

POLITECNICO DI TORINO

Master's Degree in Computer Science Engineering



MASTER's Degree Thesis

**Study on Implementation and
Optimization of Security Operation
Center Using Open-source Tools**

Supervisors:

Prof. FULVIO VALENZA
Dott. ALBERTO SCARPA

Candidate:

ROBERTO FERRAREIS

December 2024

Contents

Acronyms	IV
Summary	VIII
1 Introduction	1
2 Background Context	4
2.1 QiNet’s SOC Zero Trust Architecture	5
2.2 SOC regulatory elements	6
2.2.1 ENISA Framework	6
2.2.2 SOC Maturity Model	7
3 Security Operation Center	10
3.1 Definition	10
3.1.1 Organizational structure of the SOC	12
3.2 Architecture and Tools	14
3.2.1 SIEM	14
3.2.2 SOAR	16
3.2.3 Data sources on-boarding	17
3.3 Open source context	18
3.3.1 Licensing	19
3.3.2 Elastic Stack	20
3.3.3 The Hive Project	24
3.3.4 Netbox	27
3.3.5 Information Gathering and OSINT Tools	28
4 Security Incident Management Process	34
4.1 Alert Data Sources	36

4.2	Alert Pre-Evaluation	37
4.3	Alert Evaluation	39
4.4	Incident Classification and Evaluation	41
4.5	Impact Evaluation	44
4.6	Abort Incident Management	46
4.7	Authorized Activity Evaluation	48
4.8	Alert Fixing and Tuning Procedure	50
4.9	Incident Response Plans	51
4.10	Examples: Alert Analysis	55
4.10.1	Endpoint Detection PUP Alert	55
4.10.2	Alert Management Process for Suspicious File Execution	57
4.10.3	Failed Multi-Factor Authentication for User Alert	58
4.10.4	Suspicious URL Click Alert	59
4.10.5	Stolen or Lost Device Management	61
4.10.6	Unfamiliar Sign-In Alert Management	63
5	SOC Implementation and Simulation	67
5.1	How to implement a SOC	67
5.2	Simulation Description and Purpose	68
5.3	Simulation Tools	68
5.3.1	Primary Tools	69
5.3.2	Supporting Tools	73
5.4	Simulation Setup	74
5.5	Base Handling Process	77
6	Optimizations in SOC operations	79
6.1	Development	79
6.2	Optimized Handling Process	82
6.3	Performance Metrics and Key Indicators for SOC Evaluation	83
7	Conclusions and Future Works	87
	Bibliography	90

List of Tables

3.1	List of OSINT Tools and Their Usage	33
6.1	Comparison of Metrics Before and After Optimization (in minutes) .	86

List of Figures

2.1	ENISA Framework phases to setup a SOC and CSIRT	6
2.2	SOC Maturity Levels	9
3.1	Logical SOC components and their relation	11
3.2	SOC Tiers and their functions	14
3.3	Elastic architecture	21
4.1	SOC Workflow Summarized	36
4.2	General phases of an Incident Response Plan	51
5.1	Basic SOC Implementation	68
5.2	Detection Rules Samples in Kibana	71
5.3	Output Stage in <code>logstash.conf</code> File	72
5.4	Script to configure Elasticsearch user automatically	75
5.5	Cortex successful integration with TheHive	76
5.6	Initial Discover section in Kibana	78
6.1	Description and Kibana link in TheHive automatic case generation	81
6.2	Cortex Analyzers configuration	81
6.3	Email Notifications to analyst	82
6.4	Analyzers Test Execution Results on a public IP	83

Acronyms

CISO	Chief Information Security Officer.	4
CMDB	Configuration Management Database.	27
CSIRT	Computer Security Incident Response Team.	6
CVE	Common Vulnerabilities and Exposures.	31
ECS	Elastic Common Schema.	22
EDR	Endpoint Detection and Response.	4
FSF	Free Software Foundation.	18
GDPR	General Data Protection Regulation.	6
IAM	Identity and Access Management.	39
IOC	Indicator of Compromise.	26
IRP	Incident Response Plan.	34
KPI	Key Performance Indicator.	7
KQL	Kibana Query Language.	23
MTTA	Mean Time To Acknowledge.	38
MTTC	Mean Time To Contain.	84
MTTI	Mean Time To Investigate.	84
MTTR	Mean Time To Respond.	84

NOC Network Operation Center. 4

ORP Organizational Response Plan. 34

OSI Open Source Initiative. 18

OSINT Open Source Intelligence. 2

OSS Open Source Software. 18

PAP Permissible Actions Protocol. 43

RASCI Responsible Accountable Supported Consulted Informed. 11

SIEM Security Information and Event Management. 1

SOAR Security Orchestration, Automation and Response. 1

SOC Security Operation Center. VIII

TLP Traffic Light Protocol. 17

ZTA Zero Trust Architecture. 5

Summary

In the dynamic world of networking and data management, the security of sensitive information—whether in business or individual privacy contexts—is constantly at risk. This is due to the multitude of security domains, varying trust levels, and diverse IT tools that are often targeted by malicious actors aiming to exploit weaknesses, disrupt business operations, and compromise confidentiality.

In particular, to achieve a higher security posture, a company may establish a Security Operation Center (SOC) to monitor both internal and external IT infrastructures. The goal of an SOC is to implement proactive and reactive solutions to manage cybersecurity incidents that may impact organizations and the SOC itself. This thesis explores the strategies, workflows, and tools required to effectively implement an SOC. A comprehensive analysis is conducted on an existing, fully operational SOC within *QiNet*, a company that provides SOC services to both national and international organizations. Furthermore, an in-depth examination is performed to determine how open-source tools can be integrated to build an effective SOC, with detailed reasoning provided for the selection of specific tools based on the needs of the SOC and its operators.

The initial phase of this study focuses on a comprehensive analysis of the current SOC at *QiNet*. This includes documenting and mapping SOC workflows to identify strengths, weaknesses, and areas for improvement.

Finally, this work also explores the implementation of automated operations to aid SOC operators by enhancing response workflows with faster handling times and by providing accurate information associated with security incidents. The study aims to leverage open-source technologies and targeted strategies to improve cybersecurity resilience within business organizations.

Chapter 1

Introduction

In the dynamic and evolving landscape of the digital world, the security of various systems, each with unique specifications, has become increasingly critical. Cyber threats continue to grow in complexity and sophistication, targeting both personal and business contexts where data privacy holds particular importance. As organizations increasingly depend on various network architectures, cloud solutions, and IoT devices, the vulnerability of their infrastructures becomes higher, providing extensive opportunities for malicious entities to exploit. This rapidly changing environment demands robust and adaptable security solutions to protect systems and sensitive information effectively.

In response to these challenges, organizations increasingly rely on SOCs as central entities for monitoring and managing their IT security. A SOC serves as the backbone of a company's cybersecurity posture, enabling the detection and response to security incidents across internal infrastructures and external environments managed for business customers. Including in its staff professionals with varying skill levels and areas of expertise, SOC teams operate under the guidance of management roles, coordinating efforts to deliver a broad range of security services. This study explores the structure, functionality, and workflows of a fully operational business-oriented SOC to understand how it achieves its objectives of enhancing organizational security.

A SOC's operations rely heavily on specific tools that form the core of its functionality. The primary tools, Security Information and Event Management (SIEM) and Security Orchestration, Automation and Response (SOAR), work in cooperation to facilitate workflows. These workflows begin with the collection and analysis

of logs from monitored endpoints, transforming them into actionable alerts. In the worst-case scenarios, these alerts culminate in incident management processes, where security response teams implement mitigation measures or solutions adjusted to the specific taxonomy of the incident. This thesis examines the integration and utilization of these tools, highlighting their interconnectedness and their role in enabling SOC teams to respond effectively to security threats.

To achieve cost-effectiveness, flexibility, and customization, the study also explores the feasibility of building a SOC using open-source tools. Open-source solutions provide several advantages, including adaptability and the ability to tailor operations to specific organizational needs. This research delves into the benefits and challenges of implementing a SOC with open-source tools, including the use of Open Source Intelligence (OSINT) for enriching data associated with incidents. Furthermore, the study explores the automation of OSINT processes through scripting and integration with existing tools, enhancing efficiency and accuracy.

The latter part of the study focuses on the practical implementation of a SOC using the tools described in earlier analyses. This implementation demonstrates the feasibility of establishing a fully functional SOC capable of operating effectively in business contexts and integrating seamlessly with existing systems. Additionally, the study evaluates the optimization of SOC operations through Python scripts and specialized libraries. These optimizations are measured against key SOC performance indicators to assess their impact.

The study concludes by demonstrating the effectiveness of the implemented SOC optimizations and exploring potential paths for further improvements. These include new automation processes, customization options, and addressing emerging challenges in the cybersecurity landscape. By providing a comprehensive analysis of SOC workflows, tools, and optimizations, this thesis aims to contribute valuable insights into the evolving field of IT security.

The work shown in this thesis project is organized considering the following structure:

- **Chapter 2 - Background Context:** This chapter presents the company involved in this work and the applications context of QiNet's SOC, implementation frameworks and introductory definitions for its core conceptual elements.

- **Chapter 3 - Security Operation Center:** This chapter introduces main SOC tools used in a SOC and how they are included in the SOC architecture. It explores selected open source tools used by the company SOC and in the project implementations. Some minor tools are also here analyzed.
- **Chapter 4 - Security Incident Management Process:** This chapter follows the whole SOC management workflow considering every cases and here also some real life examples are reported.
- **Chapter 5 - SOC Implementation and Simulation:** This chapter introduces and describes a simulation to implement a SOC and its operating processes. All the tools, scripts and configurations performed to set up a SOC are described.
- **Chapter 6 - Optimizations in SOC operations:** In this chapter, all the automation optimizations are described through the operation of the various scripts. In addition, the measurements performed to be able to compare the performance of the SOC are commented here.
- **Chapter 7 - Conclusions and Future Works:** In this last chapter, the various results of the study are commented and some insights into potential future studies are introduced.

Chapter 2

Background Context

QiNet S.p.a., a relevant company part of the Impresoft Group whose mission focuses on digital transition of its clients' business processes, operates in networking and cybersecurity sectors with the goal of improving safety and informational flows of its customers. Its main provided services are Network Operation Center (NOC) and SOC, which focus on the monitoring, threat detection and incident response to protect customer assets from cyber threats.

In the SOC environment, QiNet is structured in teams of highly skilled professionals who work 24/7, both remotely scattered throughout the country and on-site. Each team monitors, identifies and responds to unusual behaviors and security incidents by conducting precise analysis on security alerts whenever these are triggered. Thanks to this kind of organization, which is efficiently coordinated by the Chief Information Security Officer (CISO), the SOC is constantly able to handle real-time threats and incidents, ensuring customer business continuity and data protection. QiNet operates both domestically and internationally, considering different types of organizations that may require a SOC-as-a-service, which typically vary from medium to large companies. As a company, it is continuously expanding and integrating different types of state-of-the-art technologies to deliver its services effectively, considering a highly dynamic and evolving area.

QiNet, in the area of cybersecurity and SOC context, provides different incident response, monitoring and assessment services, including:

- Monitoring for alerts generated from security solutions such as firewalls, End-

point Detection and Response (EDR) and SIEM;

- Engagement point monitoring, several of them can be used to contact SOC teams to handle potential security incidents;
- Security events analysis and potential impact evaluation;
- Mitigation actions, following potential confirmed attacks and security incidents;
- Vulnerability assessments, to identify and report customers security flaws;
- Penetration testing, both internally and externally to evaluate security level of the tested system;
- Cybersecurity assessment, aimed to enhance security posture of the customer;
- Threat hunting, includes proactive operations in order to identify potential hidden cyber threats;

The combination of SOC services is designed to improve the overall security of a target IT infrastructure, including every asset and actor operating in it.

2.1 QiNet's SOC Zero Trust Architecture

QiNet's SOC is described by Zero Trust Architecture (ZTA) principles which ensure that no entity, whether users, softwares and devices, is able to effectively access systems and services by default but periodical authentication processes must be carried out and role-based authorization determines whether to grant access or deny it. Each access request must comply to preconfigured rules and protocols in order to allow users to only access specific tools, tenants managed by QiNet and infrastructure sections.

In a ZTA context and in the remote work environment previously introduced, QiNet employs a VPN solution as a secure access gateway solution used to enforce security policies before accessing cloud applications, SOC tools and infrastructures. The VPN is also used to implement conditional access policies that allow users to successfully access specific resources based on location and other factors.



Figure 2.1: ENISA Framework phases to setup a SOC and CSIRT

2.2 SOC regulatory elements

QiNet also provides its services maintaining high standards of compliance. In fact, it holds certifications for ISO:27001 (Information Security Management) and ISO:9001 (Quality Management). It also handles data privacy, being fully compliant with General Data Protection Regulation (GDPR) requirements.

2.2.1 ENISA Framework

QiNet's SOC follows a framework to set up SOC and Computer Security Incident Response Team (CSIRT) released by ENISA which is the European center of cybersecurity expertise. The framework emphasizes a results-driven approach, guiding organizations through establishing and improving their CSIRT or SOC, from initial assessment to continuous operation and refinement. It is designed to enhance incident response capabilities and build an effective cybersecurity ecosystem within an organization [11].

The ENISA framework structures the SOC setup following some key phases which are here summarized and shown in Figure 2.1:

1. *Assessment for Readiness*: The first phase evaluates the organization's need for a SOC, identifying stakeholders, and establishing a preliminary operational target. It includes creating a governance structure, identifying the host organization, and setting up a high-level roadmap and budget.
2. *Design*: During the design phase, detailed plans for services, processes, workflows, technologies, and organizational structures are developed. The design should ensure that the SOC aligns with the target, resources, and security needs of the organization.

3. *Implementation*: This phase involves putting the designed structure into practice by hiring staff, setting up processes and technologies. This also includes developing IT and security management procedures and agreements with stakeholders.
4. *Operations*: Once implemented, the SOC begins its operational phase. This includes measuring Key Performance Indicator (KPI), handling security incidents, reviewing performance periodically, and adjusting services based on stakeholder evolving needs.
5. *Improvement*: Continuous improvement is essential for adapting to new threats. This phase focuses on collecting feedbacks, prioritizing improvement initiatives, and implementing updates to the SOC processes and technologies.

2.2.2 SOC Maturity Model

In general, the maturity of a SOC can be assessed by following several models. A level explains what the SOC can effectively offer in its service, and each level can introduce new enhancements but also has its drawbacks that are solved by higher maturity levels, as it is shown below. There are several models that can be used to assess the maturity of a SOC; here is presented one of them:

1. **Security Perimeter**: At this level, the SOC primarily focuses on reactive monitoring and incident response. It mainly deals with incidents that have already occurred, applying specific remediation measures manually. The main focus is on maintaining security at the perimeter, with limited visibility into ongoing threats.

While this level can be a good starting point, it presents significant drawbacks. The manual response process can lead to slower reaction times, increasing the risk of more extensive damage from incidents. Additionally, the SOC may struggle to identify advanced or coordinated attacks due to a lack of proactive monitoring.
2. **SIEM**: In Level 2, the SOC begins to employ a SIEM system. This involves the collection of logs from various sources, allowing for the storage, aggregation, and correlation of data. The aim is to facilitate incident analysis and

improve response strategies. At this stage, EDR solutions are also integrated, enhancing the SOC's ability to detect and respond to threats at the endpoint level.

This level offers greater visibility into network activities and real-time threat identification. However, it also requires specialized skilled staff to manage the SIEM and interpret the data accurately. A common challenge at this level is dealing with false positives, which may overwhelm the team.

3. **Security Automation:** Level 3 introduces the integration of SOAR tools.

These tools enable the automation of incident management processes, allowing the SOC to reduce the time required to respond to incidents. By implementing playbooks for various security scenarios, the SOC can automate repetitive tasks and help analysts to focus on more complex issues.

This increased efficiency and speed in incident response are clearly advantages. However, organizations must invest in the right infrastructure and expertise to effectively implement automation. Additionally, there is a risk of becoming overly reliant on automation, potentially neglecting the critical human analysis that can identify particular kind of threats.

4. **Advanced Analytics:** At this stage, the SOC adopts advanced analytics

techniques, such as machine learning and AI, to enhance its threat detection capabilities. This includes conducting forensic investigations to analyze incidents and learn from past situations. Predictive analysis becomes a key focus, allowing the SOC to anticipate future threats based on historical data and behavioral patterns.

The implementation of advanced analytics provides a better context for security decisions and the ability to identify unknown threats. However, it necessitates specialized skills in data analysis and machine learning, as well as significant investments in advanced technologies, which can add complexity and cost.

5. **Predictive Analytics:** In the highest maturity level, the SOC employs pre-

dictive analytics to identify potential attack patterns before they occur. By analyzing historical data and recognizing behavioral trends, the SOC becomes

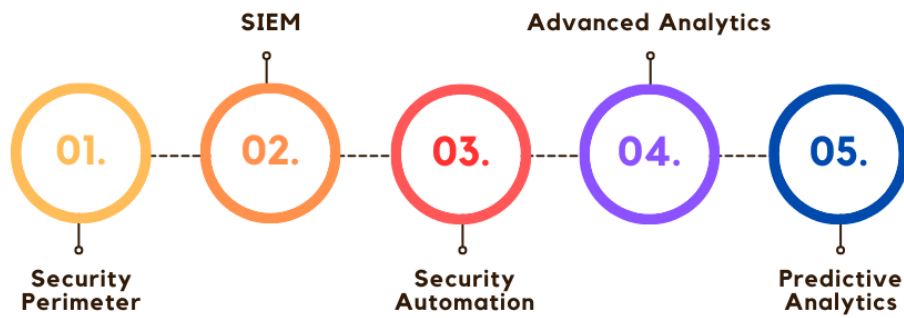


Figure 2.2: SOC Maturity Levels

more proactive, shifting its focus to prevention against threats. This may involve conducting attack simulations and penetration tests to assess and strengthen security defenses.

The ability to prevent attacks before they materialize significantly reduces potential damage and associated costs. However, achieving this level requires substantial investments in technology and training, alongside the complexities involved in managing predictive information.

Chapter 3

Security Operation Center

3.1 Definition

A **SOC** is an centralized control unit which responsibility is to monitor a specific IT infrastructure, that may be both internal and external, and protect it against multiple taxonomies of cyber threats. Its main mission is to implement proactive and reactive measures against security incidents that may compromise the CIA triad (Confidentiality, Integrity and Availability) of the infrastructure.

The **CIA triad** is composed by three essential security properties that must be achieved by a a system:

- *Confidentiality*: This means that data can only be understood and accessed by actors that have the right to do so. Any other actor must be prevented from accessing and understand protected and sensitive data. Typically, in a secure system or protocol, confidentiality is achieved through encryption techniques or access restriction based on roles and permissions.
- *Integrity*: Integrity refers to the ability to ensure that data is not changed by any unauthorized entity and unexpected change to data can be promptly detected. This is usually achieved through hash functions, which involves creating unique strings to identify data and documents, or digital signature, which allows to achieve authenticity and changes detection.
- *Availability*: Data, applications and services can always be accessed by any actor even in case of system failures, security incidents and compromises. Redundancy techniques are usually used to achieve this property as well as fault tolerance techniques.



Figure 3.1: Logical SOC components and their relation

The SOC logically consists of three basic components provided and described by the ENISA framework:

- **Processes:** In a SOC, processes are strategies defined to monitor the infrastructure, ensure proper management, analysis, and resolution of security incidents. The SOC can provide incident management processes if the customer does not have any preexisting ones.
- **People:** Teams of security analysts working 24/7 to monitor and manage what is happening within the corporate IT infrastructure perimeter. Clearly defined roles and responsibilities are critical to specifically define who is responsible for what operation, following Responsible Accountable Supported Consulted Informed (RASCI) definitions.
- **Technologies:** Tools used to identify threats, information about them, perform analysis, and determine possible strategies, including automated ones, for response and remediation.

