ENSEMBLE DEFENSE SYSTEM: COMBINING SIGNATURE-BASED AND BEHAVIORAL-BASED INTRUSION DETECTION TOOLS

by

Sarah Alharbi

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Cybersecurity

Summer 2023

ENSEMBLE DEFENSE SYSTEM: COMBINING SIGNATURE-BASED AND BEHAVIORAL-BASED INTRUSION DETECTION TOOLS

by

Sarah Alharbi

Approved: ____
Michael De Lucia, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: ____
Jamie D. Phillips, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: ____
Levi T. Thompson, Ph.D.
Dean of the College of Engineering

Approved: ____
Louis F. Rossi, Ph.D.
Vice Provost for Graduate and Professional Education and
Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my late advisor Dr. Chase Cotton for his invaluable guidance, support, and mentorship throughout the early stages of this research. His expertise, unwavering belief in my capabilities, and constant support significantly shaped this thesis. I am genuinely grateful for his mentorship, which I will always remember.

I would like to extend my appreciation to my current advisor Dr. Michael De Lucia for his continued support and guidance throughout the completion of this thesis. His insights and feedback have been invaluable in enhancing my research.

I would like to express my heartfelt gratitude to my husband, Mohammed, and my son, Yousef, for their constant encouragement, understanding, and motivation. Their support has given me the motivation and strength to overcome challenges and pursue my academic goals.

Lastly, I would also like to thank my family and friends for their support and belief in my abilities. Their encouragement and understanding have helped me during this academic journey.

TABLE OF CONTENTS

Chapter

Appendix

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Cyber attacks are becoming increasingly sophisticated, which poses significant challenges for organizations in detecting and preventing these attacks. Implementing robust defense mechanisms that can detect, prevent, and respond to these threats and attacks is crucial. In this thesis, we design, develop, and evaluate a novel Ensemble Defense System (EDS), addressing the critical need for advanced defense systems. The EDS combines the capabilities of Intrusion Detection Systems (IDS) and Security Information and Event Management (SIEM) to provide an effective defense against cyber threats. The EDS incorporates hybrid-based IDS technologies, leveraging the strengths of signature-based IDS tools like Zeek and Suricata and behavioral-based IDS tools like Slips. By utilizing hybrid-based IDS, the EDS provides a more effective system for countering cyber threats. Moreover, the EDS integrates open-source SIEM, specifically Elasticsearch, to provide data management and analysis capabilities and create user-friendly visualization. The effectiveness of the EDS has been evaluated through a designed bash script that performs several attacks, such as port scanning, privilege escalation, and Denial-of-Service (DoS). This research contributes to better cybersecurity by introducing an EDS that can detect various cyber attacks.

Chapter 1

INTRODUCTION

Organizations face increasing threats from cybercriminals who develop sophisticated attack techniques, targeting vulnerabilities in network systems and compromising sensitive data. Consequently, robust and comprehensive defense mechanisms have become essential in ensuring organizations' security in the face of these threats. One such defense mechanism, Network Intrusion Detection Systems (NIDS), has long been recognized as a crucial component of network defense strategy. NIDS systems analyze network traffic to identify potential security breaches and malicious activities [1].

There are three approaches for the NIDS: signature-based IDS, anomaly-based IDS, and hybrid-based IDS. The first approach is the signature-based IDS that identifies known attack patterns and malicious signatures to generate alerts or take preventive actions [2]. This approach often struggles to keep up with the rapidly evolving threat due to their dependence on only signatures. Moreover, the emergence of sophisticated attack techniques has caused signature-based approaches to be less effective in detecting unknown threats. Consequently, organizations increasingly seek advanced defense systems that can effectively identify and respond to these cyber attacks.

Another approach that organizations use to bolster their security is using anomaly-based IDS. This approach analyzes network traffic for deviations from normal behavior [3]. By establishing a normal baseline, anomaly-based IDS can detect and alert any suspicious activities that may harm organization security. Thus, anomaly-based IDS offers advantages in detecting previously unknown threats.

The last approach is a hybrid-based IDS that combines signature-based and anomaly-based IDS techniques. This approach can enhance security defense systems by combining the strengths of both approaches. The signature-based approach allows for the detection of known attack signatures, while the anomaly-based approach enables the identification of unknown threats by detecting abnormal network behavior. This combination of techniques provides a more comprehensive and robust defense against cyber threats. Thus, this research employs the implementation of hybrid-based IDS within an EDS.

1.1 Motivation

The motivation behind this thesis originates from three primary motivations. Firstly, there is a critical need for advanced and comprehensive defense mechanisms to counter cyber threats. Increasing cyber threats necessitate developing and implementing robust security systems to detect and mitigate these threats effectively. Secondly, evaluating the effectiveness of integrating multiple detection technologies within an EDS is essential. In particular, the combination of signature-based IDS and behavioral-based IDS has the potential to enhance overall security postures. The signature-based IDS utilize predefined patterns to detect known attacks, while behavioral-based IDS analyzes network behavior to detect anomalies and potential attacks. Lastly, increasing cyber attacks highlight the urgency to enhance the detection capability for different types of attacks. The EDS developed in this research will be specifically designed to address several attacks and evaluate its effectiveness in detecting such threats.

1.2 Aims and Objectives

The thesis aims to design, develop, and evaluate an EDS that combines the capabilities of IDS and SIEM to enhance the detection of cyber attacks. The EDS, put forth in this study, will integrate multiple detection technologies, including signature-based IDS and behavioral-based IDS, to detect various attacks. Moreover, the research will examine the overall effectiveness of EDS in improving the target system's security posture by evaluating its ability to detect attacks such as port scanning, privilege escalation, and DoS. The primary objectives of this research are:

1. To examine the effectiveness of different detection technologies within the EDS, including signature and behavioral-based IDS.
2. To assess the capabilities of IDS and SIEM in detecting various types of cyber attacks.
3. To evaluate the performance of the EDS using a designed bash script that performs specific attacks.
4. To evaluate the correlation between the IDS logs to explore more related information regarding the generated alerts.
5. To evaluate the EDS's capability to visualize and analyze attack data using Kibana Query Language (KQL) queries.
6. To identify the areas for improvement of the EDS to provide recommendations for enhancing its effectiveness and efficiency.

These aims and objectives contribute to the existing knowledge in cybersecurity and provide practical guidance for system administrators seeking to improve their defenses. This thesisâĂŹ findings and recommendations will also allow system administrators to make informed decisions regarding implementing an EDS and strengthening their cybersecurity defenses.

1.3 Thesis Outline

The remaining chapters of this thesis discuss the following topics: Chapter 2 focuses on related work, comprehensively examining previous research relevant to the thesis topic. Chapter 3 provides a background of an in-depth discussion of the most commonly used terms and concepts relevant to the research. Chapter 4 outlines the methodology employed in this research, including a detailed description of the research configuration and setup. Chapter 5 focuses on the evaluation and analysis of the implemented EDS. Finally, Chapter 6 concludes the thesis by summarizing the key contributions and discussing future research directions.

Chapter 2

RELATED WORK

This chapter overviews relevant research papers that focus on integrating Intrusion Detection Systems (IDS) and Security Information and Event Management (SIEM). The primary objective of this section is to examine the similarities and differences between the proposed EDS in this research and the approaches adopted in previous studies. This chapter also analyzes the methodology employed to conduct the tests in each study. By analyzing the existing literature, we aim to identify the unique contributions of the thesis in comparison to related research in the field.

2.1 Integrating IDS and SIEM: A Review of Related Work

El Arass and Souissi [4] focus on addressing the Big Data challenges in SIEM and introduce a novel solution called the Smart SIEM that integrates with other IDS and load-balancing tools. The authors utilize Elasticsearch [5], Logstash [6], and Kibana [7] (ELK), which are used for log management and analysis [5]. In addition to ELK, the authors also chose several open-source IDS, namely Snort [8], Zeek [9], and OSSEC [10], to ensure the detection of security incidents [4]. The chosen tools play a crucial role in the proposed system. Snort is an IDS that uses signature-based detection to identify known attack patterns [8]. Zeek is a network analysis framework that provides high-level network traffic information [9]. OSSEC, on the other hand, is a host-based intrusion detection system that monitors system logs for signs of compromise [10]. To evaluate the effectiveness of their system, the authors conduct tests using different attack scenarios performed by the Kali machine [4]. The attacks include Structured Query Language injection (SQLi), brute force, and Distributed Denial-of-Service (DDoS) [4]. The authors assess the system's ability to detect and respond to various security incidents by subjecting the system to these attacks [4]. The results obtained from these tests validate the proposed system for detecting security incidents [4].

The study by Negoita and Carabas [11] focuses on using Elasticsearch to enhance security by integrating IDS and Machine Learning (ML) techniques for attack detection. Elasticsearch offers a built-in machine-learning framework that enables the training and deployment of machine-learning models [12]. The authors also incorporate the Snort tool as an IDS in their research [11]. The study's findings indicate that while Elasticsearch's built-in machine-learning jobs can effectively identify basic vulnerabilities or anomalies within the cluster, they have certain limitations [11]. One such limitation is the requirement for manual configuration, which adds complexity to the setup process [11]. Furthermore, these built-in ML jobs may not accurately detect more sophisticated and complex attacks [11].

Priambodo et al. [13] introduce an approach to strengthen the security of work-from-home networks. The authors propose the integration of Wireguard [14], which is a secure and efficient VPN protocol, Suricata [15], which is an open-source Intrusion Detection and Prevention System (IDS/IPS), and ELK [13]. To evaluate the system's effectiveness in detecting port scanning attacks, the authors conduct a test using Nmap [16] scanning for ten open ports [13]. This test allows them to validate the system's ability to detect and generate alerts from Suricata into Kibana [13].

Esseghira et al. [17] introduce a new open-source security platform, AKER. The platform combines the capabilities of IDS and SIEM systems as well as the capability of analyzing encrypted network traffic [17]. The authors chose Suricata and Zeek as IDS solutions and Elasticsearch as the SIEM system [17]. Esseghira et al. highlight the increasing prevalence of encrypted traffic in modern network environments [17]. To tackle encrypted traffic analysis, AKER utilizes a decision-tree-based approach implemented in a Python script as part of the threat investigation module [17]. The evaluation of AKER is conducted using a malware dataset and User Acceptance Tests (UAT) [17]. The results demonstrate the platform's effectiveness in detecting malware hidden within encrypted traffic [17].

Muhammad et al. [18] present a research proposal to integrate a SIEM using the ELK stack, an IDS utilizing Zeek, and live analysis machine learning with Slips [19]. The researchers suggest sending Zeek logs, specifically conn.log, to Slips for ML analysis in their proposed system [18]. Once the analysis is completed, the alerts generated by Slips are forwarded to the ELK stack [18]. To evaluate the system's performance, the researchers choose a DoS attack as a representative cyber attack [18]. They assessed their system's effectiveness in detecting such threats [18]. Additionally, Muhammad et al. measure the CPU and RAM usage at each system stage [18].

The existing literature has primarily focused on integrating different open-source tools IDS with SIEM. These studies used Zeek or Suricata as IDS and Elasticsearch as the SIEM. Specifically, studies such as [4], [17], and [18] have utilized Zeek as an IDS, while studies like [11], [13], and [17] have employed Suricata.

2.2 Gaps in Existing Approaches

A gap remains in the existing literature regarding using hybrid-based IDS that combine signature-based and anomaly-based approaches within the SIEM. Notably, a study [4] utilized Elasticsearch's built-in ML capabilities, while a study [18] used Slips for live analysis. Therefore, previous research indicates that no research currently proposes integrating Suricata and Zeek as signature-based IDS, Slips as behavioral-based IDS, and Elasticsearch as the SIEM platform within one system. This research gap presents an opportunity to investigate the potential benefits and effectiveness of combining these specific components within an EDS.

The motivation for using the hybrid-based IDS technique in the EDS is because there are multiple research studies [20] [21] [22] [23] have suggested that a hybrid-based IDS can achieve high detection rates while keeping false positives at a low level. Thus, the EDS proposed in this study will combine the strengths of both approaches to achieve greater accuracy in detecting potential threats.

