MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

Research and implementation of network intrusion detection system based on ELK Stack

System Team	
Group Members	Nguyễn Quốc Huy - SE1720186 Võ Trọng Đức- SE172224 Dương Thành Lộc - SE171551 Huỳnh Trí Đức - SE171080 Nguyễn Đăng Khoa - SE171556
Supervisor	Lê Tiến Dũng
Ext Supervisor	
Capstone Project Code	GSP25IA05

- Ho Chi Minh City, January 2025 -

Table of contents

List of Tables

List of Figures

Chapter 1: INTRODUCTION

1.1- Project Information

Capstone project name: Research and implementation of network intrusion detection system based on ELK Stack.

Project code: GSP25IA05

Project Group Name: GSP25IA05

Timeline: 07/01/2025 - 29/5/2025

1.2- The participants

1.2.1. Name of the CP

Full name	Phone	Email	Title
Lê Tiến Dũng	0906794454	dunglt92@fe.edu.vn	Lecturer

Table 1.2.1. Supervisor

1.2.2. Team Members

Full name	Student code	Email	Role
Nguyễn Quốc Huy	SE172186	huynqse172186@fpt.edu.vn	Leader
Võ Trọng Đức	SE172224	ducvtse172224@fpt.edu.vn	Member
Huỳnh Trí Đức	SE171080	duchtse171080@fpt.edu.vn	Member
Dương Thành Lộc	SE171551	locdtse171551@fpt.edu.vn	Member
Nguyễn Đăng Khoa	SE171556	khoandse171556@fpt.edu.vn	Member

Table 1.2.2. Team members

1.3- The Problems

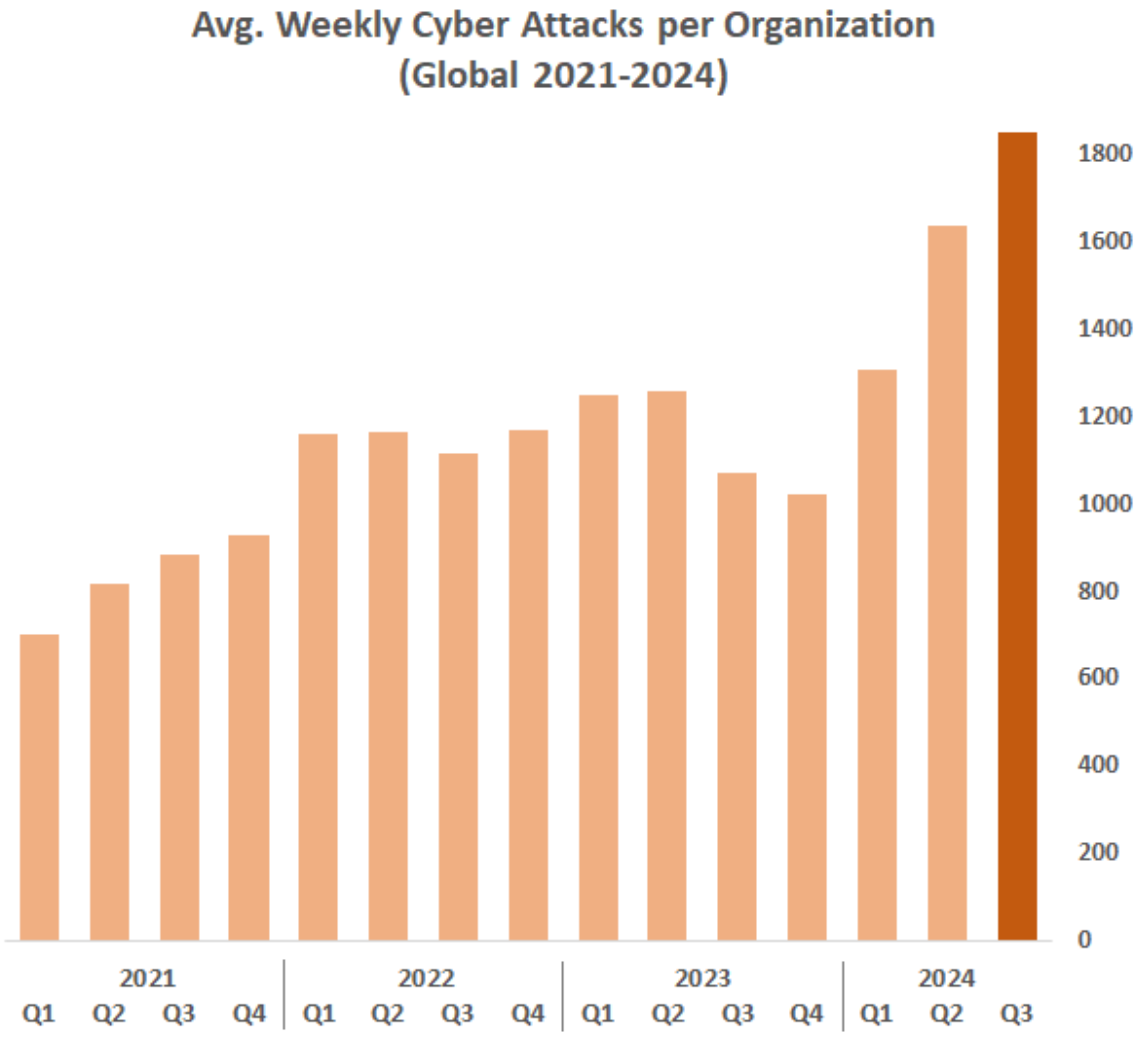
1.3.1. Overview

Organizations face difficulties in detecting and preventing network intrusions as cyber threats evolve. Traditional security measures such as firewalls and antivirus software are effective against known threats but may fail to identify sophisticated and stealthy attacks. Advanced threats, for example, zero-day exploits, can bypass these defenses and remain undetected, posing significant risks to organizations.

Web services, corporate networks, and digital platforms have become prime cyber-attack targets. Malicious activities can go unnoticed without comprehensive monitoring and detection mechanisms until substantial damage has already been done. To mitigate these risks, organizations require proactive, real-time solutions to detect and respond to network intrusions before they escalate.

One significant problem in intrusion detection mechanisms is the volume of security logs generated by various sources, including firewalls, servers, routers, intrusion detection systems (IDS), and endpoint devices. Analyzing this data in real-time is complex, as the high volume of logs can slow down the system and obscure critical security threats.

Many existing security tools struggle to efficiently integrate and analyze data from multiple sources. Organizations remain vulnerable to severe cyber threats such as data breaches, malware infections, and unauthorized access without a centralized system to monitor and correlate security events in real-time. To address these challenges, businesses must use advanced threat detection systems capable of aggregating, analyzing, and responding to security incidents in real-time, ensuring a robust defense against modern cyber threats.



Source: Checkpoint (A Closer Look at Q3 2024: 75% Surge in Cyber Attacks Worldwide - Check Point Blog)

Figure 1.3.1 Average Weekly Cyber-Attacks Per Organisation

Figure 1.3.1 refers to the average number of cyber attacks per week around the world.

1.3.2. Solution

The proposed solution is the implementation of a Network Intrusion Detection System (NIDS) using the ELK Stack (Elasticsearch, Logstash, and Kibana) as a centralized log management platform. The Elastic Stack will collect and aggregate logs from various devices and security tools, providing real-time monitoring. This centralized approach will allow the system to detect unusual or malicious activity, generate alerts when potential threats are identified, and store historical data for post-incident analysis.

This project will demonstrate how the ELK Stack can effectively integrate into a broader cybersecurity framework. By using its real-time log processing, search capabilities, and data visualization tools, organizations can improve their ability to detect, respond to, and mitigate network intrusion attempts, reducing the risk of cyber-attack damage.

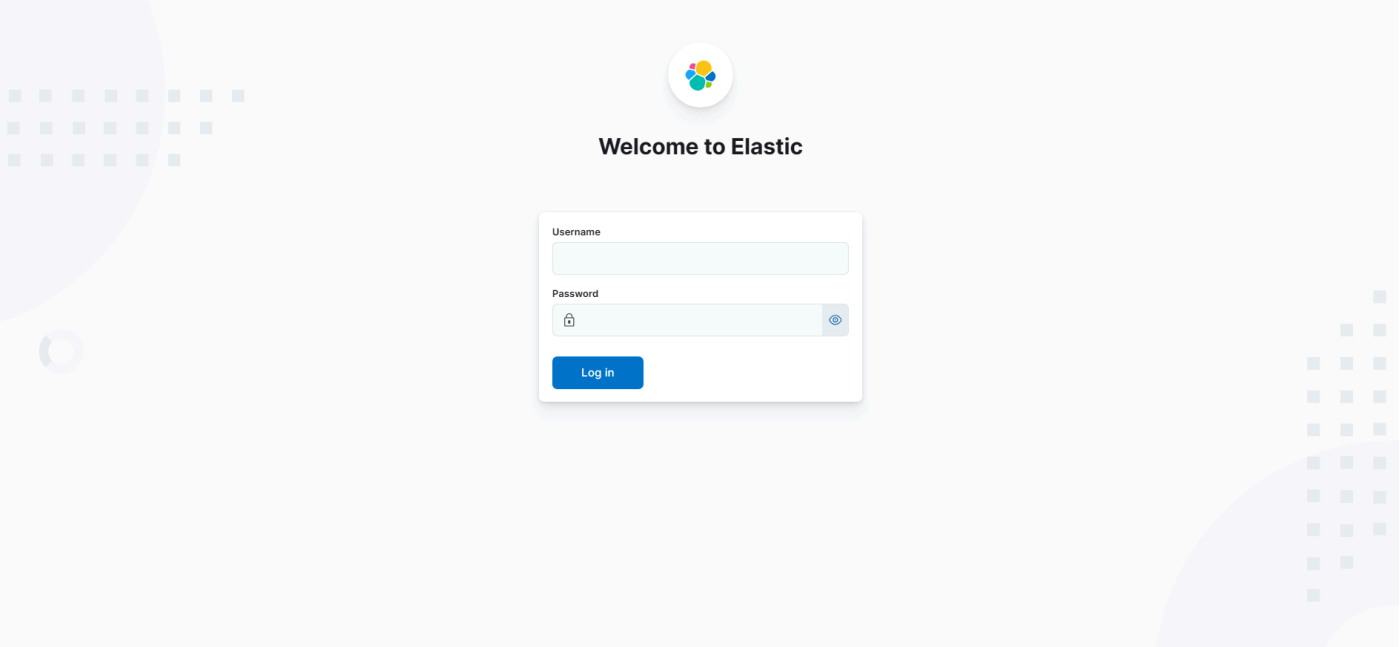


Figure 1.3.2 Elastic login page

Chapter 2: IA PROJECT MANAGEMENT PLAN

2.1- Problem Setting

2.1.1. Name of the CP

English: Research and implementation of network intrusion detection system based on ELK Stack.
Vietnamese: Nghiên cứu và triển khai hệ thống phát hiện xâm nhập mạng dựa trên ELK Stack.

2.1.2. Problem Abstraction

With cyberattacks becoming sophisticated nowadays, traditional security measures like firewalls and antivirus software may fail to detect advanced threats such as zero-day exploits. Organizations generate large amounts of logs from firewalls, servers, and endpoint devices, making it challenging to monitor and analyze security events in real-time. Without an effective system to process this data, detecting and preventing intrusions becomes harder.

This project focuses on implementing a Network Intrusion Detection System (NIDS) that enables real-time monitoring, machine learning to detect cyber-attacks, automated threat alerts, and historical data analysis. By improving visibility into network activity and potential threats, the system enhances the ability to detect, analyze, and respond to cybersecurity threats efficiently.

2.1.3. Project Overview

2.1.3.1. The Current Situation

Nowadays, cyber-attacks are becoming sophisticated. They use different techniques, such as zero-day exploits and rootkits, to infiltrate the systems. Attackers develop different strategies to bypass security measures, making detection and prevention more challenging.

While traditional security tools like firewalls or antivirus software can detect and block known threats, they often fail to detect stealthy and modern attacks. These threats can come from different sources, including threat actors, cybercriminal groups, and even insiders, to compromise an organization or corporation's sensitive data and critical infrastructure.

One of the challenges in intrusion detection is the high volume of security logs generated by firewalls, web servers, routers, and endpoint devices. Without an effective system for real-time analysis, security teams may struggle to detect threats, leaving organizations vulnerable to data breaches, financial losses, system disruptions, and

reputational damage.

Given these risks, organizations need a proactive, real-time intrusion detection system that can efficiently collect, analyze, and correlate security events across different network layers to identify potential attacks before they escalate.

2.1.3.2. The Proposed Solution

To address these challenges, this project proposes implementing a Network Intrusion Detection System (NIDS) using the ELK Stack (Elasticsearch, Logstash, and Kibana) as the central log management and monitoring platform.

The ELK Stack offers practical tools for organizations to monitor the systems efficiently. Its core features include:

- Elastic Stack offers a real-time solution that categorizes incoming data streams and automatically routes them into designated indices based on predefined rules. By leveraging ingest pipelines and tools like Logstash, messages can be parsed, enriched with additional metadata, and assigned to the appropriate categories as they arrive.

- Elasticsearch provides high-speed, full-text search capabilities with advanced query and filtering options. It allows users to explore data efficiently, identify patterns, and troubleshoot issues.

- Kibana Dashboards enables the creation of interactive and customizable dashboards that provide visual summaries of log event data. Kibana dashboards can include charts, graphs, and tables to monitor system performance and detect real-time anomalies.

- The Kibana alerting system lets users define and configure alerts when meeting specific conditions or thresholds. Alerts can be delivered via email, Slack, or other channels, ensuring fast responses to potential threats.

- Elasticsearch indexes allow efficient data storage and management with customizable retention policies, replication, and sharding. This enables organizations to optimize storage and access for specific log types.

- Elastic Agent is a unified agent that collects logs, metrics, and security data from endpoints. It simplifies deployment and management by utilizing multiple Beats functionalities in a single agent. Fleet Server centrally manages Elastic Agents, ensuring streamlined configuration, monitoring, and updates across all agents.

- Logstash pipelines allow data enrichment, transformation, and routing during ingestion. Custom processing rules allow the system to handle different types of events.

ELK Stack enables organizations to detect, respond, and prevent cyber threats in real-time, reducing the risk of system downtime and financial losses.

2.1.3.3. Boundaries of the Solution

The scope of this project is the implementation of a comprehensive Security Information and Event Management (SIEM) system using the ELK Stack to collect logs from various sources, including firewalls, Intrusion Detection Systems (IDS), and endpoint devices. The primary focus will be on monitoring the web server, where users interact the most, to track network traffic, detect unusual behavior, identify potential attacks, and generate alerts for system administrators. This real-time monitoring capability will help ensure timely threat detection and response, enhancing overall network security.

2.1.3.4. Development Environment

We used to deploy the system, including the Ubuntu server, web server, Firewall, and IDS (Intrusion Detection System). Before deploying the network intrusion detection system, log forwarding agents must be installed and configured on all devices and systems that need monitoring. These agents will collect log data (from firewalls, web servers, network devices, and EDR) and send that data to Elastic Stack for further processing.

We will deploy and configure Elasticsearch and Kibana on the Ubuntu server. This server can collect and store network logs and provide feedback if any suspicious activity is detected. Typical agents for network devices might include Filebeat, Nxlog, or Syslog.

For the test run, we will use Nmap, Burp Suite, and other penetration testing tools to simulate attacks, testing and improving the monitoring system.

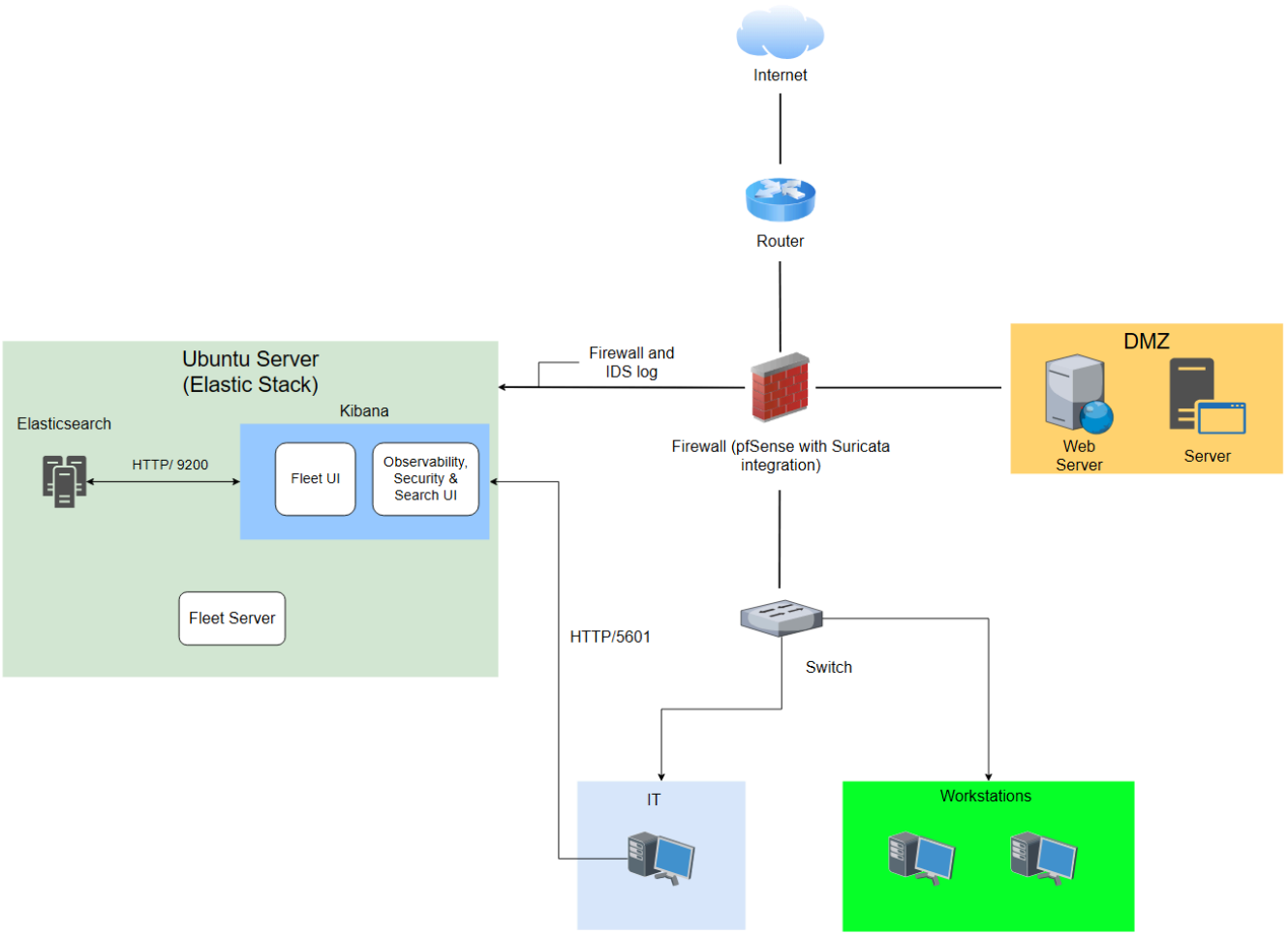


Figure 2.1.3.4 Project overview diagram

2.2- Project Organization

2.2.1. Solution Process Model

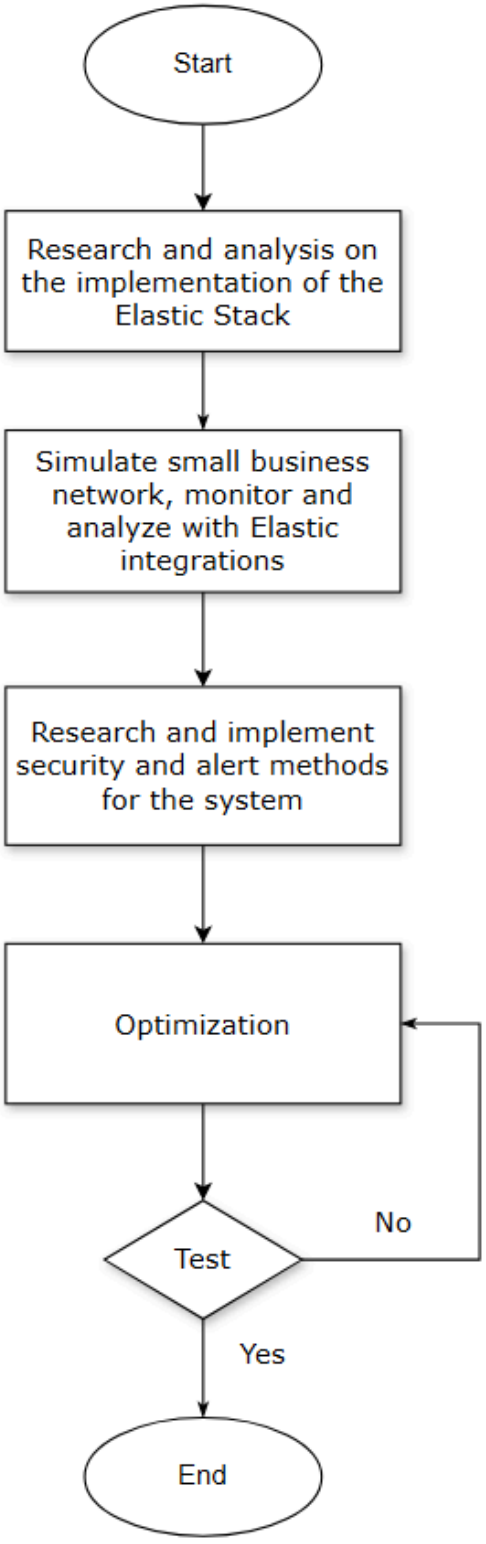


Figure 2.2.1 Solution Process Model

Step 1) Research and Analysis:

- Review IDS (Intrusion Detection System) solutions and Elastic Stack's log monitoring capabilities.
- Set up a virtual environment simulating a network with multiple devices generating traffic.

Step 2) IDS and Elastic Setup:

- Install and configure Elastic Stack (Elasticsearch, Logstash, Kibana, and Elastic Agent).
- Set up Suricata to send logs to Elastic via Filebeat.
- Configure Logstash pipelines and Kibana dashboards for log parsing and visualization.

Step 3) Testing and Simulation:

- Simulate network intrusions (e.g., DDoS, port scanning) using tools like Metasploit or Nmap.
- Verify Elastic receives and processes IDS logs accurately.
- Develop alerts and dashboards for real-time attack monitoring.

Step 4) Evaluation and Refinement:

- Assess solution effectiveness through tests and simulations.
- Collect feedback, refine configurations, and optimize detection rules iteratively.

2.2.2. Roles and Responsibilities

Phase	Full name	Role	Responsibilities
Planning	Nguyễn Quốc Huy	Leader, Technical, Consultant, Document.	Delegating tasks Analyzing goals Planning milestones Monitoring working progress
	Dương Thành Lộc	Technical, Consultant, Document.	Reviewing the plan Identify project process model Identify project environment
	Huỳnh Trí Đức	Technical, Consultant, Document.	Reviewing the plan Identifying working hour Infrastructure engineer
	Võ Trọng Đức	Technical, Consultant, Document.	Reviewing the plan Tracking on tasks Identify project environment
	Nguyễn Đăng Khoa	Technical, Consultant, Document.	Reviewing the plan Gather relevant academic literature.
	Nguyễn Quốc Huy	Leader, Technical, Consultant, Document.	Oversee the research phase, provide guidance, and ensure the team stays on track.

Research and Analysis	Dương Thành Lộc	Technical, Consultant, Document.	Reviewing the plan Conduct in-depth research on feasibility testing, analysis of technical requirements and system configuration.
	Huỳnh Trí Đức	Technical, Consultant, Document.	Reviewing the plan Conduct in-depth research on feasibility testing, analysis of technical requirements and system configuration.
	Võ Trọng Đức	Technical, Consultant, Document.	Reviewing the plan Conduct in-depth research on feasibility testing, analysis of technical requirements and system configuration
	Nguyễn Đăng Khoa	Technical, Consultant, Document.	Reviewing the plan Collect relevant academic literature. Analyze network intrusion detections and network attacks.
Design and Architecture	Nguyễn Quốc Huy	Leader, Technical,	Delegating tasks Analyzing goals

		Consultant, Document.	Planning milestones Monitoring working progress
	Huỳnh Trí Đức	Technical, Consultant, Document.	Reviewing the plan Technical researcher Infrastructure engineer
	Dương Thành Lộc	Technical, Consultant, Document.	Reviewing the plan Technical researcher Documentation
	Võ Trọng Đức	Technical, Consultant, Document.	Reviewing the plan Infrastructure consultant Documentation
	Nguyễn Đăng Khoa	Technical, Consultant, Document.	Reviewing the plan Infrastructure consultant Documentation
Implementation and Testing	Nguyễn Quốc Huy	Leader, Technical, Consultant, Document.	Delegating tasks Analyzing goals Planning milestones Monitoring working progress
	Huỳnh Trí Đức	Technical, Consultant, Document.	Reviewing the plan Technical researcher Infrastructure engineer
	Dương Thành Lộc	Technical, Consultant, Document.	Reviewing the plan Technical researcher Documentation

	Võ Trọng Đức	Technical, Consultant, Document.	Reviewing the plan Infrastructure consultant Documentation
	Nguyễn Đăng Khoa	Technical, Consultant, Document.	Reviewing the plan Infrastructure consultant Documentation
Evaluation and Refinement	Nguyễn Quốc Huy	Leader, Technical, Consultant, Document.	Delegating tasks Analyzing goals Planning milestones Monitoring working progress
	Huỳnh Trí đức	Technical, Consultant, Document.	Reviewing the plan Technical researcher Infrastructure engineer
	Dương Thành Lộc	Technical, Consultant, Document.	Reviewing the plan Technical researcher Documentation
	Võ Trọng Đức	Technical, Consultant, Document.	Reviewing the plan Infrastructure consultant Documentation
	Nguyễn Đăng Khoa	Technical, Consultant, Document.	Reviewing the plan Infrastructure consultant Documentation

Table 2.1. Roles and Responsibilities

2.2.3. Tools and Techniques

No	Software	Main Functions	Details
1	Elastic Stack (ELK)	Centralized log management and analysis	Collects, analyzes, and monitors logs from systems and applications, providing real-time alerts and reporting using Elasticsearch, Logstash, and Kibana.
2	Suricata	Intrusion Detection System (IDS)	Detects network attacks and anomalies by analyzing network traffic based on predefined rules.
3	Ubuntu	Open-source operating system based on Linux	Provides the platform for deploying the intrusion detection system (including Elastic Stack and IDS).
4	pfSense	Firewall and network security	Provides advanced firewall capabilities, traffic shaping, and VPN support to enhance network security and monitor traffic.
5	Syslog	Log collection and management	Facilitates the transmission of source logs to a remote destination using predefined filters, also allows customization and can facilitate almost any logging need.
6	PNETLab	Virtual network simulation	Runs the lab environment locally, allowing for virtualized network simulations and IDS testing.
7	Python3	Programming language	Preferred language in AI and machine learning, aid in developing predictive models and AI applications
8	Web Server	Web server platform	Hosts web applications and serves as a logging source for ELK, where web traffic logs are analyzed for anomalies.
9	Telegram	Receive instant notifications via Telegram messages	The Telegram Bot API integrates with ELK, allowing rapid notifications to users when an alert is triggered.

