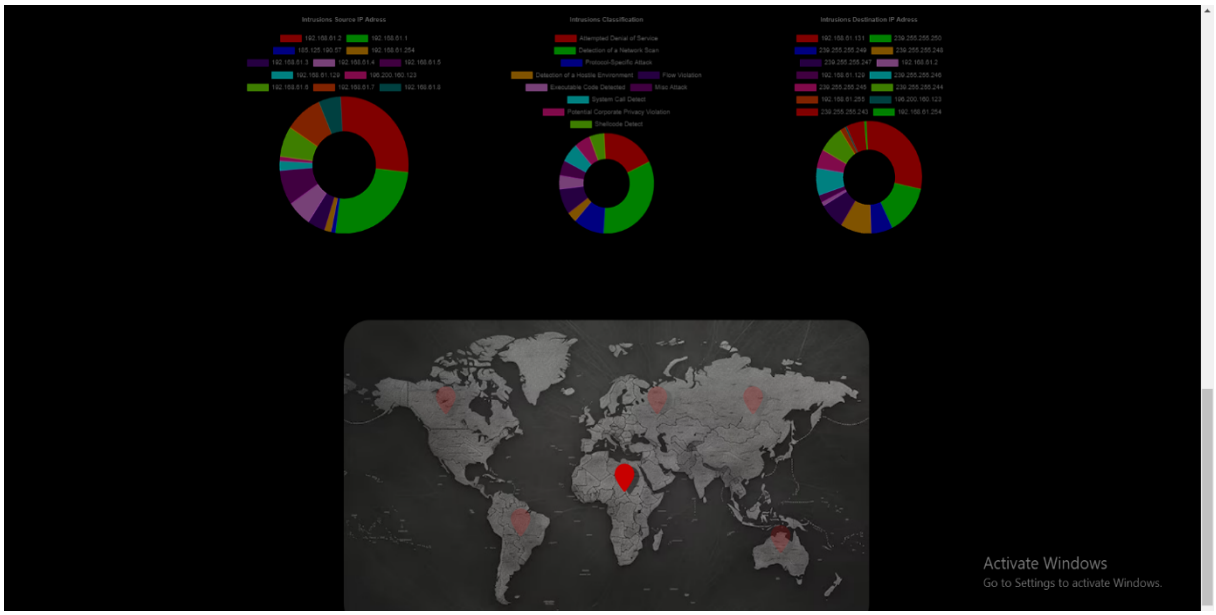

Figure 4.6: Source IP Address, Classification and destination IP Address Charts.

4.4.1.2 Log Page

The Log page is designed to present detailed records of all detected intrusion events. It offers comprehensive features for analyzing and managing these logs, including an event list that displays all detected events in a table format with columns for event description, timestamp, classification, priority, protocol, source IP, source port, destination IP, and destination port. Users can filter the events based on the timestamp and sort the logs by various criteria such as

event type, priority, and classification. Additionally, a search bar enables users to quickly find specific logs based on keywords or other parameters. An analyze button that provides advanced analysis for logs entries, using an AI algorithm, the isolation forest, for deeper insights.

- **Introduction to Artificial Intelligence**

  Artificial Intelligence (AI) is a branch of computer science that aims to create systems capable of performing tasks that typically require human intelligence. These tasks include learning, reasoning, problem-solving, perception, and understanding natural language. AI involves the development of algorithms and models that enable machines to process data, recognize patterns, and make decisions based on that information. The goal of AI is to build systems that can function autonomously and improve their performance over time, ultimately enhancing efficiency and innovation across various domains.

- **Isolation Forest Algorithm**

  - **Concept**

    The core idea behind the Isolation Forest algorithm is that anomalies are few and different, making them easier to isolate. Anomalies can be isolated by partitioning the data. Since anomalies are rare and usually distinct from the majority of the data, they tend to be isolated quickly. In contrast, normal data points require more partitions to be isolated.

