People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

UNIVERSITY OF ABDELHAMID MEHRI – CONSTANTINE 2

Faculty of New Technologies of Information and Communication (NTIC)

Department of Fundamental Computing and its Applications (IFA)

MASTER'S THESIS

to obtain the diploma of Master degree in Computer Science

Option: Networks and distributed systems (RSD)

Design and construction of an operational security center (SOC) For University of Constantine2

Realized by:

Kerrouche Aymen

Guessab Hichem

Under supervision of:

Dr.Benayoune Salim

June 2023

بسم الله الرحمن الرحيم

Acknowledgments

It is with great pleasure that we reserve these few lines as a sign of gratitude and deep gratitude to all those who, from near and far, have contributed to the realization and the outcome of this work.

We would first like to thank Mr Benayoune Salim for his support, his seriousness his kindness and especially for his invaluable help throughout the development of this work.

Our thanks to our dear parents for always giving us the courage and to be the main reason we fight every obstacle.

We willingly express our gratitude and appreciation to all our teachers for the quality of the education they have kindly provided during our studies, equipping us with valuable skills and knowledge.

Finally, we extend our heartfelt gratitude to all our loved ones who supported and encouraged us throughout this journey. We sincerely appreciate their assistance in completing this work...

Dedication

I dedicate this modest work to the two people who are very dear to me, my parents, who raised me and educated me and have always been there for me and never stopped believing in me, no word or language could express my deep gratitude to them. To the whole family,my old sisters and my big brother Hakim and to my little nieces and nephews Yahia, Anfel, Nesrine, Rania, Amir, Aymen, Majdi, Achraf. And to the family who took care of me during my last two years in Constantine. To all my friends Aymen, Zaki, Oussama, Houssam , Touhami, Amir, Akram, Ramzi and all those I met in Constantine and to my loved ones...

Guessab Hichem

I dedicate this modest work to the two people who are very dear to me, my parents, who raised me and educated me and have always been there for me and never stopped believing in me, no word or language could express my deep gratitude to them. To the whole family,my brother Newfel and my little sister Douaa. And to the family who took care of me during my last two years in Constantine. To all my friends Hichem, Zaki, Oussama, Zinou, Ahcen , Islem, Salah, Akram, and all those I met in Constantine and to my loved ones...

Kerrouche Aymen

ملخص

تكتسب مراكز العمليات الأمنية SOCs أهمية كبيرة بسبب العدد المتزايد من التهديدات الإلكترونية التي تواجهها المنظمات والحكومات والشركات، وتتولى مراكز العمليات الخاصة مسؤولية تنفيذ عمليات الأمن المركزي لمنع وقوع حوادث أمنية كبيرة، وهي تستخدم العناصر البشرية والتكنولوجية على السواء وتعتمد على تقنيات رئيسية مثل التدقيق الأمني وكشف الاقتحام لضمان موثوقية نظم المعلومات.

تحدد هذه الأطروحة عملية إنشاء مركز عمليات أمنية SOC وتناقش المراحل المختلفة التي ينطوي عليها تصوره وتنفيذه، الهدف الرئيسي للمنصة هو اكتشاف التهديدات السيبرانية المحتملة في مرحلة مبكرة، مما يسمح بمعالجتها في الوقت المناسب والكشف الدائم عن التهديدات من خلال استخدام أدوات مفتوحة المصدر. من خلال القيام بذلك، تهدف SOC إلى تقليل تأثير الحوادث الأمنية إلى الحد الأدنى.

الكلمات المفتاحية: مراكز العمليات الأمنية، التهديدات الإلكترونية، الحوادث الأمنية، التدقيق الأمني، الكشف عن الاقتحام

Abstract

Security operations centers (SOCs) are gaining importance due to the rising number of cyber threats faced by organizations, governments, and companies. SOCs are responsible for implementing central security operations to prevent major security incidents. They utilize both human and technological elements and rely on key techniques such as security auditing and intrusion detection to ensure the reliability of information systems.

This thesis outlines the process of creating a security operation center (SOC) and discusses the different stages involved in its conception and implementation. The main objective of the platform is to detect potential cyber threats at an early stage, allowing for timely remediation and permanent threat detection through the use of open source tools. By doing so, the SOC aims to minimize the impact of security incidents.

Keywords: Security Operations Centers,cyber threats,security incidents,Security auditing,intrusion detection

Résumé

Centre de sécurité opérationnelle (SOC) gagnent en importance en raison du nombre croissant de cybermenaces auxquelles font face les organisations, les gouvernements et les entreprises. Les SOCs sont responsables de la mise en œuvre des opérations de sécurité centrale pour prévenir les incidents de sécurité majeurs. Ils utilisent des éléments humains et technologiques et utilisent des techniques clés comme la vérification de la sécurité et la détection des intrusions pour assurer la fiabilité des systèmes d'information.

Cette thèse décrit le processus de création d'un centre de sécurité opérationnelle (SOC) et discute les différentes étapes de sa conception et de sa mise en œuvre. Le principal objectif de la plateforme est de détecter les cybermenaces potentielles à un stade précoce, ce qui permet de les corriger rapidement et de les détecter de façon permanente au moyen d'outils libres. Ce faisant, le COS vise à minimiser l'impact des incidents de sécurité.

Mots clés : centre de sécurité opérationnelle, cyber menaces, incidents de sécurité, audit de sécurité, détection d'intrusion

Table of Contents

List of Figures

General Introduction

Project Background

Corporate data is critical to the proper functioning of employee activities, even during their mobile and remote work. These machines are interconnected by an extensive internal network and often to the Internet.

Before the computer was used significantly, we observed various types of espionage based, With the advent of computers, networks, and the Internet, physical information theft has evolved into computer break-in.

Therefore, data stored on servers must be adequately protected and secured.To this end, a number of security techniques are employed to ensure compliance with security best practices (such as those specified in the ISO27xxx series of standards).However, we also recognize that information systems cannot be absolutely secure and foolproof.

Committed to implementing a Security Operations Center (SOC), a department within the security cell that ensures the security of the organization on both a technical and organizational level.

Problem

With the advancement of technology, cyberattacks have become a common occurrence, and their sophistication continues to grow, leaving no one safe from their devastating effects. A successful cyberattack can result the loss of sensitive data, financial loss, and damage to our University reputation like what happened to Lincoln College in Illinois in December 2021, a ransomware attack walled off the school's access to its data and halted its recruitment, retention and fund-raising campaigns [1]. So building a SOC For our

University is a necessity but it can be costly and time consuming. In fact, you may wonder if you have enough time and qualified team members to implement and manage it, This is why it is important to find ways to simplify and automate security monitoring to reduce response time in the SOC.

Proposed Solutions

In this work, we present our Security Operations Center, which combines a log manager and a threat detection service. The combination of log manager, and threat detection forms the foundation of the SOC platform. The service triggers an alert when a potential threat is detected and forwards it to you as a notification via email or text message so you can investigate further and respond to attack early to reduce the damage, giving you a safe exit to monitor the security of your IT systems.

Document Plan

This thesis is organized as follows:

- General Introduction.
- In the first chapter, we will present the state of art of the security operations center(SOC).
- The second chapter is about choosing the right tools for building a SOC.
- The last chapter is for the implementation of the tools we choose and running some tests.
- General conclusion

Chapter 1

State of the Art of SOC(Security Operations Center

Introduction

The importance of SOC has increased in recent years due to the rising number of cyber attacks on organizations of all sizes. A well-functioning SOC can significantly improve an organization's overall security posture by detecting and responding to security incidents quickly, reducing the impact of security breaches, and preventing future incidents.

1.1 Definition of Security Operations Center (SOC)

A security operations center (SOC) – sometimes called an information security operations center, or ISOC – is an in-house or outsourced team of IT security professionals that monitors an organization's entire IT infrastructure, 24/7, to detect cybersecurity events in real time and address them as quickly and effectively as possible[2].

The Security Operations Center (SOC) plays a vital role in selecting, managing, and maintaining an organization's cybersecurity solutions. It regularly analyzes threat data to uncover weaknesses and enhance overall security. This proactive approach helps prevent cyberattacks, avoiding financial losses, reputational damage, and legal consequences.

1.2 Why using SOC ?

A Security Operations Center (SOC) is a centralized facility responsible for monitoring, detecting, and responding to security incidents in an organization's IT infrastructure. SOCs are staffed with security analysts who keep tabs on an organization's systems, networks, applications, and data to identify potential security threats and quickly respond to incidents.

The primary objective of a SOC is to safeguard the confidentiality, integrity, and availability of an organization's critical data [3] , and To accomplish this, SOCs typically utilize a range of tools and methodologies such as Security Information and Event Management (SIEM) systems, threat intelligence feeds, network and endpoint monitoring, Intrusion Detection and Prevention Systems (IDS/IPS), and incident response frameworks.

1.3 SOC architecture

A Security Operations Center (SOC) architecture typically consists of four main components: people, processes, technology, and data.

1. **People:** The SOC team consists of security analysts, threat hunters, incident responders, and other security professionals. The team should be well-trained, experienced, and able to work collaboratively to investigate and respond to security incidents.
2. **Processes:** The SOC should have clearly defined processes for incident detection, triage, investigation, and response. These processes should be documented and regularly reviewed and updated to ensure that they are effective and efficient.
3. **Technology:** The SOC should be equipped with advanced security technologies, including security information and event management (SIEM) systems, intrusion detection and prevention systems (IDPS), threat intelligence platforms, and other security tools. These technologies should be integrated and automated to enable rapid incident detection and response.
4. **Data:** The SOC relies on a wide range of data sources, including logs, network traffic, and threat intelligence feeds. This data should be collected, aggregated, and analyzed in real-time to identify potential security threats.

The SOC architecture should be designed to meet the specific needs of the organization, taking into account factors such as the size of the organization, the nature of its business, and its risk profile. The architecture should be scalable and adaptable to meet changing security threats and organizational needs.

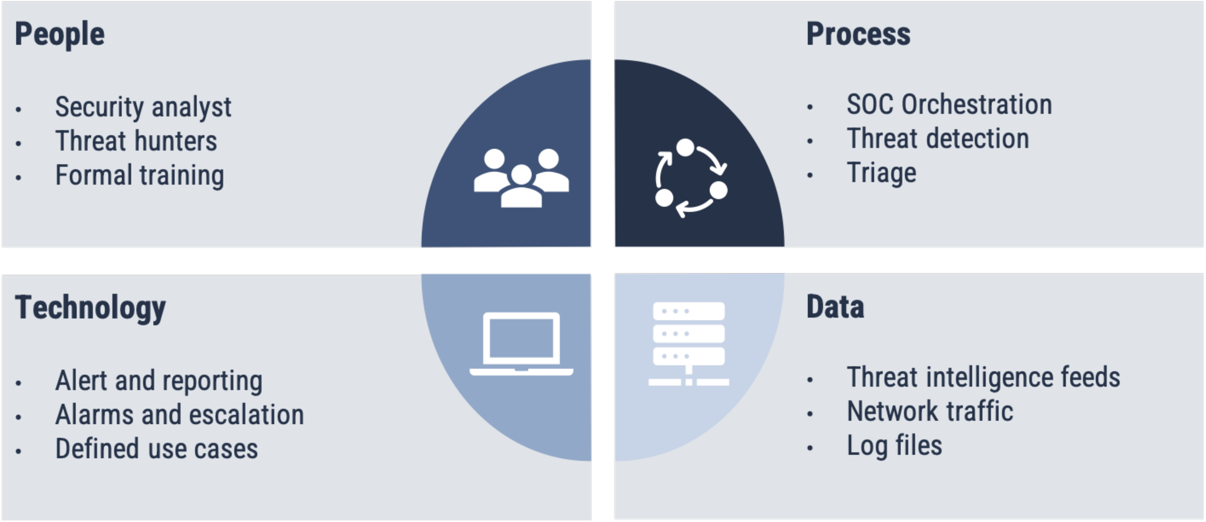


Figure 1.1: SOC Architecture

1.3.1 SOC Team Structure and Roles

The SOC team structure comprises different roles with specific responsibilities to ensure the security of the organization's assets. Here are the primary team structures and roles in a SOC:

- **Security Analysts:** They are responsible for monitoring security events and alerts generated from various sources, They investigate and analyze security incidents, identify threats and vulnerabilities, and respond to security incidents. They also provide recommendations for improving the organization's security posture.
- **Incident Responders:** they play a crucial role in coordinating and leading the response to security incidents. They collaborate with various teams and stakeholders, ensuring effective incident response and coordination. This includes engaging with external parties like law enforcement and regulatory bodies, if required.
- **Threat hunters:** responsible for proactively seeking out potential security threats before they become incidents. They use various tools and techniques to identify potential threats, including threat intelligence feeds and data analysis. They work closely with SOC analysts to investigate and respond to potential threats.
- **Security Engineers:** SOC engineers are responsible for implementing and maintaining the SOC's technologies, including security information and event management (SIEM) systems, intrusion detection systems (IDS), firewalls, and other security technologies. They also ensure that the SOC's infrastructure is up-to-date and secure.
- **SOC Manager:** The SOC manager is responsible for overseeing the entire SOC operation. They provide direction, guidance, and supervision to the SOC team, including managing the SOC's budget and resources and ensures that the SOC op-

erates according to the organization's security policies, procedures, and compliance requirements [4].

1.3.2 SOC Levels

SOCs can be categorized into various tiers according to their capabilities, maturity, and range of services. Presented below are the three widely acknowledged tiers of SOC classification:

1. **Tier 1 (Triage):** Tier 1 is the entry-level SOC providing essential security monitoring and incident management services. Analysts monitor security events and alerts, taking appropriate actions to address potential threats. They also conduct initial triage and investigations to assess the severity and impact of security incidents.
2. **Tier 2 (Investigation):** Tier 2 Positioned as an intermediate level, this SOC focuses on complex security incidents requiring in-depth analysis. Analysts possess advanced skills and broader access to diverse security tools and data sources. They excel in correlating events across systems, identifying patterns of suspicious behavior indicating significant threats.
3. **Tier 3 (Threat hunting):** It is the highest level SOC offers advanced threat hunting and response capabilities. Analysts are highly skilled, equipped with the latest threat intelligence and security technologies. Their primary role is to proactively search for advanced threats that may have bypassed other security controls and respond swiftly and efficiently.

It's worth noting that some organizations may have additional tiers or different naming conventions for their SOC levels. However, the above three tiers are the most commonly used and recognized in the industry.

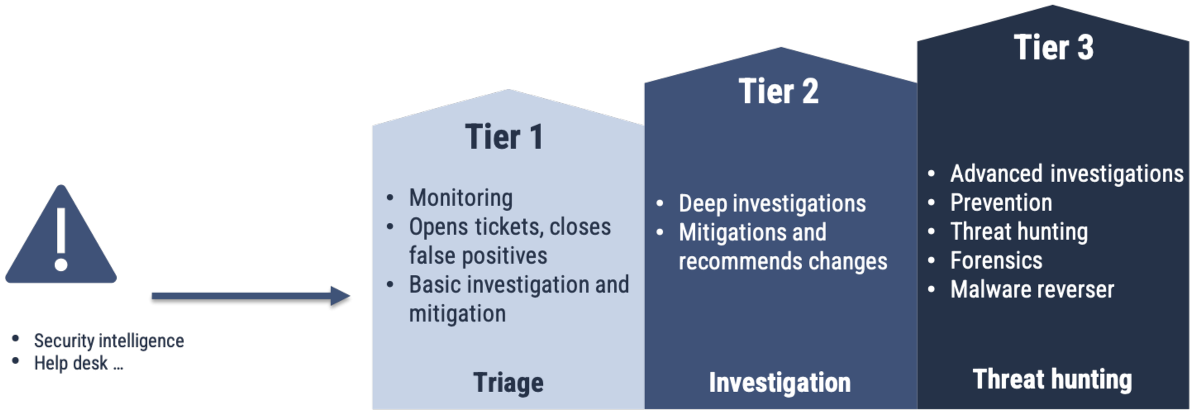


Figure 1.2: SOC Levels

1.4 SOC Activities and Functions

1.4.1 Main SOC Activities

1. **Data Collection :**
   Data collection involves gathering device and network activities, including log files and threat intelligence. It follows the company's audit policy and detection policy, overseen by endpoint engineers. The goal is to create a comprehensive view of the environment for security threat insights.
2. **Detection :**
   After the data collection function records the relevant items, the detection function applies analytics to determine if any of the data matches known bad domain lists, hashes, or IP addresses associated with malicious activity. The ultimate goal is to identify potentially harmful content with high accuracy.
3. **Triage :**
   Triage involves identifying critical alerts and managing the alert queue. SOC analysts review alerts gathered in a central location and prioritize those requiring immediate attention. Human expertise and technology work together to facilitate fast and accurate decision-making. The objective is to use technology to aid analysts in promptly addressing the most crucial alerts.
4. **Investigation :**
   After triage, the SOC team verifies if the selected high-priority alert is a genuine threat or a false positive. SOC analysts review the alert to confirm if it is an attack. Accurate and speedy verification is crucial, but can be challenging if the SIeM system is not optimized for effective data processing. The outcome of this step is either a confirmed threat, which requires incident response, or a false positive that can be disregarded..

5. **Incident response :**
   The incident response process involves addressing validated alerts to resolve confirmed attacks or incidents. Generating threat intelligence aids in identifying attack types and enhancing detection capabilities. The SOC's production of threat intelligence is a key outcome[5].

1.4.2 Auxiliary SOC Functions

1. **Threat Intelligence :**
   Threat intelligence is the identification and analysis of cyberthreats. It involves collecting, processing, and analyzing data to gain insights into threats. There are three levels of threat intelligence, representing the level of detail and specificity in the gathered information about potential or actual threats.
   - **Strategic Intelligence :** focuses on broad trends and potential threats, gathered from open sources. It informs long-term planning and decision-making by identifying emerging cyberthreats, geopolitical events, and economic trends.
   - **Operational Intelligence :** targets specific threats and tactics used by threat actors, gathered from various sources including logs and social media. It guides daily security operations and response activities.
   - **Tactical Intelligence :** it provides detailed insights into specific threats or attacks, obtained from sources like malware and forensic analysis. It offers immediate guidance during incident response efforts, aiding security teams.
2. **Forensics :**
   It provides the ability to investigate security incidents and identify the root cause of security breaches. Forensic analysis can help SOC teams determine the extent of the damage caused by a security incident, identify the data or assets that have been compromised, and provide evidence that can be used in legal proceedings.
3. **Self-assessment :**
   Self-assessment is an important aspect of maintaining a high level of performance and effectiveness in a security operations center (SOC). Here are some key areas to focus on when conducting a self-assessment in a SOC:
   - **People:** Evaluate the skills and knowledge of the SOC team members.
   - **Processes:** Review the incident response processes, procedures, and workflows in place in the SOC
   - **Tools and technologies:** Assess the effectiveness of the tools and technologies used in the SOC, such as SIEM systems, threat intelligence feeds, and EDR tools

- **Training and education:** Evaluate the training and education programs in place for SOC team members[5].

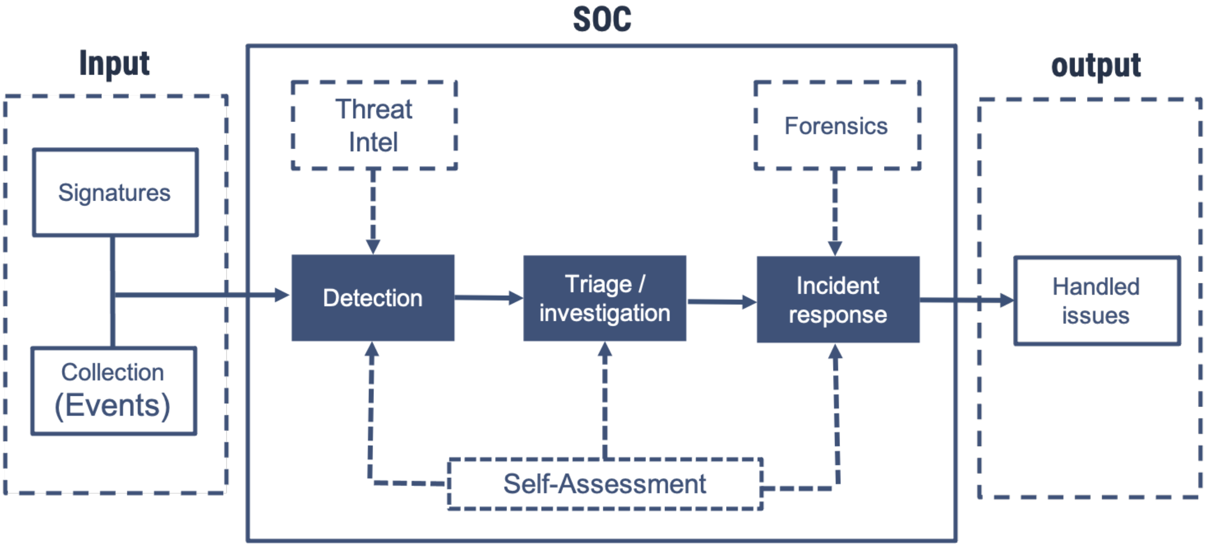


Figure 1.3: SOC Functions

1.5 SOC technologies

Security Operations Centers (SOCs) rely on various technologies to detect, monitor, and respond to security incidents. Here are some common SOC technologies:

1. **Security Information and Event Management (SIEM):** SIEM systems collect and analyze data from various sources, including network devices, servers, and applications, to identify potential security incidents. SIEM systems correlate events from different sources and generate alerts for SOC analysts to investigate.