Table 2.2. Tools and Techniques

2.3- Project Management Plan

2.3.1. Tasks

2.3.1.1. Brainstorming

Description: The brainstorming phase aims to generate ideas, define the scope, and outline the project objectives. This includes meetings with stakeholders, technical discussions, and planning to determine the system architecture and core functionalities of ELK Stack & Suricata for network security monitoring.

Deliverables: Project scope document, initial project architecture diagram, list of potential tools and technologies, and individual opinions and ideas documented during the brainstorming session.

Resources Needed: Whiteboard or virtual brainstorming tools - access to documentation and case studies on ELK and Suricata.

Dependencies and Constraints: Take input from all stakeholders to define project goals clearly. Each team member is assigned different tasks based on their expertise and interest.

Risks: Misalignment of project goals due to unclear requirements. Incomplete brainstorming, missing potential risks or challenges.

2.3.1.2. Researching

Description: Each team member researches the implementation of NIDS (Network Intrusion Detection System) using Suricata and log analysis with ELK Stack. Study how Suricata detects network threats and integrates them with ELK. Research how Elasticsearch, Logstash, and Kibana (ELK) process, store, and visualize logs from Suricata. Analyze current cybersecurity threats and how Suricata helps detect them. Review regulatory and compliance requirements for network monitoring and logging.

Deliverables: Detailed documentation on Suricata's threat detection and ELK integration.Best practices for implementing network security monitoring.

Resources needed: Access to technical documentation and user guides for Suricata and ELK; research articles on intrusion detection systems; and use the Internet to access latest security threat information.

Dependencies and constraints: Availability of external research and expertise. Time is required to understand customization options in ELK and Suricata.

Risks: Insufficient or outdated research documentation. Misconfigurations due to a lack of understanding of how Suricata sends logs to ELK and how logs are processed.

2.3.1.3. Developing Solutions

Description: Set up the ELK Stack for log storage and integrate Suricata as the Intrusion Detection System (IDS). Develop custom parsing and filtering rules in Logstash to process network logs efficiently—Configure Suricata to detect network threats and forward alerts to Elasticsearch for real-time analysis. Build Kibana dashboards to monitor and visualize security events. Implement security policies to ensure compliance with legal and organizational requirements.

Deliverables: Custom Suricata IDS rules for threat detection, Kibana dashboards and real-time alerts, and documentation on system configuration and architecture.

Resources Needed: Virtual servers for deploying ELK Stack and Suricata IDS. Development team for system configuration. Network equipment for testing (firewalls, routers). ELK and Suricata rule creation tools.

Dependencies and Constraints: Ensure the Installation and correct configuration of each component (ELK Stack, Suricata IDS). It must work within existing network architecture and security policies.

Risks: Possible integration issues between ELK Stack and Suricata IDS. System performance might be inadequate for real-time log analysis on large networks. Network security policies might restrict specific configurations or log data.

2.3.1.4. Evaluating

Description: Test the system's ability to detect network intrusion attempts using sandbox-simulated attacks. Conduct performance tests to ensure ELK Stack and Suricata can work simultaneously and scale effectively. Review and refine system configurations to optimize detection accuracy.

Deliverables: Write an evaluation report on system performance, threat detection accuracy, and scalability using ELK Stack and Suricata—a system optimization report with recommended adjustments to enhance efficiency and detection capabilities.

Resources Needed: Access to a testing environment (virtual networks or physical testbeds) for simulating attacks. The tools for generating network threats

include Suricata's built-in testing capabilities, penetration testing frameworks (e.g., Metasploit), and traffic simulation tools (e.g., Tcpreplay).

Dependencies and Constraints: Access to ELK Stack logs and Suricata security alerts from network devices. Limited time for comprehensive testing of all potential threat

Risks: The system might fail to detect certain types of attacks due to misconfigured rules.

2.3.2. Task Sheet: Assignments and Timetable

Task name	Start date	End date
Total Estimate Capstone Project	07/01/2025	29/5/2025
Project Initiating	07/01/2025	08/01/2025
Delegating tasks	08/01/2025	09/01/2025
Collection of documents and resources about ELK	09/01/2025	10/01/2025
Writing and submitting report 1	10/01/2025	11/01/2025
Discussing risk assessment and management	11/01/2025	20/01/2025
Research about IDS	11/01/2025	20/01/2025
Research about pfSense	11/01/2025	20/01/2025
Research about Web server	11/01/2025	20/01/2025
Writing and submitting report 2	20/01/2025	21/01/2025
Analyzing how ELK works	21/01/2025	24/01/2025

Identifying risks of ELK	24/01/2025	25/01/2025
Reviewing works and documents	3/02/2025	4/02/2025
Building development environment	5/02/2025	6/02/2025
Writing and submitting report 3	7/02/2025	8/02/2025
Defining core functions and requirements for demo	9/02/2025	10/02/2025
Building core functions	11/02/2025	10/03/2025
Testing core functions	10/03/2025	12/03/2025

Table 2.3. Assignments and Timetable

2.3.3. All Meeting Minutes

Subject	IAP491T
Date	07/01/2025 - 29/5/2025
Lecturer	Lê Tiến Dũng
Time	17h45 - 20h00
Location	005
Attendees	Nguyễn Quốc Huy Dương Thành Lộc Huỳnh Trí Đức Võ Trọng Đức Nguyễn Đăng Khoa
Absent	

Table 2.4. All Meeting Sessions

Chapter 3: RISK ASSESSMENT (RA)

3.1. The Need for Assessment

Risk assessment is essential in any network security project, especially when setting up a Network Intrusion Detection System (NIDS). It helps identify weaknesses in the system, making it easier to detect threats, prevent attacks, and reduce damage from security breaches. A well-planned risk assessment improves network data's security, reliability, and protection.

Risk assessment also helps in making better decisions during the system's setup and operation. By understanding potential risks, teams can use resources wisely, focus on the most important security measures, and create practical solutions to prevent cyber threats. A thorough assessment ensures that ELK + Suricata will work effectively in monitoring, detecting, and responding to network intrusions in the long run.

3.2. Identify Critical Assets

Identifying critical assets helps define the key network infrastructure components that must be protected. Critical assets in an ELK-based intrusion detection system include servers, databases, network devices, and log management systems, which are high-risk cyberattack points. Focused monitoring of these assets allows early detection of security incidents, ensuring protecting sensitive information and minimizing damage from cyber threats.

Classification	Information Assets	Asset Examples	Security Imperative
Restricted Information	Customer Database	- Customer database (MySQL, MongoDB) - Sensitive payment information	- Strong encryption - Strict access control - Continuous monitoring and incident reporting
Confidential Information	Business Strategy	- Business strategy documents - Internal emails among management	- Role-based access control (RBAC) - Periodic auditing and monitoring - Logging to detect anomalies
Internal Information	Employee Data	- Employee directory - Internal contact information	- Network segmentation - Moderate access control - Activity monitoring and access management

Table 3.1. Information Asset Classification

3.3. Risk Identification

3.3.1. Threat Identification

In the ELK-based intrusion detection system, threats refer to actions or behaviors that exploit system vulnerabilities, leading to unauthorized access, data leak, or service disruption. These threats can arise from network misconfigurations, human error, or unauthorized access.

Potential threats include:

- Unauthorized Access Attempts → Brute-force attacks, credential stuffing
- Malicious Network Traffic → Port scanning, exploitation attempts
- Data Exfiltration → Attackers stealing sensitive data
- Denial of Service (DoS) Attacks → Flooding network resources with malicious requests
- Compliance Violations and Privacy Breaches → Unsecured storage of sensitive data

3.3.2. Vulnerability Identification

In addition to threats, vulnerabilities within an ELK + Suricata-based NIDS must be considered, as attackers can exploit them.

Key vulnerabilities:

- Misconfigured logging and monitoring rules in ELK → Incomplete or incorrect rule sets may allow threats to go undetected.
- Evasion techniques by attackers → Sophisticated attackers may modify attack patterns to bypass Suricata rules.
- Limited integration with external threat intelligence feeds → Without real-time updates, the system may not detect new threats.
- Resource-intensive ELK performance → Large-scale log processing may cause system slowdowns if not optimized properly.

3.4. Risk Analysis

Risk analysis categorizes and assesses risks based on their likelihood and potential impact. This method forms the basis for designing a cost-effective security program. It helps identify high-impact risks that require careful management and those that can be accepted without stringent controls.

3.4.1. Impact Assessment

The impact assessment involves evaluating the potential consequences if a threat successfully exploits a vulnerability.

Risk Level		Likelihood Level		
		Low	Moderate	High
Impact Level	High	Moderate	Moderate	High
	Moderate	Low	Moderate	Moderate
	Low	Low	Low	Moderate

Table 3.2. Impact Levels Definitions

3.4.2. Likelihood Assessment

The system is exposed to several potential threats and vulnerabilities that could compromise security. These include Denial of Service (DoS) attacks, network scanning, exploit bypass, insider threats, and zero-day exploits. Each threat has associated vulnerabilities, such as resource exhaustion, weak authentication mechanisms, insufficient access controls, and unpatched system vulnerabilities. The impact of these threats ranges from service downtime and financial loss to unauthorized data access and system compromise.

Level	Probability Characteristic	Range of Probability
Extremely High	Very often, occur more frequently than 10% of the time/cases	0.8 - 1
High	Quite often, occur between 1% - 10% of the time/cases	0.55 – 0.8
Moderate	Many happen, occur between 0.1% - 1% of the time/cases	0.3 – 0.55
Low	Rare, occur less than 0.1% of the time/cases	0 – 0.3

Table 3.3. Likelihood Levels

3.4.3. Risk Determination (Rating)

Based on the impact and likelihood assessments, the risks were determined and assigned ratings according to a risk-level matrix.

3.4.3.1. Risk-Level Matrix

The risk-level matrix classifies risks into different levels based on their impact and likelihood. In the future, the risk matrix can be utilized as a risk register for continual risk monitoring and analysis throughout the project. The following matrix was used:

Consequence / Likelihood	Small	Moderate	Severe	Catastrophic
Low	Low	Low	Low	Moderate
Moderate	Low	Moderate	Moderate	High
High	Low	Moderate	High	High
Extreme High	Moderate	High	High	High

Table 3.4. Risk-level Matrix

3.4.3.2. Description of Risk Level

High: Risks with high impact and high likelihood require immediate attention and robust mitigation strategies.

Moderate: Risks with moderate impact and likelihood should be monitored and mitigated as part of the overall project plan.

Low: Risks with low impact and likelihood can be monitored but may not require immediate action.

3.5. Control Identification and Assessment

3.5.1. Control Identification

This section details the security measures developed to detect and mitigate network intrusion attempts. The focus is to protect and secure the systems monitored by the ELK-based intrusion detection system, enhancing network protection and minimizing the impact of cyber attacks.

3.5.1.1. Log Analysis and Anomaly Detection

Utilize log-based analysis techniques within the Elastic Stack to identify suspicious network activities. This includes detecting abnormal traffic patterns, unauthorized access attempts, or unexpected changes in system behavior.

By implementing anomaly detection mechanisms, the system can flag deviations from regular network traffic that may indicate intrusion attempts or malicious activity.

3.5.1.2. Access Control and Principle of Least Privilege

Utilize access control methods to ensure only authorized users can access the system logs and sensitive information. Implement the least privilege principle by granting users only the necessary permissions to perform their tasks. This approach can lower the risk of unauthorized access to sensitive data.

3.5.1.3. Traffic Monitoring and Integrity Control

Employ traffic monitoring tools integrated with the Elastic Stack to capture and analyze real-time network traffic. This enables the detection of attack signs, such as denial of service (DoS) or data exfiltration. Additionally, integrity checks should be implemented to protect network data, ensuring the logs remain unaltered and reliable for security analysis.

3.5.2. Control Assessment

Evaluating the effectiveness of risk control measures helps determine the level of protection the ELK Stack-based Network Intrusion Detection System (NIDS) can provide. This process involves analyzing threat detection capabilities, response effectiveness to security incidents, and the impact of control measures on system performance.

3.5.2.1. Effectiveness

Each control measure is assessed based on its ability to detect, prevent, and mitigate cyberattacks. The system must be capable of identifying common attack patterns such as port scanning, denial-of-service (DoS) attacks, and unauthorized access attempts. Utilizing ELK Stack for log analysis and Suricata for intrusion detection enhances system security by providing real-time alerts and network traffic analysis.

Assess the effectiveness against known and unknown intrusion techniques, ensuring robust protection against threats and zero-day vulnerabilities. Additionally, evaluate the impact of these controls on system performance, ensuring minimal disruptions while maintaining high detection accuracy.

3.5.2.2. Limitations

Although the ELK Stack-based intrusion detection system effectively monitors and detects abnormal activities, limitations exist. Advanced attack techniques, such as evasion tactics and zero-day exploits, may bypass existing security mechanisms. Additionally, real-time log analysis can generate many false positives, increasing the workload for security administrators and slowing response times. Therefore, optimizing detection rules and improving alert filtering mechanisms are necessary to enhance system performance.

Chapter 4: RISK MANAGEMENT PLAN

4.1. Reporting Requirements

4.1.1. List of Threats / Vulnerability

The list of vulnerabilities brought on by threats is shown in the table below so that threat management can be properly planned

No	Threats	Vulnerability
1	Denial of Service (DoS) Attacks	- Resource exhaustion (e.g., memory, CPU, bandwidth). - Service unavailability. - System crash vulnerabilities.
2	Scan	- Exposure of open ports and services. - Leak of configuration details. - Identification of weak points for exploitation.
3	Exploit Bypass	- Weak authentication mechanisms. - Exploitable bugs in software. - Unauthorized access to resources.

Table 4.1. List Of Threats/Vulnerabilities

4.1.2. Costs Associated with Risks

Every risk entails a cost. Equipment, processes, and survey results may be impacted by the damage. It establishes the risks and displays the expenses incurred. One of the things to undertake while conducting the inquiry is the table below.

Risk	Cost associated with Risk	Cost
Denial of Service (DoS)	Service downtime, loss of revenue, reputational damage, and costs for mitigation and recovery	High
Exploit Bypass	Unauthorized data access, system compromise, and potential legal implications.	High
Network Scan	Increased vulnerability exposure, potential information leakage, and additional monitoring costs.	Moderate

Table 4.2. Costs Associated With Risks

4.1.3. Recommendations to Reduce Risks

Following threat management in the plan, we have suggested a way to lessen or eliminate the threat. The list of threats suggested participants for prevention, and associated costs are shown in the table below.

Risk	Recommendations	Cost
Denial of Service (DoS)	Use load balancers, rate-limiting mechanisms, and implement IP filtering to mitigate attack impact.	High
Network Scan	Implement rate-limiting, monitor using IDS (ELK), and block suspicious IPs dynamically.	Moderate
Exploit Bypass	Regularly update and patch systems, deploy vulnerability management tools, and monitor for unauthorized access attempts.	High

Table 4.3. List Of Recommendations to Reduce Risks

4.1.4. Cost-Benefit Analysis (CBA)

Risk	Cost before applying recommendations	Cost after applying recommendations
Denial of Service (DoS)	High	Low
Network Scan	High	Moderate
Exploit Bypass	High	Moderate

Table 4.4. Cost-Benefit Analysis

4.2. Assigning Responsibilities

This method shows the duties of each person in the project. They have to take responsibility for their job and complete the work on time.

Full name	Role	Responsibilities
Nguyễn Quốc Huy	Leader	Project Planning, Offering Solutions, Tracking Progress Management, Assigning Tasks
Dương Thành Lộc	Developer, Document Writer	Developing, Testing, Reporting, Document Writer and Quality Assurance
Huỳnh Trí Đức	Developer, Document Writer	Developing, Testing, Reporting, Document Writer and Quality Assurance
Võ Trọng Đức	Developer, Document Writer	Developing, Testing, Reporting, Document Writer and Quality Assurance
Nguyễn Đăng Khoa	Developer, Document Writer	Developing, Testing, Reporting, Document Writer and Quality Assurance

Table 4.5. Assigning Responsibilities

4.3. Describing Procedures and Schedules for Accomplishment

● Denial of Service (DoS) Attacks

Recommendations

- Deploy Distributed Denial of Service (DDoS) protection
- Implement load balancing mechanisms

Procedures / Schedules

- Set up DDoS protection systems to filter and mitigate attack traffic.
- Implement load balancing across multiple servers to distribute traffic and prevent overload.
- Regularly test and update DDoS mitigation systems for optimal performance.

● Scan Attacks

Recommendations

- Utilize network monitoring tools to identify scanning activities.
- Set up firewalls (Iptables) and Intrusion Detection Systems (IDS) to block suspicious scanning attempts.

Procedures / Schedules

- Periodically review logs from monitoring tools to identify patterns of scanning behavior and adjust settings accordingly.
- Schedule vulnerability assessments to ensure the effectiveness of mitigation measures.
- Regularly update firewall (Iptables) rules to restrict unauthorized access.
- Configure IDS/IPS to detect and alert scanning attempts in real-time

● Exploit Bypass Attacks

Recommendations

- Conduct regular security audits to identify exploitable vulnerabilities.
- Apply patches and updates promptly to prevent known exploits.
- Use threat intelligence to stay informed about the latest bypass techniques.

Procedures / Schedules

- Implement automated patch management tools to reduce time delays in applying updates.

- Perform penetration testing to simulate bypass attempts and verify system defenses.
- Integrate a rule-based monitoring system in ELK to detect abnormal behavior of exploit bypass attempts.
- Provide team training to recognize and respond effectively to exploit bypass scenarios.

4.4. Reporting Requirements

4.4.1. Present Recommendations

The following recommendations were made to increase security and reduce risks associated with process injection attacks for this project:

- **Implement Rate Limiting and Traffic Monitoring:** Develop and apply Rate-limiting mechanisms to limit excessive traffic to critical services and prevent DoS attacks. Implement real-time traffic monitoring to detect unusual traffic patterns that indicate an ongoing attack.

- **Enhance Network Segmentation (for Scan):** Limit the exposure of sensitive systems by isolating them into protected zones. Use firewalls to restrict access to ports and services.

- **Apply Advanced Endpoint Protection (for Exploit Bypass):** Develop security solutions that can detect possible zero-day exploits. Keep endpoint security software up-to-date to ensure protection against the latest threats and vulnerabilities.

4.4.2. Document Management Response to Recommendations

Recommendation	Action By	Management Response
Implement Rate Limiting and Traffic Monitoring	Technical Team	Accepted
Enhance Network Segmentation (for Scan)	Technical Team	Accepted

Apply Advanced Endpoint Protection (for Exploit Bypass)	Technical Team	Accepted

Table 4.6. Document Management Responses

4.4.3. Document and Track Implementation

No.	Tasks	Accepted	Rejected
1	Implement Rate Limiting and Traffic Monitoring	X	
2	Enhance Network Segmentation (for Scan)	X	
3	Apply Advanced Endpoint Protection (for Exploit Bypass)	X	

Table 4.7. Document & Track Implementation

4.5. Plan of Action and Milestones

A comprehensive plan of action with clear milestones is crucial. The following outlines the plan of action and key milestones:

Action	Responsible	Start	Final
Brainstorming		07/01/2025	10/01/2025
First group meeting and project initializing	Group	07/01/2025	08/01/2025
Identify goals, planning and delegating tasks	Nguyễn Quốc Huy	08/01/2025	10/01/2025
Researching		10/01/2025	28/02/2025
Reading resources and documents about ELK	Group	10/01/2025	12/01/2025
Discussing risk assessment and management	Group	12/01/2025	14/01/2025
Research about IDS	Huynh Tri Duc	14/01/2025	17/01/2025

Research about pfSense	Group	17/01/2025	19/01/2025
Research about Web server	Group	19/01/2025	21/01/2025
Analyzing how ELK works	Group	05/02/2025	10/02/2025
Analyzing the process of capturing logs	Dương Thành Lộc	10/02/2025	20/02/2025
Analyzing how to block unusual IPs	Huỳnh Trí Đức	10/02/2025	20/02/2025
Analyzing reporting of abnormalities to gmail, telegram	Võ Trọng Đức	10/02/2025	20/02/2025
Analyzing Dashboard	Huỳnh Trí Đức	10/02/2025	20/02/2025
Analyzing ELK	Dương Thành Lộc	10/02/2025	20/02/2025
Analyzing putting the system on the internet	Group	20/02/2025	24/02/2025
Reporting and proposing solutions	Group	24/02/2025	26/02/2025
Reviewing works and documents	Group	26/02/2025	28/02/2025
Deploying Solutions		28/02/2025	20/04/2025
Building development environment	Group	28/02/2025	04/03/2025
Building solutions	Group	04/03/2025	20/03/2025
Developing method detecting	Group	21/03/2025	11/04/2025
Fixing and testing solutions	Group	11/04/2025	20/04/2025
Evaluating		20/04/2025	25/04/2025
Reviewing all solutions	Group	20/04/2025	22/04/2025
Collecting all documents and preparing for presentation	Group	22/04/2025	23/04/2025
Presenting	Group	23/04/2025	25/04/2025

Table 4.8. Plan of Action & Milestones

4.6. Charting the Progress of a RMP

Tracking and visualizing the progress of a Risk Management Plan (RMP) is essential to ensure effective management and timely completion of tasks. The following charts can be used to chart the progress of the RMP:

4.6.1. Board View

Board view is an effective visual tool for displaying tasks and their statuses in the Risk Management Plan. It organizes tasks into columns (often representing phases or statuses such as “Not Started,” “In Progress,” or “Completed”), providing an overview of the project’s progress and allowing the project team to quickly identify any bottlenecks.

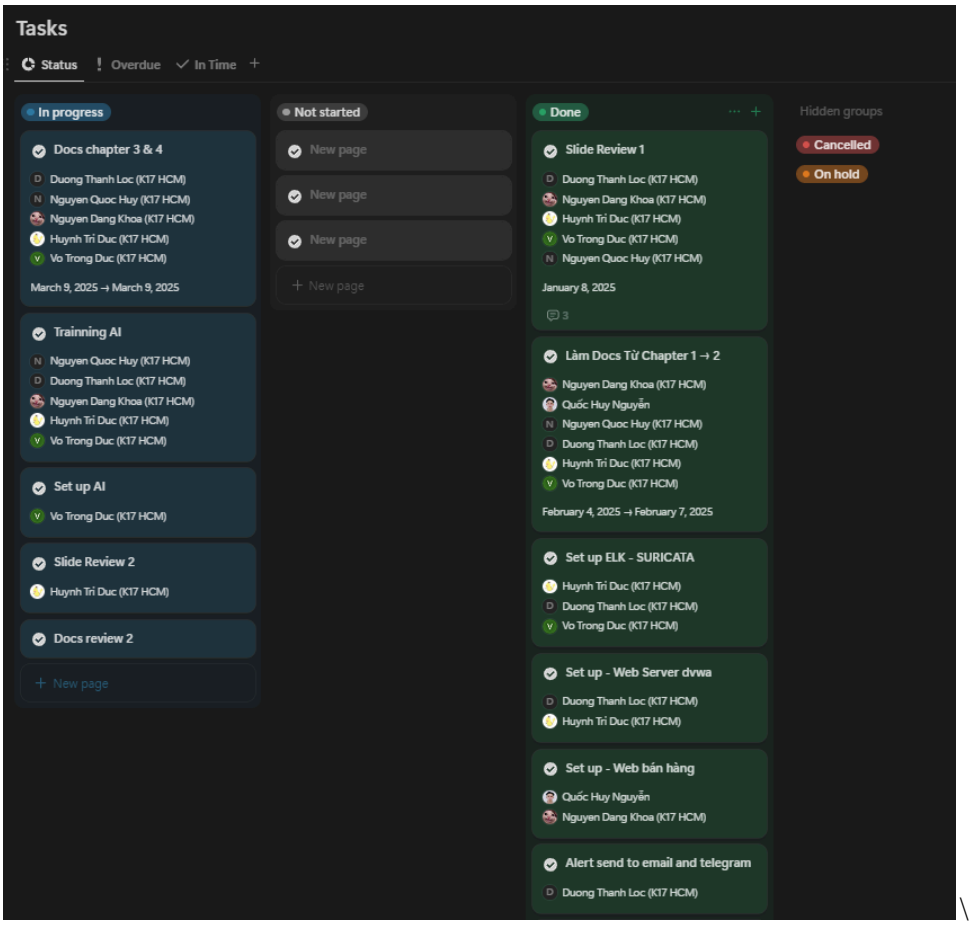


Figure 4.1. Task board view

4.7. Tools and Practices

No	Software	Main Functions	Details
1	Elastic Stack (ELK)	Centralized log management and analysis	Collects, analyzes, and monitors logs from systems and applications, providing real-time alerts and reporting using Elasticsearch, Logstash, and Kibana.
2	Suricata	Intrusion Detection System (IDS)	Detects network attacks and anomalies by analyzing network traffic based on predefined rules.
3	Ubuntu	Open-source operating system based on Linux	Provides the platform for deploying the intrusion detection system (including Elastic Stack and IDS).
4	pfSense	Firewall and network security	Provides advanced firewall capabilities, traffic shaping, and VPN support to enhance network security and monitor traffic.
5	Syslog	Log collection and management	Facilitates the transmission of source logs to a remote destination using predefined filters, also allows customization and can facilitate almost any logging need.
5	PNETLab	Virtual network simulation	Runs the lab environment locally, allowing for virtualized network simulations and IDS testing.
6	Python3	Programming language	Preferred language in AI and machine learning, aid in developing predictive models and AI applications
7	Web Server	Web server platform	Hosts web applications and serves as a logging source for ELK, where web traffic logs are analyzed for anomalies.
8	Telegram	Receive instant notifications via Telegram messages	The Telegram Bot API integrates with ELK, allowing rapid notifications to users when an alert is triggered.

Table 4.9. Tools & Practices

Chapter 5: DEVELOPMENT AND IMPLEMENT PLAN

5.1. Research on ELK Stack for Network Security

The ELK Stack (Elasticsearch, Logstash, Kibana) is a powerful toolset for real-time log monitoring, analysis, and visualization. It enables organizations to detect, analyze, and respond to security threats efficiently by centralizing and processing security logs from various network sources.

Core Functionality:

- Log Aggregation & Processing: Collects and structures logs from firewalls, IDS (Intrusion Detection System), web servers, and endpoint devices for efficient analysis.
- Real-time Monitoring & Alerts: Uses Logstash pipelines and Kibana dashboards to visualize security data and detect anomalies.
- Automated Threat Detection: Integrates with Suricata IDS to identify suspicious network activities and trigger security alerts.