In light of these findings, this current research addresses these gaps by developing and evaluating an EDS that combines Suricata, Zeek, and Slips with Elasticsearch. The integration of these components will allow for an advanced detection capability, leveraging both signature-based and anomaly-based approaches.

Chapter 3

BACKGROUND

This chapter provides a background on Intrusion Detection Systems (IDS), including their types and approaches. It discusses the two main types of IDS, namely Host-based Intrusion Detection Systems (HIDS) and Network-based Intrusion Detection Systems (NIDS), and compares their benefits and drawbacks. This chapter also examines the three detection methodologies employed by IDS solutions: signature-based, anomaly-based, and hybrid-based. The chapter briefly overviews IDS tools such as Suricata, Zeek, and Stratosphere Linux IPS (Slips). It will also discuss the Security Information and Events Management (SIEM) background and a brief overview of the existing SIEM solution, Elasticsearch.

3.1 Intrusion Detection and Prevention Systems (IDS/IPS)

IDS/IPS are network security technologies designed to protect computer networks from malicious activities. IDS and IPS are often used to provide comprehensive security coverage for the network.

An IDS is a security system designed to monitor and detect any suspicious or malicious activity on a network. It is composed of hardware, software, or a combination of both, and its primary purpose is to detect any malicious activity that may not be detected by other security measures, such as firewalls and antivirus software [24]. The IDS is intended to provide a comprehensive security solution by monitoring a network's activity, detecting any suspicious activity, and generating an alert to security administrators.

An IPS is a security technology that goes beyond detecting malicious activity and takes immediate action to prevent the threat from progressing further [25]. An IPS operates by examining network traffic in real-time and blocking traffic that matches known attack signatures or anomalous behavior.

Although IDS and IPS are often used together to provide comprehensive network security, they differ in functionality and deployment. An IDS has been designed primarily to detect security threats and generate alerts to security administrators. In contrast, an IPS is designed to actively block or modify network traffic to prevent the progression of security threats. Thus, IDS can operate as a passive system that does not impact network traffic, while IPS is an active system that can alter network traffic by blocking it.

3.1.1 Intrusion Detection System Types

IDS can be divided into two main categories: HIDS and NIDS, as shown in Figure 3.1.

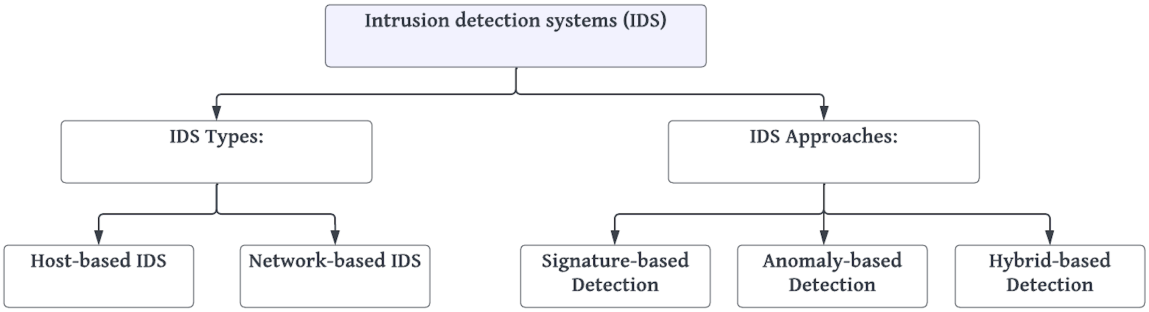


Figure 3.1: IDS Types and Approaches

3.1.1.1 Host-based Intrusion Detection Systems

HIDS monitor the activities of individual hosts and detect malicious activity by analyzing the system logs and other system-level activities [26]. HIDS helps identify any suspicious activity that may be occurring on the system, such as unauthorized access or data manipulation. As part of this process, it continuously monitors events, system, and application logs and creates a baseline for comparison. If new log entries appear and malicious activity is detected, the HIDS will trigger an alert and take action according to the predefined policy.

3.1.1.2 Network-based Intrusion Detection Systems

NIDS are widely known as a fundamental element of any organization's security infrastructure, providing a critical component of a comprehensive security strategy for network devices. NIDS are essential for detecting and notifying administrators of malicious activity on a network, thus making them an essential component of any organization's security posture. NIDS monitor network traffic and identify malicious activity by analyzing data packets flowing across the network [1] [27]. Using NIDS assists in recognizing malicious activities and protecting the network from potential threats.

3.1.2 Intrusion Detection Systems Approaches

IDS solutions employ three detection approaches: signature-based detection, anomaly-based detection, and hybrid-based detection, as shown in Figure 3.1. Each of these approaches has its unique characteristics and advantages.

3.1.2.1 Signature-Based Detection

Signature-based approach (e.g., Suricata and Zeek) is a type of security system that uses predetermined patterns or signatures to identify malicious activity [2]. These signatures are typically derived from known malicious code or activities and serve as indicators to detect potential threats. The signature-based systems maintain a database of signatures of known attacks, which are then compared to the data being analyzed. If a match is found, the system can identify the attack and take appropriate action. Signature-based IDS are very effective at detecting known attacks. However, they are not successful in detecting unknown attacks. In order to increase the effectiveness of signature-based IDS, it is essential to keep a current signature database up-to-date [2].

This database should be updated regularly, preferably daily, with the latest signatures from anti-virus labs [26]. Without regular updates, the IDS may not be able to detect some malicious activity.

3.1.2.1.1 Suricata

Suricata is an open-source software tool for detecting and preventing unauthorized access to computer networks [28]. Thus, it functions as both an IDS/IPS. Suricata is a network security tool that utilizes signature-based rules written in the Lua scripting language to analyze network traffic and identify potential security threats [28]. Suricata's architecture is similar to Snort, but it implements a multi-threaded approach for packet processing instead of a single-threaded approach used by Snort [29]. This allows for greater efficiency and improved performance in the processing of packets.

3.1.2.1.2 Zeek

Zeek, formerly known as Bro, is an open-source network security tool that monitors and analyzes network traffic for potential security threats [30]. Zeek is designed to passively monitor network traffic, creating a log of all network connections and extracting metadata about the traffic, such as the protocol type, source and destination addresses, and the size and duration of each connection [30]. Log data can then be analyzed to detect anomalies, comprehend network behavior, and provide help with network forensics [9]. For example, the connection log file records all network connection information, the HTTP log file captures details about HTTP traffic, and the DNS log file logs DNS transactions [9]. Zeek log files also include valuable security-related information. For instance, the notice log file highlights potential suspicious activities detected by Zeek's built-in signatures and scripts [9].

3.1.2.2 Anomaly-Based Detection

Anomaly-based approach (e.g., Slips) detects malicious activity by recognizing deviations from a predetermined baseline of normal user behavior [3]. The system administrator is responsible for establishing the baseline of normal behavior, and the

IDS will then detect any behavior that deviates from this baseline. Consequently, anomaly-based IDS can detect zero-day attacks since novel attacks can be identified as soon as they occur [31]. Nevertheless, the anomaly-based detection approach is highly susceptible to generating many false positives and can lead to a significant computational burden on the computing system [32].

3.1.2.2.1 Slips

Slips is a machine learning-based IPS designed to detect and prevent cyber attacks [19]. Several detection modules are available in the Slips tool to detect C&C channels, malicious flows using Machine Learning (ML), and port scanning horizontally and vertically [19]. Slips also use ML algorithms such as Recurrent Neural Networks (RNN) algorithm to detect C&C channels [19]. Additionally, it analyzes network traffic at multiple levels, including packet, flow, and application, to detect patterns and behaviors associated with known and unknown attacks [33]. Thus, Slips can identify and block new threats that traditional signature-based systems may not detect.

3.1.2.3 Hybrid-based Detection

Hybrid-based approach combines signature-based and anomaly-based detection techniques to enhance the overall efficiency of threat detection [20]. It leverages the strengths of signature-based detection in identifying known threats and anomaly-based detection to identify unknown threats. Combining these two detection techniques, hybrid-based IDS aims to provide a comprehensive security solution for detecting known and unknown attacks.

3.2 Security Information and Events Management (SIEM)

SIEM is a comprehensive security management approach that combines the capabilities of Security Information Management (SIM) and Security Event Management (SEM) [34]. SIEM solutions enable organizations to detect and mitigate security threats by allowing real-time visibility into security-related events [5]. It collects and analyzes security-related information from multiple sources (e.g., firewalls, routers, and

IDS) and provides a centralized platform for event and log management, correlation, and reporting [5]. Thus, SIEM solutions have become critical to modern security management strategies. As the cyber threat increases, SIEM solutions will play an increasingly important role in enabling organizations to maintain a robust security posture.

3.2.1 Elasticsearch

Elasticsearch utilizes as a SIEM tool due to its advanced search and analytics capabilities, which empower security teams to search and identify potential threats within their network quickly. Elasticsearch facilitates the storage, search, and analysis of various types of documents. It provides a centralized platform for managing security data from multiple sources, including firewalls and IDS. Elasticsearch can also perform complex queries on large datasets in near real-time, enabling analysts to identify patterns and anomalies quickly [5]. It can be integrated with other complementary tools, such as Logstash for data collection and log-parsing, Kibana for analytics and visualization, and Beats [35] for data shipping. Additionally, Elasticsearch can be configured to generate alerts based on specific events. Elasticsearch also enables the design of dashboards to help security administrators visualize security data and identify trends.

Chapter 4

RESEARCH METHODOLOGY

This chapter will discuss the research methodology used to design an Ensemble Defense System (EDS) that combines Intrusion Detection Systems (IDS), including hybrid-based IDS techniques, and Security Information and Events Management (SIEM). The chapter begins with an overview of the research design before describing the research configuration and setup.

4.1 Research Design

The primary goal of designing the EDS is to develop an efficient defense system that monitors network traffic, detects a wide range of potential network threats, and simplifies network supervision using user-friendly interfaces. To achieve this primary goal, the research will utilize four well-known open-source IDS and SIEM solutions: Zeek, Suricata, Slips, and Elasticsearch. When combined in the EDS, these tools offer various detection capabilities that can effectively detect a wide range of network threats.

The architecture of EDS is shown in Figure 4.1. Initially, a script will be developed to assess the EDS's ability to detect various attacks. This script will utilize several open-source penetration testing tools such as Network Mapper (Nmap), Hping [36], and SQLMap [37]. As soon as launching these tools, the network traffic will be inspected using open-source IDS tools such as Zeek, Suricata, and Slips. These IDS tools have been chosen because of their unique features and capabilities. Specifically, we will use Zeek to conduct further analysis, while Suricata and Slips will be employed to inspect the network traffic. Suricata will generate alert log files that rely on signature-based detection, while Slips will generate files based on behavioral-based detection. Filebeat [38], a lightweight data shipper for log files, is then utilized to monitor the location of Zeek, Suricata, and Slips logs and send them to Elasticsearch. Once received, Elasticsearch will index and store the logs in its database, making searching and analyzing the data accessible. In the last step of designing the EDS, Kibana will be used to visualize the stored data interactively, with the ability to filter, search, and display the information in various formats. As a result, data collected in Kibana can be analyzed effectively by network administrators.

