

IPL

escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

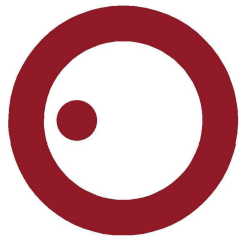
RESPOSTA A INCIDENTES DE CIBERSEGURANÇA
NO SECURITY OPERATION CENTER DO
POLITÉCNICO DE LEIRIA

*HANDLING CYBERSECURITY RELATED INCIDENTS IN
THE SECURITY OPERATION CENTER OF THE
POLYTECHNIC OF LEIRIA*

STUDENT MARCO ALEXANDRE CLEMENTE MATEUS

Leiria, November of 2021

[November 4, 2021 at 22:39 –]



IPL

escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

**RESPOSTA A INCIDENTES DE CIBERSEGURANÇA
NO SECURITY OPERATION CENTER DO
POLITÉCNICO DE LEIRIA**

*HANDLING CYBERSECURITY RELATED INCIDENTS IN
THE SECURITY OPERATION CENTER OF THE
POLYTECHNIC OF LEIRIA*

STUDENT MARCO ALEXANDRE CLEMENTE MATEUS

Number: 2190376

Project carried out under the guidance Professor Doutor Carlos Manuel da Silva Rabadão and Professor Adaíl Domingues da Silva de Oliveira

Leiria, November of 2021

[November 4, 2021 at 22:39 –]

ACKNOWLEDGMENTS

I would like to thank professor Adaíl Oliveira and professor Carlos Rabadão for all the guidance they provided while carrying out this work.

More specifically, I would like to thank professor Adaíl Oliveira for his help and guidance in matters related to technology decisions and in matters related to the security operations center of the Instituto Politécnico de Leiria, and professor Carlos Rabadão for his support on the general conception of the project.

ABSTRACT

In the present day, **IT** systems are an integral part of most organizations, and play a huge role in their success. With the necessity to connect these systems to the internet to further amplify their benefits and possibilities, comes the issue of cybersecurity. Allied to the importance of these systems for the organizations, comes the interest of attackers in disrupting these same services. When the amount of cyberattacks occurring everyday is taken into consideration, and how these might impact organizations, this issue becomes one of the greatest challenges they have to deal with.

The problems that this project deals with are fundamentally connected with this issue. With the variety of attacks that currently circulate, **Security Operations Center (SOC)** rely on many different software to monitor their systems, which in turn create too much information to be handled individually by security analysts.

In this project this issue was analyzed, as well how it can be handled, as the main objective of this project is to find a solution for the **SOC** of the **Instituto Politécnico de Leiria (IPLeiria)** which is facing this very same issue. The proposed solution to this problem is through **Security Orchestration, Automation and Response (SOAR)**. **SOAR** encompasses different concepts that help in creating effective and efficient routines to handle the incidents that a **SOC** faces on a daily basis.

To tackle this problem in the case of the **IPLeiria SOC**, the solution found relied on the use of a **SOAR** platform or software. For this effect different solutions available were analysed, including free and paid software. The choice came down to using a free software called Shuffle ¹ in conjunction with the already existent in the **IPLeiria SOC** case management platform TheHive ².

With these two tools, different playbooks were developed to handle the most prominent type of incidents the **SOC** faces.

¹ Shuffle is an opensource automation platform, focused in cyber security operations, <https://shuffler.io/>

² TheHive is an open source and free Security Incident Response Platform, <http://thehive-project.org/>

INDEX

Acknowledgments	i
Abstract	iii
Index	v
List of Figures	vii
List of Tables	ix
Acronyms	xi
1 INTRODUCTION	1
1.1 Motivation and Objectives	2
1.2 Investigation Methodology	3
1.3 Contributions	4
1.4 Document Structure	4
2 BACKGROUND	5
2.1 Security Operations Center (SOC)	6
2.1.1 Incident Response	8
2.1.2 Threat Intelligence	11
2.2 Security Automation, Orchestration and response	12
2.2.1 SOAR vs SIEM	13
2.2.2 Playbooks	15
2.2.3 Playbook vs Runbook	18
2.2.4 SOAR Architecture	19
2.2.5 SOAR Characteristics	20
2.3 SOAR Software Solutions	23
2.3.1 Opensource solutions	23
2.3.2 Commercial software solutions	29
2.4 Summary	34
3 PROOF OF CONCEPT DEVELOPMENT AND TESTING	37
3.1 Environment and constrains	37
3.1.1 Incident Response Procedures	37
3.1.2 Infrastructure	41

3.2	Choosing the SOAR Software	43
3.3	Proof of concept implementation environment	47
3.4	Use Cases	48
3.4.1	Analysing Email With Phishing Suspicion	48
3.4.2	Handling Data Breaches	60
3.4.3	Summary	69
3.5	Testing	69
3.5.1	Phishing Runbook	69
3.5.2	Data Breach Runbook	73
4	CONCLUSION	77
4.1	Conclusions	77
4.2	Future Work	78
	BIBLIOGRAPHY	79
	DECLARAÇÃO	83

LIST OF FIGURES

Figure 1	Security Incidents Frequency (Splunk, 2017)	5
Figure 2	SOC Sources of Information (Efij, 2018)	7
Figure 3	Data collection process (M. Vielberth, 2020)	7
Figure 4	NIST Framework (Standards and Technology, 2018)	9
Figure 5	NIST Framework Steps (Forsyth, 2018)	10
Figure 6	SANS Framework Steps (IR, 2020)	10
Figure 7	Threat Intelligence Platform functionality (TechEN, 2021)	12
Figure 8	SOAR Role	14
Figure 9	SIEM and SOAR Interaction (Wierzbicki, 2021)	15
Figure 10	OASIS playbook structure (Open, 2021)	17
Figure 11	Runbook example (Foolcdn, 2020)	19
Figure 12	SOAR Deployment	20
Figure 13	Security Orchestration (PaloAlto, 2019)	21
Figure 14	Walkoff Workflow builder (<i>WALKOFF Documentation</i> n.d.)	24
Figure 15	Walkoff Applications Menu (<i>WALKOFF Documentation</i> n.d.)	25
Figure 16	Shuffle Workflow (Oedegaardstuen, 2020)	26
Figure 17	Shuffle New Application Creation (Oedegaardstuen, 2020)	27
Figure 18	The Hive Dashboard (<i>TheHive Github</i> n.d.)	28
Figure 19	The Hive Workflow (<i>TheHive Github</i> n.d.)	29
Figure 20	The Hive Current Cases (<i>TheHive Github</i> n.d.)	30
Figure 21	Siemplify Runbook	31
Figure 22	Siemplify Prebuilt Runbook	31
Figure 23	Siemplify Statistics Dashboard	32
Figure 24	Splunk Phantom Runbook	33
Figure 25	Splunk Phantom Dashboard	33
Figure 26	Playbook to block IP and reset user credentials from ingested data	35
Figure 27	SOAR solutions analysed	35
Figure 28	General Procedure of CSIRT	38
Figure 29	Incident Response Lifecycle 1	39
Figure 30	Incident Response Lifecycle 2	40
Figure 31	IPLeiria SOC Architecture	42

Figure 32	IPLeiria SOC Analyst Interactions	43
Figure 33	Implementation Environment	48
Figure 34	Phishing Playbook	51
Figure 35	Email message shown through mailbox application in Shuffle	52
Figure 36	Create and alert in TheHive	53
Figure 37	Phishing alert created on TheHive by Shuffle workflow . . .	53
Figure 38	"Verdicts" included in the urlscan response	56
Figure 39	Virustotal response (VirusTotal, 2021a)	57
Figure 40	UrlScan application configuration	57
Figure 41	Virustotal application configuration	58
Figure 42	IBM XForce application configuration	58
Figure 43	Shuffle authentication section	59
Figure 44	Virustotal workflow condition	59
Figure 45	Active Directory Integration in Shuffle	60
Figure 46	Databreach runbook	63
Figure 47	Immuniweb dashboard	63
Figure 48	Schedule configuration	64
Figure 49	Immuniweb Api Parameters	64
Figure 50	Application builder Shuffle, Immuniweb basic configuration .	65
Figure 51	Application builder Shuffle, Immuniweb action configuration	66
Figure 52	Immuniweb Configuration	66
Figure 53	Shuffle Immuniweb Incidents Control	67
Figure 54	Phishing Email Sample	70
Figure 55	Execution of the Phishing Workflow	71
Figure 56	Creation of Alert on TheHive	71
Figure 57	Command Line Results from Virustotal and URLScan . . .	72
Figure 58	Command Line Results from IBMXForce	72
Figure 59	Updated The Hive Alert	73
Figure 60	Active Directory User Management	74
Figure 61	Data breach Runbook Execution Without New Incidents . .	74
Figure 62	Data breach Runbook Execution With New Incidents	75

LIST OF TABLES

Table 1	SOAR tools characteristics matrix	46
---------	---	----

ACRONYMS

API	Application Programming Interface.
CSIRT	Computer Security Incident Response Team.
DNS	Domain Name System.
ELK	Elastic Search Logstash, Kibana.
IDS	Intrusion Detection System.
IOT	Internet Of Things.
IPLeiria	Instituto Politécnico de Leiria.
IRP	Incident Response Plan.
IT	Information Technology.
MISP	Malware Information Sharing Platform.
NIST	National Institute of Standards and Technology.
OASIS	Organization for the Advancement of Structured Information Standards.
REST	Representational state transfer.
SAAS	Software as a Service.
SANS	SysAdmin, Audit, Network, and Security.
SIEM	Security Information and Event Management.
SIRP	Security Incident Response Platforms.
SOA	Security Orchestration and Automation.
SOAR	Security Orchestration, Automation and Response.
SOC	Security Operations Center.
TIP	Threat Intelligence Platforms.
URL	Uniform Resource Locator.
UUID	Universally unique identifier.
VM	Virtual Machine.
VPN	Virtual Private Network.

INTRODUCTION

This document aims to detail the process of implementing a [Security Orchestration, Automation and Response \(SOAR\)](#) system in the [Security Operations Center \(SOC\)](#) of the [Instituto Politécnico de Leiria \(IPLeiria\)](#).

With the increasing use of computer information systems to support every type of process within many different kinds of organizations, these systems not only carry very important information, which should not be access by a non authorized people, but they also support many of these processes, being crucial to the correct operation of the organization.

As a consequence to this situation, it is becoming more and more important to protect these systems. Since many of the organizations processes may rely on these types of systems, in case of failure, the organization might incur a big financial losses. Having this in consideration, it is of utmost importance to ensure that they are available and operating as expected, so that none of the organization operations are impaired.

Alongside the increasing importance of these systems for the companies, the interest in exploiting them also increases. There has never existed so many cyber-attacks as there are today and companies have never invested so much money in cybersecurity as they invest now (Morgan, 2019).

Cybersecurity breaches can lead to serious organizational and socio-economic consequences. This consequences can include loss of revenue, damage to reputation and information system, and theft of proprietary data and customer sensitive information. For example, Equifax (one of the largest credit reporting agencies in America) reported a major data breach that had affected around 148 million US consumers. The hackers were able to successfully steal sensitive information like credit card numbers, phone numbers, email addresses, and social security numbers. According to a research sponsored by IBM, the average total cost of a breach is around \$3.62 million per incident (Chandni Islam, 2019).

In order to keep these systems secure and under control, many different kinds of software applications have been developed and are available on the market, such as

firewalls, antivirus, threat intelligence, user and entity behavior analytics, among others. Nowadays it is not very hard for an organization to require a relatively large number of these application to keep track of what is happening in it's systems.

1.1 MOTIVATION AND OBJECTIVES

By using all this different software applications to ensure the security of their systems, organizations will find themselves in a situation where there is more information being generated, then they have the capability to properly handle. To be able to able to properly handle all of this information these organizations would need to employ more human resources in the area of cybersecurity. Although this would be one solution to the problem of the higher quantity of information being generated, it is a solution that isn't viable taking in consideration the general current state of cybersecurity operations in organizations. In most cases the divisions in charge of cybersecurity have small IT teams and inadequate security budgets (Michael Benza, 2020).

Another problem that arises from this new security needs, is the difficulty of integrating inherently different software applications. The use of software that comes from different security solutions providers, which use different technologies and paradigms to develop, deploy, and operate their security solutions, makes the task of integrating them to work together and inter-operate for effective and efficient support, much harder to accomplish.

Security orchestration is aimed at introducing technical and socio-technical solutions to integrate multi vendor security tools as a unified whole to support security staff in a SOC. Organizations are increasingly adopting security orchestration platforms that are proactive, autonomous, and collaborative solutions to enable security staff perform their responsibilities effectively and efficiently. A security orchestration initiative enables people, practices, and technologies to work together to improve organizations security intelligence for better security operations and management. Security orchestration is a prerequisite of security automation, which is the process of automatically detecting, preventing, and recovering from cyber-attacks without human interference using information technology, automation algorithm and artificial intelligence (Chandni Islam, 2019).

Furthermore, as a result of the previously discussed rising of cyberthreats, security automation has gained much relevance and become a major issue for many companies in this fight for cybersecurity. A recent survey by the threat detection and hunting

company Fidelis Cybersecurity has revealed this trend among 300 CISOs, CIOs, CTOs, architects, engineers, and analysts studied in a range of industries. More than half of the professionals analyzed (57 percent) said that their companies are concerned with a lack of automation. Cybersecurity automation is one of the developments in information technology. Automating human-driven, and repeatable processes will free resources to focus on the more productive problem solving tasks within organizations and individuals. Focusing on these issues will foster innovation and contribute to a more robust organization from a cybersecurity point of view (Sikender Mohsienuddin Mohammad, 2018).

This same problems can be found in the SOC of the IPLeiria, where the numbers of daily incidents have risen and the resources to properly handle them are not enough.

The aim of this project is to find and implement a solution that helps with this very problem. The solution should be able to handle some of the information that is generated from security applications and events, eliminating the need of a human resource or reducing the amount of tasks a human analyst needs to perform to handle it. This should be achieved by automating streamlined workflows, that are prepared to take some sort of event, and based on it, performed appropriate actions to handle it.

1.2 INVESTIGATION METHODOLOGY

The project development went through distinct phases, that focused on solving specific problems and ultimately lead to a prototype solution.

The first step was to gather information on how to implement security automation in an Information Technology (IT) environment, in order to have an understanding of what are good practices and methodologies to follow.

Secondly, it was necessary to conduct some research on what automation solutions were available in the market, so that later a decision could be made for the one which better satisfies the necessary constraints.

Thirdly, a prototype was developed to test the chosen solution in different scenarios. Here playbooks were developed for each scenario, and implemented with the chosen automation solution.

The last step consisted in creating test scenarios and running the developed prototype to analyse its performance and to validate its output.

1.3 CONTRIBUTIONS

This project gathered and analysed information regarding cybersecurity, with a focus on the area of automation. From this research it was possible to understand the actual state of this area, and how the current issues are handled.

On a more specific level, different software solutions to this problem were analysed and compared, having been reached conclusions on the strong points and drawbacks of each one of the tools explored.

Furthermore, the chosen software was used to implement the project requirements and it was detailed how this solution can be used to achieve such results.

Finally, a prototype was built that can be deployed to work on the target environment or be further developed to integrate new functionalities.

1.4 DOCUMENT STRUCTURE

The document is structured in 4 chapters that detail the different steps taken when developing this project.

Firstly is the present chapter, the introduction, which has the objective of explaining the reader what this report concerns and why it exists.

In the second chapter, it is performed a background review about the technologies and approaches that are being used currently, as well as the concepts that are intrinsically related to the subject matter.

The third chapter details the process of deciding how the problem this project deals with will be solved and the actions involved in implementing the actual solution.

In the fourth chapter, the work that was carried out in this project is summarized, it is made an assessment on how the development of the project went, before finally leaving some topics that may be further developed in the future.

BACKGROUND

The increase in security related information that is being generated nowadays becomes a problem because to make good use of all this information a team with many security analysts is needed, which will increase the cost of keeping a security team for the company. Taking into consideration that at the moment most security divisions of organizations are already undermanned and under-budgeted, an alternative solution is required to handle organization Cybersecurity. One way to manage this situation is to employ some sort of security orchestration automation in the **SOC** of organizations.

A study at the University of Maryland (Cukier, 2017) as found that cyberattacks were happening as often as every 39 seconds (figure 1). This situation together with the fact that security staff today have to monitor also a much larger infrastructure than in the past which represents a much larger attack surface, including mobile devices, cloud infrastructure and IoT devices, means that there are way more alerts being generated than even a well organized **SOC** can handle. Many of these tasks are mundane, repetitive, tedious tasks that contribute for the dissatisfaction of **SOC** employees with their job, a factor that is also negative for the organization as it makes the task of having a team with experienced members, not only on the skills necessary to perform their job, but also that possess a good understanding of the organization systems (M. Vielberth, 2020).



Figure 1: Security Incidents Frequency (Splunk, 2017)

With security automation, repetitive, time-consuming actions can be taken care automatically by machines, leaving security analysts with more time to focus on more important, value-adding work. In addition, security automation can also provide rapid threat detection, with response times that can't be matched by a human analyst scanning through security logs information. According to research by the Enterprise Strategy Group (ESG, 2016), IT teams ignore 74 percent of security events/alerts, even when they have security solutions in place, simply due to the sheer volume of this information. Not only can security automation detect and resolve these common issues, it also eliminates human error, which is more prone to happen in repetitive actions than others, that comes with inexperience, work overload and negligence (Splunk, 2017).

To better understand the purpose and aim of a SOAR platform, first it is necessary to understand the work that the security personnel has to carry out.

2.1 SECURITY OPERATIONS CENTER (SOC)

A Security Operations Center (SOC) is a department that has a team of information security specialists, responsible for monitoring and analyzing an organization's security posture on an ongoing basis, and where are security related systems are centralized (figure 2). A SOC can provide a solution for detecting and mitigating an attack if implemented correctly. They incorporate a mixture of people, processes, technologies, and governance and compliance, to effectively identify, detect, and mitigate threats, ideally before any damage occurs. Many organizations have created SOC, generally in the form of a group of security specialists who monitor, prevent, report, and respond to security attacks (Groot, 2020).

To have an effective and efficient SOC, it is necessary to combine multiple factors together. In the research published on "Security Operations Center: A Systematic Study and Open Challenges" (M. Vielberth, 2020), the authors identified four main building blocks, that are involved in a SOC, and which correct management is fundamental to the effectiveness of the SOC.

The first point is the people involved in the SOC. The different members of a SOC should have clearly defined roles and responsibilities, like threat hunter, incident responder, etc. It is a job that requires qualified personnel and being able to retain employees for longer is important for a strong team, opposed to having an high rotation of staff, entering and leaving the organization SOC. The members of a SOC must also have training on different areas, that enables them to work with all



Figure 2: SOC Sources of Information (Efij, 2018)

the different tools used on their job, and handle different types of incidents, as well as having a work space where collaboration and communication is made easy.

Another important factor are the processes that are implemented inside the SOC. Their should be processes in place to handle the different steps in incident handling. In preparation before an incident, where data is collected and processed in different steps, to later be of use to other processes within the SOC (figure 3), in detection and analysis where data is analysed to try to detect any intrusion, and in containment and eradication, where the threat is neutralized and the systems are brought back to normal operation.

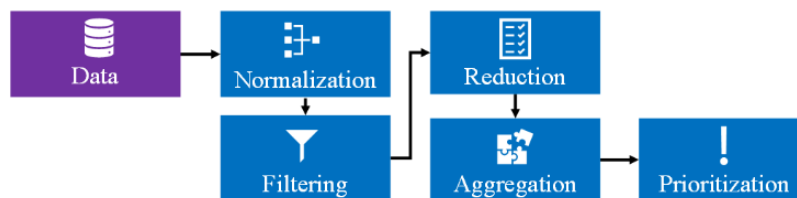


Figure 3: Data collection process (M. Vielberth, 2020)

The technology used in the SOC is also a key component, as it supports many of its processes. There are different ways of dealing with data collection, its analysis and the way it is presented. The technologies and the way they are utilized on the SOC should be well thought out and clear to all of its members.

Lastly, the governance and compliance of the SOC, is a matter of great importance. Governance refers to the way the IT systems are used as it ensures that these systems are being used in a effective and efficient way, and provides strategic direction,

developing standards, policies and procedures, and implements them. Compliance guarantees that the organization adheres to external rules, such as standards and regulations and internal rules, like policies and procedures.

A [SOC](#) surely helps organizations to be prepared for cyberattacks. However, they need to be planned thoroughly, implemented, and integrated very carefully, assessed regularly, and improved continually to unveil their full potential. If done correctly, they can improve an organization ability to prevent hacks, financial losses, and personal data breaches.

2.1.1 *Incident Response*

Incident response is a methodology that an organization uses to respond and manage a cyberattack. This methodology consists in a set of policies and procedures that can be used to identify, contain, and eliminate cyberattacks. An attack or data breach can wreak havoc potentially affecting customers, intellectual property company time and resources, and brand value. The goal of incident response is to enable an organization to quickly detect and halt attacks, minimizing damage, while also ensuring that all services are returned to normal operation as quickly as possible (Forcepoint, 2021). Because many companies today experience a breach at some point in time, a well-developed and repeatable [Incident Response Plan \(IRP\)](#) is the best way to protect the organization. An [IRP](#) is a set of documented procedures detailing the steps that should be taken in each phase of incident response. It should include guidelines for roles and responsibilities, communication plans, and standardized response protocols.

Having an incident response plan is crucial to not only to handle the incident while it occurs, but to properly handle the incident afterwards. Poor incident response can alienate customers and trigger greater government regulation, which may bring worse consequences to the organization than the immediate impact of the attack, as the organization may end up having to fines and gets its reputation tainted. In the already mentioned hack of Equifax in 2017, the decision to not share information with the public following the hack significantly hurt its brand. It is of utmost importance to report and disclose identified data breach incidents promptly, in order to comply with regulations and to avoid any legal penalties and negative public perceptions of the organization upon latent discovery of responsibilities (Christopher Johnson, 2018). Different countries have different regulations on how organizations have to communicate this incidents, which means that an incident

response plan is essential to properly handle the incident and not incur any of the problems mentioned.

There are multiple frameworks that offer guidelines on how to develop an [IRP](#) that suits a specific company needs. These frameworks are generally developed by large organizations with a significant amount of security expertise and experience. Two of the best known of these frameworks are those developed by [National Institute of Standards and Technology \(NIST\)](#) and the [SysAdmin, Audit, Network, and Security \(SANS\)](#).

2.1.1.1 *NIST Framework*

In the case of the [NIST](#) framework, it has three main components (figure 4). The framework core provides a set of desired cybersecurity activities and outcomes. It guides organizations in managing and reducing their cybersecurity risks in a way that complements an organization's existing cybersecurity and risk management processes. The Tiers guide organizations to consider the appropriate level of rigor for their cybersecurity program and are often used as a communication tool to discuss risk appetite, mission priority, and budget. Framework Profiles are an organization's unique alignment of their organizational requirements and objectives, risk appetite, and resources against the desired outcomes of the framework Core (Standards and Technology, 2018).