  - **How It Works**

    1. *Random Sampling:* The algorithm starts by randomly sampling the dataset to create a sub-sample. This helps in managing large datasets and speeds up the process.
    2. *Building Trees:* The algorithm constructs multiple binary trees (iTrees) using the sub-samples. Each tree is built by recursively partitioning the data.
       * For each partition, a random feature is selected.
       * A random split value between the minimum and maximum values of the selected feature is chosen to divide the data.
    3. *Isolation Path Length:* The process of partitioning continues until:
       * The data points are isolated (each point is alone in a partition), or
       * A predefined tree height is reached.
    4. *Anomaly Scoring:* The length of the path from the root of the tree to the point of isolation is recorded. Anomalies, being isolated quickly, tend to have shorter path lengths.

- o The average path length over all trees is used to compute an anomaly score.
- o The score is normalized to ensure it lies between 0 and 1.

5. *Thresholding:*Based on the anomaly scores, a threshold can be set to classify data points as normal or anomalous. Typically, a lower score indicates an anomaly.

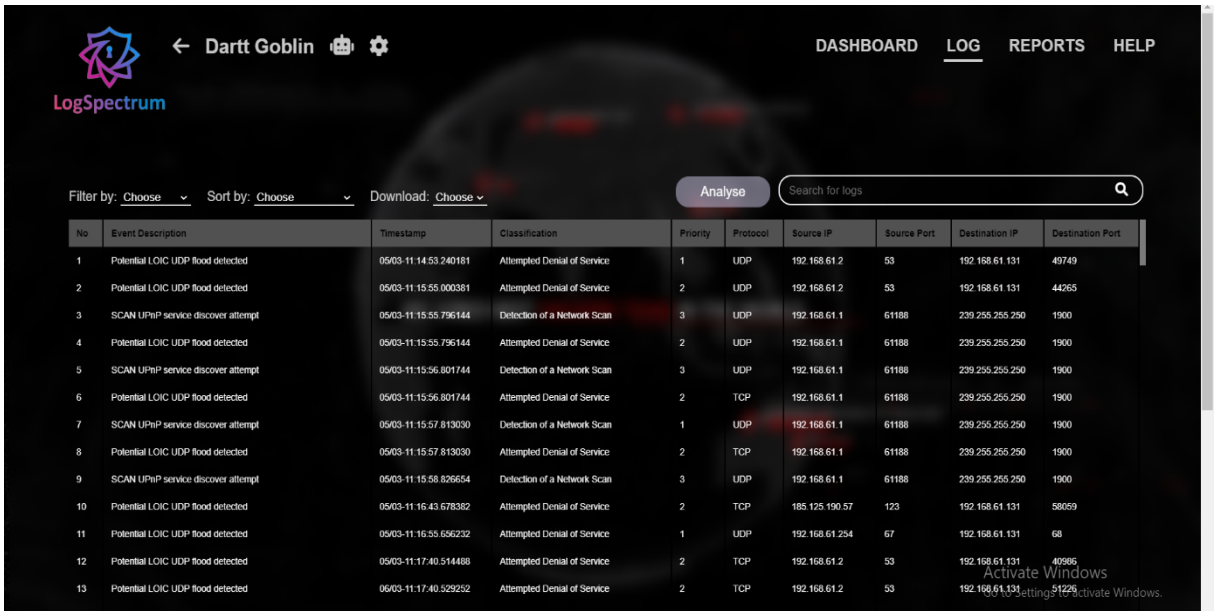

Figure 4.7: Log Entries

4.4.1.3 Reports Page

This page displays network intrusions identified as anomalies by the Isolation Forest AI algorithm. It presents two key sections: a table of standard anomalies and a list of intrusions with abnormal structures. The table shows details like timestamp, source IP, and severity, while the full-line entries below represent logs with unusual patterns. This report page provides a comprehensive view of potential security threats, allowing analysts to quickly assess and investigate suspicious network activities detected by the AI system.