The order in which the previous components are presented is critical as technology is developed and used to assist people working to achieve the goal by applying specific processes. A SOC is typically characterized by constant processes that allow improvements to be implemented and controls to be placed on the various workflows. The processes, in the context under analysis, are continuously updated to address new threats and best practices. In addition, it is critical for a SOC to invest in new technologies and constantly update the tools used to maintain a high level of security.

3.1.1 Organizational structure of the SOC

A best-practice SOC is structured into multiple tiers to optimize the analysis and response to security incidents based on their severity, with distinct roles and responsibilities at each level. Depending on the scope and size of the SOC, different teams and numbers of personnel are needed, but core roles typically include analysts at various tiers (L1, L2, CSIRT/L3) as well as dedicated managers [12]. Each tier is responsible for specific functions, from real-time monitoring and triage at the L1 level to advanced incident analysis and coordination at L2, and critical incident management and forensic investigation within the CSIRT. Dedicated managers oversee the entire incident response process and ensure that operations run smoothly, forming a comprehensive and effective cybersecurity defense strategy. Here details about each tier's responsibility follow:

- **SOC Tier 1 - L1:** L1 team acts as the front line of defense within the SOC, responsible for continuous, real-time monitoring of security events and performing initial triage. L1 analysts are tasked with identifying, classifying, and prioritizing security alerts based on their severity. They handle low-severity incidents, addressing and resolving routine security issues, such as phishing attempts or malware detections. If an L1 analyst determines that the incident is more critical or complex because it can have a significant impact on people and assets, it is escalated to the L2 team for further analysis. The L1 team is essential in ensuring that potential threats are detected and responded to promptly, and they serve as the first point of contact for incident management within the SOC.
- **SOC Tier 2 - L2:** The L2 team handles escalated incidents that require more advanced analysis and a deeper understanding of cybersecurity threats. L2 analysts conduct in-depth investigations, identifying the root cause of incidents, and implementing remediation strategies. They also coordinate incident response efforts, working closely with the L1 team and other stakeholders to ensure that threats are properly contained and mitigated. Additionally, the L2 team is responsible for consulting on preventive actions and managing threat intelligence, using data from both internal and external sources to improve the

organization's overall security posture. Their role is critical in handling incidents that exceed the capabilities of the L1 team, providing more specialized and technical expertise.

- **SOC Tier 3 - L3:** The L3, often referred to as the CSIRT, is responsible for managing the most severe and complex security incidents. This team typically consists of highly skilled cybersecurity professionals who specialize in incident response, digital forensics, and threat hunting. The CSIRT often works in collaboration with law enforcement, regulatory authorities, and external stakeholders, especially when handling incidents that involve data breaches, advanced persistent threats, or other critical events. Their work includes forensic investigations to determine how an attack occurred, the extent of damage, and ensuring the organization recovers from the incident. The CSIRT team plays a crucial role in mitigating the impact of significant cyber events and helping to protect the organization from future attacks by identifying and addressing system vulnerabilities and general weaknesses.

The whole organizational structure of the SOC is overseen by the **CISO**. The CISO is the executive leader responsible for the organization's entire cybersecurity strategy. As the top security authority, the CISO ensures that all security policies, procedures, and technologies align with business objectives and regulatory requirements. They manage the SOC, providing strategic direction and ensuring that teams are well-equipped to handle every threat taxonomy. The CISO's role also involves communicating the organization's security posture to the executive board and external stakeholders, translating complex security risks into business terms. In addition to managing security operations, the CISO is responsible for risk management, incident response coordination, and driving initiatives to spread a proactive security culture and awareness within the organization. The CISO ensures that the organization is resilient against cyber threats and capable of mitigating risks. Ultimately, the CISO is the authority that coordinates the activity of the previously mentioned teams at each level.

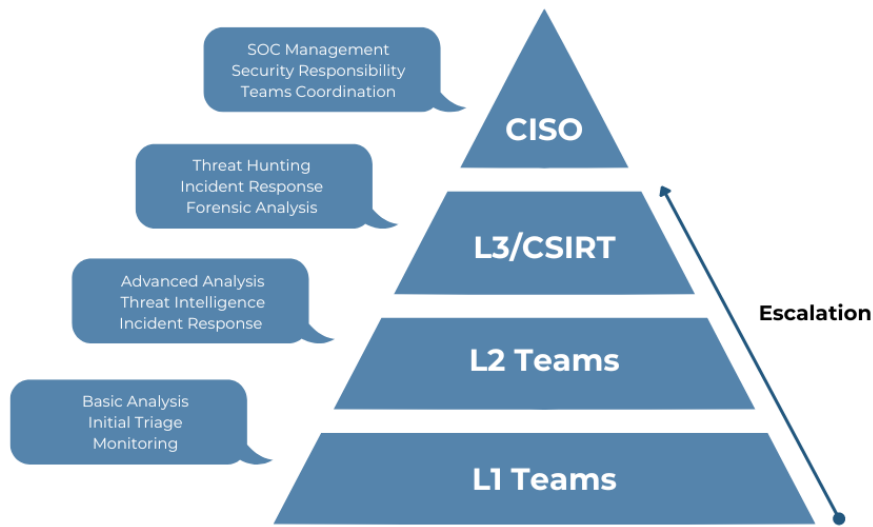


Figure 3.2: SOC Tiers and their functions

3.2 Architecture and Tools

3.2.1 SIEM

A **SIEM**, is a solution designed to provide a centralized view of IT security within an organization. The SIEM collects, aggregates, correlates and assists in the analysis of security-related logs and data from a variety of sources, such as servers, network devices, applications and other IT infrastructure. This process allows potential security threats to be identified and responded to more quickly and efficiently.

The main role of a SIEM system lies in its ability to aggregate data from a variety of sources. These log data, which can include access events, system errors and other types of activity, are normalized and stored in a common format. This normalization is crucial because it allows events from different systems to be analyzed and compared in a uniform manner. Analyzing logs from different sources can be complex because each source has a specific format for representing and communicating logs. Automation rules must be updated periodically based on the types of sources being on-boarded or off-boarded, or based on new instances that may occur.

One of the main functions of a SIEM is real-time analysis. The SIEM can analyze millions of events per second, identifying patterns of abnormal behavior that could indicate malicious activity. For example, the system might detect abnormal access to a server from an unusual location or repeated failed login attempts that could indicate a brute force attack. When a suspicious event is identified, SIEM can generate alerts and notifications, alerting SOC analysts. These alerts can be classified according to their severity, allowing analyst teams to prioritize their responses. In addition, a SIEM can offer the ability to implement incident response automation capabilities, such as automatically disconnecting a suspicious user or isolating a compromised device.

The SIEM not only detects anomalies but also provides tools for incident management. Although the SOC is agnostic to the cybersecurity tools already implemented and used in the various managed IT infrastructures, data is always extracted on the SIEM imposed by the SOC. This provides a centralized and coordinated view of security, regardless of the specific technologies used.

In addition to incident management, the SIEM offers forensic reporting and analysis tools. Analysts can use SIEM to examine historical logs in detail and reconstruct past security events. This is particularly useful for understanding the cause of an incident, assessing the impact and improving future security measures.

Correlation rules

An important feature handled by SIEM systems is the use of correlation rules. Correlation rules, in this context, are predefined logic-based conditions used to detect complex security incidents by performing correlations between multiple events across different sources. These rules can combine various types of data, such as firewall logs, authentication logs or network traffic, to identify suspicious patterns or behaviors that individual logs may not reveal. The primary objective of correlation is to enhance the detection accuracy and reduce false positives by considering the context of events across the network.

For example, a correlation rule might be configured to detect a *brute force attack* by correlating multiple failed login attempts from the same IP address within a short time frame, followed by a successful login. On their own, failed login attempts and a

successful login might not indicate a security issue, but when correlated, they could suggest malicious activity.

$$\sum_{i=1}^N 1(\text{failed login from same IP within } t) \geq k \quad (3.1)$$

$$\exists \text{successful login from same IP within } t \quad (3.2)$$

where N is the total number of login attempts, 1 is the indicator function to count login attempts, k is the threshold for the number of failed login attempts and t is a specified time window.

In this example case, if conditions (3.1) and (3.2) are true, then the brute force alert is generated and sent to SOC analysts for further analysis.

3.2.2 SOAR

The SOAR platform in the SOC is a set of tools and technologies designed to help organizations manage and respond to security incidents in a coordinated and efficient manner across analysts and other team members. SOAR integrates several capabilities that enhance the detection, analysis, response, and management capabilities of security incidents.

SOAR offers the ability to automate many of the repetitive and manual tasks that analysts must perform during investigations and analysis. For example, data collection, preliminary threat analysis, execution of containment actions, and incident notification can be automated. This reduces response time and allows analysts to focus on more complex and strategic tasks. By automating repetitive tasks and reducing manual workload, SOAR improves SOC operational efficiency. Analysts can handle more incidents in less time and with fewer resources. By integrating multiple security tools and centralizing incident management, SOAR improves the SOC's ability to detect and respond quickly to threats. Centralized visibility makes it easier to identify correlations between events.

SOAR facilitates faster and more effective incident response. Using predefined workflows typically dictated by the various incident response plans, the platform can automatically execute specific response actions by threat taxonomy. Examples of typical operations may include isolating compromised devices, blocking malicious

IP addresses, and disconnecting suspicious users.

The SOAR platform in the SOC is used primarily because of the ability to leverage tools to track and manage the entire lifecycle of various investigation cases, from opening to resolution while also considering post-case closure operations. Analysts can assign, track and document activities associated with each incident, improving visibility and collaboration within the security team.

3.2.3 Data sources on-boarding

In the existing SOC service offered to a general customer, the on-boarding process is critical to ensure proper identification, management, and monitoring of the customer's assets. It begins with the cataloging and classification of IT assets according to their severity and criticality in the infrastructure. Next, a threat assessment associated with each asset is performed to determine its potential impact in the event of an attack. Data sources are classified using Traffic Light Protocol (TLP) to determine their sensitivity and to establish specific rules for triggering incidents, validating the incident management processes together with the customer. The process of on-boarding log sources is constantly evolving: sources that are obsolete or no longer needed must be removed through an off-boarding process.

It is, in addition, necessary to define effective ingestion pipelines for data entry of new log sources for each new asset to be monitored. The ingestion pipeline process begins with gathering the requirements of the new log sources and configuring the log agents (e.g., Beats or Logstash) to collect the data. The logs are then parsed, normalized and standardized using specific tools and schemas. Next, the data are filtered to remove unnecessary information and enriched with additional useful meta-data. The transformed logs are then forwarded to SIEM.

The ingestion pipeline is tested to make sure it is working properly before deployment, followed by initial monitoring to identify any problems. Continuous monitoring and periodic maintenance are useful to ensure optimal performance and to update configurations based on changes in the customer's infrastructure, considering the variety of logs received.

3.3 Open source context

Open Source Software (OSS) refers to software released with a license that grants users the rights to study, modify, and distribute both the software and its source code freely. In general, two main standards regulate open-source software (OSS): the Free Software Foundation (FSF) [3], which prioritizes software freedom as an ethical issue and advocates for user rights to use, modify, and share software, and the Open Source Initiative (OSI) [5], which focuses on the practical benefits of open source, such as collaboration and innovation.

Unlike proprietary software, where the source code is kept confidential, OSS promotes collaboration and community-driven development. In general, one of its primary advantages is **transparency**, as the source code is publicly available for inspection. Developers and users can identify vulnerabilities, bugs, and inefficiencies more quickly, often leading to faster development cycles. Large communities can collaborate to contribute patches and improvements, which helps keep the software up to date. Additionally, OSS is typically **cost-effective** since it is free to use, reducing financial expenses for organizations compared to proprietary software licenses.

Another key benefit of OSS is its **customizability**. Users and organizations can modify the software to suit their specific needs. This is especially advantageous for companies with unique requirements that cannot be met by proprietary software.

The flexibility to adapt software to particular operational needs is an added value, especially in SOC environments, where OSS solutions introduce several advantages. In detail, SOC environments rely on multiple tools for tasks such as log management, intrusion detection, vulnerability scanning, and security monitoring. OSS offers these capabilities without the high costs associated with proprietary tools, making it a cost-effective solution for small to mid-sized organizations. OSS also allows SOC teams to customize tools to meet specific security needs, enhancing the ability to detect and respond to threats. The collaborative nature of OSS is especially beneficial in SOCs, as large communities of developers and security professionals constantly improve and update the tools. This ensures that the software evolves rapidly to meet new security challenges, with patches and enhancements

typically implemented faster than with proprietary software vendors.

Transparency is crucial in SOC operations, and OSS provides this by allowing teams to audit the source code themselves. This self-auditing capability enhances incident response by enabling the SOC team to identify potential security issues, address them directly, and release associated security patches in real-time, which can be an important feature to rely on in SOC environments where rapid response to emerging threats is critical. Moreover, OSS supports a high degree of interoperability, allowing seamless integration with tools commonly used in SOC environments and other ones used for business operations. This flexibility enhances automation and data correlation, improving the SOC's overall efficiency in detecting and responding to threats.

3.3.1 Licensing

Licensing is a critical aspect of OSS, as it defines the legal context under which the software can be used, modified, and distributed. Unlike proprietary software, where users are typically restricted, OSS licenses grant users more freedoms. These licenses allow anyone to access the source code, modify it to suit their needs, and redistribute both the original and modified versions, often with certain conditions attached.

There are several types of OSS licenses, each with varying levels of restrictions and freedoms. Here are described license types of software that have been used and will be described in the next sections:

- **Apache-Licence 2.0:** The Apache License 2.0 is a permissive open-source license that offers significant flexibility for users to freely use, modify, and distribute software. It allows derivative works to be released under any license, including proprietary ones, making it highly attractive for commercial use. The license includes explicit patent protection, meaning contributors grant patent rights, reducing the risk of patent litigation. However, users must provide proper attribution to the original authors in the source code and documentation. Since it does not impose copyleft requirements, modifications do not need to be open-sourced, making it ideal for integrating open-source software into proprietary projects.

- **Affero General Public License 3.0:** The AGPL-3.0 is a strong copyleft license designed to ensure that software, especially when used over a network, remains free. It adds a crucial condition for network-based applications: if the software is run as part of a service, users must have access to the source code of the modified version. This network interaction clause protects users by ensuring they retain the freedom to access and modify software, even when interacting with it over the web. By requiring modifications to be released under the same AGPL-3.0 license, it enforces strict software freedom, particularly for online or hosted services.

3.3.2 Elastic Stack

Elastic [2] ELK Stack is a suite of tools designed for the analysis and management of large volumes of data, particularly logs and metrics; it can be used as SIEM in a SOC environment. The name ELK comes from the initials of its three main components: Elasticsearch, Logstash and Kibana. This suite offers a comprehensive solution for data ingestion, processing, storage and visualization, making it particularly useful in areas such as IT infrastructure monitoring, security analysis and operational intelligence. Elastic is licensed as an open source tool with Apache 2.0 up to the version 7.10.2 which is the one used in this study. However, Elastic has also transitioned some of its features to a more restrictive dual-license model, introducing the Elastic License and the Server Side Public License (SSPL) for certain versions and features. An open source fork of the Elastic project based on the version 7.10.2 has been called OpenSearch.

Here follows an explanation of each tool in the suite that composes the workflow shown in Figure 3.3

Elasticsearch

Elasticsearch is an open-source distributed search and analysis engine based on Apache Lucene. It is designed to be scalable and to provide fast answers to queries. Its primary use is indexing and searching large volumes of data, providing a robust platform for information analysis and monitoring, and is particularly well-suited for searching for information within the large volume of logs collected by an SOC.

Furthermore, it is built around a distributed architecture consisting of nodes and

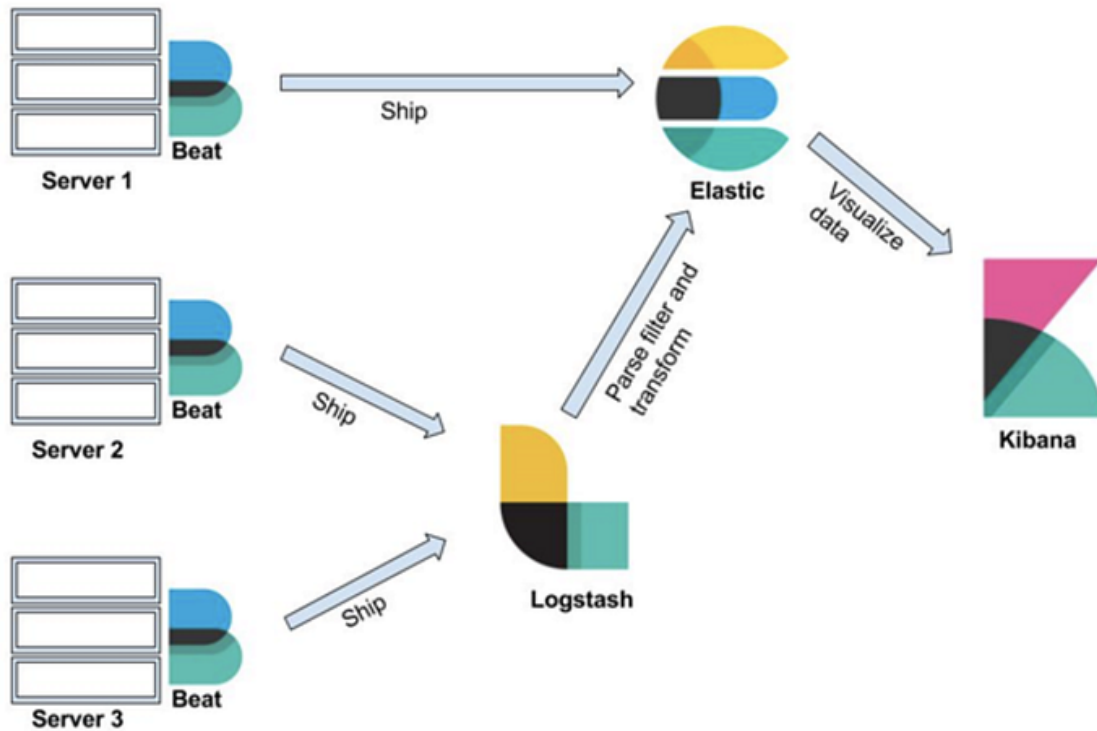


Figure 3.3: Elastic architecture

clusters. A node is a single instance of Elasticsearch that can run on a physical or virtual machine. A cluster is a collection of one or more nodes that work together to manage and distribute data. Each cluster is identified by a unique name, and each node has a unique identifier within the cluster. The data structure in Elasticsearch is based on indexes, which are similar to databases in a relational database management system. Indexes are in turn divided into types, which contain documents, each of which is a JSON representation of an entity, which in the case of SOC is primarily a log from a specific source.

Elasticsearch uses a document architecture, where each document is a JSON containing data organized into fields. When a document is indexed, Elasticsearch parses the contents and stores them in a structure that enables fast and efficient searching. In addition, it provides the ability to scale horizontally. New nodes can be added to a cluster to increase processing capacity and data storage capacity. Elasticsearch automatically distributes data among cluster nodes, managing replication to ensure reliability and fault tolerance.