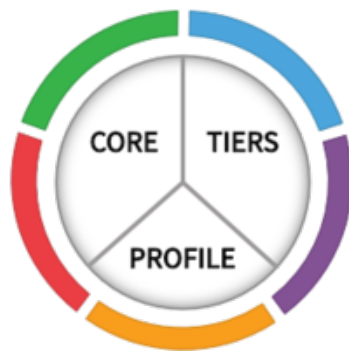


Figure 4: NIST Framework (Standards and Technology, 2018)

The Core component, there are five high level functions specified: Identify, Protect, Detect, Respond, and Recover (Standards and Technology, 2018). These functions act as the backbone of the framework core that the rest of the elements are organized around. These five functions were chosen on the basis that they represent the five primary pillars for a successful and holistic cybersecurity program, according no [NIST](#).

The identify function assists in developing an organizational understanding to managing cybersecurity risk to systems, people, assets, data, and capabilities. The protect function outlines appropriate safeguards to ensure delivery of critical infrastructure services. The detect function defines the appropriate activities to identify the occurrence of a cybersecurity event. The respond Function includes appropriate activities to take action regarding a detected cybersecurity incident. And finally, the recover function identifies appropriate activities to maintain plans for resilience and to restore any capabilities or services that were impaired due to a cybersecurity incident (Standards and Technology, 2018).



Figure 5: NIST Framework Steps (Forsyth, 2018)

2.1.1.2 SANS Framework

The SANS framework aims to just like the NIST framework, provide a guide on how a company should set up its resources, so that they can handle cybersecurity incidents efficiently. The SANS institute published a 20-page handbook that lays out a structured 6-step plan for incident response (figure 6).



Figure 6: SANS Framework Steps (IR, 2020)

In the Preparation step organizations should review and codify an organizational security policy, perform a risk assessment, identify sensitive assets, define which are critical security incidents the team should focus on, and build a [Computer Security Incident Response Team \(CSIRT\)](#). The Identification specifies that an organization monitors its IT systems and detects deviations from normal operations, monitoring if they represent actual security incidents. In the Containment step, the threat should be contained by performing short-term containment, for example by isolating the network segment that is under attack, and later adopt a more long term solution. The Eradication step refers to the removal of the threat from the affected systems, identifying the root cause of the attack. The recovery step entails the process of bringing the systems back to normal operations. Finally the Lessons Learned step explain how to perform a retrospective of the incident, creating appropriate documentation of the incident (Kral, 2021).

2.1.2 *Threat Intelligence*

One key aspect of a successful attack is being ahead of the security programs and general threat information that the organization possesses. This permits the attackers to find a new ways of breaking into the organization networks. In order avoid letting this happen it is of utmost importance that the security team has the most recent information on possible cyber threats (Seker, 2020).

Gartner defines threat intelligence as evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to IT or information assets (Gartner, 2021). This is information that organizations collect, either from past incidents they had to deal with or obtained from external sources, and can use to better understand past, present, and future threats. Amongst others, examples of threat intelligence include indicators (system artefacts or observables associated with an attack), security alerts, incident reports (Andrew Ramsdale and Kolokotronis, 2020).

One excellent way for an organization to obtain threat intelligence is to use a [Threat Intelligence Platforms \(TIP\)](#). A TIP is a technology solution that collects, aggregates and organizes threat intel data from multiple sources and formats. With a TIP, analysts save much time by not having to continuously look for the latest information on known threats, since a TIP can aggregate and provide this information in a convenient way. Also, a TIP not only provides information, but

also enables security and threat intelligence teams to easily share threat intelligence data with other stakeholders and security systems (PaloAlto, 2019).



Figure 7: Threat Intelligence Platform functionality (TechEN, 2021)

By making use of a [TIP](#), security teams can stay updated in the latest information on cyber threats much more easily when comparing with the traditional methods of obtaining this information manually. This removes a lot of work from the analyst, much of it repetitive and human error prone, which increases the efficiency of the team. Furthermore, it also enables them to monitor and quickly detect, validate and respond to potential security threats much faster.

Another interesting aspect of the a [TIP](#), is its potential to integrate with another security tools like [SIEM](#), [IDS](#), which can give tools like these another level of functionality, by enriching the actions this tools normally execute with the information they can provide.

2.2 SECURITY AUTOMATION, ORCHESTRATION AND RESPONSE

[Security Orchestration, Automation and Response \(SOAR\)](#) is a relatively recent term that tries do define a new set of security tools that are becoming more popular lately, as a result of the previously described situation of increasing security needs.

The term [SOAR](#) refers to technologies that enable organizations to collect inputs monitored by the security operations team. For example, alerts from the [SIEM](#) system and other security technologies — where incident analysis and triage can be performed by leveraging a combination of human and machine power — help define, prioritize and drive standardized incident response activities. [SOAR](#) tools allow

an organization to define incident analysis and response procedures in a digital workflow format (Gartner, 2020).

The current definition of the term **SOAR** was set in 2017, however, the term has actually been used as early 2015 by Gartner to describe “Security Operations, Analytics, and Reporting”. Later the term was revised to refer to its current definition in 2017 as it saw a convergence of existing technologies such as **Security Orchestration and Automation (SOA)**, **Security Incident Response Platforms (SIRP)**, and **Threat Intelligence Platforms (TIP)** (Scott, 2020).

2.2.1 *SOAR vs SIEM*

As a result from the issue of the large quantities of security information being generated nowadays, software platforms for **Security Information and Event Management (SIEM)** and **SOAR** are becoming more common. A **SIEM** and a **SOAR** have different capabilities that can be used in conjunction, and the correct use of each one strengths is key to a good performance when handling an incident.

SIEM is a technology that helps in the detection of threats and security incidents, which in turn helps in the prompt incident response. By using of near real time event log collection and analysis of various, disparate event data, a **SIEM** makes sense of the immense quantity of information that is generated by all the security applications, by collecting, aggregating, categorizing and analyzing this data.

Detection is only possible if **IT** events are gathered and appropriately analysed, which is becoming an increasingly complex task, due to the many different sources of information from different applications and large number of events, which ultimately makes the job of discovering all types of incoming threats hard. In a scenario where all the information is separated, events that are mutually related on different platforms, will usually stay undetected. A **SIEM** solves this problem by centralizing event logs together, and correlating/analysing this information. **SIEM** offers capabilities like (Mario Zgela, 2019):

- agent or agentless event collection;
- aggregation and normalization of events;
- near real time event monitoring;
- pre-defined engine for threat identification, with possibility of custom rule definition;

- searching and reporting on various threats.

A **SIEM**, may often include features that try to identify patterns in the data it collects, that may indicate a cyberattack, eventually issuing alerts accordingly. These functions, generally rely on machine learning techniques. In the context of a **SIEM**, machine learning takes cybersecurity rules and data to try to find evidences of an attack on the organization systems.

Although this type of platform already gives a great help to security analysts, **SIEM** mostly focus on managing the information that is inside the organization systems. There is still another set of actions needed to be taken on a daily basis by security analysts where **SIEM** software falls short on providing any help, although, this is where **SOAR** software comes in (Froehlich, 2021).

A **SOAR** is able to establish integration's with many different types of security applications and tools which enables it to achieve highly automated and complex incident response workflows. With this automation, it's not only easier and faster to deliver results when an incident happens, but it also reduces the amount of work a security analyst has to carry out (figure 8), especially in terms of simple repetitive actions (Kirtley, 2020).

While some features of a **SOAR** and **SIEM** may overlap, there are some features mainly related to automation that are only generally found on **SOAR**, and while it is typically possible to execute most of a **SIEM** actions on a **SOAR**, it is much more time consuming and inefficient.

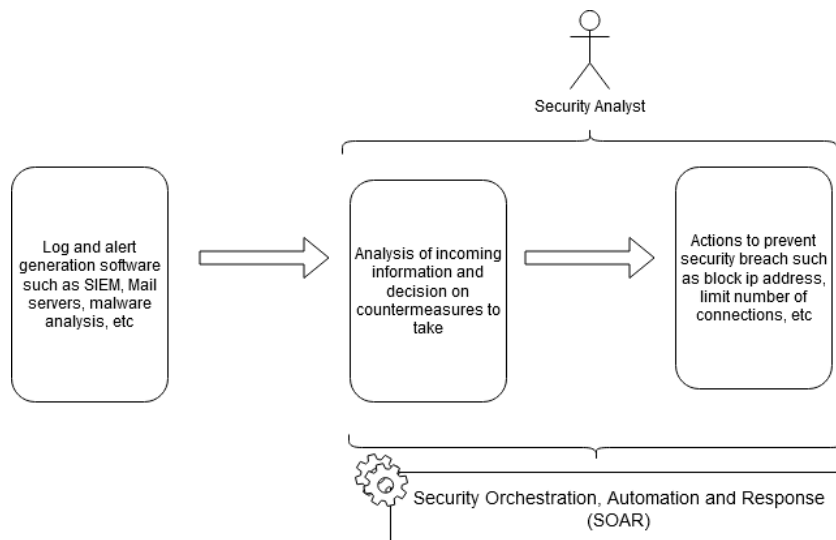


Figure 8: SOAR Role

Both **SIEM** and **SOAR** products generally consume data feeds, though, **SIEM** tools are better suited for larger volumes of data with disparate sources and formats. Both tools are meant to provide automation to detecting and managing security incidents, however, **SOAR** tools offer many more possibilities in the field of automation, even being capable of taking a human analyst out of the process in some incident routines.

Concluding, what this means is that both solutions can be used together by taking advantage of the **SIEM** ability to ingest large volumes of data and generate alerts, with a **SOAR** solution layered on top of the **SIEM** (figure 9), to manage the incident response process to each alert, automating and orchestrating a number of mundane and repetitive tasks, for an optimal configuration (Moran, 2018).

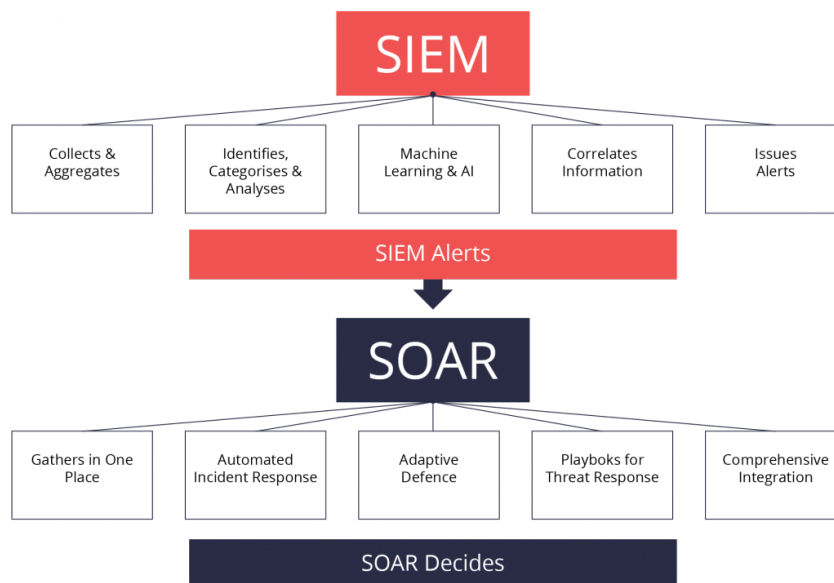


Figure 9: SIEM and SOAR Interaction (Wierzbicki, 2021)

2.2.2 Playbooks

A security playbook, is a flow required steps and actions that provide guidance on how to address a certain security event successfully. Playbooks provide a step-by-step approach to orchestration, helping security teams establishing standardized incident response processes and ensuring the steps are followed in compliance with regulatory frameworks.

While an incident response plan should be more concise and relevant to a wide variety of security incidents, a security playbook acts as a helpful manual for more specific situations, focusing in step by step directions for a well scoped incident task. The combination of these two resources provides the organization with a

incident response plan which is applicable in most incidents, and incident response playbooks that are applicable for specific incidents (McGeehan, 2018). Taking this into consideration, it is very important for an organization to possess different playbooks for the many situations it might find itself in.

Playbooks document an approved process, that can be followed either daily or once a year, by either an experienced employee or a new hire, and the end result of all actions taken should be the same. The actions defined by security playbooks are the basis for the incident handling, and as such, these playbooks will be the elements that this projects aims to automate.

The [Organization for the Advancement of Structured Information Standards \(OASIS\)](#) is a global nonprofit consortium that works on the development, convergence, and adoption of open standards areas such as cybersecurity, blockchain, [Internet Of Things \(IOT\)](#), among others. [OASIS](#) has developed a specification that offers a schema for security playbooks and how these can be created, documented, and shared in a structured and standardized way across organizational boundaries (Open, 2021).

In the [OASIS](#) specification, playbooks are classified into seven different types as follows:

- Notification Playbook - A playbook that is primarily focused on the orchestration steps required to notify and disseminate information and other playbooks about a security event, incident, or other threat. For example, a notification playbook can be used to notify multiple entities about an attack and disseminate other playbooks to detect and mitigate it as quickly as possible;
- Detection Playbook - A playbook that is primarily focused on the orchestration steps required to detect a known security event, other known or expected security-relevant activity, or for threat hunting;
- Investigation Playbook - A playbook that is primarily focused on the orchestration steps required to investigate what a security event, incident, or other security-relevant activity has caused. Investigation playbooks will likely inform other subsequent actions upon completion of the investigation;
- Prevention Playbook - A playbook that is primarily focused on the orchestration steps required to prevent a known or expected security event, incident, or threat from occurring. Prevention playbooks are often designed and deployed as part of best practices to safeguard organizations from known and perceived threats and behaviors associated with suspicious activity;

- Mitigation Playbook - A playbook that is primarily focused on the orchestration steps required to mitigate a security event or incident that has occurred when remediation is not initially possible. Organizations often choose to mitigate a security event or incident until they can actually remediate it, like quarantining affected users/devices/applications from the network temporarily to prevent additional problems;
- Remediation Playbook - A playbook that is primarily focused on the orchestration steps required to remediate, resolve, or fix the resultant state of a security event or incident, and return the system, device, or network back to a nominal operating state.
- Attack Playbook - A playbook that is primarily focused on the orchestration steps required to execute a penetration test or attack simulation to test or verify security controls or identify vulnerabilities within an organization's environment. This is often represented by a penetration test that is used to verify how security systems or other systems respond to various aspects of the test or attack.

CACAO playbooks are structured in five sections; playbook metadata, the workflow logic, a list of targets, a list of extensions, and a list of data markings (figure 10).

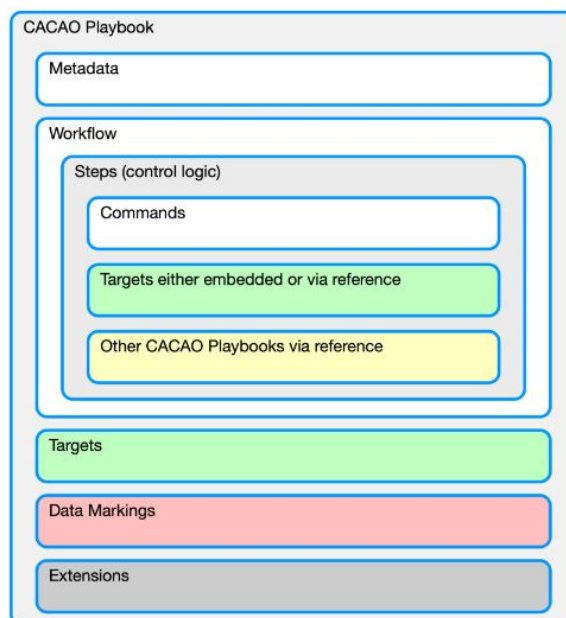


Figure 10: OASIS playbook structure (Open, 2021)

The playbook metadata is a group of information that characterizes the playbook. This section should contain the type of playbook, which should specify the opera-

tional functions the playbook addresses, who and when the playbook was created, a name and a description for it, some versioning information, among another fields that should be present in the case that they can be applied to the playbook.

The workflow section contains a series of steps, that long with the associated commands form the building blocks for playbooks. Workflows process steps either sequentially, in parallel, or both depending on the type of steps required by the playbook. The commands contain detailed information about the commands that are to be executed or processed automatically or manually as part of a workflow step. Targets contain detailed information about the entities or devices that accept, receive, process, or execute one or more commands as defined in a workflow step. Targets contain the information needed to send commands as defined in steps to devices or humans. There also the possibility that inside one playbook, another playbooks may be triggered.

Data markings represent restrictions, permissions, and other guidance for how playbooks can be used and shared. For example, playbooks may be shared with the restriction that it must not be re-shared, or that it must be encrypted at rest.

Extensions can be used to refer to all objects that may be used in other parts of a playbook reference something of importance.

This are some of the more general guidelines offered in this specification. These concepts can all be found in greater depth in the original specification document. In conclusion, this specification offers good guidance when building a playbook, which certainly helps creating an effective playbook, that can be well understood by the parties involved.

2.2.3 *Playbook vs Runbook*

Another term that is becoming more popular recently in the context of security automation recently is the term runbook. In most [SOAR](#) software, the automation of the actions that the playbooks previously mentioned implement, are achieved by creating and deploying runbooks. Runbooks are a way of defining multiple sequential actions to be taken after a certain trigger event occurs. The terms runbook and playbook are many times used interchangeably which often lead to confusion in their meaning. They both serve as guides for a set of actions that need to be taken when handling a specific situation. Although, playbooks deal with overarching responses to larger issues, giving higher level instructions which

are more targeted to be followed by humans, while runbooks are more focused on defining individual processes, that are often more easily automated. Runbooks are often applied to perform actions, such as data enrichment, threat containment and sending notifications while handling an incident response, which help to accelerate the incident response process (DFLabs, 2019).

Runbook automation via **SOAR** software, allows the creating runbooks and performing them automatically (figure 11). In this situation there can still be room for human intervention. Some of this runbooks may be implemented in a way that a part of the actions needed to be taken are automated, but human interaction still occurs at some predetermined points in the execution.

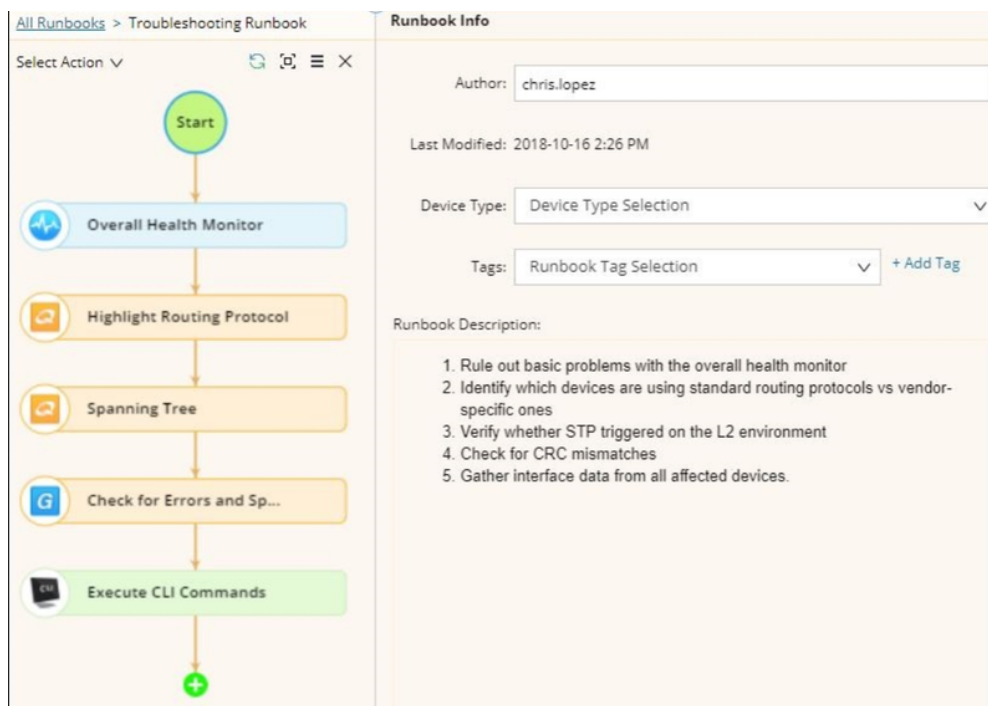


Figure 11: Runbook example (Foolcdn, 2020)

2.2.4 *SOAR Architecture*

A **SOAR** tool is design to be integrated inside the company technology infrastructure and work directly with the other organization software. As is the case with many other types of software applications, **SOAR** software may be deployed as **SAAS**, not necessarily in the physical infrastructure of the organization. The figure 11 summarizes the main interactions that the **SOAR** software has inside the company systems.

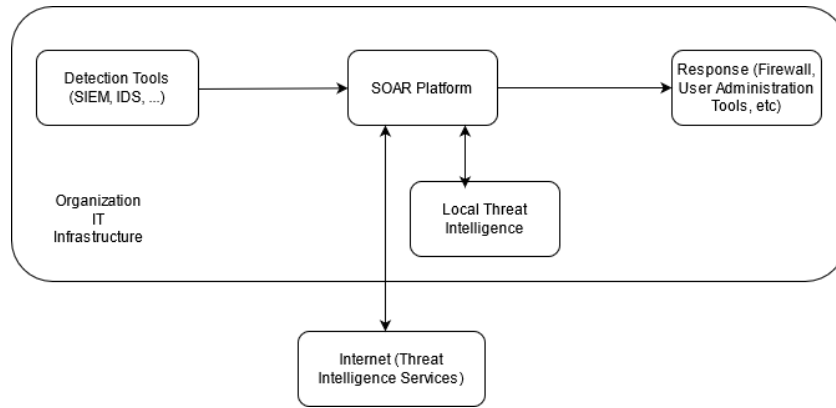


Figure 12: SOAR Deployment

The **SOAR** software obtains logs from the organizations monitoring tool such as **SIEM** and **IDS**, and processes this information. Then, it can use threat intelligence services to further enrich the original information or to classify it. These threat intelligence services can either be services available on the internet such as Virustotal, or a local service that the organization owns to keep track of its incident history. Finally, the **SOAR** platform, acts on other company software to take measures to handle the incident, like blocking an IP address on the firewall or reset an user access credentials.

2.2.5 *SOAR Characteristics*

In order to later be able to choose a **SOAR** software to use in the implementation of this project, it is important to understand what are the characteristics that one should possess (figure 13).

According to Exabeam (Exabeam, 2020), a **SOAR** software should be able to provide three main features: Orchestration, Automation and Case Handling.