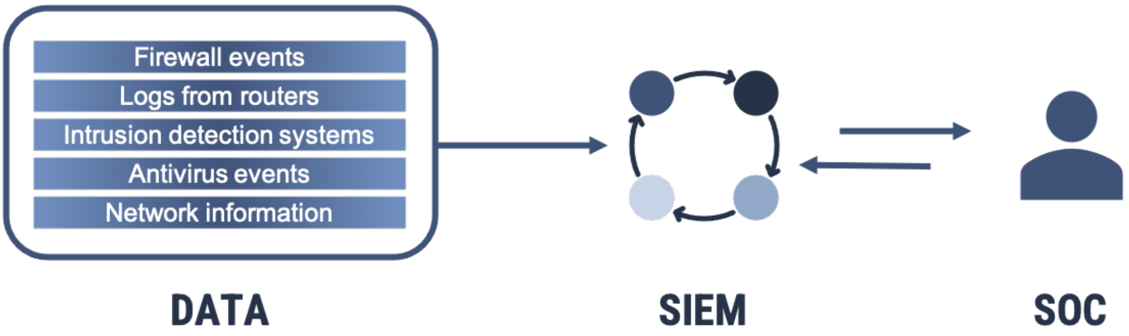


Figure 1.4: SIEM diagram

2. **Intrusion Detection Systems (IDS):** IDS are network security tools that monitor network traffic for potential security breaches. IDS can detect suspicious activity such as port scanning, network intrusion attempts, and malware infection.
3. **Endpoint Detection and Response (EDR):** EDR tools monitor endpoint devices such as laptops, desktops, and servers for suspicious activity. EDR tools can detect and respond to malware infections, unauthorized access, and other security incidents.
4. **Network Traffic Analysis (NTA):** NTA tools analyze network traffic to detect and respond to security incidents. NTA tools can detect suspicious activity such as data ex-filtration, lateral movement, and malware command and control traffic.
5. **Security Orchestration, Automation, and Response (SOAR):** SOAR platforms integrate various SOC technologies to automate and orchestrate security operations. SOAR platforms can streamline incident response, automate repetitive tasks, and improve the efficiency of SOC operations.
6. **Data Loss Prevention (DLP):** DLP tools monitor network traffic and endpoints to prevent sensitive data from leaving the organization. DLP tools can detect and block attempts to ex-filtrate sensitive data through email, web uploads, or other channels.

These are just a few examples of SOC technologies. Different organizations may use different combinations of technologies based on their specific needs and budgets.

1.6 Different types of security operations center

security operations center can be classified based on various criteria, such as their specialization, tiered levels, localization, and organizational techniques. The main types of SOCs include :

- **Centralized SOC:** A SOC that is located in a central location and serves as the primary point of contact for all security-related activities across the organization.
- **Virtual SOC:** A SOC that is entirely cloud-based, allowing for remote monitoring and management of security incidents. This type of SOC may be centralized or decentralized, depending on the organizational structure.
- **Co-managed SOC:** A SOC that is managed jointly by the organization and a third-party provider, with the organization retaining some control over security operations.
- **Specialized SOC:** SOC that focuses on a specific area of cybersecurity, such as cloud security, network security, industrial control systems security, or incident response.

Each type of SOC has its unique strengths and weaknesses, and organizations need to consider their specific security needs and resources when selecting the appropriate SOC model.

1.7 SOC performance metrics

The Security Operations Center (SOC) performance metrics can be divided into several categories:

1. **Incident Response Metrics:** These metrics measure how well the SOC handles and resolves security incidents. Some examples of incident response metrics include:
    - **Mean Time to Detect (MTTD):** Measures the time taken to detect a security incident.
    - **Mean Time to Respond (MTTR):** is the average time it takes DevOps teams to respond after receiving an alert [6].
    - **First Contact Resolution (FCR):** Measures the percentage of incidents resolved on first contact with the SOC.
2. **Threat Intelligence Metrics:** These metrics measure how well the SOC leverages threat intelligence to detect and respond to security threats. Some examples of threat intelligence metrics include:
    - **Detection Ratio:** Measures the number of threats detected by the SOC compared to the total number of threats identified.
    - **False Positive Ratio:** Measures the number of false positives generated by the SOC compared to the total number of alerts generated.
    - **Threat Hunting Success Rate:** Measures the effectiveness of proactive threat hunting activities conducted by the SOC.
3. **SOC Operations Metrics:** These metrics measure the overall efficiency and effectiveness of SOC operations. Some examples of SOC operations metrics include:
    - **Security Posture Improvement:** Measures the effectiveness of security controls and processes implemented by the SOC in improving the overall security posture.
    - **Training and Development:** Measures the effectiveness of training and development programs for SOC staff in improving their skills and knowledge.
    - **Budget Utilization:** Measures the effectiveness of SOC budget utilization in achieving the desired security outcomes.

Conclusion

In conclusion, Security Operations Centers (SOCs) are critical components of modern cybersecurity strategies for organizations of all sizes and types. With the ever-increasing sophistication and frequency of cyberattacks, a SOC can help an organization proactively monitor, detect, and respond to security incidents in real-time. By deploying the appropriate SOC model that matches the organization's specific needs, resources, and risk appetite.

Ultimately, an effective SOC can help organizations stay ahead of evolving cyberthreats, maintain business continuity, and protect their reputation, assets, and customers.

Chapter 2

Tools and technologies used in SOC

Introduction

Security Operations Centers (SOCs) play a critical role in protecting an organization's digital assets, and having the right tools and technologies is essential to their success.

Additionally, automation has become a critical component of SOC operations, enabling faster and more efficient incident response. This chapter will cover how these tools can be automated using technologies such as Security Orchestration, Automation, and Response (SOAR) platforms.

2.1 Security Operations Center used tools (open source)

SOCs rely on a variety of tools and technologies to monitor network traffic, detect anomalies, and investigate potential threats.

Some common tools and technologies used in SOCs include:

2.1.1 ELK Stack

The ELK stack is a popular open-source platform used for log management and analysis. It consists of three main components: Elasticsearch, Logstash, and Kibana.

1. **Elasticsearch :** is a distributed RESTful search and analysis engine that allows us to store large amounts of information in JSON format in a powerful indexing engine. Indeed, Elasticsearch can be seen as a search engine that uses Lucence (Apache library) to index the content.
2. **Logstash :**is a data collector and data processor based on a set of different plug-ins. These plug-ins allow you to easily configure the tool to collect, load and transfer data in a number of different architectures and send it to Elasticsearch.

3. **Kibana :** is a web-based visualization tool used to analyze and visualize data stored in Elasticsearch. It allows users to create interactive visualizations, dashboards, and reports, making it easy to understand and share insights with others [7] .
4. **Beats :** Beats are lightweight data shippers that are used to collect, parse, and send various types of data to Elasticsearch or Logstash for further processing and analysis.

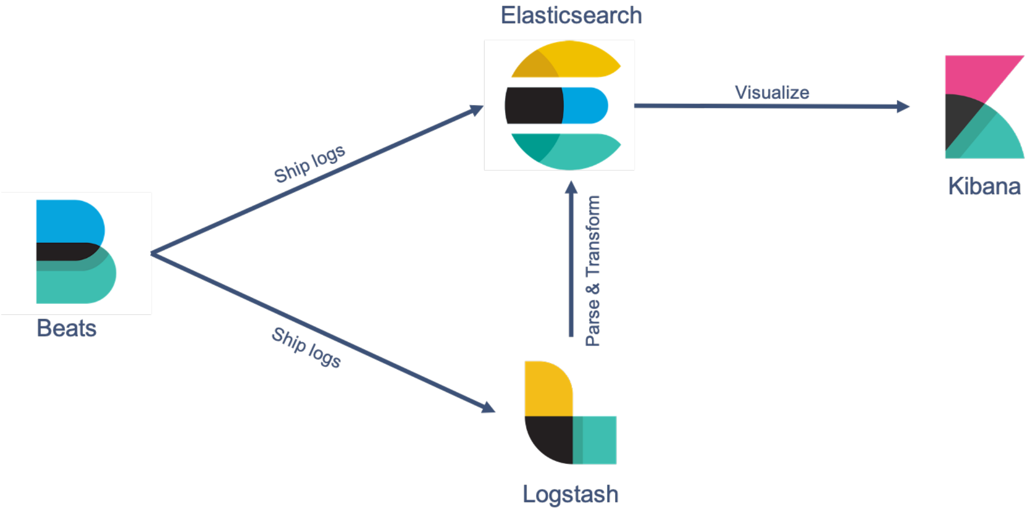


Figure 2.1: ELK-Stack

Together, the ELK stack provides organizations with a powerful platform for analyzing log data, identifying trends, and investigating potential security incidents. It is widely used in Security Operations Centers (SOCs) to detect and respond to security threats.

2.1.2 TheHive

TheHive is an open-source Security Incident Response Platform that provides a scalable and collaborative environment for managing and responding to security incidents. It is designed to simplify and streamline the incident response process for SOC, CSIRT, CERT, and other security practitioners [8].

Advantages of using TheHive

TheHive offers several advantages compared to other Incident Response (IR) platforms. Here are some key benefits:

- **Scalable and Collaborative:** Enables collaboration and scales to fit the needs of organizations.
- **Seamless Integration:** Integrates with various security tools and data sources for comprehensive investigations.

- **User-Friendly Interface:** Provides an intuitive interface for easy navigation and efficient operations.
- **Customization and Extensibility:** Allows customization of workflows and supports integration of additional analyzers and responders.

2.1.3 Cortex

Cortex is an open-source analysis and triage engine that complements TheHive and automates the investigation and analysis of security observables. It integrates with threat intelligence feeds and security tools, providing a comprehensive view of security incidents. With its flexible architecture, Cortex allows users to create custom analyzers and responders. It streamlines incident response by automating the analysis process.

Advantages of using Cortex

Cortex has many advantages compared to other analyzing tools, we mention :

- **Automation:** automates the analysis and triage of security observables, saving time and effort for security teams.
- **Integration:** integrates with various threat intelligence feeds, malware analysis tools, and other security tools, providing a comprehensive and interconnected security ecosystem.
- **Customization:** allows customization through a flexible and modular architecture, empowering users to tailor analyzers and responders to their specific needs.
- **Efficiency:** Cortex improves the efficiency of incident response, enabling faster and more effective detection, investigation, and response to security incidents.

2.1.4 MISP

MISP - Open Source Threat Intelligence and Sharing Platform allows organizations to share information such as threat intelligence, indicators, threat actor information or any kind of threat which can structured in MISP. The aim of this trusted platform is to help improving the counter-measures used against targeted attacks and set-up preventive actions and detection [9].

Advantages of using MISP

When compared to other threat intelligence platforms, MISP offers several advantages:

- **Open Source:** MISP is freely available and can be customized according to specific needs.

- **Flexibility and Customization:** MISP allows users to define their own data models, taxonomies, and information sharing formats.
- **Collaborative Sharing:** MISP facilitates secure sharing and exchange of threat intelligence data among different organizations and communities.
- **Integration Capabilities:** MISP seamlessly integrates with a wide range of security tools, enabling consolidated and correlated threat intelligence data from multiple sources.

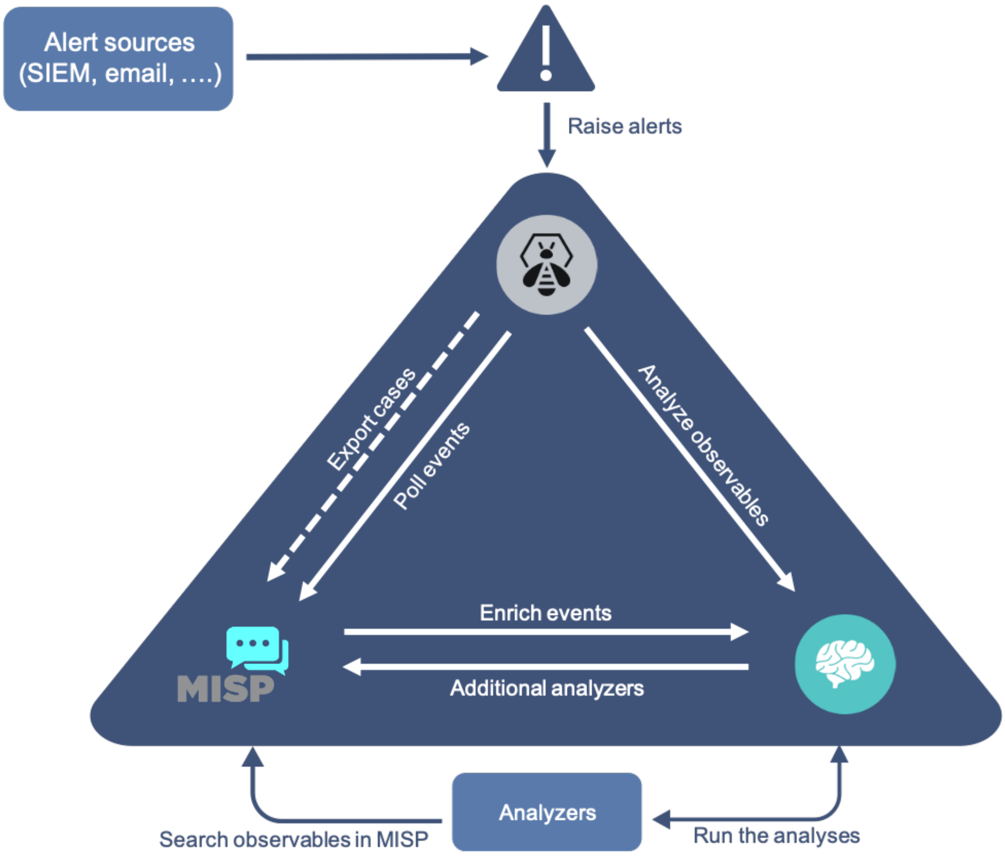


Figure 2.2: TheHive,Cortex and MISP integration

2.2 Automation solution

2.2.1 SOAR Technology

Security orchestration, automation and response, or SOAR, is a stack of compatible software programs that enables an organization to collect data about security threats and respond to security events with little or no human assistance. The goal of using a SOAR platform is to improve the efficiency of physical and digital security operations.

SOAR platforms have three main components:

1. **Security orchestration :** it connects and integrates disparate internal and external tools via built-in or custom integrations and application programming interfaces.
2. **Security automation :** fed by the data and alerts collected from security orchestration, ingests and analyzes data and creates repeated, automated processes to replace manual processes.
3. **Security response :** offers a single view for analysts into the planning, managing, monitoring and reporting of actions carried out after a threat is detected. This single view enables collaboration and threat intelligence sharing across security, network and systems teams[10].

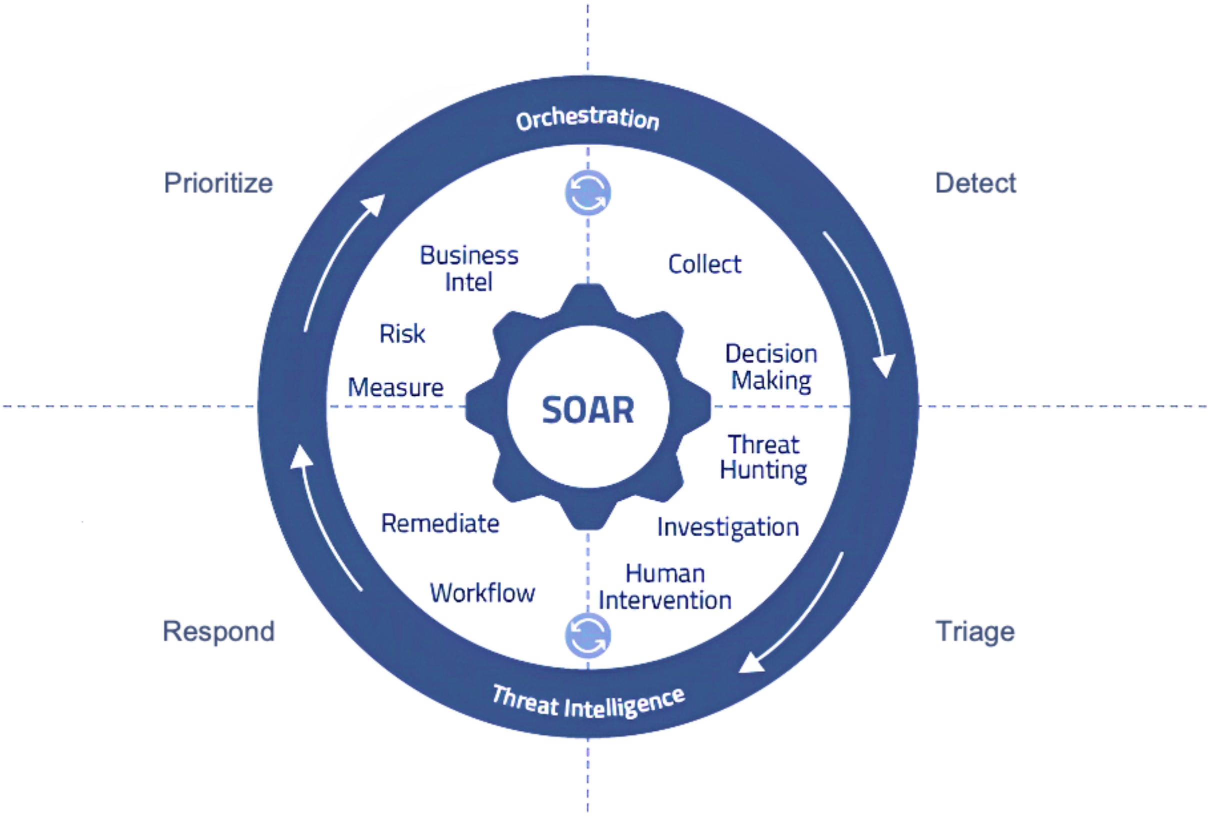


Figure 2.3: SOAR model

2.2.2 Benefits of SOAR

SOAR platforms can provide several benefits for Security Operations Center (SOC) teams. These benefits include faster incident response, improved efficiency, enhanced collaboration, better threat hunting, and increased visibility. SOAR can automate routine security tasks, allowing SOC team members to focus on high-level tasks such as incident response, threat hunting, and vulnerability management. and also provide a centralized platform that allows SOC teams to collaborate more effectively, share information and insights across the team, and make more effective decisions.

By automating threat intelligence gathering and analysis, SOAR can help SOC teams to proactively identify and mitigate potential security threats, leading to a more proactive security posture and reduced risk of security incidents.

2.2.3 Open Source SOAR solution

Shuffle

Shuffle is a SOAR platform that allows security teams to orchestrate and automate end-to-end security incident handling. Its operation is mainly based on applications or integrations, triggers and workflows. Shuffle is oriented towards a fully open-source ecosystem. This includes, but is not limited to, open-source workflows, applications and standards (OpenAPI, Swagger)[11].

2.3 Proof of Concept (POC)

We have devised a comprehensive proof of concept (POC) for our Security Operations Center (SOC), focusing on essential use cases and key phases to bolster threat detection and incident response capabilities, fortify security incident management, and mitigate the impact of security breaches.

To kickstart the process, we have diligently established the requisite infrastructure and deployed cutting-edge tools to facilitate SOC operations. The SOC is now actively monitoring the network and collecting logs from diverse endpoints, meticulously scrutinizing them for any anomalies that could potentially raise red flags. Once an anomaly is detected, it is promptly transformed into an alert ticket, triggering the immediate attention of a skilled SOC analyst.

Our SOC boasts a sophisticated array of correlation rules, expertly developed and fine-tuned to pinpoint various types of attacks that may target our university. Notably, we have equipped our SOC to detect network attacks, malware incidents, denial-of-service (DOS) attacks, and more, reinforcing our proactive security posture.

Upon detection, the incident response phase springs into action, encompassing vital activities such as triaging, categorizing, and promptly responding to security incidents. The subsequent phase revolves around Threat Intelligence, wherein we diligently gather pertinent intelligence from both internal and external sources, empowering our SOC to stay ahead of emerging threats.

Moreover, we have judiciously automated certain SOC functionalities, enabling streamlined and efficient operations. Lastly, our SOC is committed to continuous improvement, perpetually evaluating and enhancing its operations. We conduct regular exercises and diligently test the effectiveness of our detection and response capabilities, ensuring we remain resilient in the face of evolving security challenges.