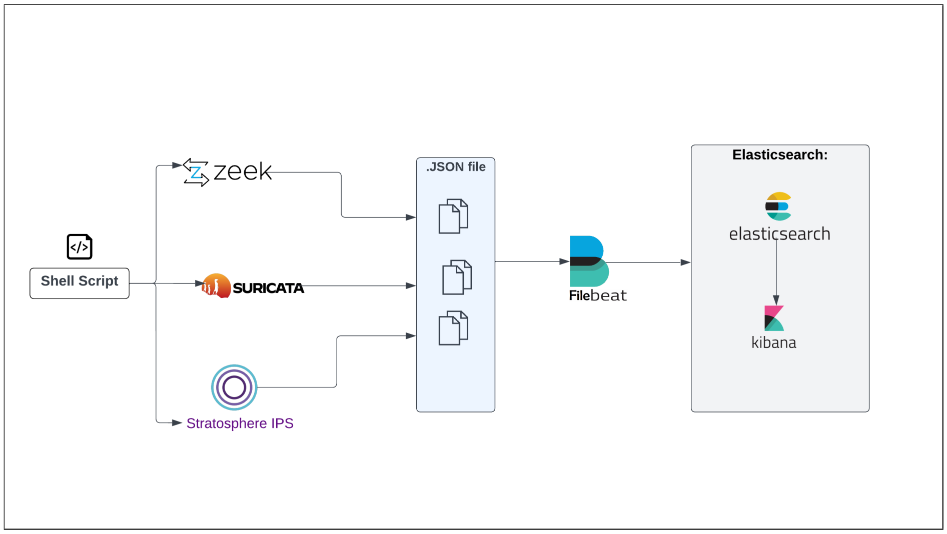
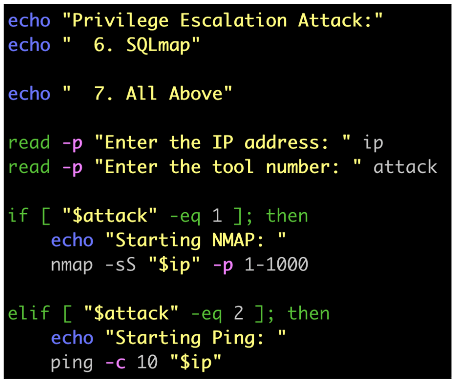


Figure 4.1: Ensemble Defense System Architecture

4.1.1 Various Shell Script Attacks

We implemented a bash script that performs several attack techniques, such as port scanning, privilege escalation, and DoS. The primary purpose of this script is to evaluate the EDS's capability to detect and visualize these attacks based on Kibana Query Language (KQL). A sample of the shell script can be seen in Figure 4.2, with the complete script in appendix A. The script first prompts the user to select the desired attack tool, including Nmap, Nikto [39], Ping [40], Hping, and SQLMap. Through these tests, we aim to identify and visualize these attacks meaningfully and to prove the EDS's ability to analyze and present attack data.



```
echo "Privilege Escalation Attack:"
echo "  6. SQLmap"

echo "  7. All Above"

read -p "Enter the IP address: " ip
read -p "Enter the tool number: " attack

if [ "$attack" -eq 1 ]; then
    echo "Starting NMAP: "
    nmap -sS "$ip" -p 1-1000

elif [ "$attack" -eq 2 ]; then
    echo "Starting Ping: "
    ping -c 10 "$ip"
```

Figure 4.2: Shell Script for Launching Multiple Attacks

4.1.2 Intrusion Detection System

The EDS uses several IDS: Zeek, Suricata, and Slips. All these systems have unique features and capabilities. For example, Zeek will extract files and metadata from network traffic, Suricata can detect known attacks, and Slips can detect unknown attacks using ML algorithms. Configuring and integrating these systems into the EDS will provide a comprehensive approach to network security monitoring and threat detection.

4.1.2.1 Zeek

Zeek generates different network traffic logs, and configuring the zeek.yml file is crucial to determine the specific logs to be generated. In this context, enabling conn.log, dns.log, dhcp.log, and http.log files is essential to get valuable insights from network activity records. Among these logs, the conn.log is the most important file due to its comprehensive record of network connection activity. To support efficient analysis of Zeek logs, they need to be converted from their default tab-separated values (TSV) format to JavaScript Object Notation (JSON) format. We can accomplish this conversion by utilizing the Filebeat tool, configuring to read Zeek logs in their original TSV format, parse the log lines, and converting them into JSON format before sending them to Elasticsearch. Therefore, the Filebeat tool will ship JSON-formatted Zeek logs to Elasticsearch for processing and visualization.

4.1.2.2 Suricata

Complementing the EDS with other cybersecurity solutions acting as NIDS that utilize signature-based detection methods is necessary. Therefore, adding Suricata to the system can provide further information on network connections and detect malicious activities in the network traffic. Suricata relies on a predefined set of rules to detect malicious activity. Additionally, Suricata can generate alerts or block network traffic if it matches a defined security rule. Within Suricata, the information regarding network events is stored in the eve.json file. This file contains a wide range of information, including the source and destination IP addresses, the type of protocol being used, the event type that has occurred, as well as other metadata related to the network traffic. The Suricata eve.json log file will be sent to Elasticsearch by Filebeat.

4.1.2.3 Slips

To enhance the detection capabilities of the EDS, integrating signature-based detection with other NIDS solutions that utilize anomaly-based detection becomes crucial. One such solution is Slips which can analyze real-time network traffic, PCAP files, or network flows generated by Zeek or Suricata and then create alert files [19]. In the EDS, we will leverage the Slips's detection modules to detect C&C channels, malicious flows using ML, and port scanning horizontally and vertically [19]. In addition, we will leverage Slips's ML algorithms feature to detect several attacks [19]. Thus, when Slips finds any suspicious traffic, it generates alert files called alerts.json, which can be forwarded to Elasticsearch for further analysis.

4.1.2.4 Comparison of IDS Solutions and Their Role in EDS

As demonstrated in Table 4.1 below, this comparison provides a comprehensive overview of IDS and their roles within EDS.

IDS	Release Date	Type	Role
Zeek	1996 (as Bro), 2018 (as Zeek) [30] [9].	Signature-based IDS.	It can analyze protocol and extract metadata from network packets.
Suricata	2009 [15].	Signature-based IDS.	It can generate alerts for suspicious network traffic that matches its signature rules.
Slips	2016 [41].	Behavioral-based IDS.	It can generate alerts using signature-based detection and anomaly-based detection techniques.

Table 4.1: Comparison Between the IDS and Their Role in EDS

4.1.3 Filebeat

Filebeat is essential in centralizing and forwarding log data to Elasticsearch [38]. Filebeat collects various logs, such as ZeekâĂŹs logs files, Suricata's eve.json files, and Slips's alert.json files. Once collected, the logs are sent to Elasticsearch, allowing for a better understanding of network traffic. Through Filebeat, we can efficiently send log data streaming in real-time to Elasticsearch, simplifying network monitoring and analysis.

4.1.4 Elasticsearch and Kibana

Elasticsearch is an open-source search and analytics engine for all data types [5]. We will collect the JSON logs files from Zeek, Suricata, and Slips and then forward and store them in Elasticsearch through Filebeat.

To visualize and interact with the data stored in Elasticsearch, we will use Kibana, a browser-based user interface that enables searching, analyzing, and visualizing the data [7]. The primary purpose of using Kibana is to analyze network traffic through customized interfaces. These interfaces will facilitate the exploration of the incoming network data in a user-friendly manner.

4.2 Configuration and Set Up

The section will be divided into three phases to complete the EDS configuration. The first phase will focus on configuring the IDS systems, including Zeek, Suricata, and Slips. In the second phase, data generated by the IDS systems will be integrated and stored using Filebeat and Elasticsearch. This will involve setting up the inputs and outputs in Filebeat and configuring the log paths for each IDS system in Docker [42]. Finally, in the third phase, the Kibana will be utilized to build a dashboard that visualizes the log data stored in Elasticsearch. The dashboard will provide an easy-to-understand format of the network traffic and user-friendly dashboards to simplify the detection of any potential security threats.

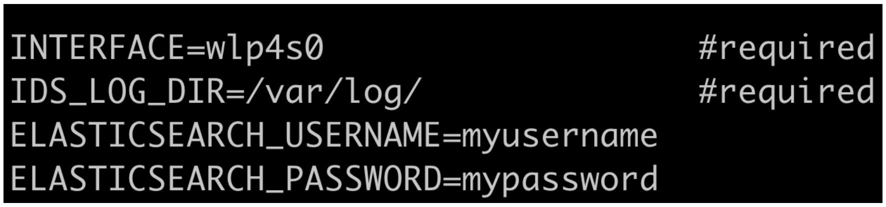
4.2.1 IDS Configurations

In this system, we have used a combination of signature-based IDS (Zeek and Suricata) and behavioral-based IDS (Slips) to enhance the overall network security. For the implementation, we will use Docker Compose [43] to use the latest images for the three IDS employed in this system. The complete Docker Compose files can be found in a repository on GitHub [44].

A key reason for choosing Docker is its ease of management and deployment. As an example of how Docker facilitates the configuration of the EDS, it allows for saving Elasticsearch dashboard settings for future uses. Additionally, Docker Compose allows using environment variables to pass dynamic values to containers at runtime. These environment variables can be utilized across IDS's YAML files to specify essential details such as network interface, log paths, and Elasticsearch credentials. Thus, the configuration process of IDS's YAML files becomes more flexible and customizable.

Figure 4.3 illustrates an example of environment variables used in a Docker Compose file for the EDS. These variables include:

- `INTERFACE`: Specifies the network interface for the host.
- `IDS_LOG_DIS`: Specifies the directory for log files.
- `ELASTICSEARCH_USERNAME_PASSWORD`: Specifies the Elasticsearch credentials.



```
INTERFACE=wlp4s0                    #required
IDS_LOG_DIR=/var/log/               #required
ELASTICSEARCH_USERNAME=myusername
ELASTICSEARCH_PASSWORD=mypassword
```

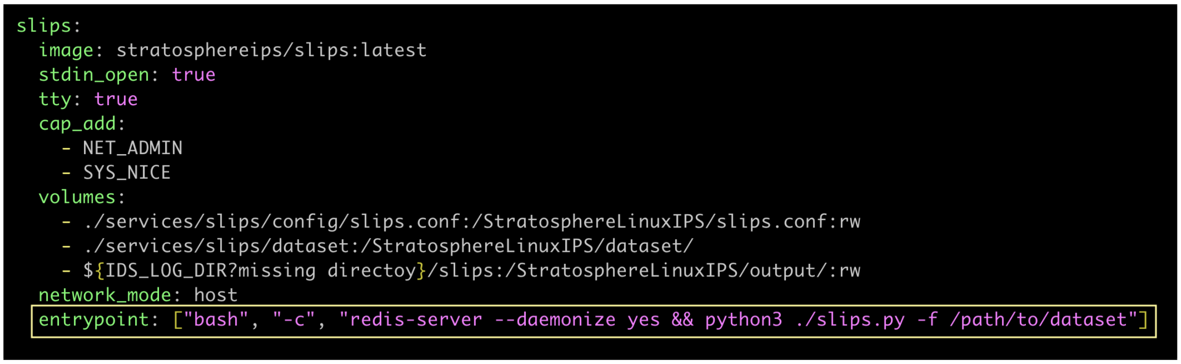
Figure 4.3: EDS Configuration in the Docker .ENV File

4.2.1.1 Signature-based IDS: Suricata and Zeek

Suricata can detect known threats based on predefined rules. Thus, Suricata will be configured to use the latest several rules to generate alerts for the latest known attacks. Additionally, for Zeek, we specified custom configurations and scripts in the local.zeek file to load the `@load ja3` and `@load hassh` packages, allowing Zeek to detect and analyze JA3 and HASSH fingerprints for SSL/TLS connections [45]. As a result, these packages can help identify potential threats.

4.2.1.2 Behavioral-based IDS: Slips

After using the latest Slips image in the Docker Compose file, we configured the entrypoint section, as shown in Figure 4.4. The purpose of this configuration was to enable Slips to start by launching a Redis [46] database in daemon mode. This Redis database is a cache for Slips [19]. Once the Redis database runs, a Slips Python script scans the dataset file in the specified directory path or inspects the packets on the defined network interface.

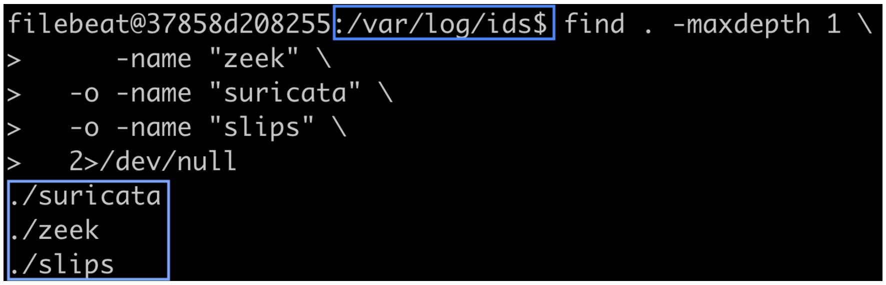


```
slips:
  image: stratosphereips/slips:latest
  stdin_open: true
  tty: true
  cap_add:
    - NET_ADMIN
    - SYS_NICE
  volumes:
    - ./services/slips/config/slips.conf:/StratosphereLinuxIPS/slips.conf:rw
    - ./services/slips/dataset:/StratosphereLinuxIPS/dataset/
    - ${IDS_LOG_DIR?missing directoy}/slips:/StratosphereLinuxIPS/output/:rw
  network_mode: host
  entrypoint: ["bash", "-c", "redis-server --daemonize yes && python3 ./slips.py -f /path/to/dataset"]
```

Figure 4.4: Configuration of Entrypoint in Docker for Slips

4.2.2 Configuration of Data Integration and Storage

In order to store all the IDS logs using Docker Compose, we created a volume mount that mapped the IDS logs on the host machine to the container directory `/var/log/ids/`, as shown in Figure 4.5. This allowed easy access to log data and ensured that log data was preserved even if the container was removed or re-created.