2.2.5.1 *Orchestration*

One of the key capabilities is the capacity to work as a security orchestration platform. Security orchestration is the process of integrating a disparate ecosystem of **SOC** tools and processes to automate tasks for simpler, more effective security operations. This is one of the main features that will help solve the before mentioned problem of the many security related software that **SOC** teams need to use today, to cover all their security issues. Security orchestration solves these problems by creating connections/integration's between processes and technologies, so that most

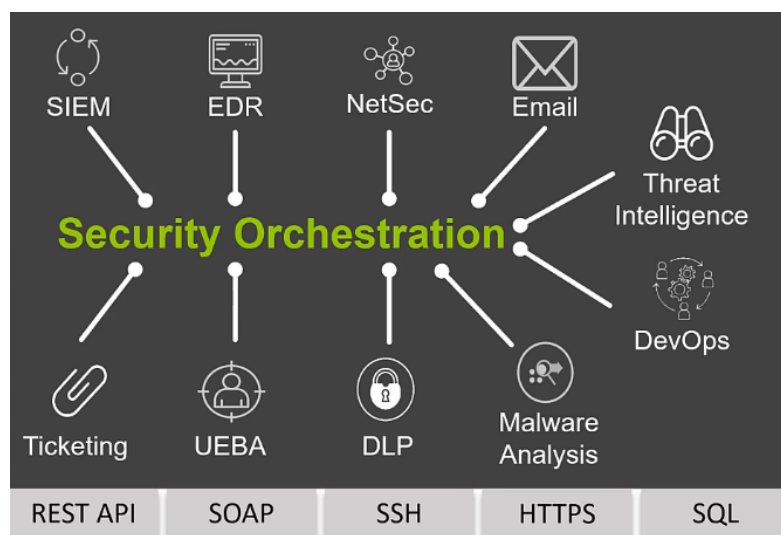


Figure 13: Security Orchestration (PaloAlto, 2019)

day-to-day [SOC](#) tasks can be completed with much less effort. [SOAR](#) tools can do this by integrating with other security solutions in a way that lets them “pull” data and also “push” proactive actions. [SOAR](#) provides a generic interface, allowing analysts to define actions on security tools and [IT](#) systems without having to be experts in those systems or their APIs.

One common example for orchestration, that is an issue often found in [SOC](#), is the handling of a suspicious email. One possible set of actions that a [SOAR](#) tool may take is (Exabeam, 2020):

- A [SOAR](#) tool can investigate whether the sender has a bad reputation, via threat intelligence, and use `()dns` tools to confirm the origin;
- The tool can automatically extract hyperlinks and validate them via [URL](#) reputation, detonate the links in a secure environment, or run attachments in a sandbox;
- Then, if an incident is confirmed, a playbook is run. The playbook looks in the email system to find all messages from the same sender or with the same links or attachments and quarantines them.

2.2.5.2 Automation

Automation is related to orchestration, it is machine-driven execution of actions on security tools and IT systems, as part of a response to an incident. This automation is done on security actions with the power to programmatically detect, investigate and remediate cyberthreats, with or without human intervention. [SOAR](#) tools, should

allow security teams to define standardized automation steps and a decision-making workflow, with enforcement, status tracking and auditing capabilities(Exabeam, 2020). This automation can as soon as an alert comes in, determine whether an action is required, based on previous similar incidents. If the alert truly represents a threat, then it can then take further actions or, if on the other hand the alert is found to be false, it can be immediately discarded, without a security analyst having to waste time analysing it

In order to build automation in these type of tools, generally the security analysts develop security runbooks, which analysts can create using a visual interface or a programming language. An example of an automation playbook can be the handling of a malware file:

- Scanning the malware file and detonating the file in a sandbox using external services;
- Checking the file with external reputation services such as VirusTotal;
- Identifying the geolocation of the source or originating IP address;
- Notifying the user about the malware and performing a post-analysis cleanup.

2.2.5.3 *Incident Management and Collaboration*

Another important feature of a **SOAR** tool is its features that enable collaboration between the members of the security team, and let's them manage incidents in a efficient and transparent way.

There are multiple features that contribute for a good management and collaboration platform (Exabeam, 2020):

- Case Management - As previously mentioned, case management is how each case is managed throughout its lifetime within the system. A **SOAR** tool should allow to record actions and decisions made by the security team, making them visible to the entire organization, as well as external auditors. Over time, this will create the an organizational knowledge base of the past incidents, historical responses and decisions and their outcomes, which can be referenced when dealing with future cases;
- Management of Threat Intelligence - Threat intelligence significantly reduces the time needed to manually research and triage alerts by supplying **SOAR** solutions with automated intelligence in real time. A **SOAR** tools brings in threat data from open-source databases, industry leaders, coordinated response

organizations, and commercial threat intelligence providers. Afterwards, it attaches the relevant threat information to specific incidents, and makes threat intelligence easily accessible to analysts as they are investigating an incident;

- Alert Processing and Triage - One key factor to reduce the alert fatigue that analyst face, is being able to do a triage of the alert as soon as they are generated. A [SOAR](#) tool can gather and analyze security data and correlate data to identify priority, and automatically generates incidents for investigation. This way, analyst can focus on analysing the more relevant cases first, and when they do, this cases can already be enriched with relevant information that the [SOAR](#) tool has already added to it. This removes the need for a human to notice the relevant security data, identify it as a security incident and manually set up an incident in the system;
- Journaling and Evidentiary Support - Finally, it is also very useful for a team analyst working in a [SOC](#), to have an easy and efficient way of storing and consulting artifacts that are created during the handling of a case current and future analysis.

2.3 SOAR SOFTWARE SOLUTIONS

The second phase of this project consisted on carrying out an analysis of the available solutions on the market. In this step the objective was to gather information on the different solutions available, so that later an informed choice could be made on what solution to use for the existent environment.

This meant searching for [SOAR](#) software solutions and analysing its features through their documentation or even by deploying and exploring some of the solutions. Due to time constraints it was not possible to fully analyse all the found solutions. Because of this it was decided to explore options that for some reason were found be of more interest to this project.

2.3.1 *Opensource solutions*

Since security orchestration is still a relatively new concept the number of opensource solutions is limited. Three different software were identified in this category.

2.3.1.1 Walkoff

Walkoff is an automation framework, which through integration with various different software, allows users to define sequences of actions. The goal of this software is, like any other [SOAR](#), to provide users a way of automating repetitive tasks ([WALKOFF github 2020](#)).

Walkoff aims to offer:

- Easy-to-use: Drag-and-drop workflow editor. Sharable apps and workflows;
- Flexibility: Deployable on Windows or Linux;
- Modular: Plug and play integration of almost anything with easy-to-develop applications;
- Visual Analytics: Send workflow data to custom dashboards.

In this software, users have a drag and drop interface (figure 14) which let's them build workflows. These workflows represent a sequence of actions that are executed once a certain event occurs. ([WALKOFF Presentation n.d.](#))

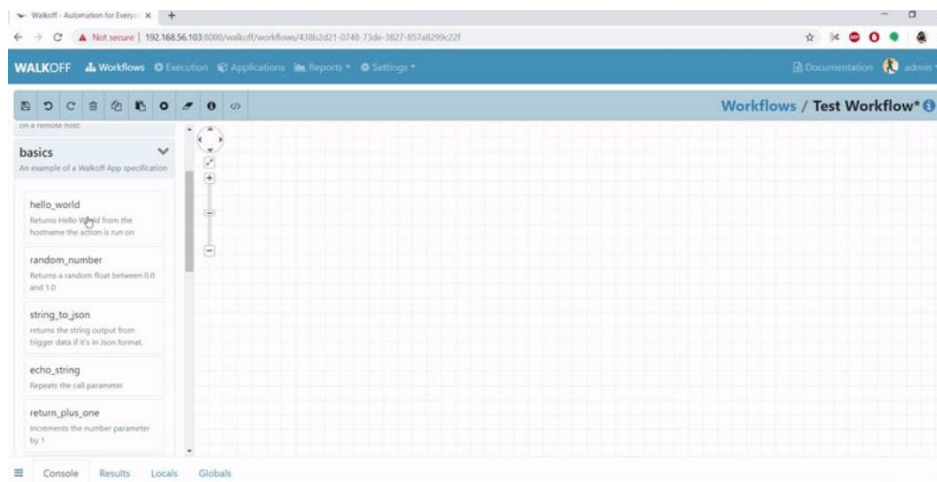


Figure 14: Walkoff Workflow builder ([WALKOFF Documentation n.d.](#))

Applications are what enables the integration with other software. These applications have the specifications of the [API](#) of the corresponding software already defined and only require the user to input the specific information he wishes to send like api keys and parameter values. Walkoff already comes with many application built out of the box, this way users only have to drag them to their workflow and start integrating. If it an application to a certain software does not exist already, users may create their own to integrate with their desired softwares. On (figure

15) we can see the application menu, where users can create, edit or delete their applications (*WALKOFF Documentation n.d.*).

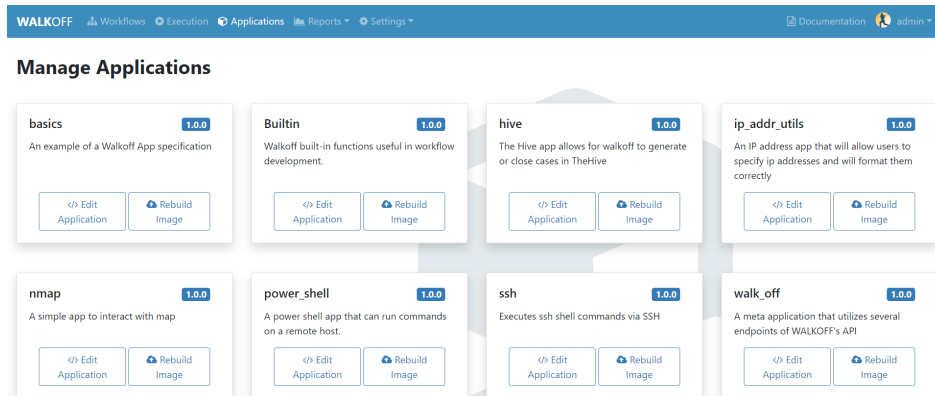


Figure 15: Walkoff Applications Menu (*WALKOFF Documentation n.d.*)

Walkoff is a competent tool in terms of orchestration and automation capabilities. It allows the creation of workflows, which can be used to implement runbooks built for the specific needs of a **SOC**, with some integration's with common apps already included. On the other hand it lacks tools to help analysts with incident management and collaboration. It does not offer a centralized area where incidents can be created and followed by the analysts or any history feature. The feature that Walkoff provides can certainly be of use in a **SOC**, if used together with another tool that is able to deliver some of the features missing.

Although this software does have some feature of value, its development was dropped. This means that no new features will be developed, nor any existing bugs will get fixed in the future. In the context of cybersecurity this is a huge problem, as everyday are new threats and issues that need new ways to be tackled. This situation makes Walkoff a very poor choice for an implementation in a production environment.

2.3.1.2 *Shuffle*

Shuffle is another opensource **SOAR** platform with many similarities with Walkoff. The creator of Shuffle started this project when he found himself writing the same code again and again in an effort to integrate multiple systems, which is not only a tedious process but also a time consuming one. It was with this idea in mind, of streamlining the process of integrating different applications, that this software was created (Oedegaardstuen, 2020).

Shuffle follows a principle similar to Walkoff to automate processes. The actions that are taken after a certain event is detected are built and defined using workflows (figure 16).



Figure 16: Shuffle Workflow (Oedegaardstuen, 2020)

This workflows contain the many applications which enable the interaction/integration with other software. On the connections between applications it is possible to define conditions based on many factors such as the information returned from the execution of a application. This allows to create flows like, if a scan of an IP Address from a service with blacklists returns a match, automatically make a call do the firewall API to block this IP.

Shuffle also uses the same integration approach and structure of Walkoff, meaning that their apps work with Shuffle as well. On top of this Shuffle also uses OpenAPI, together with a builder which should allow the user to create a integration with a new application much more easily and quickly, than having to build everything from scratch (figure 17).

Shuffle can be deployed in two forms, either with a on premise installation or with a [Software as a Service \(SAAS\)](#) approach, the latter requires a monthly subscription which price depends on the subscribed plan, from 15 to 999 \$.

Similarly to Walkoff, Shuffle also has some very interesting capabilities in terms of orchestration and automation of security tasks, and, still similarly to Walkoff, it lacks features geared towards incident management and team collaboration. However, unlike Walkoff, Shuffle is a newer software, still under active development, which was the major drawback that Walkoff had.

General information
[Click here to learn more about app creation](#)

Name
 Virustotal V3

Description
 A description for the service

API information
 Base URL - leave empty if user changeable
 https://www.virustotal.com/api/v3
Must start with http(s):// and CANT end with /

Authentication
 API key

API key
 x-apikey
Can't be empty. Can't contain any of the following characters: !#\$%&^*+_-~|]+\$

Field type
 Header

Figure 17: Shuffle New Application Creation (Oedegaardstuen, 2020)

Ultimately, Shuffle presents some useful features for security automation, and overall is a valuable solution for what is required for this project, and although it is a relatively new project (first release was on 26 June 2020), it can bring value to a SOC and has an active community supporting its development.

2.3.1.3 The Hive

TheHive is also an opensource platform dedicated to automate processes related to information security. TheHive lets the user analyze bulks of *observables*, which can be an IP or email addresses, URLs, domain names, files or hashes, they have collected, all at once, by querying a single tool instead of several.

On (figure 18) we can see a list of Alerts created in TheHive. This Alerts correspond to one or more observables and can be created either by user manually or automatically from other applications like SIEM. Analysts can then choose to analyze the *observables* in this Alerts by using *analyzers*. This *analyzers* are part of another tool TheHive integrates closely with called Cortex. Cortex has already hundreds of these *analyzers* built in, that analysts can use to automatically get

feedback from DomainTools, VirusTotal, PassiveTotal, Joe Sandbox, geolocation, threat feed lookups and so on (*TheHive Github n.d.*).

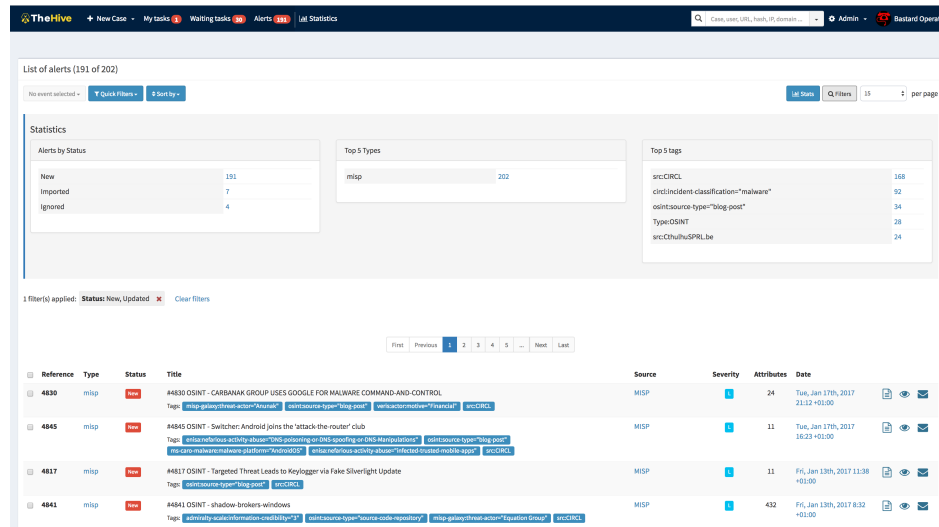


Figure 18: The Hive Dashboard (*TheHive Github n.d.*)

TheHive also integrates with [Malware Information Sharing Platform \(MISP\)](#). [MISP](#) is an opensource threat intel sharing platform, that allows any organization to store and maintain a database of the threats it has experienced in a structured way, together with any information that was learn from them. This results in a searchable history of threat events. The fact that this information is saved in a well defined and structured way enables the possibility of multiple organizations sharing the information they have gathered, which is what makes [MISP](#) a very powerful tool when dealing with threats.

Through the integration of [MISP](#) with TheHive it is possible to create a case out of a [MISP](#) event. TheHive can be configured to receive Alerts from [MISP](#) events, then this Alerts can be processed and be previewed to decide whether they warrant an investigation or not (figure 18). If it is decided that the Alert should be investigated further, a case in TheHive is created where an analyst can choose which actions to take.

Another very interesting feature of TheHive, is the way it enables collaboration between members of the security team. By having a centralized dashboard (figure 20), where all the cases that are happening in the moment are displayed, all members of the team know whats happening. They can easily see which cases are still unattended, if there are any high severity cases, who is already working on what, etc.

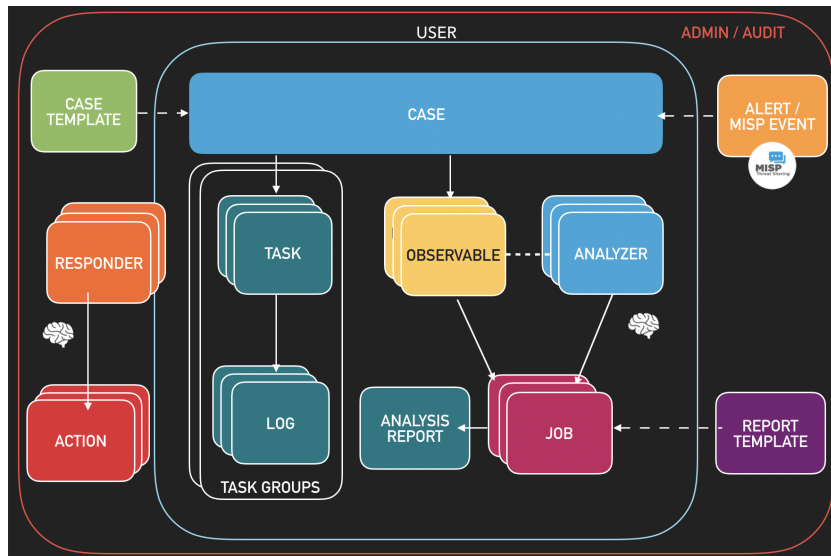


Figure 19: The Hive Workflow (*TheHive Github n.d.*)

Unlike the two previously explored solutions, TheHive lacks in terms of automation and orchestrations capabilities, as it only integrates with a specific set of security tools. This makes it a very limited solution, as it cannot automate any desired process that may already exist in the SOC. Nonetheless, TheHive is definitely a very powerful tool when dealing with security incidents, already packed from the start with very useful capabilities. It excels in some of the feature that the other solutions are lacking, mainly in case management and threat intelligence, as it comes with many features in these areas out of the box.

While TheHive on its own is not capable of providing all the necessary functions that are required of a SOAR software, it still brings very valuable tools to a SOC. Moreover, it brings some features that the other two free solutions fail to provide, which may make it a great choice to work in conjunction with the other tools.

2.3.2 Commercial software solutions

When it comes to the market of paid software solutions there are many more options available. The issue with this solutions, is that they are generally very expensive. Based on the prices of some commercial SOAR software investigated, like Siemplify that as a 2500\$ monthly for fee (Siemplify, 2021), or XSOAR by PaloAlto with price in the tens of thousands of dollars depending on the version (ITPrice, 2021), any paid options are out of question, as there is no budget for this project.

The screenshot shows the TheHive web interface. The top navigation bar includes 'TheHive', 'New Case', 'My tasks', 'Waiting tasks', 'Alerts', and 'List Statistics'. The main content area displays a 'List of cases (11 of 26)' with a table of case details. The table has columns for Title, Severity, Tasks, Observables, Assignee, and Date. The right sidebar shows a detailed view of a case, including its status, resolution status, and impact status.

Title	Severity	Tasks	Observables	Assignee	Date
#19 - [MISP] #1310 OSINT - Sobey's 'Mammoth' OS-X Trigen by Palo Alto networks	High	5 Tasks	4	[Avatar]	01/24/17 9:00
#24 - [MISP] #3329 OSINT - ASERT Threat Intelligence Report 2016-03 The Four Element Sword Engagement	High	5 Tasks	53	[Avatar]	02/09/17 12:03
#21 - [MISP] #4855 OSINT - Nemucod downloader spreading via Facebook	High	5 Tasks	5	[Avatar]	01/24/17 11:37
#20 - [MISP] #1107 OSINT - Turbo Taxis: Two 64-bit Dersabi Stearns Converge	High	5 Tasks	10	[Avatar]	01/24/17 9:04
#17 - #3024 OSINT - In the Shadows: Vavtrak Aims to Get Stealthier by adding New Data Cloaking	High	No Tasks	20	[Avatar]	01/22/17 12:17
#15 - #13-#1395 Malpam 2016-09-22 (1.jp in .jsp) - campaign: "Delivery ID [integrity] / #14-Suspicious URL	High	No Tasks	16	[Avatar]	12/13/16 13:17
#12 - #11 (Malpam) 2016-09-15 - "SCAN" Campaign 7 / #10-#3410 Malpam 2016-09-15 (waf in .jsp) - campaign: "SCAN"	High	7 Tasks	12	[Avatar]	12/13/16 10:24
#6 - #2111 OSINT - Malpam delivers NanoCore RAT	High	No Tasks	1	[Avatar]	12/07/16 22:23
#4 - #3414 OSINT OSX/Printzid Backdoor Additional Details by Zanz / Eric Romang	High	No Tasks	2	[Avatar]	12/07/16 22:20
#3 - #3413 Malpam (2016-04-28) - Lucky RDD	High	No Tasks	19	[Avatar]	12/07/16 22:18
#2 - #3407 NanoCore related activities	High	No Tasks	2	[Avatar]	12/07/16 22:17

Figure 20: The Hive Current Cases (*TheHive Github n.d.*)

Even though, some of this solutions have free versions (generally referred as *Community Edition*) which come with many different types of limitations. Having this in consideration it was decided to investigate some of this options so that it could be made a comparison with the opensource solutions, to identify what advantages these may bring, or to understand how limiting the free version are.

2.3.2.1 *Siemplify*

Siemplify was one of the commercial options analysed. The fact that Siemplify made a partnership with Checkpoint (another security software provider) which is used in the *IPLeiria SOC (Siemplify and Checkpoint Partnership 2020)*, made this a particular interesting options to analyse.

Similarly to the other solutions analysed, Siemplify makes use of runbooks and applications to create workflows and integration's respectively (figure 21).

One of the big advantages of Siemplify is the amount of applications that are available from the start. In a section called marketplace, it is possible to find all the many applications already built. Simply clicking install on the desired application, downloads it and makes it available to use on a local runbook.