5.1.1. Log Collection and Threat Detection Process

To enhance network security, the ELK Stack is used to establish a robust system for collecting and analyzing logs. The process includes:

- Log Collection: Deploys log forwarders (e.g., Filebeat, Syslog) to gather security logs from network devices and servers.
- Data Processing & Filtering: Using Logstash to parse and enrich log data for more accurate analysis.
- Threat Detection & Response: Configure Kibana dashboards and alerts to detect unusual activities, enabling security teams to respond promptly to potential threats.

The operating mechanism:

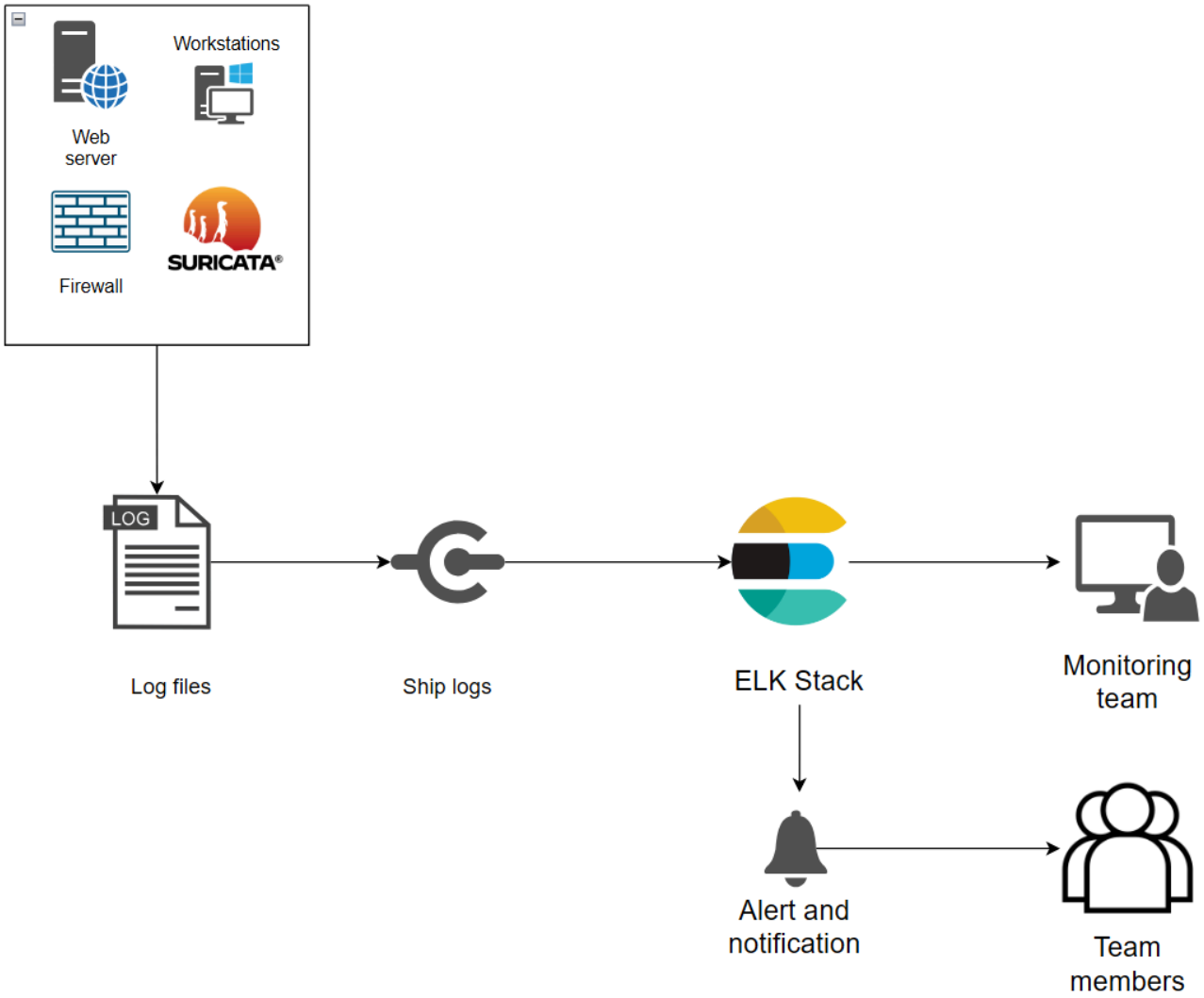


Figure 5.1.1 The process of catching logs

Example:

The ELK Stack comprises Elasticsearch, Logstash, and Kibana, which are widely adopted open-source platforms for centralized log management and analysis. It enables organizations to collect logs from various sources, including routers, firewalls, and servers. It uses protocols like TCP and UDP to ensure comprehensive data collection. Once collected, Logstash processes these logs by parsing and structuring the raw data, extracting key fields such as IP addresses and timestamps, which facilitates efficient analysis. The structured data is then indexed and stored in Elasticsearch, a distributed search and analytics engine optimized for swift retrieval. For visualization and exploration, Kibana provides an intuitive interface that allows users to create dynamic dashboards and perform in-depth analyses. Beyond standard log data, the ELK Stack is good at capturing and analyzing logs related to specific security threats, including Denial of Service (DoS) attacks, Reconnaissance attempts, and SQL Injection attempts, enhancing the ability to detect and respond to potential threats.

5.1.2. Blocking Abnormal IPs

Blocking suspicious IP addresses is crucial for mitigating security threats. By integrating ELK Stack with pfSense Firewall, the system can automatically identify and block malicious IPs based on security logs.

Mechanism:

- Suricata IDS identifies suspicious network activities.
- Logstash processes alerts and forwards relevant data to pfSense.
- Kibana dashboards provide real-time monitoring of threats.
- pfSense Firewall updates blocklists and prevents access from flagged IPs.

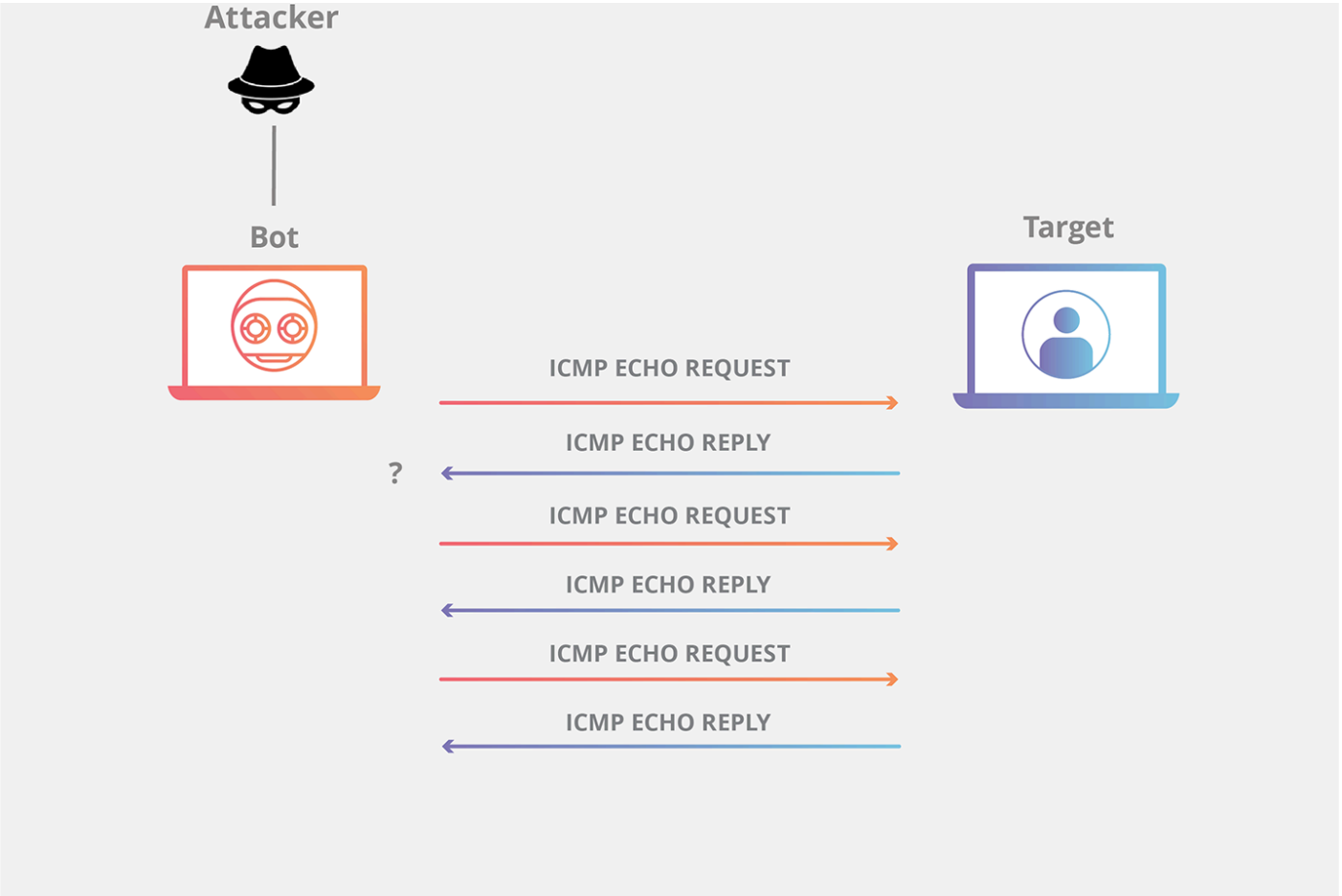


Figure 5.1.2 The process of blocking abnormal IP addresses

Example:

Integrating the ELK Stack with pfSense enables organizations to detect and block anomalous IP addresses effectively. Initially, pfSense is configured to forward its logs to the ELK Stack via syslog protocols. Logstash processes these logs, parsing and structuring the data to extract critical information such as IP addresses, timestamps, and event types. This data will be indexed and stored in Elasticsearch, facilitating efficient analysis. Upon detecting

such anomalies, Kibana can generate automated alerts quickly to notify administrators. To further enhance network security, pfSense can automatically add these identified IP addresses to a blocklist, preventing further access.

5.1.3. Alerting System: Gmail & Telegram Integration

Real-time notifications allow security teams to respond quickly to threats. By configuring ELK Stack, alerts can be sent via Gmail and Telegram whenever abnormal activities are detected.

Implementation:

- **Gmail Alerts:** Logstash sends email notifications through SMTP when critical threats are detected.
- **Telegram Notifications:** Logstash integrates with Telegram API to push alerts to security teams.

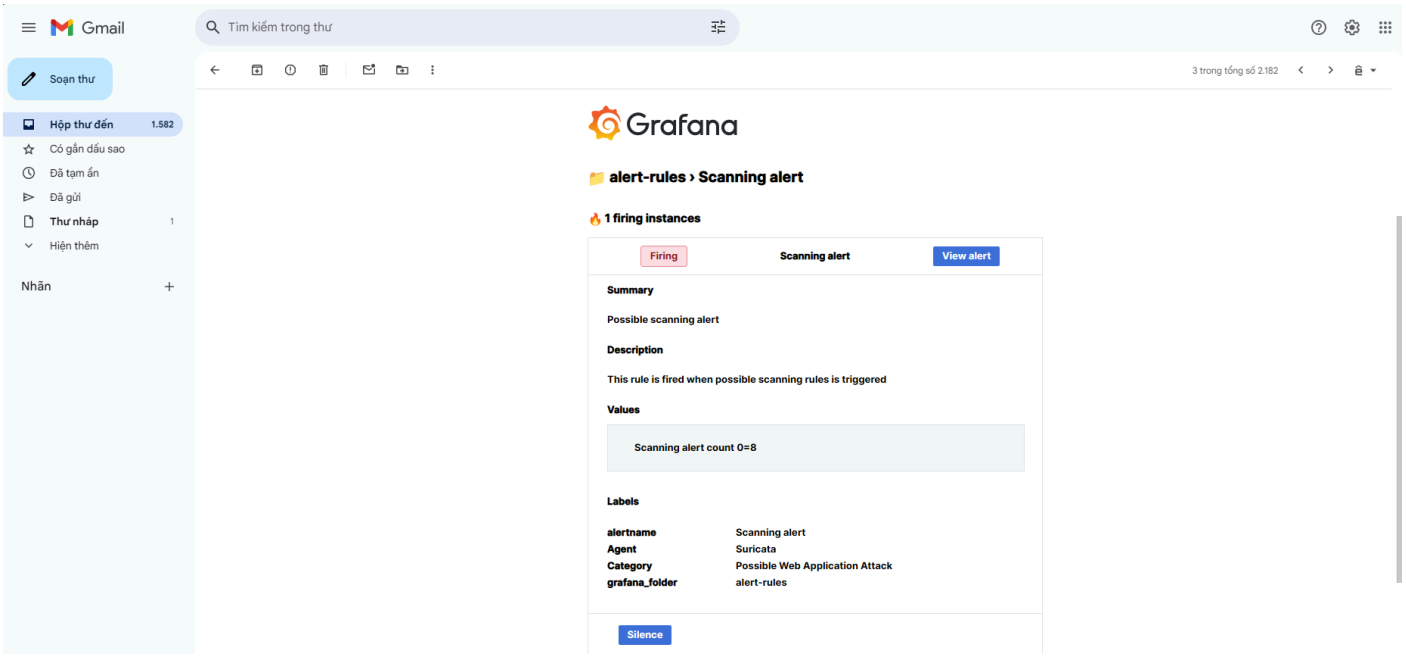


Figure 5.1.3.1 Anomaly detection notification sent to Gmail

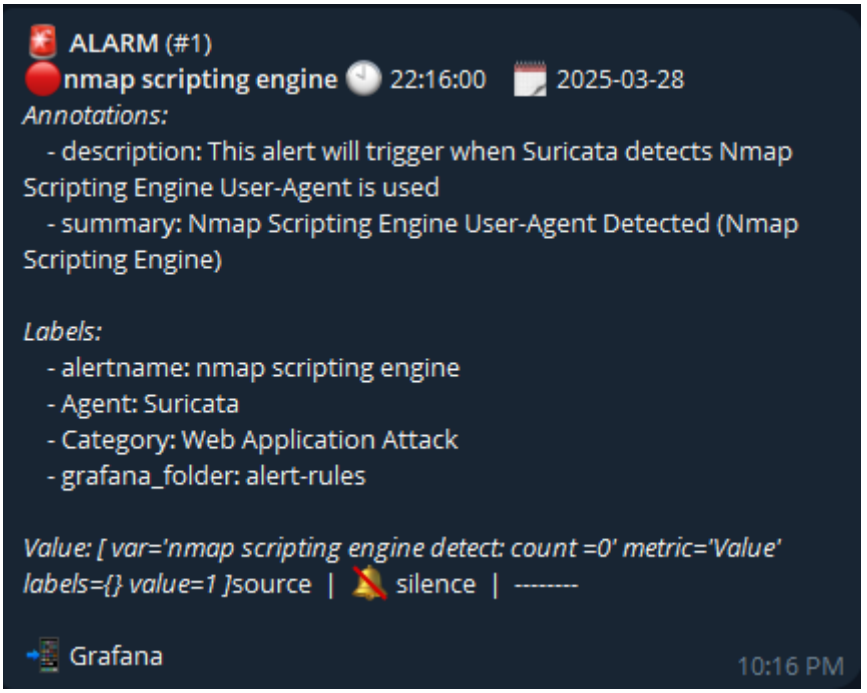


Figure 5.1.3.2 Anomaly detection notification sent to Telegram

Example:

Elastic leverages its alerting system and integrates it with external notification services to notify of abnormalities via Gmail and Telegram. Elastic configures SMTP settings for Gmail alerts, allowing it to automatically send email notifications to specific recipients upon detecting predefined alert conditions. For Telegram, Elastic can use Telegram's API to send alerts directly to an administrator's Telegram account or group. When Elastic generates an alert when an abnormality is identified based on set thresholds or patterns (e.g., repeated failed logins, unusual IP traffic), it will send the alerts through the configured communication channels, ensuring administrators are informed immediately for quick and effective response.

5.1.4. Define how to put the system on the internet

Deploying the hybrid Suricata and machine learning-based Network Intrusion Detection System (NIDS) to a public-facing environment substantially enhances security monitoring capabilities while ensuring comprehensive network protection against sophisticated cyber threats. The implementation begins with strategic placement of the Suricata engine at network boundaries, configured in IDS mode to analyze traffic flows from the gateway router. This setup enables the system to capture and process network traffic through its multi-tiered architecture, which combines signature-based detection with anomaly-based machine learning capabilities. The machine learning component—utilizing Random Forest, and Decision Tree algorithms trained on the CICIDS2017 dataset—significantly improves detection of zero-day attacks and sophisticated threats that evade traditional signature-based systems.

The detection workflow operates as an efficient eight-step process: First, the system establishes a baseline of normal network behavior through machine learning algorithms

trained on benign traffic data. When deployed, a high-speed network sniffer connected to a SPAN port captures traffic without introducing latency. Each captured packet is first analyzed against the signature database (SNIDS component); if a match is found, an immediate alert is sent to administrators. Packets without signature matches are passed to the anomaly detection component (ADNIDS), which compares them against the established behavioral baseline. Any deviation from this baseline triggers an alert identifying the packet as suspicious. The system continuously improves by feeding newly identified suspicious patterns back to the signature database for future detections, while normal-behaving packets are allowed to pass without alerts. This dual-detection approach ensures comprehensive coverage against both known threats and previously unseen attack vectors, making it particularly effective in defending internet-facing deployments where novel attack methods are frequently encountered.

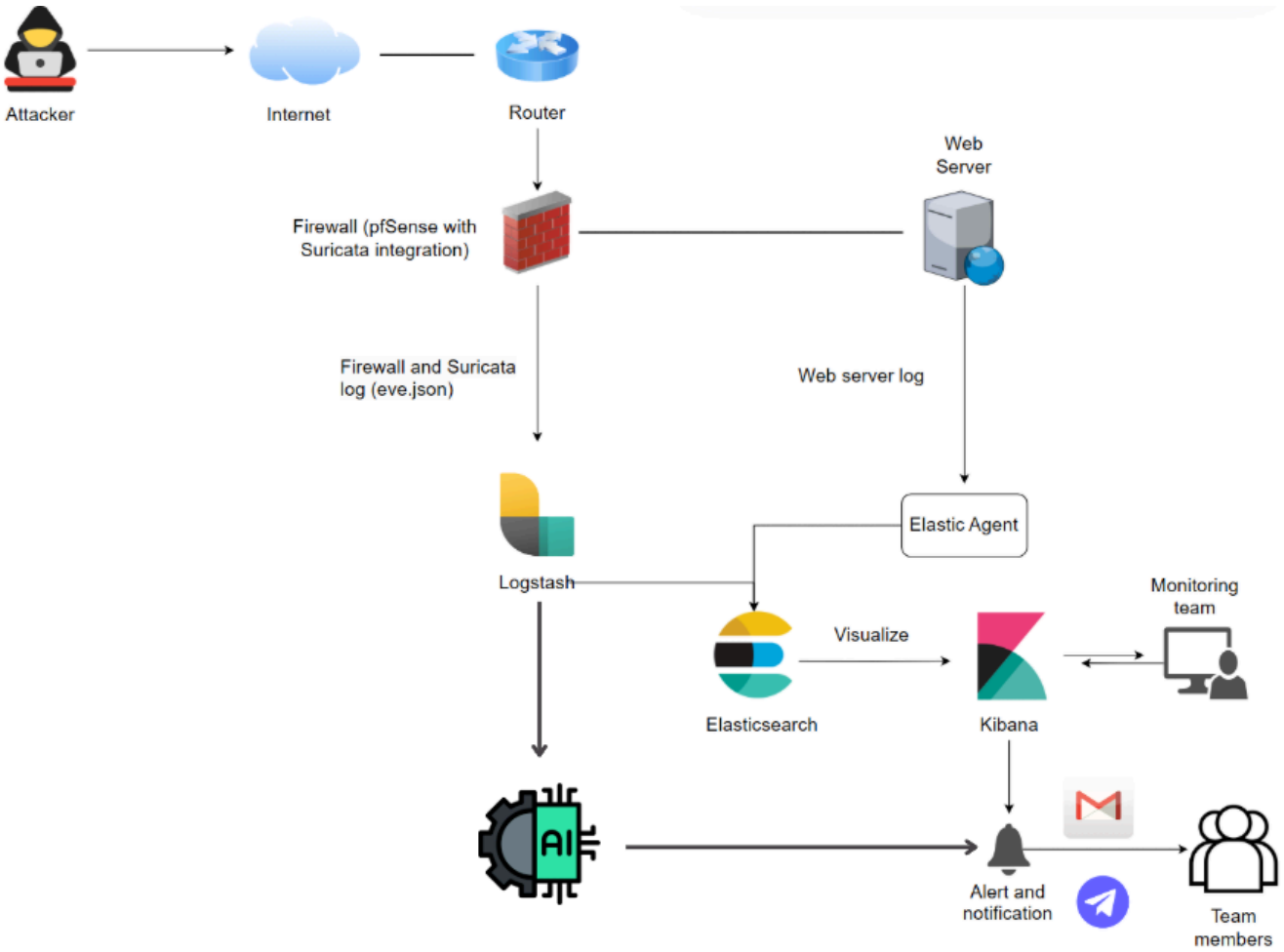


Figure 5.1.4 Design Diagram for Accessing from the Internet

Example:

To make ELK Stack accessible on the internet, begin by configuring the VM's network settings for public access. This usually involves assigning a public IP address to the

VM or configuring port forwarding from a public-facing firewall to the VM's internal IP. Ensure that all required ports are open, primarily HTTP (80) or HTTPS (443).

5.1.5. Define how to export to dashboard

Exporting logs and analytics to a dashboard provides a centralized, real-time overview of network activity, allowing administrators to quickly identify and respond to potential security incidents.

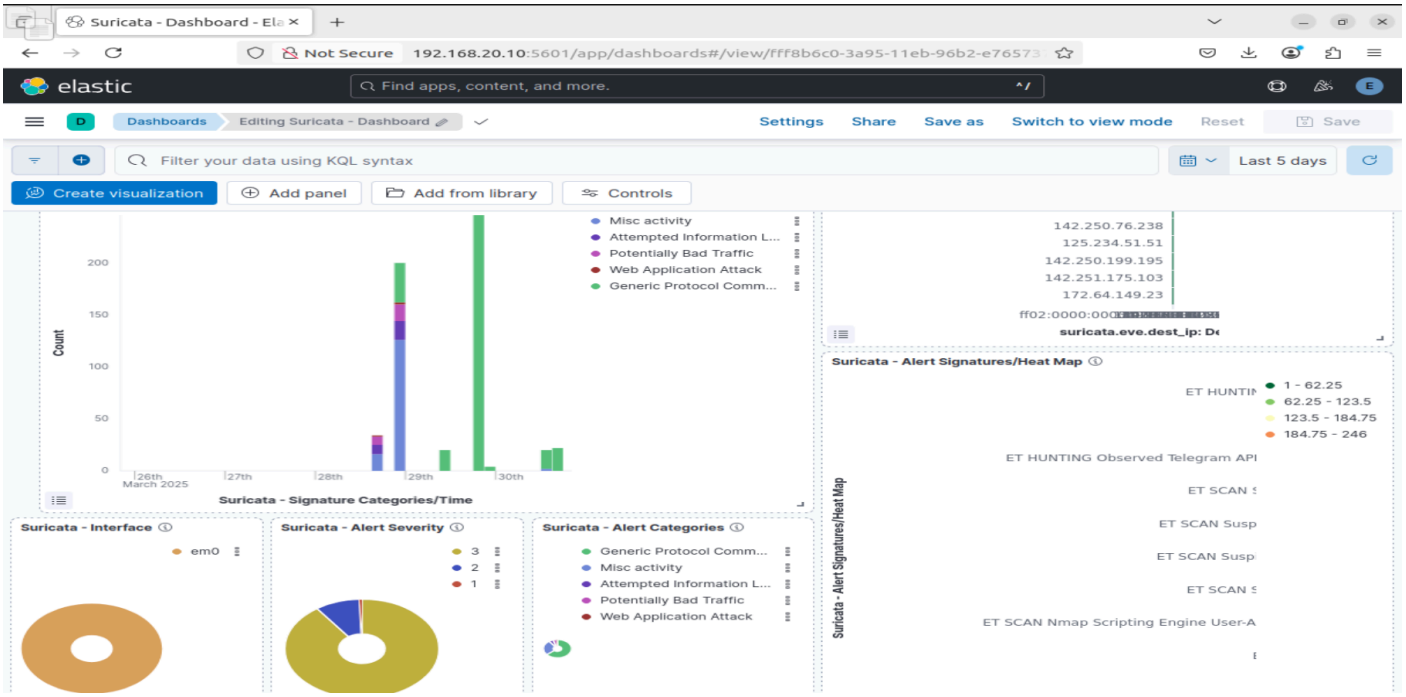


Figure 5.1.5 Export to Kibana dashboard

Example:

To export Elastic Stack data to a dashboard, create visualizations using its built-in tools. Begin by defining search queries or filters to display specific log data. Save those queries and add them as a visualization to the dashboard, combining multiple widgets for a comprehensive view of network activities. Configure each visualization based on data types such as IP traffic, error rates, or alerts for abnormal activities to ensure clarity and actionability. Once set up, the dashboard becomes accessible through the Elastic Stack interface, enabling real-time monitoring and customizable views that can be shared with authorized users.

5.2. Hybrid Machine Learning-Based Anomaly Detection System

5.2.1. Overview

The hybrid network intrusion detection system (NIDS) developed in this project combines the strengths of signature-based detection via Suricata with machine learning-based anomaly detection. While signature-based systems excel at identifying known threats with

high precision, they often fail to detect novel or zero-day attacks. To address this limitation, we implemented a machine learning component that analyzes network flow characteristics to identify deviations from normal behavior patterns.

Our machine learning subsystem functions as the Anomaly Detection-based NIDS (ADNIDS) component within the overall architecture. When network packets do not match known signatures in Suricata, they are forwarded to the ADNIDS for analysis. This dual-detection approach creates a more robust security posture capable of identifying both known and unknown threats.

The following sections detail the specific implementation of our machine learning-based anomaly detection system, including dataset preparation, feature engineering, model selection, training methodology, and integration with the broader ELK-based monitoring infrastructure.

5.2.2. Dataset and Preprocessing

5.2.2.1. Dataset Selection and Analysis

For training our machine learning models, we selected the CICIDS2017 dataset, which contains benign and attack network flows captured in a realistic network environment. The dataset includes various attack types such as DoS, DDoS, brute force, web attacks, and infiltration, making it ideal for training a comprehensive detection system.

The CICIDS2017 dataset presented several challenges:

1. **Class Imbalance**: The dataset contains significantly more benign traffic (2,273,097 instances) than attack traffic (557,646 instances) (see Table 1).
2. **Data Quality Issues**: The dataset contains missing values, infinite values, and duplicates that needed to be addressed
3. **High Dimensionality**: The dataset includes 78 features, many of which are redundant or irrelevant

Class of network traffic	Frequency
Normal	2,273,097
Abnormal	557,646

Table 5.1. The CICIDS2017 dataset is unbalanced

5.2.2.2. Preprocessing Pipeline

We implemented a comprehensive preprocessing pipeline to address these challenges:

1. **Duplicate Removal**: We identified and removed 308,381 duplicate records to prevent bias in the model training process.