Logstash

Logstash is an open-source tool designed for efficient data collection, processing, and transformation. It operates as a data processing pipeline, making it highly versatile for dealing with structured and unstructured data from various sources. Logstash's architecture is composed of three main phases: *input*, *filter*, and *output*. A key enhancement to its capabilities is its support for the **Elastic Common Schema (ECS)**, which helps standardize data representation, making it easier to analyze and correlate across different sources and datasets.

In the **input** phase, Logstash collects data from numerous sources, such as log files, databases, network messages, and other formats. This stage supports a wide array of input plugins, including protocols like HTTP, syslog, TCP/UDP, and file-based inputs like CSV and JSON. Logstash's ability to interact with different data types is crucial for SOC environments, where different kinds of logs are gathered. With ECS, the data collected at this stage can be normalized right from the start. The Elastic Common Schema provides a standardized way of structuring fields across various types of logs. For instance, fields like IP addresses, user agents, timestamps, and hostnames can be stored consistently across different input types, ensuring a uniform format that aids downstream analysis. This is particularly useful for correlating events from sources like firewalls, IDS/IPS, and application logs in security monitoring.

Once data is ingested, it moves to the **filter** phase, where it undergoes transformation, enrichment, and restructuring. Logstash provides various filter plugins such as *grok* or *mutate*, which will be used later in the actual implementation. Incorporating ECS in this phase ensures that fields generated or transformed by filters adhere to a common structure. This standardization is essential for maintaining data consistency when logs are analyzed. In detail, *Grok* is a filter plugin in Logstash used for parsing and extracting structured data from unstructured log messages. It works by matching log patterns with predefined regular expressions, called Grok patterns, to break down complex log data into meaningful fields.

In the **output** stage, the processed data are sent to one or more destinations. Logstash supports numerous output plugins, allowing data to be sent to Elasticsearch, relational databases, files, message queues, and other destinations. In this

study, only the link between Logstash and Elasticsearch is considered.

Logstash is a flexible tool that can be installed where needed. However, to improve performance in the existing SOC architecture, it can be installed directly in the customer's network. This relieves the central SOC cluster from pre-processing log data before it is stored in Elasticsearch.

Kibana

Kibana is a browser-based user interface for visualizing and analyzing data stored in Elasticsearch; it is designed to make data easily understandable through interactive dashboards and advanced visualizations. This tool provides a variety of data visualization options, including bar graphs, line graphs, pie charts, heat maps, and so on. These visualizations can be configured to suit the analyst's specific needs and according to the task they need to conduct. One of the features of Kibana is the ability to create customizable *dashboards* that are useful for getting a complete overview of the data. Dashboards can display unauthorized access attempts, suspicious activity, and other critical security metrics. Kibana simplifies exploration and search of the data stored in Elasticsearch. In addition, the large amount of data can be filtered by primarily considering the time periods associated with the alert, then log fields can be used as an additional filter and combined with the **Kibana Query Language (KQL)** to achieve a more precise filter. This is particularly useful for identifying trends, anomalies, and specific insights.

Kibana is tightly integrated with Elasticsearch, which means it can take advantage of all the advanced indexing and search capabilities of Elasticsearch. Data are retrieved from Elasticsearch, processed, and then displayed in Kibana. This integration allows Kibana to handle large volumes of data in real time.

One of the most common uses of Kibana is the monitoring and analysis of log files. Combined with Logstash and Beats, Kibana can display log data from a variety of sources, making it easier to diagnose problems and optimize application performance. This helps the SOC detect and respond quickly to threats.

Kibana is highly extensible due to its plugin-based architecture. Users can develop and install custom plugins to add new features or integrate Kibana with other applications and services. This flexibility allows Kibana to be adapted to a wide range of specific needs and to allow integration with other tools already used by customers,

such as most EDR solutions like Defender and CrowdStrike Falcon.

Beats

Beats is a data collection platform in the Elastic suite. It includes several specialized agents to be installed in the monitored infrastructure:

- Filebeat: Collects and sends log files from various sources.
- Metricbeat: Collects system and service metrics.
- Packetbeat: Monitors and analyzes network traffic in real time.
- Heartbeat: Monitors availability and response times of services.
- Auditbeat: Monitors system-level security events.

Beats integrates seamlessly with Elasticsearch and Logstash. It is used for real-time monitoring and analysis of log data, metrics, network traffic, service availability and security events. Beats is the component that, in the Elastic infrastructure used for monitoring, is responsible for collecting logs and sending them to Logstash or into Elasticsearch to be then indexed for analysis and search. Each host in the monitored infrastructure must install a Beats agent in order to be fully tracked.

3.3.3 The Hive Project

TheHive 4 [9] is an open-source SOAR platform under AGPL-3.0 license designed for security incident management and automation. This platform offers a wide range of features that make it an ideal choice for SOCs seeking a flexible and customizable solution to improve their incident response capability.

Being open-source, TheHive is highly customizable, which means it can be adapted to the specific needs of the SOC and its customers. The platform easily integrates with a wide range of security and SIEM tools (including Elastic) facilitating data collection and correlation. This integration is critical because it allows information from multiple sources to be aggregated, creating a comprehensive and centralized view of IT infrastructure security. The ability to correlate this data from various tools enables rapid identification of threats and anomalies, improving the SOC's ability to respond effectively to incidents.

One of the strengths of TheHive is its support for collaboration among analyst teams. The platform enables real-time information sharing, which is crucial for coordinating incident responses in a quick and organized manner. With this feature, team members can work together more effectively, exchanging information and updates on ongoing incidents, reducing response times, and improving problem resolution and case analysis.

TheHive offers the ability to integrate automation operations, which significantly reduce manual workload and increase the efficiency of incident response operations. Automations allow repetitive and routine tasks to be handled, freeing security professionals to focus on more strategic and critical activities. In a SOC, TheHive's automations can perform several key operations, including:

- **Enrichment:** This process involves enriching logs with additional key information regarding each case, called *observables* (observable objects, such as IP addresses, domains, file hashes). Enrichment can take place using an OSINT approach, which leverages public sources to gather additional information. TheHive uses specific tools for analysis called *analyzers*, which can include services such as VirusTotal, IPAbuseDB, and Netbox. These tools help to gain a deeper, more contextual understanding of incidents, improving the quality and effectiveness of investigations.
- **Response:** TheHive enables automated incident response operations through *responders*, such as case closure or blocking malicious IPs. Automated responses are essential for responding quickly to threats and minimizing the impact of incidents.

In addition to these capabilities, TheHive also includes advanced tools for case management and forensic analysis. Case management is facilitated by an intuitive user interface that allows tracking and documenting every stage of an incident, from initial investigation to resolution. This structured approach helps ensure that all actions are documented and that incidents are managed consistently and comprehensively. TheHive's forensic analysis tools make it possible to examine historical logs and reconstruct past security events, providing a detailed understanding of the sequence of events and causes of incidents.

Cortex

Cortex [7] in TheHive is a security incident analysis and response platform that provides a structured, centralized way to perform automated and semi-automated analysis. Cortex is a key component of TheHive, designed to facilitate and automate the analysis of security events. Cortex, being part of TheHive's open-source ecosystem under the AGPL-3.0 license, benefits from an active community of developers and users. Cortex integrates seamlessly with TheHive to enhance incident response capabilities. Through this integration, TheHive can send analysis requests to Cortex and receive the results automatically, making the incident management process more efficient and faster.

Cortex uses a number of analyzers, which are predefined or customizable modules, typically Python scripts, designed to perform specific analysis tasks automatically. The analyzers can cover a wide range of functions, including URL scanning, file hash verification, IP address analysis, Indicator of Compromise (IOC) look-up and more. The analyzers can be run automatically in response to specific events or manually by the analysts.

Cortex is mainly used for its automation capabilities. It allows analysts to reduce their manual workload by quickly performing repetitive and complex analyses. This allows analysts to focus on more critical and value-adding tasks. The platform supports the creation of automated workflows that can handle a variety of analyses and responses based on specific triggers. Although there are other open source tools for SOAR capabilities like Wazuh or Shuffle, TheHive can be chosen to implement a SOC, as previously explained, for its comprehensive capabilities, including integrated threat and case management that tracks the entire incident lifecycle and supports multi-user collaboration. Its native integration with Cortex allows for automated analysis of observables and IOCs, streamlining incident handling. The platform's scalability and customization options enable SOCs to adapt workflows and response plans to their specific needs. Additionally, TheHive provides features for collaboration, role-based access control, and detailed incident tracking, ensuring extensive documentation and effective management of security incidents.

3.3.4 Netbox

NetBox [6] is an open source IT infrastructure management platform, under Apache 2.0 license, designed to aid network administrators in documenting and managing various aspects of networks and data centers. NetBox is used because of the need to include a robust system for managing and documenting network infrastructure in the SOC monitoring system. Its main features include management of IP addresses, subnets, VLANs, racks in data centers, network devices, and the physical and logical connections between them. Administrators can track used and available IP addresses, assign them to specific devices and document associated details. NetBox also supports subnet and VLAN management, allowing documenting address ranges, subnet masks and routing information.

NetBox main strength is data center management, allowing documentation of racks, including their physical location, capacity and occupancy. Users can document details about the devices mounted in the racks and the wiring configuration between them. It also supports documentation of network devices such as routers, switches, firewalls, and servers, including details such as model, manufacturer, and hardware specifications, and tracks the network interfaces of each device.

As for connection management, NetBox allows documentation of cables and connectors, mapping physical connections between components and keeping track of layer 2 and layer 3 connections. It also supports management of virtual machines and virtualization clusters, including configuration details and allocated resources.

NetBox supports automation through its REST API, which allows integration with other systems and automation of management operations. It also supports custom plugins and scripts to extend the platform's native functionality. This automation and integration capability improves operational efficiency.

Since QiNet also provides the NOC service, NetBox is used as an asset inventory or Configuration Management Database (CMDB) of network devices and hosts with associated IPs that allows SOC analysts to retrieve additional information if they are involved within security alerts and incidents considering a monitored infrastructure for which those details are known and fully mapped within the tool. In addition, in Cortex it is possible to include an analyzer that interfaces with NetBox in case of specific IP addresses, this option is later explored.

3.3.5 Information Gathering and OSINT Tools

OSINT is a particular framework [4] that involves the systematic collection and analysis of publicly available information to detect and address security threats, making it a critical element of SOC operations. OSINT leverages a wide range of open and freely accessible data sources, including websites, social media platforms, public databases, online forums, and even WHOIS records or DNS lookups, to provide a broader context about potential attackers, vulnerabilities, and suspicious activities. By enriching log data with this external information, analysts can better understand the motivations, methods, and infrastructure behind cyber threats.

In a SOC, OSINT tools and techniques complement traditional log analysis by offering insights that internal systems alone cannot provide. For instance, analysts can discover details about emerging malware, track threat actor activities across social platforms, and identify exposed credentials or misconfigurations by scanning repositories. Threat intelligence feeds can also be integrated, providing real-time updates on vulnerabilities or threat actors based on OSINT data sources.

While OSINT typically refers to publicly available information, it's important to distinguish that the term does not inherently refer to the use of open source software or free tools. Instead, it reflects the open and public nature of the information being accessed. SOC teams may use both open source tools (e.g. theHarvester) and proprietary OSINT platforms (e.g. Shodan) to enhance their analysis capabilities.

In addition to OSINT tools, sandboxing tools play a vital role in SOC environments but are not classified as OSINT. Sandboxes provide a controlled environment to execute and observe potentially malicious files or scripts without affecting production systems. Tools like these allow analysts to safely analyze malware behavior, network communications, and potential IOCs. Though sandboxing tools are more focused on behavior analysis rather than information gathering, they may provide crucial data and employ public sources to retrieve additional information like IOCs. Also, when combined with OSINT, they have the ability to provide a comprehensive view of a security incident.

Here follows the complete list of OSINT tools used in SOC analysis for information gathering and sandboxing; Although they are employed manually, they can be

integrated as Cortex analyzers in certain situations..

- **AbuseIPDB:** This tool provides information on IP addresses reported for malicious activity. Users can report and share their experiences regarding suspicious IP addresses, allowing the analyst to identify potential IOCs. As a best practice, it is always best to consider the trustworthiness and activity of any reporters,
- **CentralOps.net:** This is an online platform that offers a variety of tools for network infrastructure analysis and intelligence. This tool provides detailed information on domains, IP addresses and other aspects of the network.
- **Cisco Talos:** Cisco Talos is Cisco's threat intelligence group, which focuses on cybersecurity research and analysis to protect users, data and infrastructure. Talos provides in-depth intelligence on malware, vulnerabilities, phishing campaigns, and other threats. The group analyzes large amounts of data from Cisco security appliances, partners, and other sources to identify and mitigate emerging threats.
- **GreyNoise:** GrayNoise is a threat intelligence platform that collects and analyzes global network traffic data to identify malicious traffic on the Internet. GrayNoise helps filter out false positives from their security monitoring systems, reducing the number of unnecessary alerts and allowing analysts to focus on real threats. The platform provides information on suspicious IP addresses, including details on behavior, geolocation, and frequency of activity.
- **Scanalytics:** This tool provides information on IP addresses, proxies, and servers associated with online scams, frauds, and other fraudulent activities on the Internet. It uses a large database to identify suspicious patterns and behavior and helps companies protect their users from online threats. In analysis, it is primarily useful for identifying the use of anonymizing proxies or VPNs.
- **Spur:** This tool is designed to identify the use of VPNs or public proxies, which could be used to hide users' identities or bypass geographical restrictions. Spur

collects information on a wide range of VPN services and public proxies and helps identify potential security risks associated with the use of such services.

- **Microsoft Message Header Analytics:** Online tool provided by Microsoft that allows users to analyze email headers. Users can copy and paste an e-mail header into the Web interface to get a detailed analysis. This tool decodes and displays information contained in the headers, such as mail servers traversed, delivery times, IP addresses, and other technical details. It is designed to help diagnose mail delivery problems, detect phishing and spam emails, and trace the origin of emails.
- **Filescan.io:** FileScan.io is an online platform that allows users to upload and scan suspicious files. The platform scans files for malware or malicious activity, providing a detailed analysis of file behavior. FileScan.io uses a combination of static and dynamic analysis techniques to identify threats and provides detailed reports that include file information, hashes, suspicious behavior, and IOC. Although not primarily an OSINT tool, Filescan.io relies on publicly accessible file analysis techniques.
- **VirusTotal:** It is an online platform that allows users to analyze files, URLs, and domains for malware, viruses, and other security threats. It works by uploading files or URLs, followed by a scan that makes use of about 70 antivirus and anti-malware engines. Once the scan is complete, it provides a detailed report of the results, including positive and negative detections by the different antivirus engines. VirusTotal can be used within an SOC for suspicious file analysis, URL verification, and IOC searching. During the analysis, it is best practice to consider the trustworthiness of any vendor associated with the engine that reports the indicated object, URL or hash file, as malicious. when it comes to analyzing custom and private files, VirusTotal is unable to provide information since these files are not included in its public database. In such cases, it is necessary to contact the owners or development teams associated with the files directly to obtain additional information. This may include information about the nature of the file, its provenance, purpose, and authenticity.

- **Abuse.ch:** Abuse.ch is a project that aims to combat cyber threats, particularly malware. The website collects, analyzes and shares information on various forms of malicious activity, such as botnets, ransomware and other threats, providing various blocklists and intelligence feeds that can be used to improve network defenses and for research activities. Among the resources offered by Abuse.ch are URLhaus, a platform that collects and shares information about malicious URLs used to distribute malware; Feodo Tracker, which monitors and shares information about the Feodo botnet used to spread banking malware; SSL Blacklist, a list of SSL certificates associated with malicious servers; and ThreatFox, a collaborative platform for sharing IOCs.
- **AlienVault:** This tool provides threat detection and security intelligence capabilities, including identification of compromised servers, malware and other suspicious activities. It is primarily used for analysis of servers involved in analysis. It is not open-source except for its OSSIM tool.
- **Shodan:** It is a search engine specializing in Internet-connected devices, including servers, IoT devices, network devices, and more. It collects information about services exposed on the Internet, including details about protocols, software versions and known Common Vulnerabilities and Exposures (CVE), which is the main reason why it can be useful for analysis.
- **AppAnyRun:** This is an online platform that provides automated analysis of executable files, documents, and suspicious URLs. It uses a combination of sandboxing and dynamic analysis to detect suspicious behavior and potential threats. In a SOC, analysts can use AppAnyRun to analyze suspicious files or URLs and evaluate their behavior in a secure environment with being careful about avoid sharing sensitive data since the tool employs and shared informations publicly.
- **Joe Sandbox:** Joe Sandbox is an advanced malware analysis platform that allows suspicious files and URLs to be securely run and analyzed within an isolated environment. The platform offers a variety of features, including detection of malicious behavior, generation of detailed reports, and visualization

of malware activity. Joe Sandbox supports a wide range of file types and provides tools for static and dynamic malware analysis.

- **Browserling:** Browserling is an online service that allows users to test Web sites on different browsers and operating systems in real time. Users can access remote browser sessions to test the compatibility of their website on various browsers without having to install anything on their computer. Browserling supports a wide range of browsers and versions, providing a quick way to perform cross-browser testing.
- **Windows Sandbox:** Windows Sandbox is an isolated execution environment built into Windows allows users to run untrusted or suspicious applications safely and in isolation. It uses hardware-based virtualization to create a temporary, isolated instance of Windows, separate from the user's main environment.
- **URLScan.io:** This is an online service that allows dynamic analysis of suspicious URLs. It uses sandboxing to analyze the behavior of a URL when it is visited, identifying potential threats such as phishing, malware or malicious websites.

Tool	Category	Primary Usage	OSINT
AbuseIPDB	IP Analysis	Database of malicious IP addresses	Yes
CentralOps.net	IP Analysis	Platform for IP and domain analysis	Yes
Cisco Talos	IP Analysis	Threat intelligence feed	Yes
GreyNoise	IP Analysis	Threat intelligence platform	Yes
Scamalytics	IP Analysis	Analysis of malicious IPs	Yes
Spur	IP Analysis	Search engine for domain and IP information	Yes
Microsoft MHA	E-mail Analysis	E-mail header analysis	Yes
Filescan.io	File Scanner	File scanner for malware	Yes
VirusTotal	File Scanner	File and URL scanner for malware	Yes
Abuse.ch	Malware and IoC	Collaborative database of malicious IP addresses	Yes
AlienVault	Malware and IoC	Analysis of compromised servers	Yes
Shodan	Malware and IoC	Search engine for internet-connected devices	Yes
AppAnyRun	Sandbox	Sandboxing and dynamic threat analysis	No
Browserling	Sandbox	Website analysis across different browsers and OS	No
Joe Sandbox	Sandbox	Static and dynamic malware analysis	No
Windows Sandbox	Sandbox	Environment for analyzing suspicious applications	No
URLScan.io	URL Scan	Website scanner for malware and phishing	Yes

Table 3.1: List of OSINT Tools and Their Usage

Chapter 4

Security Incident Management Process

The security incident management process, or SOC workflow, is structured into several steps and components involved, and outlines the operations performed by the SOC L1, SOC L2, and CSIRT (L3) teams. Each level has specific and coordinated tasks to ensure a structured response to security incidents, from initial identification to final resolution and ticket closure.

The SOC L1 team operates 24/7 and is responsible for the early stages of incident management. Their operations begin with the pre-evaluation of alerts from various sources. Once an alert is received, SOC L1 opens a case investigation on the SOAR. It then proceeds with the assessment of the alert following an internal workflow. This step includes the assignment and classification of the incident, in case the alert is related to an event that impacts the security of company assets. If no security issue is detected, the team creates an operational ticket to resolve any associated operational issues and closes the investigation case on the SOAR. If the incident is classified as a security problem, a security incident ticket is created or updated. SOC L1 then passes the case to SOC L2 (this is the escalation process) for further investigation and more specific handling.