In addition to the applications it is also possible to download entire runbooks. This runbooks come already built around a specific use case, and try to solve a specific problem. After downloading a playbook, a configuration wizard is launched. In this wizard the user can introduce the proprietary information needed like API

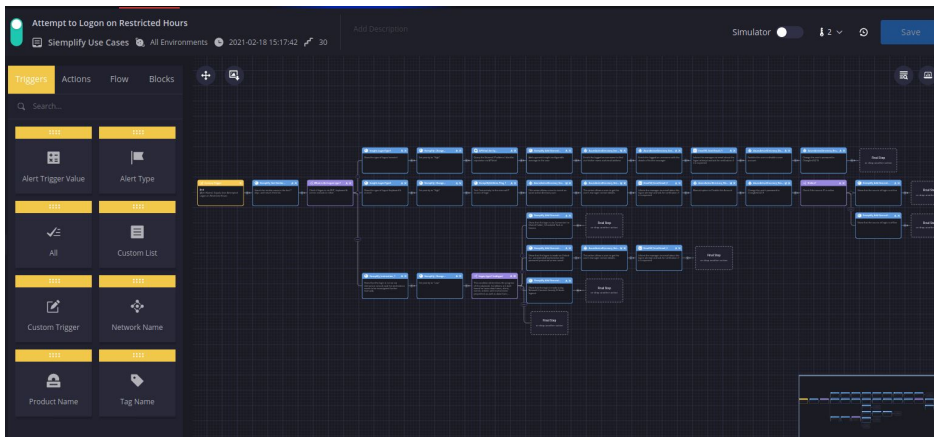


Figure 21: Siemplify Runbook

keys of the software that is being used and other necessary information (figure 22). Once this setup is made the runbook can be further customized or simply deployed.

Another convenient feature of Siemplify, is that from the close development it had with Checkpoint software, there are already some already built in runbooks for checkpoint software out of the box. This runbooks include flows to automate firewall audits and remediate any rule violations with the Checkpoint Firewall or network threat prevention with automated malware analysis together with Checkpoint SandBlast.

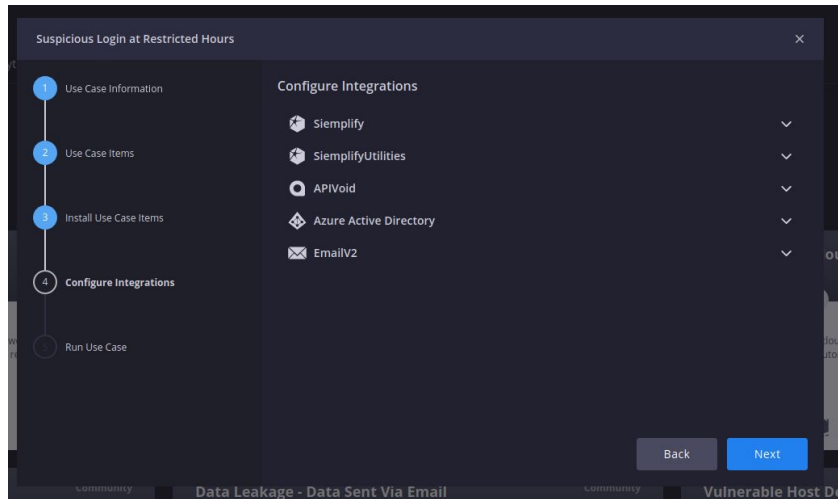


Figure 22: Siemplify Prebuilt Runbook

Siemplify also includes a case management feature, that can create cases automatically from alert information and prioritize them, threat intelligence resources, among other useful features for team collaboration, such as a dashboard where many different statistics can be seen (figure 23), which is automatically populated

with information that Simplify captures and can be further customized to suit the user needs.

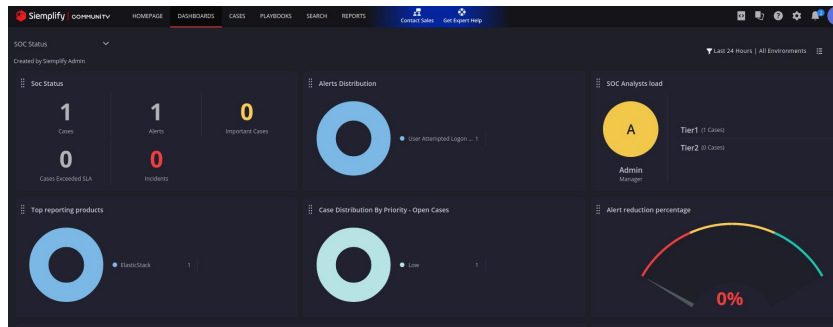


Figure 23: Simplify Statistics Dashboard

Simplify is able to bring together many of the features that are required to make a good security orchestration and automation platform in a single environment. Moreover, it does this while providing a clean and intuitive interface with a more fluid and stable operation compared to the previous analysed solutions. This makes Simplify a vastly superior option relatively to the opensource tools, in term of its usability and features, although the price point for the commercial version, is very high.

Simplify also offers a free version, which comes with various limitations over the commercial version. These limitation include the limit of having at most 5 runbooks, a maximum of 25 daily alerts that can be handled, the limit of only one user on the platform, among others.

2.3.2.2 Splunk Phantom

Another commercial solution analyzed was the **SOAR** from Splunk, Splunk Phantom. This solution was chosen based on fact the the **IPLeiria SOC** already uses Splunk as a **SIEM**, and using multiple products from the same company may bring benefits in integration's and general inter-operation between them.

Again, like the other solutions, Splunk Phantom uses of runbooks and applications to create workflows and integration's respectively (figure 24).

Similarly to Simplify, Splunk Phantom already as available many runbooks ready to be deployed for common security operations tasks. This include tasks like reset accounts passwords upon suspicious activity, automatic analysis, etc. In terms of integration's built in, Splunk Phantom also has large number of them, covering many of the more common security software.

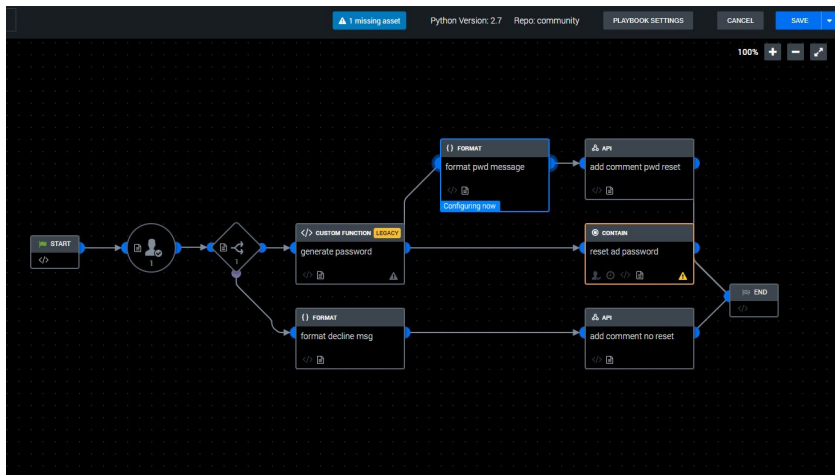


Figure 24: Splunk Phantom Runbook

Also similarly to Siemplify, Splunk Phantom also brings case management features built in, focusing in creating an environment where it is easy for different teams to collaborate with each other. Together with many different configurable statistics, there are areas where the current incidents can be seen, as well as if they are already being investigated or not (figure 25).

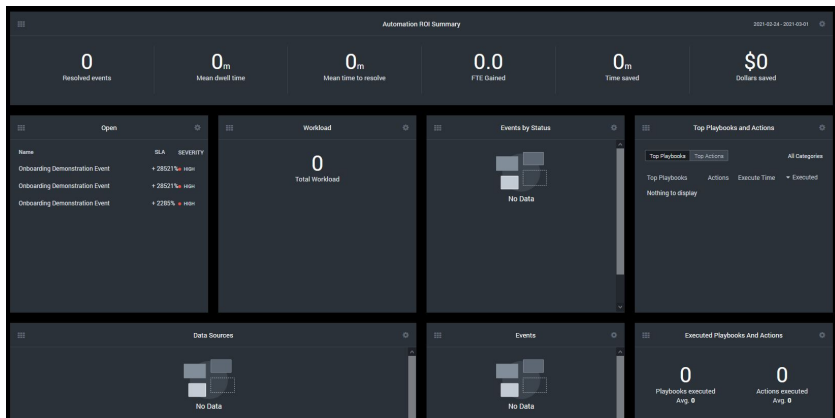


Figure 25: Splunk Phantom Dashboard

In terms of functionalities, Splunk Phantom is very similar to Siemplify. It brings together case management features with security automation and orchestration. The main difference identified when comparing these two solutions is the interface that is not as user friendly and intuitive, while also having a slightly more dated appearance in the case of Splunk Phantom. In any case, Splunk Phantom is also a commercial software with a very high price point as states previously.

Just like Siemplify, Splunk Phantom as a community version available. Many of the limitations are similar to the limitations of Siemplify, like only allowing one user on the platform. However, while the Siemplify community version has a limitation in

terms of the the number of alerts that can be processed, Splunk Phantom community edition has a limitation in the number of daily actions that can be executed of 100 (*Splunk community versions details 2021*). With mostly everything that occurs inside a runbook being an action, this limit can be easily reached.

2.4 SUMMARY

In light of the evident increase in cyberattacks that organizations are victim today, many new technologies and software solutions have been appearing on the market, to help them handle this issue. Organizations nowadays, also rely heavily on their **IT** systems to operate normally, and failure's in these systems may lead to serious financial costs, not only directly by not being able to supply their services to costumers, but also by falling in non compliance with legislative measures, that may eventually lead to significant fines. Because of this situation, most organizations, that have event moderately small **IT** systems, are already using many of these software, in an effort to avoid any downtime of their services or having any security breaches that may damage their reputation.

The many security related applications that the organizations generate a huge amount of information, that in many cases cannot be handled properly by the security staff, which is lacking as well in many cases. To help analysts make use of all these like **SIEM** and **SOAR**. While **SIEM** main objective is to aggregate and categorize the information, **SOAR** expands in this concept by providing more functionalities mainly focused on automation, to reduce the workload needed to be carried out by analysts.

Regarding cybersecurity incidents, there is already many research and procedures developed, on how these can be handled by an organization. The **NIST** and the **SANS** frameworks offer a model on how an **IRP** can be developed to suit a particular organization needs. This **IRP** should contain all the information that an element of the organization might need when handling any kind of security incident the organization suffers effectively. Furthermore, for a more efficient incident response process, there should also be playbooks, defined by the organization, that explain in a more concrete way how more specific actions involved in the incident handling should be carried out (figure 27). Again, for these type of documents, there are also models on how to effectively develop them, being the specification by **OASIS** a good reference.

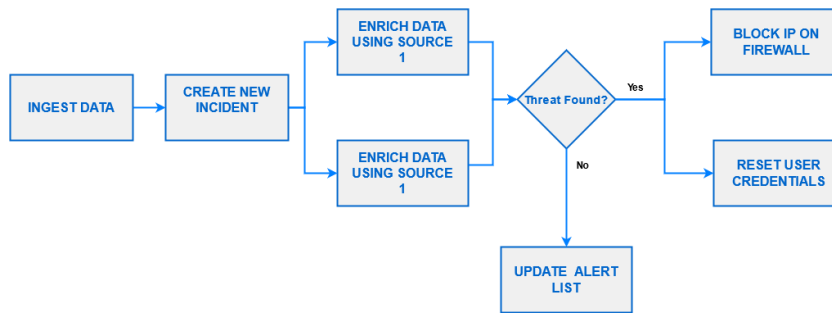


Figure 26: Playbook to block IP and reset user credentials from ingested data

To help SOC addressing the issue of the enormous amount of information being generated while implementing an effective incident response mechanism, there have been developed platform designated as **Security Orchestration, Automation and Response (SOAR)** tools. These are software solutions that are deployed in the organization IT systems, and that are able to integrate with the many other applications there. These integration combined with their automation capabilities provide a way of automating many of the operations analysts usually have to carry out manually, while ,in some cases, also providing a centralized platform for team collaboration.

An analysis conducted in the current market of SOAR solutions, revealed that there are both commercial and free options available currently (figure 27).



Figure 27: SOAR solutions analysed

The commercial options, are being developed, in most cases, by big companies in the IT industry and come with a high price tag associated. The free solutions offer significantly less quality and combination of functionalities, although, still prove useful for the job at hand. The items studied in this chapter, help contextualize the problem that this project fundamentally deals with, and will guide its development. The next chapter will detail how a SOAR solution was chosen to be used in the IPLeiria SOC, with the information that was gathered here.

PROOF OF CONCEPT DEVELOPMENT AND TESTING

This section will detail the process of implementing a **SOAR** solution on the **SOC** of the **IPLeiria**.

First the structure of the **SOC** will be analysed to better understand the environment where the solution is going to be deployed.

After a good understanding of the **SOC** structure is achieved and together with the information gathered in the previous chapter about the solutions available in the market, a choice will be made on what software to use.

Once the software is decided, a deeper analysis of the solution is done in order to better understand its capabilities and potential, where some real world scenarios will be replicated.

Finally the process of implementing the solution on the **IPLeiria SOC** is detailed.

3.1 ENVIRONMENT AND CONSTRAINS

The **IPLeiria** is an organization that has been operating and evolving since it was established back in 1987. Likewise, its **IT** infrastructure, has also grown. This means that there is already in place an infrastructure with different systems and technologies where this project has to be deployed.

3.1.1 *Incident Response Procedures*

The **SOC** already has in place general procedures to deal with the security incidents, that should be followed by analysts when dealing with an incident. In figure 28, a diagram of the general procedure followed by the **CSIRT** team can be seen.

In the above diagram the different groups involved in the different steps of the incident response can be identified. The security analysts are the one who start handling any incident, and then do most of the subsequent actions. The **SOC** coordinator has to intervene in particular situations only to approve certain actions

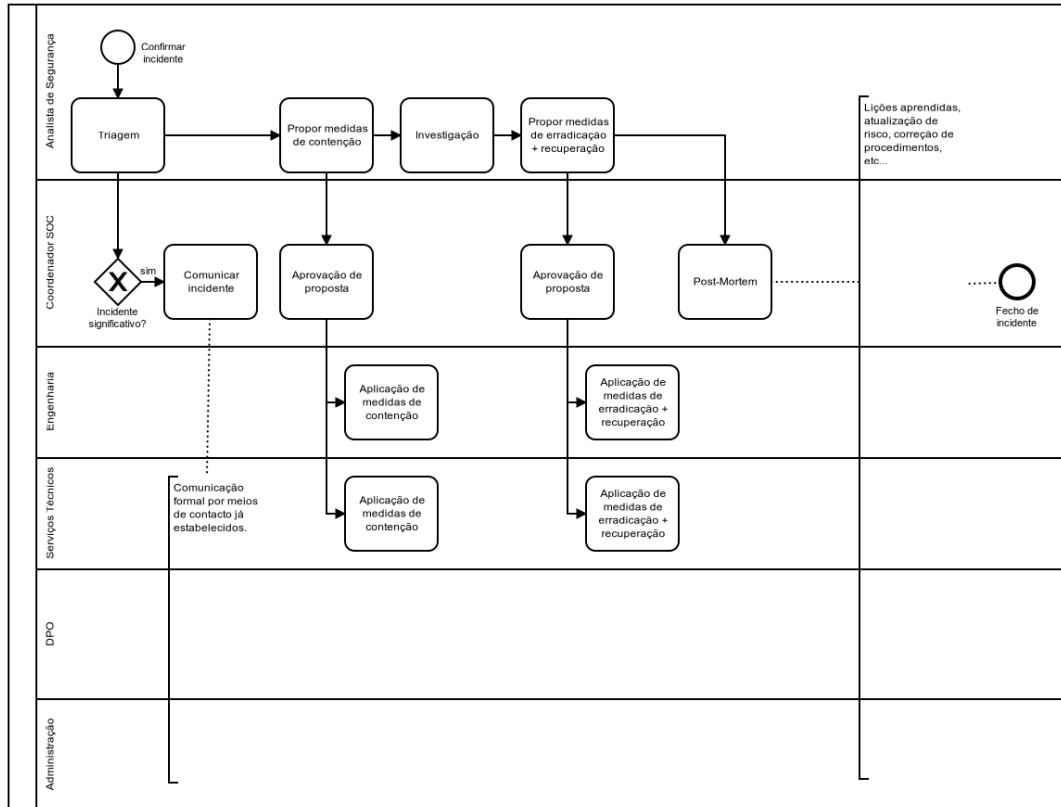


Figure 28: General Procedure of CSIRT

that may need to be taken and to communicate with relevant third party entities in relevant situations. The engineering and technical services teams will handle specific actions that may need to be taken for certain types of incidents. Finally the administration and the data protection officer, only act when communication of the incident is deemed necessary to other parties.

The actions that are supposed to be taken by the SOAR platform, are mainly the ones who are taken by the security analyst, to alleviate the amount of work he has to carry out. The incident response life cycle followed by an analyst of the SOC can be seen in figures 29 and 30.

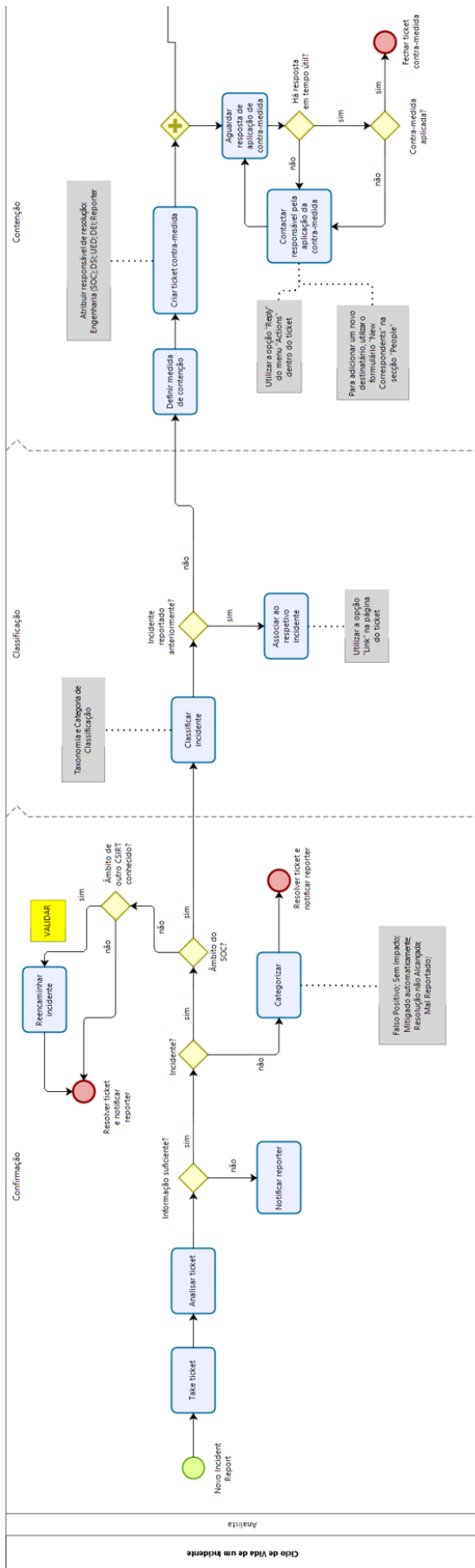


Figure 29: Incident Response Lifecycle 1

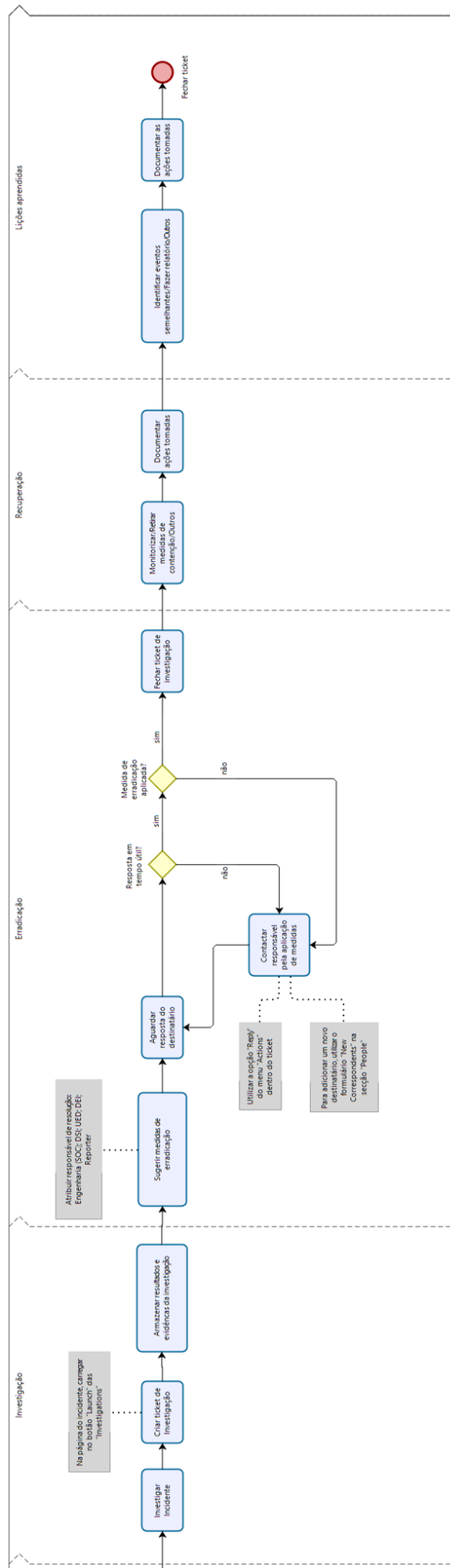


Figure 30: Incident Response Lifecycle 2

The first section contains a set of actions that the analysts carry out during the confirmation of an incident. This step has the objective of quickly analysing the event that was reported and understand if it represents a real incident or not, as well as if it should be handled by the [SOC](#).

Once the incident is found to be of concern to the [SOC](#), it is classified. The objective of this phase is to assess the gravity of the incident and categorize it accordingly.

Afterwards, the containment phase takes place. Here the incident is registered in the case management system of the [SOC](#) and a the entity that should implement the counter measures for the incident is assigned to the incident.

With the counter measures in place further investigation of the incident begins, an investigation ticket is opened and any info regarding the incident is associated with it.

Once the previous investigation finds more information on what caused the incident, the eradication phase starts. Here, a solution to remove whatever negative effects the incident had on the systems is found, and the entities responsible for implementing this actions contacted. With the eradication performed, the investigation ticket is finally closed.

In the last two phases, the analyst monitors the affected systems to ensure everything is now fixed and registers information that might help in dealing with future similar incidents.

This details the processes taken by the analyst from figure 28. The technical services and the engineering, have very different approaches to handling the incidents depending on it's type, for this reason it is not possible to streamline their actions in a single diagram.

The remaining entities are not relevant for the scope of this project.

3.1.2 *Infrastructure*

Since this is not a standalone project, on the contrary, it has to integrate with already existent systems, it is important to understand what these systems are and how they are setup, so that the integration can be done correctly and efficiently. As this project only concerns the security of the [IT](#) systems and is design to be implemented in the [SOC](#), it's systems will be the focus of this analysis.

Currently two SIEM solutions are being used, Elastic Search Logstash, Kibana (ELK) and Splunk. The version of ELK in use is free while Splunk is a paid SIEM. Since Splunk monthly cost is based on the amount of data processed, only logs of high severity are processed by this SIEM, while the rest of the logs are processed and stored in ELK (figure 31). The reason why the rest of the less important logs are still stored in ELK, is because these logs may prove useful if an incident is only detected after it has occurred, to find out more about the incident.