2. **Handling Missing and Infinite Values**: We observed that 0.02% of values (5,734 instances) in columns 'Flow Bytes/s' and 'Flow Packets/s' were either missing or infinite. These were replaced with median values to maintain the statistical integrity of the dataset.

3. **Dataset Balancing**: To address class imbalance, we created a balanced dataset with 425,878 benign samples and a proportional number of attack samples. This balanced dataset prevents the model from being biased toward the majority class (see Table 2).

4. **Feature Selection**: We implemented a three-step feature selection process:

- Removed constant or quasi-constant features that provide little discriminative value
- Eliminated duplicated features using the Python duplicated() function
- Removed highly correlated features using correlation analysis

Class of network traffic	Frequency
Normal	425,878
Abnormal	425,878

Table 5.2. The balanced CICIDS2017 dataset

The preprocessing pipeline results in a clean, balanced dataset suitable for training machine learning models while maintaining the essential characteristics needed for effective intrusion detection.

5.2.3. Feature Engineering

5.2.3.1. Feature Alignment with Suricata

A critical aspect of our hybrid system is ensuring compatibility between features extracted from Suricata logs and those used in the machine learning model. We developed a specialized feature mapping approach that aligns Suricata's flow data fields with the CICIDS2017 dataset features used for training (see Table 3).

CICIDS2017 Feature	Suricata Field	Description
Destination Port	dest_port	Target port of the connection
Flow Duration	duration	Time from first to last packet
Total Fwd Packets	total_fwd_packets	Number of packets sent from source to destination
Total Backward Packets	total_bwd_packets	Number of packets sent from destination to source
Total Length of Fwd Packets	total_fwd_bytes	Total bytes sent from source to destination

Total Length of Bwd Packets	total_bwd_bytes	Total bytes sent from destination to source
Flow Bytes/s	flow_bytes_per_sec	Rate of bytes per second (derived)
Flow Packets/s	flow_packets_per_sec	Rate of packets per second (derived)
Down/Up Ratio	down_up_ratio	Ratio of download to upload traffic (derived)

Table 5.3. The selected features of the CICIDS2017 dataset

5.2.4. Model Training and Evaluation

5.2.4.1. Model Selection

We evaluated several machine learning algorithms for their effectiveness in detecting network intrusions. Our analysis included:

- **Decision Tree Classifier**: A simple, interpretable model that can capture complex decision boundaries
- **Random Forest Classifier**: An ensemble method that improves upon decision trees by reducing overfitting
- **XGBoost Classifier**: A gradient boosting algorithm known for its performance in structured data tasks

Based on preliminary cross-validation tests, we conducted more detailed evaluations of these algorithms.

5.2.4.2. Training Methodology

For each algorithm, we:

- Split the preprocessed data into training (75%) and testing (25%) sets
- Applied Standard Scaling to normalize feature values
- Trained the model using cross-validation to prevent overfitting
- Performed hyperparameter tuning using grid search for optimal performance

5.2.5. Performance Metrics and Evaluation

5.2.5.1. Confusion matrix

The confusion matrices demonstrate that all three models—Decision Tree, Random Forest, and XGBoost—achieve high classification accuracy. The Decision Tree model performs well but shows a slightly higher tendency to misclassify attacks (see Fig 1). Random Forest offers the best balance between precision and recall, making it the most reliable option for practical deployment (see Fig 2). XGBoost exhibits outstanding

precision with the fewest false positives, which is advantageous in reducing false alarms, though it marginally sacrifices recall (see Fig 3). Overall, Random Forest provides the most consistent and balanced performance across both benign and malicious classifications.

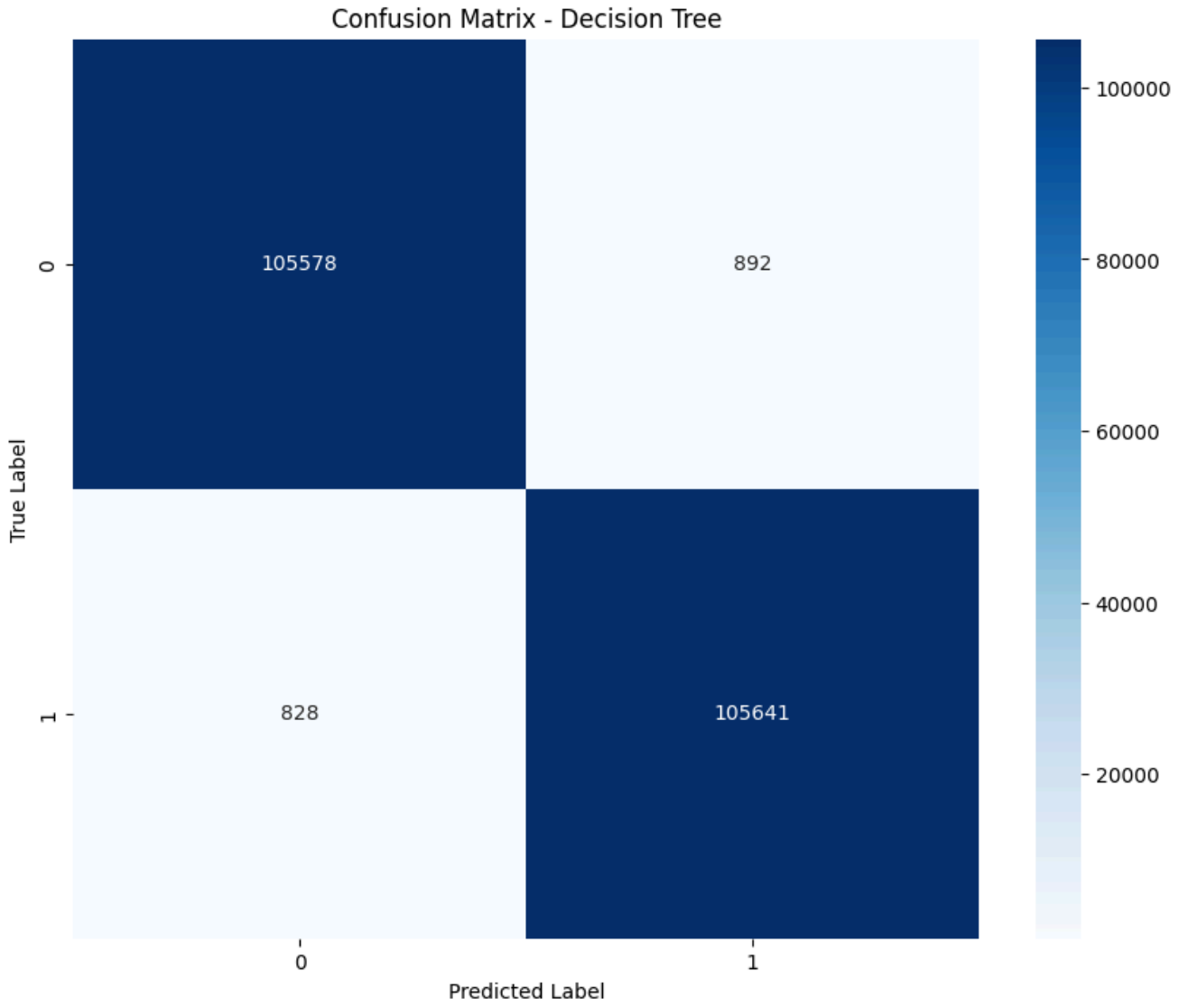


Figure 5.2.5.1 Decision Tree - Confusion Matrix

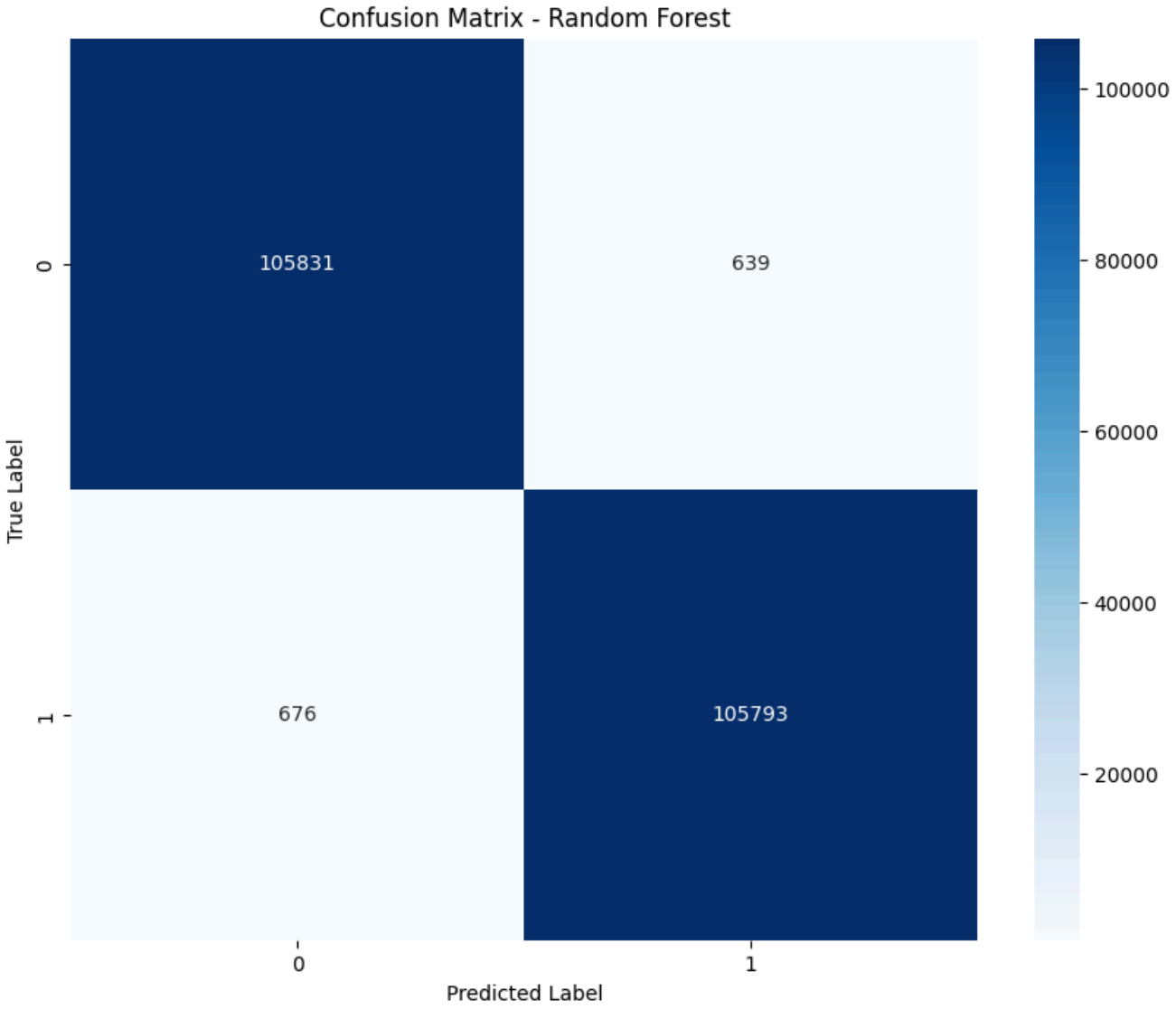


Figure 5.2.5.2 Random Forest - Confusion Matrix

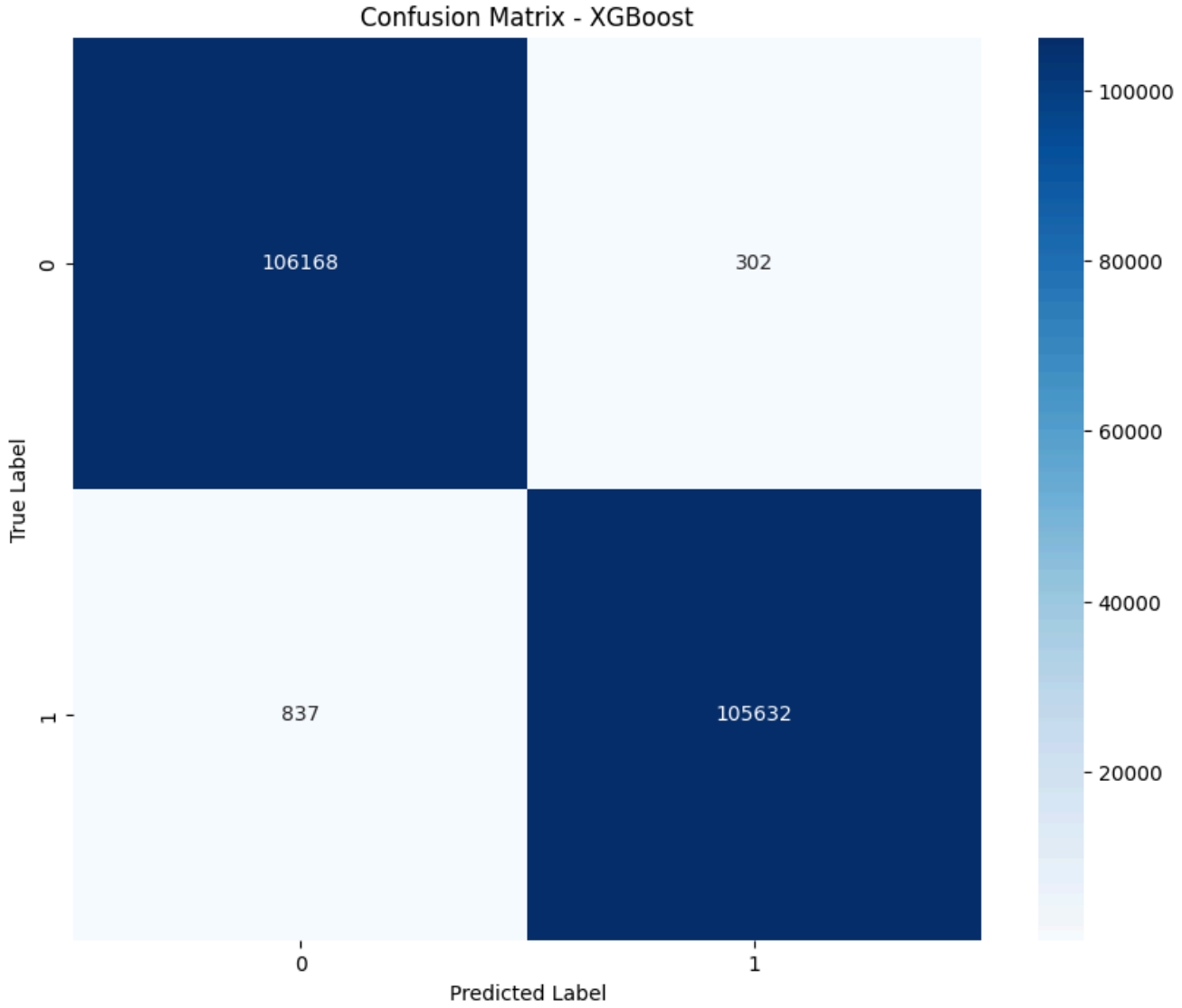


Figure 5.2.5.3. XGBoost - Confusion Matrix

5.2.5.2. Evaluation

The comparison indicates that all three models deliver strong performance on the classification task. Among them, XGBoost demonstrates the most favorable balance between predictive accuracy and training efficiency. Random Forest also achieves high performance, while the Decision Tree model, despite being slightly less effective, remains a viable baseline.

Metric	Decision Tree	Random Forest	XGBoost
Accuracy	99.19%	99.38%	99.46%
F1-Score (Macro)	0.9919	0.9938	0.9946
Recall	0.9919	0.9938	0.9946
Precision	0.9919	0.9938	0.9946
Training Time (seconds)	4.37	133.45	1.60

Table 5.4: Comparison between three models

5.2.6. Real-Time Detection Pipeline

5.2.6.1. Integration with Suricata and ELK Stack

The machine learning models are integrated into the broader detection system through a multi-stage pipeline:

- Log Collection: Suricata captures network traffic and generates structured logs, then sends it to SIEM for Machine Learning to monitor.
- Flow Aggregation: The SessionManager class groups related packets into flow sessions
- Feature Extraction: The AdaptiveFlowFeatureExtractor extracts relevant features from completed flow sessions
- Anomaly Detection: The AnomalyDetector class applies the trained models to detect abnormal behavior
- Alert Generation: Detected anomalies trigger alerts via the ELK Stack's alerting mechanisms

5.2.6.2. Detection Logic

The anomaly detection process follows this workflow:

1. Network traffic is captured and processed by Suricata
2. If Suricata identifies a known signature match, an alert is generated immediately
3. If no signature match is found, the packet is forwarded to the ADNIDS component.
4. The ADNIDS extracts features from the packet and associated flow.
5. These features are preprocessed (scaled, normalized) to match the training data format.
6. The machine learning models analyze the features and generate prediction probabilities.
7. A combined anomaly score is calculated using both machine learning and statistical

methods.

8. If the anomaly score exceeds the threshold, an alert is generated.

5.2.7. Alert Format and Severity Logic

5.2.7.1. Alert Structure

The machine learning component generates detailed alerts that include:

1. **Basic Connection Information**:
   - Source IP and port
   - Destination IP and port
   - Protocol and application protocol
   - Timestamp and duration
2. **Detection Details**:
   - ML algorithm predictions (Decision Tree, Random Forest, XGBoost)
   - Prediction confidence scores
   - Statistical anomaly indicators
   - Combined anomaly score
3. **Feature Analysis**:
   - Top anomalous features
   - Z-scores showing deviation from normal baseline
   - Statistical thresholds that were exceeded
4. **Behavioral Indicators** (if applicable):
   - Port scan/host scan scores
   - Brute force attempt indicators
   - Traffic volume metrics

5.2.7.2. Severity Calculation

Alert severity is determined using a multi-factor approach:

1. **ML Model Confidence**: Higher confidence scores from the models increase severity
2. **Statistical Anomaly Magnitude**: Larger deviations from the baseline increase severity
3. **Behavioral Factors**: Detected scanning, brute force, or volume anomalies increase severity
4. **Application Context**: Anomalies in sensitive protocols (e.g., HTTPS, SSH) receive higher severity

The final severity score is a weighted combination of these factors, categorized as:

- Low (0.3-0.5): Unusual but not necessarily malicious
- Medium (0.5-0.7): Suspicious activity that warrants investigation
- High (0.7-1.0): Likely attack that requires immediate attention

5.2.7.3. Integration with Notification Systems

As described in the previous sections, alerts are sent to administrators via: Telegram messages for immediate mobile notifications

5.3. Technologies

5.3.1. Elastic Stack (ELK)

Elastic Stack, commonly known as the ELK Stack, is a powerful open-source log management and analysis solution widely used in IT environments to monitor, search, and analyze logs from various sources, including applications, network devices, and servers. It enables system administrators and security professionals to efficiently collect, analyze, and set alerts on logs, facilitating early detection of potential issues and enhancing system performance.

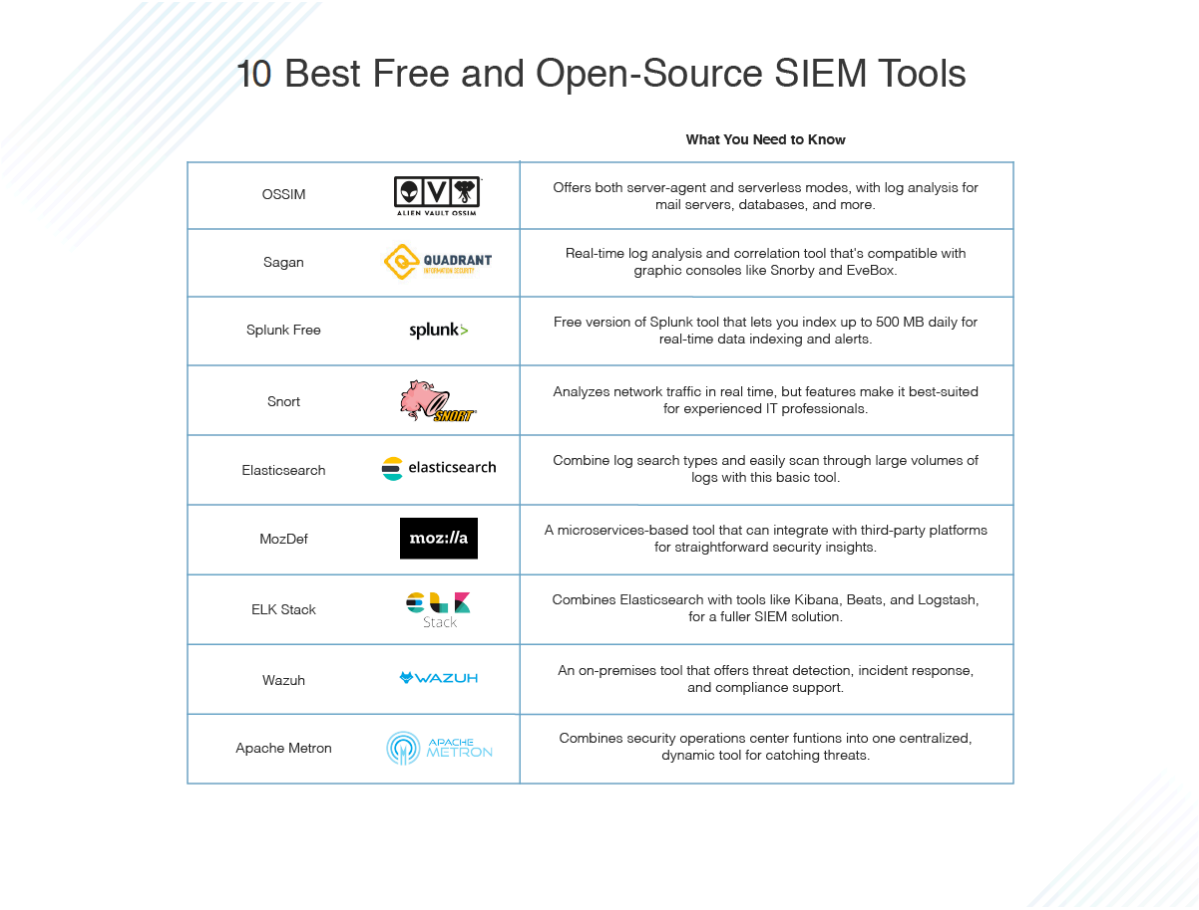


Figure 5.3.1 10 Best Free and Open-Source SIEM Tools

(Source: https://www.dnsstuff.com/free-siem-tools/)

The ELK Stack comprises three primary components:

- Elasticsearch: A distributed search and analytics engine designed for horizontal scalability and real-time data retrieval

- Logstash: A server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a designated storage.

- Kibana: A visualization layer that works on top of Elasticsearch, providing users with the ability to create real-time dashboards and visualizations for data analysis.

Since its introduction, the ELK Stack has become a popular log management tool with a large user community worldwide. Its high customizability, user-friendly interface, and seamless integration with various systems make it a preferred choice for many organizations.

Advantages of the ELK Stack:

- The ELK Stack aggregates logs from diverse sources into a single, searchable repository, simplifying log management and accelerating issue detection and resolution.

- Kibana enables users to create and share real-time visualizations of their data, supporting immediate monitoring and informed decision-making.

- Elasticsearch is designed for scalability. It distributes data across a cluster of servers to manage large data volumes and high query loads.

Disadvantages of the ELK Stack:

- Deploying and maintaining the ELK Stack requires configuring multiple components to function together. Tasks include setting up data ingestion pipelines, defining index policies, and ensuring system security.

- While the ELK Stack software is free, building, scaling, and maintaining the infrastructure requires significant resources. Computing and data storage costs depend on daily log volume and data retention policies.

5.3.2. Intrusion Detection System (Suricata)

Suricata is a well-known Intrusion Detection System (IDS) that monitors network traffic in real-time to detect potential threats, including malware, denial of service attacks, and port scans. It is an open-source tool that identifies suspicious patterns by comparing network traffic to a set of predefined rules and can alert administrators when it detects unusual activity.

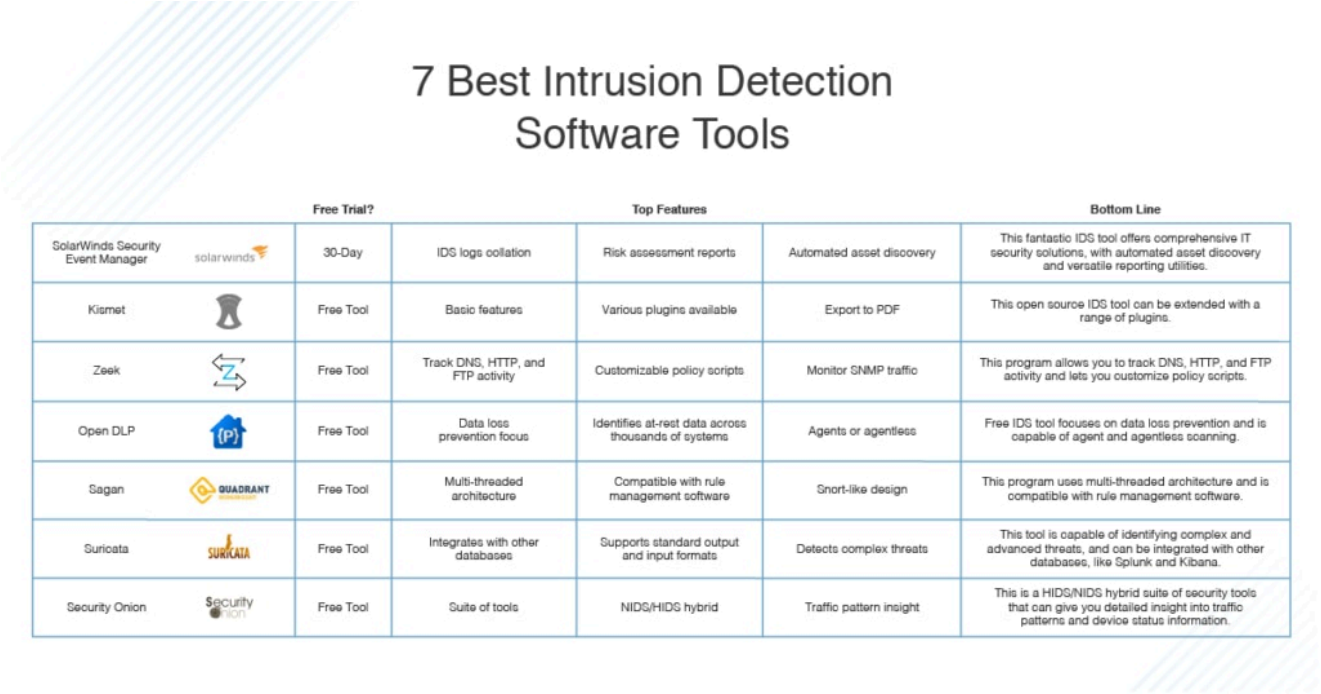


Figure 5.3.2 7 Best Intrusion Detection Softwares

(Source: https://www.dnsstuff.com/network-intrusion-detection-software/)

This intrusion detection system (IDS) was developed by Martin Roesch in 1998 and was initially released as an open-source project. This intrusion detection system (IDS) is a widely deployed IDS solution. Its popularity is due to its effectiveness in identifying a wide range of network threats and its community support, which keeps its rule sets up to date.

