INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Análise prática de ferramentas de gestão de conteúdo de SOCs

TIAGO PORTUGAL GIL RIBEIRO GONÇALVES
setembro de 2024

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Practical Analysis of SOC Content Management Frameworks

Tiago Gonçalves

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Cybersecurity and Systems Administration

Supervisor: Professor Jorge Manuel Canelhas Pinto Leite
Co-Supervisor: Dr. Eva Catarina Gomes Maia

Evaluation Committee:
President:
Dr. Isabel Praça, Professor, Polytechnic of Porto

Members:

Porto, September 19, 2024

Statement of Integrity

I hereby declare having conducted this academic work with integrity.

I have not plagiarised or applied any form of undue use of information or falsification of results along the process leading to its elaboration.

Therefore the work presented in this document is original and authored by me, having not previously been used for any other end.

I further declare that I have fully acknowledged the Code of Ethical Conduct of P.PORTO.

ISEP, Porto, September 19, 2024

Dedicatory

Firstly i extend my sincere gratitude to all those who contributed to the success of the work presented in this thesis. Secondly, i am especially grateful to my family, friends, and my loyal dog. Their unwavering support and encouragement throughout this endeavor have been invaluable. I would like to express my deepest appreciation to my mother, whose love, support, and companionship have been a constant source of strength, not only during this thesis but throughout my life. Words cannot adequately convey my gratitude. Lastly, would also like to thank my supervisors, Professor Jorge Pinto Leite and Dr. Eva Maia. Their guidance, expertise, and availability have been instrumental in the completion of this work. I am particularly grateful for the knowledge they have shared and for their patience and support throughout the process.

Abstract

Cyberattacks have increased significantly in recent years, especially since the COVID-19 pandemic began [1]. In addition to the increase in quantity, the new types, tools, and techniques of attacks, as well as the increasing complexity of attacks, are allowing more cybercriminals to succeed [2].

In today's world, it is inevitable that organizations will be compromised. The question is not if they will be cyber-attacked, but rather when [3]. In this context, SOCs (Security Operations Centers) have emerged as crucial entities responsible for defending against these threats and helping organizations protect their assets. A SOC is an essential cybersecurity department within an organization, primarily helping to detect, manage, and respond to cyber threats [4]. However, SOCs deal with large and complex amounts of data. This can easily lead to rapid disorganization and inefficiency of an SOC, affecting its ability to detect, monitor, and mitigate cyber threats effectively. It can also reduce stakeholders' confidence in the SOC's ability to protect the organization.

Given this difficulties, this thesis aims to address existing methods for a continuous improvement of a SOC's maturity and capability, with a particular improvement in monitoring and detection of threats through use cases, by using Content Management Frameworks (CMFs). CMFs offer a structured approach to managing the content of SOC and allow for an easier identification of areas for improvement. By leveraging CMFs, organizations can their streamline the SOC content management, leading to an improved effectiveness and efficiency of the SOC, which then leads to greater stakeholder trust.

The proposed solution and its implementation leverage up-to-date and well-known continuous improvement methodologies such as Define, Measure, Analyze, Improve and Control (DMAIC) and Plan, Do, Check, Act (PDCA). It also uses frameworks for enchaining monitoring and detection of cyber threats, such as the Cyber Kill Chain, the MITRE ATT&CK and DeTT&CT frameworks, as well as its integration with CMFs.

Keywords: SOCs, Content Management Frameworks, Use Cases, Maturity

Resumo

Nos últimos anos, tem existido um aumento significativo de, especialmente desde o início da pandemia do COVID-19 [1]. Para além do aumento da quantidade, os novos tipos, ferramentas e técnicas de ataques, como tabem da complexidade dos mesmo têm permitido, aos cibercriminosos, cada vez sucederem [2].

No mundo de hoje, é inevitável que as organizações sejam comprometidas. A questão não é se serão atacadas ciberneticamente, mas sim quando [3]. Neste contexto, os SOCs surgiram como entidades cruciais encarregadas de defender contra essas ameaças e ajudar as organizações a proteger seus ativos. Um SOC consiste num departamento de cibersegurança essencial de uma organização, que auxilia principalmente na detecção, gestão e resposta a ameaças cibernéticas [4]. No entanto, os SOCs lidam com grandes e complexas quantidades de dados. Isso pode facilmente levar a uma rápida desorganização e ineficiência de um SOC, prejudicando a sua capacidade de detectar, monitorizar e mitigar eficazmente as ameaças cibernéticas. Pode também reduzir a confiança dos stakehlders na capacidade de um SOC de proteger a organização.

Dadas estas dificuldades, esta tese pretende endereçar os métodos existentes de melhoria continua da maturidade e capacidade de um SOC, com um foco em particular na melhoria de deteção e monitorização através de cases de uso, com recurso a ferramentas de gestão de informação (FGI). As FGI oferecem uma abordagem estruturada para gerir o conteúdo de um SOC e permitem identificar com maior facilidade areas que necessitam de melhoria. Ao utilizar FGIs, as organizações podem simplificar o processo de gestão de informação, o que levan a uma maior eficácia e eficiência do SOC, o que se traduz numa maior confiança das stakeholders.