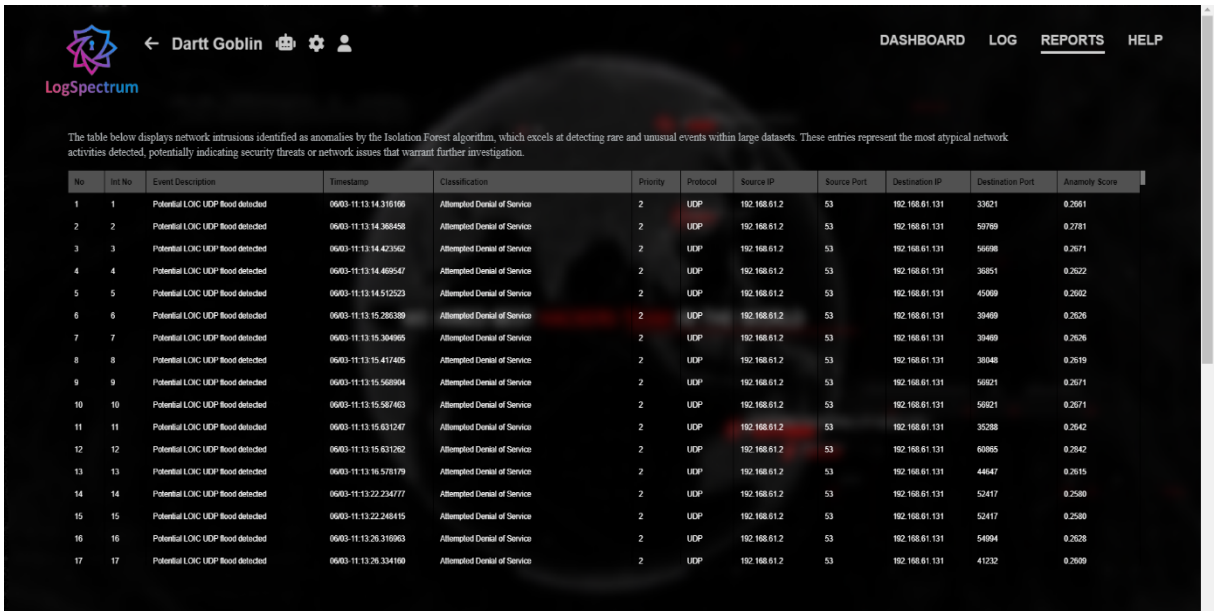

Figure 4.8: Anomaly Intrusions (AI Algorithm Results)