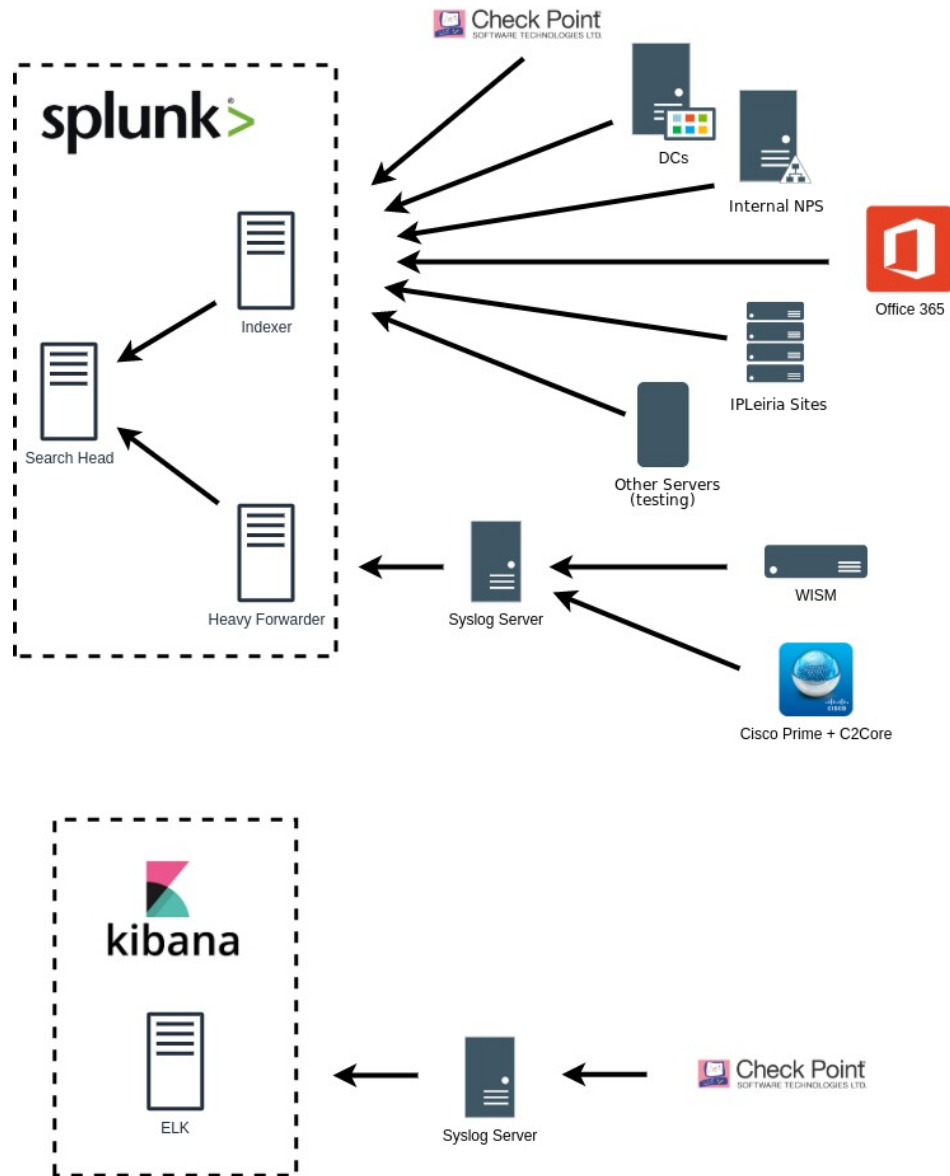


Figure 31: IPLeiria SOC Architecture

Concerning the security operations, it is also important to take into consideration that the organization uses systems such as Cisco network devices, checkpoint

firewalls, Microsoft office365 and Active Directory in its ecosystem and that the chosen **SOAR** solution will eventually have to interact with this systems.

In figure 32, it can be seen the interactions that an analyst of the **SOC** carries out when doing his job. An analyst monitors what is happening through application like Splunk, **ELK**, IPAM and Cisco Prime. In order to keep track of current issues and to also track their progress, the incidents are logged into TheHive and GLPI. If some actions is required to be executed in the Checkpoint Firewall or Office365, like blocking an IP address, the analyst contact the engineering team for this effect.

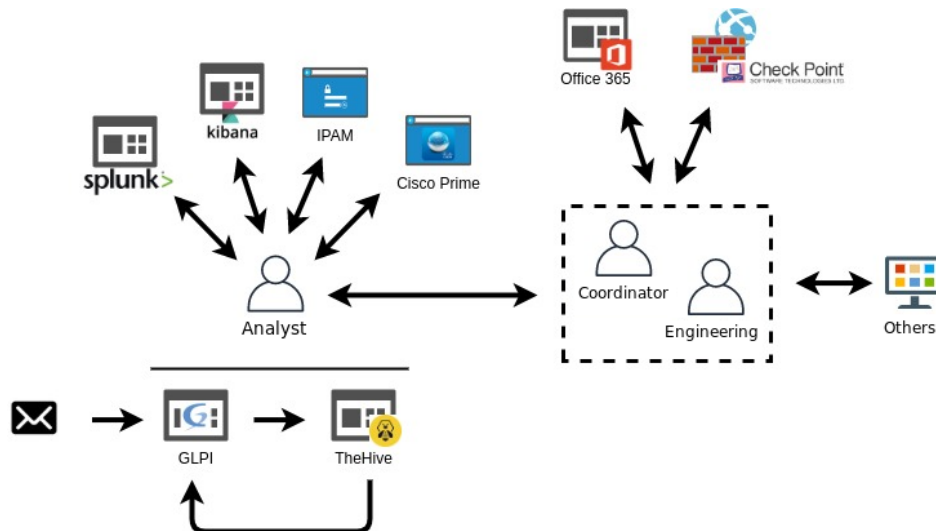


Figure 32: IPLeiria SOC Analyst Interactions

Additionally, another very important constraint of this project is the lack of a budget to acquire commercial software. Because of this a solution with no costs will have to be chosen. This means either working with an opensource solution or using a free version of commercial software.

Lastly, the solution to be implemented, should preferably be a solution with potential to scale with future needs the **SOC** might have. Since the objective of this project is to implement a **SOAR** that can help the **SOC** staff to carry out daily tasks, and these are likely to change in the future, a solution that is only able to satisfy current issues with no margin for new functionalities is not ideal.

3.2 CHOOSING THE SOAR SOFTWARE

One key step of this project is choosing the **SOAR** software to implement the projects requirements. This will have to be a solution that satisfies all the necessary

requirements to work in the [SOC](#) of the [IPLeiria](#), that can be afforded by the institution and preferably one that can serve as a good basis to continue to support future security issues that may appear in the [SOC](#).

From the analysis conducted in the previous chapter, it can be easily seen that the commercial software versions come with clear advantages over the opensource solutions analysed. This advantages come in the shape of different functionalities, like:

- A greater array of integration's already built;
- Playbooks for certain common workflows included;
- Incident management and team collaboration resources built in;
- More intuitive and user friendly interface.

Either Siemplify or Splunk Phantom bring significant advantages over the open-source solutions. Not only are they stabler and easier to work with they also bring many features that are at all not provided by the other solutions. Although, since the full commercial version of this software can't be afforded, the only way to use one of these to develop this project is through the use of the community versions. Since these version come with many limitations to their functionalities, it is then important to first analyse what features are impacted and how they may impact the [IPLeiria SOC](#) necessities, in order to understand if these versions are still viable solutions.

In the case of Splunk Phantom, it comes with the limitation of only being able to automate 100 actions per day. Taken into consideration that one single runbook may include easily ten actions, this would only allow for 10 alerts to be handled by the tool automatically per day. This issue is even more aggravated in the case of developing more complex runbooks which may include a larger number of actions. This limitation makes this the Splunk Phantom a very a weak choice for this project. While some level of automation could still be achieved with this free version, it would not allow to scale the solution in the future to eventually meet new requirements, which would force the [SOC](#) to change to a new tool. This would invalidate the effort made setting up the tool and all the knowledge that was gained during the time working with this solution. This factor alone makes Splunk Phantom Community Version a non viable solution to the project.

This type limitations may not be of concern in the case of a team who wants to start implementing a [SOAR](#) solution in their [SOC](#), knowing they can later afford to evolve to a paid version, removing then these limitations. However, this is not the

case of this project, as it is important that the software chosen is able to handle present and future security requirements for the [IPLeiria SOC](#).

In Siemplify, this same situation happens, although with slightly different characteristics. The main concerning limitation here, is the limit of daily alerts that can be processed, not in the number of actions that are taken inside the runbooks. This is much less restrictive than the Splunk Phantom case, as runbooks can have all the necessary actions performed, enabling the construction of complex playbooks.

Here another limitation comes into play, one of them is the fact that there can only be one user in the platform, which certainly stands in the way of having multiple analysts handling cases in the application. Although, the main purpose of this tool is its automation capabilities, which are not impaired by this factor.

The limitation of only allowing 5 runbooks to be active on the platform can also become problematic. Currently this number of runbooks should be enough for the [SOC](#) needs, but in the future it may become a problem if more automation cases are identified.

Even though there is more flexibility in this solution, it still comes with some limiting factors that can impact how the solution might scale in the future with the [SOC](#) needs.

The only way to avoid this type of software constraints, is to choose an open-source solution, even though these may require more effort to set up, they are more likely to accommodate any future need.

WALKOFF was the first open-source solution analysed, which as explained in the previous chapter is able to accomplish most of the tasks required in terms of orchestration and automation. Still, the fact that this solution is no longer being developed, even though it currently may satisfy all the necessities, will make it obsolete in a short time period. Taking in consideration how fast the field of security information is changing currently, it shouldn't take long before some sort of issues it cannot solve start appearing, leaving the [SOC](#) with an inept and obsolete tool.

Another open-source solution is TheHive. This solution as explained in the previous chapter is already being used in the [SOC](#) of the [IPLeiria](#). TheHive lacks many of the features required of a true [SOAR](#) tool, as it does not have the capability to automate or integrate with many security applications, it simply comes with some automation features, mostly focused in threat intelligence. However, it does bring some very good features in terms of incident management, serving as a very good platform for team collaboration. Alone, this software simply is not adequate to

	Orchestration	Automation	Case Handling	Price	Free Version
Walkoff	✓	✓	✗	Free	-
TheHive	✗	✗	✓	Free	-
Shuffle	✓	✓	✗	Free	-
Siemplify	✓	✓	✓	Out Of Budget	Limited
Splunk Phantom	✓	✓	✓	Out Of Budget	Very Limited

Table 1: SOAR tools characteristics matrix

meet all the necessary security requirements to satisfy with this project, but it can serve as a complementary tool to another that can implement the features it lacks, provided an integration can be made between both.

Shuffle was the last opensource software analysed. It is able to satisfy most of the necessities that have been identified for the [IPLeiria SOC](#), and it also has an active development and community behind it, which are bringing new features and general improvements to the software each week, making it a more solid tool over time. This makes it a good option since not only can the current necessities be met, but may also serve as a good platform for the [SOC](#) to continue to develop and expand its security practices.

The matrix table 1, summarizes these findings.

As previously mentioned, there is no budget for any commercial version of [SOAR](#) software, so the two paid version of the analysed tools, Siemplify and Splunk Phantom, are simply out of question. In terms of the free versions of these commercials options, Siemplify is a clear superior choice due to the fact that it's limitations are not as restrictive as the ones from Splunk Phantom.

In the opensource solutions, Walkoff is comparable to Shuffle in many way, with the huge drawback of being a discontinued tool, which leaves Shuffle as a better option. Lastly TheHive is a tool that just does not provide all the features required for the intended application.

With this, the decision is between Siemplify Community Version and Shuffle. The major drawback of Shuffle is its lack of case handling features. However, this tool can be integrated with TheHive, which is already used in the [SOC](#) of [IPLeiria](#), and has very good case handling capabilities. Taking this into consideration, it is

possible to form a very competent [SOAR](#) solution using this two software, that certainly meets all the requirements of this project.

In the end, the fact that Siemplify Community Version has some limitations that may hinder future developments, weights heavily against it. Even though that Siemplify is a more stable and polished solution than the Shuffle/TheHive combination, the fact that it is possible to use a solution which is able to meet all the basic needs, without any limitation, makes this solution a preferable choice.

3.3 PROOF OF CONCEPT IMPLEMENTATION ENVIRONMENT

In the next sections, the chosen [SOAR](#) solutions will be used to develop runbooks that will tackle some scenarios that are relevant for the reality of the [IPLeiria SOC](#).

For this type of prototyping a [Virtual Machine \(VM\)](#) running the Ubuntu distribution of Linux will be used. In this [VM](#) Shuffle will be deployed, as indicated in its documentation, using Docker (*Shuffle installation guide 2021*). Shuffle is provided with a docker compose file which specifies all the different dependencies required to run shuffle as well as configurations for these. In total Shuffle will use four different containers:

- Frontend - Where the web interface is available, used by any user of the applications to carry out any actions within it, developed in ReactJS (JavaScript);
- Backend - [REST](#) api which is used by the frontend to execute all types of actions the user takes, developed in Go;
- Database - The databased used to persist all the data from the application, based on Google Datastore (NoSQL);
- Orborus - Execution environment which runs Shuffle workflows, developed in Python.

Once this setup is done, the frontend can be accessed through a web browser on the port 3001.

Additionally to Shuffle, as mentioned before, TheHive will also be part of the runbooks to develop, as it will be the case management platform. This means that a local instance of TheHive need to be set up alongside shuffle for this to be possible. Since this is an application that is already being used by the [IPLeiria SOC](#) for this very purpose, this report will only focus on the aspects that directly concern the developed Shuffle runbooks.

Apart from the local setup, there will be another elements used in the upcoming implementation, namely external web services and an Active Directory instance from the [IPLeiria](#) internal network. Both of these elements will be detailed in a later section. Figure 33, illustrates the general elements of this implementation.

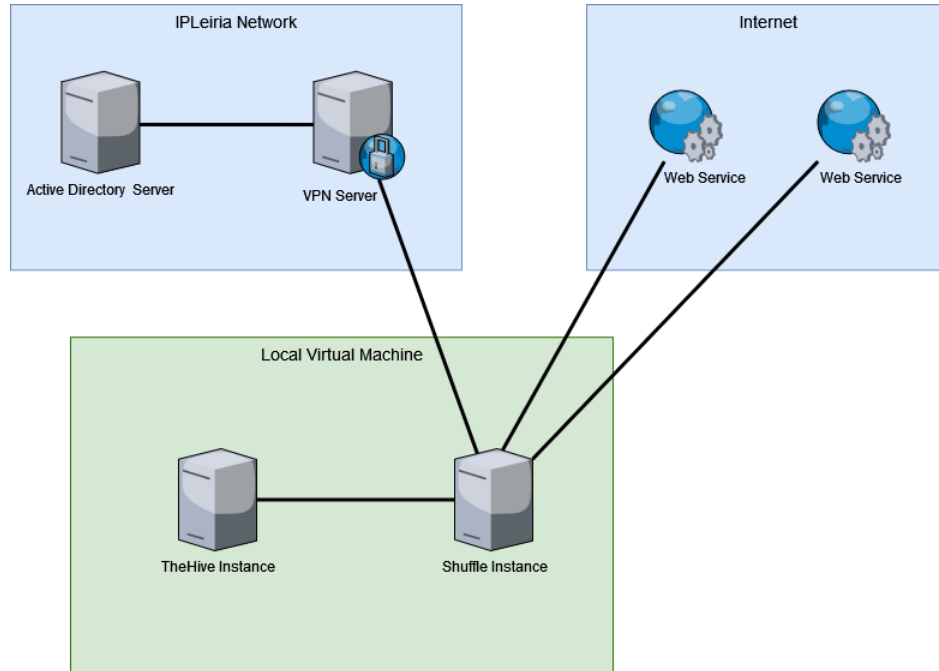


Figure 33: Implementation Environment

3.4 USE CASES

In this section, it will be detailed how this [SOAR](#) solution can be used to build routines that can help with the daily challenges that the [SOC](#) is currently facing.

3.4.1 *Analysing Email With Phishing Suspicion*

Phishing cyber-attacks have been around for a while now, and even though their practice is no secret, they still represent one of the biggest threats online (Gurinaviciute, 2021),(Dosal, 2021). For this reason, one of the use cases to be tackled by this project will be one dealing with a phishing incident.

3.4.1.1 *Phishing*

In a phishing attack, attackers send counterfeit communications that appear to come from a trustworthy source. The most common mean of communication for this attacks is email. In this emails, attackers lure the user into giving sensitive data, such as login information or into installing malware software on their own machines. In some cases the final goal of this attacks is in obtaining information that the attacker can use for it's own benefit, like the victims credit card number. Other attacks are carried out with the goal hijacking computer networks/systems of corporation or individuals until a ransom fee is delivered (Cisco, 2021).

3.4.1.2 *Phishing Incident Response Guidelines*

Before starting the process of trying to streamline some of the processes in the SOC of the IPLeiria, related to phishing incidents, it is important to gather some information of what are the general objectives when dealing with this type of incidents.

The article on building a phishing response playbook (Das, 2021), was followed to create the basic structure of the playbook and to define which actions need to take place during its execution.

As with most security incidents, there is a different set of actions that should be taken in different phases of the incident.

The first phase consists in the identification of the threat. This is the moment when something that might indicate the existence of a phishing attack happens, and it must be further investigated. At this moment it is important to gather as much information about the phishing email as possible to be used in future steps. The message should be carefully examined by a security analyst, which should further investigate any attachment or suspicious link it contains, obviously taking all the necessary precautions to not endanger anything with these actions.

After this first moment of dealing with the threat, it is important to proceed to the triage of the alert. This is an important step as it will determine the priority of the incident. If an incident has the potential of damaging the organization greatly it should be immediately tackled, while if something of minor importance, it is preferable to not disrupt any other processes to deal with it. Once the priority of the attack has been determined, it should be assigned a level of priority, so that

every member working on incident response knows when it is appropriate to handle it.

Once it is appropriate, the investigation of the incident should begin. In this step different components of the email should be analysed.

- The header of the email will contain information about the name, email and mail server used by the attacker;
- The body of the email which contains the actual message should be analysed to search for any telltale signs of a phishing message;
- Any links the email might contains, as they could point to spoofed websites.

Lastly, while investigating the case, it is important to also ascertain the number of employees impacted by the email, what actions were taken with regards to the phishing email and what type of systems may have been impacted by this (servers, workstations, wireless devices,etc).

With the investigation of the incident completed, and knowing now the full extent of the attack, it is time to take measures to contain and remediate the situation. The main objective of this phase is to regain control of any system the attackers may have gained access to, or ensure they lose access. This translates on revoking any authentication credentials they may have obtained such as:

- Changing the passwords or usernames of any directly impacted employees;
- If a point of the [IT](#) infrastructure was impacted, change these credentials for anyone who has access to it;
- Block any discovery harmful IP/Domain on the organization firewall to avoid further damage;
- Wipe any affected smartphones, so that any sort of sensitive information/data that resides on them cannot be accessed;
- Monitor all systems within the [IT](#) infrastructure for any unusual anomalies that may be occurring, and if they occur, consider shutting them down until further analysis.

Lastly, ensure that some measure are taken to avoid future incidents of the same type. One key step of this phase is ensuring that all the information gathered while the incident was being handled is properly stored, so that it can be used in any future event of the same kind. On top of this there are many other actions to take like reviewing with the organization members what went wrong and how to avoid it in the future, as well as conducting training programs targeted to the type of

incident that happened, checking/reviewing the mechanisms used to deploy software upgrades on companies systems, among other general procedures.

This is an example of a standard approach on what to do when faced with a phishing incident, although, the steps presented here are designed with the mindset that a security analyst will accomplish them, and because of this not all are directly implementable on an automated playbook, as it is required for this project.

3.4.1.3 Runbook

This particular runbook should automate various different actions in the procedures previously mentioned in the section 3.1.1. While this runbook will not match perfectly with the flow of the previously explored diagram, it automated different actions in different sections. This runbook will register the incident in the case management system, investigate it and with this information classify it, as well as apply some counter measures.

In the SOC of the IPLeia there is designated mailbox to which are sent emails of which there are suspicions of phishing. The objective of the playbook to be developed is to automatically analyse these emails to try to find out if they really are phishing emails and enhance the case related to them with additional information. The playbook built on shuffle can be seen in figure 34.

As an additional source of information on building this playbook, the article on (DFLabs, 2017) was also used as a reference.

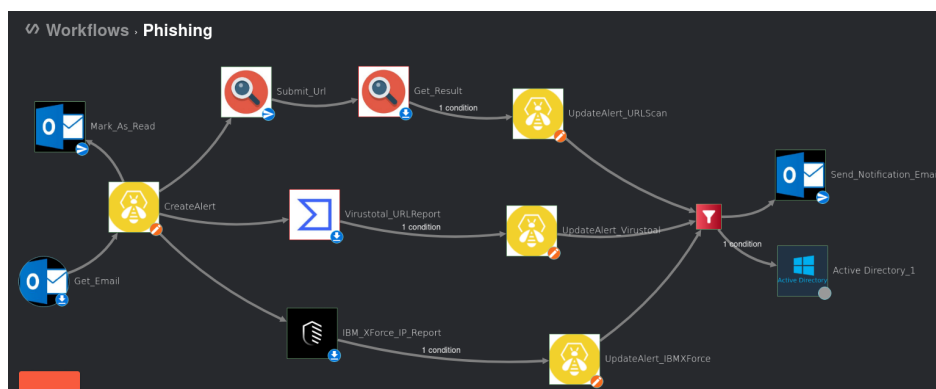


Figure 34: Phishing Playbook

The first step when dealing with a security incident is the identification of the threat. In this phase the objective is to gather as much information as possible about the phishing email. In this case it is important to look for the email address

of the sender, the subject of the email, it's content and attachments and also the recipient of the email.

In the playbook developed, the first action carried out is obtaining one unread email from the phishing mailbox. This is done through an application designed to integrate with the outlook webmail. For this, it necessary to configure the connection to the mailbox inside the app by supplying all the required authentication information and the connection properties. In this particular situation it's intended to read only one message at a time and process it, so it is also specified that only one unread message should be retrieved from the inbox folder.

When the application is ran, the necessary information mentioned before, is all available in the data returned by this app in the structure shown on figure 35.