Through this holistic and well-defined POC, we aim to demonstrate the tangible benefits of a robust SOC in safeguarding our organization's critical assets and maintaining a robust security posture.

Conclusion

By leveraging powerful open-source tools and embracing automation, our SOC is equipped to effectively protect our assets. This combination reduces manual work for analysts, enabling faster response times and ensuring a successful defense against attacks.

Chapter 3

Contributions

Introduction

In this chapter, we discuss the contribution of our research to the field of SOC operations, which provides valuable insights for improving the effectiveness and efficiency of SOC operations

Additionally, we discuss SOC implementation and introduce various development tools that provide organizations with a road-map to improve their SOC operations and better protect their assets.

3.1 Theoretical Proposal

3.1.1 Project description

The objective of this project is to design and implement a Security Operation Center (SOC) using open source tools, namely TheHive, Cortex, and MISP, with the integration of Shuffle as a SOAR solution to automate and orchestrate incident response processes. The SOC will serve as a central hub for monitoring,detect, analyzing, and responding to security incidents within the organization's information systems and networks.

This project aims to establish a cost-effective and efficient SOC that enables proactive monitoring, rapid incident response, and effective utilization of threat intelligence within the organization.

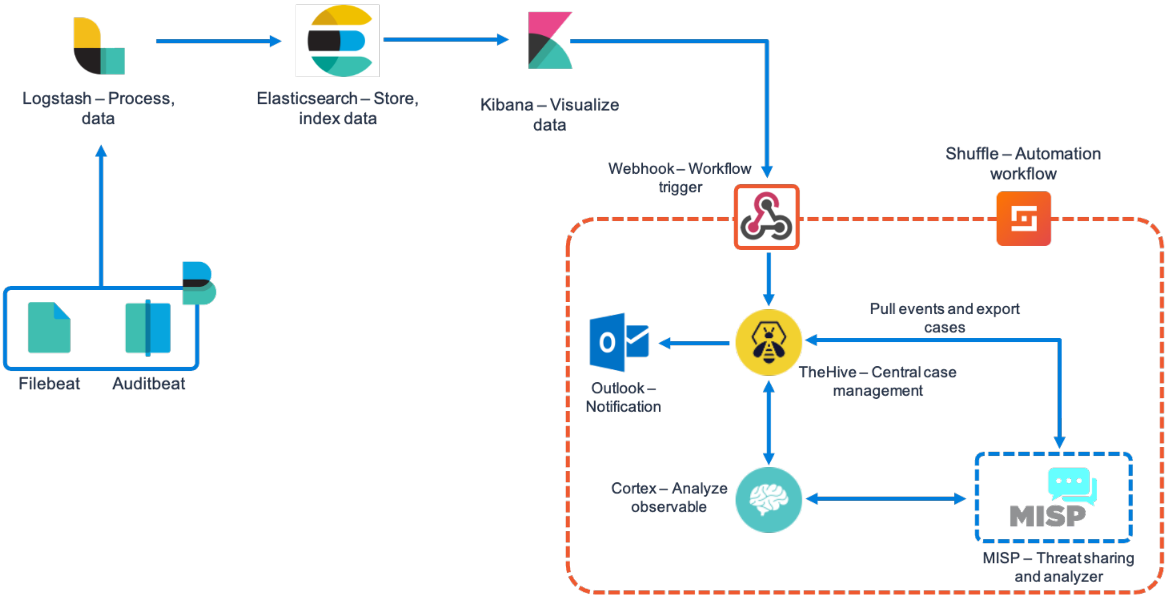

Logstash – Process, data
Elasticsearch – Store, index data
Kibana – Visualize data
Webhook – Workflow trigger
Shuffle – Automation workflow
Pull events and export cases
Outlook – Notification
TheHive – Central case management
Cortex – Analyze observable
MISP
MISP – Threat sharing and analyzer
Filebeat
Auditbeat

Figure 3.1: SOC Architecture

3.1.2 Work Environment

The good work environment for our Security Operation Center project involves the separation of the tools by installing each of them in a different server or machine, which offers improved performance, security, flexibility, fault tolerance, and ease of maintenance, contributing to the overall effectiveness of the SOC environment. and for each machine we have used a set of hardware with the following main characteristics:

- Server with 8GO RAM and 100GO of space
- Exploitation System : Linux Debian 11 /64bit

3.2 Operation system hardening (Debian 11)

We have made the decision to install Debian version 11, a Linux distribution, to serve as the foundation for our secure system. Recognizing the importance of protecting our system against cyber threats, we have dedicated our time and effort to hardening the OS. By implementing various security measures and configurations, we aim to boost the security posture and resilience of our system. To assess the security status, we have utilized the open-source project Lynis. It is a script that will run various of tests, than will provide a security percentage score after running it. We will guide you through the process of installing Lynis and the steps we took to harden Debian, ensuring a robust and secure operating system.

Installing Lynis

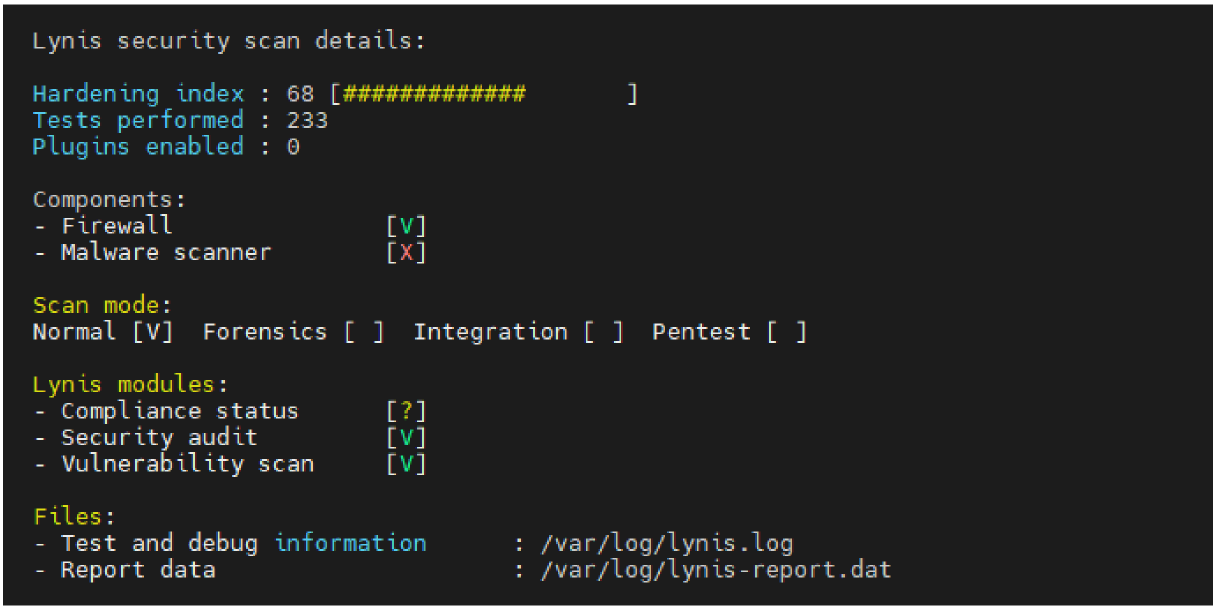
Download the Lynis package and Extract it by executing the following commands:

```
wget https://cisofy.com/files/lynis-3.0.8.tar.gz
tar xvfz lynis-3.0.4.tar.gz
```

Navigate into the lynis directory and perform a system audit with Lynis, by running the following command:

```
cd lynis
./lynis audit system
```

Lynis now initiate the security audit of our system and displayed the results in the terminal as the screenshot shows :



```
Lynis security scan details:

Hardening index : 68 [##############       ]
Tests performed : 233
Plugins enabled : 0

Components:
- Firewall               [V]
- Malware scanner        [X]

Scan mode:
Normal [V]  Forensics [ ]  Integration [ ]  Pentest [ ]

Lynis modules:
- Compliance status      [?]
- Security audit         [V]
- Vulnerability scan     [V]

Files:
- Test and debug information      : /var/log/lynis.log
- Report data                     : /var/log/lynis-report.dat
```

Figure 3.2: The security percentage of the OS

You can see that the rate of OS is 68 based on 239 tests performed. So now it's our job to harden the OS and make it more secure.

Password protection

When we perform a system audit on our OS , it reveals this weakness: "Checking for password protection [NONE]". Which means that our OS does not have a password protection enabled for a specific component. To solve it you have to follow this steps:

1. open the file "/etc/default/grub" and find the line that starts with : "GRUB CMDLINE LINUX DEFAULT"

   ```
   nano /etc/default /grub
   ```

2. Put this parameters in the first "splash grub.pbkdf2.sha512=1" and your file will look like this: "GRUB CMDLINE LINUX DEFAULT=quiet splash grub.pbkdf2.sha512=1"
3. Set a GRUB2 password using this command :

   ```
   grub-mkpasswd-pbkdf2
   ```

   and copy the generated password hash
4. Update the GRUB configuration file

   ```
   nano /etc/grub.d/40_custom
   ```

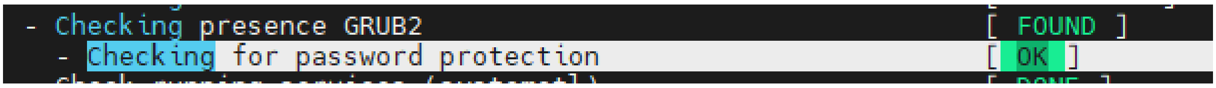
   Add the this lines at the end of the file:

   ```
   set superusers=username
   password_pbkdf2 username generated_password_hash
   ```

   The username must be replayed with your username
5. Then save the file and update GRUB2

   ```
   update-grub
   ```

And after you complete all this steps it will show:



```
- Checking presence GRUB2                                   [ FOUND ]
  - Checking for password protection                        [ OK ]
```

Figure 3.3: Status after hardening the passwords

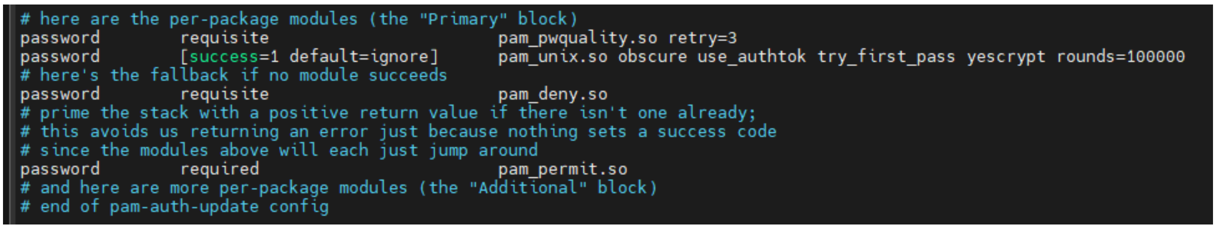
NOTE: all this steps must be done using the root privileges.
and the percentage of the OS security will be up to 69

Secure Authentication by configuring pam

1. **Configure a hashing method for the Password :** You just need to update the common-password file which you can do it using a text editor: You can just use this command:

   ```
   nano /etc/pam.d/common-password
   ```

   and then look for the line where you find "pam.unix.so" and update it like in the screenshot



```
# here are the per-package modules (the "Primary" block)
password        requisite                       pam_pwquality.so retry=3
password        [success=1 default=ignore]      pam_unix.so obscure use_authtok try_first_pass yescrypt rounds=100000
# here's the fallback if no module succeeds
password        requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password        required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
# end of pam-auth-update config
```

Figure 3.4: Configuration file of common-password

2. **Configure password hashing rounds :** It's the same file that must be configured just add this parameter: "rounds= (the number of rounds needed)"
3. **Configure password aging (minimum, maximum) :** Now we have to update the "login.defs" file to do it use this command

   ```
   nano /etc/login.defs
   ```

   and configure this parameters according to your need
   PASS_MAX_DAYS 100
   PASS_MIN_DAYS 7
4. **Determining default umask for files :** Open the file that Lynis tool detected that must be update using a text editor and put in the end of the file: "umask 027"

Set the permission of recommended files

After you run the script, you will see in the result some files that you have to harden the permissions of them. we used "chown" command to set the file owner and group ownership to "root" And the "chmod" command to set correct permission (read and execute).

Mount directories recommended

Here after got the result of the lynis script. There is a number of directories you will be recommended to mount like "/home /tmp /var"
So let us show an example of those directories the way we mount it (/tmp):

1. Create a new directory Then we mounted the "/mnt/tmp mount"

   ```
   mkdir /mnt/tmp_mount
   mount /dev/sda  /mnt/tmp_mount
   ```

2. Copy the content of "/tmp" to "/mnt/tmp mount"

   ```
   cp --a /tmp/. /mnt/tmp_mount/.
   ```

3. Now we have to add the new entry to the "/etc/fstab", add this line to the file "/dev/sdX1 /tmp ext4 defaults 0 2"

4. Finally mount the "/tmp" with the updated "/etc/fstab"

```
mount --a
```

Configure the networking

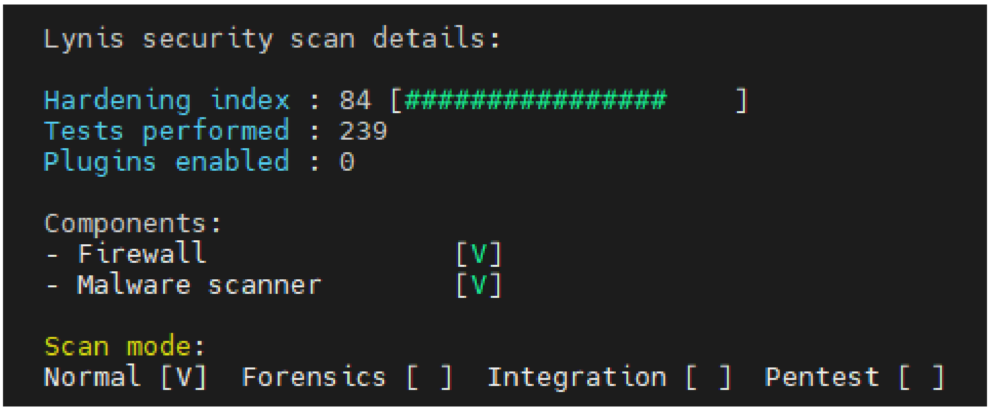
First we have to make sure that our OS at least have 2 working DNS
To do that you just have to configure "/etc/resolv.conf" file

1. Open the file using :

```
nano /etc/resolv.conf
```

2. Append those servers like a backup to the DNS server you will find already:
   nameserver 8.8.8.8
   nameserver 8.8.4.4
   After this you can close any open port that you don't need.

There are other things we have done to reduce the risk like installing malware scanner, IDS/IPS, accounting software, firewall, auto upgrade software and some other things. . . . we can't write all the steps that we have did in order to harden the Debian OS that we have used to deploy our SOC in it, because it will be a too long.
But here is the percentage of our harden Debian 11 OS for now



```
Lynis security scan details:

Hardening index : 84 [################    ]
Tests performed : 239
Plugins enabled : 0

Components:
- Firewall               [V]
- Malware scanner        [V]

Scan mode:
Normal [V]  Forensics [ ]  Integration [ ]  Pentest [ ]
```

Figure 3.5: Status of the OS after hardening

3.3 Solutions implementation

When implementing the tools we faced several difficulties, as ensuring compatibility between the software versions and dependencies required by each application. the configuration issues, such as setting up the necessary network connectivity, configuring

database connections, and managing user permissions.
Additionally, troubleshooting and resolving potential conflicts between the applications or resolving issues with the operating system's package manager posed further difficulties. Overall, we give our careful attention to documentations, thorough testing, and seeking assistance from the respective communities so we overcome these challenges during the implementation process.

3.3.1 Tools Installation

Install Required Packages

we run system package cache update and install required packages:

```
sudo apt update
sudo apt install wget gnupg2 apt-transport-https git ca-certificates curl
   jq software-properties-common lsb-release python3-pip iproute2
```

Install Java Runtime Environment

then we have installed Java and define the JAVA_HOME environment variable :

```
sudo apt install openjdk-11-jre-headless
```

Set the JAVA_HOME :

```
echo JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64" | sudo tee -a /etc/
   environment
source /etc/environment
```

3.3.1.1 ELK stack installation

Installation of Elastic Stack follows a specific order. we followed the order bellow to install Elastic Stack components :

1. **Install Elasticsearch**

   To simplify the installation of all Elastic Stack components, we will create Elastic Stack repos,and import the Elastic stack PGP repository signing Key :

   ```
   curl -sL https://artifacts.elastic.co/GPG-KEY-elasticsearch | gpg
       --dearmor > /etc/apt/trusted.gpg.d/elastic.gpg
   echo "deb https://artifacts.elastic.co/packages/7.x/apt stable
      main" | tee /etc/apt/sources.list.d/elastic-7.x.list
   ```

   Update package cache and install Elasticsearch :

   ```
   apt update
   apt install elasticsearch
   ```

Configure Elasticsearch

we changed the default cluster name :

```
sed -i '/cluster.name:/s/#//;s/my-application/PFE-RSD/' /etc/elasticsearch/elasticsearch.yml
```

we defined an address on which to expose Elasticsearch node on the network "192.168.56.101". By default Elasticsearch is only accessible on localhost :

```
sed -i '/network.host:/s/#//;s/192.168.0.1/192.168.56.101/' /etc/elasticsearch/elasticsearch.yml
```

we kept the default Elasticsearch port Elasticsearch listens for HTTP traffic on the first free port it finds starting at 9200 :

```
sed -i '/http.port:/s/#//' /etc/elasticsearch/elasticsearch.yml
```

When you set the network.host to an IP address, Elasticsearch expects to be in a cluster. But since we are running a single node Elasticsearch in our setup, you need to specify the same in the configuration by adding the line, "discovery.type: single-node" on Elasticsearch configuration file and also adding the line "xpack.security.enabled: true" to turns on some of the security features that are included with Elasticsearch. and we disable Swapping:

```
echo 'discovery.type: single-node' >> /etc/elasticsearch/elasticsearch.yml
echo 'xpack.security.enabled: true' >> /etc/elasticsearch/elasticsearch.yml
sed -i '/bootstrap.memory_lock:/s/^#//' /etc/elasticsearch/elasticsearch.yml
```

Running Elasticsearch

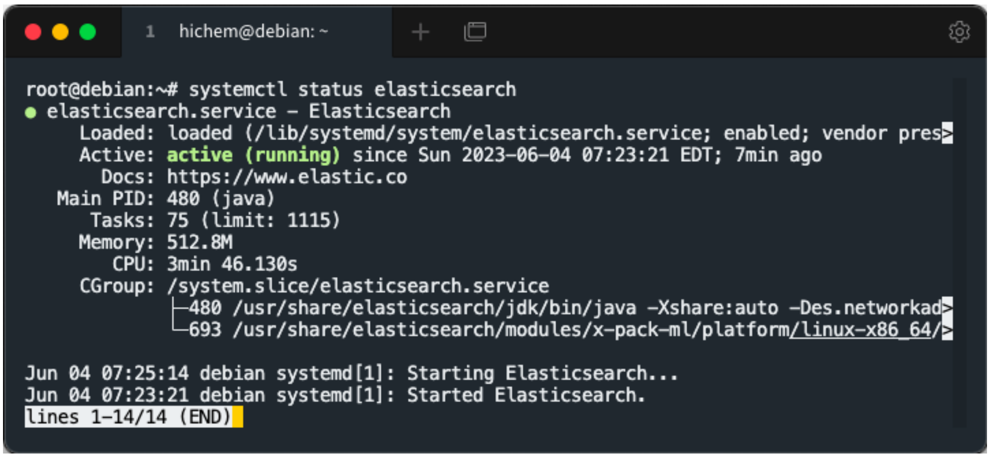
and then we Started and enabled Elasticsearch to run on system boot :

```
systemctl enable --now elasticsearch
```

To check the status

```
systemctl status elasticsearch
```

and then we get this output, which means is all well:



```
root@debian:~# systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor pres>
     Active: active (running) since Sun 2023-06-04 07:23:21 EDT; 7min ago
       Docs: https://www.elastic.co
   Main PID: 480 (java)
      Tasks: 75 (limit: 1115)
     Memory: 512.8M
        CPU: 3min 46.130s
     CGroup: /system.slice/elasticsearch.service
             ├─480 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkad>
             └─693 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/>

Jun 04 07:25:14 debian systemd[1]: Starting Elasticsearch...
Jun 04 07:23:21 debian systemd[1]: Started Elasticsearch.
lines 1-14/14 (END)
```

Figure 3.6: Elasticsearch status

- **Configuring Elasticsearch Passwords**

  Now that we have enabled the "xpack.security.enabled" setting, we need to generate passwords for the default Elasticsearch users. Elasticsearch includes a utility in the "/usr/share/elasticsearch/bin" directory that can automatically generate random passwords for these users Run the following command to cd to the directory and then generate random passwords for all the default users:

  ```
  cd /usr/share/elasticsearch/bin
  ./elasticsearch-setup-passwords auto
  ```

2. **Install Kibana**

   Since we already setup Elastic repos, simply install Kibana by running the command:

   ```
   apt install kibana
   ```

- **Configuring Kibana**

  Kibana is set to run on "localhost:5601" by default.