The SOC L2 team goes into action after the initial classification performed by SOC L1. Their first responsibility is the classification of safety incidents. If the incident is recognized as a known security incident, they follow standard operating procedures which may include Organizational Response Plan (ORP) or Incident Response Plan (IRP) for resolution. If the incident is not classified as known or requires further review, the L2 SOC may continue with the analysis and management to ensure that

all appropriate measures are taken. Their activity may involve creating updates in security tickets and passing the case to the CSIRT if necessary.

CSIRT is involved for critical incidents that require a higher level of intervention. Once the case is received from SOC L2, the CSIRT conducts a detailed assessment of the incident. If the incident is classified as critical, the team handles the resolution with high priority. If the incident is not critical, the CSIRT still handles the resolution, but with a lower level of urgency. If the need for forensic analysis is determined, the CSIRT proceeds with the investigation to collect and analyze detailed evidence. Once incident management is complete and all necessary actions have been implemented, the CSIRT closes the case on SOAR and the ticket, marking the conclusion of the incident management process. From a general point of view, the security incident management process begins when, following the collection of various logs from the monitored assets, an alert is generated and needs to be analyzed. As a preventive operation, the SOC assesses whether the alert belongs to its sphere of competence, that is, whether it is responsible for its assessment and resolution of any associated problems. If the alert does not fall within the SOC's competence, the related information is forwarded to other responsible business units.

If the alert is taken over by the SOC, a case is opened for investigation. Next, the alert is evaluated, classified, and related triaged, determining all associated information and the severity of the reported issue.

The first step is to check for any elements and evidence that indicate an attack or malicious activity is in progress or has already occurred. If malicious activity is found, mitigation is promptly initiated. Regardless, the analyst continues with the analysis of logs and information useful to the case. The search for indicators that prove the presence of malicious activity is constant and occurs after each phase in a cyclic manner.

When the analysis is considered exhaustive, remedial suggestions are sent to the customer or the person involved in the investigation via operation ticket. In the case of serious incidents, escalation of the case to the L2 or CSIRT team for advanced analysis is carried out. At the end of these operations, the case can be considered closed. In the next sections, each stage of the security incident management process is expanded and explained in full detail, considering each possible occurrence.

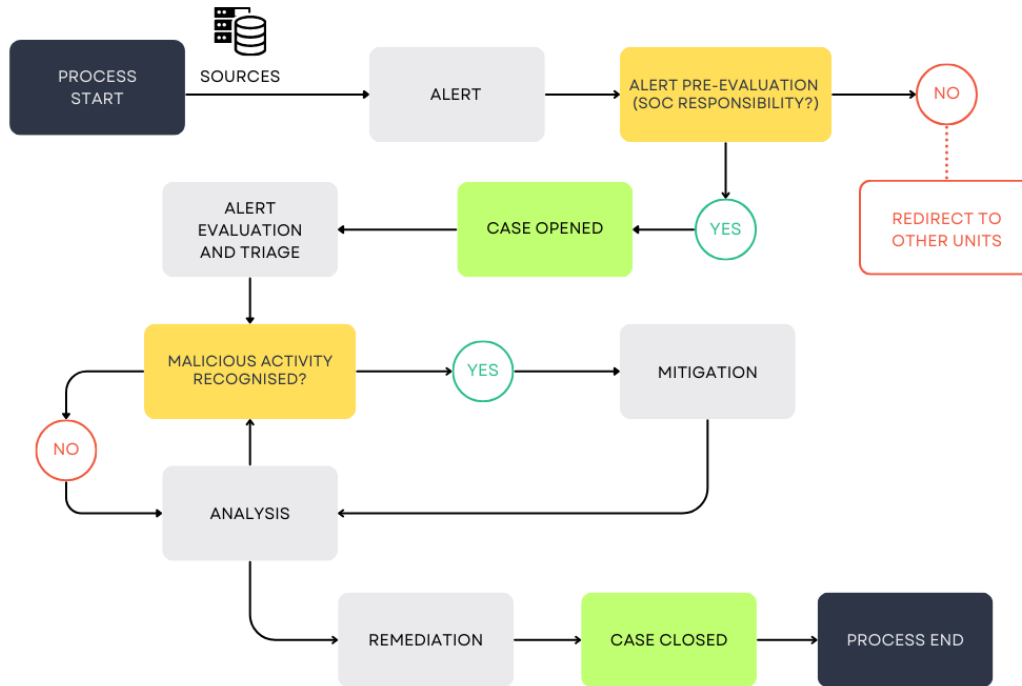


Figure 4.1: SOC Workflow Summarized

4.1 Alert Data Sources

The workflow within a SOC begins by identifying the sources of alerts, which are subsequently evaluated and analyzed to support investigations by dedicated teams. **Automated alert systems**, such as the Elastic ELK stack, play a critical role in incident detection. Specifically, Kibana enables the configuration of alert rules that define the conditions for triggering notifications when suspicious or anomalous behavior is recorded. Once triggered, these alerts require detailed evaluation by the SOC to determine their relevance and severity.

User-reported incidents constitute another significant source of alerts. Reporters, often external to SOC teams, may observe unusual activities or behaviors and notify the SOC via communication channels such as email or telephone. These reports often provide initial information for investigation. The process is streamlined using ticketing systems which facilitate the collection and tracking of such reports.

In addition to responding to automated alerts and user reports, SOC analysts engage in proactive **threat-hunting** activities. This involves deliberately searching for evidence of malicious activity across dashboards (e.g., Kibana as part of the SIEM),

EDR platforms, or sensors installed on organizational assets. Analysts focus on identifying key indicators, including:

- **IOCs:** Forensic evidence signaling potential intrusions within systems or networks, essential for detecting breaches or malicious activities.
- **Suspicious Operations:** Unusual behaviors or activities that may indicate the early stages of an attack.
- **Vulnerability Indicators:** Evidence of weaknesses within assets, components, or communication pathways that attackers might exploit. Reporting and addressing these vulnerabilities is a critical component of maintaining robust security postures.

The hunting process complements automated detection systems by addressing potential gaps and identifying threats that may have been missed. While performing these activities, SOC analysts must also continuously monitor the SIEM, data sources, and consoles to ensure standard alerts are not overlooked and that the core functions of the SOC are effectively executed. This dual focus on reactive and proactive measures enhances the overall effectiveness of the SOC.

4.2 Alert Pre-Evaluation

When an alert is received by the SOC, the analyst is responsible for conducting a preliminary evaluation on it.

In the initial phase of alert analysis in the SOC, the first step is to determine whether the alert is related to activities or events that fall within the SOC's area of responsibility. This process includes consulting the **RASCI** matrix which clearly defines the roles and responsibilities within the organization in relation to the activity or event that has occurred. For example, operations that fall within the SOC's responsibility include clicks on malicious or suspicious links and running unrecognized programs. In contrast, examples of operations not associated with the SOC might include password resets or VPN configuration. If the alert does not fall within the SOC's scope, the associated incident handling is concluded, and the case is forwarded to other specialized teams.

Once an alert is recognized as within the SOC's responsibility, it is essential that the analyst promptly **acknowledges** it to manage the analysis and resolution of the associated case. This step is critical for the customer, as it indicates how quickly the SOC can respond to security incidents. Often, this metric is agreed in advance and is known as the *Mean Time To Acknowledge (MTTA)*. The time taken to acknowledge an alert and begin the response process can significantly impact the overall performance of the SOC.

Once the alert has been assigned, the actual preliminary evaluation phase begins. The first step in this process is conducting an **alert health check**, which involves a series of targeted checks to ensure the alert was correctly generated and contains all relevant information in an accurate manner, even after data parsing and export processes. In addition, the analyst checks whether the data sources are functioning properly and if threat detection sensors are operational. It is essential that all elements associated with the alert are identified as *observables* for subsequent analysis (e.g., IP address, username, user email, etc.). If necessary, the analyst responsible for the case must verify that the data sources are working correctly and that the sensors are operational. If observables are missing or of low quality, the analyst is tasked with manually retrieving and entering all the necessary information into the case to ensure an accurate analysis. If manual remediation of the alert is not possible, it is labeled as non-functional, and the abort incident management phase (4.6) starts.

The same series of events or multiple related events may trigger several simultaneous or consecutive alerts, which, in reality, pertain to the same context and may require a single comprehensive analysis encompassing all the cases involved. Once this aspect is assessed, the analyst either opens a new investigation case or updates an existing one with all the information obtained from the consolidation of multiple alerts. In cases where similar or related active cases are identified on the SOAR platform, referring to the same user or device, the best practice is to consolidate and analyze them within a single case. This approach requires analyzing the entire context. Typically, the primary case to be analyzed, which encompasses the others, is the oldest in the chronology. However, if the cases report different severity levels, the investigation's priority is given to the case with the highest severity. Another

task conducted during this phase is the **correlation of multiple alerts**. Analysts always seek to identify connections or patterns between very similar alerts. This type of correlation enables a more comprehensive understanding of the threats and allows for more informed decision-making during subsequent analysis. Such correlations are managed and supported by data analysis tools that associate events with each other.

The fundamental outputs of this phase are primarily feedback communicated to the customer, confirming that the case associated with the security incident has been appropriately handled or reporting any issues encountered. In the latter case, it is crucial to communicate the problem to the stakeholders responsible for managing it. These stakeholders can include operational units if the incident should not be handled by the SOC, SIEM operators, data source platforms in cases of malfunctions, or specific problems such as errors in data logs. Identity and Access Management (IAM) stakeholders may also be involved when handling account and credential management issues.

4.3 Alert Evaluation

The alert evaluation phase involves the detailed analysis of the case by a SOC analyst. Fundamentally, it implies the study of actors, devices, technologies, and protocols to enable the analyst to identify the cause that triggered the alert and determine whether the activities performed are false positives or malicious, thus requiring classification and remediation actions.

During this phase, the SOC analyst examines in detail the data and information available to understand the context of the event and identify patterns or anomalies that could indicate confirmed malicious activity. This process can involve analyzing system logs, network traffic flows, user activity records, and other relevant data. The SOC analyst uses their expertise, SIEM data, and OSINT tools to gather additional information and enrich the existing data to assess the severity and credibility of the alert. This may include consulting known threat databases, applying behavioral rules and models, and analyzing the characteristics of the activity to determine its nature and the intent of the users or, in the case of a confirmed incident, the malicious actors.

Based on the analyst's evaluation, the alert may be classified as a **false positive**, meaning that no actual incident occurred, or as a legitimate security event that requires immediate action. If the alert is confirmed as an attack or a threat, incident response operations are initiated to mitigate the impact and restore security through specific IRPs, ensuring the protection of critical systems and data.

A critical step in alert evaluation involves **identifying the actors** associated with the triggered alert. These may be the targets of the suspicious activity or users whose actions triggered the alert due to unauthorized behaviors in the organizational environment. For internal users, tools like OSINT and the organization's CMDB are utilized to analyze visibility and public exposure. External actors are similarly assessed through OSINT to identify their potential roles in the activity. Analysts also leverage SOAR platforms to correlate cases involving the same or related actors, enabling a broader context for analysis.

During the evaluation, the SOC analyst delves into the data to understand the context of the alert and detect patterns or anomalies indicative of malicious activity. To reach this conclusion, it is essential to first evaluate the **nature of the alert**. This involves assessing how the EDR system analyzes behavior and the indicators it uses to identify suspicious activity. The SIEM is also useful in this context, as it allows rules to be defined on logs, which, based on specific parameters, can generate alerts. It is also beneficial to examine similar previous cases that reported indicators and evidence of false positives, to evaluate whether similar patterns are present in the current case. This process involves analyzing system logs, network traffic, user activity records, and other relevant data sources. Using SIEM systems, analysts enrich their understanding by correlating logs and defining rules that may have triggered the alert. In addition, OSINT tools and threat databases are consulted to validate the presence of known indicators of compromise. If sufficient evidence points to a false positive, the alert is classified accordingly, and the case is closed. However, if the evidence remains inconclusive, the incident is treated as impactful until further proven otherwise.

If the case is not identified as a false positive, it may involve authorized internal activities within the organization. In this context, it is crucial to determine whether this is the case. **Authorized activities** include vulnerability assessments, pene-

tration testing, or scheduled maintenance activities. All entities involved must be informed of such activities, and they must be marked as authorized. The SOC analyst verifies details such as timeframes, associated systems, and the nature of the activity to confirm its legitimacy. Information such as IP addresses, hostnames, and approved schedules is reviewed to ensure alignment with authorized tasks. If the activity is confirmed and recognized as authorized, the case can be closed. Otherwise, the incident will be confirmed, and it will be necessary to identify its impact and perform incident response activities.

This phase of the SOC operations analysis flow requires a properly configured organizational or customer CMDB. The CMDB is essential for analysts as it allows them to identify the users, devices, and software involved in the potential security incident. Additionally, OSINT tools play a critical role in this context, as they allow analysts to determine whether any information associated with the investigation case is present on public sources. Such information can assist the analyst in retrieving IOCs useful to the case.

To keep the SOC informed of authorized activities such as vulnerability assessments, penetration testing, and scheduled maintenance of organizational assets, these activities must be known and shared. In the event of malfunctions or various issues, such as loss of operability, missing assets, or inaccessible databases, all CMDB and SOAR stakeholders must be informed to request the restoration of full functionality. This approach ensures a thorough and accurate analysis of security incidents and a prompt response to operational issues.

4.4 Incident Classification and Evaluation

The **classification** of incidents within a business or IT context depends on an evaluation of their severity and priority. This evaluation considers factors such as urgency, impact, and the specific systems or individuals affected. Severity levels are typically determined using a matrix or point-based scoring system, providing a structured approach to incident assessment. The severity level directly influences the response times and management procedures. Incidents involving critical users, such as executives (e.g., CEOs or CFOs), or essential business systems are assigned higher severity due to their potential operational impact. Similarly, incidents that

involve sensitive information, such as Personally Identifiable Information (PII) or Protected Health Information (PHI), are prioritized due to their reputational implications. Other factors include the number of affected users, meaning that an incident impacting a single user or a limited group is generally considered less severe than one affecting an entire department or organization. Resolution times are also critical, as any delays may indicate an underestimated severity level, in which case it is necessary to re-evaluate it.

Once a security incident is confirmed, the **incident classification and evaluation** phase begins. This phase involves defining the nature and impact of the incident, which includes analyzing the damage, identifying affected systems and data, and assessing the severity. The evaluation forms the basis for planning subsequent management and mitigation operations. Mitigation actions may include isolating compromised systems, removing malware, or blocking unauthorized access, with the aim of limiting damage and preventing further escalation. Effective classification ensures incidents are handled according to their priority and impact.

Communication plays a vital role throughout the incident response process. The SOC must maintain clear and **continuous communication** with internal stakeholders, such as analyst teams, and external entities, such as technical experts and customers. This coordination ensures a unified and effective response, minimizing delays and potential mismanagement. Effective communication also helps align the efforts of all involved parties, ensuring that mitigation and recovery actions are properly executed.

After an incident is resolved, a **post-incident** evaluation is conducted to analyze its root causes, the effectiveness of the mitigation measures, and lessons learned. This process aims to refine incident response strategies and implement improvements to prevent similar incidents in the future. By addressing the gaps identified during the evaluation, organizations can enhance their resilience and readiness for future threats. Post-incident reviews contribute significantly to an organization's long-term security posture.

The **data collection** phase is critical to determining and evaluating the nature of a security incident. This process involves identifying both internal and external actors involved, as well as compromised or affected assets. Analysts assess the

scope and perimeter of the incident's impact, identifying which components of the IT infrastructure were affected. Detailed information from log sources is essential to reconstruct the sequence of events and understand how the incident occurred. Protocols like TLP and Permissible Actions Protocol (PAP) ensure proper classification and handling of shared information. Additional insights from threat analysis, including attack vectors, techniques used and attackers' potential objectives, are also incorporated into this phase.

Impact assessment involves analyzing the extent of damage and the specific assets and data affected by the incident. Analysts aim to identify all impacted systems while excluding those that remain unaffected. This phase includes reviewing documented vulnerabilities, such as CVEs, to evaluate their relevance to the incident. Historical cases and tickets for similar incidents are consulted to identify patterns or inconsistencies that may inform the evaluation. This thorough analysis ensures a clear understanding of the incident's impact and provides a foundation for prioritization and response.

This step determines whether the incident qualifies as a cybersecurity issue by assessing its impact on the CIA properties of critical assets or data. **Cybersecurity incidents** require specialized attention, while other issues, such as physical security breaches or misconfigurations, may not. However, incidents with potential cybersecurity implications, such as tampering with assets or suspected denial-of-service attacks, are investigated further. Even if an incident is only suspected to be cybersecurity-related, it is treated as such until confirmed otherwise, ensuring appropriate caution and response.

In critical situations, such as ransomware attacks or compromises of key assets, incidents are escalated to L2 SOC analysts or, in severe cases, to L3/CSIRT teams. The findings and analysis from the initial investigation are handed over to these advanced teams for further evaluation and resolution. **Escalation** ensures that critical incidents are managed by skilled professionals, minimizing the potential for impact or escalation.

The classification and evaluation phase concludes with ticket creation, documenting the incident's findings and necessary actions. Tickets may be operational, for non-cybersecurity issues, or security-related, for incidents requiring IRP/ORP protocols.

For recurring incidents, existing tickets are updated to avoid redundancy and align with organizational policies. Tools such as the CMDB and SOAR, alongside ticketing platforms, are employed to manage incident records, ensuring accurate tracking and efficient resolution.

Throughout the incident evaluation and classification process, tools such as the CMDB and OSINT platforms are utilized to enrich information from the initial analysis. Communication with stakeholders managing the CMDB, SOAR, and ticketing systems is crucial to address any technical issues, asset accessibility problems, or operational challenges. Effective collaboration ensures seamless management of incidents and prevents delays in their resolution.

4.5 Impact Evaluation

The impact assessment phase occurs as a complementary component of the incident evaluation and classification phase 4.4. This process is critical to understanding the scope and implications of a security incident on company assets. During this phase, various sources of logs, trigger conditions, and other relevant information are analyzed. Analysts review logs from IDS, WAF, EDR systems and other IT infrastructure to gather information on how the incident developed and what parts of the infrastructure were compromised.

An effective approach to impact assessment also includes **replication of the attack** flow. This means that analysts try to reconstruct the actions taken by the attacker to better understand the dynamics of the incident. This operation makes it possible to identify any vulnerabilities exploited, the movement followed by the attacker within the network, and the potential consequences of his actions.

In addition, during the impact assessment, it is crucial to involve all stakeholders within the organization. This includes the teams responsible for IT asset management, information security, and business operations.

This phase also requires effective communication with external stakeholders, such as security partners and managed security service providers, to collect additional data and confirm the results of the analysis. Thus, impact evaluation and incident classification are closely interrelated because a detailed understanding of the impact is essential to determine the correct categorization of the incident.

The evaluation of CVEs, exploits, and POC attacks is a critical step in incident analysis. Once the vulnerability responsible for a security incident is identified, research is conducted on public sources to find associated CVEs. This research aims to identify confirmed exploits that could be used for similar attacks, determining the level of risk associated with the vulnerability. Simultaneously, documented POCs are evaluated to understand how an attack could exploit the vulnerability and its potential impact on organizational assets.