Advantages of Suricata:

- Suricata is free to use and highly customizable, making it a good choice for organizations.
- Suricata analyzes network traffic in real-time, allowing for rapid detection and response to potential threats.
- Suricata is multi-threaded, meaning it can use multiple CPU cores simultaneously. This allows it to handle tasks and analyze traffic in real-time.

Disadvantages of the ELK Stack:

- While Suricata offers a user-friendly interface, the proper configuration requires an understanding of network security concepts and IDS/IPS functionalities.
- Suricata can consume significant CPU and memory resources, especially with high-bandwidth networks. Upgrading hardware or implementing distributed deployments might be necessary to ensure optimal performance.

5.3.3. Operating System (Ubuntu)

Ubuntu is a popular, open-source Linux distribution. It is based on Debian and is designed for desktops, servers, and IoT devices. Developed by Canonical Ltd., a UK-based company founded by Mark Shuttleworth, Ubuntu aims to provide an easy-to-use, stable, and secure operating system for users worldwide.

Figure 5.3.3 Types of OS

(Source: https://www.candtsolution.com/news_events-detail/what-is-an-operating-system/)

Key Features of Ubuntu:

- Ubuntu offers an intuitive graphical user interface, making it accessible for users transitioning from other operating systems.
- New Ubuntu versions are released every six months, with Long-Term Support (LTS) versions available every two years, offering five years of support and updates.
- Ubuntu provides access to a vast collection of free and open-source software through its package management system, allowing users to easily install and update applications.

5.3.4. pfSense (Firewall)

The concept of firewalls emerged in the late 1980s as networks became more connected and needed better security measures. Firewalls are one of the most deployed network security tools across all types of organizations, ranging from small businesses to large enterprises. They are a standard component of most network security setups. Firewalls come in many types (hardware-based, software-based, cloud-based) and are available from many popular vendors, including Cisco, Fortinet, Palo Alto Networks, and pfSense.

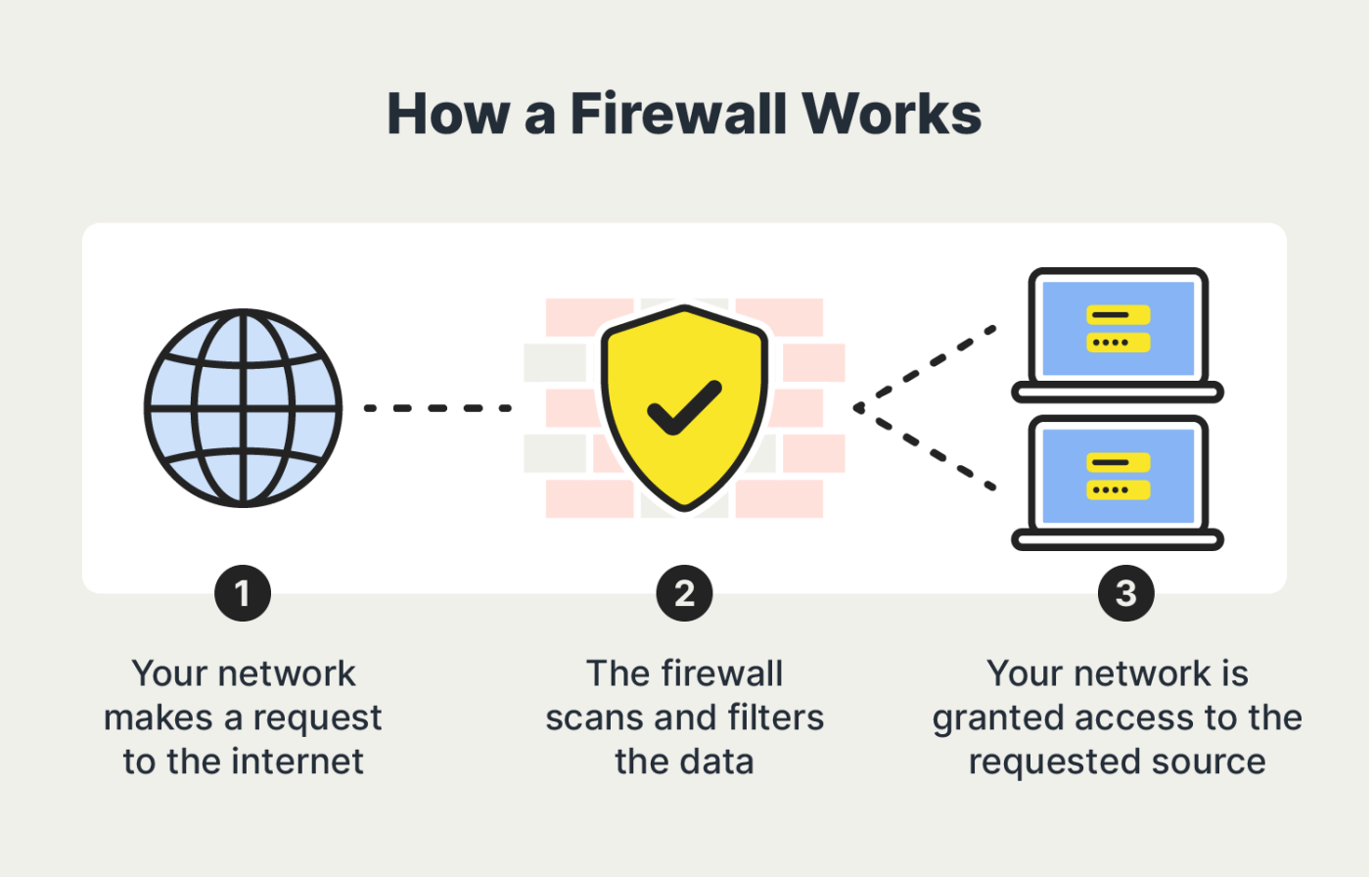


Figure 5.3.4 How a Firewall works

(Source: https://ms.codes/blogs/internet-security/firewall-in-computer-is-used-for/)

pfSense is an open-source firewall and router platform based on FreeBSD designed to provide robust network security solutions for organizations of all sizes. This can monitor and manage incoming and outgoing traffic using predefined or custom rules, acting as a gate between internal networks and external networks, such as the Internet. This protects data and resources within private networks by preventing unauthorized access, detecting potential threats, and allowing safe traffic.

pfSense Advantages:

- pfSense is free to use and customize, offering a cost-effective solution without licensing fees. Its open-source nature allows for transparency and community-driven enhancements.

- In addition to standard firewall capabilities, pfSense supports VPN integration, load balancing, and intrusion detection/prevention systems (IDS/IPS).

- pfSense provides an intuitive web-based interface, simplifying configuration and management tasks. This accessibility makes it suitable for both novices and experienced users.

pfSense Disadvantages:

- While the basic setup is straightforward, configuring advanced features can be complex, especially for users unfamiliar with network security concepts. This may necessitate additional learning or professional assistance.

- Implementing features like deep packet inspection or intrusion detection can be resource-intensive. High-throughput networks may require robust hardware to maintain optimal performance.

5.3.5. Syslog-ng

Syslog-ng is an open-source implementation of the syslog protocol designed to collect, process, and forward log messages from various sources to designated destinations. Developed by Balázs Scheidler in 1998, syslog-ng enhances traditional syslog functionalities by offering content-based filtering, flexible configuration options that support a wide range of protocols, and support for reliable transport protocols like TCP and TLS encryption.

Figure 5.3.5 Syslog-ng - a log management software

(Source: https://www.syslog-ng.com/products/log-management-software/)

Advantages of syslog-ng:

- Syslog-ng enhances SIEM systems' performance and efficiency by reducing the log volume and improving the quality of logs fed into them.

- With solutions like syslog-ng Store Box, users can perform rapid searches across billions of logs using full-text queries with Boolean operators. This facilitates quick identification of critical logs and expedites troubleshooting processes.

- Syslog-ng Store Box provides secure storage and customizable reporting features.

Disadvantages of syslog-ng:

- While syslog-ng offers customization options, configuring it to suit specific organizational needs can be complex and requires a deep understanding of log management principles and the application's configuration syntax.

- Implementing advanced features such as real-time log analysis, filtering, and secure data transmission can use many computer resources.

5.3.6. PNETlab

PNETLab is a free, open-source platform used to create, share, and practice networking labs across multiple vendors. It provides a comprehensive environment for network simulation, enabling users to design and test network topologies without buying and using physical hardware.

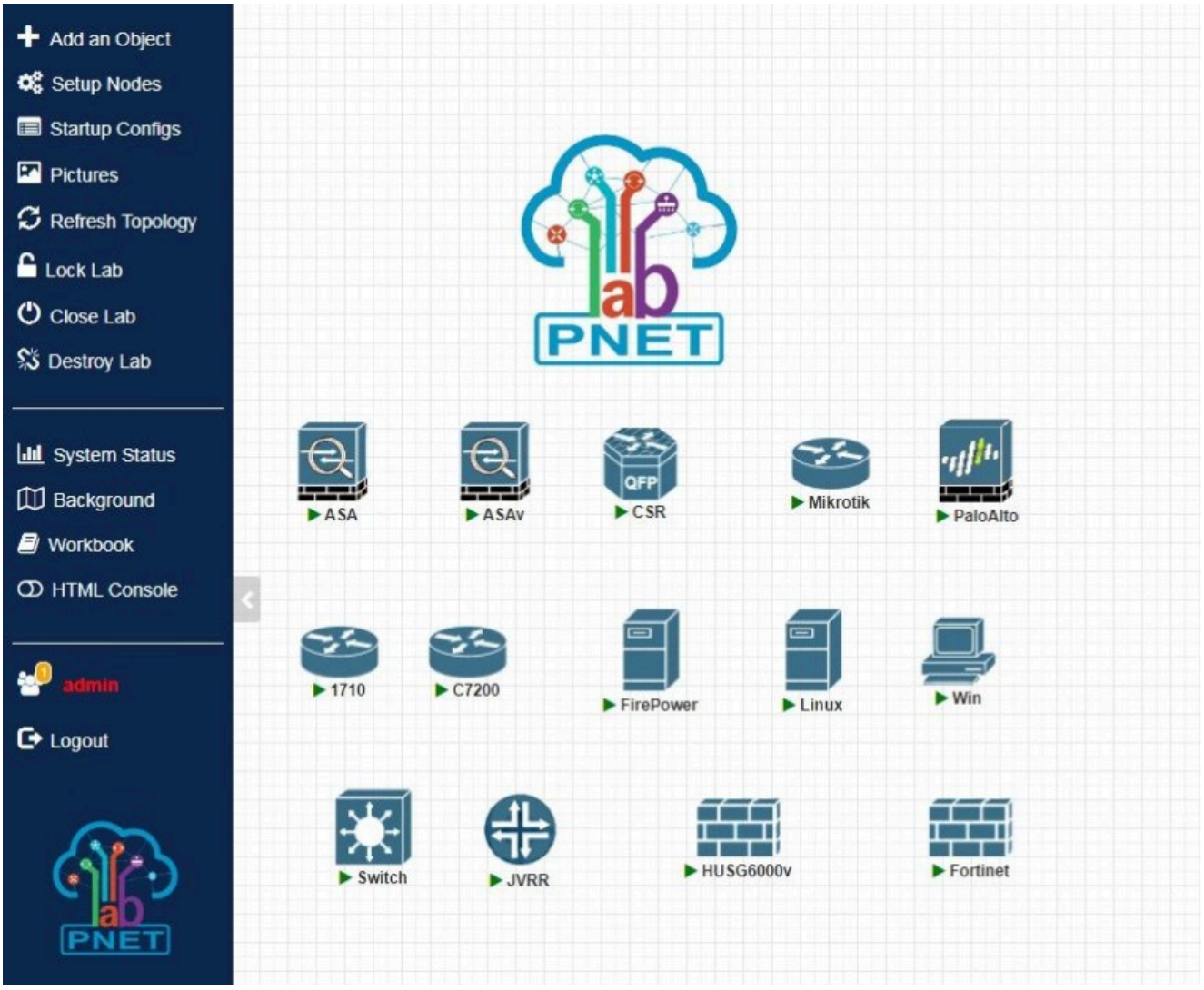


Figure 5.3.6 PNETLab: A full & free Networking Emulator platform

(Source: https://pnetlab.com/pages/main)

PNETLab has a community of users who contribute to its development and share labs through the platform's store. The official website provides documentation, tutorials, and a donation option to support ongoing development. Additionally, PNETLab maintains a presence on social media platforms like Facebook, which users can follow to receive updates and engage with the community.

5.3.7. Python3

Python 3 is a modern, high-level programming language renowned for its readability, versatility, and community support. Building upon the foundation of Python 2, Python 3 introduced significant enhancements that have proved its position as a leading language across various domains, including web development, data science, artificial intelligence, automation, and education.

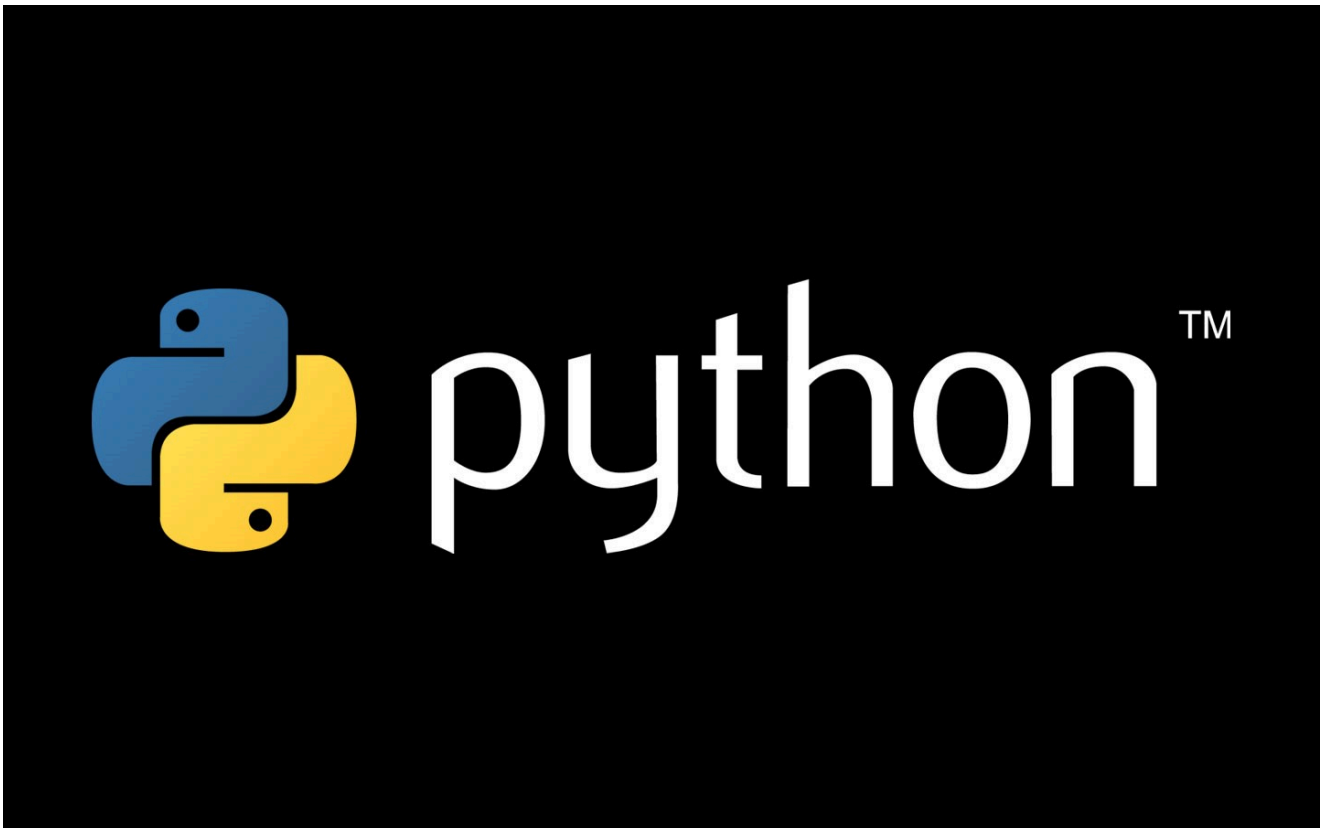


Figure 5.3.7 Python - a programming language

Advantages of Python 3:

- Python's straightforward syntax makes it accessible for beginners and allows quick and easy development from the beginning.
- Python provides a rich standard library and an ecosystem of third-party packages, facilitating development in web development, machine learning, and other areas.
- There is a large and active community that contributes to Python's continuous improvement and offers resources and documentation for learning and troubleshooting.

Disadvantages of Python 3:

- As an interpreted language, Python may exhibit slower execution speeds than compiled languages like C++ or Java.
- Python's flexible data types and ease of use can increase memory usage, making it less suitable for memory-intensive tasks.

5.3.8. Web Server

A web server combines hardware and software to deliver web content to users via an intranet. On the hardware side, it involves a computer that stores website components such as HTML documents, images, CSS stylesheets, and JavaScript files. The software component includes programs that use protocols like HTTP (Hypertext Transfer Protocol) and HTTPS (HTTP Secure) to process requests from clients—typically web browsers—and serve the appropriate content in response.

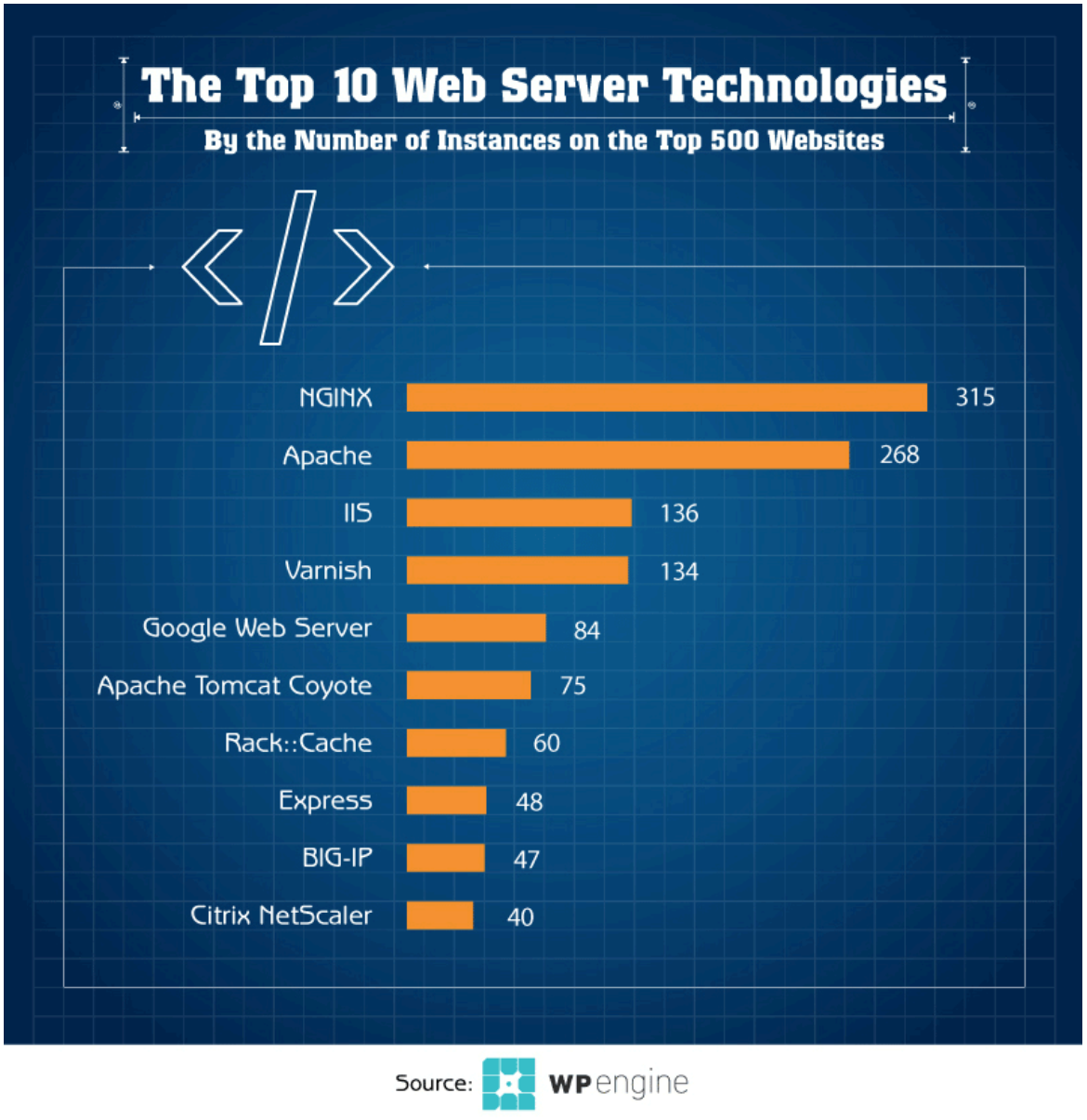


Figure 5.3.8 Top 10 Web server technology

(Source: https://wpengine.com/whats-the-internet-built-on/)

Advantages of Web servers:

- Web servers can handle many connections, making them suitable for high-traffic websites.
- Web servers can support different programming languages, databases, and frameworks, enabling developers to build different applications and websites.
- Web servers can integrate security features such as web app firewalls or intrusion detection systems to minimize the chances of data breaches or being targeted by cyber-attacks.

Disadvantages of Web servers:

- Web servers are often targeted by cyber threats like Denial of Service (DoS) attacks, necessitating robust security protocols to protect data and maintain service availability.
- Setup and maintaining a web server can be challenging; this requires a notable investment in computer hardware and skilled personnel for management and administration.

5.3.9. Telegram

Telegram is a messaging app that is built based on security, speed, and privacy. Developed in 2013, it allows users to send messages, images, videos, and files and create group chats with a large number of members.

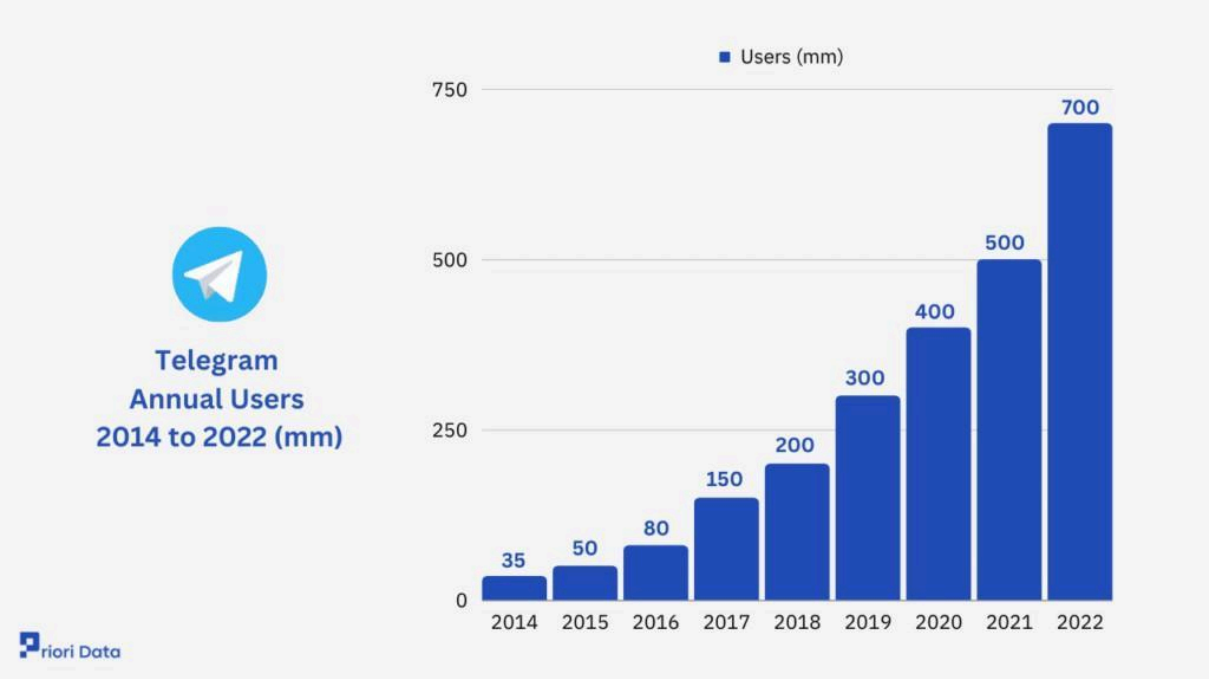


Figure 5.3.9 Telegram user statistics are increasing over the years

(Source: https://prioridata.com/data/telegram-statistics/)

Telegram was founded by Russian brothers Pavel and Nikolai Durov, the founders of the popular Russian social network VKontakte (VK). Telegram has over 700 million monthly users worldwide (as of 2023). Telegram is especially popular in countries with strong restrictions on internet freedom and is often used by users for freedom of information and privacy protection.

Telegram's advantages:

- Telegram is open source and has a public API, allowing developers to create integrated tools and bots easily.

- Telegram offers end-to-end encryption for Chats and allows users to use other security features, such as passwords, two-factor authentication, and self-destructing messages.

Disadvantages of Telegram:

- Telegram can consume a lot of storage and device memory due to unlimited cloud storage and large data downloads.

5.4. Technologies

We use the Ubuntu virtual machine to build the ELK Stack system (including Elasticsearch, Logstash, and Kibana). Around this system, we will integrate an Intrusion Detection System (IDS) to detect abnormal network traffic or abnormal access through the Web server. When someone accesses, the findings will be saved as logs. Then, the Syslog-ng integrated into pfSense will send those logs to the ELK system for synthesis so the administrator can detect them in time.

In the given context, we will assume a scenario where the attacker has accessed the website to perform attacks. The assumptions we aim for are Scan attacks, SQL Injection, and DOS attacks. In this situation, the Intrusion Detection System (IDS) will detect abnormal signs and send them to Elastic Stack. It will send messages to the system administrator via Gmail and Telegram. Finally, ELK will send the attacker's IP to the Firewall (Iptables) to promptly stop the attacks.

5.5. Environment

To test the “ELK-based network intrusion detection system” we created, we used several virtual machines with custom configurations.

5.5.1. Ubuntu

We build a Ubuntu machine with the following installation:

- Operation System_version: Ubuntu 20.04 LTS.
- Software installed: ELK (Elasticsearch, Logstash, Kibana)

-> Acts as the server for receiving and processing logs from other machines to be analyzed by ELK for intrusion detection.

5.5.2. Kali

We build a Kali linux machine thewith following installation:

- Operation System_version: Kali
- Pre-installed tools: Nmap, Burp Suite, and other penetration testing tools.
- Provides attack traffic.