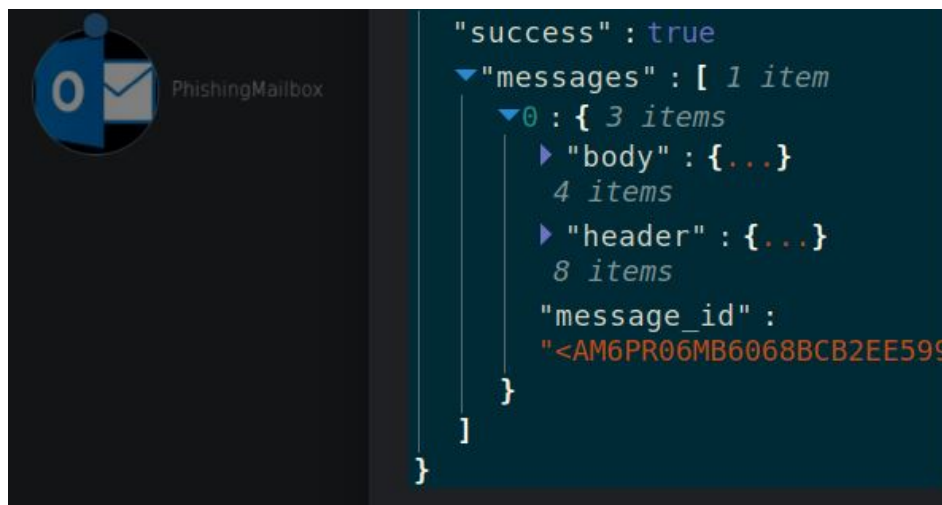


Figure 35: Email message shown through mailbox application in Shuffle

All of this information regarding the email in analysis is now available to be used in subsequent actions in the playbook.

After the message has been retrieved it is marked has read in the mailbox, to avoid it being processed multiple times.

Next, an alert will be created in TheHive. This will be a dedicated alert to this phishing incident and will gather all the information generated in the process of dealing with this incident. As soon as an email arrives in the phishing mailbox, it is important to open an alert TheHive to generate information on the incident, even if the suspicion turns out to be false it is relevant to store this information.

In order to create an alert, it's used another application to communicate with TheHive. To be able to interact with the [API](#) of TheHive, it is first necessary to generate an APIKey that serves as the authentication mechanism. Once this key is

obtained the connection can be configured by specifying the address of the TheHive and this key. Additionally it is necessary to configure the application to execute the intended action. In this case, it must create an alert, with the type of "Phishing Suspicion", source "Phishing Mailbox" and with a reference equal to the subject of the phishing email (figure 36).



Figure 36: Create and alert in TheHive

After this action, a new alert is present in TheHive alert list. In figure 37 it is possible to see this new entry, with the values that were specified in the Shuffle runbook.

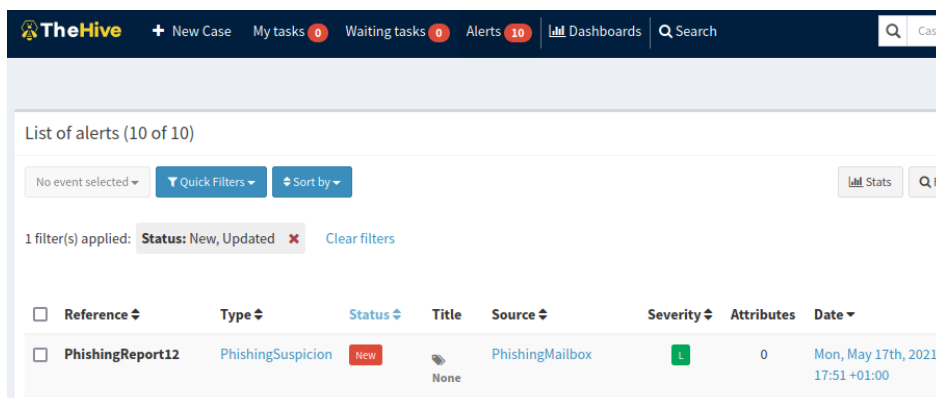


Figure 37: Phishing alert created on TheHive by Shuffle workflow

After the incident has been logged in TheHive, the analysis of the email begins. In order to find out more information about the phishing email two external services are used VirusTotal and URLScan.

The main objective at this phase is to try to find out if the email being processed really is a threat or not. To achieve this, the information of the email will be submitted to a service capable of analyzing it and determine if it is malicious or not.

Many phishing emails achieve their goal either by deceiving the user into accessing dangerous websites through [URL](#) present in the email. As such, analysing these with appropriate tools/services can provide a great insight in terms of determining if the email is malicious or not.

To analyse [URL](#) and domains, there are a multitude of services on the internet where this [URL](#) can be submitted and that return a response informing if it has been associated with anything malicious. Rather than integrating with a huge number of this it is preferably, both in terms of reducing the amount of time required to develop the playbook and in terms of the playbook efficiency, to use services that already gather information from multiple sources. Also, in order to use this services, it is required to have an integration with them on Shuffle, meaning it is convenient to use services which Shuffle already has integration's built. If it doesn't exists, it must be created from scratch, which for multiple services, can be time consuming. After the analysing the article on (How S., [2021](#)), two services were identified.

URLScan is service that performs scans on websites. By submitting an URL to this service, an automated process will browse to the website, executing actions like a regular user and records the activity that is generated by this. This process analysis information like IP addresses contacted, resources such as javascript and css and if anything is identified as potentially dangerous, it will be marked as so in the scan results (Gilger, [2021](#)).

VirusTotal is another service capable of analysing not only webpages and domains but files as well. Through virustotal api it is possible to upload files or submit urls that will be analysed and cross checked against multiple blacklisting services (VirusTotal, [2021b](#)).

Before integrating these services in the runbook, to better understand how they work and to test their performance, some tests were executed. It was required to analyse how their [API](#) are built, in order to understand how requests are made, and to afterwards analyse the structure of the response.

Also, this is an important step since shuffle, at the moment, still has a very basic (and incomplete in some regards) handling of these type of integration's, which can make any debug process very hard to execute. For example, if something is wrong with one of these integration's, like a misspelled [API](#) key, shuffle only returns a

generic connection error. In this case, a more complete response would include more information about the error, like the HTTP status and other relevant info, which is extremely helpful in identifying the issue.

Both virustotal and urlscan, require an user account in order to use their services. They also both use an apikey as the method of authentication on their [API](#), although, they handle the requests in slightly different ways. In the case of urlscan, the apikey is specified as an header on the request and it is necessary to execute two requests. In the first request the information to be analysed is submitted and then in the second the results are queried through and [UUID](#) returned from the first request. On virustotal, the apikey is specified as a query parameter, and the response from this request already has the result of the scan.

The response formats are also different between the two. Urlscan provides a very extensive response with many properties about it's many scans, and in the end a summary with its verdicts about the scans (figure [38](#)).

For the purpose of this runbook, this verdicts will be the only information used to classify the threat, more specifically the malicious flag on the "overall" property.

Virustotal returns a list of the assets it used to analyse the url, with their respective classifications (malicious or not). There is also a summary field which indicates the number of positives that were returned in all the scans. This will be the field which will be used to classify the url in the scope of the runbook (figure [39](#)).

To carry out these tests, it was also necessary to find real phishing emails to use as an example of a threat that these services should classify as malign. To obtain examples of phishing emails, samples were downloaded from malware-traffic-analysis.net.

From the tests that were carried out, all the samples were classified as malicious by at least one of these services, which is a good indicator on the reliability of these services. Also, it was possible to identify that some samples were classified as malicious by one service and clean by the other. This means that using both services together helps in achieving a better decision on the threat under examination.

In the developed playbook both of these services will be used to analyse eventual [URL](#) contained in the email. Since they use different methods of analysing the provided resource, using both means gathering more information about the threat.

In the playbook implementation, after the creation of the alert on TheHive, these services are called. In the case of UrlScan, as metioned earlier two interactions are

```

"verdicts": {
  "overall": {
    "score": 0,
    "categories": [],
    "brands": [],
    "tags": [],
    "malicious": false,
    "hasVerdicts": 0
  },
  "urlscan": {
    "score": 0,
    "categories": [],
    "brands": [],
    "tags": [],
    "detectionDetails": [],
    "malicious": false
  },
  "engines": {
    "score": 0,
    "malicious": [],
    "benign": [],
    "maliciousTotal": 0,
    "benignTotal": 0,
    "verdicts": [],
    "enginesTotal": 0
  },
  "community": {
    "score": 0,
    "votes": [],
    "votesTotal": 0,
    "votesMalicious": 0,
    "votesBenign": 0,
    "tags": [],
    "categories": []
  }
}

```

Figure 38: "Verdicts" included in the urlscan response

required. This is achieved by using two applications in the playbook, first one using an action called "Submit URL", and then a second one using a different action called "Get Result" (figure 40). The first action returns an **UUID** which is then used by the second as a parameter, to get the result on that submission. The get result actions will return a response with the structure previously illustrated in figure 38.

As for the integration with Virustotal, only one action is needed (figure 41). This single action will make a request with the provided url as a parameter and its response will contain the results of the analysis.

Another relevant characteristic of the email that can help in identifying it as phishing or not, is the domain of the sender of the email. Certain services use databases to store information about different domains, relative to many factors


```

Example response
{
  'response_code': 1,
  'verbose_msg': 'Scan finished, scan information embedded in this object',
  'scan_id': '1db0ad7dbcec0676710ea0eaacd35d5e471d3e11944d53bcbd31f0cbd11bce31-1390467782',
  'permalink': 'https://www.virustotal.com/url/___urlsha256___/analysis/1390467782/',
  'url': 'http://www.virustotal.com/',
  'scan_date': '2014-01-23 09:03:02',
  'filescan_id': null,
  'positives': 0,
  'total': 51,
  'scans': {
    'CLEAN MX': {
      'detected': false,
      'result': 'clean site'
    },
    'MalwarePatrol': {
      'detected': false,
      'result': 'clean site'
    }
  }
}

```

Figure 39: Virustotal response (VirusTotal, 2021a)

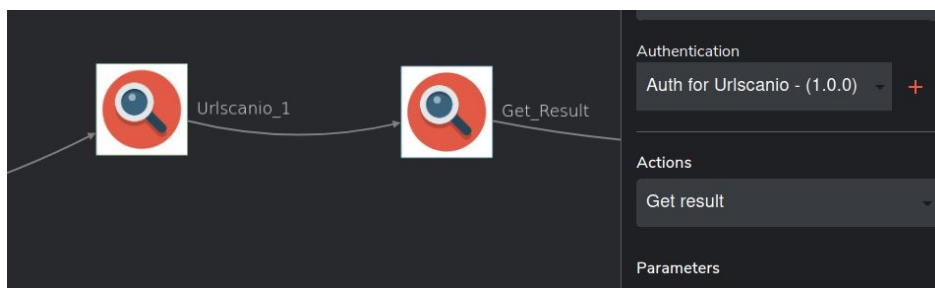


Figure 40: UrlScan application configuration

regarding them. The IBM XForce Exchange is one of such services. This service is able to classify a domain and associate it with a normalized value that is produced from processing the threat intelligence information available, known as risk score. For example, if an IP is identified as sending a high volume of spam frequently, it will have a high risk score. If the IP then becomes less active in its spam output, its score will start to decrease over time (IBMXForce, 2021).

Similarly to the previous two services, IBM XForce also exposes an [API](#) which can be used to access its services programmatically, although the authentication strategy is different, which instead of an API key, uses the basic authentication method with a username and password. Inside the application, the action to retrieve the IP reputation is specified, which will have included in its response the previously mentioned "risk score", upon which further actions will be performed (figure 42).

For all these integrations, their authentication is configured through shuffle administration authentication management (figure 43). Here different profiles are

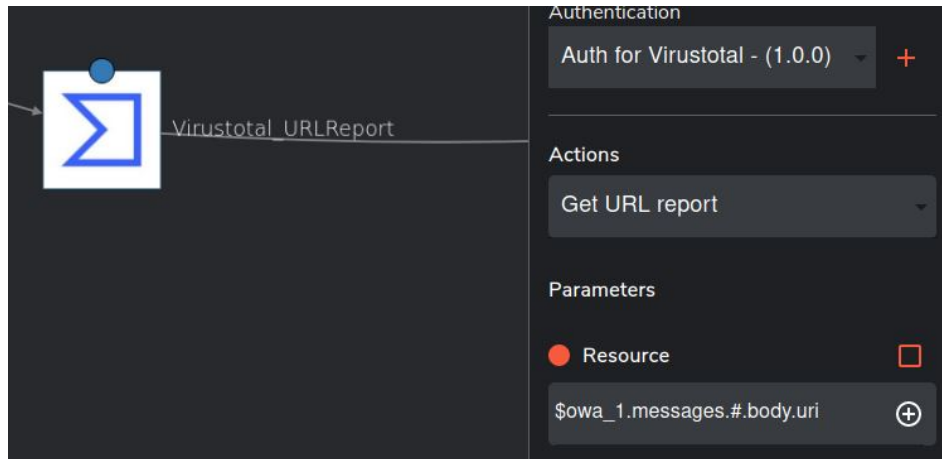


Figure 41: Virustotal application configuration

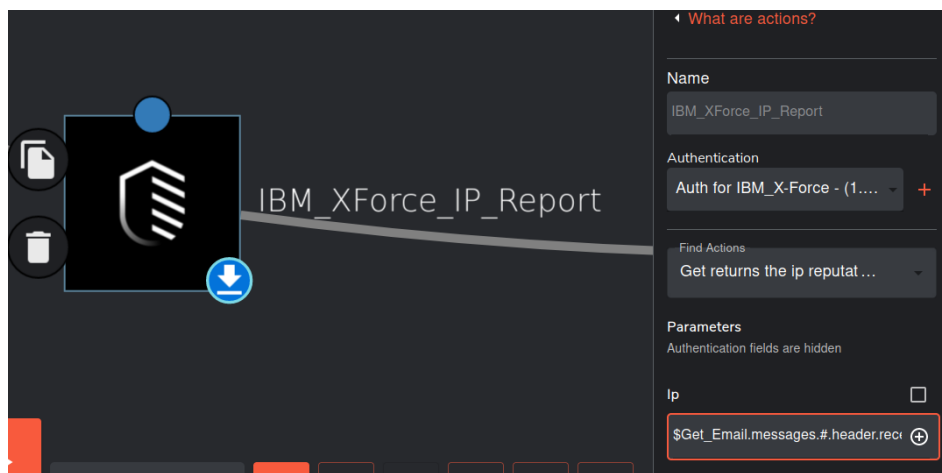


Figure 42: IBM XForce application configuration

created for each integration, with all the specific credentials and configurations necessary, which will then be used on the respective applications on workflows.

After all of these services are executed, the previously created alert will be updated if relevant information is found. To do this conditions are used in the connection between nodes. This conditions will analyse the responses from both services, and update the alert if the criteria specified is matched (figure 44).

Different conditions are required for the different services as they differ in the response format. For Virustotal, the condition is checking if the "positives" counter is larger then 0, for urlscan the malicious flag is checked and for IBMXForce the risk parameter is checked to evaluate if it is larger then 1.

Finally, two last actions are executed. If the playbook has reached this point (if any of the services indicated a threat), an email is sent to warn users that there is a phishing email circulating and advising them on how they should proceed.

Icon	Label	App Name	Ready	Workflows	Actions	Fields	Actions
	Auth for Immuniweb	Immuniweb	Yes	0	1	username_basic, password_basic	
	Auth for Urlscanio	Urlscanio	Yes	0	2	apikey	
	Auth for owa	owa	Yes	0	1	username, password, server, build, account, verifyssl	
	Auth for Virustotal	Virustotal	Yes	0	1	apikey	

Figure 43: Shuffle authentication section

Virustotal_URLReport 1 condition

Condition

source destination

- \$Virustotal_1.positives DOES NOT EQUAL 0

Autocomplete Autocomplete

Conditions can't be used for loops [#] [Learn more](#) **SUBMIT**

Figure 44: Virustotal workflow condition

Additionally, another verification is performed to find out if the email address that sent this message is from the [IPLeiria](#).

One possible scenario that eventual attacker may take advantage of, is using a compromised institutional [IPLeiria](#) account to spread malicious emails. To tackle this scenario a final check is made and if the email of the sender of the message is in fact an account from the [IPLeiria](#) domain, this account is blocked using the active directory services.

To block the user account in active directory, it is necessary to configure the integration with its server (figure 45). For this, first a connection with the server needs to be established by configuring the server information (IP Address, port and domain). Additionally it is necessary to have an account with administrator rights over the group of users that it is intended to act upon. With this two sets of information, the integration is able to authenticate on the active directory server and execute the necessary actions.

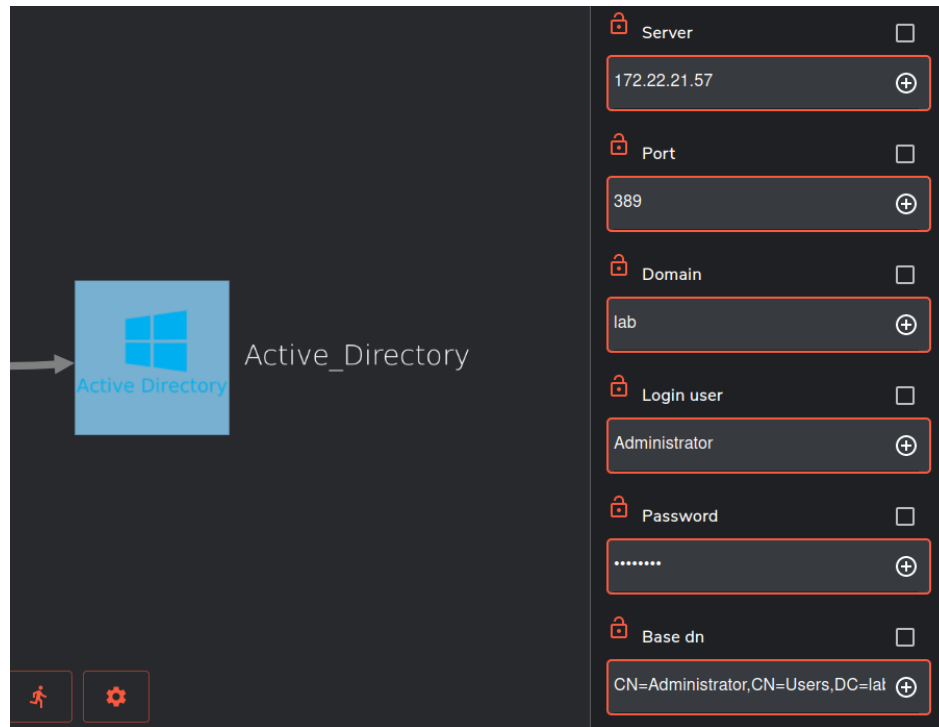


Figure 45: Active Directory Integration in Shuffle

In this case it used the action of "Deactivate Account", which will block the account from taking any further actions.

To be able to test this integration, a remote [VM](#) on the [IPLeiria](#) network with an instance of active directory configured was provided, which could be access via a [Virtual Private Network \(VPN\)](#).

3.4.2 Handling Data Breaches

Data breaches are another threat that can have a negative impact in the [IPLeiria](#) infrastructure. For this reason the [SOC](#) as taken precautions to mitigate any possible attacks stemming from this type of incidents.

3.4.2.1 Data Breach

A data breach occurs when information held by an organization is stolen or accessed without authorization. Once the information is leaked from the organization databases it can be used in multiple different ways to try to deceive the involved entities ([Şahin, 2021](#)).

One common example of this exploitation, is when attackers use some information when creating phishing messages (such as emails and texts) to make them look legitimate. By incorporating some real details about a user, taken from the data breach, in a phishing email, the user may be more easily lead to think that the email is in fact genuine, since it presents information that is supposed to be private. This greatly increases the chance of a phishing email being successful. Another possible case is when sensible information like passwords are leaked without being encrypted (hashed more commonly). This gives possible attackers direct access to the user's account.

3.4.2.2 *Data breach Incident Response Guidelines*

Data breaches are an inevitable part of most [IT](#) systems, as some of the cases that lead to breaches are completely outside of the organization control, like an employee o creates an account with the same credentials as the ones he uses inside the company on an external website, where a breach occurs. However, some of the negative impacts cause by these breaches, can be reduced or nullified, by developing a solid playbook that charts a course to recovery. The playbook by the Incident Response Consortium (Consortium, 2021), offers a general course of action when dealing with a data breach.

A data breach incident starts with the detection of said breach. There are a number of different signs that can be used to alert for a possible data breach, like large data dumps of databases, network shares, or other computer systems, large number of emails sent by single user, reports of removable devices used to copy data, identification of proprietary information outside the organization, among others. Procedures should be put in place to monitor for these types of activity and issue alerts accordingly.

If a data breach is detected, then first it must be analysed to understand its impact. Here is important to understand a series of different factors, many of them may depend on the organization. Some general factors to take into consideration include understanding if the breach violates any compliance regulation, if costumers or other business partner are affected by breach, if there is external knowledge of the breach or if the stolen data can be damaging for the brand of the organization.

The next step is to identify what systems have been compromised (servers, desktops, mobile, etc) and identify any user credentials that may also be compromised, which may grant access to these systems. Additionally, the source of the data leak must be identified, by analysing the leaked data and matching it to the organization

resources to find out where it may have come from. This can be done in parallel with the analysis of the logs of these systems to confirm any unusual activity.

Once the previous information has been gathered, the eradication phase can take place, where the compromised systems are repaired. Here any malware that may have infiltrated the systems is removed, compromised credentials revoked and analyses to ensure that now further systems are compromised taken. Moreover, there should also be set in motion the necessary meetings with the affected stakeholders in this incident, internal and external to the organization.

In the last step, all the organization systems are brought back to normal operation, by wiping and base lining systems, updating any system that may have represent a vulnerability, and other similar actions specific to the the organization systems, that help insure their secure operation.

3.4.2.3 *Runbook*

The runbook to be developed for data breaches will mainly automated the case management actions that the analyst needs to take, as well the counter measure that is taken in these situations.

Many of the operations when dealing with data breaches, as described above, cannot be easily automated, and for the context of the [SOC](#) of the [IPLeiria](#), the main use case for automation is associated with dealing with leaked credentials, where the domain of the [IPLeiria](#) is concerned.

In order to deal with data breaches, it is essential find a way of knowing when something of relevance to the [IPLeiria](#) happens. One way to get this type of information, is to use external services that gather information on this type of incidents.

To handle this issue specifically, the [SOC](#) of [IPLeiria](#) has a subscription to Immuniweb. Immuniweb is a company that provides automated web security services that work through their machine learning and artificial intelligence technologies. One of the services they provide is the issuing of alerts when there is a data leak. After an initial setup is performed, to specify which domains should be searched for, anytime there is a data leak involving some sort of asset from the organization, an incident on their platform is generated. By analysing this incident further actions can be taken to mitigate any following attack.

The goal of the following runbook, is to whenever a new alert is issued from Immuniweb, automatically take preventive actions to mitigate any attack based on that incident (figure 46).