  To allow external access, we edited the configuration file and replace the value of server.host with the interface IP "192.168.56.101" :

  ```
  sed -i '/server.port:/s/^#//' /etc/kibana/kibana.yml
  sed -i '/server.host:/s/^#//;s/localhost/192.168.56.101/' /etc/
      kibana/kibana.yml
  ```

  In this setup, Elasticsearch is listening on "192.168.56.101:9200" . So we replaced the address accordingly.

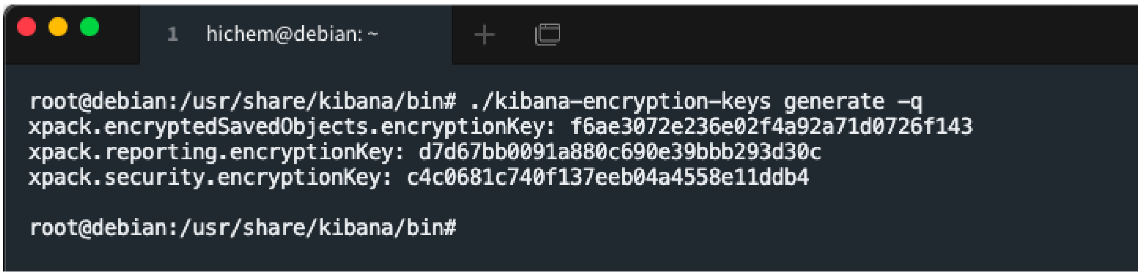
  ```
  sed -i '/elasticsearch.hosts:/s/^#//;s/localhost/192.168.56.101/'
       /etc/kibana/kibana.yml
  ```

- **Enabling "xpack.security" in Kibana**

We generated the required encryption keys using the kibana-encryption-keys utility that is included in the "/usr/share/kibana/bin" directory.by running the following to cd to the directory and then generate the keys:

```
cd /usr/share/kibana/bin/
./kibana-encryption-keys generate -q
```

The "-q" flag suppresses the tool's instructions so that you only receive output like the following:



```
root@debian:/usr/share/kibana/bin# ./kibana-encryption-keys generate -q
xpack.encryptedSavedObjects.encryptionKey: f6ae3072e236e02f4a92a71d0726f143
xpack.reporting.encryptionKey: d7d67bb0091a880c690e39bbb293d30c
xpack.security.encryptionKey: c4c0681c740f137eeb04a4558e11ddb4

root@debian:/usr/share/kibana/bin#
```

Figure 3.7: Kibana generated keys

and then we added these keys to Kibana's "/etc/kibana/kibana.yml" configuration file by copy and paste them in the end of the file

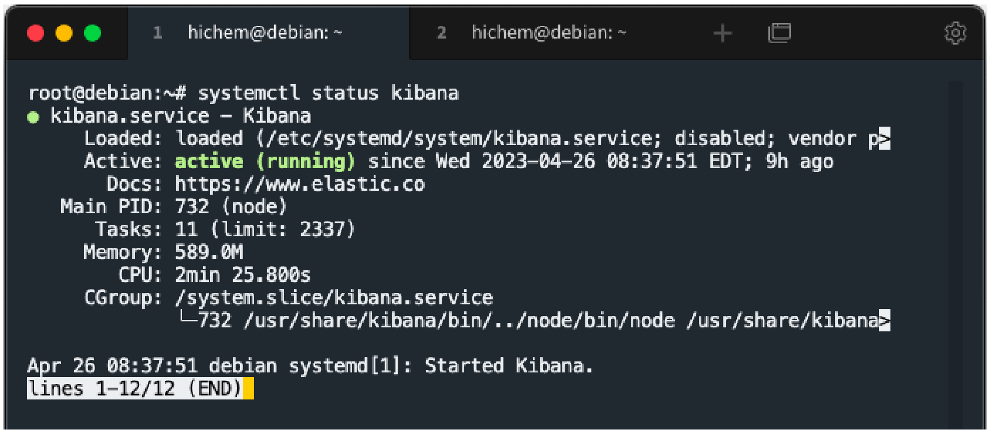
- **Configuring Kibana Credentials**
  we store the values in Kibana's keystore, which is an obfuscated file that Kibana can use to store secrets. first we navigate to "/usr/share/kibana/bin" directory. Next, we run the following command to set the username and password for Kibana:

```
cd /usr/share/kibana/bin
./kibana-keystore add elasticsearch.username
./kibana-keystore add elasticsearch.password
```

  We will receive a prompt to enter the Elasticsearch username and password

- **Running Kibana**
  Once the installation is done,we start and enable Kibana to run on system boot.

```
systemctl enable --now kibana
systemctl status kibana
```



```
root@debian:~# systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; disabled; vendor p>
     Active: active (running) since Wed 2023-04-26 08:37:51 EDT; 9h ago
       Docs: https://www.elastic.co
   Main PID: 732 (node)
      Tasks: 11 (limit: 2337)
     Memory: 589.0M
        CPU: 2min 25.800s
     CGroup: /system.slice/kibana.service
             └─732 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana>

Apr 26 08:37:51 debian systemd[1]: Started Kibana.
lines 1-12/12 (END)
```

Figure 3.8: Kibana status

3. **Install Filebeat**

   Filebeat is a lightweight shipper for collecting, forwarding and centralizing event log data.

   To install Filebeat from Elastic repos;

   ```
   apt install filebeat
   ```

   - **Filebeat modules**

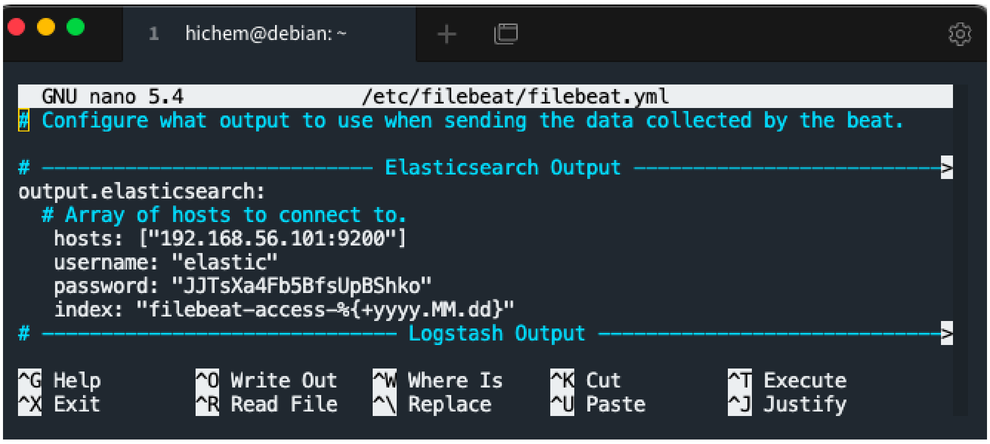
     Filebeat modules simplify the collection, parsing, and visualization of common log formats, so we enabled the most important one "System module" :

     ```
     filebeat modules enable system
     ```

   - **Configure Filebeat Output**

     Filebeat can send the collected data to various outputs. We are using Elasticsearch in this case.

     ```
     nano /etc/filebeat/filebeat.yml
     ```



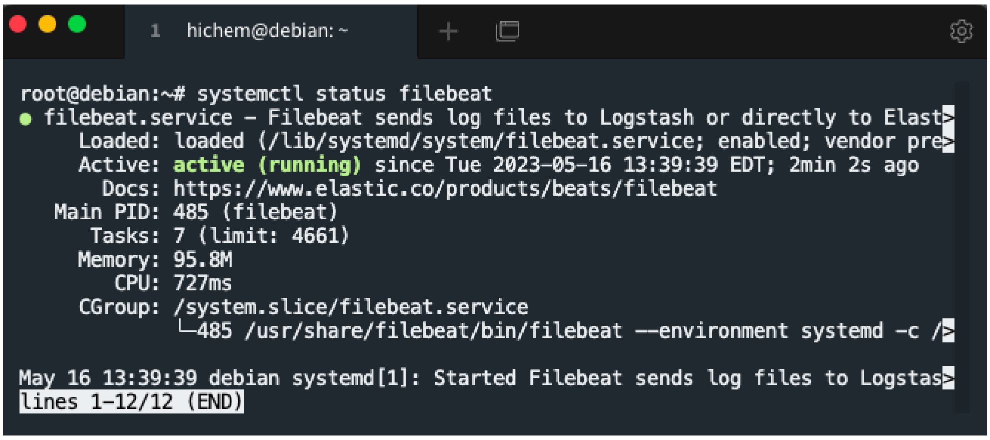
```
  GNU nano 5.4                    /etc/filebeat/filebeat.yml
# Configure what output to use when sending the data collected by the beat.

# ---------------------------- Elasticsearch Output ---------------------------->
output.elasticsearch:
  # Array of hosts to connect to.
   hosts: ["192.168.56.101:9200"]
   username: "elastic"
   password: "JJTsXa4Fb5BfsUpBShko"
   index: "filebeat-access-%{+yyyy.MM.dd}"
# ------------------------------ Logstash Output ------------------------------>

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify
```

Figure 3.9: Filebeat outputs to Elasticsearch

- **Running Filebeat**

  Once the installation is done,we start it and we check the status

```
systemctl start filebeat
systemctl status filebeat
```



```
root@debian:~# systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elast>
     Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor pre>
     Active: active (running) since Tue 2023-05-16 13:39:39 EDT; 2min 2s ago
       Docs: https://www.elastic.co/products/beats/filebeat
   Main PID: 485 (filebeat)
      Tasks: 7 (limit: 4661)
     Memory: 95.8M
        CPU: 727ms
     CGroup: /system.slice/filebeat.service
             └─485 /usr/share/filebeat/bin/filebeat --environment systemd -c />

May 16 13:39:39 debian systemd[1]: Started Filebeat sends log files to Logstas>
lines 1-12/12 (END)
```

Figure 3.10: Filebeat status

Accessing Elasticsearch Web Interface

Once we open the browser and visit "`http://192.168.56.101:5601`". we will be redirected to Kibana's login page:

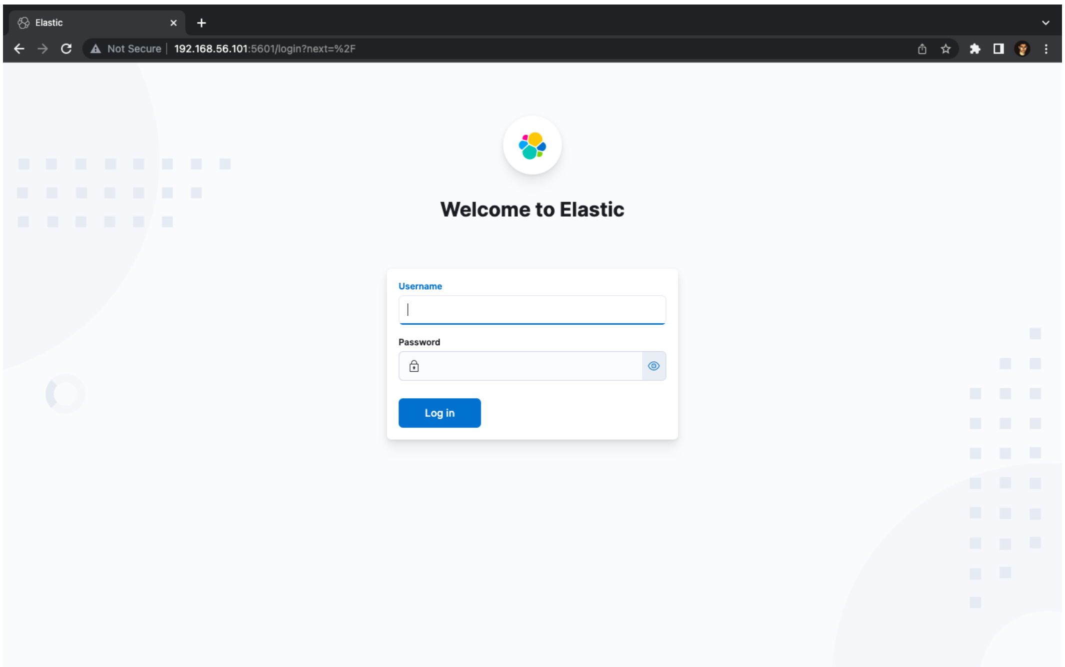
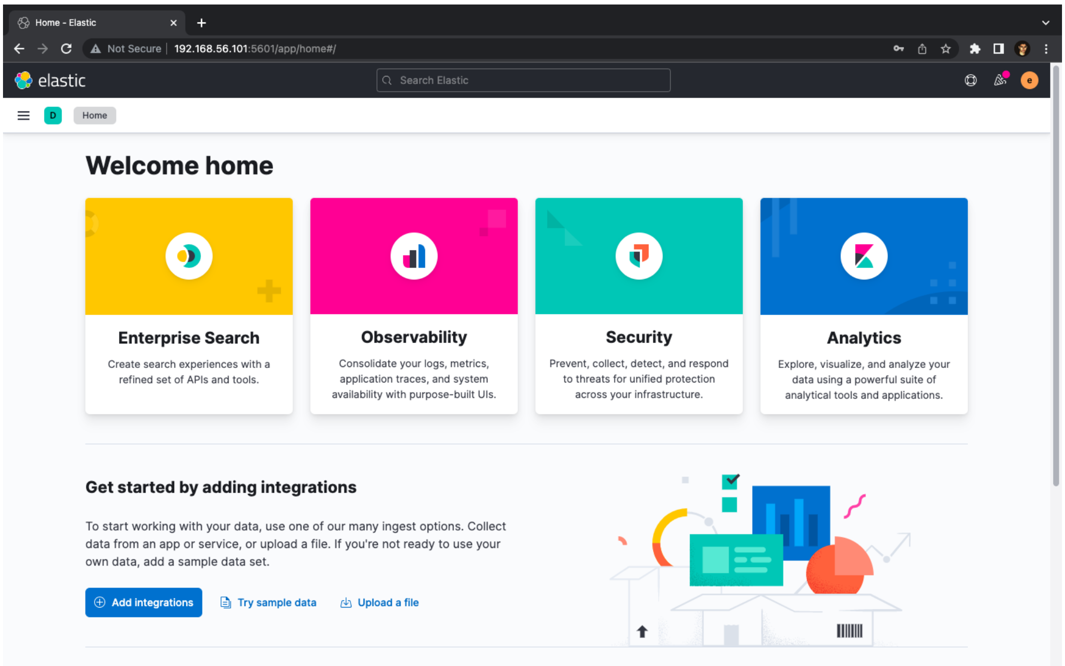


Figure 3.11: Kibana's login page

After Log in to our Kibana server using "elastic" for the Username, and the password, we land on the Kibana's home page:



Welcome home

Enterprise Search
Create search experiences with a refined set of APIs and tools.

Observability
Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.

Security
Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.

Analytics
Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

Get started by adding integrations

To start working with your data, use one of our many ingest options. Collect data from an app or service, or upload a file. If you're not ready to use your own data, add a sample data set.

Figure 3.12: Kibana's home page

3.3.1.2 Cortex installation

- First,Install Cortex and TheHive repository:

```
wget -qO- "https://raw.githubusercontent.com/TheHive-Project/Cortex/master/PGP-PUBLIC-KEY"
sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/cortex.gpg
wget -qO- https://raw.githubusercontent.com/TheHive-Project/Cortex/master/PGP-PUBLIC-KEY
sudo gpg --dearmor  -o /etc/apt/trusted.gpg.d/thehive.gpg
echo 'deb https://deb.thehive-project.org release main' | sudo tee -a /etc/apt/sources.list.d/thehive-project.list
```

And then we installed Cortex :

```
sudo apt update
sudo apt install cortex -y
```

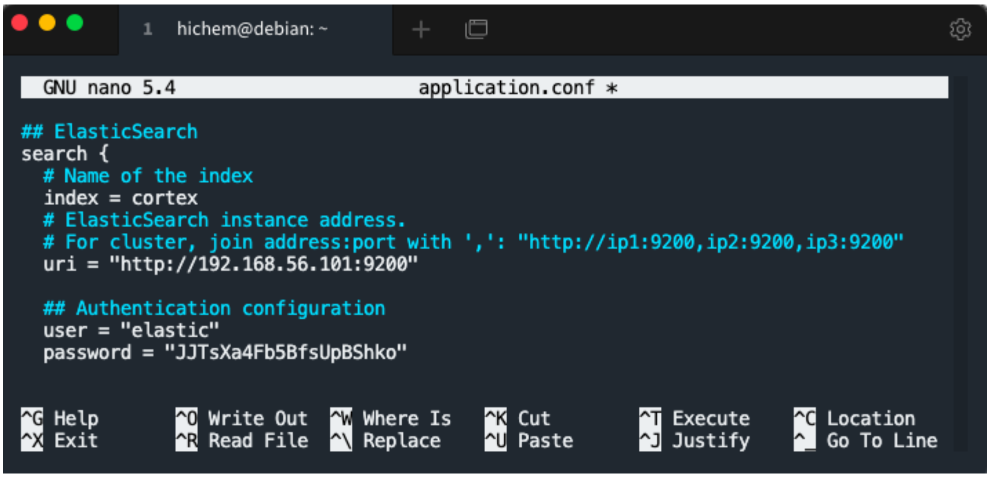
- **Configure Cortex**

We created Cortex Secret Key required for secure cryptographic Cortex functions:

```
sudo sed -i "/play.http.secret.key/s/^#//;s/\*\*\*CHANGEME\*\*\*/`cat \/dev\/urandom
tr -dc 'a-zA-Z0-9' | fold -w 64 | head -n 1`/" /etc/cortex/application.conf
```

And then, we configured Elasticsearch connection settings, Since we are running Cortex in a different node from Elasticsearch ,we changed the "uri" parameter and add our Elasticsearch IP address "192.168.56.101:9200" ,and also we configured the Elasticsearch authentication configuration by adding the user and password of our ELK service :

```
sudo nano /etc/cortex/application.conf
```



```
  GNU nano 5.4                    application.conf *

## ElasticSearch
search {
  # Name of the index
  index = cortex
  # ElasticSearch instance address.
  # For cluster, join address:port with ',': "http://ip1:9200,ip2:9200,ip3:9200"
  uri = "http://192.168.56.101:9200"

  ## Authentication configuration
  user = "elastic"
  password = "JJTsXa4Fb5BfsUpBShko"


^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Figure 3.13: Cortex connection settings

- **Enable and Configure Cortex Analyzers**

Cortex ships with the support of various analyzers. Some that are free to use, some that requires special access or valid subscription or product license.

By default, Cortex is configured to get the list of analyzers from "`https://download.thehive-project.org/analyzers.json`" but We have installed and host our analyzers on the Cortex server.

First, we installed required packages :

```
sudo apt install -y --no-install-recommends python2.7-dev python3-pip
    \python3-dev ssdeep libfuzzy-dev libfuzzy2 libimage-exiftool-perl
    libmagic1 \python3-testresources build-essential git libssl-dev
```

Next, we installed Python setup tools;

```
sudo pip3 install -U pip setuptools
sudo pip install -U pip setuptools
```

we cloned the Cortex-analyzers repository in the directory of our preferred directory;

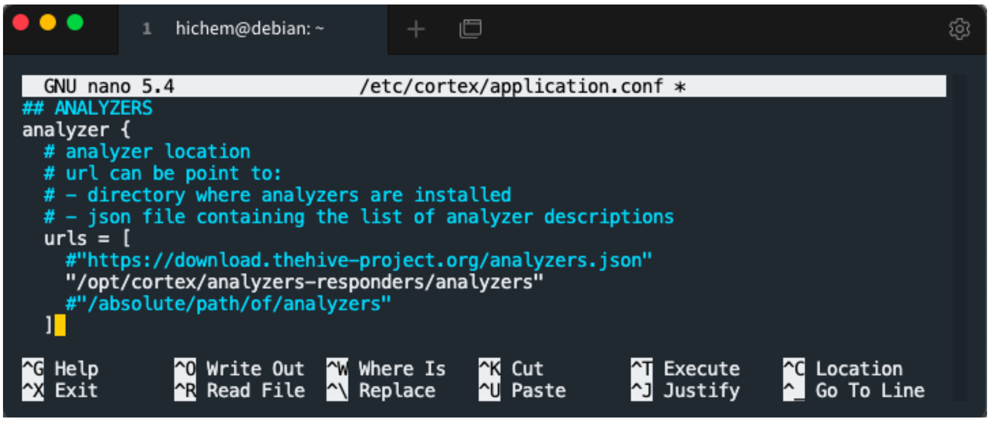
```
sudo git clone https://github.com/TheHive-Project/Cortex-Analyzers /
   opt/cortex/analyzers-responders
```

Install Python requirements of each analyzer thereafter;