Trigger verification involves analyzing logs and network traffic to gain additional insights into the incident's potential impact. This process includes verifying trigger conditions for events and alerts linked to the incident, which helps identify its cause. The analyzed logs are compared against IOCs and POCs from previously observed attacks. This comparison confirms whether the events recorded match the documented POCs and whether the IOCs are present in the network traffic, providing evidence of the attack's nature and scope.

Target verification is aimed at gathering detailed information about the assets affected by the incident. The process begins by consulting the CMDB to check the versions of operating systems and applications on the impacted assets. By analyzing software versions and configurations, it is determined whether known exploits can be applied to the target assets. If the target asset is incompatible with the exploit, it is considered immune or unaffected, ruling out its involvement in the security incident. Historical checks of resolved tickets can also reveal whether similar vulnerabilities or exploits have been erroneously reported in the past, helping to reduce false positives and ensuring focus on assets that genuinely present a risk.

Target **logs verification** follows, involving a detailed analysis of available application logs to determine whether the attack was successful. This step is crucial in confirming whether the attack impacted the asset. Log analysis may uncover specific traces such as errors, abnormal behaviors, or unauthorized access attempts that validate the attack's execution. If the logs show no impact, the incident can be labeled as a false alarm, confirming that the asset remained unaffected by the attack.

Finally, attempting an attack on the target asset may be conducted, but only for authorized or publicly exposed assets. These tests must be non-destructive to avoid

damage or service disruption. By simulating the attack, analysts can interpret the results to confirm the asset's vulnerability. This feedback loop helps prevent the incorrect classification of incidents, ensuring accurate assessments and informed decision-making in incident response.

At the end of this phase, it is necessary to determine whether there has been a confirmed impact from the security incident, based on previously conducted activities. If the impact is confirmed, even if only potentially, an in-depth assessment of the incident is carried out and the relevant mitigation and remediation operations are implemented. Otherwise, if no impact is detected, the case is closed as a false positive, avoiding further action and unnecessarily spent resources.

At this stage, several elements need to be analyzed to determine the final result of the assessment. Depending on the type of asset involved, it may be useful to examine web server and WAF logs to identify unauthorized access attempts or web attacks. In parallel, it may be necessary to examine EDR or operating system logs to identify suspicious activity or compromise at the endpoint level. Application logs, along with IAM, PAM and VPN logs, can provide additional details on user behavior and access, contributing to a complete view of the incident. The CMDB plays a key role in this phase, enabling the collection of detailed information on the operating systems, applications, plugins, and versioning of the assets involved.

Finally, conducting vulnerability assessment activities on the potentially affected asset can provide essential information on vulnerabilities and CVEs to be associated with attacks and POCs. These assessments help identify specific weaknesses that could be exploited and confirm whether the security incident had a real impact, thus supporting the final decision on whether to close the case or take mitigation and remediation measures. This is a useful operation to learn how to improve security measures and address potential vulnerabilities or bugs in software assets.

4.6 Abort Incident Management

This phase includes all the operations necessary to conclude the investigation cases and communicate the results to the interested parties involved in the alert or incident. **Reports** documenting the analysis performed and the conclusions reached are provided to ensure transparency. If any issues are identified with the alert's parsing

or storage in the SIEM, remediation procedures are initiated to correct these errors. However, if no alert fixes are required, the SOC workflow typically ends after this phase. All actions and operations undertaken must be documented within the SOAR system, maintaining a detailed record for future reference.

A critical task in this phase is identifying the appropriate recipients of the **analysis results and reporting** findings accurately. The SOC's responsibility is limited to its operational scope, meaning it must determine the correct business unit or department to receive the report. When the appropriate unit is unknown, the SOC adopts a best-effort approach, providing guidance to the reporter to identify the relevant business unit. For instance, if a reporter contacts the SOC regarding a VPN issue, the SOC may direct them to the responsible unit if known, or advise consulting company documentation or the helpdesk. This ensures that the analysis results are appropriately communicated, even when IT security is not the root cause of the issue.

This step involves assessing whether the alert requires fixing due to receiving or parsing errors. Such errors can often highlight other issues in the monitoring system. If an alert is determined to need fixing, the SOC investigates and documents the source of the problem and the resolution steps. If no investigation case is associated with the alert, the relevant operations are still tracked within the SOAR system to ensure accountability and improve processes. Effective alert fixing contributes to enhancing the accuracy of future incident detection.

It is useful, then, to verify that all required steps have been completed and all relevant stakeholders are properly informed during this phase. The RASCI table serves as a tool for clarifying roles and responsibilities among team members, ensuring that every action item is addressed. Communication platforms are used to coordinate with stakeholders, while changes and updates are documented. Secondary platforms may also be utilized for specific operational requirements.

Communication remains a key component of this phase, focusing on engaging stakeholders listed in the RASCI table to confirm functionality and address any change requests. SOAR stakeholders are informed of any malfunctions or operational issues detected during the incident to refine response capabilities for future cases. Similarly, ticketing platform stakeholders are notified of issues, ensuring all activities

are accurately tracked. Providing detailed feedback, including status updates and explanations of resolutions through the engagement platform, ensures all parties remain informed and aligned.

The feedback process not only resolves current cases but also contributes to the overall enhancement of incident management processes. Clear documentation, structured communication, and a focus on addressing malfunctions collectively strengthen the SOC's ability to handle future alerts and incidents effectively.

4.7 Authorized Activity Evaluation

The process of evaluating authorized activities is essential for ensuring that planned operations do not compromise system security or cause unplanned disruptions. This evaluation distinguishes between legitimate activities and potential security incidents, ensuring that only authorized operations are recognized and properly handled.

Initially, planned maintenance activities are evaluated to rule out any service interruptions or inaccessibility not related to security events. This involves verifying IP/subnets, hostnames, and the perimeter associated with the maintenance, as well as confirming compliance with the planned maintenance time window. These checks ensure that planned operations align with system security policies and that any interruptions are the result of authorized activities. If the activity is confirmed as planned, the case is closed.

The next step involves verifying authorized Vulnerability Assessment and Penetration Testing (VAPT) activities. This process identifies whether the activities match authorized and planned operations. Targeted checks confirm that these activities are consistent with event flows and pose no security risks.

A critical element in this evaluation is verifying the correctness of the source IP address associated with the current activities. This includes analyzing private and public IP/subnets, examining potential intermediate NATs that may obscure the actual source, and reviewing the `X-Forwarded` field. These measures validate the legitimacy of the source and its compliance with authorized configurations. If the source address is incorrect, the incident proceeds to the classification and evaluation phase.

Similarly, the destination IP address is examined to confirm its validity. Private and public IP/subnets of the destination are analyzed, and intermediate NATs and front-ends that might conceal the actual destination are investigated. The destination domain and involved assets are verified against the target perimeter to ensure they align with authorized configurations. If the target IP is found to be incorrect, the process advances to the incident classification and assessment phase.

The authorized time window for the activity is then verified to confirm that the operation occurs within the specified period. This involves checking the validity of the time window and identifying events that exceed it. If an extension to the time window is not provided, such events are treated as security incidents. This step prevents potential risks from unauthorized time extensions or delays in planned activities.

After completing these checks, if no errors or anomalies are identified, the SOAR investigation case may be closed. Proper evaluation of authorized activities relies on structured inputs such as detailed authorization information, start and end times, and potential extensions for planned maintenance and VAPT activities. This information must be communicated to the SOC for continuous monitoring and must be integrated into the SIEM for event correlation. Tracking these activities in ticketing and SOAR tools ensures synchronization and efficient event analysis.

The outputs of this assessment are crucial for security risk management. They include communication with stakeholders regarding incomplete or incorrect notifications, events exceeding the authorized time window, or activities occurring outside the defined perimeter. Such information ensures all activities are properly monitored and managed.

Limitations on L1 analyst engagements in cases of improper activities or time window violations should be clearly defined. L1 analysts may escalate these issues to stakeholders when necessary, but all detected incidents and communications must be verified by L2 or L3 analysts before classification as actual events. Additionally, risks linked to supply chain technologies must be assessed to ensure a thorough evaluation of potential security impacts, further enhancing the overall security posture.

4.8 Alert Fixing and Tuning Procedure

In this phase, the procedure for managing and resolving **alert failures** is developed. The primary objective is to ensure that all alerts and events are effectively monitored, managed, and resolved, thereby preventing critical events from being missed or the system from becoming overwhelmed with unnecessary alerts. A key component of this phase involves recalibrating the **rules** that trigger alerts.

The first step involves gathering all relevant information about the alert failure. This information is essential for understanding the nature of the issue and determining the corrective actions needed. Accurate data collection enables a precise diagnosis of the problem and informs the subsequent steps in the process.

Once the necessary information is collected, an operational ticket is created on the ticketing platform. This step facilitates tracking the activity and ensures that a clear record of the changes made is maintained. Documenting the issue and actions taken is crucial for accountability and for reviewing or auditing the resolution process in the future.

The **operational ticket** is then assigned to the L2/L3 teams, who are responsible for managing the issue. These teams verify the conditions that trigger alerts and events and, if necessary, correct them. Their expertise ensures that the alert mechanisms are refined to function as intended, addressing any anomalies or inefficiencies identified during the analysis.

After the L2/L3 teams have attempted to correct the issue, the next step is to evaluate whether the alert has been successfully resolved. If the issue is resolved, the operational ticket is closed, marking the completion of the process. However, if the alert remains unresolved, the L2/L3 teams may involve the L1 team to manually open operational or security tickets. This manual intervention ensures that critical events are not overlooked due to ongoing alert failures, preserving the security system's overall effectiveness.

In cases where alerts or events cannot be easily corrected or are found to be redundant, the L2/L3 teams may consider temporarily or permanently deactivating them. This step prevents unnecessary system overloads and ensures that the focus remains on alerts that provide meaningful insights for security analysis. Deactivation decisions are made to maintain system efficiency without compromising monitoring

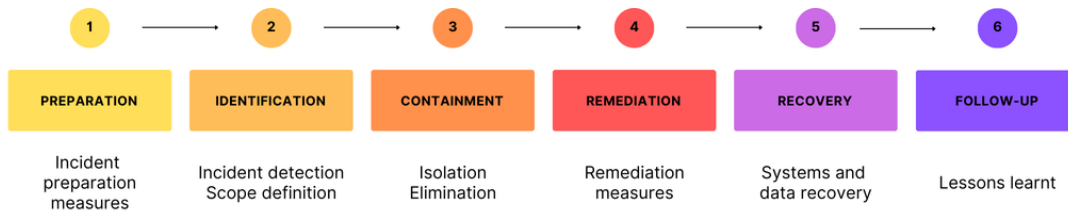


Figure 4.2: General phases of an Incident Response Plan

effectiveness.

The final step involves resolving and closing the operational ticket. Once the issue has been addressed or a decision has been made regarding the deactivation of alerts, the operational ticket is closed. This marks the conclusion of the alert failure management procedure and ensures that all relevant actions are properly documented for future reference.

4.9 Incident Response Plans

In a SOC, well-structured processes are defined for incident management based on its taxonomy (e.g., phishing, unauthorized login, data breach, etc.), so for each of them an Incident Response Plan (IRP) is created. An IRP is a formal document that clearly defines the procedures to follow for effectively managing incidents and developing a response.

The IRP includes the following steps, also shown in Figure 4.2:

- 1. Preparation:** Preparation is the step in the IRP process that focuses on establishing a robust incident response capability. This phase involves the formation of dedicated incident response teams, each with clearly defined roles and responsibilities. Additionally, the development of reporting and documentation procedures for various incident types is crucial. This includes creating standardized templates for incident reports and checklists to streamline the response process. Preparatory measures must also be adopted, an example is ensuring that all security solutions, such as firewalls and EDR software, are updated and fully working.
- 2. Identification:** The identification phase is critical for detecting and analyzing potential security threats to assess their impact on the organization. Continuous monitoring of systems and networks is essential for detecting unusual

activities or anomalies that may indicate an incident. The deployment of technologies such as IDS and SIEM solutions facilitates the aggregation and analysis of security data. Clear guidelines must be developed for logging, categorizing, and escalating incidents to ensure that critical events receive prompt attention.

3. **Containment:** Once an incident is identified, the next step is containment, which involves implementing measures to isolate the incident and mitigate its impact. Short-term containment strategies may include disconnecting affected systems from the network to prevent further damage, while long-term strategies could involve adjustments to access controls or network configurations to prevent recurrence. It is imperative that all actions taken during the containment phase are meticulously documented to maintain a clear record of decisions made and their outcomes, thus enhancing the overall response capability.
4. **Remediation:** The remediation phase focuses on addressing the underlying issues caused by the incident. The SOC undertakes a comprehensive analysis of the incident to identify its root causes and any exploited vulnerabilities. Remediation actions may include applying patches to vulnerable systems, enhancing security measures, and conducting user training to promote safe practices. In some cases, collaboration with external experts may be necessary to ensure a detailed and effective remediation process, enhancing the organization's overall security posture.
5. **Recovery:** The subsequent phase, recovery, includes restoring affected systems and data to ensure the resumption of normal organizational operations. This process may involve cleaning compromised systems, restoring data from secure backups, and verifying the integrity of systems before reintegrating them into the operational environment. It is essential to monitor systems closely during the recovery phase to identify any lingering issues or signs of potential compromise. Detailed documentation of the recovery process is vital for informing future updates to the IRP and enhancing organizational resilience.

6. **Follow-up:** Lastly, the follow-up phase emphasizes the importance of learning from the incident to prevent similar occurrences in the future. A detailed analysis of the incident, including a retrospective review, is conducted to identify effective strategies and areas for improvement. Forensic investigations may be employed to gain deeper insights into the incident's nature and impact. The findings from these analyses should be disseminated across the organization to raise awareness and improve overall security practices. Furthermore, training sessions based on lessons learned should be organized to reinforce best practices and strengthen the organization's security awareness.

In IRPs, various communication flows are also defined in the event of incidents, indicating which parties need to be informed based on the type of attack. Key stakeholders, such as the CISO, users, system owners, and Level 2 support teams, must be kept informed to ensure a coordinated response.

Practical example: Malware IRP

In this section, a practical example of IRP is reported. A SOC may handle an incident taxonomy associated with malware considering the following plan:

1. *Preparation:* Effective preparation for malware incidents involves several key measures. First, ensure all desktops and servers have updated anti-malware solutions. Incident response procedures should be reviewed, clearly defining roles, responsibilities, and escalation protocols for critical incidents. Developing a threat intelligence program is essential to identify emerging risks, vulnerabilities, and threats to the organization and its brand. Additionally, assign roles across relevant teams—IT, HR, legal, PR, and others—to ensure coordinated efforts during incidents. Finally, establish clear procedures for IT teams to receive actionable malware alerts, enabling a swift and effective response.
2. *Identification:* The goal of this phase is to detect malware incidents and determine their extent. Key steps include malware identification through the implementation of antivirus solutions and the detection of unusual system resource usage, such as abnormal CPU, disk space, or RAM consumption. Additionally, anomalous network traffic or suspicious connection attempts should

be monitored closely. A comprehensive global log monitoring process is essential for tracking system behavior. Finally, identifying the infected systems is critical to contain and address the malware efficiently.

3. *Containment*: The objective here is to mitigate the effects of the malware attack. Key actions include disconnecting infected devices from the network while ensuring they are not powered down or system data deleted without further investigation. Infected devices should be isolated, and suspicious traffic blocked. If possible, contact the owner of the affected product, service, or device. Additionally, force the disconnection of the affected user from all company platforms and temporarily suspend their access. If business-critical traffic cannot be blocked, ensure it is not a potential infection vector.
4. *Remediation*: The goal of remediation is to take decisive steps to halt the malware incident. This includes conducting malware analysis if necessary and identifying appropriate tools and methods for remediation, such as vendor-provided updates or patches, antivirus signature databases, and external support contacts or security websites. It's crucial to determine how the malware gained access to the systems. Additionally, if Personally Identifiable Information (PII) or Protected Health Information (PHI) has been compromised, involve the DPO and SOC to manage the breach appropriately.
5. *Recovery*: The objective of recovery is to return systems to normal functionality. This involves cleaning or reinstalling affected systems from scratch and applying all necessary updates. Once secured, reconnect the affected devices to the network and remove containment measures. Reactivate user accounts, ensuring password resets are performed. If required, restore any lost or compromised data to fully reinstate system operations.
6. *Follow-up*: To improve future response efforts, conduct forensic investigations if necessary to understand the full scope of the incident. Review the details of the attack and hold a training session to share lessons learned, ensuring that the incident informs improvements to current procedures and response strategies.

4.10 Examples: Alert Analysis

In this section, several examples of basic alert taxonomies are presented, which are critical for understanding the operational framework of a SOC. These **taxonomies** serve as foundational structures for categorizing and prioritizing security alerts, thereby facilitating a more organized and effective response to potential threats. By standardizing the classification of alerts, SOCs can enhance their ability to manage security incidents systematically and ensure that appropriate resources are allocated based on the severity and type of threat.

A well-defined taxonomy provides a common language that enables analysts, managers, and other stakeholders to discuss security alerts and incidents with clarity and precision. This shared understanding is crucial in high-pressure environments where rapid decision-making is necessary. Furthermore, by employing a taxonomy, SOCs can more effectively identify patterns and trends in security alerts, facilitating proactive measures and improving overall incident response strategies.

In the context of the simulation (Chapter 5) of SOC operations described in the subsequent sections, these examples of alert taxonomies will be instrumental in shaping the scenarios and workflows that are tested. They allow for a structured approach to analyzing how alerts are triaged, investigated, and resolved.

4.10.1 Endpoint Detection PUP Alert

Endpoint alerts can indicate the execution of Potentially Unwanted Programs (PUPs). It is crucial to consider the context in which the alert was generated, as the activity may not be harmful depending on the role of the user involved. For instance, if the user is a network specialist and the PUP is a network scanner, this may represent a routine business practice rather than malicious behavior.

Evaluating the specific type of PUP is fundamental. Even if tools like VirusTotal or other OSINT platforms do not classify it as harmful, the software could still be exploited for malicious purposes. For example, a network scanner could be used to gather sensitive information. In instances where the PUP is classified as *greyware*, which is a software that is not inherently malicious but carries potential risks, it is typically blocked by default. It may only be whitelisted if valid justifications are provided or if its use has been explicitly approved by the user.

Another important consideration is the management of PUPs that are portable software. In such cases, the EDR solution may quarantine a file that differs from the one executed after decompression. Therefore, it is essential to analyze both the programs and their associated information to ensure a detailed and accurate assessment in the final analysis.

When VirusTotal indicates matches from vendors, it is considered best practice to evaluate the security of the reported flag (confidence level) and the credibility of the vendor that flagged it. This assessment aids in determining whether the file genuinely poses a significant threat requiring immediate mitigation actions, or if it can be managed with a less invasive strategy.

Here follow the management steps for a case of this taxonomy:

1. The case is taken over by analyst on SOAR system.
2. Using Kibana, the alert is analyzed to extract pertinent information such as the user, User Agent, IP address, hash of the potentially harmful program, and so forth.
3. In this instance, the EDR platform (e.g. CrowdStrike Falcon) is employed directly for investigations.
4. The user is searched for within EDR to identify the associated activity.
5. A detailed analysis of the process tree leading to the PUP execution must be conducted.
6. Each quarantined file is examined, and the corresponding hashes are uploaded to VirusTotal.
7. Based on the VirusTotal score, the file is classified as either harmful or benign. If labeled as non-harmful, further research may be conducted using search engines, forums, and other resources for confirmation.
8. All relevant information must be documented in the SOAR analysis:
 - User Information: User name, email, activities, and timestamps.
 - Host Information: Host account details, host name, and host type.