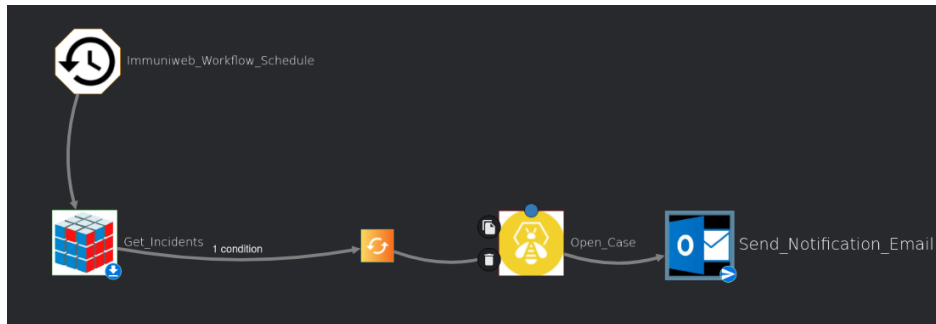


Figure 46: Databreach runbook

Immuniweb provides an [API](#) where all their services can be accessed. These include information on the discovered assets from penetration tests, for each type of application or services that the organization might have (web, mobile, cloud, network). There is also an incident report section, where discovered incidents that involve the organization are logged. For this runbook, what has the most interest is this second functionality, which will allow to set in motion a set of actions when an incident occurs (figure 47).

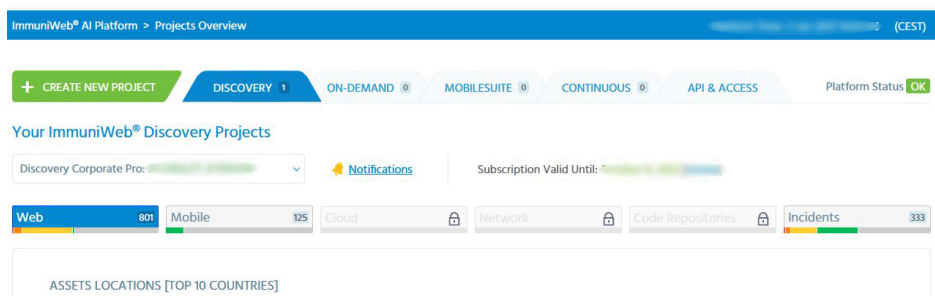


Figure 47: Immuniweb dashboard

The first element of this runbook is a schedule. A schedule acts as a trigger for the runbook. In order to know if a new incident has happened it is required to make a request to the Immuniweb [API](#). Taking this into account, it is necessary to execute this runbook continuously over time to detect new incidents, hence the use of the schedule. The schedule is configured to run every thirty seconds, which gives a reasonably quick response time to any new incident that is created, while keeping the usage of resources required for the execution low (figure 48).

When the schedule is triggered, the first action executed will be the request to Immuniweb API to retrieve the list of incidents. This request needs to be made

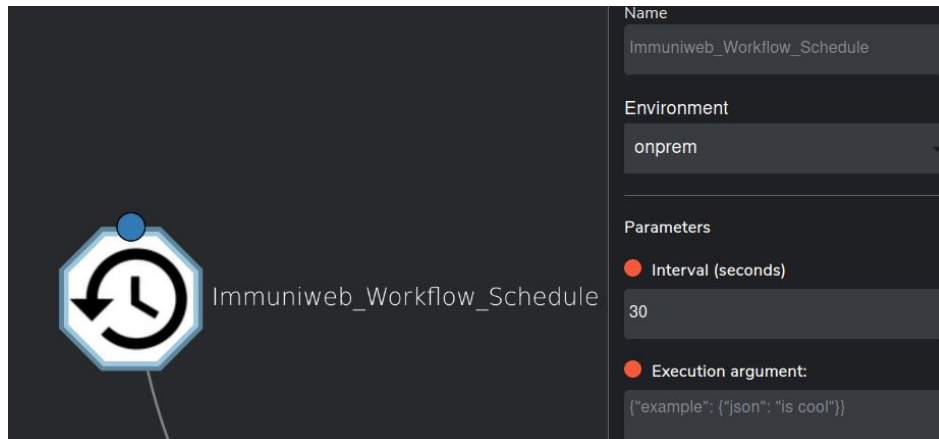


Figure 48: Schedule configuration

(similarly to the services of the previous runbook) through a specific application that integrates with Immuniweb. However, unlike with the previous used services, Shuffle does not have an application for Immuniweb built, which means it will need to be developed.

To build this integration in Shuffle the first step is to analyse how the Immuniweb [API](#) works. By consulting the documentation on their website, it is possible to understand they use an the HTTP basic authentication scheme to control access to their [API](#). Furthermore, to create credentials for this mechanism, they need to be generated from an authenticated user dashboard. These credentials will consist of an "apikey ID" and an "apikey secret", which will act as a username and password respectively. The [URL](#) of the api endpoint where the requests must be sent is also specified (<https://portal.immuniweb.com/client/project/discovery/exportdomains/>) together with the HTTP method to use (GET) and also the different parameters that can be used to customize the response to the users needs (figure 49).

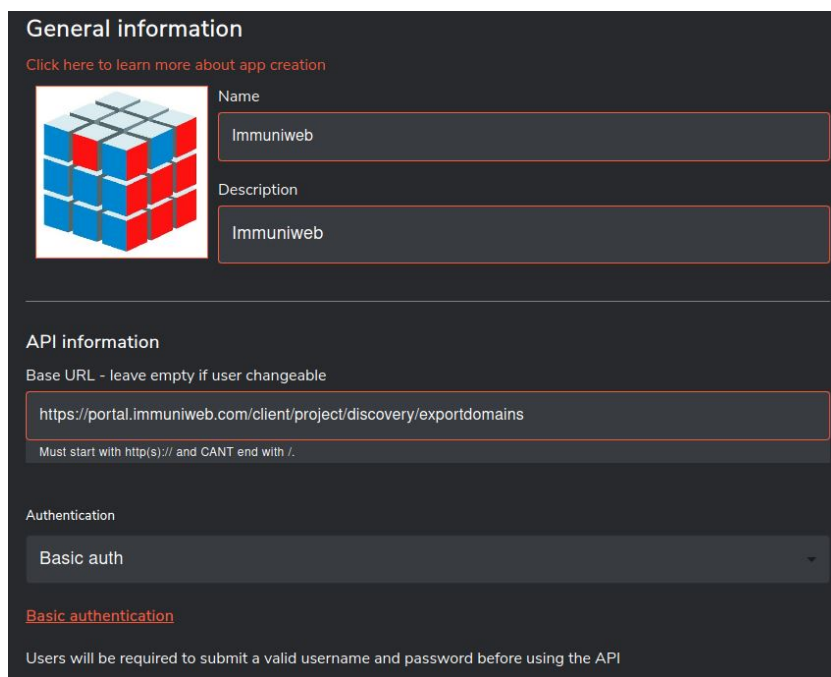
HTTP GET Parameters		
Name	Type	Description
discovery_id	integer (required)	Project ID
discovery_tab_type	string (optional)	Data tab: <ul style="list-style-type: none"> • webapps • mobileapps • cloud • network • repositories • incidents

Figure 49: Immuniweb Api Parameters

Immuniweb uses "Discoveries" as a way to create different profiles with different configurations to be used by the users. This is one of the options that must be

specified through parameters on the request. The different discoveries are presented on the user dashboard and have an ID associated with them, which will be the ID used in the request, specified in the "discovery_id" parameter. Finally it must be also specified the "discovery_tab_type", which is used to indicate what kind of information is intended to be retrieved.

With all this information gathered, it is now possible to create the application to use in the Shuffle workflow. To build this application on Shuffle, it is used Shuffle's own application builder functionality (figure 50).



The screenshot shows the 'General information' section of the application builder. It includes a link to learn more about app creation, a 3D cube icon, and input fields for 'Name' (Immuniweb) and 'Description' (Immuniweb). Below this is the 'API information' section, which includes a 'Base URL' field containing 'https://portal.immuniweb.com/client/project/discovery/exportdomains' and a note that it must start with 'http(s)://' and cannot end with a period. The 'Authentication' section shows a dropdown menu set to 'Basic auth' and a link to 'Basic authentication' with a note that users will be required to submit a valid username and password before using the API.

Figure 50: Application builder Shuffle, Immuniweb basic configuration

In the first section some basic information must be specified such as the name, the base [URL](#) which can later be concatenated with additional routes to access different endpoints for different actions, the authentication method used by the service and the image used to easily identify the application in a workflow.

Each application can then have a list of actions which will execute different requests. These actions have additional information like the specific [URL](#) path for them, HTTP method and additional parameters that may or may not be specified for it. Here is where the different parameters identified above (figure 49) are specified to be used in the application.

After this the application can be saved built, which will then make it available for use in a workflow.

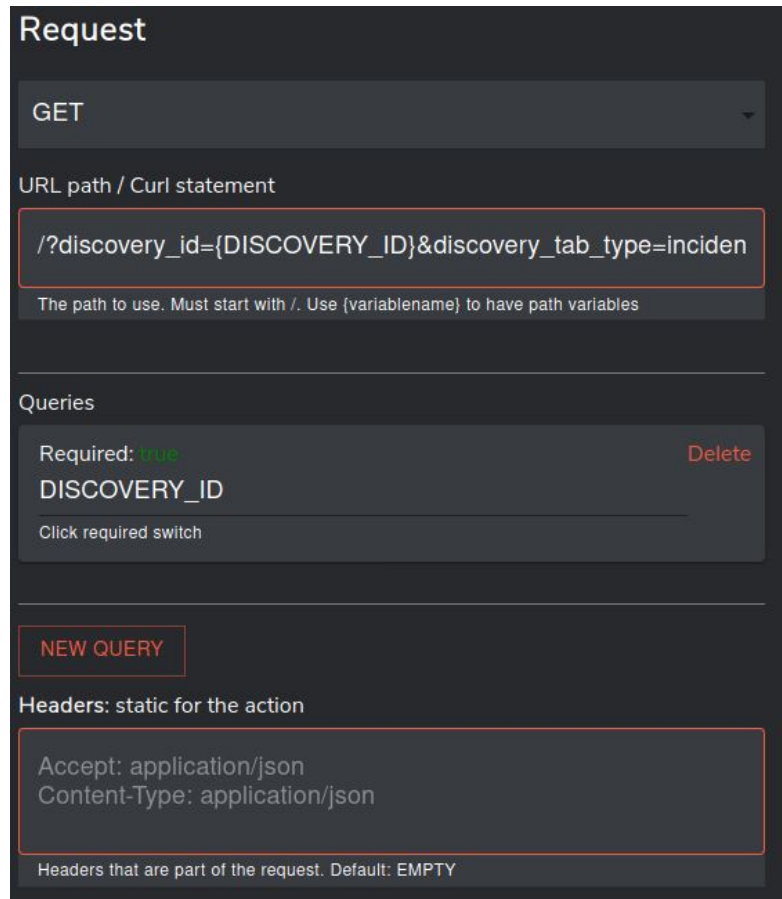


Figure 51: Application builder Shuffle, Immuniweb action configuration

In the workflow the immuniweb application will be using the action previously created (GetIncidents) and will require the ID of the discovery to used to be defined (figure 52). The response from this service will contain an array of the incidents logged in the platform.

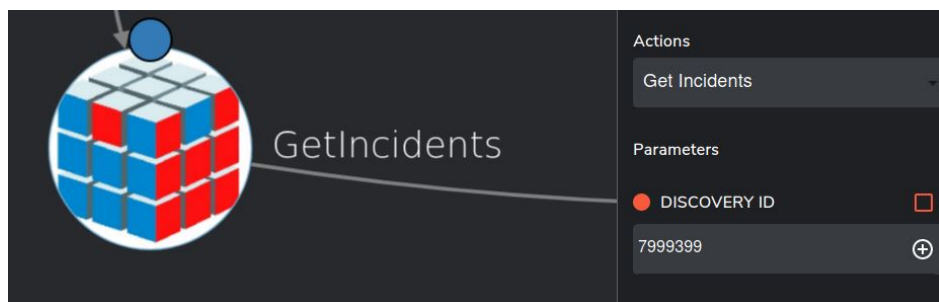


Figure 52: Immuniweb Configuration

Since it only makes sense to handle each incident retrieved from the platform once, and when making the request to Immuniweb the response always contains all incidents reported, it is necessary to implement some logic to check only for new

incidents. In Shuffle this was achieved by using workflow variables. Shuffle offers two types of control variables to use inside of workflows, workflow variables and execution variables.

Workflow variables are variables that can be set before the start of the execution of a workflow and define static reusable data. This means that after the workflow is finished, they will keep their value and can then be reused in subsequent executions. These type of variables can be used store information such as an [URL](#) or an [apikey](#) to be used inside the workflow.

Execution variables, are used inside a workflow in the same way as a workflow variable, although they can only be assigned values inside this same workflow. Their intended use is to store temporary data while the workflow is executing, that is not saved anywhere else, therefore losing its value from execution to execution.

To solve the problem with the response from the Immuniweb [API](#), what is necessary is to keep track between executions of how many incidents exist in the platform, so that in the next execution it can be analysed if there are any new occurrence and process these ones accordingly. For this purpose, a workflow variable was used (figure 53).

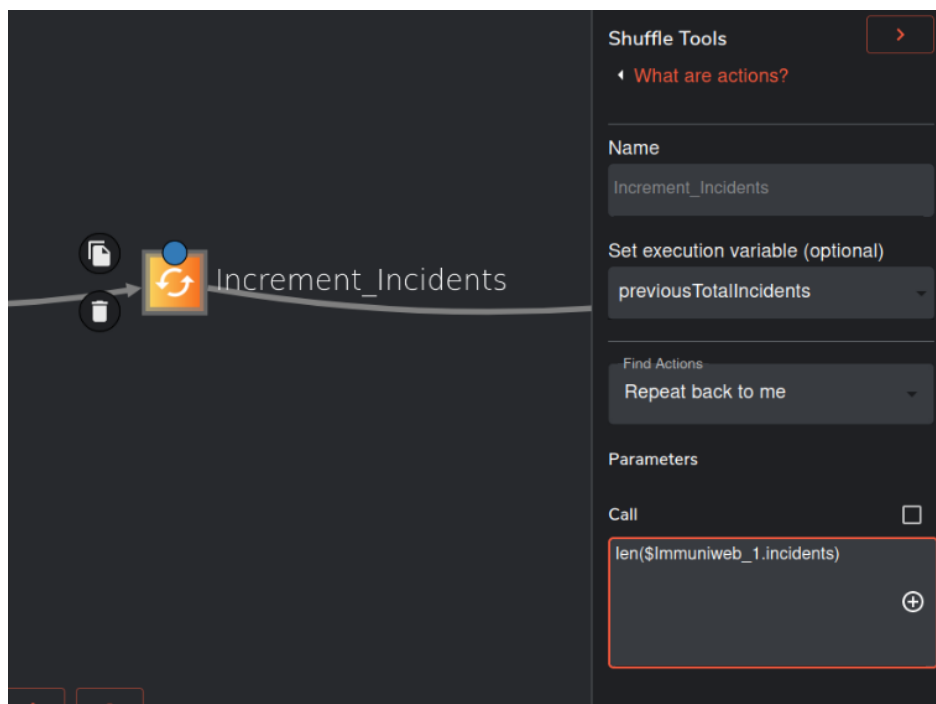


Figure 53: Shuffle Immuniweb Incidents Control

After retrieving the list of incidents, the total number of incidents reported is compared to the previous value recorded through a branch condition. If the value is

equal, then the runbook stops at this point, as the condition to continue to execute is not met. If the number is greater than the previous value, then first, the control variable is incremented, to register the new incident, next the rest of the runbook is executed.

Since multiple incidents may be reported at once, it is important to handle all of them. To ensure this the control variable is only incremented by one, and only this new incident will be processed by the rest of the workflow. On the next execution, the succeeding report will be handled. This is not the ideal way to handle these events, as there may be a relatively huge number of incidents reported at once. Preferably, there should be a loop in the runbook to handle all the incidents, as in this way all the new reported incidents would be handled in a single execution. However, Shuffle does not yet support this functionality, as it is a request feature that is still under development at the time of the construction of this runbook.

The next step in this runbook is to open a new case in TheHive. This step is similar to what was done previously in the phishing runbook to open an alert. Here a case is immediately created with a higher severity level (2 instead of 1), as it is a confirmed security issue. When the case is created, it is immediately enriched with the information that the response from Immuniweb contains. The case will be created with the title "Compromised Credentials (Immuniweb)" and with a description containing the email that has been compromised.

Lastly, it is sent an email to the affected user account, informing about the incident that has just occurred. Since there is no way to be certain that the same password is used in the [IPLeiria](#) account, and because that if a password reset is issued the user may end up locked out of his own account without receiving any information on what happened, as this process can only be carried out while inside the institution network, the only advisable action to take in this situation is notifying the user. This email aims to let the user know that his institutional email account address has been found in a data breach and that the associated password has also been exposed, advising that if the same password is used elsewhere it should be changed.

To send the email to affected account, a predefined email account for notifications is used, and the integration with the email server is configured in the same way as it was in the previous runbook. With this integration a new email based on a template is sent to the affected account.

3.4.3 *Summary*

For the particular case of the SOC of the IPLeia, it was chosen a solution that best encompasses its necessities with its limitations, based on the research that was previously carried out. This chosen solution combines two different software, each with its own strengths, that are able to integrate with each other in order to form a solution that can accomplish every base requirement of a SOAR software.

These solutions were then used to create two different runbooks, which automate actions that are currently executed on a daily basis in the SOC. These runbooks do not handle an incident from beginning to end completely, as that is not easily achievable without the presence of a human at some points, although this was not the objective. The goal of the two playbooks was to help the analysts in scenarios where some degree of automation eases the daily tasks, that are currently required to be carried out manually. Referring to the previously mentioned IPLeia SOC incident life cycle procedure, various steps throughout the process, in incidents of the type that the runbooks are designed to handle, are automated, mainly in terms of case management, but also in terms of contention and eradication. Furthermore, these playbooks provide valuable knowledge to create other playbooks for future use cases with these tools.

3.5 TESTING

As this is a practical project, which results in the creation of a deliverable, it is important to devise tests that ensure that the specified requirements are met. The purpose of this section is to develop different test scenarios that are designed to test the two runbooks created previously.

3.5.1 *Phishing Runbook*

To test the functionalities of the phishing runbook it is necessary to obtain an email that has been identified as a phishing email, and verify that all the actions defined in the runbook are executed as expected.

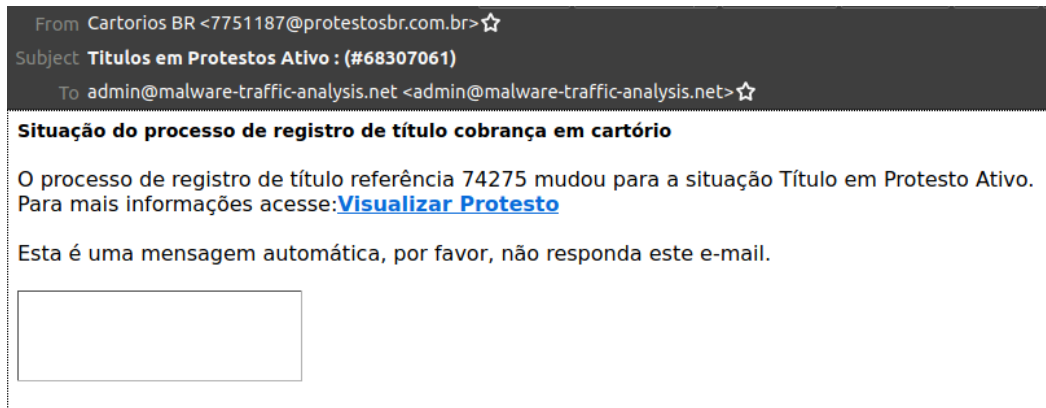


Figure 54: Phishing Email Sample

The sample of a malicious email was obtained from the website malware-traffic-analysis¹. The sample email (figure 54), is a phishing email that contains a link with an [URL](#) to a phishing page. This [URL](#) is what can be used to automatically detect if the email under analysis is a phishing email. If it is associated with a malicious page, the services used in the runbook should report it as such.

After obtaining this sample phishing email it is necessary to upload it to an existent mailbox to be used in the test scenario. For this purpose a test email account was created and configured in the respective Shuffle application.

After this configurations are into place, and the TheHive instance is also running it is possible to execute the runbook (figure 55). After the execution finishes it is possible to quickly identify the executed applications by observing the connections that have changed their color to green. Through this, it is possible verify that apparently, Virustotal has identified the contents of the email to be malicious, and accordingly with the condition configured in the connection between the Virustotal application and the respective TheHive application, execution flowed to execute the UpdateAlert TheHive node.

The first step of the workflow is to obtain the email from the configured mailbox. To check that this was done correctly, the output of the email application can be inspected, and it can be verified that the subject and content matches with the email message.

After this an alert should be opened in TheHive. This is an alert that is created with some specified parameters from the Shuffle application, the reference, type, source and severity. In the figure 56 the generated alert can be seen with the fields correctly filled with the information that was specified in Shuffle.

¹ Malware-traffic-analysis is a blog that focuses on network traffic related to malware infections, and contains multiple samples of a diverse type of malware, <https://www.malware-traffic-analysis.net>

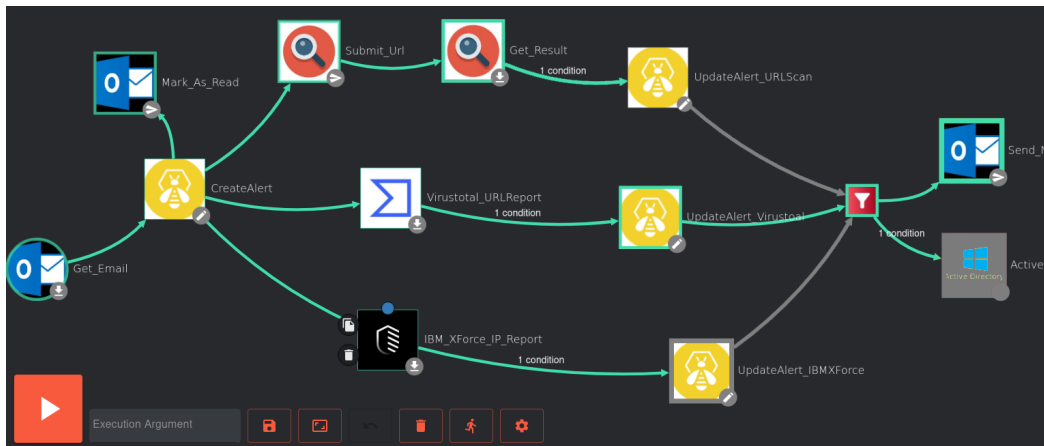


Figure 55: Execution of the Phishing Workflow

<input type="checkbox"/>	Reference ↕	Type ↕	Status ↕	Title	Source ↕	Severity ↕	Attributes
<input type="checkbox"/>	Titulos em Prot estos Ativo : (# 68307061)	PhishingSuspicion	New	Non e	PhishingMailbox	L	0

Figure 56: Creation of Alert on TheHive

Immediately after this, the message should be marked as read, so that it is not processed again in the future. By inspecting the mailbox this action can be confirmed.