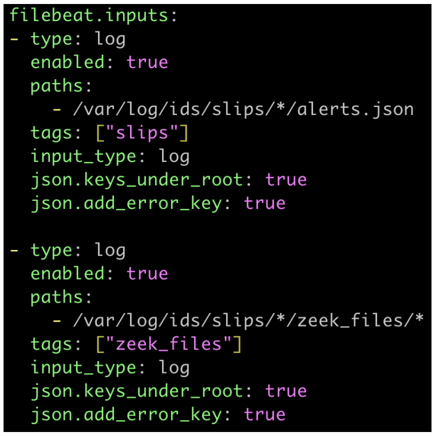


```
filebeat@37858d208255:/var/log/ids$ find . -maxdepth 1 \
>      -name "zeek" \
>   -o -name "suricata" \
>   -o -name "slips" \
>   2>/dev/null
./suricata
./zeek
./slips
```

Figure 4.5: Folder Containing IDS Logs at /var/log/ids

4.2.2.1 SIEM Solution: Filebeat

Within the YAML file (filebeat.yaml) located in the Filebeat directory, we have enabled the Zeek and Suricata modules. We also made the necessary configurations to read and forward the Slips alert log to Elasticsearch for indexing and analysis. It is crucial to ensure that all output from Slips is shipped to Elasticsearch, considering that Slips has its own Zeek configuration. For the Slips alert log and its Zeek logs, we set the type property to log, enabled it, and specified the paths of the logs to be collected, as illustrated in Figure 4.6. In addition, we set the tags property to ["slips"] to label all log data collected from `/StratosphereLinuxIPS/output/` directory and ["zeek_files"] to label all Slips' Zeek logs. As a result of using this field, Slips log data could be filtered within Elasticsearch by tags.



```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/log/ids/slips/*/alerts.json
  tags: ["slips"]
  input_type: log
  json.keys_under_root: true
  json.add_error_key: true

- type: log
  enabled: true
  paths:
    - /var/log/ids/slips/*/zeek_files/*
  tags: ["zeek_files"]
  input_type: log
  json.keys_under_root: true
  json.add_error_key: true
```

Figure 4.6: Slips Configuration in the Filebeat YAML File

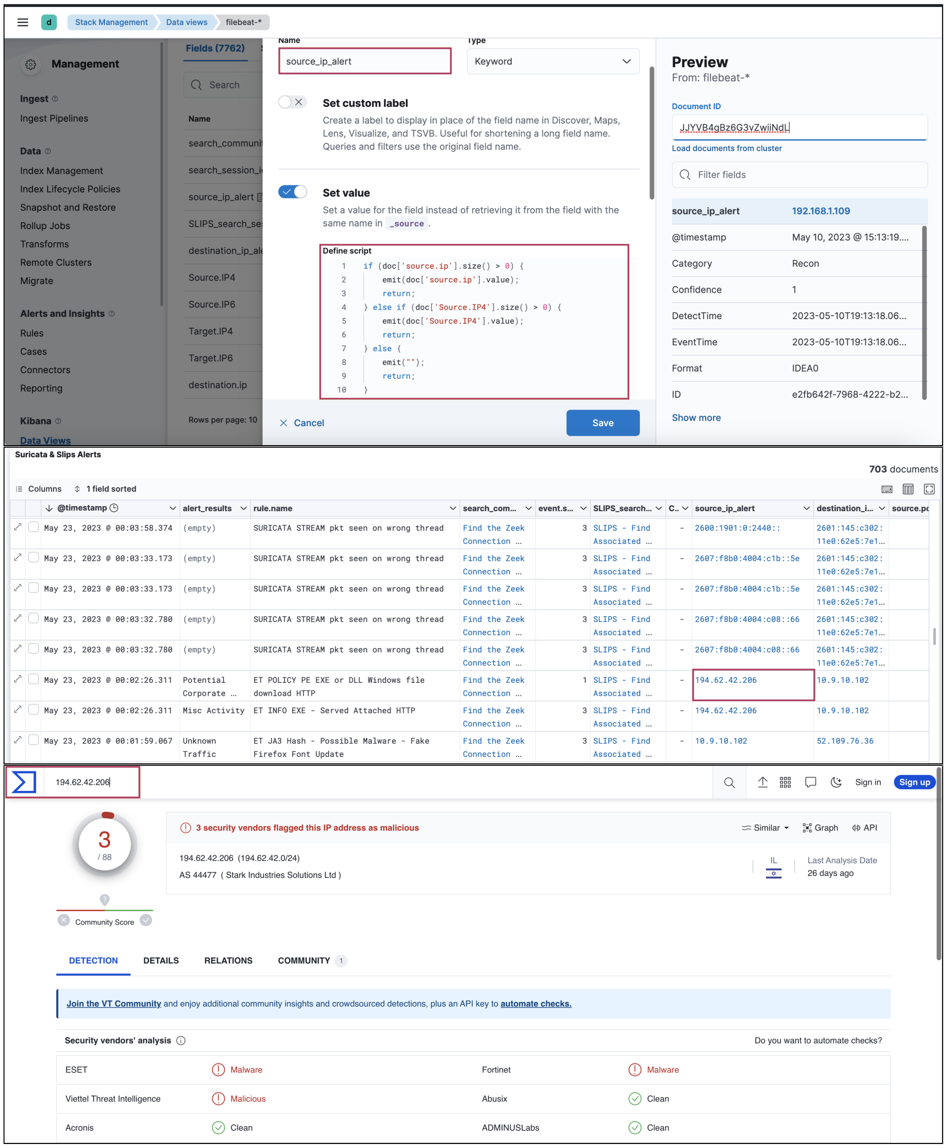
4.2.2.2 SIEM Solution: Elasticsearch

Data integration and storage are crucial components of Elasticsearch configuration. To collect logs from Zeek, Suricata, and Slips and forward them to Elasticsearch for storage and indexing, we used the Filebeat tool. Integrating Filebeat with Elasticsearch allows us to centralize and analyze log data from signature and behavioral-based

IDS in a single platform. Additionally, the integration will help enhance the efficiency of the EDS in the threat detection process.

4.2.2.3 Elasticsearch Runtime Fields

Leveraging Elasticsearch runtime fields can enrich IDS logs during runtime and enhance their contextual relevance and usefulness. Therefore, we added numerous runtime fields to Elasticsearch by writing scripts. For example, as shown in Figure 4.7, we wrote a script to merge the destination and source IP addresses from Suricata and Slips and then redirect to the VirusTotal [47] website to get insights and analysis on IP addresses. VirusTotal includes information about these IP addresses from various security sources and databases [47]. This information can help network administrators make more informed decisions about threat detection and response.



Name: source_ip_alert
Type: Keyword

Set custom label
Create a label to display in place of the field name in Discover, Maps, Lens, Visualize, and TSVB. Useful for shortening a long field name. Queries and filters use the original field name.

Set value
Set a value for the field instead of retrieving it from the field with the same name in `_source`.

Define script

```
if (doc['source.ip'].size() > 0) {
   emit(doc['source.ip'].value);
   return;
} else if (doc['Source.IP4'].size() > 0) {
   emit(doc['Source.IP4'].value);
   return;
} else {
   emit("");
   return;
}
```

Preview
From: filebeat-*

Document ID: JJYVB4gBz6G3vZwiiNdL
Load documents from cluster
Filter fields

source_ip_alert	192.168.1.109
@timestamp	May 10, 2023 @ 15:13:19....
Category	Recon
Confidence	1
DetectTime	2023-05-10T19:13:18.06...
EventTime	2023-05-10T19:13:18.06...
Format	IDEA0
ID	e2fb642f-7968-4222-b2...

Show more

Suricata & Slips Alerts

703 documents

Columns 1 field sorted

@timestamp	alert_results	rule.name	search_com...	event.s...	SLIPS_search...	C..	source_ip_alert	destination_i...	source.pc
May 23, 2023 @ 00:03:58.374	(empty)	SURICATA STREAM pkt seen on wrong thread	Find the Zeek Connection ...	3	SLIPS - Find Associated ...	-	2600:1901:0:2440::	2601:145:c302:11e0:62e5:7e1...	
May 23, 2023 @ 00:03:33.173	(empty)	SURICATA STREAM pkt seen on wrong thread	Find the Zeek Connection ...	3	SLIPS - Find Associated ...	-	2607:f8b0:4004:c1b::5e	2601:145:c302:11e0:62e5:7e1...	
May 23, 2023 @ 00:03:33.173	(empty)	SURICATA STREAM pkt seen on wrong thread	Find the Zeek Connection ...	3	SLIPS - Find Associated ...	-	2607:f8b0:4004:c1b::5e	2601:145:c302:11e0:62e5:7e1...	
May 23, 2023 @ 00:03:32.780	(empty)	SURICATA STREAM pkt seen on wrong thread	Find the Zeek Connection ...	3	SLIPS - Find Associated ...	-	2607:f8b0:4004:c08::66	2601:145:c302:11e0:62e5:7e1...	
May 23, 2023 @ 00:03:32.780	(empty)	SURICATA STREAM pkt seen on wrong thread	Find the Zeek Connection ...	3	SLIPS - Find Associated ...	-	2607:f8b0:4004:c08::66	2601:145:c302:11e0:62e5:7e1...	
May 23, 2023 @ 00:02:26.311	Potential Corporate ...	ET POLICY PE EXE or DLL Windows file download HTTP	Find the Zeek Connection ...	1	SLIPS - Find Associated ...	-	194.62.42.206	10.9.10.102	
May 23, 2023 @ 00:02:26.311	Misc Activity	ET INFO EXE - Served Attached HTTP	Find the Zeek Connection ...	3	SLIPS - Find Associated ...	-	194.62.42.206	10.9.10.102	
May 23, 2023 @ 00:01:59.067	Unknown Traffic	ET JA3 Hash - Possible Malware - Fake Firefox Font Update	Find the Zeek Connection ...	3	SLIPS - Find Associated ...	-	10.9.10.102	52.109.76.36	

194.62.42.206

3 security vendors flagged this IP address as malicious

194.62.42.206 (194.62.42.0/24)
AS 44477 (Stark Industries Solutions Ltd)

Last Analysis Date
26 days ago

DETECTION   DETAILS   RELATIONS   COMMUNITY 1

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis   Do you want to automate checks?

ESET	Malware	Fortinet	Malware
Viettel Threat Intelligence	Malicious	Abusix	Clean
Acronis	Clean	ADMINUSLabs	Clean

Figure 4.7: Runtime Field for IP Address Redirection to VirusTotal Website

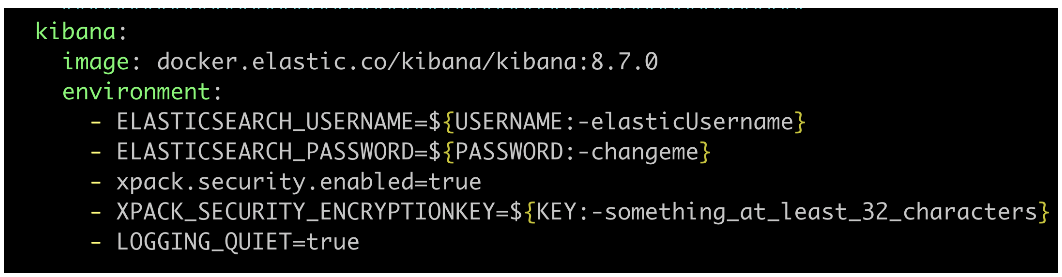
Additionally, utilizing the same method, we incorporated a runtime field to merge Suricata and Slips alerts under a single title to facilitate efficient monitoring and analysis of IDS alerts within the system's visualization. Moreover, we created separate runtime fields for `community_id` and `session_id`, enabling easy navigation to related dashboards and providing a more comprehensive view of network activity and associated alerts.