- Sensor Information: All details regarding the sensor that triggered the alert.
 - PUP File Information: File name, company, version, signature, VirusTotal score, etc.
 - Alert Triggers: IOCs or Indicators of Attack (IOAs).
9. The involved user is subsequently notified through predefined channels (e.g., email, Freshdesk, Matrix).
 10. As a remediation procedure, it is advised to remove the PUP, instruct the user not to install such software in the future, and conduct a comprehensive antivirus scan of the device.

4.10.2 Alert Management Process for Suspicious File Execution

This alert is triggered following the detection of a file considered suspicious or exhibiting suspicious behavior by the EDR system. Upon detection, the EDR takes action to block the file in question, prompting the analyst to determine the actual purpose behind the file's execution.

Here follow the management steps for a case of this taxonomy:

1. The first step involves the engagement of SOC team, which can be triggered either by an alert from the SOAR system or through an email notification.
2. An analyst is assigned to the case and initiates preliminary operations for analysis.
3. The analyst conducts an initial assessment of relevant observables that may aid in the analysis, such as username, IP address, and file hash.
4. Within Kibana, the analyst looks for useful information related to the alert. This information is critical for understanding the reason behind the alert being triggered, and all relevant data is contained within the alert logs.
5. Typically, the analyst evaluates the process tree within the EDR to gain insights into the execution timeline of the flagged file. The blocked file is further analyzed on VirusTotal using its hash, usually a SHA value.

6. The analyst seeks to understand the basis on which the EDR determined that the file in question is malicious. Online research can be beneficial in this context to justify findings.
7. The analyst then requests the involved user to recognize the file and the associated activity, facilitating a clearer understanding of the context.
8. Finally, the analyst draws conclusions regarding the alert and determines whether it is a true positive, indicating an actual threat, or a false positive.

4.10.3 Failed Multi-Factor Authentication for User Alert

The alert regarding *Failed Multi-Factor Authentication for User* indicates that a user has attempted to register a device for Multi-Factor Authentication (MFA) unsuccessfully. This type of alert may arise from various attempts by the user to access or register a device for MFA.

The user could be trying to complete the login process on multiple devices or enable MFA as part of a security enhancement for their account. Such attempts might reflect normal behavior, such as a user attempting to properly configure their MFA settings across different devices to strengthen account security.

Additionally, events related to `UserAccountModified` may be observed, as the user might also attempt to alter account information during the MFA setup process. These actions could be innocuous or indicative of malicious intent, highlighting the importance of evaluating the context and sequence of events to ascertain the user's intentions.

It is crucial to note that in platforms like Microsoft, a login attempt is considered failed until the entire MFA procedure is successfully completed. This detail is significant during event analysis, as a high number of failed attempts could suggest a potential attack or unauthorized use of the account.

In managing this type of alert, it is advisable to thoroughly examine access and account modification logs, identifying any suspicious or unusual patterns that may indicate security risks. Appropriate responses might include temporarily blocking the account, implementing additional security checks, or notifying the user to verify ongoing activity, thereby ensuring the protection of the account and associated data. Here follow the management steps for a case of this taxonomy:

1. The case is taken over by an analyst on the SOAR system.
2. Within Kibana, the alert is analyzed to gather pertinent information such as user details, User Agent, IP address, and other relevant data.
3. When analyzing logs in the *Discover* section related to the corporate tenant within the correct timeframe, the `EventAction` field is crucial as it effectively clarifies the user's intended action.
4. For alerts related to Microsoft Defender, Azure error codes can be researched online to gain a better understanding of the user's associated actions.
5. After obtaining all relevant information from Kibana, including user actions, timestamps, devices, User Agent, and IP information, data is transferred into the SOAR analysis.
6. Depending on whether the analyst determines the activity to be harmful or not, the user may be contacted to acknowledge the activity or to classify the alert as a false positive.

4.10.4 Suspicious URL Click Alert

This type of alert is triggered when a user clicks on a URL that has been flagged as suspicious and potentially malicious by the EDR system. Even if the email containing the URL has been quarantined in advance, the user might still choose to click on the link, resulting in the generation of an alert by the EDR.

When analyzing malicious domains, it is essential to recognize that these domains often operate on a *fire-and-forget* basis, being used for brief periods before being abandoned or replaced by new ones. This temporal aspect can provide significant clues regarding their potential danger.

Another critical point concerns the analysis of reported IP addresses. It is generally beneficial to examine multiple sources and reports to accurately assess whether an IP address is indeed malicious. A single report may not provide sufficient evidence to definitively determine the nature of an IP, even if it comes from a reliable source. In the context of analyzing suspicious emails, the `Return-Path` field in email headers is particularly important. This field indicates the address to which delivery failure

notifications are sent. Significant discrepancies between this address and the one displayed as the sender can suggest attempts at obfuscation by attackers, signaling potential malicious intent.

Additionally, the *Internet Message ID*, a unique identifier assigned to each email message, is crucial for tracking and reconstructing messages in forensic analysis. This tool aids in monitoring the message's path and identifying any manipulations during transit.

Analysts can further refine their investigations by considering *Authentication Pass* checks, which provide information on the outcome of the user's authentication process, indicating whether the user successfully completed all authentication stages during their interaction with the suspicious URL or email.

Finally, implementing Zero-Hour Auto Purge (ZAP) for suspicious emails represents a proactive defensive action. This automatic mechanism identifies and isolates potentially harmful emails before they reach users' inboxes, thereby limiting exposure to cyber risks.

Here follow the management steps for a case of this taxonomy:

1. After taking on the case, the analyst begins by evaluating access logs for the 24 hours following the click on the suspicious URL, utilizing Kibana and applying filters related to the involved user. This step aims to identify any unusual activities, such as logins from unusual geographic locations or at odd hours.
2. The analyst employs OSINT tools to analyze the IP address used for access.
3. The analyst reconstructs and evaluates the timeline of suspicious activities leading up to the click on the suspicious URL to gain a better understanding of the context.
4. The analysis of the suspicious email focuses on the sender and any attachments. This step is crucial for understanding the email's actual purpose and IOCs.
5. The analyst uses OSINT tools like URLscan and AppAnyRun to analyze the sender and verify if there are any reports labeling them as malicious. Additionally, the sender's domain can be analyzed on CentralOps.net for further insights.

6. Attachments from the suspicious email are examined using sandbox environments or tools like VirusTotal to check for malware or other harmful content. If necessary, and if the file cannot be downloaded, the analyst may request the involved user to provide the attachment through an alternative channel.
7. If there are other URLs present in the suspicious email, they are analyzed to determine their nature and whether they are associated with previously known malicious events.
8. All information obtained from various analyses, including the Internet Message ID and other relevant details, is documented within the analysis, following the correct timeline within the SOAR system.
9. Following a confirmed phishing incident, the SOC analyst may recommend the following actions to the customer:
 - Block the malicious domain to prevent further access.
 - Change user credentials, as they may have been compromised following a successful phishing attempt.
 - Remove all sessions related to the involved user and device.
 - Delete the involved email to eliminate a confirmed threat factor.

4.10.5 Stolen or Lost Device Management

This case pertains to the management of a stolen or lost device. It is crucial to erase all sensitive data on the device to prevent the disclosure of confidential information pertaining to the user or the organization.

At the core of this management process is the CMDB, which is as previously explained an essential tool that serves as a comprehensive repository of hardware and software information related to devices registered within the organization. The CMDB not only catalogs these data but also plays a critical role in correlating various hardware and software components.

Continuous monitoring of devices is fundamental in this context. By collecting data on activities performed with the device, it is possible to suggest appropriate remediation measures. This process often relies on the analysis of past cases involving

similar activities, allowing for more targeted and effective recommendations.

In this scenario, the responsibilities of the analyst include identifying the issue, gathering and conducting a thorough analysis of information regarding the device, and proposing remediation interventions. It is important to note that while the SOC analyst provides recommendations, the actual implementation of security measures remains the responsibility of the customer's technical staff.

Additional information sources, such as reports filed with law authorities in the case of theft or loss, can provide valuable details for the overall analysis of the case. This information can contribute to a more comprehensive understanding of the circumstances and support the formulation of more effective response strategies.

While remote wipe functionality is an important protective measure, it should not be considered a completely foolproof solution. In scenarios involving sophisticated tampering, there is a risk that the software component responsible for receiving the wipe signal may be disabled, for instance, by removing the hard disk or battery.

Here follow the management steps for a case of this taxonomy:

1. The lost or stolen device is reported to the SOC by the customer. The reporting party may be the user themselves or a higher-level user, such as the responsible manager or the CISO of the customer company. Reports are typically submitted via email or through a helpdesk system.
2. The analyst retrieves all relevant information about the user's device from the CMDB, which could be Azure in the context of the Microsoft ecosystem.
3. As a precautionary measure, the analyst requests a *remote wipe*, which sends a command to the device to restore factory settings and erase all data deemed critical or sensitive to the user and the customers' organization.
4. All login operations following the report of the lost or stolen device are evaluated.
5. The device must be blocked from VPN access to prevent it from accessing protected assets and resources.
6. The analyst notifies the customer's Regional Service Desk (RSD) of the collected information and requests additional relevant details while suggesting

further remediation activities in accordance with the previously established IRP:

- Determine whether the device has an encrypted disk.
- Ask whether the device contains sensitive data.
- Suggest marking the device in the CMDB as *stolen* or *lost*.
- Recommend the replacement of the device as soon as possible.
- Suggest removing the MAC address from any whitelist.
- Propose placing the device on a blacklist for Wi-Fi and Network Access Control (NAC).
- Recommend changing passwords for any networks accessible from the device.
- Suggest enabling device tracking and periodically checking for its activation.
- Advise disconnecting from any active sessions on the device.

7. Based on the responses received from the RSD, the analyst documents all actions taken in the analysis to maintain a complete history of the case.

4.10.6 Unfamiliar Sign-In Alert Management

The alert is triggered by the SIEM system, which analyzes logs to identify suspicious activities based on predefined rules. This particular alert occurs when an access attempt exhibits unusual attributes and properties, such as geographic anomalies e.g., a login from a location outside Italy for a company that operates exclusively within the country. Alternatively, the alert may be raised by an EDR system if it detects unusual behavior.

The primary goal in this situation is to determine whether the user intentionally accessed the account under atypical circumstances. This involves evaluating access patterns from foreign countries, analyzing potential VPN usage, and assessing whether malicious activity has occurred.

It is important to note that geolocation based on IP addresses can be imprecise due to the dynamic nature of IP assignment. This limitation should be taken into

account when interpreting alerts based on geographical anomalies. Additionally, identifying the event provider that generated the log e.g., in Office 365, can provide valuable context for the analysis.

A specific signal to consider is the **Unfamiliar Sign-In**, which indicates a successful access attempt with parameters that differ from typical user behavior or predefined patterns. These parameters may include the use of unrecognized devices, atypical access locations, or unusual access times.

A suspicious pattern to monitor is a series of login attempts occurring in rapid succession, often just milliseconds apart. This activity could indicate the use of automated scripts or tools for unauthorized access attempts, distinguishing it from normal human behavior.

For a more thorough analysis, it is useful to gather information about the application and device used for login. In particular, the use of a false user agent can be a strong indicator of malicious activity, as attackers often attempt to conceal information about the actual device. OSINT tools can be employed to verify the authenticity of the user agent.

If the IP address belongs to a wireless network, this may suggest the absence of a VPN. In such cases, analyzing a broader range of IP addresses for a more extensive evaluation can be beneficial. Similarly, the name of the ISP can serve as a useful filter for the final analysis, providing additional clues about the nature and origin of the access.

Here follow the management steps for a case of this taxonomy:

1. The analyst takes ownership of the case in the SOAR system and begins the analysis.
2. A precise taxonomy is chosen for the case. Typically, this case is classified under the *Unprivileged Account Compromise* taxonomy in the SOAR system.
3. In Kibana, the tenant related to the company is selected to search for relevant logs.
4. The analyst reviews key fields in the alert log e.g., action performed such as `UserLoggedIn`, IP, user agent, email, etc.. Information that explains the cause

for triggering the alert, which may be **Unfamiliar Sign-In** or **Unfamiliar Location**, is particularly useful.

5. After navigating to the *Discover* section in Kibana and setting the correct timestamp for the case to narrow down the number of logs to analyze, additional filters are applied, and useful fields are added to the log list for easier comparison. The aim is to understand what the user was attempting to do, starting with the log that triggered the alert. It may also be useful to conduct separate searches for IP and user information such as username or email.
6. In the case of alerts related to Microsoft Defender, Azure error codes can be searched online to better understand the user's actions. Additionally, error messages in the logs may provide further insights.
7. Based on the analysis of the previously gathered information, the analyst verifies the unusual behavior of the user.
8. If unusual behavior is suspected, the analyst should check IP information using various OSINT tools:
 - AbuseIPDB for IP information.
 - Scamalytics for IPs, proxy and server information and so on.
 - Shodan for server information and CVE vulnerability checks.
 - Spur to check for public VPNs or proxies.
 - AlienVault to perform checks on server.

The activities performed immediately after a successful login must also be analyzed.

9. The analyst should communicate the results of the analysis directly to the user using a predefined engagement method e.g., email, reference numbers, Matrix or Freshdesk.
10. Recommended Remediation Actions:
 - **Session Revocation:** This action helps mitigate the risk of unauthorized activity and ensures that only legitimate users can reaccess their accounts.

- User Account Password Reset: Invalidating the current password blocks potential malicious actors who may have obtained compromised credentials.
- MFA Activation: Enabling Multi-Factor Authentication adds an extra layer of security.

Chapter 5

SOC Implementation and Simulation

5.1 How to implement a SOC

As introduced in the first part of the study, a SOC is a versatile entity composed of various tools, processes, and people, predominantly analysts, alongside managers and other roles that may not be directly associated with cybersecurity. The effectiveness of a SOC relies on the coordination of these diverse tools and processes, which must adhere to a defined workflow to manage security alerts. This workflow subsequently addresses any alerts identified as confirmed security incidents.

Each SOC possesses unique requirements and may prioritize different types of tools when selecting SIEM and SOAR systems. These systems may be proprietary or open-source, depending on the specific functionalities required within the given context. Furthermore, it is essential to specify that a SOC is not a static entity; it is inherently dynamic, evolving in response to the changing demands of customers, particular needs, and the characteristics of the monitored infrastructures.

Despite this inherent dynamism, a SOC must maintain a level of independence from the specific customer infrastructure it monitors. This independence is crucial to ensure that the SOC can effectively support a wide array of IT tools that are already in use within the client's environment. As such, a working SOC is characterized not only by its adaptability but also by its ability to integrate seamlessly with diverse technological ecosystems, thereby enhancing its operational efficacy in the face of evolving cybersecurity threats.

Considering the tools introduced in previous sections, open-source tools may be

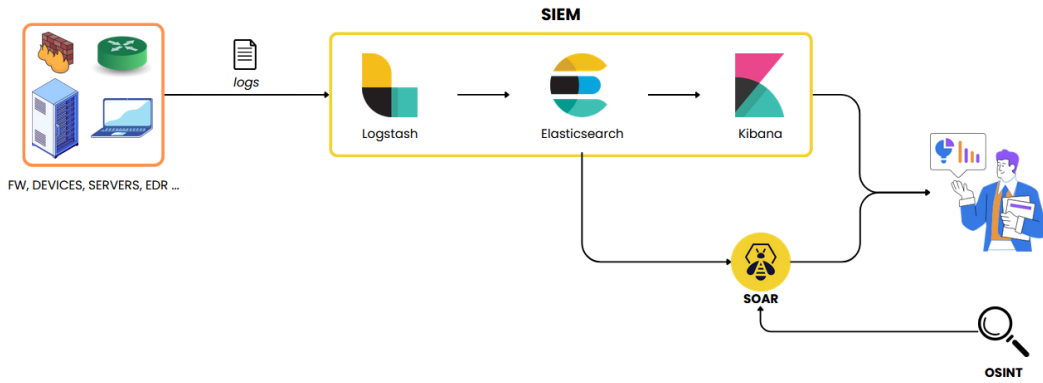


Figure 5.1: Basic SOC Implementation

effective when implementing SOC core elements because they are cost-effective, customizable, and flexible. They allow organizations to build up their customized solutions based on specific needs without expensive licenses, while benefiting from community-driven updates and transparency. This adaptability makes open-source tools ideal for tasks like threat detection, incident response, and intelligence sharing in a SOC environment. Based on this concept, tools are used to implement a SOC following the simple architecture presented in Figure 5.1.

5.2 Simulation Description and Purpose

The proposed simulation, which is accessible in a GitHub repository [1], focuses on the utilization of key tools employed in SIEM and SOAR functions within a SOC. The goal is to demonstrate, through the analysis of operations and performance indicators, the feasibility of implementing a SOC with open-source tools and how optimizations implemented in SOC workflows can significantly enhance the quality of service provided. Specifically, the tangible benefits of these optimizations are highlighted in terms of operational efficiency, response speed, and accuracy in threat management.

5.3 Simulation Tools

To run the simulation, a Docker Compose configuration file has been used to define and manage the infrastructure. This setup establishes an integrated SOC environment by orchestrating multiple services through Docker Compose. Each service

operates in its own Docker container, representing essential tools for monitoring, analyzing, and managing security incidents.

The configuration combines a range of open-source tools, each selected to facilitate efficient incident analysis and response. Detailed descriptions of each component and its interactions are outlined below, making this setup replicable for future simulations and further research.

5.3.1 Primary Tools

This section provides an overview of essential tools utilized within a SOC environment for effective log management, data visualization, incident response, and device inventory management. Each tool fulfills a distinct role in enhancing security monitoring, analysis, and response capabilities, collectively establishing a comprehensive framework for security operations.

The core tools, which have been detailed in Section 3.3, include the ELK Stack, which supports centralized logging and visualization; Cortex, used for threat intelligence enrichment; TheHive, serving as the SOAR platform; and NetBox, which acts as CMDB. This setup provides a comprehensive security monitoring and response ecosystem suitable for modern SOC environments as considered for this study.

Here follow details about the functionalities and contributions of each tool to the SOC infrastructure.

The ELK Stack v7.10.1, as the latest open-source SIEM platform version, offers comprehensive monitoring and log management capabilities that are critical for the effective operation of a SOC. The core components of the ELK Stack include Elasticsearch, Kibana, Logstash, Filebeat, and Metricbeat, each serving distinct functions to ensure efficient log collection, data analysis, and system monitoring.