```
for i in `find /opt/cortex/analyzers-responders -name 'requirements.
   txt'`; do sudo -H pip install -r $i; done && \
for i in `find /opt/cortex/analyzers-responders -name 'requirements.
    txt'`; do sudo -H pip3 install -r $i || true; done
```

After we have locally installed Analyzers, we configured Cortex to use these local analyzers by changing the urls to the local file system path containing Analyzers;

```
sudo nano /etc/cortex/application.conf
```



```
  GNU nano 5.4                    /etc/cortex/application.conf *
## ANALYZERS
analyzer {
  # analyzer location
  # url can be point to:
  # - directory where analyzers are installed
  # - json file containing the list of analyzer descriptions
  urls = [
    #"https://download.thehive-project.org/analyzers.json"
    "/opt/cortex/analyzers-responders/analyzers"
    #"/absolute/path/of/analyzers"
  ]

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Figure 3.14: Analyzers configuration

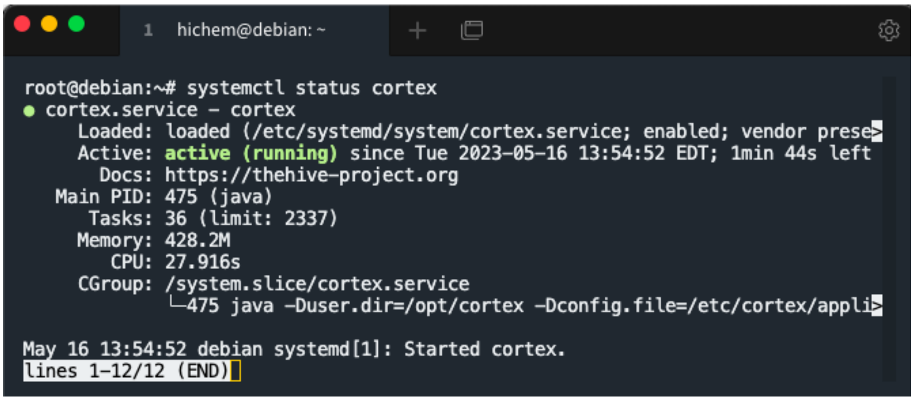
- **Running Cortex**

We started Cortex service :

```
sudo systemctl enable --now cortex
```

And the we checked the status :

```
systemctl status cortex
```



```
root@debian:~# systemctl status cortex
● cortex.service - cortex
     Loaded: loaded (/etc/systemd/system/cortex.service; enabled; vendor prese>
     Active: active (running) since Tue 2023-05-16 13:54:52 EDT; 1min 44s left
       Docs: https://thehive-project.org
   Main PID: 475 (java)
      Tasks: 36 (limit: 2337)
     Memory: 428.2M
        CPU: 27.916s
     CGroup: /system.slice/cortex.service
             └─475 java -Duser.dir=/opt/cortex -Dconfig.file=/etc/cortex/appli>

May 16 13:54:52 debian systemd[1]: Started cortex.
lines 1-12/12 (END)
```

Figure 3.15: Cortex status

- **Accessing Cortex Web Interface**

Open Cortex ports (9001/tcp) on Firewall :

```
ufw allow 9001/tcp
```

Once we open the browser and visit "`http://192.168.56.102:9001`". we will be redirected to Cortex login page:

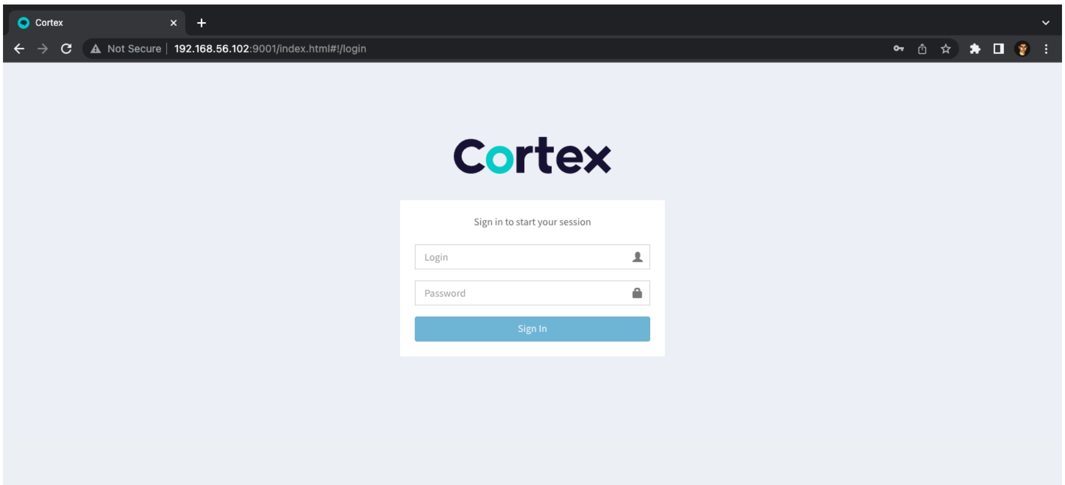

Cortex
Not Secure | 192.168.56.102:9001/index.html#!/login
Cortex
Sign in to start your session
Login
Password
Sign In

Figure 3.16: Cortex login page

After creating a Cortex Organization and Organization administrator, this will be the homepage :

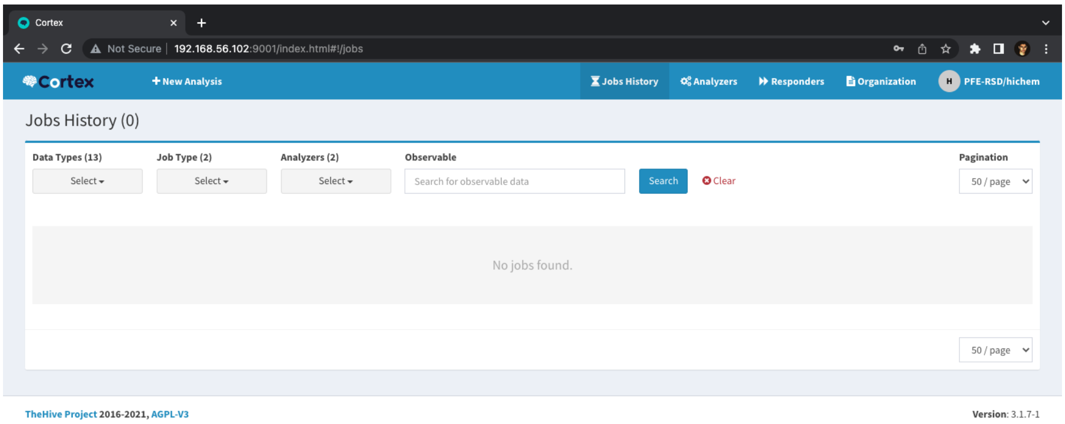


Figure 3.17: Cortex homepage

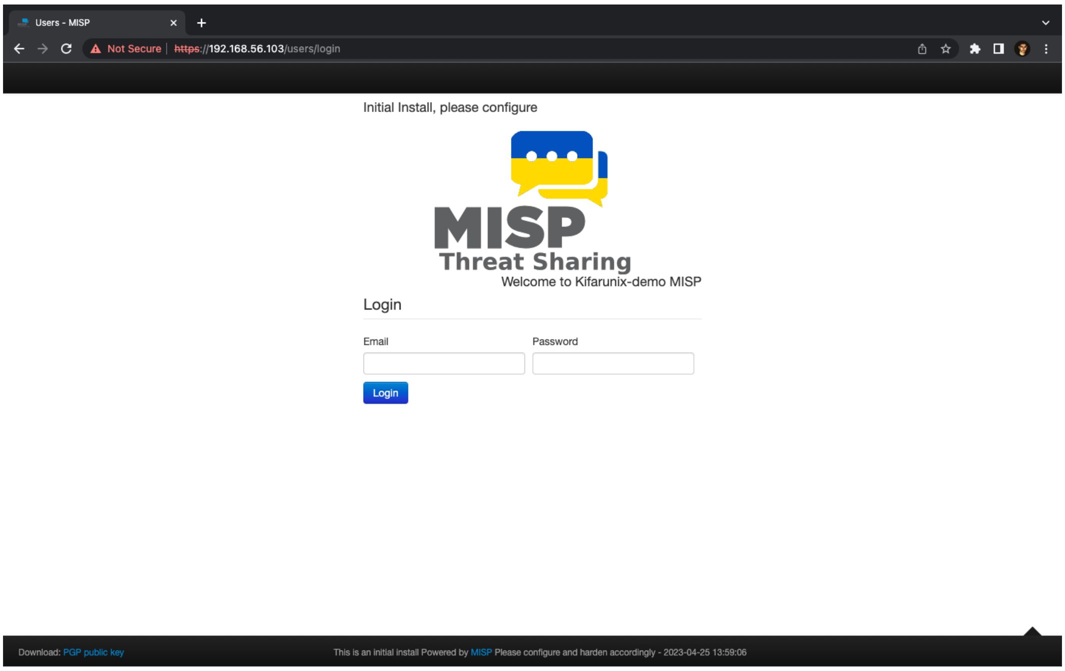
3.3.1.3 MISP installation

We have installed MISP with "INSTALL-misp.sh" script :

```
sudo apt-get update -y && sudo apt-get upgrade -y
wget --no-cache -O /tmp/INSTALL.sh https://raw.githubusercontent.com/MISP/
    MISP/2.4/INSTALL/INSTALL.sh
bash /tmp/INSTALL.sh
```

- **Accessing MISP Web Interface**

At this point, we can now login to MISP, using the address : "`https://192.168.56.103`"



Initial Install, please configure

Welcome to Kifarunix-demo MISP

This is an initial install Powered by MISP Please configure and harden accordingly - 2023-04-25 13:59:06

Figure 3.18: MISP Login page

Default credentials :

Username: admin@admin.test

Password: admin

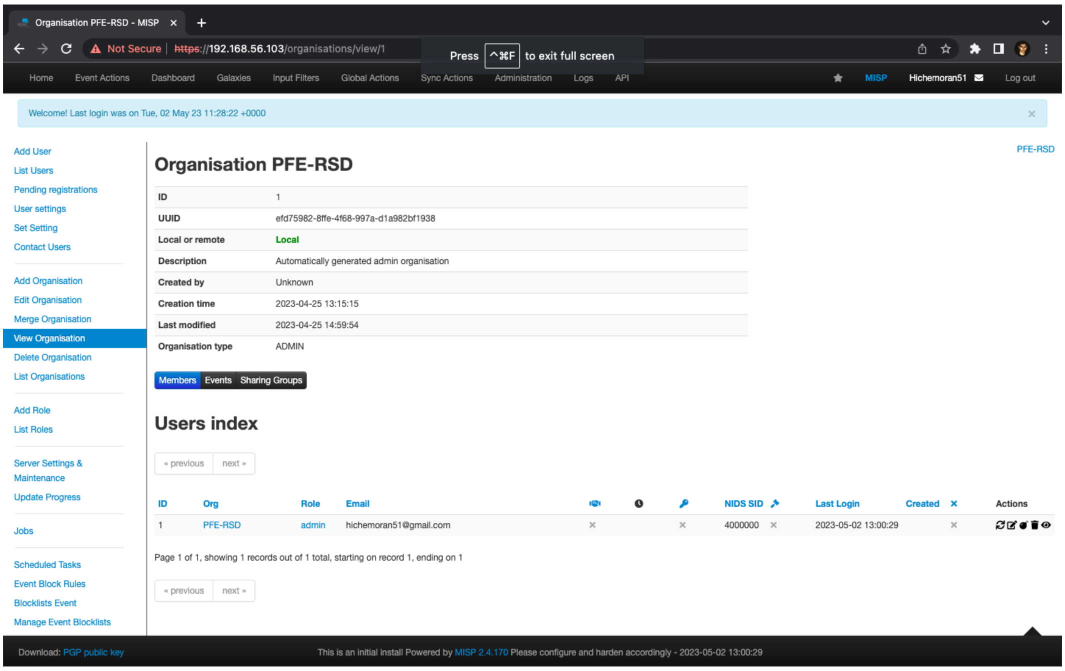
When we login, we reset this admin credentials to proceed;

- navigate to Administration > List Users.
- Click the edit button against the admin user.
- logout and login to confirm the user account changes.

- **Create an organisation**

  We have created a MISP organisation named "PFE-RSD" ,and generating a new API for it :

  - navigate to Administration > Add Organisation.
  - Fill out the fields.
  - click on the "Submit" button.



Welcome! Last login was on Tue, 02 May 23 11:28:22 +0000

Organisation PFE-RSD

ID	1
UUID	efd75982-8ffe-4f68-997a-d1a982bf1938
Local or remote	Local
Description	Automatically generated admin organisation
Created by	Unknown
Creation time	2023-04-25 13:15:15
Last modified	2023-04-25 14:59:54
Organisation type	ADMIN

Users index

ID	Org	Role	Email				NIDS SID		Last Login	Created	×	Actions
1	PFE-RSD	admin	hichemoran51@gmail.com	×		×	4000000	×	2023-05-02 13:00:29		×	

Page 1 of 1, showing 1 records out of 1 total, starting on record 1, ending on 1

This is an initial install Powered by MISP 2.4.170 Please configure and harden accordingly - 2023-05-02 13:00:29

Figure 3.19: MISP Organisation

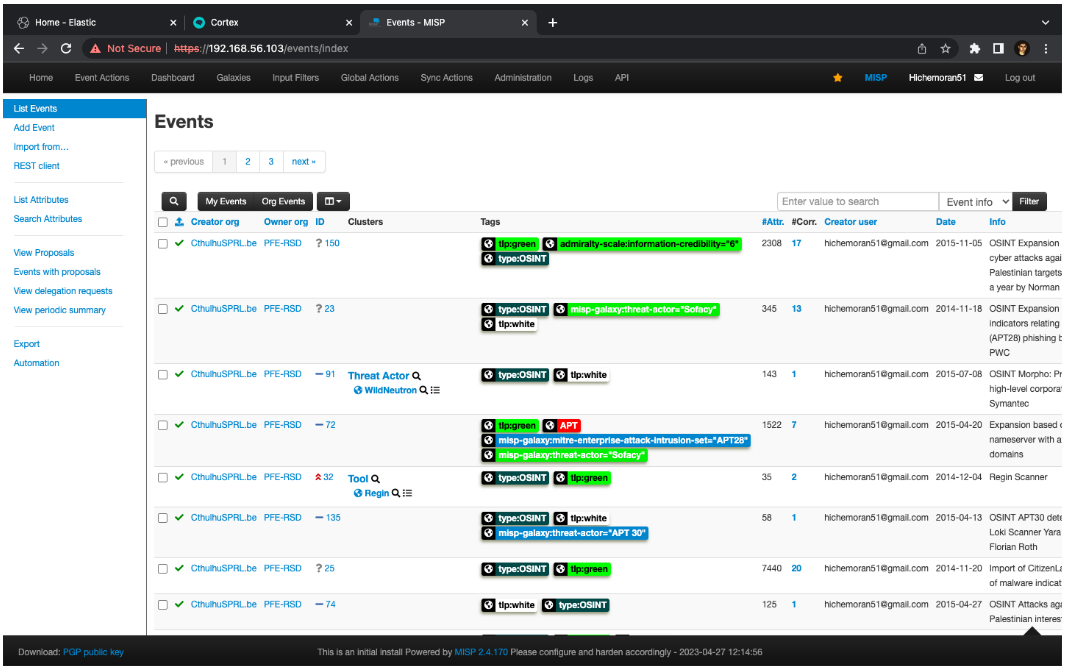
- **The MISP Events**

  On a fresh install, MISP has no events on it yet,but it ships with ability to pull events with patterns that can be used to detect malicious activities from some default open-source feeds.which are disabled by default.

  To enable the default feeds :

  - navigate to Sync Actions > List Feeds.
  - Select the two default feeds and click Enable Selected.

  When the feeds are enabled, it will start to download the events related to known malware, APTs, ransomware and all their attributes from the sources automatically, we confirm that by navigating to Event Actions > List Events.



☐	Creator org	Owner org	ID	Clusters	Tags	#Attr.	#Corr.	Creator user	Date	Info
☐ ✓	CthulhuSPRL.be	PFE-RSD	? 150		tlp:green admiralty-scale:information-credibility="6" type:OSINT	2308	17	hichemoran51@gmail.com	2015-11-05	OSINT Expansion cyber attacks agai Palestinian targets a year by Norman
☐ ✓	CthulhuSPRL.be	PFE-RSD	? 23		type:OSINT misp-galaxy:threat-actor="Sofacy" tlp:white	345	13	hichemoran51@gmail.com	2014-11-18	OSINT Expansion indicators relating (APT28) phishing b PWC
☐ ✓	CthulhuSPRL.be	PFE-RSD	— 91	Threat Actor WildNeutron	type:OSINT tlp:white	143	1	hichemoran51@gmail.com	2015-07-08	OSINT Morpho: Pr high-level corpora Symantec
☐ ✓	CthulhuSPRL.be	PFE-RSD	— 72		tlp:green APT misp-galaxy:mitre-enterprise-attack-intrusion-set="APT28" misp-galaxy:threat-actor="Sofacy"	1522	7	hichemoran51@gmail.com	2015-04-20	Expansion based nameserver with a domains
☐ ✓	CthulhuSPRL.be	PFE-RSD	⇑ 32	Tool Regin	type:OSINT tlp:green	35	2	hichemoran51@gmail.com	2014-12-04	Regin Scanner
☐ ✓	CthulhuSPRL.be	PFE-RSD	— 135		type:OSINT tlp:white misp-galaxy:threat-actor="APT 30"	58	1	hichemoran51@gmail.com	2015-04-13	OSINT APT30 dete Loki Scanner Yara Florian Roth
☐ ✓	CthulhuSPRL.be	PFE-RSD	? 25		type:OSINT tlp:green	7440	20	hichemoran51@gmail.com	2014-11-20	Import of CitizenLa of malware indicat
☐ ✓	CthulhuSPRL.be	PFE-RSD	— 74		tlp:white type:OSINT	125	1	hichemoran51@gmail.com	2015-04-27	OSINT Attacks aga Palestinian interes

Download: PGP public key

This is an initial install Powered by MISP 2.4.170 Please configure and harden accordingly - 2023-04-27 12:14:56

Figure 3.20: MISP Dashboard

3.3.1.4 TheHive installation

We will be installing TheHive 4 version, to install it we start with updating package list.

```
apt-get update && apt-get upgrade
```

Setup apt configuration with the release repository:

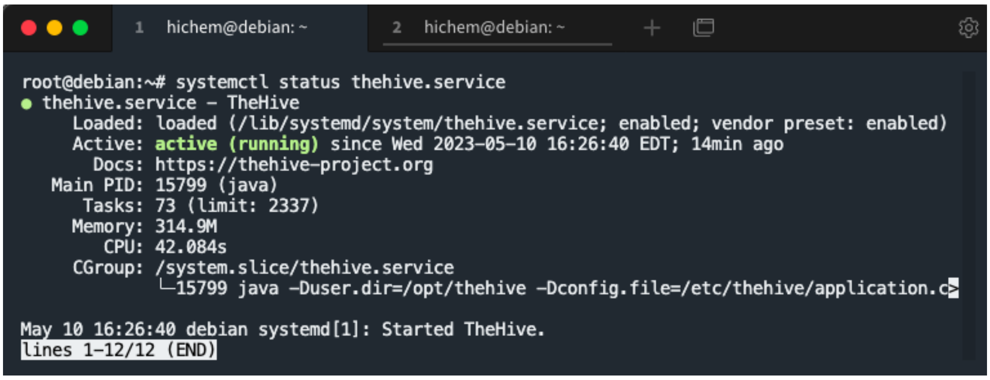
```
curl https://raw.githubusercontent.com/TheHive-Project/TheHive/master/PGP-PUBLIC-KEY | apt-key add -
echo 'deb https://deb.thehive-project.org release main' | tee -a /etc/apt/sources.list.d/thehive-project.list
 apt-get update
```

Then you will able to install TheHive 3.5.0+ the package using apt command:

```
apt install thehive4
```

We enabled and run TheHive service :

```
systemctl enable thehive
systemctl start thehive
systemctl status thehive
```



```
root@debian:~# systemctl status thehive.service
● thehive.service - TheHive
     Loaded: loaded (/lib/systemd/system/thehive.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-05-10 16:26:40 EDT; 14min ago
       Docs: https://thehive-project.org
   Main PID: 15799 (java)
      Tasks: 73 (limit: 2337)
     Memory: 314.9M
        CPU: 42.084s
     CGroup: /system.slice/thehive.service
             └─15799 java -Duser.dir=/opt/thehive -Dconfig.file=/etc/thehive/application.c>

May 10 16:26:40 debian systemd[1]: Started TheHive.
lines 1-12/12 (END)
```

Figure 3.21: TheHive status

- **Accessing TheHive Web Interface**

  Once we open the browser and visit "`http://192.168.56.104:9000`". we will be redirected to TheHive login page:

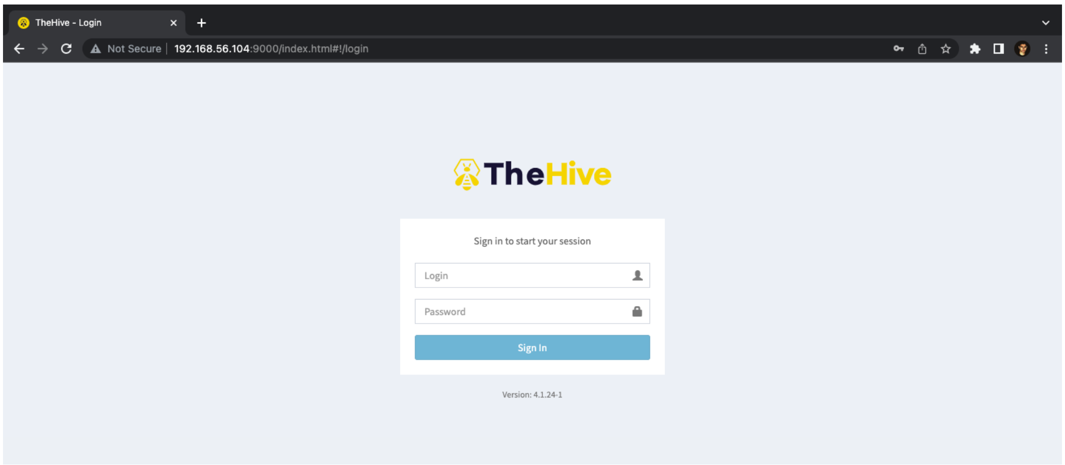
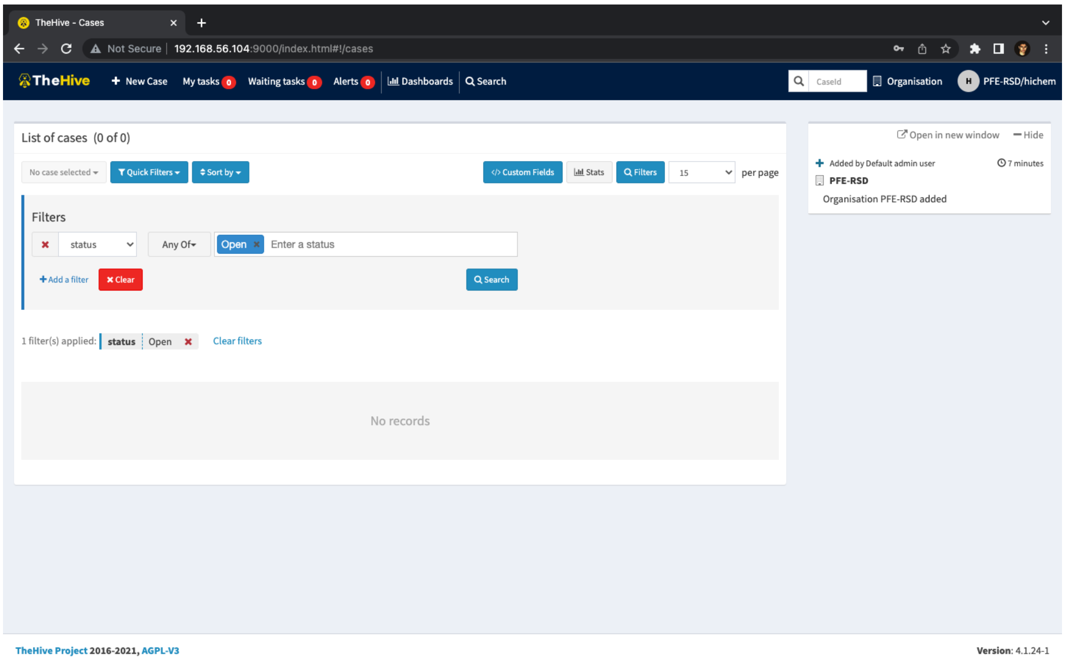

TheHive - Login
Not Secure | 192.168.56.104:9000/index.html#!/login
TheHive
Sign in to start your session
Login
Password
Sign In
Version: 4.1.24-1

Figure 3.22: TheHive login page

After creating TheHive Organization and Organization administrator, this will be the homepage :



List of cases (0 of 0)

1 filter(s) applied: status Open

No records

Added by Default admin user 7 minutes

PFE-RSD

Organisation PFE-RSD added

Figure 3.23: TheHive homepage

3.3.1.5 Shuffle installation

Installation of Shuffle is currently only available in docker, so we will install docker first :

```
apt install docker.io
apt install docker-compose
```

and then we download Shuffle :

```
git clone https://github.com/frikky/Shuffle
cd Shuffle
```

Fix prerequisites for the Opensearch database (Elasticsearch):

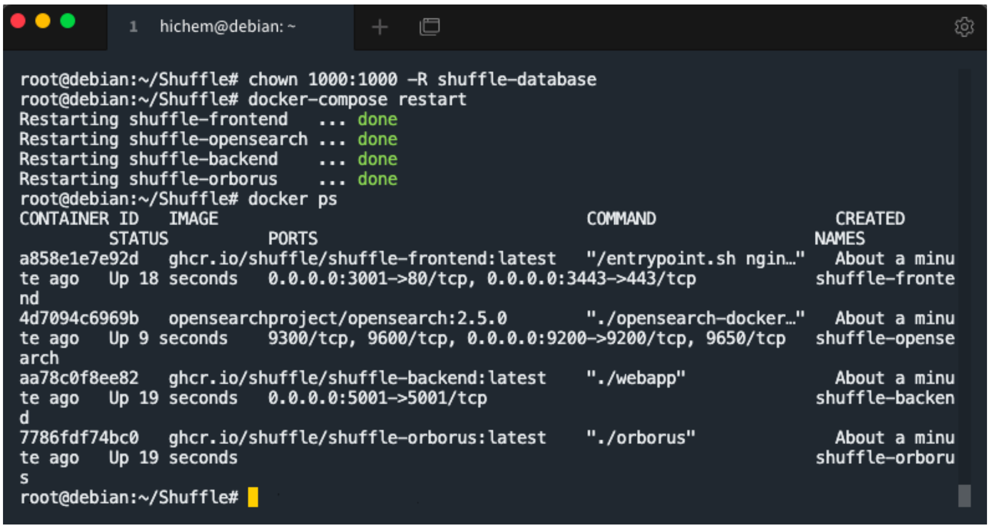
```
mkdir shuffle-database
chown -R 1000:1000 shuffle-database
```

The Docker setup is done with docker-compose, so we run docker-compose

```
docker-compose up -d
```

and the we verify if the service is running :

```
docker ps
```



```
root@debian:~/Shuffle# chown 1000:1000 -R shuffle-database
root@debian:~/Shuffle# docker-compose restart
Restarting shuffle-frontend   ... done
Restarting shuffle-opensearch ... done
Restarting shuffle-backend    ... done
Restarting shuffle-orborus    ... done
root@debian:~/Shuffle# docker ps
CONTAINER ID   IMAGE                                    COMMAND                  CREATED
         STATUS          PORTS                                                      NAMES
a858e1e7e92d   ghcr.io/shuffle/shuffle-frontend:latest  "/entrypoint.sh ngin…"   About a minu
te ago   Up 18 seconds   0.0.0.0:3001->80/tcp, 0.0.0.0:3443->443/tcp                shuffle-fronte
nd
4d7094c6969b   opensearchproject/opensearch:2.5.0       "./opensearch-docker…"   About a minu
te ago   Up 9 seconds    9300/tcp, 9600/tcp, 0.0.0.0:9200->9200/tcp, 9650/tcp       shuffle-opense
arch
aa78c0f8ee82   ghcr.io/shuffle/shuffle-backend:latest   "./webapp"               About a minu
te ago   Up 19 seconds   0.0.0.0:5001->5001/tcp                                     shuffle-backen
d
7786fdf74bc0   ghcr.io/shuffle/shuffle-orborus:latest   "./orborus"              About a minu
te ago   Up 19 seconds                                                              shuffle-orboru
s
root@debian:~/Shuffle# 
```

Figure 3.24: Shuffle service running

- **Accessing Shuffle Web Interface**

  Once we open the browser and visit `https://192.168.56.105:3443`. we will be redirected to Shuffle login page:

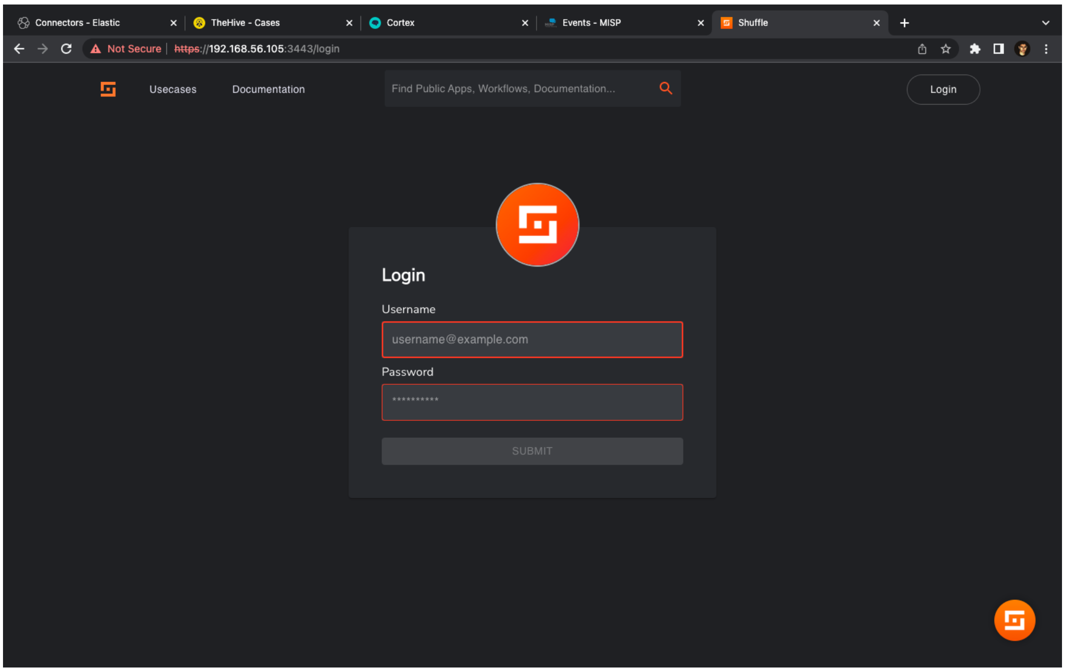


Figure 3.25: Shuffle login page

After creating a new user, this will be redirected to the homepage :

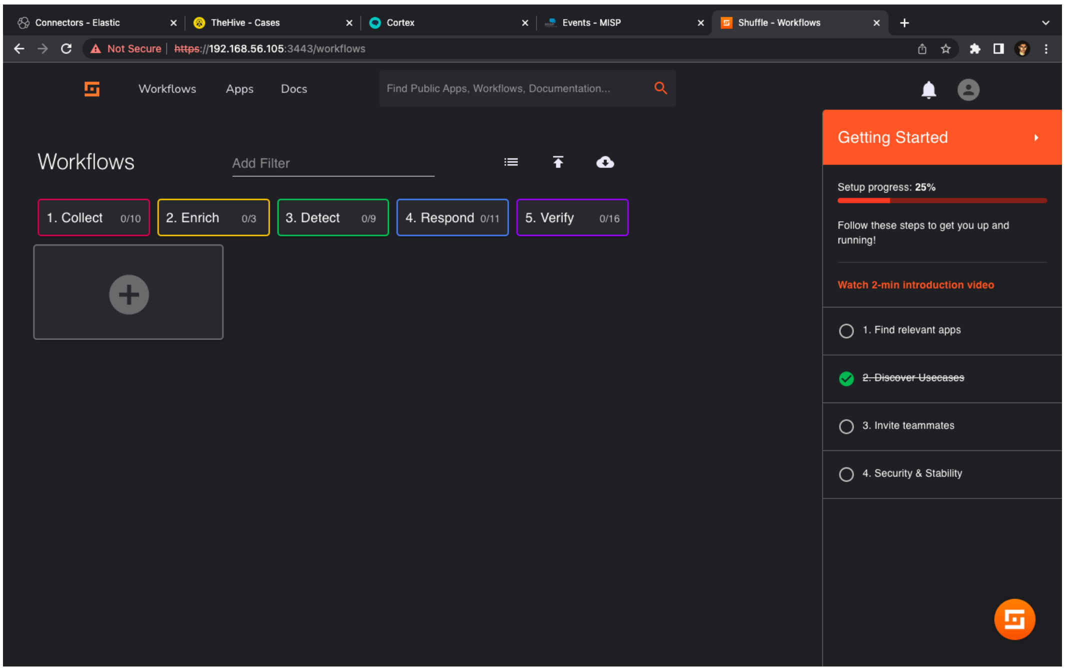


Figure 3.26: Shuffle homepage

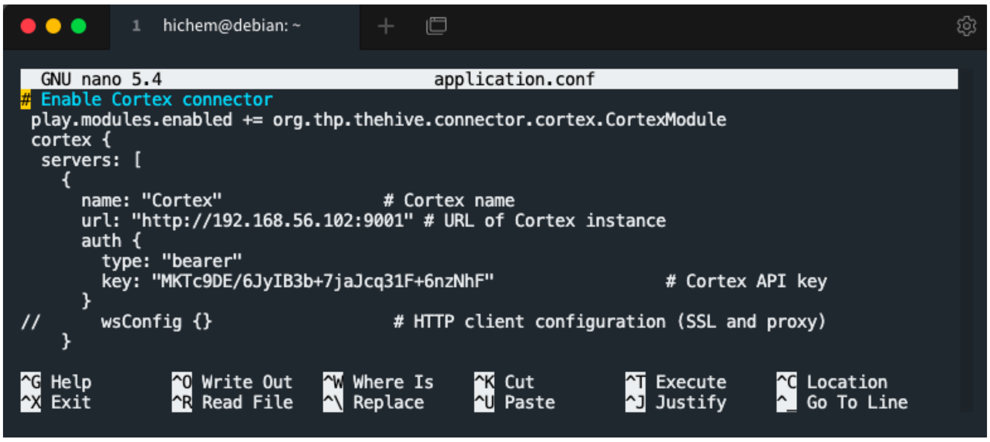
3.3.2 Tools Integration

3.3.2.1 TheHive with Cortex and MISP

After Creating a user in Cortex and in MISP ,we create an API key for each user and we use it to integrate TheHive with them by adjusting TheHive configuration file ,adding the IP address and the API key for Cortex and for MISP .

```
nano /etc/thehive/application.conf
```

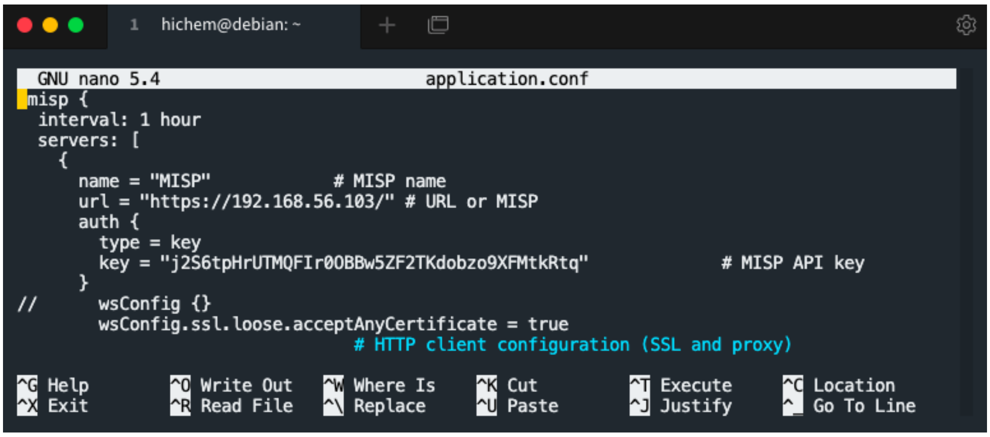
- Adding the IP address and the API key for Cortex



```
  GNU nano 5.4                       application.conf
# Enable Cortex connector
 play.modules.enabled += org.thp.thehive.connector.cortex.CortexModule
 cortex {
  servers: [
    {
      name: "Cortex"                # Cortex name
      url: "http://192.168.56.102:9001" # URL of Cortex instance
      auth {
        type: "bearer"
        key: "MKTc9DE/6JyIB3b+7jaJcq31F+6nzNhF"                # Cortex API key
      }
//      wsConfig {}                  # HTTP client configuration (SSL and proxy)
    }

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^_ Go To Line
```

Figure 3.27: TheHive connection settings with cortex

- Adding the IP address and the API key for MISP



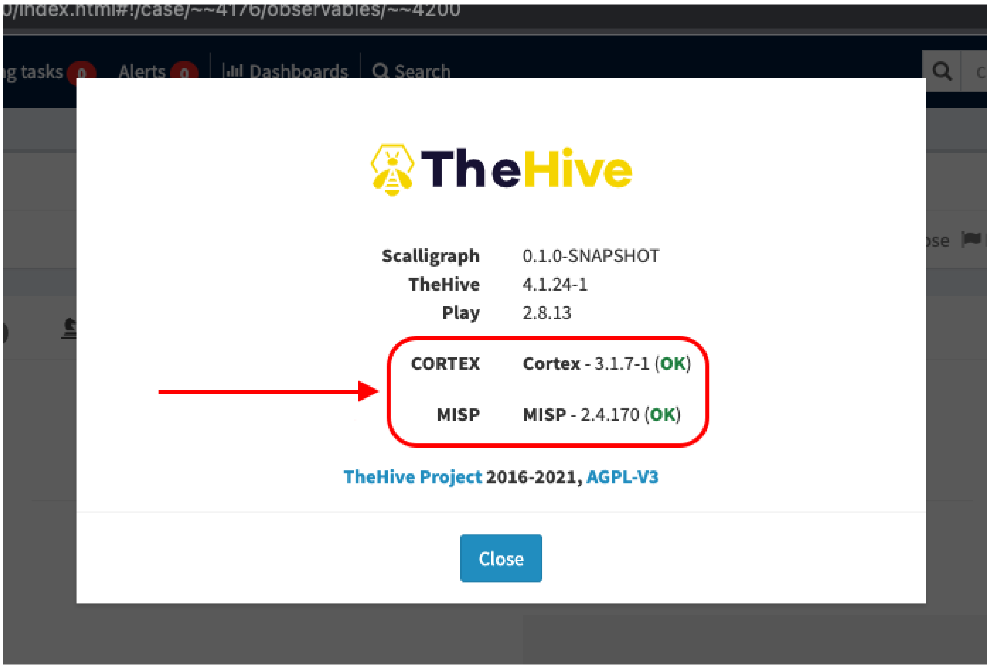
```
  GNU nano 5.4                          application.conf
misp {
  interval: 1 hour
  servers: [
    {
      name = "MISP"              # MISP name
      url = "https://192.168.56.103/" # URL or MISP
      auth {
        type = key
        key = "j2S6tpHrUTMQFIr0OBBw5ZF2TKdobzo9XFMtkRtq"            # MISP API key
      }
//      wsConfig {}
        wsConfig.ssl.loose.acceptAnyCertificate = true
                                     # HTTP client configuration (SSL and proxy)

^G Help          ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location
^X Exit          ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^_ Go To Line
```

Figure 3.28: TheHive connection settings with MISP

restart TheHive service

```
systemctl restart thehive
```

And we can check if the integration is done in TheHive by clicking on the username in the top right and then select "About"



TheHive

Scalligraph 0.1.0-SNAPSHOT
TheHive 4.1.24-1
Play 2.8.13
CORTEX Cortex - 3.1.7-1 (OK)
MISP MISP - 2.4.170 (OK)

TheHive Project 2016-2021, AGPL-V3

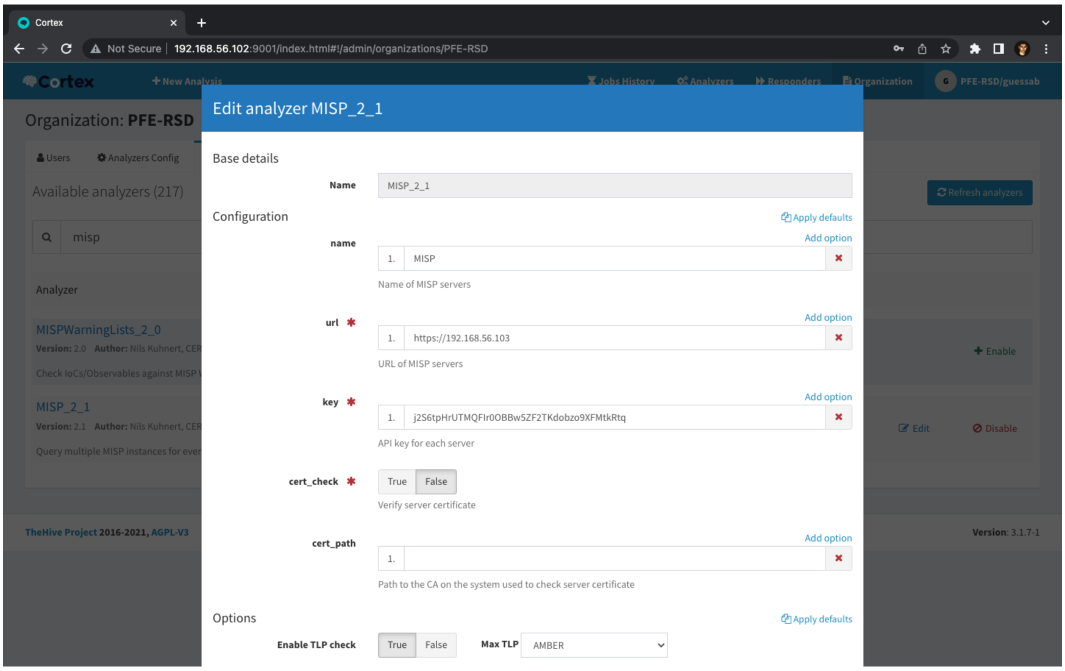
Figure 3.29: TheHive connection with cortex and MISP

3.3.2.2 Cortex with MISP

We added the MISP analyzer to cortex as follow :

- Login to Cortex UI and go to Organization > Analyzers
- Search for MISP and Fill out the fields.