4.2.3 Configuration of Data Visualization

In addition to data integration and storage, it is essential to visualize these data meaningfully using user-friendly interfaces. For this purpose, we will utilize Kibana. Kibana allows us to create custom dashboards and visualizations and generate reports and alerts for the collected logs. Thus, network administrators can quickly identify patterns, trends, and anomalies in IDS logs by analyzing their collected data.

4.2.3.1 SIEM Solution: Kibana

As illustrated in Figure 4.8, in the Docker Compose file, we have set up login credentials to secure Kibana dashboard access. In addition, to ensure the encryption of sensitive data, we have also specified an encryption key using the "XPACK SECURITY ENCRYPTIONKEY" environment variable. This key is used to secure communications between Kibana and Elasticsearch. Finally, we have also specified a volume mount in the Docker Compose file to ensure that Elasticsearch data persists even if the container is stopped. Specifically, we have mounted the 'Elasticsearch' image to `/usr/share/elasticsearch/data` path, where Kibana stores dashboards, visualizations, settings, and other data in Elasticsearch.

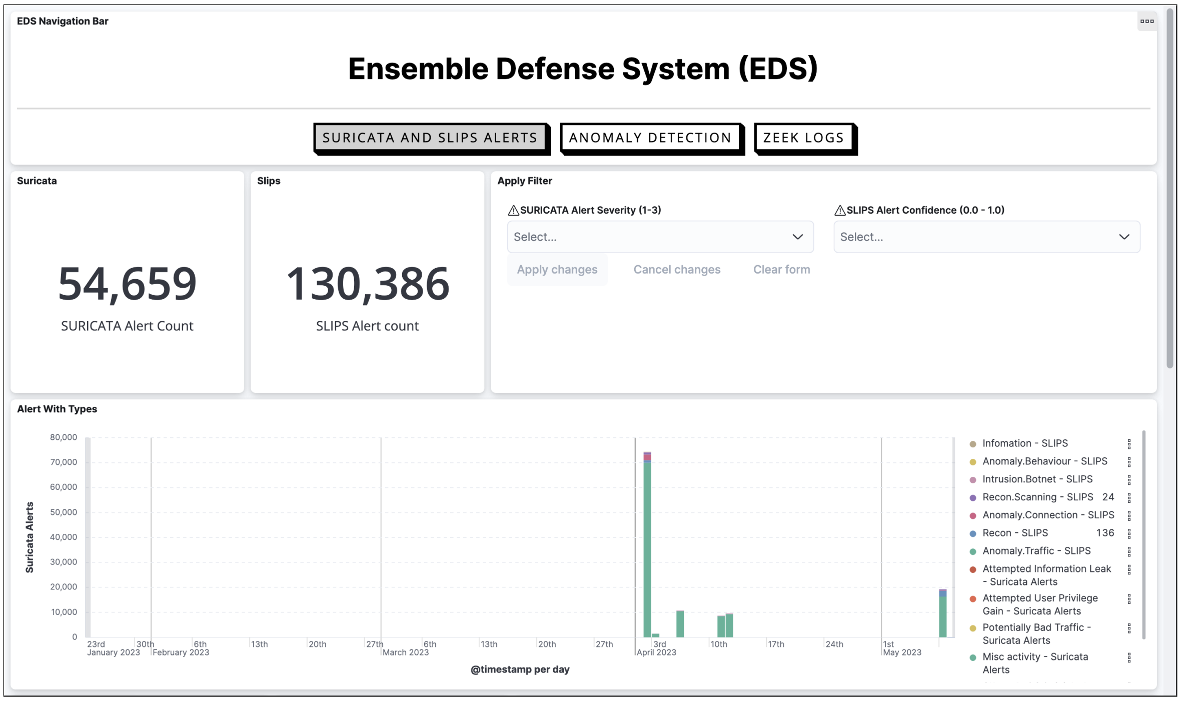


```
kibana:
  image: docker.elastic.co/kibana/kibana:8.7.0
  environment:
    - ELASTICSEARCH_USERNAME=${USERNAME:-elasticUsername}
    - ELASTICSEARCH_PASSWORD=${PASSWORD:-changeme}
    - xpack.security.enabled=true
    - XPACK_SECURITY_ENCRYPTIONKEY=${KEY:-something_at_least_32_characters}
    - LOGGING_QUIET=true
```

Figure 4.8: Secure Access Mechanism for Kibana Dashboard

4.2.3.2 Slips and Suricata Alert Dashboard

The alert outputs from IDS tools should be considered when detecting potential threats. Since Slips and Suricata IDS tools generate alerts, we merged their alert results on one dashboard within the EDS. As shown in Figure 4.9, in addition to the navigation section and the IDS alert count, we included a chart that provides valuable information about the alert type and its generation source. We also added a filter for the output based on two elements: alert severity for Suricata and the alert confidence score for Slips. Users can identify potential threats quickly and efficiently using this filter.



EDS Navigation Bar

Ensemble Defense System (EDS)

SURICATA AND SLIPS ALERTS | ANOMALY DETECTION | ZEEK LOGS

Suricata

54,659

SURICATA Alert Count

Slips

130,386

SLIPS Alert count

Apply Filter

⚠SURICATA Alert Severity (1-3): Select...

⚠SLIPS Alert Confidence (0.0 - 1.0): Select...

Apply changes | Cancel changes | Clear form

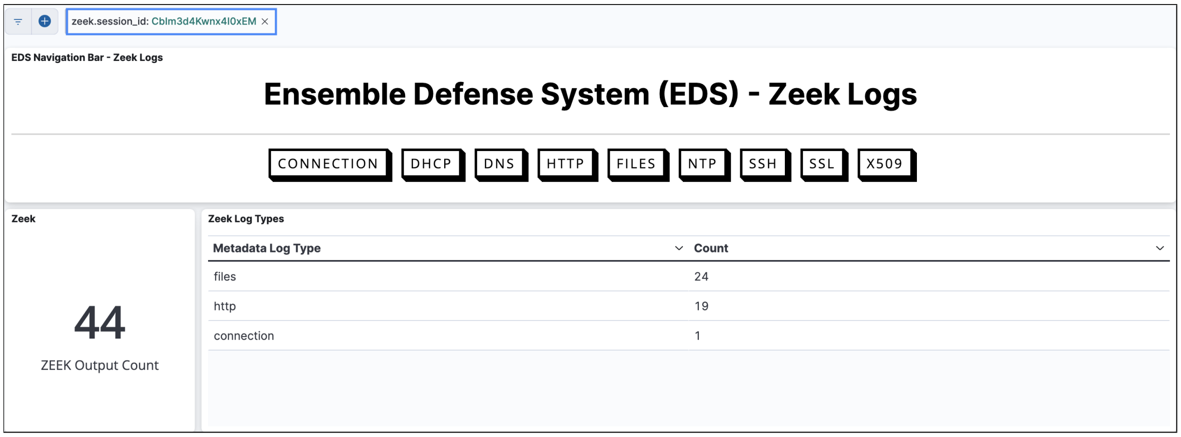
Alert With Types

Suricata Alerts
80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
23rd January 2023
30th
6th February 2023
13th
20th
27th
6th March 2023
13th
20th
27th
3rd April 2023
10th
17th
24th
1st May 2023
@timestamp per day
● Infomation - SLIPS
● Anomaly.Behaviour - SLIPS
● Intrusion.Botnet - SLIPS
● Recon.Scanning - SLIPS 24
● Anomaly.Connection - SLIPS
● Recon - SLIPS 136
● Anomaly.Traffic - SLIPS
● Attempted Information Leak - Suricata Alerts
● Attempted User Privilege Gain - Suricata Alerts
● Potentially Bad Traffic - Suricata Alerts
● Misc activity - Suricata Alerts

Figure 4.9: Suricata and Slips Alerts Dashboard

4.2.3.3 Overview of Metadata Dashboards

We developed a metadata dashboard, as shown in Figure 4.10, which provides valuable information about the Zeek logs by utilizing the `session_id` field. The `session_id` assigns to each session or connection, allowing easy correlation of logs and events related to that session [9]. We filtered the Zeek logs using the `session_id` field to present information about a specific session or connection. We focus on listing protocol usage for a particular session in order to gain insights into the characteristics of network sessions.



zeek.session_id: Cblm3d4Kwnx4l0xEM

EDS Navigation Bar - Zeek Logs

Ensemble Defense System (EDS) - Zeek Logs

Zeek

44

ZEEK Output Count

Zeek Log Types

Metadata Log Type	Count
files	24
http	19
connection	1

Figure 4.10: Metadata Dashboard for Zeek Logs Using session_id Field

4.2.3.4 Anomaly Detections Using KQL Queries

KQL is a powerful query language designed for use with Elasticsearch [5]. It allows users to perform complex searches and aggregate data to enable the identification of potential anomalies and unusual patterns in network traffic. Additionally, KQL queries can be employed to analyze specific attributes of network traffic, such as the frequency and duration of connections, to detect potential threats. Furthermore, it is worth noting that combining KQL with ML algorithms and statistical models can enhance the accuracy and effectiveness of anomaly detection in network traffic [5]; however, this aspect is outside the scope of the current thesis. Thus, we will use the KQL to identify several attacks:

1. **Port scanning attack:** This attack can be identified as multiple connections to different ports originating from the same IP address within a short period of time. [48].

2. **DoS attack:** The attack can be observed as a sudden surge in requests, such as Ping, from a particular source IP address or an unusual volume of traffic targeting specific ports [48].

3. **Privilege escalation attack:** This attack may be indicated by a sequence of failed login attempts and followed by a successful login [48].

By efficiently analyzing large volumes of network traffic data, KQL queries can provide valuable insights into detected security threats.

Chapter 5

ANALYSIS AND EVALUATION

This chapter presents the analysis and evaluation of the Ensemble Defense System (EDS) that combines Intrusion Detection Systems (IDS) and Security Information and Events Management (SIEM). This chapter aims to assess the implemented EDS's effectiveness and performance in achieving its goals of providing a usable and efficient defense system based on multiple open-source tools. We will evaluate the EDS's ability to detect specific types of attacks, such as port scanning, privilege escalation, and DoS. We will also examine the EDS's ability to correlate several logs to detect potential attacks.

5.1 Detection of Attacks Utilizing Common Attack Tools

As part of the EDS evaluation, we implemented a script that perform several attack techniques, including port scanning, privilege escalation, and DoS attacks. Through the use of this script, we aimed to evaluate the effectiveness of the EDS in detecting and visualizing these attacks using the KQL.

5.1.1 Port Scanning

We have used the KQL query to detect different port scanning attacks and evaluate the EDS's capability to identify and visualize various port scan attempts. Port scanning attacks are detected using the following KQL query:

```
not (network.direction: "outbound")
and ((not (network.transport: "icmp")
and not (zeek.connection.history: /Sh*|F*|D*/))
or (network.transport: "icmp"
and zeek.connection.icmp.type: "8"))
```

The KQL query excludes outbound traffic as a first step in filtering network logs to identify port scanning attempts, followed by two alternative conditions. The first condition filters network traffic that does not use the ICMP transport protocol and does not have a Zeek connection history that matches the specified patterns. This can help identify TCP/UDP port scan attempts. The second condition filters for network traffic that uses the ICMP transport protocol and has an ICMP type of "8" (echo request). This can help identify ICMP-based port scan (ping scan) attempts.