-> Used to perform various types of attacks such as network scanning, Denial of Service (DoS), and SQL Injection on the target machine (Ubuntu).

5.5.3. Ubuntu Server

We build a Ubuntu machine with the following installation:

- Operation System_version: Ubuntu 20.04 LTS.
- Software installed: Apache2

-> Acts as the victim web server to be attacked. Collects and sends web server logs to be analyzed by ELK for intrusion detection.

5.6. Building Test

5.6.1. Building a network intrusion detection system based on ELK Stack

5.6.1.1. Interface

In the current landscape of increasingly complex and diverse network attacks, robust log management and analysis tools are essential for effective intrusion detection and network security. The ELK Stack, comprising Elasticsearch, Logstash, and Kibana, serves as a powerful solution by enabling the collection, storage, and analysis of logs from multiple sources, facilitating the early detection of network intrusion activities.

To enhance network defense mechanisms, we propose establishing an integrated system that leverages the ELK Stack's capabilities to support key features such as comprehensive logging, blocking of anomalous IP addresses, automated reporting of suspicious activities, and the creation of intuitive dashboards. By harnessing these features, administrators can efficiently detect and respond to abnormal signs indicative of network intrusions. The effectiveness of each component will be demonstrated in detail in the following sections.

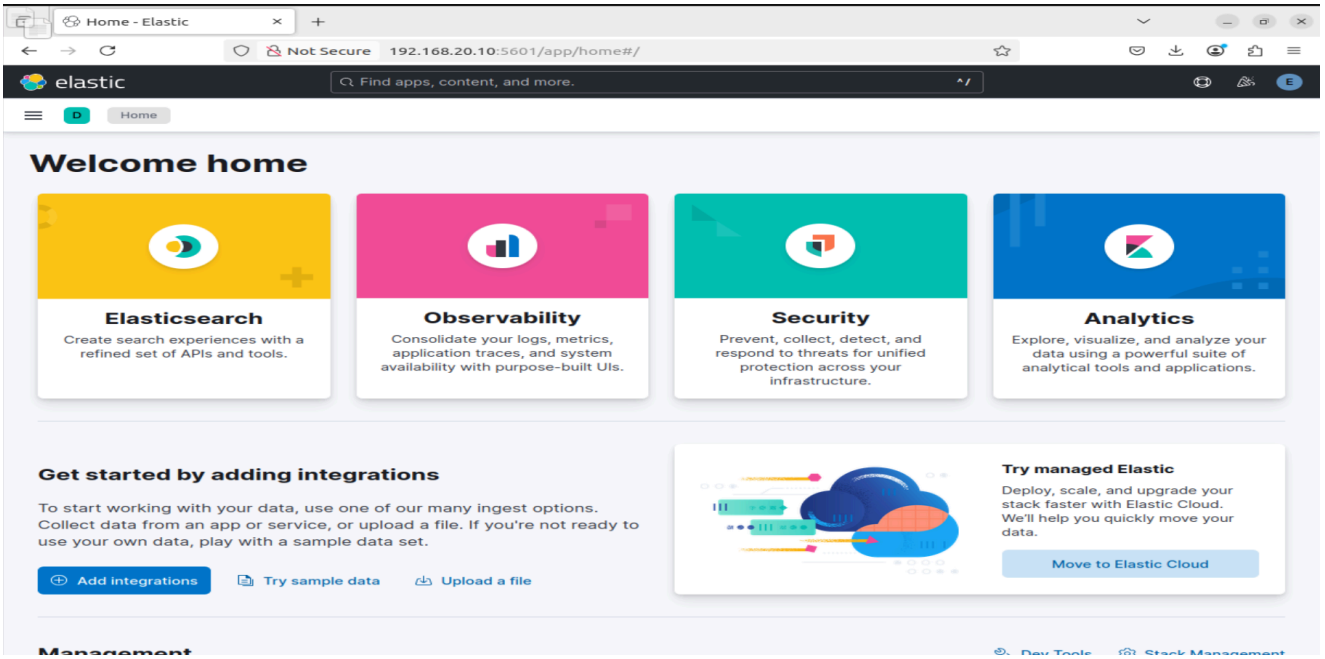


Figure 5.6.1.0 Elastic homepage

5.6.1.2. Logging

ELK is capable of collecting logs from various sources (Web Server, IDS, attack types), providing comprehensive data on system activities. By analyzing logs, administrators can detect unusual signs or unwanted activities, which is the basis for intrusion detection.

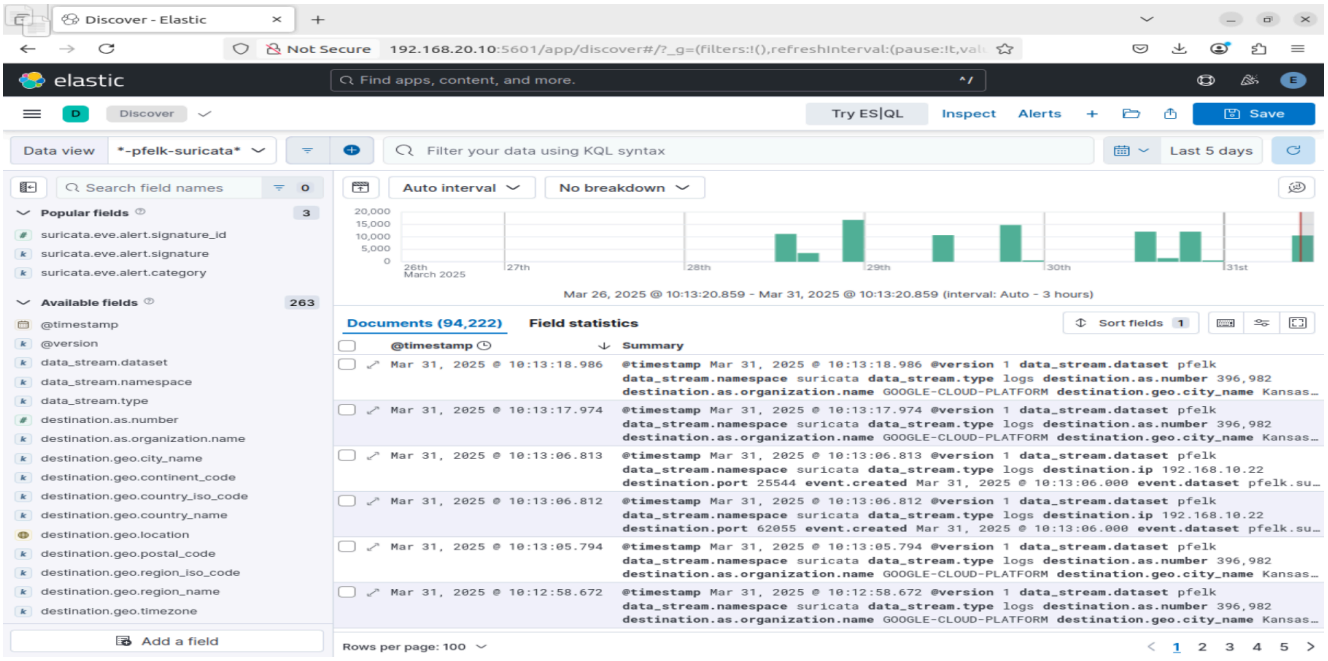


Figure 5.6.1.1 Display of successfully captured log

5.6.1.3. Blocking abnormal IPs

Through log streams and alert rules, Suricata allows monitoring and detection of IPs with abnormal behavior. When detecting intrusions, the system integrates with Blocking Feature to automatically block suspicious IPs, reducing the risk of attacks.

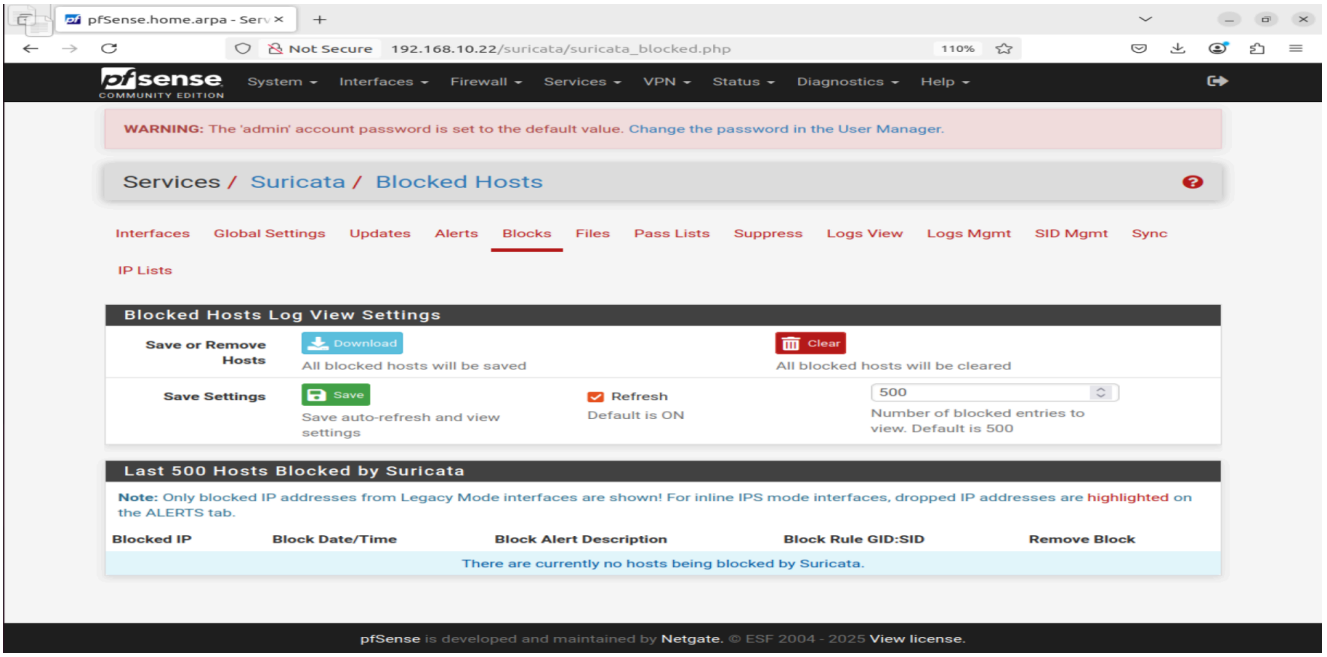


Figure 5.6.1.2 Blocking unusual IPs in Suricata

5.6.1.4. Sending abnormal reports

Elastic Stack supports integration with notification services such as Gmail and Telegram, allowing administrators to receive alerts as soon as an anomaly occurs. This ensures that security issues are addressed promptly, minimizing response and remediation time.

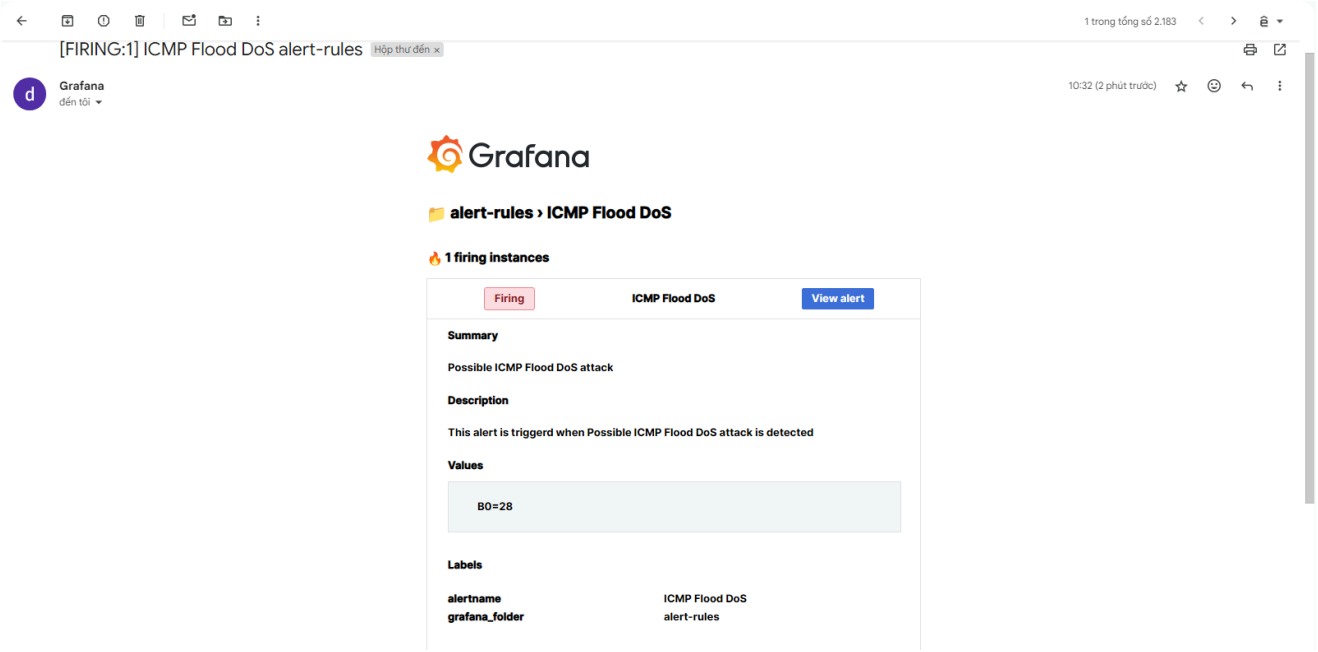


Figure 5.6.1.3.1 Gmail Intrusion Detection Notification

Figure 5.6.1.3.2 Telegram Intrusion Detection Notification

5.6.1.5. Creating dashboards

Elastic Stack provides highly customizable Dashboards that visually display information about network activity, unusual events, and system status. Through the

Dashboard, administrators can monitor the system status and take necessary actions based on real-time information.

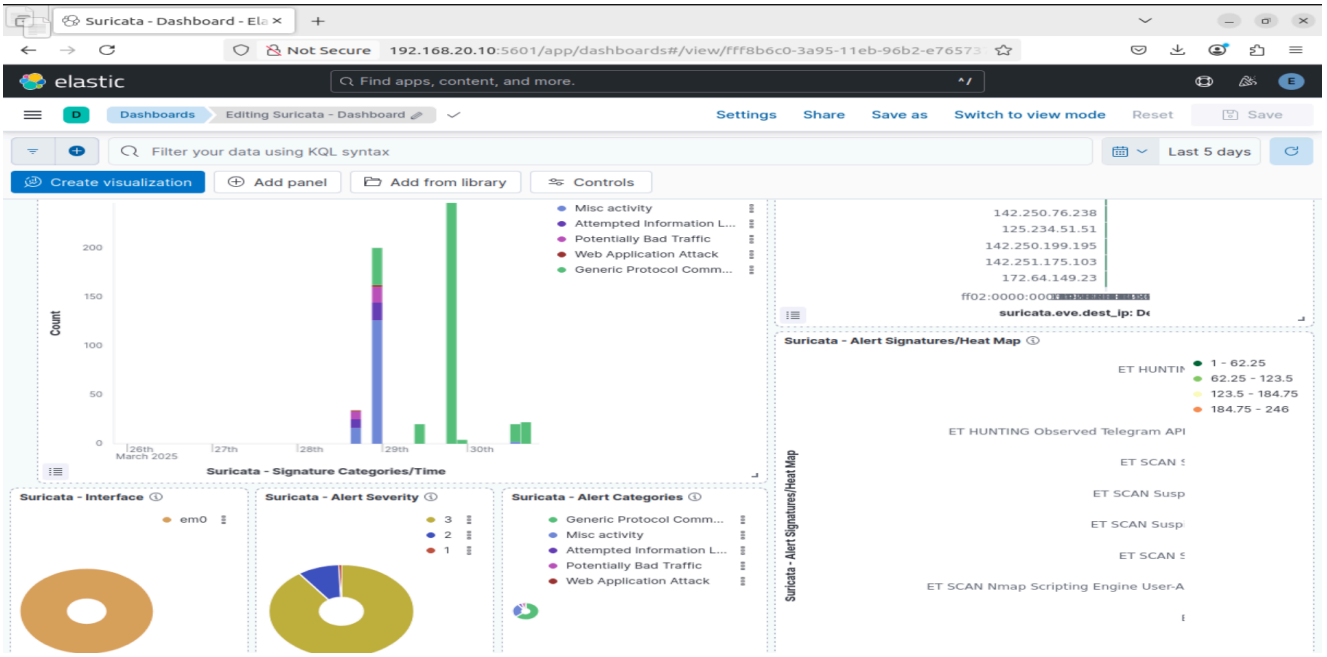


Figure 5.6.1.4 Creating dashboards

5.7. Demonstration

5.7.1. Scenario 1: DOS attack

5.7.1.1. Scenario

First, the attacker will enter a Kali Linux virtual machine to perform a DOS attack. He will executes a command like this:

hping3 --flood -1 doancuoinam.id.vn

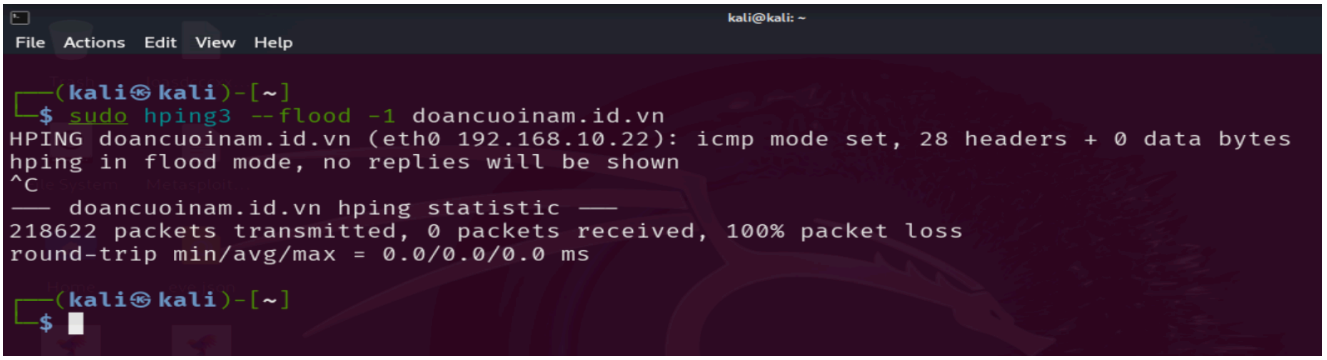


Figure 5.7.1 Enter DOS attack command on Kali Linux

5.7.1.2. Result

Elastic Stack successfully captured the log of DOS attack using hping3 command on Kali machine by sending a series of payloads to the Web server.

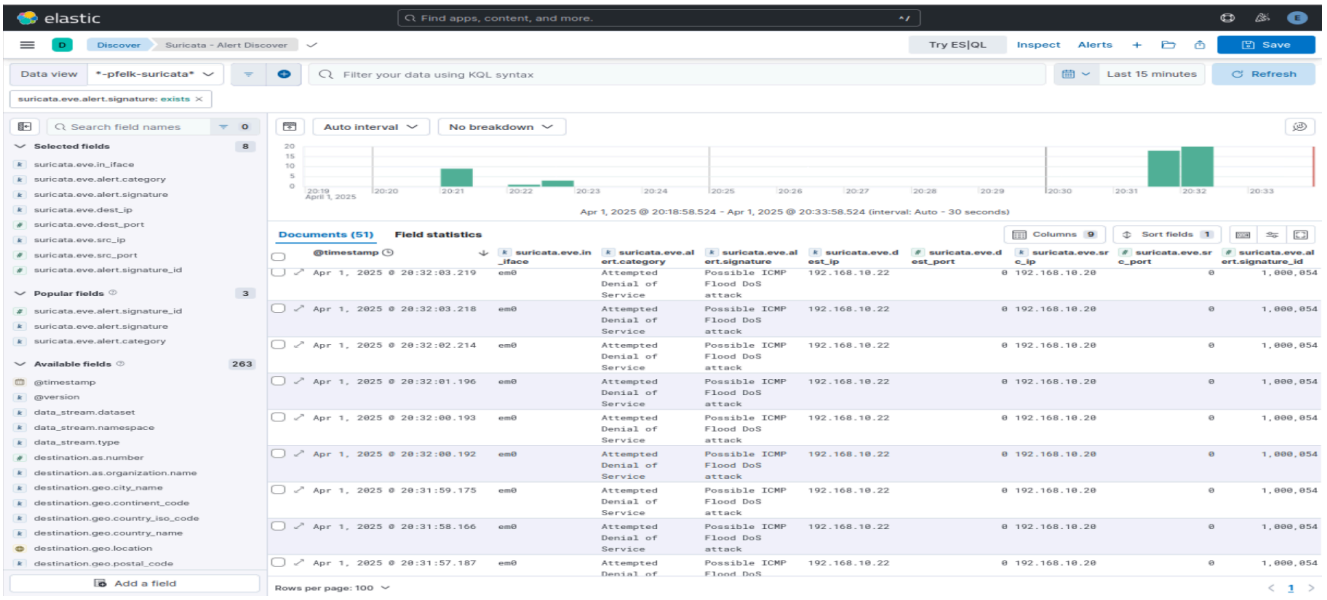


Figure 5.7.1.1 The result shows that ELK has captured the log

5.7.2. Scenario 2: SQL Injection attack

5.7.2.1. Scenario

First, we will identify the victim's website. Then, we will Sign up a user account in order to perform the attack

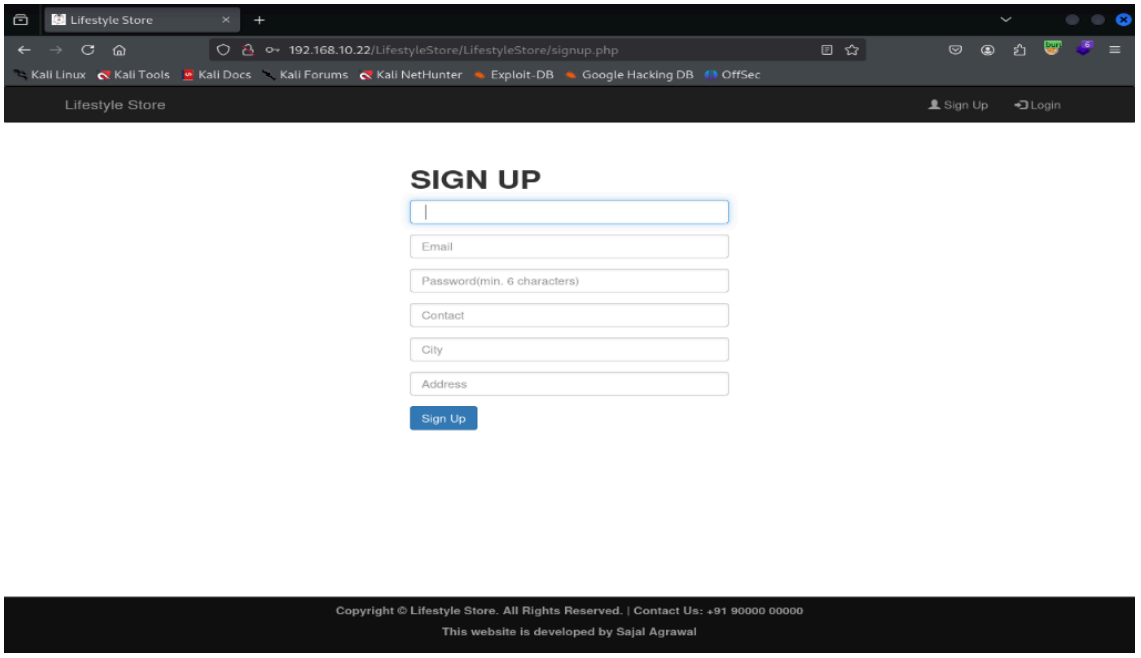


Figure 5.7.2.0.1 Sign up page of victim server

Signing up with the information

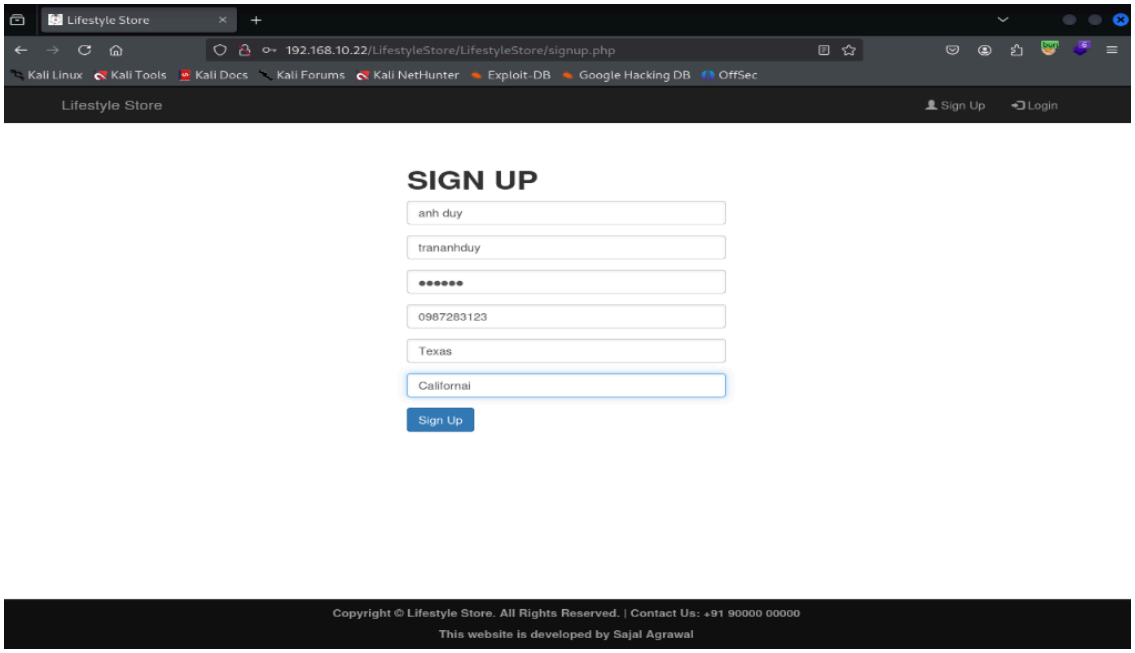


Figure 5.7.2.0.2 Input some information in the sign up page

After signing up, we will see the victim web page. We will perform SQL Injection attacks on the Cart page when we add products to cart.