In the [URL](#) analysis, only Virustotal appeared to have flagged the [URL](#) that was contained in the email as malicious. To further verify that each application executed correctly in the runbook, it is necessary to replicate these same actions, via another method, and compare the results. The most reliable way of doing this verification is to use each service [API](#) directly, as was previously done, in the last chapter, to set up these integrations. The results obtained by using each service [API](#) to analyse the [URL](#) the email contains, can be seen in [figure 57](#).

As can be seen in the result, only virustotal reports the submitted [URL](#) as malicious, which matches what the applications returned when being executed inside the runbook. Furthermore, the fact that only the next node under the virustotal application executed, demonstrated that the condition configured in the connection, is also working as intended.

Relating to the domain analysis, provided by the service from IBMXForce, from the execution of the playbook we can see that the service did not mark the domain as suspicious. This is confirmed by repeating the process done for the two other

VirusTotal

```
06170D/6C8T98GBRTZ/52738686", "response_code": 1, "scan_date": "2021-06-17 08:13:03",
"permalink": "https://www.virustotal.com/gui/url/e56fa0e091813912707ea4a8722a7f3373d47216bbd9b4e28290dfe507a442bb/detection/u-e56fa0e091813912707ea4a8722a7f3373d47216bbd9b4e28290dfe507a442bb-1623917583", "verbose_msg": "Scan finished, scan information embedded in this object", "filescan_id": null, "positives": 1, "total": 88, "scans":
```

URLScan

```
},
"verdicts": {
  "overall": {
    "score": 0,
    "categories": [],
    "brands": [],
    "tags": [],
    "malicious": false,
    "hasVerdicts": 0
  },
  "urlscan": {
```

Figure 57: Command Line Results from Virustotal and URLScan

services, of manually calling their [API](#), and verifying that in fact the score associated to the email domain is 1 (lowest possible score), which means the runbook behaved accordingly.

```
4.62.206.0/23"}], "cats": {"Dynamic IPs": 71},
"geo": {"country": "Portugal", "countrycode": "PT"}, "score": 1, "reason": "DNS heuristics", "reasonDescription": "Based on statistical DNS analysis.", "categoryDescriptions": {"Dynamic IPs": "This category contains
```

Figure 58: Command Line Results from IBMXForce

By analysing figure 55, we can see that only the virustotal path was executed, both the other paths aborted (correctly) on the condition evaluation, which mean that only the action to update the TheHive alert on this path should be executed. This alert update is the next action to take place. The update action has two jobs, one is two raise the severity of the alert from low to medium, and to add to the description of the alert, the information that virustotal has reported it as malicious. In figure 59, these two changes can be confirmed. The severity is now medium (M) and the description was correctly updated. Furthermore, we can validate that the only text added to the description is related to virustotal, no other text from the two other paths has been added, confirming that the runbook did not execute the other paths, as expected.

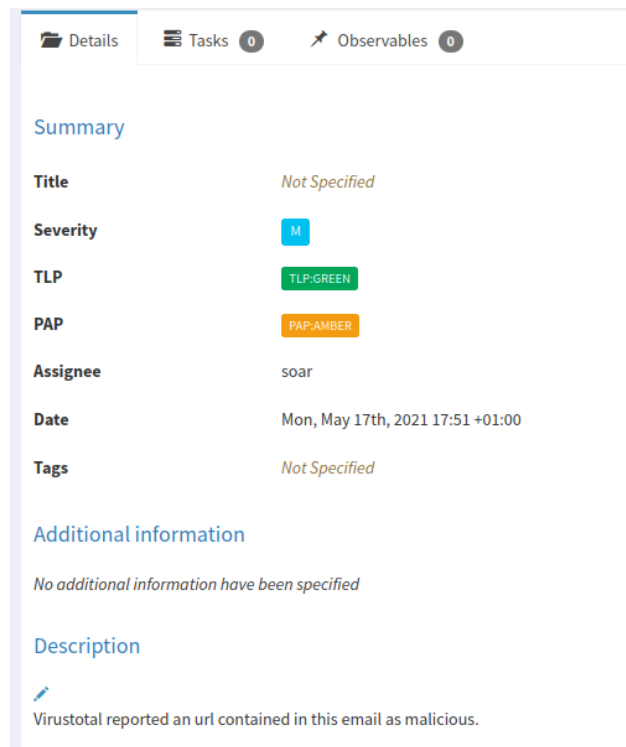


Figure 59: Updated The Hive Alert

The last action taken is to send the warning email, since the playbook has in fact reached this point. This can be easily checked by verifying the mailbox of an email account that was in the destination of the phishing email.

Since in this execution of the runbook the sender of the email wasn't an [IPLeiria](#) account, the flow in which the integration with active directory is not executed.

To test this scenario another test was ran. This test used a new email message, which had as sender email address a fake address with the domain of [IPLeiria](#) and that corresponded to an account created in the active directory instance provided to run these tests. This email also included the same [URL](#) that was present in the last test, so that it is considered a malicious email.

After executing the runbook in this second scenario the active directory integration is executed and by logging into the active directory admin dashboard, it is possible to check that the account corresponding account has been flagged has locked/disabled.

3.5.2 Data Breach Runbook

The data breach runbook execution is dependent on the response that is given by the Immuniweb service. Unless this response has a new incident, compared to the

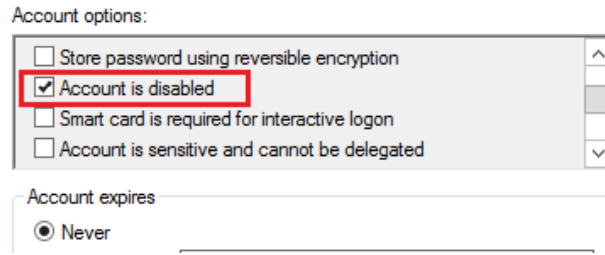


Figure 60: Active Directory User Management

previous, the runbook will not be executed. Many services similar to this, have special test endpoints or sandbox environments, that can be used by developers to build and test their integration's, which can be used to emulate the operations of the real endpoints, with options to manually generate and control the various events they deal with. Although, this is not the case with Immuniweb, as this type of endpoints are not provided, which makes the testing process more complex.

To solve this issue, the solution that was adopted was to create a simple [API](#) that returns a response with the same structure as the response from Immuniweb. With this [API](#) running in an environment that the Shuffle instance has access, the integration with Immuniweb was changed to instead make a request to the IP address of this local [API](#), allowing thus to customize the response.

The [API](#) was developed using Quarkus, a Java framework, which exposes one endpoint where Shuffle can interact with. By specifying a query parameter named "increment" as true, the response comes with an additional incident in its structure, this allows to test both of the conditions that can happen in the playbook.

After a baseline has been established (to set the base number of incidents), the execution of the runbook is displayed on figure 61. This execution shows how the actions after the check for new incidents is performed, are not executed, since no new incidents were found.

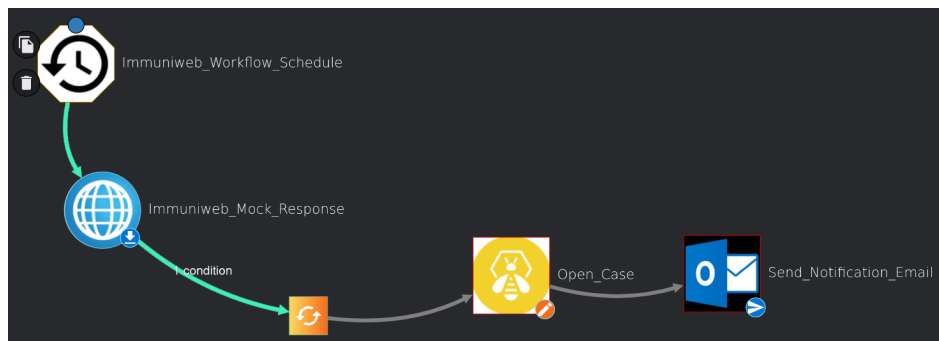


Figure 61: Data breach Runbook Execution Without New Incidents

After this first execution of the runbook, the number of incidents was incremented on the [API](#) and the runbook was executed a second time. This time, as can be seen in figure 62, the flow of execution continued after the evaluation of the number of incidents, opening a new case on TheHive and dispatching an email to the account mentioned in the incident report.

Another action that was also possible to test thanks to using this local [API](#) to "mock" the response from Immuniweb, was the sending of the notification email. The data returned by the [API](#) contains an email that can be used for testing, enabling the check of this mailbox to be performed.

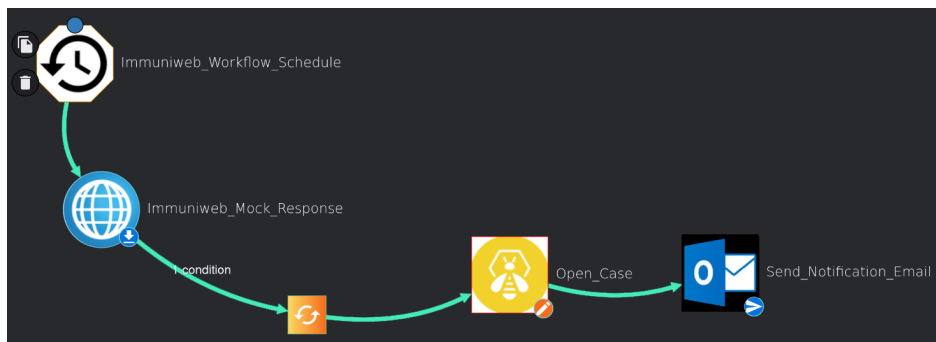


Figure 62: Data breach Runbook Execution With New Incidents

CONCLUSION

Nowadays, security of the **IT** systems of organizations is a big issue, as failures in this department may lead to serious financial, legal and reputation problems. However with the enormous amounts of cyberattacks occurring, it is also very hard to have a team of security analysts capable of handling them individually. The **SOC** of the **IPLeiria** is no exception to this problem, and to help with this issue this project implements a **SOAR** solution capable of reducing the workload of analysts.

4.1 CONCLUSIONS

After conducting some research in security incident response, and on **SOAR** solutions, and taking also into consideration the constraint imposed by the **SOC** of the **IPLeiria**, a decision was made on the solution to implement. The solution chosen makes use of an opensource software called Shuffle, that is able to provide many features in terms of security orchestration and automation, in conjunction with the already utilized in the **SOC** TheHive.

The Shuffle **SOAR** platform is a very recent opensource project, that is being developed since June 2020. Being such a recent platform, it still has many incomplete features, specially at the time this project started in September of 2020. Although, its quick progression on the implementation of new features and the active community behind is development make it a worthy choice, as it has the potential to evolve into a good platform to support the future needs of the **IPLeiria SOC**.

Nevertheless, it still is a very unstable software, that still has many bugs. This translated in many lost hours trying to work around its faulty features, and recurring to the development community to try to implement the requirements of this project. This used a huge amount of time to solve software related issues and in the development of some features that initially weren't available, which in part, hindered further development of the project.

Despite this issues, two runbooks that are relevant for the daily operation of the **IPLeiria SOC** were developed. This runbooks integrate with the tools that the **SOC**

already uses and are able to automate actions that are usually carried out by [SOC](#) analysts, related to phishing and data breach incidents. Through the use of this runbooks it is possible to automate actions that effectively save the analysts time and effort.

Beyond providing these runbooks, this project was able to gather much information, explore and put into practice how the two chosen solutions can be used to perform [SOAR](#) duties. This, together with the fact that the solution chosen is not bound by any limiting factor, gives a solid foundation to the [SOC](#) for further developments using these platforms.

4.2 FUTURE WORK

The main goal of this project was to create a proof of concept solution of a [SOAR](#) solution to be used in the [SOC](#) of the [IPLeiria](#). Taking into consideration the time constraints the project has, it was only targeted developing two runbooks, of the most common and meaningful cases for automation within the [SOC](#). This gives room to study new scenarios where automation might help within the [SOC](#), and develop new runbooks that tackle more of these procedures.

Since Shuffle is such a new tool and new developments and optimization are coming at an immense rate, it can also worth reviewing the already develop runbooks, to further optimize these. In the case of the phishing runbook, the fact that Shuffle is the no capable to deal with file attachments on emails at the moment, did not allow to perform an analysis on these, which is also an important step when dealing with phishing emails. Another limitation, now in the case of the data breach runbook, is the lack of loop functions in Shuffle. Because of this, a less effective solution was adopted, as explained in the previous chapter. Once features that can handle these situations become available, these runbooks can be significantly optimized with them.

BIBLIOGRAPHY

- Andrew Ramsdale, Stavros Shiaeles and Nicholas Kolokotronis (2020). «A Comparative Analysis of Cyber-Threat Intelligence Sources, Formats and Languages». In: *MDPI Electronics*.
- Chandni Islam Muhammad Ali Babar, Surya Nepal (2019). «A Multi-Vocal Review of Security Orchestration». In: *ACM Computing Surveys*.
- Christopher Johnson, Ping Wang (2018). «Cybersecurity Incident Handling: A case study of the equifax data breach». In: *Issues in Information Systems*.
- Cisco (2021). *What Is Phishing?* URL: <https://www.cisco.com/c/en/us/products/security/email-security/what-is-phishing.html>. (accessed: 06.04.2021).
- Consortium, Incident Response (2021). *Playbook - Data Theft*. URL: <https://www.incidentresponse.com/playbooks/data-theft>. (accessed: 10.05.2021).
- Cukier, Michel (2017). *Hackers Attack Every 39 Seconds*. URL: <https://eng.umd.edu/news/story/study-hackers-attack-every-39-seconds>. (accessed: 10.12.2020).
- Das, Ravi (2021). *The Phishing Response Playbook*. URL: <https://resources.infosecinstitute.com/topic/the-phishing-response-playbook/>. (accessed: 03.04.2021).
- DFLabs (2017). *The Phishing Response Playbook*. URL: <https://www.dflabs.com/resources/blog/r3-rapid-response-runbook-for-spear-phishing/>. (accessed: 08.05.2021).
- (2019). *The Difference Between Playbooks and Runbooks in Incident Response*. URL: <https://www.dflabs.com/resources/blog/the-difference-between-playbooks-and-runbooks-in-incident-response/>. (accessed: 05.01.2021).
- Dosal, Eric (2021). *8 Cybersecurity Threats to Watch Out for in 2020*. URL: <https://www.compuquip.com/blog/cybersecurity-threats-watch-out>. (accessed: 04.04.2021).
- EfiJ (2018). *Security Operations Center*. URL: <https://nextsecurity.co/security-operations-center/>. (accessed: 15.05.2021).
- ESG (2016). *Phantom and ESG Research Finds Companies Ignore Majority of Security Alerts*. URL: <https://www.businesswire.com/news/home/20160315005555/>

- [en/Phantom-ESG-Research-Finds-Companies-Ignore-Majority](#). (accessed: 10.12.2020).
- Exabeam (2020). *Incident Response Automation and Security Orchestration with SOAR*. URL: <https://www.exabeam.com/siem-guide/incident-response-and-automation/>. (accessed: 15.12.2020).
- Foolcdn (2020). *Overview: What is a runbook?* URL: <https://www.fool.com/the-blueprint/runbook/>. (accessed: 15.02.2021).
- Forcepoint (2021). *What is Incident Response*. URL: <https://www.forcepoint.com/cyber-edu/incident-response>. (accessed: 15.01.2021).
- Forsyth, Elliot (2018). *Dealing with Cyber Attacks—Steps You Need to Know*. URL: <https://www.nist.gov/blogs/manufacturing-innovation-blog/dealing-cyber-attacks-steps-you-need-know>. (accessed: 20.05.2021).
- Froehlich, Andrew (2021). *What is SIEM? What is SOAR? How are they different?* URL: <https://searchsecurity.techtarget.com/answer/SOAR-vs-SIEM-Whats-the-difference>. (accessed: 15.04.2021).
- Gartner (2020). *Gartner Glossary*. URL: <https://www.gartner.com/en/information-technology/glossary/security-orchestration-automation-response-soar>. (accessed: 10.10.2020).
- (2021). *What are Security Threat Intelligence Products and Services*. URL: <https://www.gartner.com/reviews/market/security-threat-intelligence-services>. (accessed: 25.04.2021).
- Gilger, Johannes (2021). *URLScan - About*. URL: <https://urlscan.io/about/>. (accessed: 03.04.2021).
- Groot, Juliana De (2020). *What is a Security Operations Center (SOC)?* URL: <https://digitalguardian.com/blog/what-security-operations-center-soc>. (accessed: 10.01.2021).
- Gurinaviciute, Juta (2021). *5 biggest cybersecurity threats*. URL: <https://www.securitymagazine.com/articles/94506-5-biggest-cybersecurity-threats>. (accessed: 04.04.2021).
- How S., hustlelead (2021). *URL Analysis: How to Determine Maliciousness*. URL: <https://hustlelead.medium.com/url-analysis-how-to-determine-maliciousness-f630b4e51b9e>. (accessed: 02.04.2021).
- IBMXForce (2021). *IBM X-Force Exchange API*. URL: <https://api.xforce.ibmcloud.com/doc/>. (accessed: 01.02.2021).
- IR, Flexible (2020). *Incident Response : Phases understanding them better*. URL: <https://playbooks.flexibleir.com/tag/sans-incident-response-framework/>. (accessed: 20.05.2021).

- ITPrice (2021). *PALO ALTO PRICE LIST 2021*. URL: <https://itprice.com/paloalto-price-list/xsoar.html>. (accessed: 17.02.2021).
- Kirtley, Ellyn (2020). *What is SIEM? What is SOAR? How are they different?* URL: <https://swimlane.com/blog/siem-soar>. (accessed: 01.10.2020).
- Kral, Patrick (2021). *Incident Handler's Handbook*.
- M. Vielberth F. Bohm, I. Fichtinger (2020). «Security Operations Center: A Systematic Study and Open Challenges». In: *IEEE Open Access Journal*.
- Mario Zgela, Ivan Penga (2019). «Security Information and Event Management – Capabilities, Challenges and Event Analysis in the Complex IT System». In: *Proceedings of the Central European Conference on Information and Intelligent Systems*.
- McGeehan, Ryan (2018). *Incident Response: Writing a Playbook*. URL: <https://magoo.medium.com/incident-response-writing-a-playbook-773e7920f171>. (accessed: 05.4.2021).
- Michael Benza, Dave Chatterjee (2020). «Calculated risk? A cybersecurityevaluation tool for SMEs». In: *Kelley School of Business, Indiana Universit*.
- Moran, John (2018). *How to leverage your existing SIEM solution with SOAR technology – a DFLabs and LogPoint use case*.
- Morgan, Steve (2019). *Global Cybersecurity Spending Predicted To Exceed 1 Trillion dollars From 2017-2021*. URL: <https://cybersecurityventures.com/cybersecurity-market-report/>. (accessed: 01.11.2020).
- Oedegaardstuen, Fredrik (2020). *Shuffle Introduction Medium*. URL: <https://medium.com/shuffle-automation/introducing-shuffle-an-open-source-soar-platform-part-1-58a529de7d12>. (accessed: 01.10.2020).
- Open, OASIS (2021). *Security Playbooks Version 1.0*.
- PaloAlto (2019). *What is a Threat Intelligence Platform*. URL: <https://www.paloaltonetworks.com/cyberpedia/what-is-a-threat-intelligence-platform>. (accessed: 10.05.2021).
- Şahin, Yağmur (2021). *Was My Data Stolen? Potential Data Breach Scenarios and Brief Map*. URL: <https://medium.com/databulls/was-my-data-stolen-potential-data-breach-scenarios-and-brief-map-e090174bb921>. (accessed: 10.05.2021).
- Scott, Andrew (2020). *SIEM vs SOAR, What's the Difference?* URL: <https://medium.com/swlh/siem-vs-soar-whats-the-difference-f81cf830fd03>. (accessed: 15.04.2021).
- Seker, Ensar (2020). «Cyber Threat Intelligence (CTI) in a Nutshell». In: *Medium*.
- Shuffle installation guide* (2021). URL: <https://github.com/frikky/shuffle/blob/master/.github/install-guide.md>. (accessed: 15.02.2021).

- Siemplify (2021). *Siemplify Pricing*. URL: <https://www.siemplify.co/pricing/>. (accessed: 17.02.2021).
- Siemplify and Checkpoint Partnership* (2020). URL: <https://www.siemplify.co/press/siemplify-and-check-point-software-partner-to-enhance-and-accelerate-orchestration-automation-and-response-in-security-operations-centers/>. (accessed: 10.02.2021).
- Sikender Mohsienuddin Mohammad, Surya Lakshmisr (2018). «Security automation in Information technology». In: *International Journal of Creative Research Thoughts*.
- Splunk (2017). *What Is Security Automation?* URL: https://www.splunk.com/en_us/data-insider/what-is-security-automation.html. (accessed: 10.12.2020).
- Splunk community versions details* (2021). URL: <https://docs.splunk.com/Documentation/Phantom/4.10.2/Admin/License>. (accessed: 15.01.2021).
- Standards, National Institute of and Technology (2018). *Framework*. URL: <https://www.nist.gov/cyberframework/new-framework>. (accessed: 15.04.2021).
- TechEN (2021). *The six most powerful threat intelligence platforms for your security teams*. URL: <https://tech-en.netlify.app/articles/en528514/index.html>. (accessed: 25.05.2021).
- TheHive Github* (n.d.). URL: <https://github.com/TheHive-Project/TheHive>. (accessed: 03.10.2020).
- VirusTotal (2021a). *VirusTotal - Api documentation*. URL: <https://developers.virustotal.com/reference#url-report>. (accessed: 01.02.2021).
- (2021b). *VirusTotal - How it works*. URL: <https://support.virustotal.com/hc/en-us/articles/115002126889-How-it-works>. (accessed: 03.04.2021).
- WALKOFF Documentation* (n.d.). URL: <https://walkoff.readthedocs.io/>. (accessed: 01.10.2020).
- WALKOFF github* (2020). URL: <https://github.com/nsacyber/WALKOFF>. (accessed: 01.10.2020).
- WALKOFF Presentation* (n.d.). URL: https://github.com/nsacyber/WALKOFF/blob/master/1.%5C%20AVENGER_CON_WORKSHOP/AvengerCon%5C%20Presentation.pdf. (accessed: 01.10.2020).
- Wierzbicki, Maciej (2021). *Azure Sentinel vs. Azure Security Center*. URL: <https://www.predicagroup.com/blog/azure-sentinel-vs-azure-security-center/r>. (accessed: 15.05.2021).

DECLARAÇÃO

Declaro, sob compromisso de honra, que o trabalho apresentado nesta dissertação, com o título *"Resposta A Incidentes De Cibersegurança No Securityoperation Center Do Politécnico De Leiria"*, é original e foi realizado por Estudante Marco Alexandre Clemente Mateus (2190376) sob orientação de Professor Doutor Carlos Manuel da Silva Rabadão e Professor Adail Domingues da Silva de Oliveira .

Leiria, Novembro de 2021

Marco Mateus

Estudante Marco Alexandre Clemente Mateus