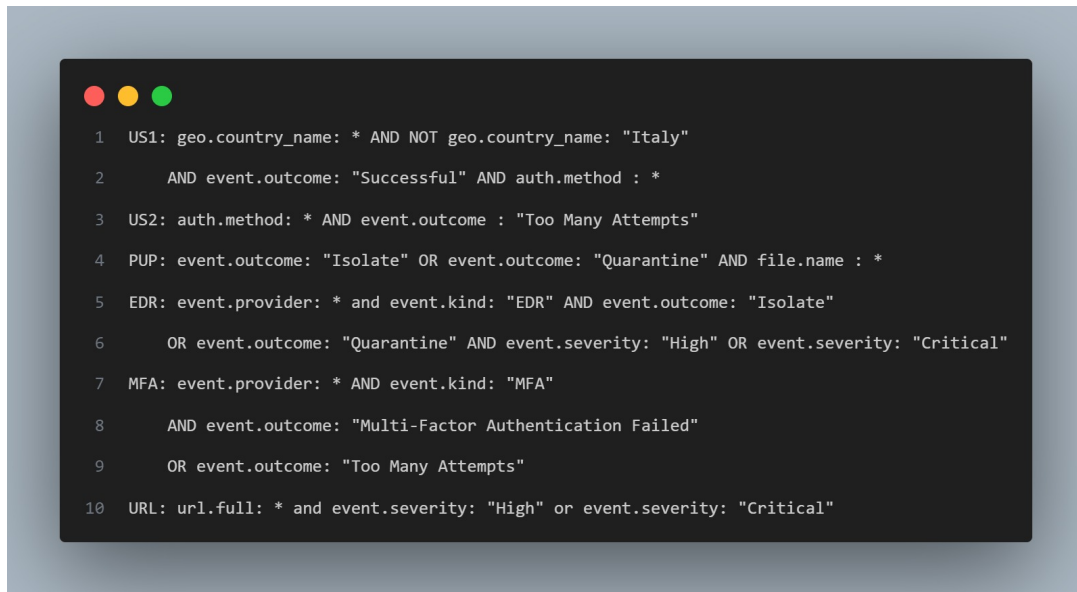
Elasticsearch is configured as a single-node instance with security features enabled, serving as the central monitoring and analysis hub within the SOC. This setup manages and indexes log data, operating as the backend for components such as TheHive, Cortex, Kibana, Logstash, and Filebeat. Each of these components interacts with Elasticsearch through a secure authentication process that uses designated usernames and passwords, ensuring controlled access. Data persistence is

maintained through a dedicated Docker volume, and initial setup requires configuring reserved user credentials for secure operation.

Kibana, another essential component in the SOC environment, provides an advanced graphical interface for visualizing and analyzing data indexed by Elasticsearch. It authenticates securely with Elasticsearch through a dedicated user, which is called `elastic` by default in the simulation, supporting secure data access with encryption for data in transit and at rest. Notably, this SOC configuration utilizes spaces or namespaces within Kibana, creating simulated working environments that are especially useful in a multi-tenant SOC framework. For example, the `big-company` namespace is configured to simulate a high-profile client's environment, allowing SOC analysts to concentrate solely on logs and data flows pertinent to that entity. Kibana further enhances security monitoring by supporting detection rules that trigger real-time alerts for specified events. In the simulation, in compliance with the example cases reported in section x, the following detection rules have been provided:

- *US1*: Alerts on successful authentication events from outside Italy to detect unauthorized access.
- *US2*: Monitors multiple failed authentication attempts to identify potential brute-force attacks.
- *PUP*: Activates when a file is isolated or quarantined, indicating potential malware.
- *EDR*: Detects high-severity EDR events, such as isolation or quarantine.
- *MFA*: Monitors failed multi-factor authentication attempts to detect possible account compromise attempts.
- *URL*: Triggers when URLs associated with high or critical severity events are detected, flagging potential threats from dangerous web sources.

Logstash functions as the SOC primary log transformation and normalization tool, tasked with collecting and processing log data from multiple sources according to the Elastic Common Schema before forwarding the data to Elasticsearch for indexing. The Grok language is used to parse logs, and data persistence is achieved

A screenshot of a terminal window showing ten numbered lines of detection rules. The rules are: 1. US1: geo.country_name: * AND NOT geo.country_name: "Italy"; 2. AND event.outcome: "Successful" AND auth.method : *; 3. US2: auth.method: * AND event.outcome : "Too Many Attempts"; 4. PUP: event.outcome: "Isolate" OR event.outcome: "Quarantine" AND file.name : *; 5. EDR: event.provider: * and event.kind: "EDR" AND event.outcome: "Isolate"; 6. OR event.outcome: "Quarantine" AND event.severity: "High" OR event.severity: "Critical"; 7. MFA: event.provider: * AND event.kind: "MFA"; 8. AND event.outcome: "Multi-Factor Authentication Failed"; 9. OR event.outcome: "Too Many Attempts"; 10. URL: url.full: * and event.severity: "High" or event.severity: "Critical".

```
1 US1: geo.country_name: * AND NOT geo.country_name: "Italy"
2   AND event.outcome: "Successful" AND auth.method : *
3 US2: auth.method: * AND event.outcome : "Too Many Attempts"
4 PUP: event.outcome: "Isolate" OR event.outcome: "Quarantine" AND file.name : *
5 EDR: event.provider: * and event.kind: "EDR" AND event.outcome: "Isolate"
6   OR event.outcome: "Quarantine" AND event.severity: "High" OR event.severity: "Critical"
7 MFA: event.provider: * AND event.kind: "MFA"
8   AND event.outcome: "Multi-Factor Authentication Failed"
9   OR event.outcome: "Too Many Attempts"
10 URL: url.full: * and event.severity: "High" or event.severity: "Critical"
```

Figure 5.2: Detection Rules Samples in Kibana

through a dedicated volume, with optimized Java configurations supporting efficient resource usage within the SOC infrastructure. Logstash pipeline is configured in a file called `logstash.conf`. In the *Input stage*, logs are received from Beats, so Filebeat or Metricbeat, through port 5044, feeding the data into the Logstash pipeline for processing. The *Filter stage* then structures and enriches the log data for easier analysis. JSON data is parsed from the message field, and Grok patterns extract key information, such as the original event details, IP addresses, ports, and protocols, placing them into standardized fields like `[source]`, `[destination]`, and `[network][protocol]`. Additional details, such as usernames, user agent info, and operating system details, are similarly parsed and organized. Fields for event outcomes, severity levels, and detection methods are translated and standardized, ensuring that crucial information about authentication, file and process details, and action outcomes are clearly labeled. Redundant fields are removed to streamline the log entries. In the *Output stage*, the structured and enriched data is sent securely to Elasticsearch, with each log entry indexed by date (e.g., `logstash-%{+YYYY.MM.dd}`) for efficient querying and time-based searches. This final step stores the logs in a format optimized for analysis, making data readily available for monitoring, alerts, and reports.

Filebeat is configured as a lightweight logging agent responsible for collecting log data and securely forwarding it to Logstash for parsing. This `filebeat.yml` configu-

A screenshot of a terminal window with a dark background and light-colored text. The terminal shows a configuration snippet for the output stage in a logstash.conf file. The code is as follows:

```
1 output {
2   elasticsearch {
3     hosts => ["http://elasticsearch:9200"]
4     index => "logstash-%{+YYYY.MM.dd}"
5     user => "elastic"
6     password => "changeme"
7   }
8 }
```

The terminal window has three colored window control buttons (red, yellow, green) in the top-left corner.Figure 5.3: Output Stage in `logstash.conf` File

ration file sets up Filebeat to monitor log files located in a configured directory, which are populated with generated logs from the script, forwarding them to Logstash on port 5044. Custom fields are included to specify log types, and Kibana integration is enabled to allow data visualization within the `big-company` namespace, with dashboards set up for automatic configuration.

Metricbeat contributes vital monitoring data related to system performance, capturing metrics on CPU usage, memory utilization, and filesystem status directly from host operating systems. It forwards this data to Elasticsearch, supporting the SOC's overall monitoring and performance analysis capabilities.

Cortex v3.1.1 is an advanced tool that enhances SOC capabilities by providing automated analysis and response functions. As part of the SOAR platform, Cortex conducts actions on observables within case data. Its functionality integrates directly with Elasticsearch. Cortex is configured with designated directories for analyzers and responders, which streamline security workflows and support automated responses across various observable types. The analyzers and responders considered in this simulation include specific scripts which are **AbuseIPDB** for IP address reputation checks, **URLHaus** for domain, URL, hash, and IP analysis, **URLscan.io** for domain and URL threat intelligence, and **VirusTotal** for both

file and URL scanning. These analyzers extend Cortex's analytical range, enabling security teams to efficiently assess and classify security incidents based on real-time OSINT. Cortex configuration is written in `application.conf` file which includes URI and authentication information related to Elasticsearch and locations on where to find analyzers and responders; for example, analyzers scripts are found in directory `/opt/cortex/analyzers/`.

TheHive v4.1.24 operates as the SOC's SOAR platform, supporting centralized security incident management. Integrated with Cortex for automated incident analysis and response, TheHive uses Elasticsearch for case storage and retrieval. It is configured with dedicated volumes for data persistence, and it initiates only once Elasticsearch and Cortex are operational. TheHive records cases opened following SIEM alerts, enabling analysts to conduct investigations, analyze logged activities, and perform automated responses on observables associated with each case. Its main configuration happens, similarly to Cortex but in a different file, in the `application.conf` file. In this simulation, Cortex is integrated with TheHive by enabling the Cortex connector module and specifying details for a Cortex server instance in the configuration file. The configuration includes an entry for a single server labeled `Cortex-Server` and defines the connection endpoint at `http://cortex:9001`. Authentication for secure access to Cortex is managed using a bearer token authorization method, with an authentication key specified. This setup allows TheHive to leverage Cortex's analysis and response capabilities, creating a seamless, unified environment for threat intelligence and incident response within the SOC.

NetBox v2.9.1 acts as the SOC's CMDB, maintaining information on all network devices within the SOC and client environments and more. NetBox is crucial in the simulation for identifying devices linked to private IP addresses, enhancing the SOC's network visibility and contextual analysis capabilities.

5.3.2 Supporting Tools

To conduct the simulation, several supporting tools have been used, which are also suitable for replicating all analyzed processes in future work.

Docker, an open-source platform, plays a central role by hosting SOC services in

lightweight, portable containers that encapsulate applications and their dependencies. This containerized setup not only ensures consistent environments across development, testing, and production but also enables smooth communication among services. Docker Compose, an integral part of this setup, facilitates the management of multi-container applications through a single YAML file, thereby streamlining the orchestration of complex stacks and enabling rapid deployment and scaling of services.

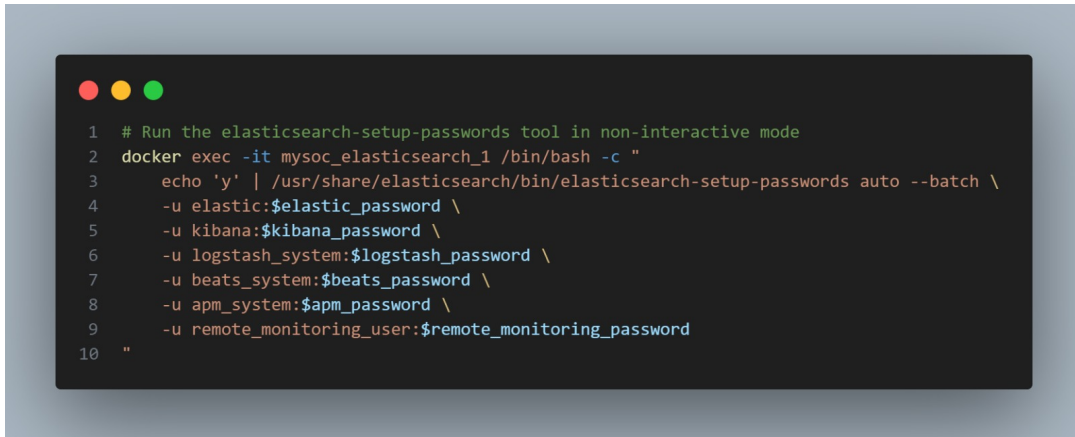
In addition, **Python 3**, along with its package manager **pip3**, is employed to execute automation scripts designed for generating logs for the SIEM system, automating case creation from SIEM to SOAR systems, and measuring performance metrics derived from simulations. Python is further used to develop tools that assist in analysis and automated response, enhancing various aspects of incident management and response automation.

5.4 Simulation Setup

This section outlines a detailed guide for establishing and configuring key tools within a simulated SOC environment. By leveraging Docker for efficient deployment and management, the setup integrates Elasticsearch, Kibana, Cortex, TheHive, and accompanying scripts, with each tool fulfilling essential roles in data indexing, visualization, incident detection, and response coordination. The objective is to develop a unified and secure configuration that allows seamless integration across these tools, supporting robust threat monitoring, data analysis, and incident response capabilities. This type of configuration improves the SOC's operability and improves the ability to test various operations within its operational flows.

To configure **Elasticsearch** within a Docker environment, first access the console and navigate to the `bin` directory.

Here, execute the command `./elasticsearch-setup-passwords` to initiate the password setup process in interactive mode or it is possible to execute an automated script shown in Figure 5.4 to streamline this process. It is advised to set a default password consistently across users to streamline credential management and maintain security uniformity within the simulated SOC environment, acknowledging

A terminal window with a dark background and light text. The text is a Docker command to run the elasticsearch-setup-passwords tool in non-interactive mode. The command is:

```
1 # Run the elasticsearch-setup-passwords tool in non-interactive mode
2 docker exec -it mysoc_elasticsearch_1 /bin/bash -c "
3   echo 'y' | /usr/share/elasticsearch/bin/elasticsearch-setup-passwords auto --batch \
4   -u elastic:$elastic_password \
5   -u kibana:$kibana_password \
6   -u logstash_system:$logstash_password \
7   -u beats_system:$beats_password \
8   -u apm_system:$apm_password \
9   -u remote_monitoring_user:$remote_monitoring_password
10 "
```

Figure 5.4: Script to configure Elasticsearch user automatically

that this practice applies specifically to simulation purposes rather than production environments.

Within **Kibana**, the setup starts with defining a tenant, called a *space* (e.g., "Big Company"), and defining an index pattern. The space can be created through **Stack Management > Spaces**, while the index pattern is configured after indices are created by the simulation and by navigating to **Stack Management > Index Patterns**, and creating an index pattern with the prefix `logstash-*`. This setup aligns with the simulation environment, where each generated log is directed to an index named `logstash-+YYYY.MM.dd`, following a standardized naming convention based on the current date. The defined index pattern aids in organizing and visualizing log data effectively. To enhance detection capabilities within the SOC, pre-existing detection rules explained in Section 5.3.1 may be incorporated by accessing the **Detection > Manage Detection Rules** section, and importing rules from a designated file path.

The setup of **Cortex** begins with the creation of a dedicated database automatically from the UI, followed by configuring an administrative user responsible for oversight within the SOC environment. Once the database and user configurations are complete, a new organization should be established with clearly defined settings to facilitate structured roles and responsibilities. Within this organization, an integration account is created to support seamless connectivity with other SOC components. Subsequently, an API key is generated for the integration account and added to the Cortex configuration file under the authentication settings. To apply these configurations, the Docker containers are restarted by executing `docker-compose down`



Scallicgraph	0.1.0-SNAPSHOT
TheHive	4.1.24-1
Play	2.8.13
CORTEX	Cortex-Server - 3.1.1-1 (OK)

Figure 5.5: Cortex successful integration with TheHive

followed by `docker-compose up -d`, ensuring the implementation of the changes and completing the Cortex setup process. The analyzers used in the simulation are configured in Cortex using the UI, which mainly requires the corresponding OSINT service API key., this will be discussed in Chapter 6.

TheHive configuration begins with accessing the platform using default administrative credentials, followed by the creation of a new organization. Relevant organizational details are specified to establish structured user and role management. Within the organization settings, custom observable types, which are `username` and `filepath`, are defined to monitor specific data artifacts aligned with SOC requirements, thereby enhancing the visibility of pertinent information. Additionally, the establishment of an `org-admin` role allows for comprehensive organizational data management and access to essential tools for incident response.

The **NetBox** environment is configured separately from the main compose file, and once pulled and started, the administrator account should be set with `manage.py createsuperuser` which should be executed within the main NetBox container. There are several components to be created in order to set up NetBox in a way that is ready for the simulation, so a setup script called `setup_netbox.py` is used. This script facilitates the automated setup of the NetBox environment by setting up essential network and device metadata, streamlining configuration for IT infrastructure management. It defines a logical structure for a simulated organization, *Big Company* by creating entities such as sites, device roles, manufacturers, device types, and devices with distinct roles (e.g., domain controllers, web servers, user devices, and sandboxes). Each device is configured with network interfaces, and

specific IP addresses are assigned to these interfaces. Moreover, the script updates each device with its corresponding primary IPv4 address, ensuring accurate network representation and asset tracking.

5.5 Base Handling Process

The Python script `log_simulation.py` generates simulated security log entries for various types of alerts and logs them in JSON format to a specified log file. The script defines five alert types: *Unfamiliar sign-in*, *Potentially Unwanted Program (PUP)*, *Endpoint Alert*, *Failed MFA for User* and *Malicious URL Click Detected*, each assigned an arbitrary probability. Data for generating log entries is loaded from a JSON file, which contains fields such as public and private IP addresses, usernames, device IDs, and security event metadata. The `generate_log_entry` function randomly selects alert types and populates unique details for each alert, such as IP addresses, protocols, device information, file details, and geo-location data. The `log_event` function then logs these generated entries at regular intervals. Through repeated execution in a timed loop, the script continuously creates several log entries that simulate real-world security alerts, which could be useful for testing log analysis or monitoring systems.

Filebeat then retrieves generated JSON logs and ships them towards Logstash on port 5044. This will normalize them into ECS and store them in Elasticsearch, following the process that has been previously explained.

In Kibana, logs can be analyzed by filtering them by modifying the timerange, selecting specific fields and associated values; KQL is highly used in this context. Without filters, the Discover section appears as shown in Figure 5.6. The detection rules mentioned above are periodically executed, considering an arbitrary period that is typically 5 minutes, on the logs and in case of a confirmed match, an internal security alert is generated in Kibana accessible in **Security > Detection**. In this section, a visualization similar to Figure 5.6 is shown but distributed considering types of alerts.

The analyst then manually opens an investigation case in TheHive reporting information associated with the alert. Next, the information needed is the observables and tasks to be able to perform performance measurements later; both are man-

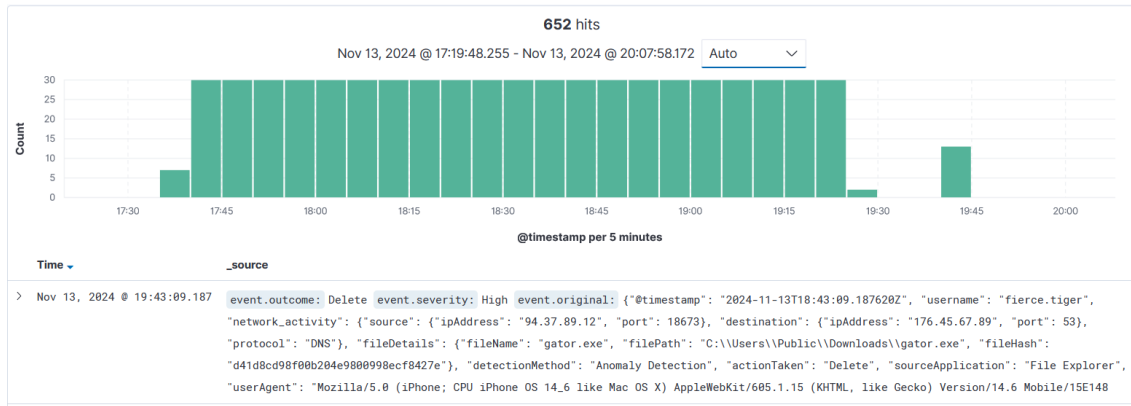


Figure 5.6: Initial Discover section in Kibana

ually created within the case environment. In detail, observables are typically IP addresses, file names, hashes, URLs and usernames, used as a correlating factor between different cases and as information to be enriched using public OSINT sources. The tasks that must be created are the following:

1. **Takeover:** The case is correctly taken over by the SOC analyst.
2. **Start Analysis:** The analyst focuses solely on the analysis of the case and for now on it actually started.
3. **Containment:** The analyst successfully applied containment actions if it is required. This task is optional.
4. **Remediation:** The analyst successfully applied and/or suggested remediation actions associated with the case.