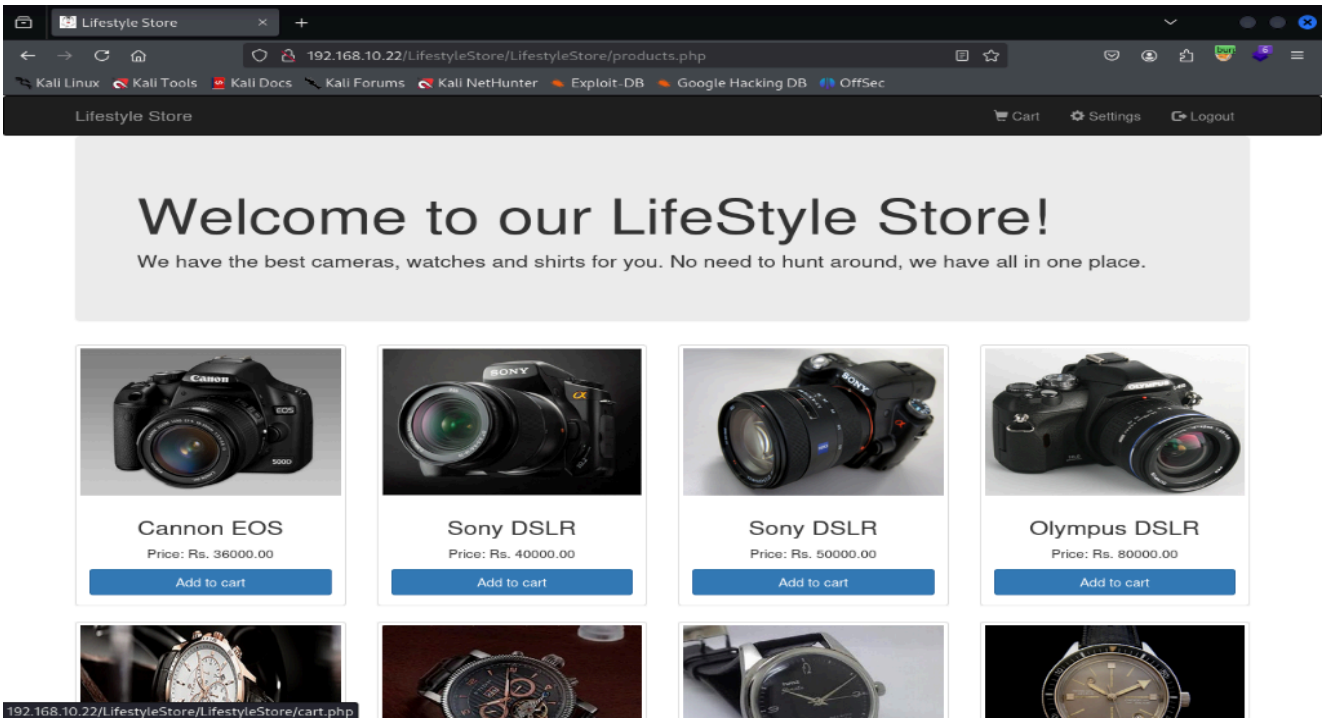


Figure 5.7.2.0.3 Homepage of the victim website

Using Burp Suite to capture the action of adding a product to cart

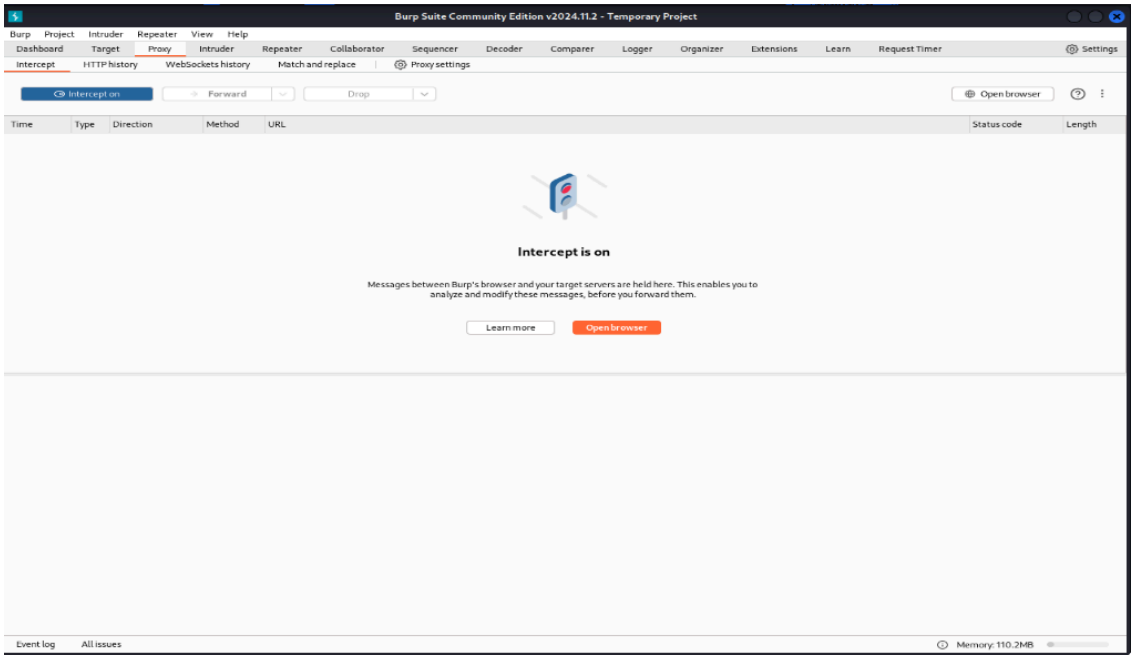


Figure 5.7.2.0.4 Burp suite home page

Next, to perform SQL Injection on the website, we capture the “add product to cart” action by using Burp Suite proxy

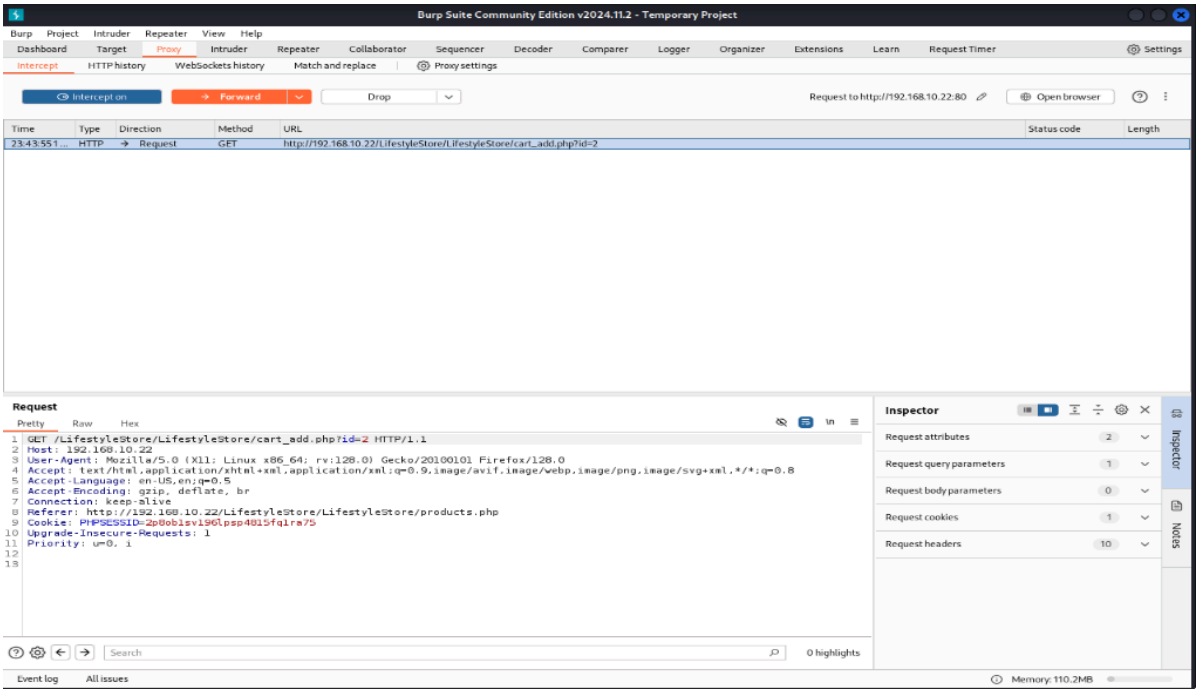


Figure 5.7.2.0.5 “Add to cart” action captured

We will use sqlmap to perform SQL injection attack on the URL: http://192.168.10.22/LifestyleStore/LifestyleStore/cart_add.php?id=2 and submit them.

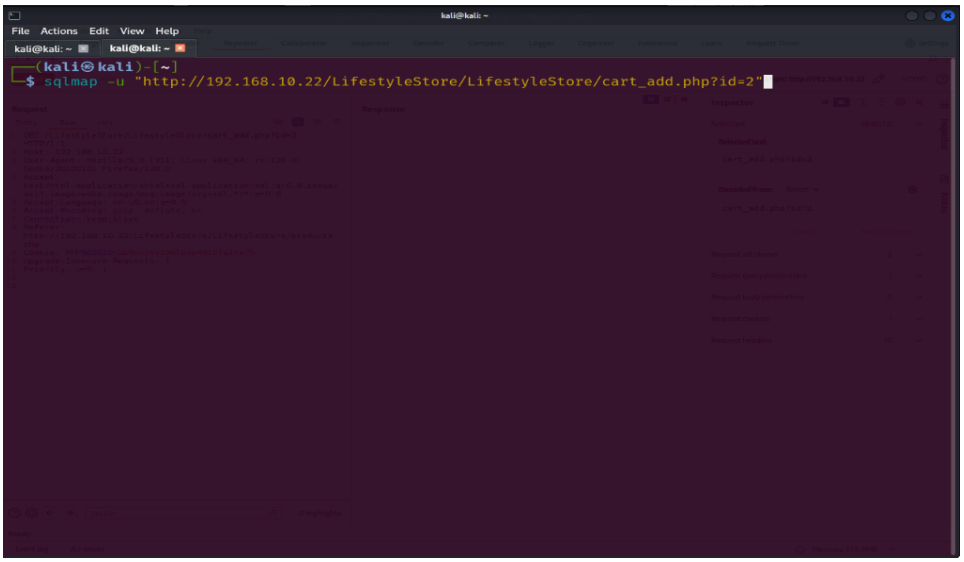


Figure 5.7.2.0.6 Perform SQL Injection using sqlmap

5.7.2.2. Result

After that, the SQL Injection attack is complete, and shows some information about the victim server.

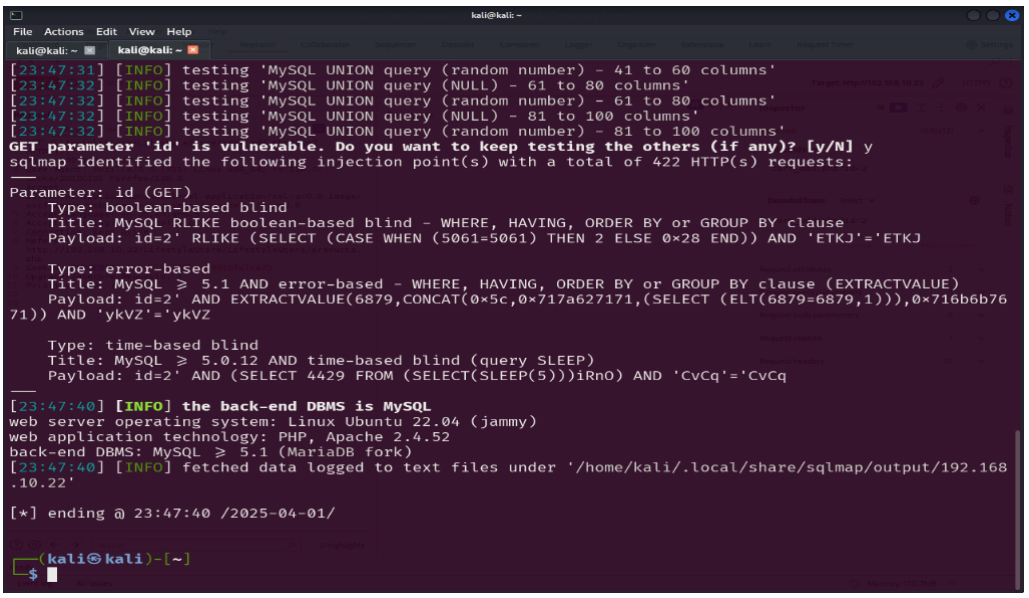


Figure 5.7.2.1 Successful SQL Injection attack

ELK Stack has captured logs of attacks: SQL Injection attack.

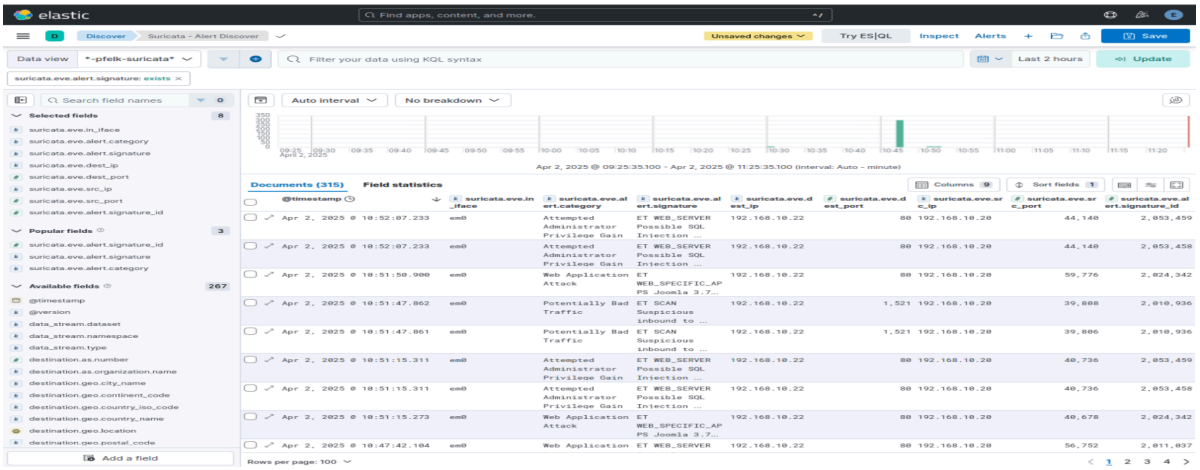


Figure 5.7.5.2. The result shows that ELK has successfully captured the log

5.7.3. Scenario 3: Scan attack

5.7.3.1. Scenario

First, the attacker opens the terminal. We will use Nmap to scan the target website for vulnerability. The command is nmap –script=vuln

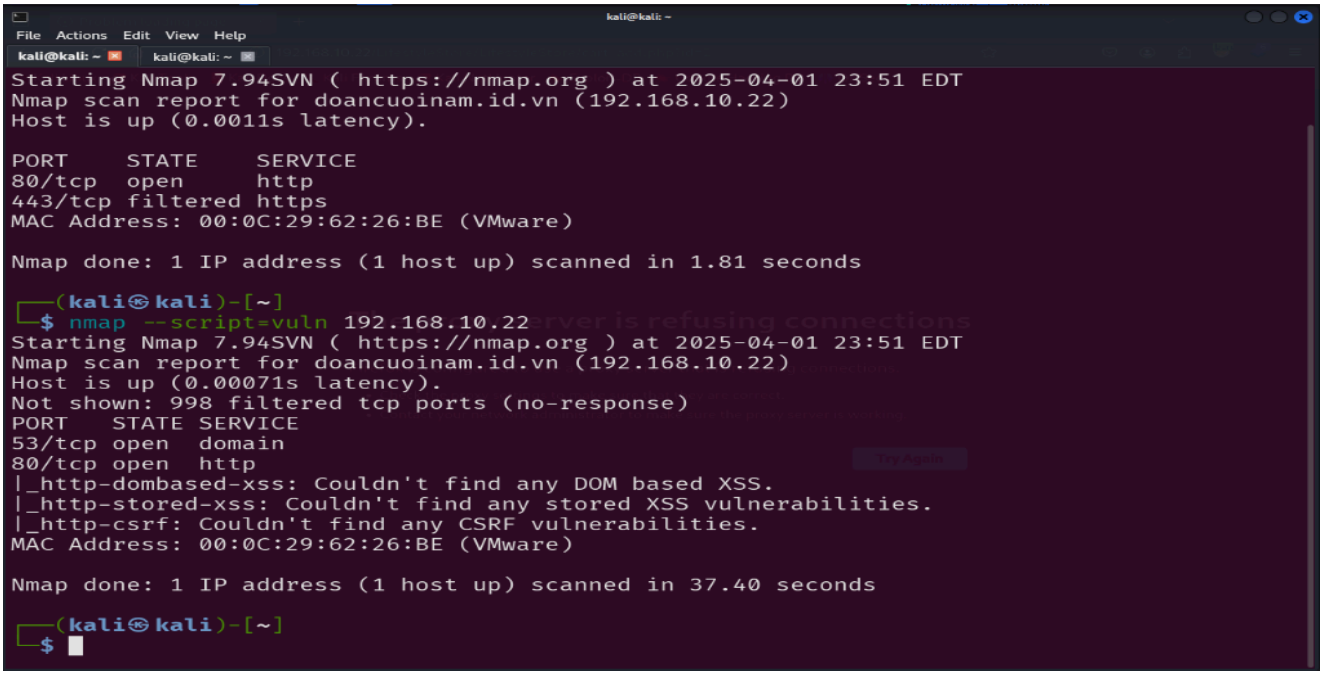


Figure 5.7.3.0 The attacker uses the nmap tool

5.7.3.2. Result

ELK successfully captured the Scan attack log using Burp Suite by sending a series of payloads to ELK.

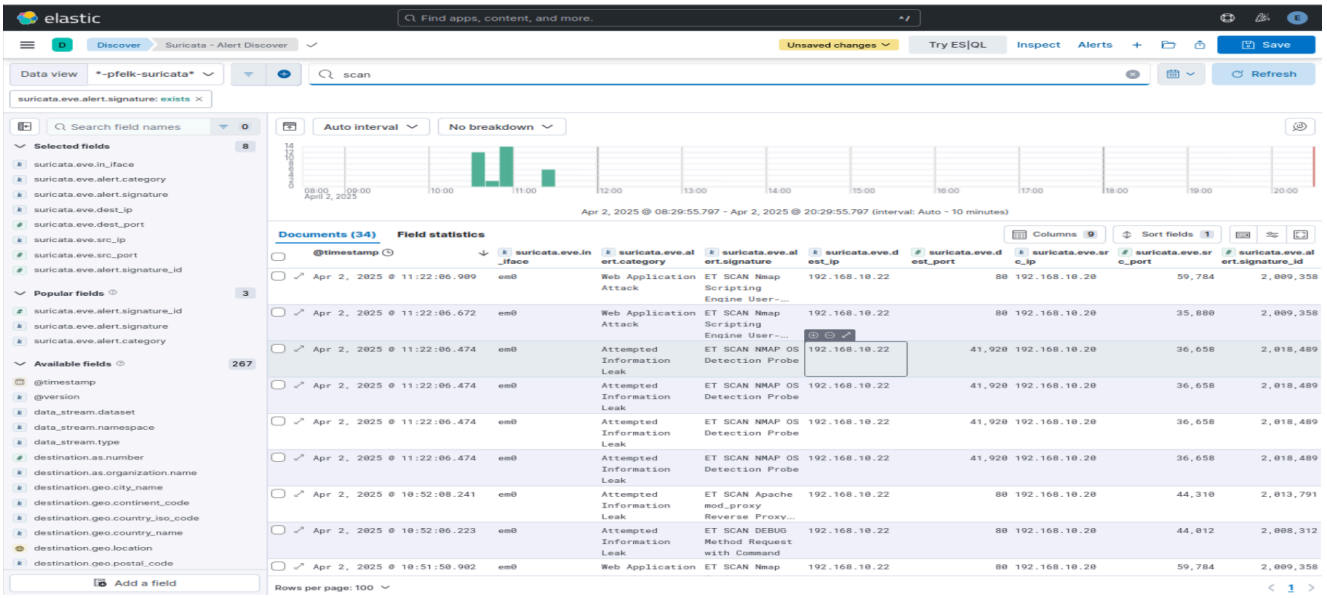


Figure 5.7.3.1 The result shows that ELK has successfully captured the log

5.7.4. Scenario 4: Command Injection Attack: Execution and Backdoor Installation

5.7.4.1. Executing Command Injection on DVWA

In this step, the **Command Injection** vulnerability in DVWA (Damn Vulnerable Web Application) is exploited by injecting malicious commands into an input field that directly interacts with the system shell. The goal is to execute arbitrary system commands on the target machine.

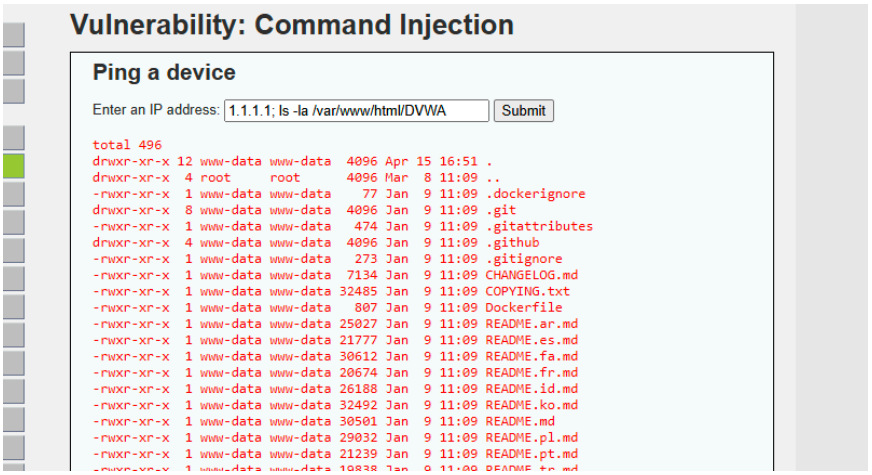


Figure 5.7.4.0 DVWA Command Injection

5.7.4.2. Identifying the secret.txt File Containing SSH Credentials

The next step is to locate sensitive information that could be used to gain further access to the system, such as SSH credentials.

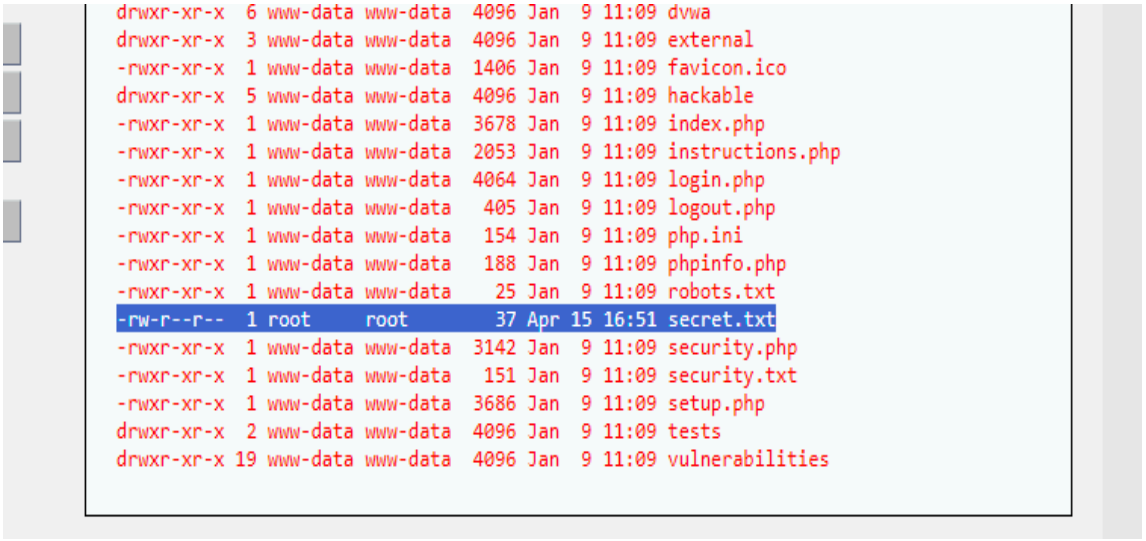


Figure 5.7.4.1.1 A file contains sensitive information

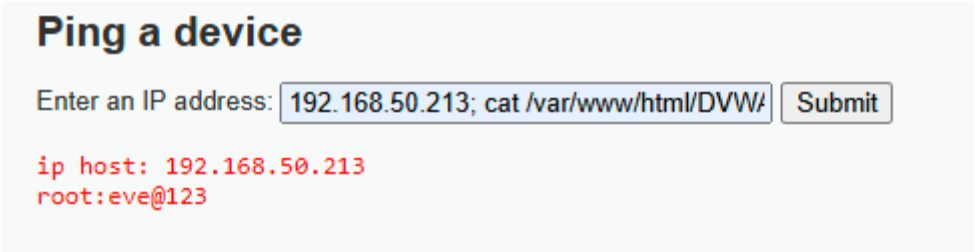


Figure 5.7.4.1.2: Read the content of secret.txt

5.7.4.3. Gaining SSH Access

After discovering the SSH login credentials stored in the secret.txt file, the attacker proceeds to access the victim's server using the SSH protocol with the obtained information. This step grants the attacker remote access to the target system under a legitimate user account, allowing them to carry out further malicious actions such as installing backdoors or maintaining long-term access.

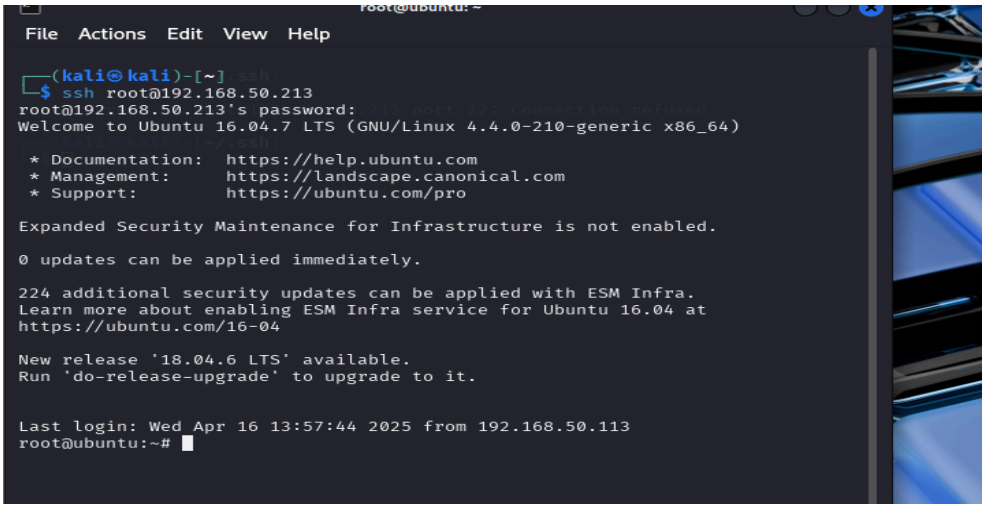


Figure 5.7.4.2 SSH to web server

5.7.4.4. Installing a Backdoor and Maintaining Access

Maintain access by installing a **backdoor** that will allow the attacker to reconnect at any time, even if the system is restarted.