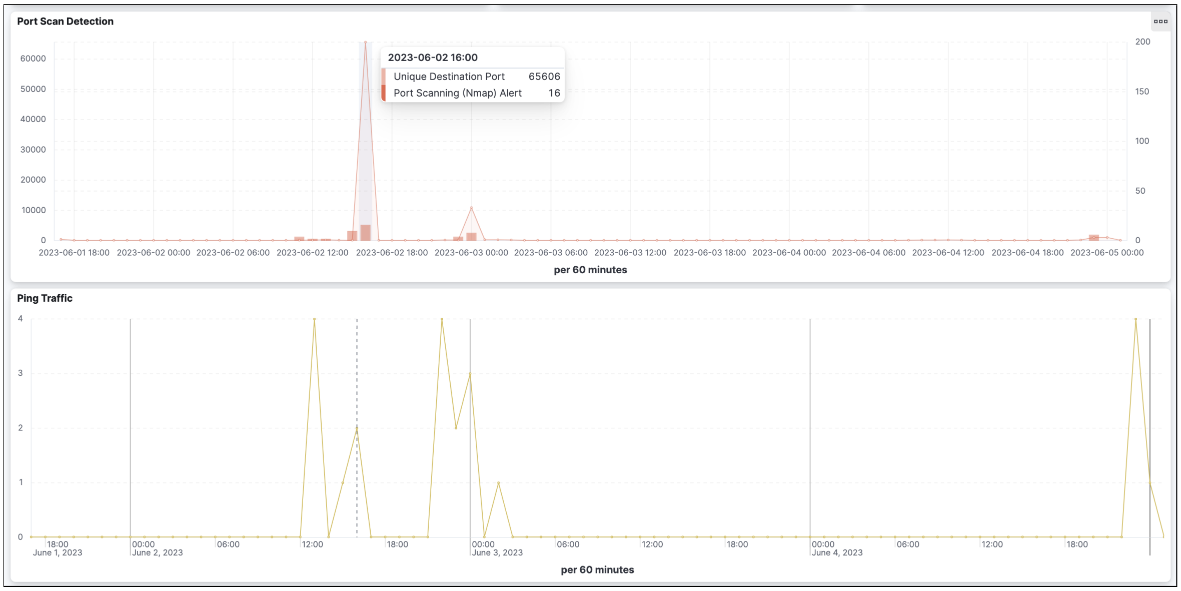
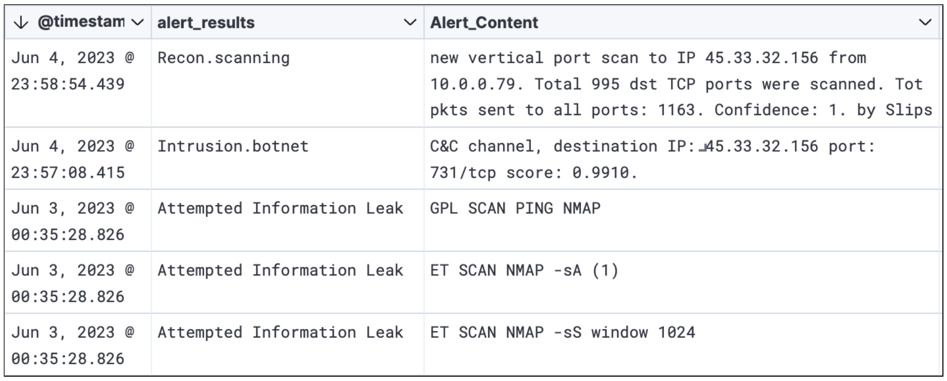


Figure 5.1: Port Scanning Attack Detection Using KQL

As part of our port scanning attack evaluation, we utilized several tools through the implemented bash script, including Nmap (`nmap -sS <ip> -p 1-1000`), Ping (`ping -c 10 <ip>`), and Nikto (`nikto -h <ip>`.) As a demonstrator in Figure 5.1, the first graph shows the port scanning attempts by the shaded light red region beneath the straight line graph. These attempts were primarily caused by the usage of the Nmap tool. Furthermore, the red bars represent the number of Nmap attack alerts generated by Suricata and Slips. These alerts are filtered using the following KQL:

```
(event.module:"suricata" AND
rule.category:"Attempted Information Leak")
OR (tags:"slips" AND Category:"Recon.Scanning")
```

These filters allowed us to identify alerts about attempted information leaks from Suricata and reconnaissance scanning from Slips, as shown in Figure 5.2. Additionally, in Figure 5.1, the graph below also depicts the number of ping requests recorded during each scan. These requests were generated through the use of Nmap, Ping, and Nikto tools.



↓ @timestam	alert_results	Alert_Content
Jun 4, 2023 @ 23:58:54.439	Recon.scanning	new vertical port scan to IP 45.33.32.156 from 10.0.0.79. Total 995 dst TCP ports were scanned. Tot pkts sent to all ports: 1163. Confidence: 1. by Slips
Jun 4, 2023 @ 23:57:08.415	Intrusion.botnet	C&C channel, destination IP: 45.33.32.156 port: 731/tcp score: 0.9910.
Jun 3, 2023 @ 00:35:28.826	Attempted Information Leak	GPL SCAN PING NMAP
Jun 3, 2023 @ 00:35:28.826	Attempted Information Leak	ET SCAN NMAP -sA (1)
Jun 3, 2023 @ 00:35:28.826	Attempted Information Leak	ET SCAN NMAP -sS window 1024

Figure 5.2: Port Scanning Attack Detection by Slips and Suricata

5.1.2 Denial-of-Service (DoS)

The following KQL: `not (network.direction: "outbound")` are used to detect the DoS attack, where the main goal is to exclude outbound traffic, focus on inbound or internal traffic, and narrow down the visualization in the graph to detect any potential DoS attack. We used the Hping tool to conduct the DoS attack simulations (`hping3 -c 100 -p 21 -w 64 -d 120 --flood --rand-source <ip>`), which aims to flood the target with a large volume of packets. We launched attacks on both port 21 and port 80. In Figure 5.3, the graph illustrates the targeted port numbers for the DoS attacks. The shaded green region beneath the straight line graph represents the DoS attack on port 21, while the blue shade represents the same attack on port 80. In the second graph, the red bars represent the total number of alerts generated by Suricata and Slips. These alerts are filtered using the following KQL:

```
(event.module : "suricata"
and (rule.category: ("Attempted Denial of Service"))
or (tags: "slips"
and (Category: "Malware" or Category: "anomaly.traffic"))
```

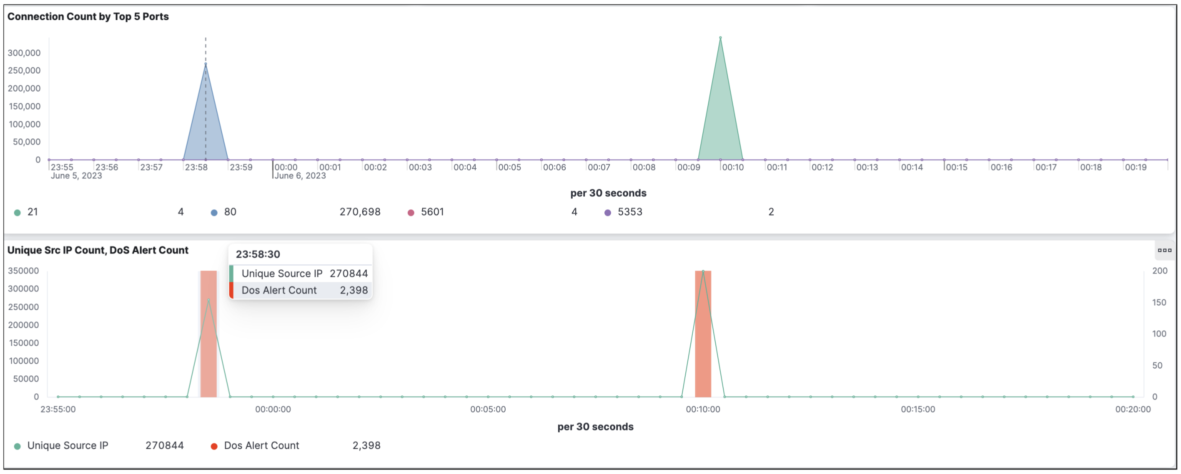


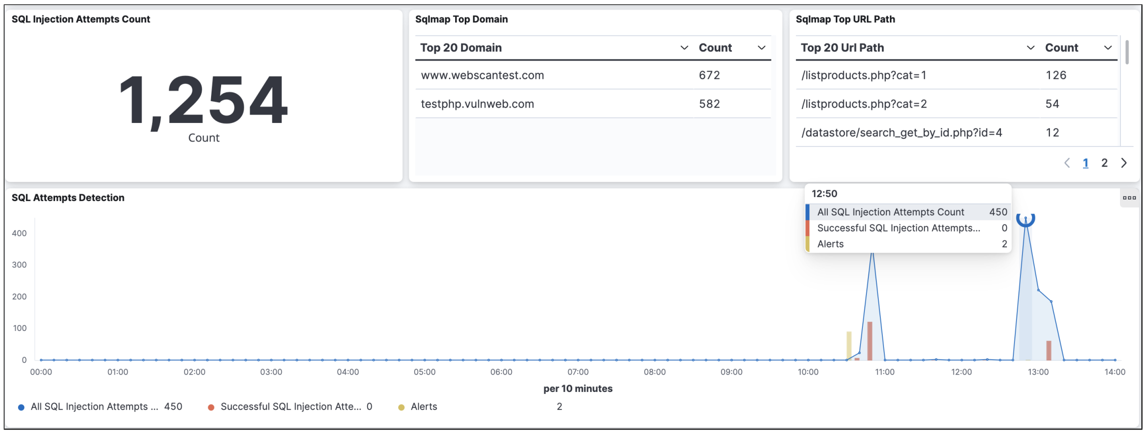
Figure 5.3: DoS Attack Detection Using KQL

The number of packets transmitted through port 80 is 270,717, and the total number of generated alerts from Suricata and Slips is 2,398. Additionally, the number of packets transmitted through port 21 is 214,224, and the total number of generated alerts from both Suricata and Slips is 670. These provide insights into the volume of network traffic during a DoS attack and the EDS's capability to generate alerts from the IDS tools.

5.1.3 Privilege Escalation

To evaluate the EDS's capability in detecting privilege escalation attacks, we utilized the SQLMap tool, commonly used to perform SQL injection (SQLi) attacks. The KQL query `user_agent.original: sqlmap*` was used for this purpose. This query filters the network logs and specifically searches for user agent strings that contain the term "sqlmap." With this query, we captured and analyzed the network traffic associated with sqlmap requests. As depicted in Figure 5.4, the provided information, such as the URL domain name and URL path, provides valuable insights into the attacker's methodology when attempting SQLi attacks. Furthermore, the graph below provides a general view of EDS's ability to detect potential SQLi attacks on the system by presenting the total number of SQLi attacks in the shaded light blue region beneath the straight line graph, the number of successful SQLi attacks in the red bar, and the count of alerts generated by Suricata and Zeek in the yellow bar. These alerts are filtered using the following KQL:

```
(event.module: "suricata"
and rule.category: "A Network Trojan was detected")
or (rule.name: "HTTP::SQL_Injection_Attacker"
or rule.name: "HTTP::SQL_Injection_Victim")
```



SQL Injection Attempts Count

1,254
Count

Sqlmap Top Domain

Top 20 Domain	Count
www.webscantest.com	672
testphp.vulnweb.com	582

Sqlmap Top URL Path

Top 20 Url Path	Count
/listproducts.php?cat=1	126
/listproducts.php?cat=2	54
/datastore/search_get_by_id.php?id=4	12