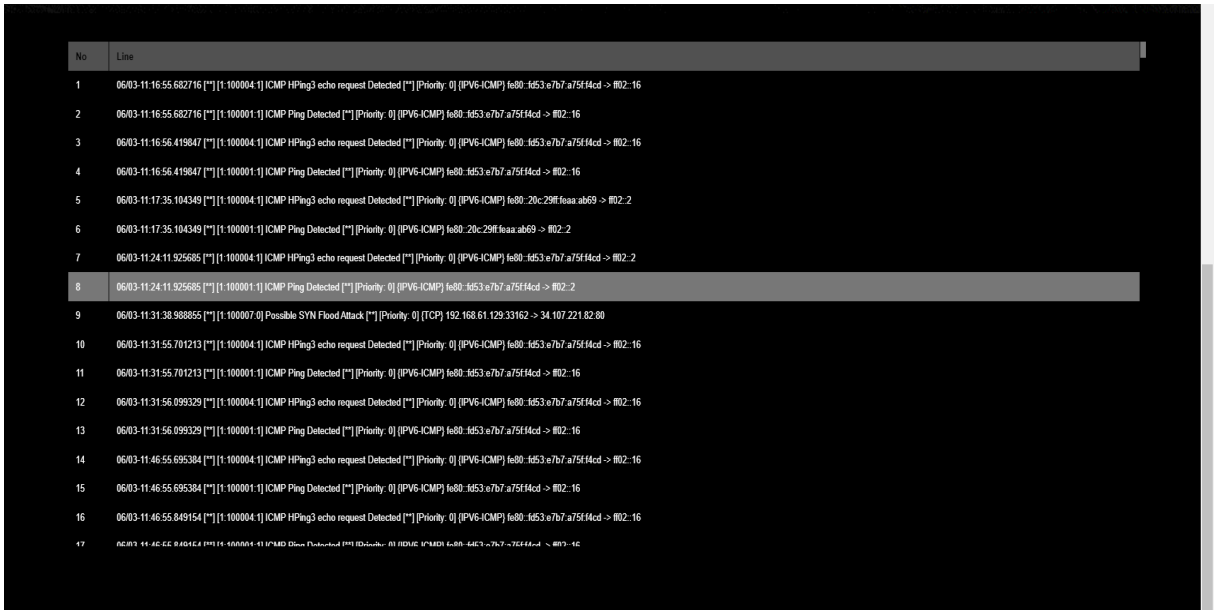

Figure 4.9: Abnormal Intrusions

4.4.1.4 Help Page

The Help page provides an AI-powered assistant specialized in network and cybersecurity. It offers quick access to frequently asked questions about Snort configuration and best practices. Users can also input custom queries to get expert responses on various network security topics. This functionality combines pre-set information with on-demand assistance, helping users quickly find solutions to their security-related questions and challenges, using the Claude API [4] using

the its documentaion [5] in addition of StackOverFlow [6].