The analyst begins the investigation by analyzing mainly logs in Kibana and previous cases associated with the observables of the case currently being analyzed. The main operations associated with a preliminary L1 analysis are those of identifying the appropriate time range, the study and enrichment of the various observables, and consequently the analysis of the network or behavioral flow associated with an asset or a user. Enrichment occurs manually by considering the OSINT resources mentioned above. The analysis ends by manually closing both the alert on Kibana and the case on TheHive; on the latter, it is also necessary to specify further comments and the final outcome, which can be a true positive or a false positive.

The next section introduces automation operations that optimize the flows previously described.

Chapter 6

Optimizations in SOC operations

Considering the existing SOC infrastructure and deployed tools, a series of automations could be implemented to optimize workflows and reduce the time needed to handle various security cases.

This chapter examines specific optimizations within the SOC simulation environment, aimed at enhancing analysis efficiency and supporting the analyst in streamlining operational workflows.

Key improvements include the automated opening of cases on the SOAR when alerts trigger from the SIEM, which reduces the time required for manual case creation and allows analysts to focus more on investigative tasks. Furthermore, data enrichment associated with cases is expedited through automated searches leveraging OSINT tools, enabling quicker gathering of contextual information. Finally, the integration between SOAR and SIEM platforms facilitates seamless tracking of monitored activities, providing a clearer view of the nature and flow of operations associated with alerts. These optimizations collectively aim to reduce management overhead, allowing for a greater volume of analyses to be completed within a given timeframe, thereby improving the SOC's overall operational efficacy.

6.1 Development

Starting from the initial management process of the various security alerts, some optimizations have been developed using Python scripts in order to automate some operations.

A Python library called **TheHive4Py** [10] has been used to link Elastic and TheHive so it can be exploited to link SIEM and SOAR.

A script called `elastic2hive.py` has been designed and developed to integrate an Elastic Stack and TheHive to automate the detection and case management of security alerts as previously introduced. The code continuously queries Elasticsearch for open alerts, updates their status, creates a detailed case payload, and submits it to TheHive for further analysis.

The script initiates by querying Elasticsearch for open alerts. Using a specific query, it searches for alerts with a status of *open* in the `.siem-signals-big-company` index which is what indexes the alerts inside the SIEM. Authentication is managed via `HTTPBasicAuth`, and the query results are processed to extract relevant logs. The method incorporates exception handling to manage any errors that may arise during the request.

Once an alert is retrieved, the script updates its status in Elasticsearch to *in-progress*. This step ensures that alerts are marked as being processed, reducing redundancy and enabling efficient workflow management. The alert's unique identifier (`_id`) is used to locate and modify its status.

The script generates a structured payload for creating a case in TheHive. The payload includes a detailed description derived from the log's metadata, such as timestamps, user information, IP addresses, and associated rules. It also constructs a link to Kibana for visualizing the alert within a specific timeframe; everything is shown in Figure 6.1. This section maps the severity of the incident to numerical levels and organizes actionable **tasks** for analysts which have already been explained in Section 5.5. The **observables** section of the payload aggregates relevant data points, including IP addresses, usernames, file details, and URLs. These observables serve as key indicators for investigation and are attached to the case for streamlined analysis. The case payload and observables are submitted to TheHive via its REST API. Authentication is handled using an API key, and the payload is sent as a JSON object. If the case creation is successful, the script iteratively adds each observable to the newly created case using the `TheHiveApi` Python library within `TheHive4Py`.

In addition to the improvements introduced by the script presented in the previous section, some analyzers have been used to perform automatic enrichment on the observables of a case. Analyzers are extremely useful in this sense because they

Description

A case has been generated based on the following Elasticsearch log data:

Timestamp: The event occurred at 2024-11-13T18:22:43.128Z. This is the exact time when the log entry was recorded, providing a precise timeline for the incident.

Rule Triggered: The case was triggered by the rule named 'MFA'. This rule is designed to identify specific patterns or anomalies in the log data.

User Involved: The user involved in this event is 'happy.panda'.

Source IP Address: The source IP address is 88.37.45.67.

Destination IP Address: The destination IP address is 102.123.45.67.

Device ID: The device ID involved in this event is 29175.

Kibana Alert Link: [http://localhost:5601/s/big-company/app/kibana#/discover?_g=\(time:\(from:'2024-11-13T18:12:43.128000Z',to:'2024-11-13T18:32:43.128000Z'\)\)&_a=\(columns:!\(source\),index:'88e931f0-50ed-11ef-8255-4fe67c27d121',query:\(match_all:{}\)\)](http://localhost:5601/s/big-company/app/kibana#/discover?_g=(time:(from:'2024-11-13T18:12:43.128000Z',to:'2024-11-13T18:32:43.128000Z'))&_a=(columns:!(source),index:'88e931f0-50ed-11ef-8255-4fe67c27d121',query:(match_all:{})))

Figure 6.1: Description and Kibana link in TheHive automatic case generation

2 Options	<p>AbuseIPDB</p> <ul style="list-style-type: none"> ✓ key: API key for AbuseIPDB ✓ days: Check for IP Reports in the last X days (<i>Default: 30</i>)
2 Options	<p>NetBoxIPAnalyzer</p> <ul style="list-style-type: none"> ✓ url: URL of the NetBox instance ✓ token: API token for the NetBox instance
1 Option	<p>Urlscan.io</p> <ul style="list-style-type: none"> ✓ key: API key for Urlscan.io
6 Options	<p>VirusTotal</p> <ul style="list-style-type: none"> ✓ key: API key for Virustotal ✓ polling_interval: Define time interval between two requests attempts for the report (<i>Default: 60</i>) ✓ highlighted_antivirus: Add taxonomy if selected AV don't recognize observable ✓ rescan_hash_older_than_days: Rescan hash observable if report is older than selected days (<i>Default: 30</i>) ✓ download_sample: Download automatically sample as observable when looking for hash ✓ download_sample_if_highlighted: Download automatically sample as observable if highlighted antivirus didn't recognize

Figure 6.2: Cortex Analyzers configuration

are specific to the type of input, whether IP, name, user, domain and so on, and can exploit OSINT tools that are typically queried via REST API through authenticated requests using API keys.

Analyzers are nothing more than Python scripts executed through explicit operations in TheHive, only if enabled and configured in Cortex for the associated organization. They are described by a through a JSON file that also includes the configuration parameters that are then set in Cortex. On the official repository [8], some analyzers are already ready to be installed, configured and enabled. Some refer to OSINT resources, so in the context of the simulation, the analyzers for AbuseIPDB, Urlscan.io, and VirusTotal have been slightly modified and used to enrich observables. Instead, for NetBox there is no official analyzer, and it has been created following the same logic as the previous ones, using a Python script, so that it interacts with the platform starting from TheHive.



Figure 6.3: Email Notifications to analyst

6.2 Optimized Handling Process

Starting from the management process described in Section 5.5, performances are significantly improved as cases are no longer opened manually but are opened in TheHive thanks to the `elastic2hive.py` script that, as previously described, periodically extracts information from alerts triggered in the SIEM and reports them to a case in TheHive. Furthermore, the observables and tasks described above are also added inside the case ready for execution and analysis. The analyst is notified of this operation through emails; this functionality is tested and reported using Mailtrap, an email delivery platform. When an analyst identifies that a new case has been opened in TheHive, they can immediately take responsibility for it by executing the **Takeover** task, even if the analysis does not start immediately. Once the analysis begins, the analyst executes the **Start Analysis** task. During this phase, the analyst, following a methodology similar to the one outlined in the basic process, conducts the necessary analyses. They enhance the automatically generated description by performing log analysis operations in Kibana, leveraging direct connections to the relevant time frame. Additionally, the analyst utilizes configured analyzers on the observables, depending on their type, to perform automated OSINT enrichment and gather supplementary information.

Based on the analyst's decisions, the **Containment** task may be executed to signal

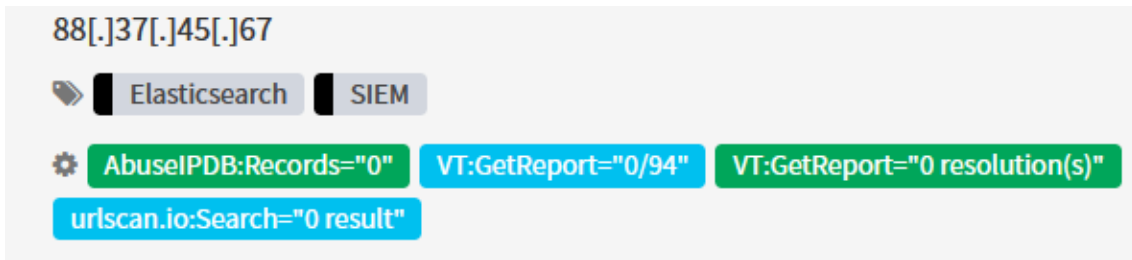


Figure 6.4: Analyzers Test Execution Results on a public IP

the implementation of containment measures. Similarly, the **Remediation** task can be executed if the case transitions into the remediation phase. The execution of these tasks has the purpose of outlining key milestones, marking progress in the analysis and resolution of the case. This structured approach aids not only the analyst but also managers and clients by providing a detailed timeline of operations that have been taken. The timeline documents all actions that culminate in the case's conclusion, which may result in a determination of either a false positive or a true positive. In instances of true positives, it is essential that the analysis is both precise and complete to facilitate a seamless escalation process to L2 or CSIRT.

Analyzing in detail the OSINT enrichment operations with analyzers, these occur by accessing the "Observables" section once the case under analysis has been accessed. Subsequently, by accessing the single observable, a list of available analyzers is shown based on the type of observable. The analyst can then execute analyzers, even simultaneously, on the observable and access the reports in an integrated way in the platform without the need for external resources. The results of a test execution are shown in Figure 6.4 in which reports that the IP is not involved in suspicious or malicious operations on public sources.

6.3 Performance Metrics and Key Indicators for SOC Evaluation

SOC performance can be evaluated through a variety of factors, including Service Level Agreement (SLA) metrics. SLAs, established in collaboration with the customer, provide a time-based framework that defines the type of management required for incidents, specifying whether incidents are to be handled by L1, escalated to L2, or addressed by CSIRT. Each alert triggered must comply with the agreed

SLAs; therefore, the various phases of analysis must be conducted within the agreed times and methods.

In evaluating SOC performance, various metrics and performance indicators are employed. This simulation focuses specifically on operational metrics that evaluate the general activities of an analyst rather than on performance metrics directly related to analysis results. Key metrics include:

- **Mean Time To Acknowledge (MTTA):** The average time between case creation and its acknowledgment by an analyst (also called the **Takeover** operation). MTTA is generally higher before optimizations, as cases are manually created, taking longer to prepare for takeover compared to automated case creation, where observables and tasks are pre-configured.
- **Mean Time To Investigate (MTTI):** The average time from case creation to the start of analysis. This period may align closely with or be immediately after the takeover, assuming the analyst initiates the analysis soon after taking responsibility for the case. However, analysts may sometimes take over cases to meet client SLA requirements but begin the analysis later, for example, due to concurrent analysis tasks. MTTI tends to be higher pre-optimization, as manual task initiation takes longer than an automated process. In some cases, this metric may remain unchanged even after optimization.
- **Mean Time To Respond (MTTR):** The average time from case creation to the implementation of a response to the alert or incident. In the simulation, this task focuses on **Remediation**, involving recommendations for corrective actions to the client. Before optimizations, this metric is estimated to be higher, as response times improve with faster analysis made possible through optimization, thus reducing the time interval.
- **Mean Time To Contain (MTTC):** The average time from case creation to the implementation of containment actions, such as IP blocking or isolating compromised devices. Due to CMDB and EDR tool limitations in the simulation, containment actions are not performed. MTTC, similar to MTTR, is typically higher before optimizations, as containment decisions generally follow or coincide with analysis operations.

- **Average Case Closure Time:** The average time from case creation to its final closure. This value is higher pre-optimization, as all operations are performed manually.
- **Number of Cases Closed Daily:** This metric is generally lower pre-optimization, as the number of cases closed within a specific timeframe is fewer due to longer analysis and closure times.
- **Total Open and Closed Cases:** This KPI indicates the total cases recorded in TheHive, including both open and closed cases.

In order to compare the various measurements performed with a specific Python script `metric_measurements.py` within the simulation, it is necessary to contextualize the various operations by making assumptions. The comparisons shown in Table 6.1 consider basic operations performed during the flow of a simulation that includes the analysis of 5 cases per measurement. Three measurements per scenario were performed to make the overall comparison more reliable. In this scenario, the analyst, pre-optimizations, manually opens a case in TheHive manually including the 4 tasks presented previously and including basic information in the description. In any case, the case is taken in charge as soon as possible to reduce MTTA as much as possible. The analysis operations included in the MTTI are, by hypothesis, very simple and include the ability to search for information on logs in Kibana and perform OSINT enrichment using external resources, and to manually insert observables within each case, extending the analysis times. Post-optimization, however, this operation is automatic thanks to the `elastic2hive.py` script, greatly reducing setup times for the actual analysis; enrichment instead occurs automatically and in an integrated way using analyzers. MTTC instead is virtual in both scenarios since the possibility of managing containment operations using, for example, EDR solutions is not foreseen in this type of study; the reported time refers to the execution of the Containment task in the analysis timeline. MTTR, instead, refers to remediation operations that in normal scenarios consider operations that can require actions by customers and therefore consider indefinite times; within the simulation, it is assumed that a potential customer responds immediately.

	Base			Optimized			Avg Base	Avg Optimized
MTTA	4.19	4.04	3.39	2.31	0.37	0.47	4.27	1.05
MTTI	10.53	9.17	10.35	2.59	2.03	1.46	10.02	2.03
MTTC	14.08	10.31	12.14	4.08	2.27	3.03	12.18	3.39
MTTR	15.32	11.00	13.09	4.17	2.39	2.25	13.14	3.34
Closure	16.03	12.13	13.53	5.12	4.16	3.03	14.30	4.10

Table 6.1: Comparison of Metrics Before and After Optimization (in minutes)

Chapter 7

Conclusions and Future Works

The study conducted on the implementation and optimization of a SOC aimed to document and map the operational context within which it operates and the various processes that lead to the management of the various types of alerts and security incidents through log analysis and enrichment, also relying on public sources. Furthermore, the feasibility study of the implementation of a SOC through its fundamental tools, SIEM and SOAR, was also essential to demonstrate that open-source tools can be considered to achieve this goal by introducing significant advantages. In addition to the possibility of implementing a SOC, optimizations associated with automation operations were also analyzed and implemented.

In this study, the key results are, as previously anticipated, of different types. As a first result, it is necessary to intrinsically report a guide to the SOC, its definition, architecture and operational processes. Show how a SOC can manage and monitor certain assets, users and processes within a customer's infrastructure and how it is possible to respond to possible security incidents. This was possible in collaboration with QiNet, analyzing a functioning operational SOC. The success of this type of operation depended on the precision with which all the phases of managing an alert, and subsequently of a confirmed security incident, were described, starting from the description of the sources considered for the generation of an alert up to the closure of an investigation case considering all the inputs, outputs, tools and participants for each phase.

Secondly, after having studied the architecture and the various processes of a SOC, it was possible to implement a SOC through open-source tools using Elastic as

SIEM and TheHive as SOAR. Elastic, as SIEM, has proven to be extremely versatile starting from the possibility of customizing the configurations of all the tools included in its stack to the possibility of seamless integration with TheHive, making subsequent optimizations possible; once the initial configuration phase has been overcome, which can be complex, Elastic is a fundamental tool for searching and analyzing logs even in real time. TheHive, as SOAR, has been a tool suitable for managing incident response and has a focus on investigation cases. Its most effective feature is that of allowing integration with various types of tools such as Cortex, which allows automation in analysis operations. This tool is also quite complex to configure and heavily depends on external resources to be able to offer complete functionality, such as Elastic and Cortex.

As a final aspect to consider, the optimizations, in the described contexts and following the hypotheses presented, have significantly reduced the management times of the various cases, also indicating an overall acceptable improvement in performance. Results of this type, the result of rather basic simulation operations, lay the foundations for considering improvements in performance even in more complex scenarios where it could be more useful to manage a critical safety case in the most precise and fast way possible.

The study conducted considers some limitations. First, the examples conducted consider simple types of analysis, mainly of L1 type; L2 type analysis or the impact of other types of SOC activities such as vulnerability assessment and penetration testing can be addressed and explored in additional studies. Furthermore, within the SOC implementation simulation, external EDR platforms were not included as they are outside the scope of the study conducted, as these types of platforms are mainly proprietary software. Subsequently, even the measurements conducted are associated, as previously described, with simple operations and analysis cases with the sole purpose of showing the potential and simplicity of automation optimizations.

Some details to be explored in any future work could concern the development of an integrated platform that includes both SIEM and SOAR tools. This would introduce further improvements for the analyst who can perform analysis, monitoring

and incident response operations in a fast, simplified and accurate way.

In addition, the increasing integration of AI tools into professional environments offers significant opportunities for enhancing SOC tools and operations. However, it is essential to emphasize the responsible use of AI to protect sensitive information and ensure data security. Key guidelines for responsible AI usage should be established, validated, and continuously monitored. These guidelines include implementing robust data anonymization practices, such as avoiding the inclusion of identifiable information in AI tools. Additionally, using AI for projects involving sensitive documents, reports, or code containing configuration details should be avoided. To mitigate potential data leaks, organizations must recognize that AI providers often use input data for training purposes, potentially exposing it to third parties; therefore, such practices should be also strictly avoided. Verification and validation of AI-generated outputs are equally critical, as these tools are prone to errors or inaccuracies. By adhering to these practices, SOCs can maintain data integrity and operational security while leveraging AI to enhance efficiency and effectiveness.

In conclusion, this thesis has provided a comprehensive analysis of the strategies, workflows, and tools essential for implementing and optimizing SOC operations. By integrating effective methodologies, leveraging open-source technologies, and proposing enhancements to existing workflows, the study contributes to advancing the capabilities of modern SOC environments. The research highlights the importance of continuous improvement, particularly through automation, to address the dynamic challenges of cybersecurity.

The findings underscore the necessity for SOCs to balance efficiency and security, ensuring the integrity of sensitive information while responding to evolving threats. Moving forward, the adoption of emerging technologies and adherence to best practices will be critical in enhancing resilience against future cyber threats. This thesis aspires to serve as a foundation for further exploration and innovation in the domain of SOC, contributing to stronger cybersecurity ecosystems across industries.

Bibliography

- [1] Github repository. <https://github.com/ferraroby00/MySOC>.
- [2] Elastic. Elastic github repository. <https://github.com/elastic>.
- [3] Free Software Foundation. Free software foundation. <https://www.fsf.org/>.
- [4] OSINT Framework. Osint framework. <https://osintframework.com/>.
- [5] Open-Source Initiative. The steward of the open source definition, setting the foundation for the open source software ecosystem. <https://opensource.org/>.
- [6] Netbox Labs. Netbox oss documentation. <https://netboxlabs.com/docs/netbox/en/stable/>.
- [7] TheHive Project. Cortex. <https://github.com/TheHive-Project/Cortex>.
- [8] TheHive Project. Cortex analyzers. <https://github.com/TheHive-Project/Cortex-Analyzers>.
- [9] TheHive Project. Thehive project 4. <https://github.com/TheHive-Project/TheHive>.
- [10] TheHive Project. Thehive4py. <https://github.com/TheHive-Project/TheHive4py>.
- [11] Edgars Taurins. How to setup soc and csirt - good practice guide. Technical report, ENISA, 2020.
- [12] Manfred Vielberth, Gunther Pernul, and Fabian Bohm. Security operations center: A systematic study and open challenges. 2020.