SQL Attempts Detection

12:50

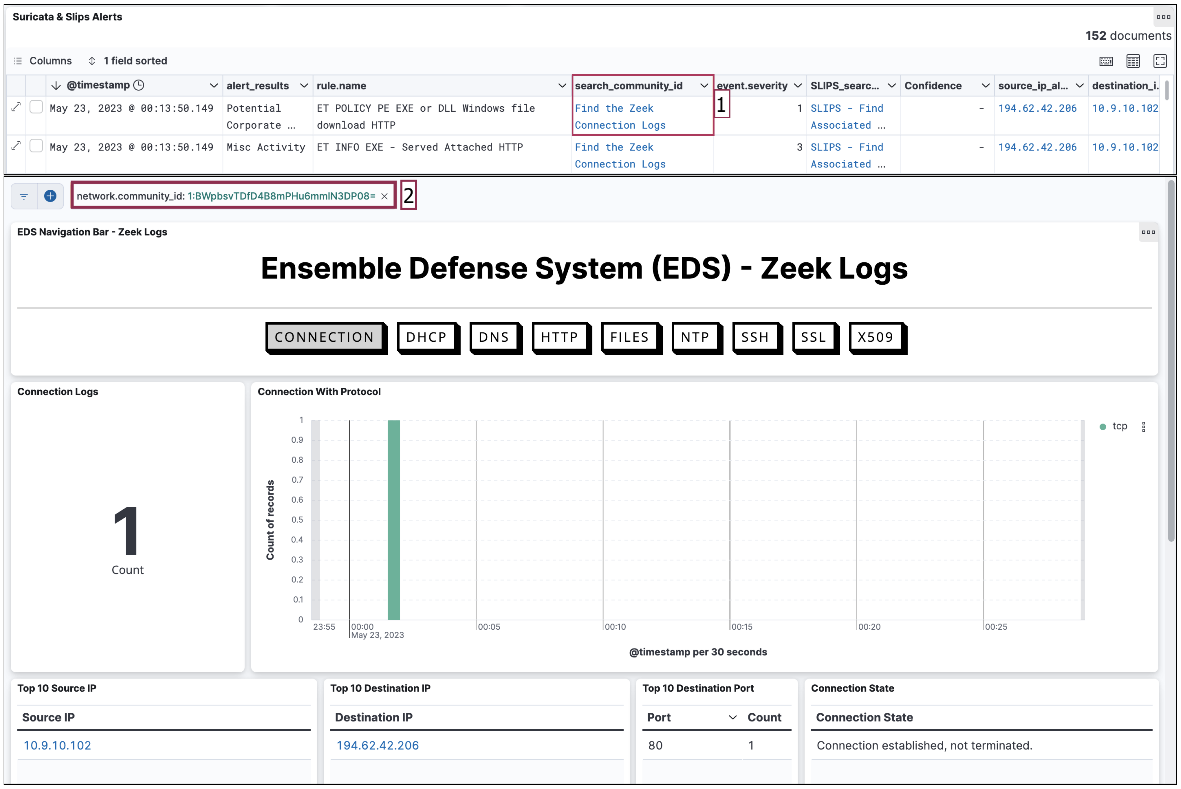
All SQL Injection Attempts Count	450
Successful SQL Injection Attempts...	0
Alerts	2

Figure 5.4: SQLi Attack Detection Using KQL

5.2 Benefits of Correlation in the EDS

We can correlate Suricata and Slips alert logs with Zeek logs using `community_id` and `session_id`. Leveraging these fields within IDS logs will help establish correlations between network flows. An IDS's `community_id` field identifies network flow characteristics based on five-tuple features: source IP address, destination IP address, source port, destination port, and protocol, while IDS's `session_id` field is a unique identifier assigned to each network session [9]. The information in these fields can be utilized to correlate network flows in order to explore more related information regarding the generated alerts.

To analyze and evaluate the correlation process in the EDS, we first analyzed a dataset from a malware traffic analysis website that examined and documented various forms of malicious network traffic and malware samples [49]. As indicated by number 1 in Figure 5.5, when Suricata generates alerts, we can locate the corresponding connection logs in Zeek logs using the `community_id` field. As indicated by number 2, we could present more details related to the Zeek connection log that drove the Suricata to generate an alert.


Suricata & Slips Alerts

152 documents

@timestamp	alert_results	rule.name	search_community_id	event.severity	SLIPS_searc...	Confidence	source_ip_al...	destination_i...
May 23, 2023 @ 00:13:50.149	Potential Corporate ...	ET POLICY PE EXE or DLL Windows file download HTTP	Find the Zeek Connection Logs	1	SLIPS - Find Associated ...	-	194.62.42.206	10.9.10.102
May 23, 2023 @ 00:13:50.149	Misc Activity	ET INFO EXE - Served Attached HTTP	Find the Zeek Connection Logs	3	SLIPS - Find Associated ...	-	194.62.42.206	10.9.10.102

network.community_id: 1:BWpbsvTDfD4B8mPHu6mmIN3DP08=

EDS Navigation Bar - Zeek Logs

Ensemble Defense System (EDS) - Zeek Logs

Connection Logs: 1 Count

Top 10 Source IP
Source IP
10.9.10.102

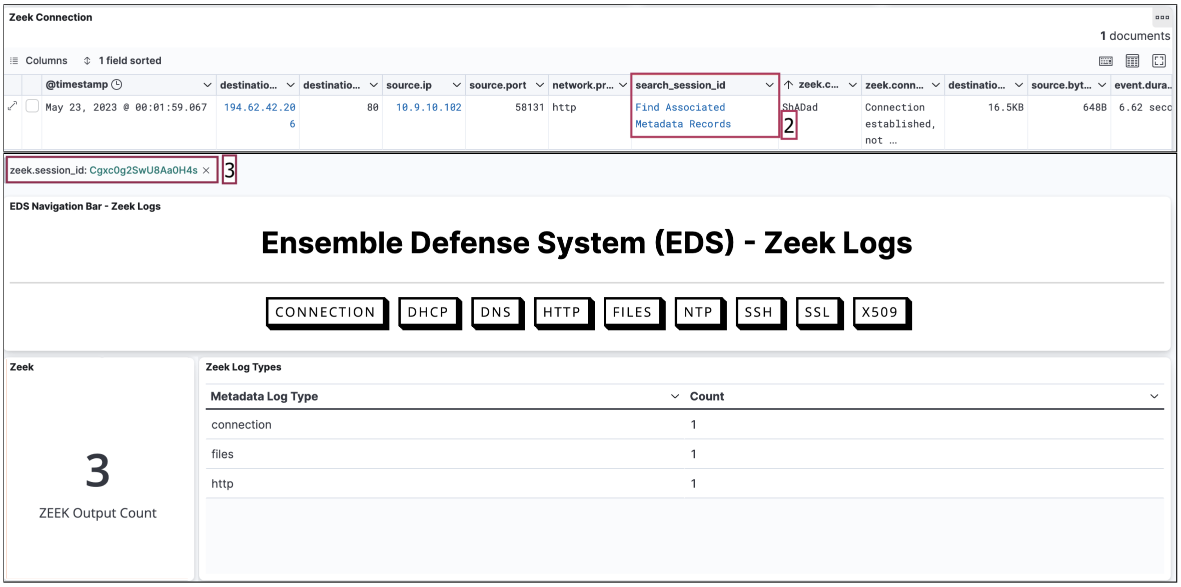
Top 10 Destination IP
Destination IP
194.62.42.206

Port	Count
80	1

Connection State
Connection established, not terminated.

Figure 5.5: Part 1: Correlation Between Zeek Logs and Suricata Alert Logs

To further investigate, we can use the `session_id` field to discover relevant Zeek metadata logs for each session, as depicted in numbers 2 and 3 in Figure 5.6. The importance of examining the Zeek metadata enables us to observe the frequency of each Zeek log. In this particular Figure 5.6, we can identify the presence of a file exchange within the session.



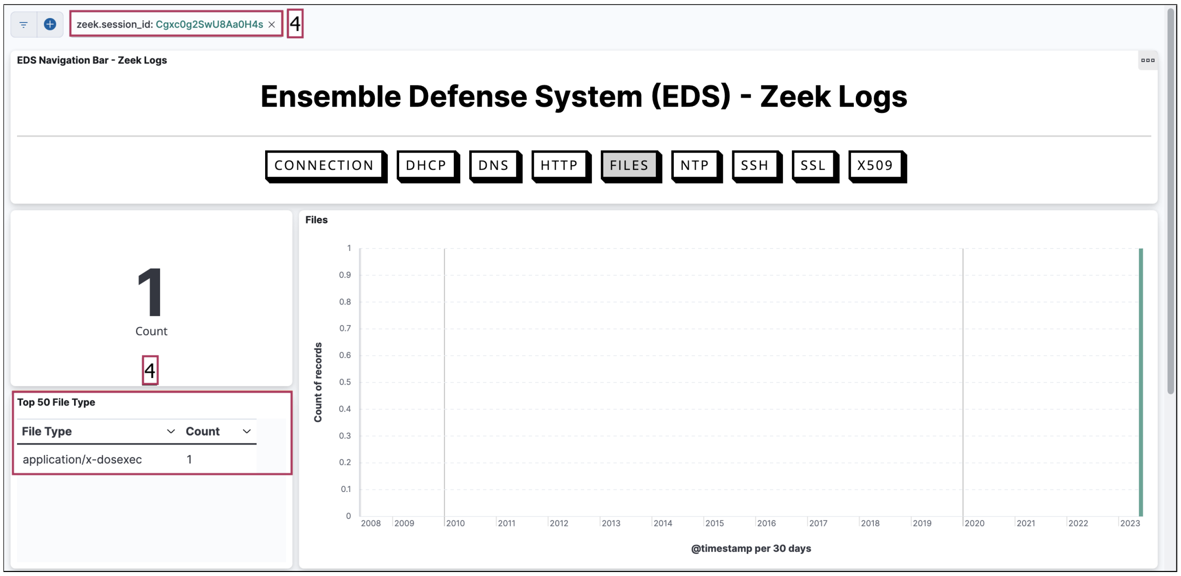
@timestamp	destinatio...	destinatio...	source.ip	source.port	network.pr...	search_session_id	zeek.c...	zeek.conn...	destinatio...	source.byt...	event.dura..
May 23, 2023 @ 00:01:59.067	194.62.42.206	80	10.9.10.102	58131	http	Find Associated Metadata Records	ShADad	Connection established, not …	16.5KB	648B	6.62 secc

2
zeek.session_id: Cgxc0g2SwU8Aa0H4s
3
EDS Navigation Bar - Zeek Logs
Ensemble Defense System (EDS) - Zeek Logs
CONNECTION
DHCP
DNS
HTTP
FILES
NTP
SSH
SSL
X509
Zeek
3
ZEEK Output Count
Zeek Log Types

Metadata Log Type	Count
connection	1
files	1
http	1

Figure 5.6: Part 2: Correlation Between Zeek Logs and Suricata Alert Logs

Lastly, the number 4 in Figure 5.7 shows the file log dashboard, wherein we can extract comprehensive information regarding all file-related activities. It is worth noting that our analysis identified a dangerous file type, namely x-dosexec. Thus, this correlation allows for identifying correlations between network flows and detecting potential attacks more efficiently.



zeek.session_id: Cgxc0g2SwU8Aa0H4s

Top 50 File Type

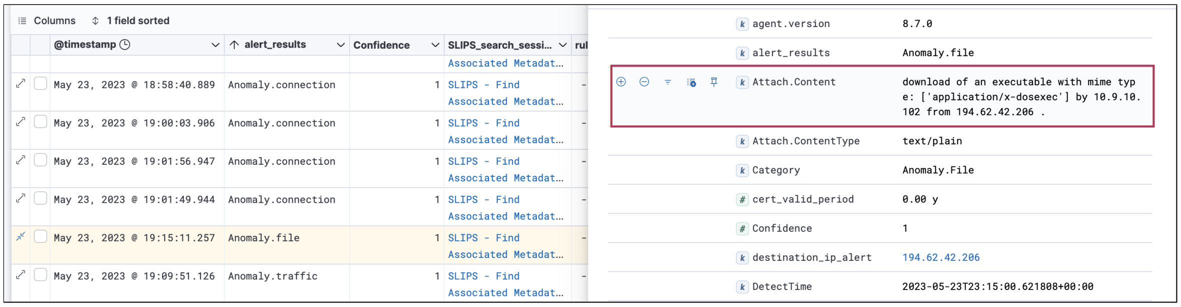
File Type	Count
application/x-dosexec	1

Figure 5.7: Part 3: Correlation Between Zeek Logs and Suricata Alert Logs

The `community_id` field in the Zeek, Suciata, and Slips outputs and `session_id` in Zeek logs are helpful when tracking and identifying suspicious network traffic. Using these fields, we can track and analyze the network traffic associated with an alert output generated by Suricata or Slips. Furthermore, due to the `community_id` and `session_id` use, we can group the network traffic associated with an alert, which allows us to identify patterns and correlations with other alerts and logs. As a result, we are able to provide a more comprehensive understanding of the attack.

5.3 Validation of Attack Severity and Urgency by the EDS

As shown in Figure 5.5 and 5.8, detecting the same attack by another IDS, such as Slips, reinforces and validates its significance. Additionally, it emphasizes that this is an actual and potentially threatening attack. When multiple IDS, such as Suricata, Zeek, and Slips, identify and alert the same attack, it emphasizes the severity and urgency of the situation.