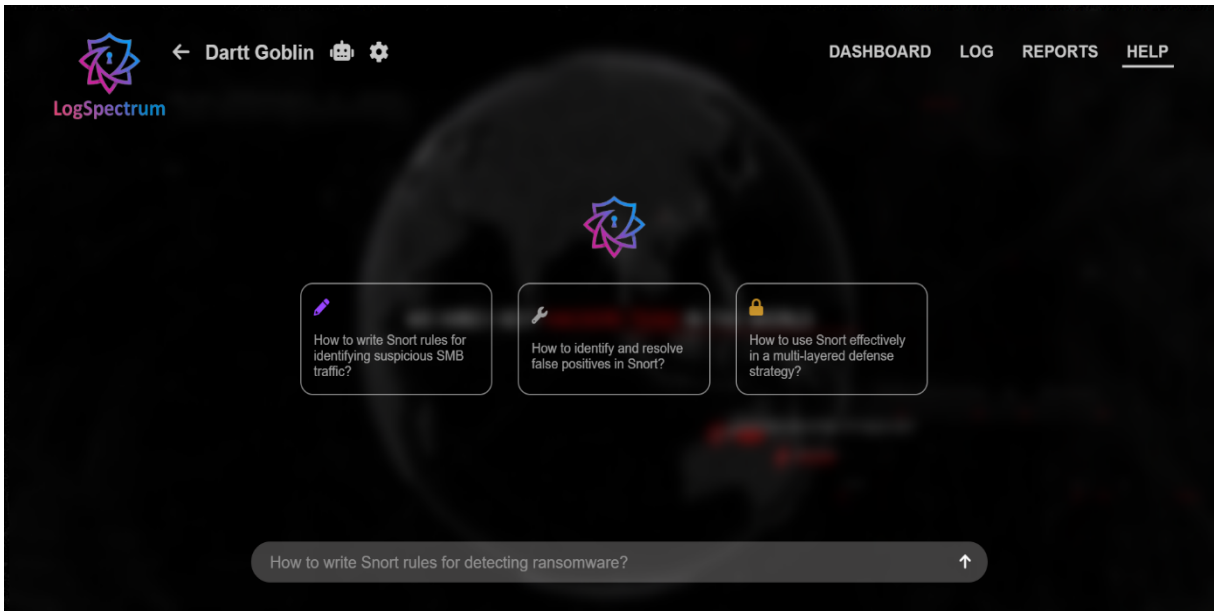

Figure 4.10: LogSpectrum AI Assistance and FAQ

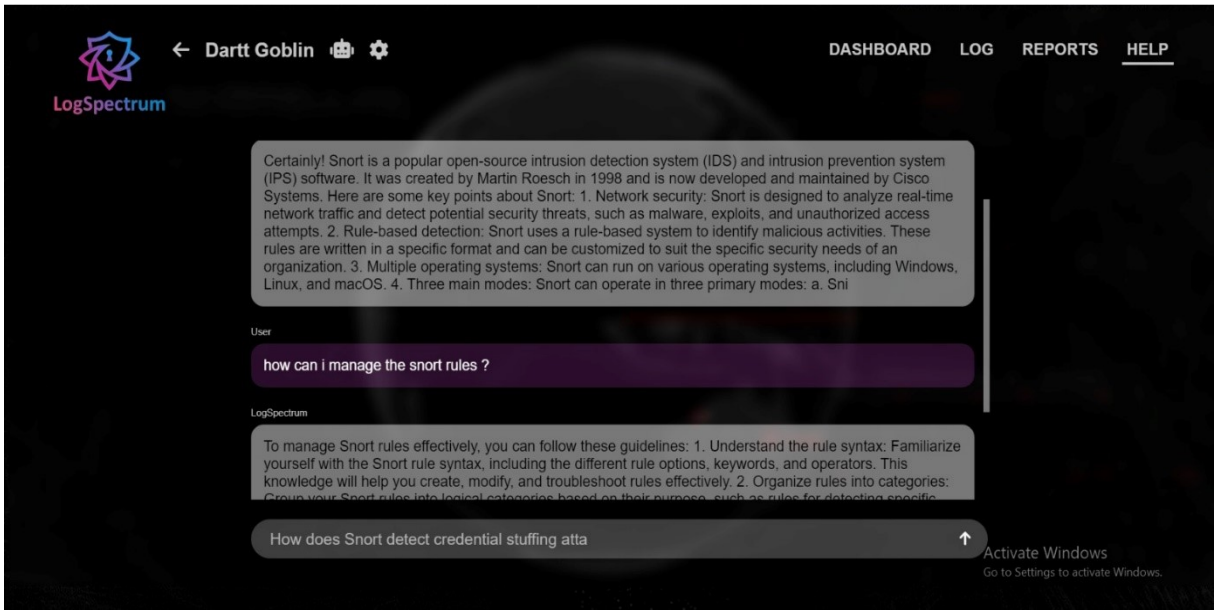

Figure 4.11: LogSpectrum AI Assistance Conversation

4.4.2 Conclusion

By structuring the interface in this way, we ensure that users have easy access to critical information and tools necessary for effective network security management. The visual descriptions, log management capabilities, reporting features, and help resources collectively contribute to a robust intrusion detection and prevention system.

General Conclusion

In conclusion, the development of the Intrusion Detection and Prevention System (IDPS) presented in this project underscores the critical importance of robust cybersecurity measures in safeguarding network infrastructures. By meticulously researching the various facets of cybersecurity, including common attack vectors and effective defense strategies.

The project successfully combined open-source tools with custom-developed scripts to create a dynamic and responsive system capable of identifying and mitigating cyber threats in real-time. Our implementation of a Demilitarized Zone (DMZ) provided an additional layer of security, further enhancing the system's ability to protect sensitive data and services.

Throughout the project, we encountered and overcame significant challenges related to system integration, detection rule configuration, and response mechanism optimization. These challenges were addressed through persistent effort and iterative testing, resulting in a functional and efficient IDPS.

Our rigorous testing in a simulated environment demonstrated the system's efficacy in detecting and responding to a variety of intrusions, including Distributed Denial of Service (DDoS) attacks, malware infections, and unauthorized access attempts. The project's outcomes not only validate the system's capability to enhance network security but also pave the way for future advancements in intrusion detection and prevention technologies.

Ultimately, this project contributes to the ongoing efforts to protect organizational networks against evolving cyber threats, emphasizing the need for continuous improvement and innovation in the field of cybersecurity.

Bibliography

[1] Qu'est-ce que la cybersécurité ? https://www.kaspersky.fr/resource-center/definitions/what-is-cyber-security, December 2023.

[2] The Fundamentals of Cyber Security | Online | University Of Adelaide. https://online.adelaide.edu.au/blog/cyber-security-fundamentals.

[3] Aliesana Sandoval. 7 Cybersecurity Defenses to Prepare for Cyber Warfare, November 2021.

[4] Anthropic Console. https://console.anthropic.com/settings/keys.

[5] Welcome to Claude. https://docs.anthropic.com/en/docs/welcome.

[6] Stack Overflow - Where Developers Learn, Share, & Build Careers. https://stackoverflow.com/.