- Click on the "False" in the cert check field.
- Click on the "Save" button.



Not Secure | 192.168.56.102:9001/index.html#!/admin/organizations/PFE-RSD

Organization: PFE-RSD

Available analyzers (217)

MISPWarningLists_2_0 — Version: 2.0 Author: Nils Kuhnert, CER

MISP_2_1 — Version: 2.1 Author: Nils Kuhnert, CER

Edit analyzer MISP_2_1

Base details

Name: MISP_2_1

Configuration

name: 1. MISP — Name of MISP servers

url *: 1. https://192.168.56.103 — URL of MISP servers

key *: 1. j2S6tpHrUTMQFIr0OBBw5ZF2TKdobzo9XFMtkRtq — API key for each server

cert_check *: True [✓ False] — Verify server certificate

cert_path: 1. — Path to the CA on the system used to check server certificate

Options

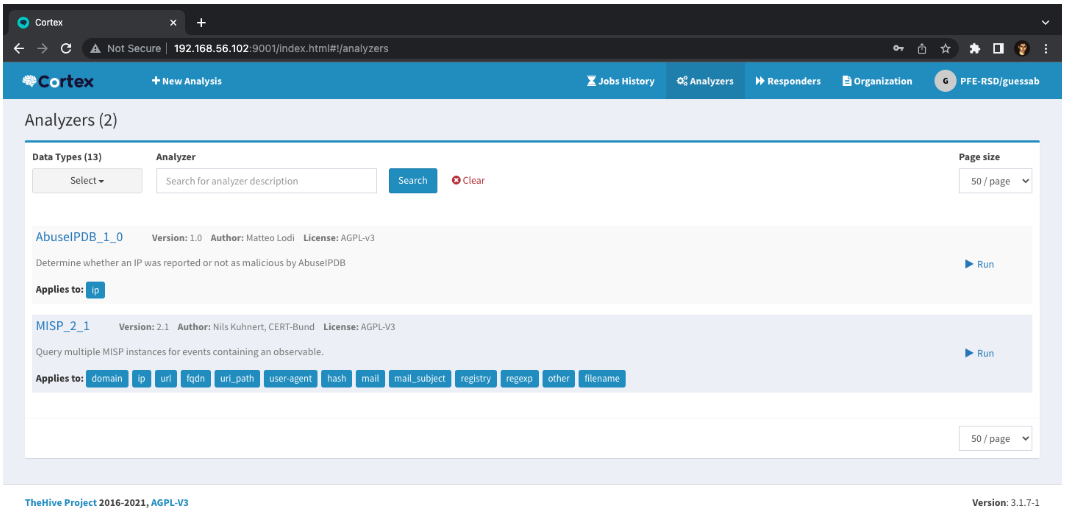
Enable TLP check: [✓ True] False

Max TLP: AMBER

TheHive Project 2016-2021, AGPL-V3 — Version: 3.1.7-1

Figure 3.30: adding MISP analyzer

and then go to Analyzers page to check if it's added :



Not Secure | 192.168.56.102:9001/index.html#!/analyzers

Analyzers (2)

AbuseIPDB_1_0 — Version: 1.0 Author: Matteo Lodi License: AGPL-v3

Determine whether an IP was reported or not as malicious by AbuseIPDB

Applies to: ip

MISP_2_1 — Version: 2.1 Author: Nils Kuhnert, CERT-Bund License: AGPL-V3

Query multiple MISP instances for events containing an observable.

Applies to: domain, ip, url, fqdn, uri_path, user-agent, hash, mail, mail_subject, registry, regexp, other, filename

TheHive Project 2016-2021, AGPL-V3 — Version: 3.1.7-1

Figure 3.31: MISP analyzer

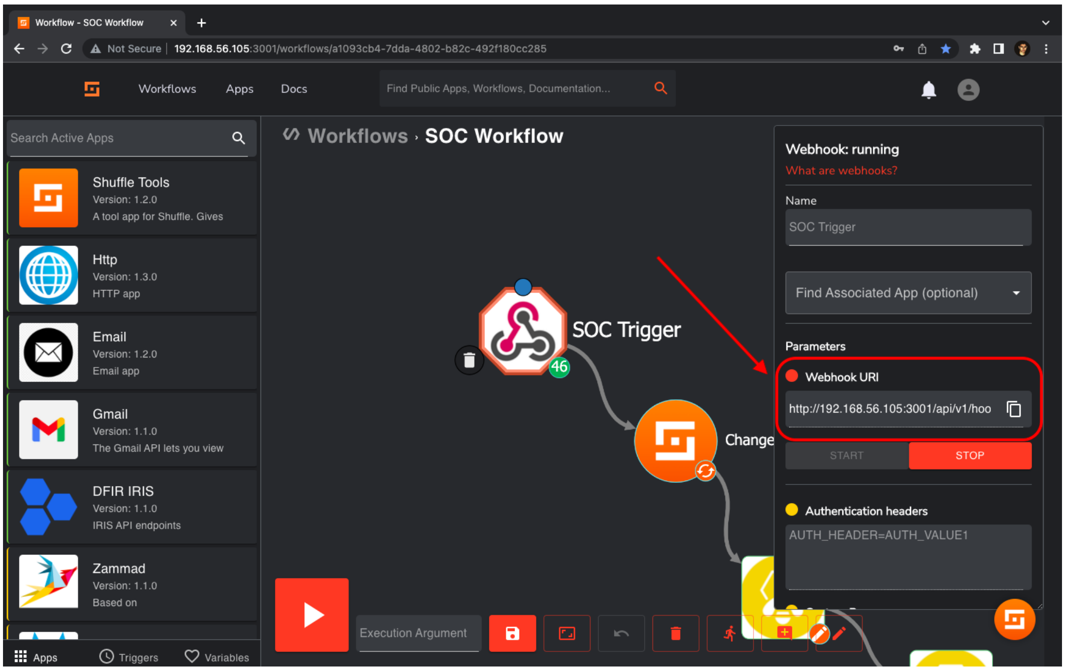
3.3.2.3 ELK with Shuffle (SOAR)

The integration of ELK with Shuffle is made by using a webhooks, which provides a powerful combination for security incident management and automation.

Webhooks provide a mechanism for communication between different systems by allowing real-time data exchange through HTTP callbacks.

The first step is to get the Webhook trigger URL from Shuffle by accessing to Shuffle Web interface and to our Workflow, Clicking on the Webhook app

- Accessing to Shuffle Web interface > Workflows > SOC workflow (our workflow)
- Choose the webhook trigger app
- Copy the Webhook URI



Not Secure | 192.168.56.105:3001/workflows/a1093cb4-7dda-4802-b82c-492f180cc285

Webhook: running

Name: SOC Trigger

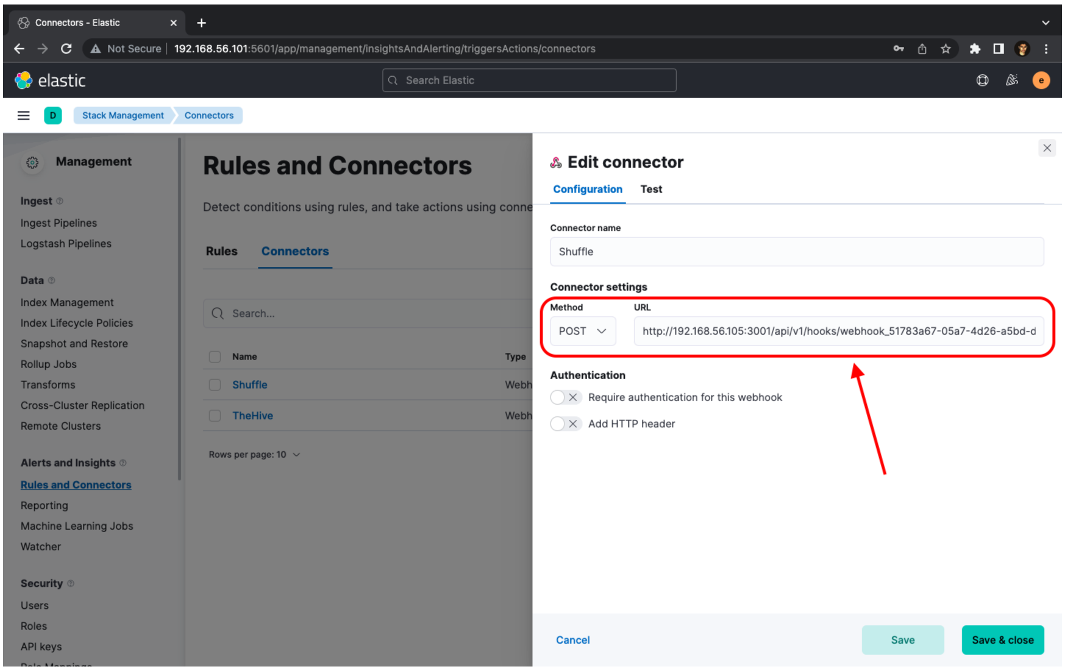
Webhook URI: http://192.168.56.105:3001/api/v1/hoo

Authentication headers: AUTH_HEADER=AUTH_VALUE1

Figure 3.32: Shuffle's Webhook setup

To setup a webhook in ELK we followed the steps below :

- After accessing to the Kibana web interface we go to Stack management > Rules and Connectors
- Choose Connectors and then click on "Create connector"
- We add the connector name and we choose the "POST" method
- Paste the copied URL from Shuffle and click "Save"



Name	Type
Shuffle	Webh
TheHive	Webh

Connector name: Shuffle

Connector settings

Method: POST

URL: http://192.168.56.105:3001/api/v1/hooks/webhook_51783a67-05a7-4d26-a5bd-d

Authentication

○ ✕ Require authentication for this webhook

○ ✕ Add HTTP header

Figure 3.33: Kibana's Webhook setup

And then we just need to add the webhook connector to the rule so it work as a trigger for our Shuffle workflow every-time we get an alert from that rule

3.3.3 Shuffle Workflow description

The following workflow description outlines the key steps involved in a Shuffle SOAR workflow. This workflow integrates various tools and platforms, including TheHive, Cortex, and the Outlook app, to enable efficient incident response and collaboration within a SOC environment :

1. The workflow begins with a webhook trigger, which serves as the starting point for initiating the workflow. This webhook is configured to receive notifications or events from the kibana alert console.
2. Once the webhook is triggered, the workflow proceeds to create an alert in TheHive,The alert contains relevant information about the security event or incident that triggered the workflow.
3. After the alert is generated, the workflow continues by creating an observable within TheHive. which represents a piece of data or an indicator related to the alert, such as an IP address, domain name, file hash, or URL.
4. In the next step, a case is created in TheHive based on the alert and associated observable. A case represents the container for managing and tracking the incident investigation. It allows SOC analysts to collaborate, document findings, assign tasks,

and track the progress of the incident response process.

5. Concurrently with the creation of the case in TheHive, The workflow sends the observable associated with the alert to Cortex for analysis,and once Cortex completes the analysis, it generates analysis results or reports. These results provide insights into the nature of the observable, its reputation, potential threats, or any additional information that can aid in the investigation and response process.
6. Finally, the workflow sends a notification via the Outlook app to relevant stakeholders, informing them about the newly created case.

   This workflow enables efficient incident response by leveraging automated alert creation, analysis in Cortex, and timely notifications to facilitate collaboration and response coordination.

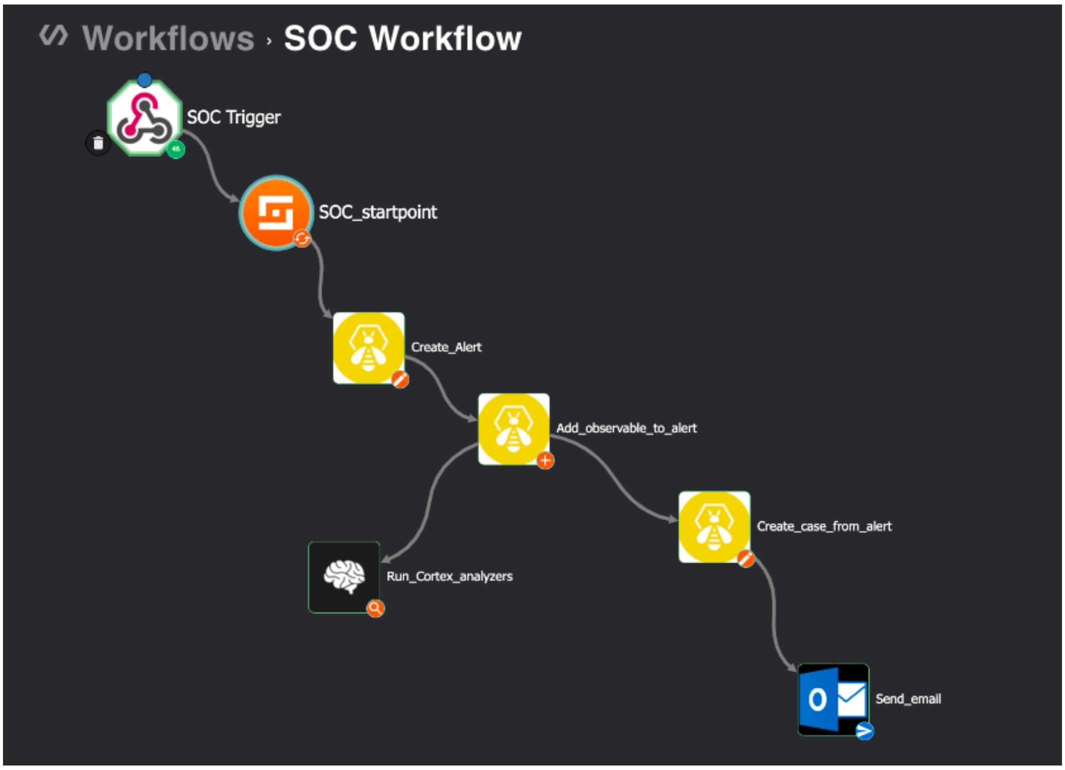


Figure 3.34: SOC Workflow

Flexibility advantage

Our SOC is designed to be highly adaptable and scalable, ensuring that it can effectively handle the security needs of our university. In scenarios where there is a significant increase in network traffic or an expanding number of endpoints, we have the capability to add additional servers to augment the main SOC server and alleviate the workload. To achieve this, we securely store the credentials of the new server within our the shuffle system, allowing it to control and manage it over all SOC components. This flexible approach enables us to dynamically respond to evolving security demands and ensure optimal performance and protection across our university's digital infrastructure.

3.4 Experiments and tests

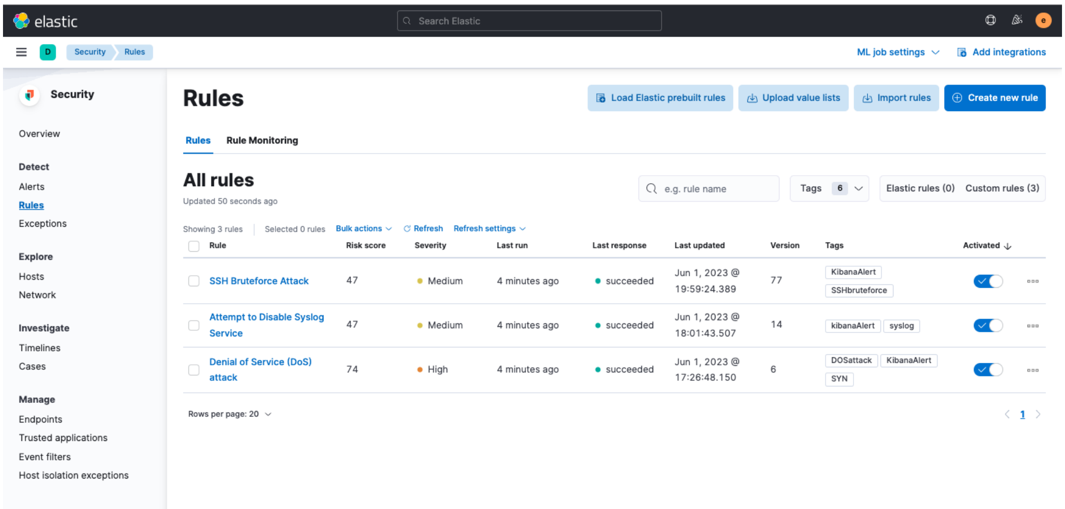
Before diving into our testing process, we would like to clarify an important point. Although our Security Operations Center (SOC) was initially intended for our university, due to the unavailability of university logs, we decided to conduct comprehensive testing by simulating live attacks. This allowed us to evaluate the effectiveness of our SOC under realistic conditions and identify potential vulnerabilities. By subjecting our system to these simulated attacks, we gained valuable insights into the robustness and resilience of our SOC's security measures. Let us now share the details of our testing methodology and the remarkable outcomes we achieved.

3.4.1 Objectives of the experiments

- Assess the detection capabilities of our SOC when exposed to simulated cyberattacks.
- Measure the effectiveness of the SOC in generating timely alerts and conducting automated analysis of the attack artifacts.
- Evaluate the response capabilities of our SOC in orchestrating incident response actions based on the detected attacks.

3.4.2 Creation of the rules

Elastic Security rules enable the detection and response to security incidents within an Elastic Stack environment. They identify malicious activity, unauthorized access, and suspicious network traffic, based on the analytics applied on the log files, Elastic Security provide a pre-built rules and these rules cover a wide range of security concerns. in this experiments we will create a customized rules to detect the attacks and then activate them.



Rule	Risk score	Severity	Last run	Last response	Last updated	Version	Tags	Activated
SSH Bruteforce Attack	47	● Medium	4 minutes ago	● succeeded	Jun 1, 2023 @ 19:59:24.389	77	KibanaAlert SSHbruteforce	✓
Attempt to Disable Syslog Service	47	● Medium	4 minutes ago	● succeeded	Jun 1, 2023 @ 18:01:43.507	14	kibanaAlert syslog	✓
Denial of Service (DoS) attack	74	● High	4 minutes ago	● succeeded	Jun 1, 2023 @ 17:26:48.150	6	DOSattack KibanaAlert SYN	✓

Figure 3.35: Elastic security rules

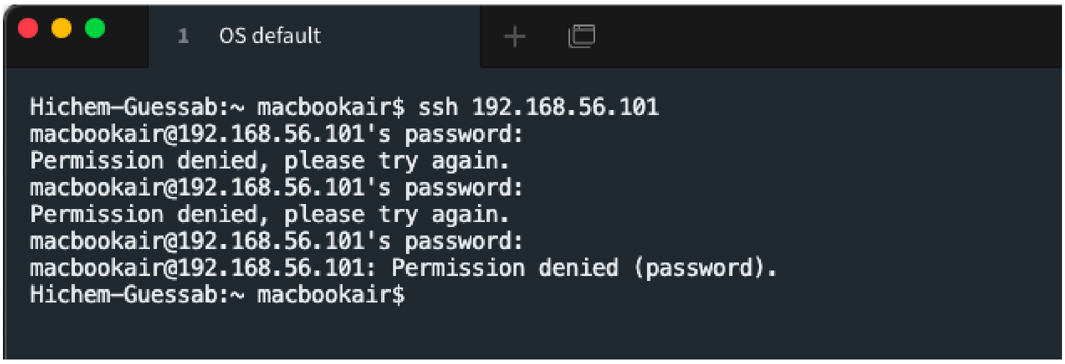
3.4.3 Running attacks scenario

we will execute a series of simulated cyberattacks on the test environment, targeting various attack methods, such as SSH bruteforce attack,attempt to disable syslog service and a Dos attack.