Columns 1 field sorted

@timestamp	alert_results	Confidence	SLIPS_search_sessi...	rul
			Associated Metadat...	
May 23, 2023 @ 18:58:40.889	Anomaly.connection	1	SLIPS - Find Associated Metadat...	-
May 23, 2023 @ 19:00:03.906	Anomaly.connection	1	SLIPS - Find Associated Metadat...	-
May 23, 2023 @ 19:01:56.947	Anomaly.connection	1	SLIPS - Find Associated Metadat...	-
May 23, 2023 @ 19:01:49.944	Anomaly.connection	1	SLIPS - Find Associated Metadat...	-
May 23, 2023 @ 19:15:11.257	Anomaly.file	1	SLIPS - Find Associated Metadat...	-
May 23, 2023 @ 19:09:51.126	Anomaly.traffic	1	SLIPS - Find Associated Metadat...	-

agent.version	8.7.0
alert_results	Anomaly.file
Attach.Content	download of an executable with mime type: ['application/x-dosexec'] by 10.9.10.102 from 194.62.42.206 .
Attach.ContentType	text/plain
Category	Anomaly.File
cert_valid_period	0.00 y
Confidence	1
destination_ip_alert	194.62.42.206
DetectTime	2023-05-23T23:15:00.621808+00:00

Figure 5.8: Similar Attack Alerts Generated by Suricata and Slips

Chapter 6
CONCLUSION

The implemented Ensemble Defense System (EDS) that integrates Intrusion Detection Systems (IDS) and Security Information and Event Management (SIEM) has proven to be a promising approach to enhancing network security. The primary objective of this thesis is to design, develop, and evaluate a novel EDS that leverages the strengths of hybrid-based IDS technologies and SIEM. The EDS successfully proves its threats detection capabilities by integrating multiple detection technologies, including signature-based and behavioral-based IDS. This integration enables the system to leverage pre-defined patterns for detecting known attacks while analyzing network behavior to identify anomalies. By combining the strengths of both approaches, the EDS achieves effective threat detection. Throughout this thesis, we have explored the benefits of the hybrid approach through its implementation and evaluation. The results show that the EDS can effectively detect attacks and strengthen network security.

6.1 Contributions

This thesis has made significant contributions to the defense against cyber attacks. Firstly, an EDS has been designed, developed, and evaluated as a significant contribution to research in strengthening network security. The EDS integrates IDS with SIEM to provide a comprehensive defense solution. This research has contributed to evaluating the effectiveness of integrating multiple IDS technologies, signature-based and behavioral-based IDS, within the EDS. In addition, using SIEM enables the EDS to provide user-friendly interfaces to facilitate effective threat detection. Furthermore, this thesis evaluated the EDS's capability to detect various types of attacks.

6.2 Future Research

Future research in this area can focus on several aspects. Firstly, further enhancements can be made to this EDS to improve its accuracy and efficiency in detecting sophisticated attacks. This EDS's enhancements can involve incorporating advanced machine learning algorithms by adding a costume script to the hybrid-based IDS. Additionally, the rise in encrypted network communications challenges traditional IDS systems. Future research may take into their consideration developing scripts to handle encrypted traffic within the EDS. Hence, developing innovative approaches to analyzing encrypted traffic to detect malicious communications is necessary. In addition to the scripts, implementing a data management strategy is recommended, including periodically removing the data from the folder `/var/log/ids/*`, which is made to store both the Zeek logs and Suricata and Slips alert files. This practice will optimize the system's CPU and memory utilization to enhance the system's performance. Furthermore, conducting more extensive evaluation studies using real-world datasets would provide valuable insights into the system's performance.

REFERENCES

[1] G. Vigna and R. A. Kemmerer, “Netstat: A network-based intrusion detection system,” *J. Comput. Secur.*, vol. 7, p. 37âĂŞ71, jan 1999.

[2] M. Ozkan-Okay, R. Samet, Ã. Aslan, and D. Gupta, “A comprehensive systematic literature review on intrusion detection systems,” *IEEE Access*, vol. 9, pp. 157727–157760, 2021.

[3] Y. Otoum and A. Nayak, “As-ids: Anomaly and signature based ids for the internet of things,” *J. Netw. Syst. Manage.*, vol. 29, jul 2021.

[4] M. El Arass and N. Souissi, “Smart siem: From big data logs and events to smart data alerts,” vol. 8, pp. 3186–3191, 06 2019.

[5] Elastic, “elk.” https://www.elastic.co/elasticsearch/.

[6] Elastic, “Logstash.” https://www.elastic.co/logstash/, 2021.

[7] Elastic, “Kibana.” https://www.elastic.co/kibana/, 2021.

[8] Snort Project, “Snort.” https://www.snort.org/.

[9] The Zeek Development Team, “Zeek network security monitor.” https://zeek.org/, 2021.

[10] OSSEC, “Ossec - open source host-based intrusion detection system.” https://www.ossec.net/, 2021.

[11] O. Negoita and M. Carabas, “Enhanced security using elasticsearch and machine learning,” pp. 244–254, 07 2020.

[12] Elastic, “What is elasticsearch machine learning?.” https://www.elastic.co/what-is/elasticsearch-machine-learning.

[13] D. F. Priambodo, Amiruddin, and N. Trianto, “Hardening a work from home network with wireguard and suricata,” in *2021 International Conference on Computer Science and Engineering (IC2SE)*, vol. 1, pp. 1–4, 2021.

[14] WireGuard, “WireGuard.” https://www.wireguard.com/.

[15] Open Information Security Foundation, “Suricata.” https://suricata.io/.

[16] "Nmap - the Network Mapper." https://nmap.org/.

[17] A. Esseghir, F. Kamoun, and O. Hraiech, "Aker: An open-source security platform integrating ids and siem functions with encrypted traffic analytic capability," *Journal of Cyber Security Technology*, vol. 6, no. 1-2, pp. 27–64, 2022.

[18] A. R. Muhammad, P. Sukarno, and A. A. Wardana, "Integrated security information and event management (siem) with intrusion detection system (ids) for live analysis based on machine learning," *Procedia Computer Science*, vol. 217, pp. 1406–1415, 2023. 4th International Conference on Industry 4.0 and Smart Manufacturing.

[19] Stratosphere Project, "Stratospherelinuxips - intrusion detection and prevention system." https://github.com/stratosphereips/StratosphereLinuxIPS.

[20] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks," *Expert Systems with Applications*, vol. 29, no. 4, pp. 713–722, 2005.

[21] K. Q. Yan, S.-C. Wang, and C. W. Liu, "A hybrid intrusion detection system of cluster-based wireless sensor networks," 2009.

[22] A. Abduvaliyev, S. Lee, and Y.-K. Lee, "Energy efficient hybrid intrusion detection system for wireless sensor networks," in *2010 International Conference on Electronics and Information Engineering*, vol. 2, pp. V2–25–V2–29, 2010.

[23] H. Sedjelmaci, S. M. Senouci, and M. Feham, "Intrusion detection framework of cluster-based wireless sensor network," in *2012 IEEE Symposium on Computers and Communications (ISCC)*, pp. 000857–000861, 2012.

[24] A. Reshamwala and K. Bhowmik, "A review of intrusion detection system using neural network and machine learning technique," *International Journal of Computer Science and Engineering*, vol. 2, no. 2, pp. 62–67, 2013.

[25] Z. Wang and X. Li, "Intrusion prevention system design," in *Proceedings of the International Conference on Information Engineering and Applications (IEA) 2012* (Z. Zhong, ed.), (London), pp. 375–382, Springer London, 2013.

[26] U. H. Rao and U. Nayak, *Intrusion Detection and Prevention Systems*, pp. 225–243. Berkeley, CA: Apress, 2014.

[27] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns," *IEEE Transactions on Computers*, vol. 63, pp. 807–819, 2014.

[28] The OISF development team, "Suricata: Open source next generation intrusion detection and prevention engine." https://suricata-ids.org/.

[29] B. Caswell, J. Beale, and A. Baker, *Snort IDS and IPS Toolkit*. Syngress, 2007.

[30] V. Paxson, “Bro: a system for detecting network intruders in real-time,” *Computer Networks*, vol. 31, no. 23, pp. 2435–2463, 1999.

[31] J. Veeramreddy, V. Prasad, and K. Prasad, “A review of anomaly based intrusion detection systems,” *International Journal of Computer Applications*, vol. 28, pp. 26–35, 08 2011.

[32] P. GarcÃŋa-Teodoro, J. DÃŋaz-Verdejo, G. MaciÃą-FernÃąndez, and E. VÃązquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges,” *Computers & Security*, vol. 28, no. 1, pp. 18–28, 2009.

[33] S. Garcia, K. Babayeva, A. Gomaa, and O. Lukas, “Demo of slips, a free-software ips with behavioral machine learning detection.” https://www.ieeelcn.org/prior/LCN46/lcn46demos/Demo_2_1570753964.pdf, 2020.

[34] A. Buecker, J. Amado, D. Druker, C. Lorenz, F. Muehlenbrock, R. Tan, and I. Redbooks, *IT Security Compliance Management Design Guide with IBM Tivoli Security Information and Event Manager*. IBM redbooks, IBM Redbooks, 2010.

[35] Elastic, “Beats.” https://www.elastic.co/beats/, 2021.

[36] antirez, “hping.” https://github.com/antirez/hping.

[37] sqlmap, “sqlmap.” https://sqlmap.org/.

[38] Elastic, “filebeat.” https://www.elastic.co/beats/filebeat, 2021.

[39] sullo, “nikto.” https://github.com/sullo/nikto.

[40] “Ping.” https://ping.com/en-us/.

[41] Stratosphere IPS, “Stratosphere IPS.” https://www.stratosphereips.org/.

[42] Docker, “Docker documentation.” https://docs.docker.com/.

[43] Docker, “Docker compose documentation.” https://docs.docker.com/compose/, 2021.

[44] Sarah Alh, “EDS.” https://github.com/SarahAlh/EDS.

[45] P. Matousek, I. Rudolfova, O. Rysavy, and V. Malombe, *On Reliability of JA3 Hashes for Fingerprinting Mobile Applications*, pp. 1–22. 02 2021.

[46] Redis Labs, “Redis.” https://redis.com/.

[47] VirusTotal, “Virustotal.” https://www.virustotal.com/gui/home/search.

[48] J. M. Stewart, *Threats, Attacks, and Vulnerabilities*, pp. 1–122. 2021.

[49] B. Duncan, “Malware traffic analysis.” http:/malware-traffic-analysis.net.

Appendix A

SHELL SCRIPT FOR LAUNCHING MULTIPLE ATTACKS

```
#!/bin/bash
echo "Valid choices are:"
echo "PORT SCANNING:"
echo "  1. NMAP"
echo "  2. Ping"
echo "  3. Nikto"
echo "DoS Attack:"
echo "  4. hping"
echo "Privilege Escalation"
echo "  5. SQLmap"
echo "  6. All Above"
read -p "Enter the IP address: " ip
read -p "Enter the tool number: " attack

if [ "$attack" -eq 1 ]; then
    echo "Starting NMAP: "
    nmap -sS "$ip" -p 1-1000

elif [ "$attack" -eq 2 ]; then
    echo "Starting Ping: "
    ping -c 10 "$ip"

elif [ "$attack" -eq 3 ]; then
    echo "Starting Nikto: "
    nikto -h "$ip"

elif [ "$attack" -eq 4 ]; then
    echo "Starting hping"
    hping3 -c 100 -p 21 -w 64 -d 120 --flood --rand-source "$ip"

elif [ "$attack" -eq 5 ]; then
    echo "Starting SQLmap"
    sqlmap -u "$ip:80"

elif [ "$attack" -eq 6 ]; then
    echo "Run All Attacks"
    nmap -sS "$ip" -p 1-1000
    ping -c 10 "$ip"
    nikto -h "$ip"
    hping3 -c 100 -p 21 -w 64 -d 120 --flood --rand-source "$ip"
    sqlmap -u "$ip:80"
else
    echo "It is not a valid choice"
fi
echo "Done"
```