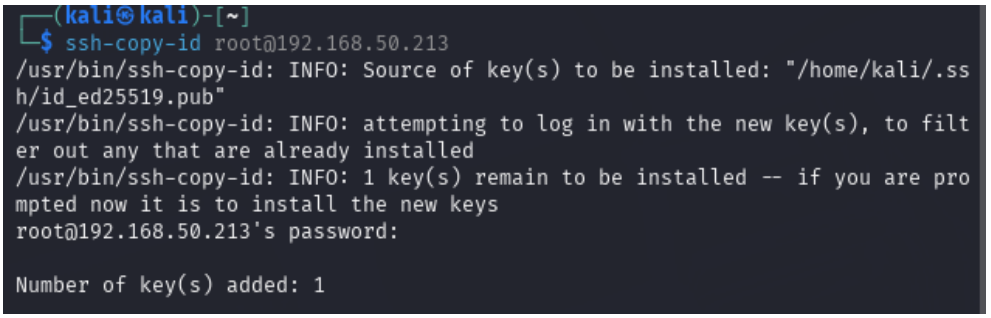


Figure 5.7.4.3 Install a backdoor

5.7.4.5. Reconnecting via SSH and Backdoor

After the system reboots and the backdoor is active, the attacker can reconnect to the victim machine using Netcat, maintaining persistence.

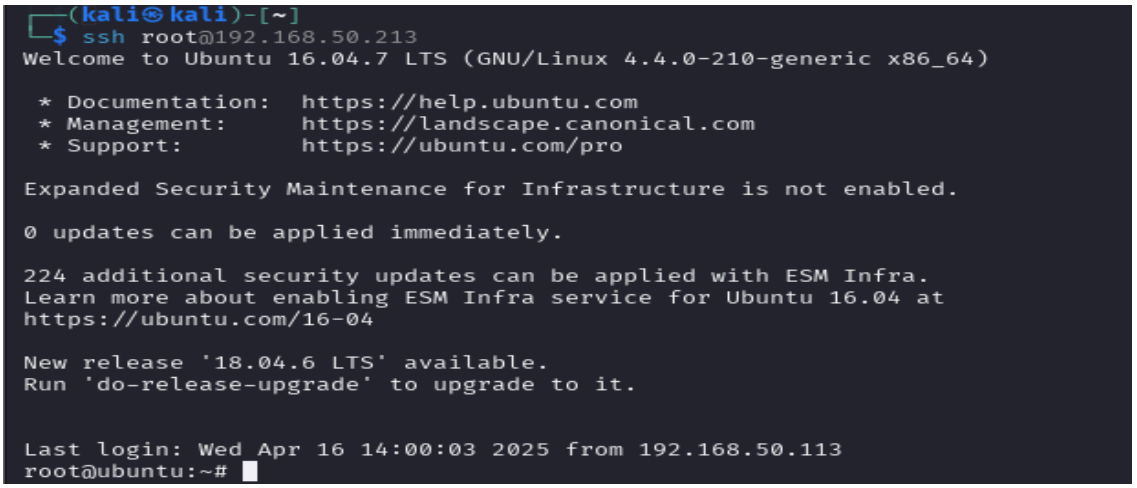


Figure 5.7.4.4 Successfully maintain persistence

5.7.4.6. Result

The ADNIDS system successfully detects these abnormal attacks and sends alerts through Telegram for admin.

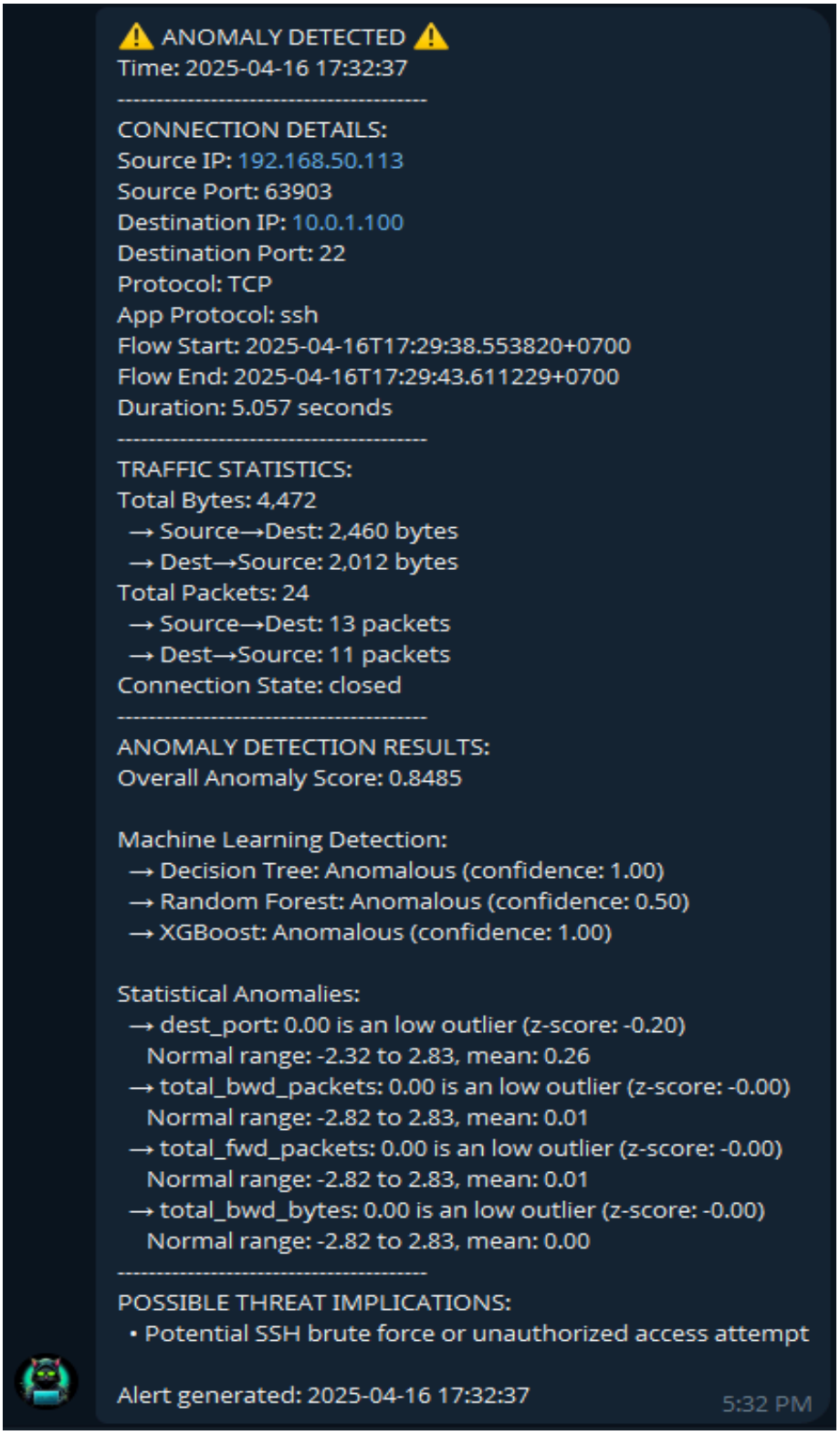


Figure 5.7.4.5 ADNIDS system successfully detects

5.8. Detection and processing

5.8.1. Introduction

After analyzing cyber attack techniques targeting websites, we have identified several highly dangerous and particular threats in today's network environment. The key challenge is detecting and preventing these attacks on the Website in real-time. To address this, our team has researched and agreed on using Suricata, which is capable of quickly identifying network intrusions. When suspicious activity is detected, it generates an alert, enabling administrators to take immediate action.

We found a shared characteristic by studying three common website attacks: they all exploit vulnerabilities and overload systems to disrupt operations or steal sensitive data. Based on this insight, we will use Suricata IPS to continuously monitor website access patterns and track IP addresses. If a match with a known threat is found, it logs the event and issues an alert, allowing administrators to respond swiftly and mitigate potential risks.

5.8.2. Demonstration of detection and processing

In this demo, we will do a DOS attack on a website. Here, we will perform a DOS attack on the website **http://doancuoinam.id.vn** and the ELK Stack will catch this unusual intrusion detection.

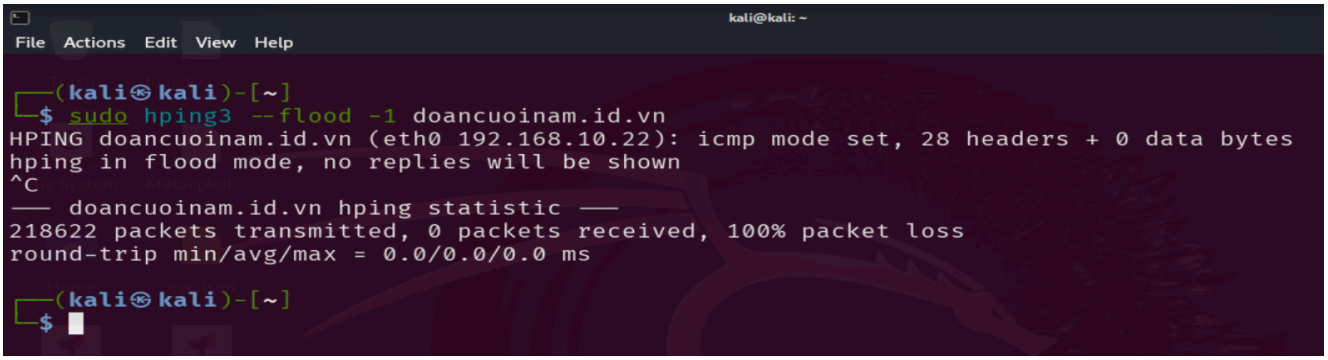


Figure 5.8.2.1 Perform a DOS attack on the website

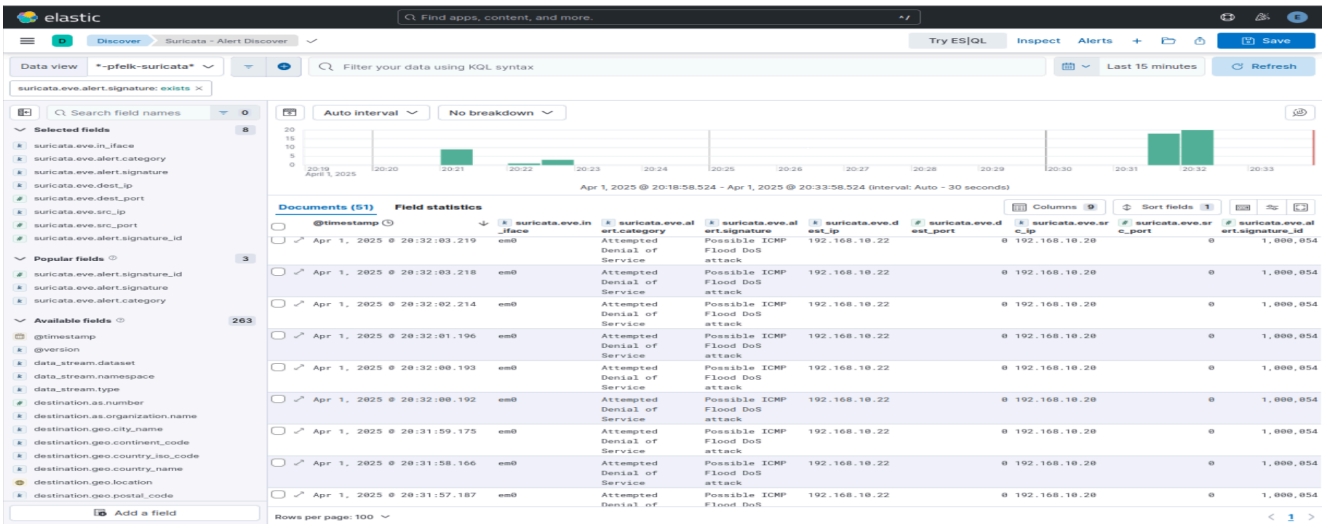


Figure 5.8.2.2 The result shows that Elastic Stack has successfully captured the log

After a successful DOS attack, Elastic will send a notification to 2 messaging services, Gmail and Telegram. At the same time, it will block IPs that detect abnormalities.

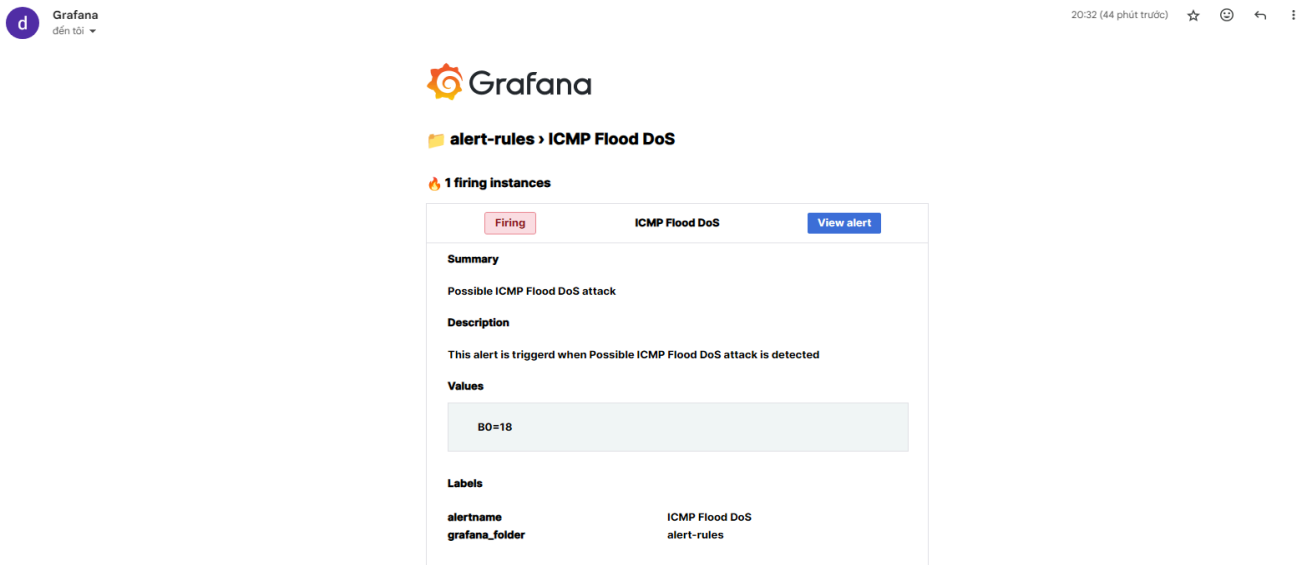


Figure 5.8.2.3 Elastic will send notification to Gmail

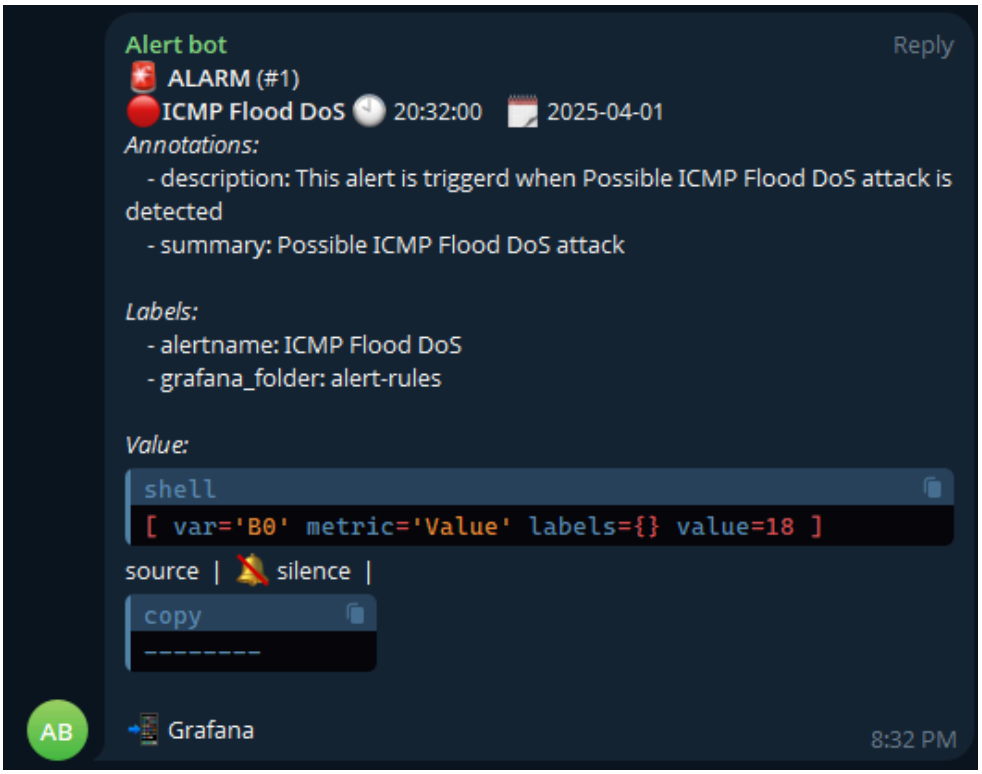


Figure 5.8.2.4 Elastic will send notification to Telegram

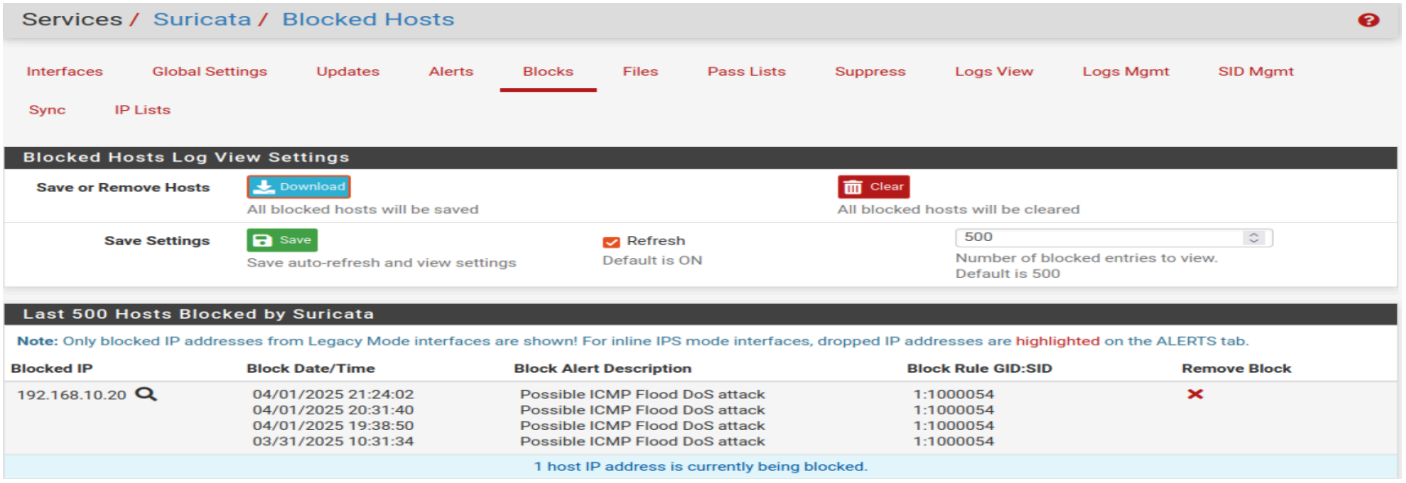


Figure 5.8.2.5 Suricata will block IPs that detect abnormalities based on rules

After a SSH attack, which cannot be alerted by Elastic because of not setting rules. The ADNIDS system can detect it by analyzing by machine learning and will send a notification to Telegram.

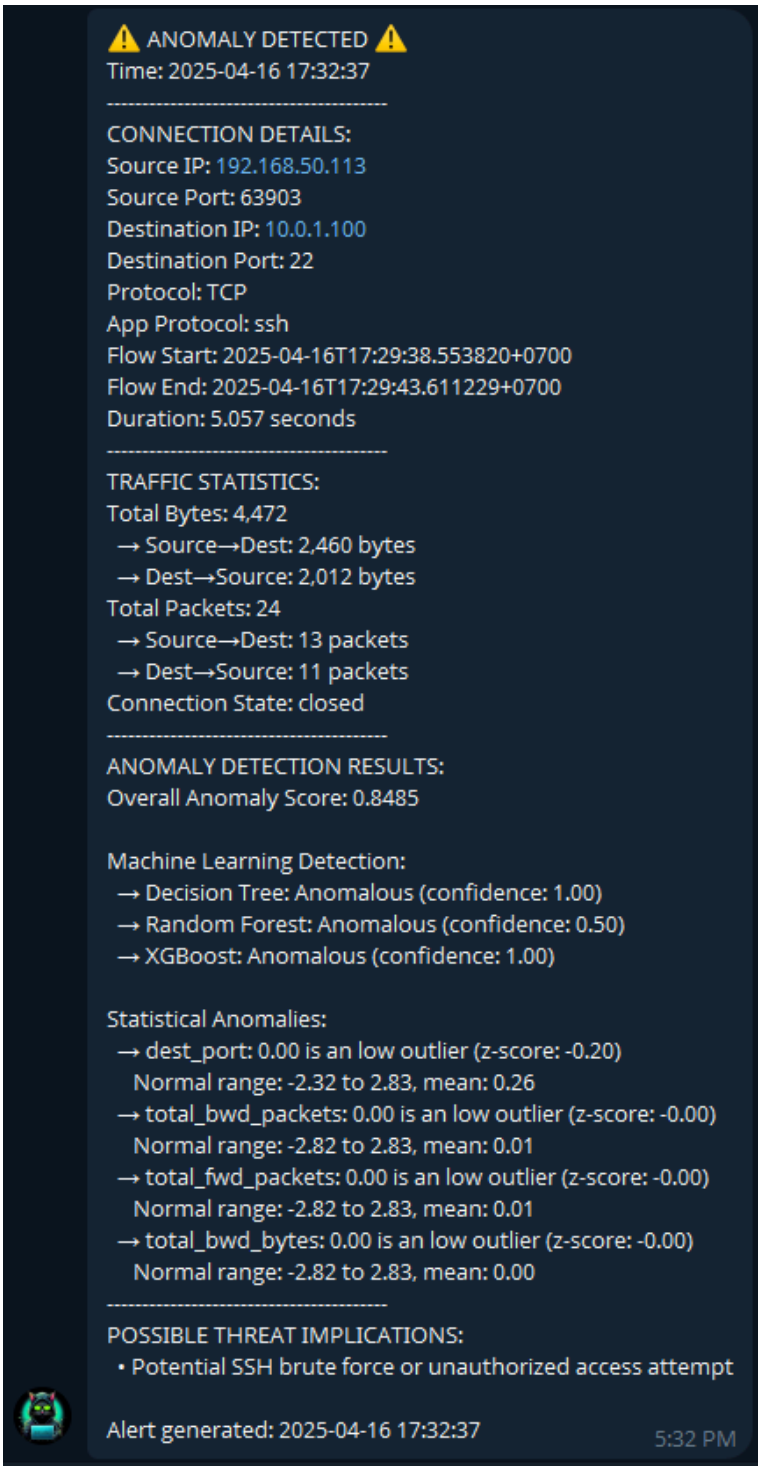


Figure 5.8.2.6 Suricata will block IPs that detect abnormalities based on rules

Chapter 6: VALIDATION DOCUMENT

6.1. Project Result

The project has produced significant findings in the analysis, simulation, monitoring, and mitigation of various network-based attacks. By extensively researching and simulating different network intrusion scenarios, we gained valuable insights into the methods commonly employed by the attackers. Using these insights, we developed a robust method to detect and monitor suspicious network activity, enhancing the ELK Stack's capabilities to

actively identify and alert on malicious behaviors. This advancement reinforces the ELK Stack as an essential tool for network security, efficiently identifying, tracking, and mitigating potential threats.

6.2. Advantages and Disadvantages of Solution

6.2.1. Advantages

- The proposed method demonstrates high efficiency in detecting and monitoring network attack activities.
- The method not only focuses on detecting anomalous activities but also provides timely alerts to administrators.
- The solution is designed to address current monitoring limitations by providing a scalable, flexible and stable platform for log monitoring.

6.2.2. Disadvantages

- Although the proposed solution has shown promising results, it may not be universally applicable in all scenarios or capable of detecting all variants of network attacks.
- Some specialized attack techniques may pose challenges to the solution in terms of detection or mitigation effectiveness.

6.3. Development Strategy of Solution

- **Conduct Research and Information Collection:** Conduct a thorough research of the techniques to collect logs and understand how they operate.
- **Simulate various cyber attack scenarios:** Simulate various cyber attack scenarios to better understand the functionality and feasibility of these techniques.
- **Analyze and propose detection and mitigation methods:** Analyze collected data and propose detection and mitigation methods based on technical aspects and behavioral patterns related to network intrusions.
- **Deploy and Test:** Deploy the proposed technique and test it in a real environment to evaluate its performance and effectiveness.
- **Optimize the technique and propose improvements:** Optimize the technique based on the test results and propose further improvements to improve detection and mitigation capabilities.

6.4. References

[1]	*Installing a self-managed Elastic Stack on Ubuntu* https://www.elastic.co/guide/en/elastic-stack/current/installing-stack-demo-self.html
[2]	*Kibana Dashboard* https://www.elastic.co/guide/en/kibana/current/create-dashboard.html
[3]	*Kibana Alert* https://www.elastic.co/guide/en/kibana/current/alerting-getting-started.html
[4]	*Access log in ELK* https://www.elastic.co/guide/en/enterprise-search/current/logging-view-query-logs.html
[5]	*Block unusual IPs* https://snapshooter.com/blog/how-to-block-ip-accessing-your-linux-server-with-iptables-and-ufw-firewall
[6]	*Install pfSense and Suricata IDS* https://tech.lobobrothers.com/en/implementing-pfsense-with-suricata/
[7]	*Configure Suricata on pfSense* https://tech.lobobrothers.com/en/configuring-suricata-in-pfsense/
[8]	*Telegram notification* https://gist.github.com/gelldur/94b57b2fa276fe9de180378bf6855877
[9]	*Setup Logstash* https://www.elastic.co/guide/en/logstash/current/setup-logstash.html
[10]	*Dos Attack* https://www.cloudflare.com/learning/ddos/ddos-attack- tools/how-to-ddos/
[11]	*Scan Attack* https://portswigger.net/burp/documentation/desktop/running- scans
[12]	*Sql Injection Attack* https://www.evolvesecurity.com/blog-posts/tools-of-the-trade-your-ally-in-uncovering-sql-injection-vulnerabilities
[13]	*Slips: Behavioral Machine Learning-Based Intrusion Prevention System* *https://stratospherelinuxips.readthedocs.io/en/develop/index.html*
[14]	*A Suricata and Machine Learning Based Hybrid Network Intrusion Detection System* *https://www.researchgate.net/publication/357785493_A_Suricata_and_Machine_Learning_Based_Hybrid_Network_Intrusion_Detection_System*
[15]	*Intrusion Detection (CIC-IDS2017)* *https://github.com/noushinpervez/Intrusion-Detection-CICIDS2017*

[17]	*NIDS_ML_HWR* *https://www.kaggle.com/code/janrggeberg/nids-ml-hwr*
[18]	*MTH-IDS: A Multi-Tiered Hybrid Intrusion Detection System for Internet of Vehicles* *https://arxiv.org/pdf/2105.13289.pdf*
[19]	*Suricata* *https://docs.suricata.io/en/latest/what-is-suricata.html*