- Note : Here in this tests we will be targeting the ELK stack server because we have already installed the beat tools (Filebeat , Auditbeat) in it and configured them to send log files to Elasticsearch.

1. **First attack scenario(SSH Bruteforce Attack) :**

   After creating the customized rule to detect an SSH bruteforce attack we will be applying this attack by doing a high number of an ssh connection with a wrong password :

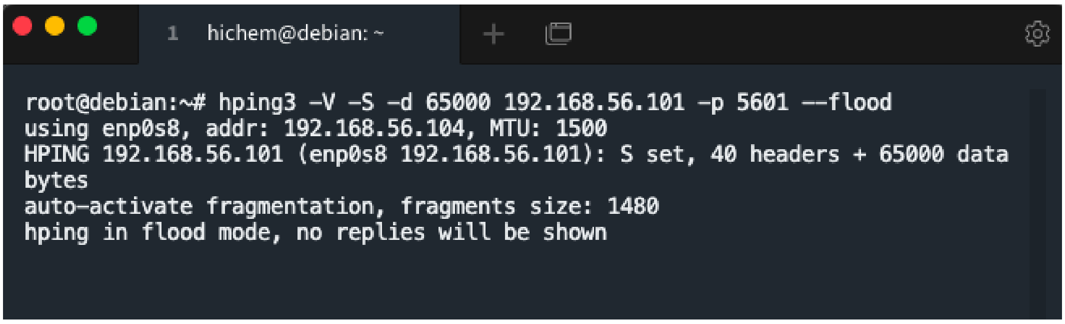


```
Hichem-Guessab:~ macbookair$ ssh 192.168.56.101
macbookair@192.168.56.101's password:
Permission denied, please try again.
macbookair@192.168.56.101's password:
Permission denied, please try again.
macbookair@192.168.56.101's password:
macbookair@192.168.56.101: Permission denied (password).
Hichem-Guessab:~ macbookair$
```

Figure 3.36: SSH bruteforce attack

2. **Second attack scenario(Denial of Service (DOS) attack) :**

   After creating the customized rule to detect an Denial of Service (DOS) attack we will be applying this attack by overwhelm the targeted server by flooding it with a large number of TCP connection requests (SYN packets) using the hping tool on the 5601 port :



```
root@debian:~# hping3 -V -S -d 65000 192.168.56.101 -p 5601 --flood
using enp0s8, addr: 192.168.56.104, MTU: 1500
HPING 192.168.56.101 (enp0s8 192.168.56.101): S set, 40 headers + 65000 data
bytes
auto-activate fragmentation, fragments size: 1480
hping in flood mode, no replies will be shown
```

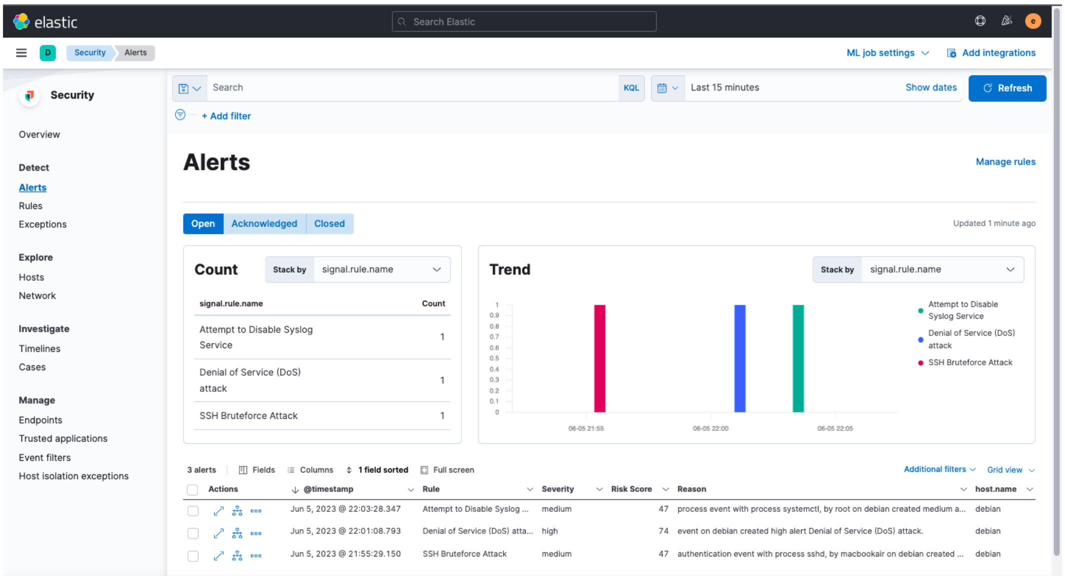
Figure 3.37: Denial of Service (DoS) attack

3. **Third attack scenario(Attempt to Disable Syslog Service) :**
   After creating the customized rule to detect an Attempt to Disable Syslog Service we will be applying this attack by trying to stop the syslog service with one of the stopping commands :

```
systemctl kill rsyslog
```

3.4.4 Attacks scenario results

By checking the Kibana security alerts we can see that the previous attacks have been successfully detected by the kibana detection rules :



Alerts

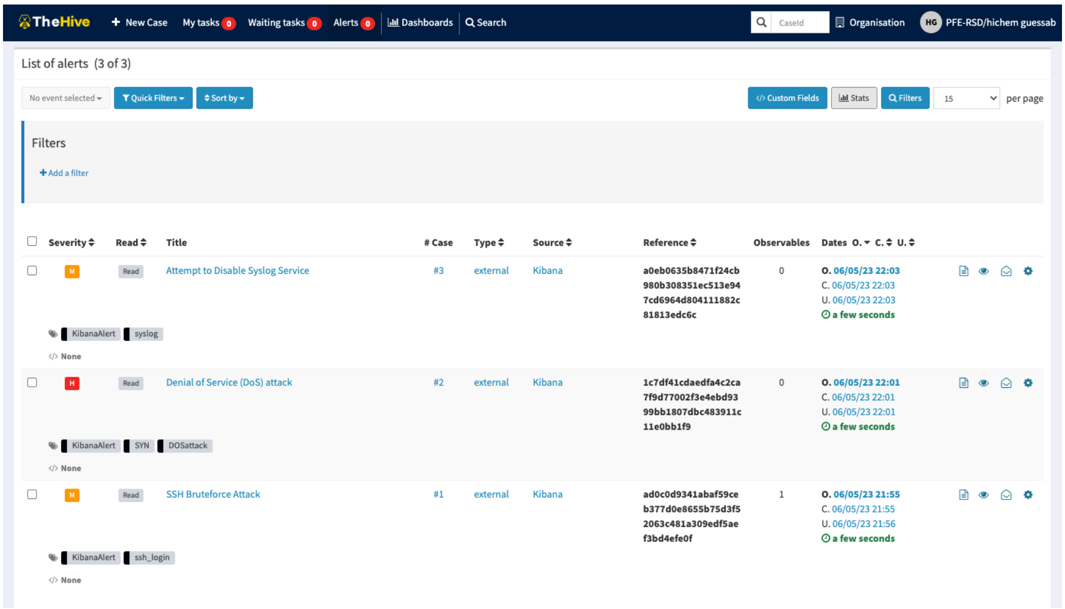
Count — Stack by signal.rule.name

signal.rule.name	Count
Attempt to Disable Syslog Service	1
Denial of Service (DoS) attack	1
SSH Bruteforce Attack	1

Actions	@timestamp	Rule	Severity	Risk Score	Reason	host.name
	Jun 5, 2023 @ 22:03:28.347	Attempt to Disable Syslog ...	medium	47	process event with process systemctl, by root on debian created medium a...	debian
	Jun 5, 2023 @ 22:01:08.793	Denial of Service (DoS) atta...	high	74	event on debian created high alert Denial of Service (DoS) attack.	debian
	Jun 5, 2023 @ 21:55:29.150	SSH Bruteforce Attack	medium	47	authentication event with process sshd, by macbookair on debian created ...	debian

Figure 3.38: Kibana security alerts

After the detection of the attacks , alerts have been created in TheHive based on the detections :



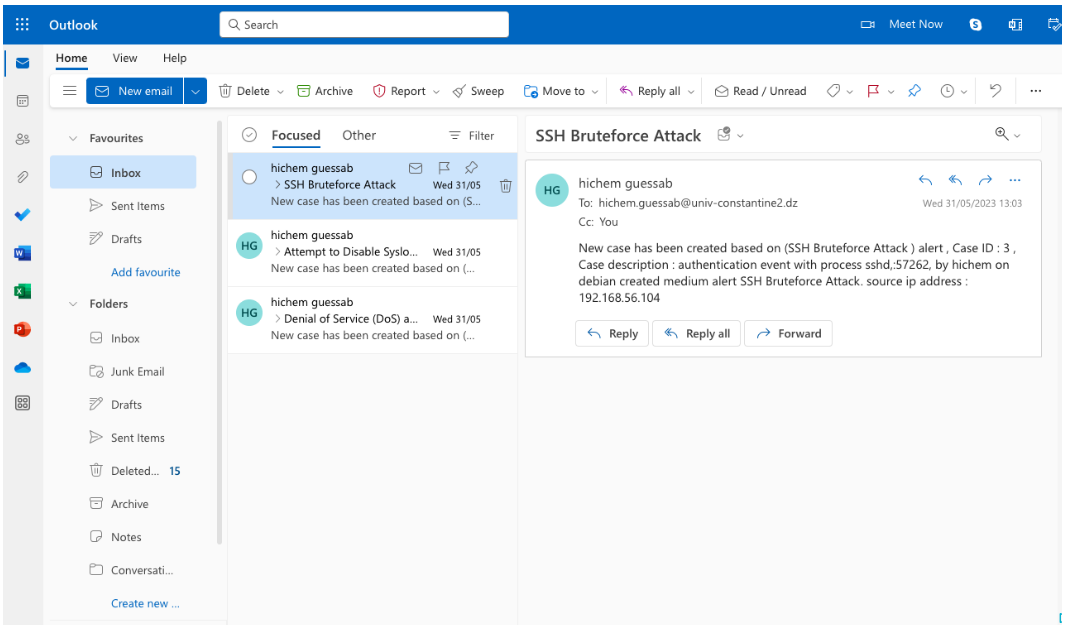
List of alerts (3 of 3)

Filters

□	Severity	Read	Title	# Case	Type	Source	Reference	Observables	Dates O. C. U.
□	M	Read	Attempt to Disable Syslog Service KibanaAlert syslog None	#3	external	Kibana	a0eb0635b8471f24cb 980b308351ec513e94 7cd6964d804111882c 81813edc6c	0	O. 06/05/23 22:03 C. 06/05/23 22:03 U. 06/05/23 22:03 a few seconds
□	H	Read	Denial of Service (DoS) attack KibanaAlert SYN DOSattack None	#2	external	Kibana	1c7df41cdaedfa4c2ca 7f9d77002f3e4ebd93 99bb1807dbc483911c 11e0bb1f9	0	O. 06/05/23 22:01 C. 06/05/23 22:01 U. 06/05/23 22:01 a few seconds
□	M	Read	SSH Bruteforce Attack KibanaAlert ssh_login None	#1	external	Kibana	ad0c0d9341abaf59ce b377d0e8655b75d3f5 2063c481a309edf5ae f3bd4efe0f	1	O. 06/05/23 21:55 C. 06/05/23 21:55 U. 06/05/23 21:56 a few seconds

Figure 3.39: TheHive list of alerts

Additionally, Even if The Soc analysts are not physically present in front of the monitoring system, they will receive instant email notifications whenever an attack is detected. This enables them to take immediate action to mitigate the potential impact of the detected attacks



hichem guessab
> SSH Bruteforce Attack Wed 31/05
New case has been created based on (S...

HG hichem guessab
> Attempt to Disable Syslo... Wed 31/05
New case has been created based on (...

HG hichem guessab
> Denial of Service (DoS) a... Wed 31/05
New case has been created based on (...

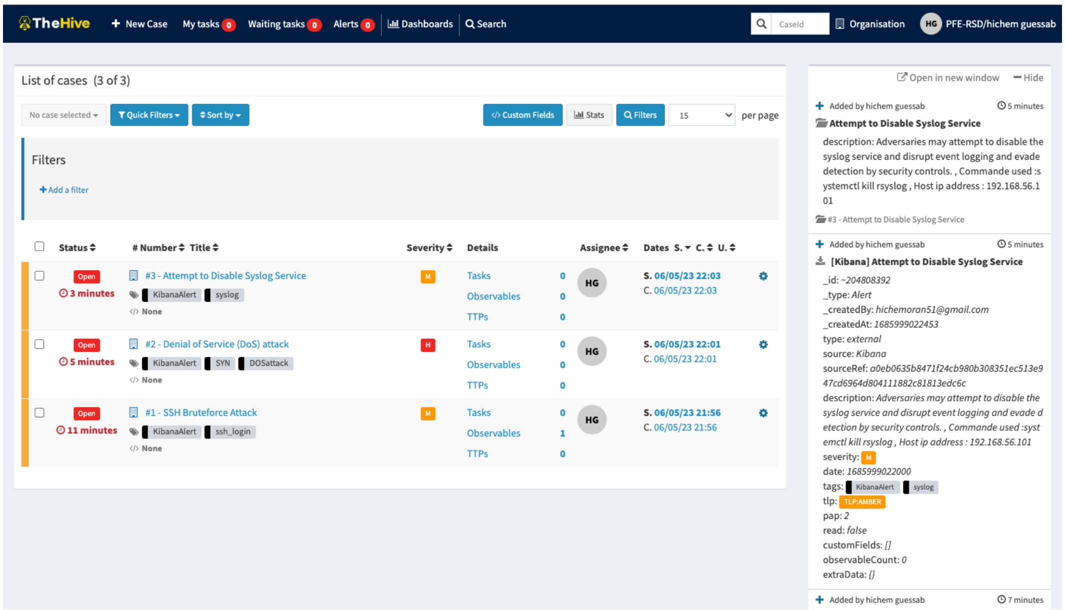
SSH Bruteforce Attack

HG hichem guessab
To: hichem.guessab@univ-constantine2.dz
Wed 31/05/2023 13:03
Cc: You

New case has been created based on (SSH Bruteforce Attack) alert , Case ID : 3 , Case description : authentication event with process sshd,:57262, by hichem on debian created medium alert SSH Bruteforce Attack. source ip address : 192.168.56.104

Figure 3.40: Notification sent

After the creation of the alerts , cases have been created in TheHive based on these alerts :



List of cases (3 of 3)

Filters

+ Add a filter

□	Status	# Number Title	Severity	Details		Assignee	Dates S. C. U.
□	Open 3 minutes	#3 - Attempt to Disable Syslog Service KibanaAlert syslog None	M	Tasks Observables TTPs	0 0 0	HG	S. 06/05/23 22:03 C. 06/05/23 22:03
□	Open 5 minutes	#2 - Denial of Service (DoS) attack KibanaAlert SYN DOSattack None	H	Tasks Observables TTPs	0 0 0	HG	S. 06/05/23 22:01 C. 06/05/23 22:01
□	Open 11 minutes	#1 - SSH Bruteforce Attack KibanaAlert ssh_login None	M	Tasks Observables TTPs	0 1 0	HG	S. 06/05/23 21:56 C. 06/05/23 21:56

+ Added by hichem guessab 5 minutes

Attempt to Disable Syslog Service

description: Adversaries may attempt to disable the syslog service and disrupt event logging and evade detection by security controls. , Commande used :s ystemctl kill rsyslog , Host ip address : 192.168.56.1 01

#3 - Attempt to Disable Syslog Service

+ Added by hichem guessab 5 minutes

[Kibana] Attempt to Disable Syslog Service

_id: *~204808392*
_type: *Alert*
_createdBy: *hichemoran51@gmail.com*
_createdAt: *1685999022453*
type: *external*
source: *Kibana*
sourceRef: *a0eb0635b8471f24cb980b308351ec513e9 47cd6964d804111882c81813edc6c*
description: *Adversaries may attempt to disable the syslog service and disrupt event logging and evade d etection by security controls. , Commande used :syst emctl kill rsyslog , Host ip address : 192.168.56.101*
severity: M
date: *1685999022000*
tags: KibanaAlert syslog
tlp: TLP:AMBER
pap: *2*
read: *false*
customFields: *[]*
observableCount: *0*
extraData: *{}*

+ Added by hichem guessab 7 minutes

Figure 3.41: TheHive list of cases

Conclusion

In conclusion for this chapter, leveraging Shuffle as a SOAR platform, along with TheHive, Cortex, and MISP, has empowered our SOC to streamline incident response, enhance collaboration, and strengthen our overall cybersecurity capabilities. By integrating these tools, we have achieved efficient threat detection, investigation, and mitigation, while effectively protecting our critical assets. This SOC framework enables us to proactively address evolving security challenges and maintain a robust defense against cyber threats.

General Conclusion

In conclusion, The establishment of a Security Operations Center (SOC) has significantly enhanced the cybersecurity of our university. Through research and analysis, we identified the key components for a successful SOC and selected open-source tools like Elastic Search and TheHive for their versatility. These tools enabled us to actively monitor, detect, and respond to security incidents promptly, leveraging advanced technologies such as threat intelligence platforms and real-time incident response systems. Our SOC implementation has created a culture of resilience, pro-activity, and early incident addressing, protecting our valuable assets.

This thesis outlines the steps undertaken to achieve the desired outcome. Initially, we conducted an assessment of the existing requirements and provided an overview of the project's general context.

In the pursuit of our SOC implementation, we carefully followed this steps that included strengthening the operating system and setting up security rules. By installing and configuring the necessary tools with great care, we established a strong foundation for our SOC to operate on. Our meticulous attention to detail and commitment to best practices have improved the effectiveness and reliability of our SOC.

As we reach the end of this transformative journey, we take pride in the remarkable results we have achieved. Our SOC demonstrates our strong commitment to cybersecurity, positioning our university as a leader in higher education security. By embracing new technologies, enhancing our incident response capabilities, and fostering a culture of continuous learning and improvement, our SOC will continue to grow and adapt to the ever-changing threat landscape.

Future Prospects

In the dynamic world of cybersecurity, continuous updates to our Security Operation Center (SOC) are essential. By staying current with new tools and creating new rules, we strengthen our defense against evolving attacks, ensuring the security of our assets. Our commitment to ongoing improvement and adaptation empowers us to stay ahead of potential threats and maintain a robust cybersecurity posture.

Bibliography

[1] Lincoln college to close, hurt by pandemic and ransomware attack. *the NewYork times*. URL https://www.nytimes.com/2022/05/09/us/lincoln-college-illinois-closure.html.

[2] What is a security operations center (soc). *IBM*. URL https://www.ibm.com/topics/security-operations-center.

[3] What is a soc and why it matters for security. *Cybriant*. URL https://cybriant.com/what-is-a-soc-and-why-it-matters-for-security/.

[4] Soc processes, operations, challenges, and best practices. *Sapphire*, 2023. URL https://www.sapphire.net/security/soc-processes/.

[5] Online course mgt551: Building and leading security operations centers. *SANS*. URL https://www.sans.org/cyber-security-courses/building-and-leading-security-operations-centers/.

[6] Saif Gunja. What is mttr? how mean time to repair helps define devops incident management. *dynatrace*, 2022. URL https://www.dynatrace.com/news/blog/what-is-mttr/.

[7] Jad Hamdouch and Ismael Bouarfa. What is the elk stack and how to apply it to information security. *medium*, 2019. URL https://medium.com/jad-ismael/what-is-the-elk-stack-and-how-to-apply-it-to-information-security-38a91088b0e6.

[8] A 4-in-1 security incident response platform. *TheHive*. URL https://thehive-project.org/.

[9] Misp - open source threat intelligence platform. *circl.lu*. URL https://www.circl.lu/services/misp-malware-information-sharing-platform/.

[10] Sharon Shea. What is soar? *techtarget*, 2023. URL `https://www.techtarget.com/searchsecurity/definition/SOAR`.

[11] Azouman Bertin Tougma. Shuffle, le soar open source. *conix*, 2022. URL `https://www.conix.fr/shuffle-le-soar-open-source/`.

Acronyms

SOC Security Operations Center

ISO International Organization for Standardization

DLP Data loss prevention

SIEM Security Information and Event Managemen

ELK Elasticsearch, Logstash, and Kibana.

MISP Malware Information Sharing Platform

IR Incident Response

IP Internet Protocol

SOAR Security Orchestration, Automation and Response

IDS Intrusion Detection Systems

IPS intrusion prevention system

DOS Denial of Service

ISOC information security operations center

IT Information technology

EDR Endpoint Detection and Response

NTA Network Traffic Analysis

MTTD Mean Time to Detect

MTTR Mean Time to Respond

FCR First Contact Resolution

JSON JavaScript Object Notation

REST Representational State Transfer

LOG The automatically produced and time-stamped documentation of events relevant to a particular system

CERT Computer Emergency Response Team

CSIRT Computer Security Incident Response Team

PGP Pretty Good Privacy

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

URL Uniform Resource Locator