A solução proposta e sua implementação utilizam metodologias de melhoria contínua conhecidas e atualizadas, como a Definir, Medir, Analisar, Melhorar e Controlar (DMAIC) e Planear, Fazer, Verificar, Agir (PDCA). Também utiliza frameworks para aprimorar a monitorização e a detecção de ameaças cibernéticas, como o Cyber Kill Chain, o MITRE ATT&CK e DeTT&CT, bem como sua integração com CMFs.

Keywords: Ferramentas de Gestão de Conteúdo, Casos de Uso, Maturidade, Melhoria Contínua

Contents

List of Figures

List of Tables

List of Source Code

Chapter 1

Introduction

A Security Operations Center (SOC) is a critical component of an organization's cybersecurity strategy. It is responsible for monitoring, detecting, and responding to security incidents, threats, and vulnerabilities across the organization's infrastructure [3]. However, most SOCs lack the necessary processes, procedures, and improvements required to operate efficiently and effectively [5]. Frameworks for dealing with these processes, procedures, and improvements can be used to streamline processes and improve decision-making.

This chapter outlines the context and motivation behind this thesis, detailing the research problem, objectives, and associated research questions. It also presents the scientific contributions and the overall structure of the document.

1.1 Context and Motivation

Technology has become increasingly interconnected in recent years, offering various opportunities to improve human life. These advances could be medical innovations, improved enjoyment through interactive games, and enhanced communication effectiveness, among many other advantages [6]. However, while technology has brought numerous improvements and benefits, it has also presented a growing challenge to the increased ease and frequency with which these technologies can be exploited for malicious purposes [7]. As technology evolves in complexity and interdependence, the potential for cyberattacks' complexity and success increases accordingly.

The increase in cyberattacks in recent years, particularly when the COVID-19 pandemic began, has severely affected businesses in all sectors [8]. These attacks often result not only in significant financial losses, but also leakage of sensitive personal information, such as addresses, passwords, or credit card details. According to the International Business Machines (IBM) corporation, the average cost of a data breach in 2024 is **4.88 million** United States dollars (USD) [9], representing a 10% increase over last year and the highest total ever. Also, in 2022, Cybersecurity Ventures predicted that, in 2023, cybercrime could cost a staggering 8 trillion USD a year, equivalent to 667 billion USD a month [10]. This figure would place cybercrime as the third largest economy in the world if measured as a country, behind the United States of America and China.

The growing costs and sophistication of cyberattacks underscore the importance of cybersecurity for businesses of all sizes. The ExPetr attack of 2017, more commonly known as the WannaCry ransomware attack, is the most expensive cyberattack to date, with estimated 10 billion dollars in losses [11, 12]. On 7 February 2022, Vodafone Portugal fell victim to a cyberattack that led to a significant disruption in critical infrastructures, resulting in a

communication shutdown that lasted for an hour. This incident had a severe impact on various sectors, affecting ATMs, disrupting hospital services, and even causing interruptions in police phone lines [13]. In a separate incident in May 2023, a software vulnerability in MOVEit (software for file transfer) led to a data breach. This breach had far-reaching consequences compared to the Vodafone attack, affecting over 17.5 million individuals globally and exposing confidential personal data of both customers and employees across several companies [14].

As cyberattacks continue to grow in volume and sophistication, and given that governments, organizations, institutions, universities, and other businesses collect, process, and store a large amount of confidential information, it is crucial for these businesses to prioritize cybersecurity measures to protect their data, systems, and reputations. To make efforts to prevent security breaches and all the negative impacts that can come with them, some organizations have a dedicated unit for cybersecurity, mostly known as a Security Operations Centers (SOC) [5, 15]. SOCs can be viewed as a complex structure responsible for managing and controlling the security of a company. This structure incorporates people, processes, technologies, governance, and compliance, with the aim of effectively preventing, detecting, and mitigating threats, ideally before any damage occurs [16]. SOCs are one of the most critical defense systems of modern organizations and are also responsible for coordinating and managing cyber incidents in case of possible threats [5].

As it can be seen, establishing a SOC has become an essential step for organizations seeking to enhance their overall security posture and proactively prevent and mitigate risks. However, creating a SOC is only the the first of many steps to achieve efficient cyber-safety [17]. Following the establishment of a SOC, a crucial next step is continuously improving its effectiveness and efficiency, thus making it more mature and capable. This continuous improvement process ensures that the SOC remains adaptable, effective, and aligned with the evolving security needs of the organization [5].

Despite the importance of continually improving the effectiveness and efficiency of SOCs, organizations with a SOC often face significant challenges to achieve this goal [5]. SOCs typically handle vast amounts of data, and managing such information efficiently requires a structured manner and substantial investment of time and resources [18]. Without such efforts, SOCs may fail to monitor, detect and respond to cyber threats, and may also misaligned with the strategic objectives of the business [19], all resulting in poor maturity and capabilities.

To address these challenges, SOCs can implement Content Management Frameworks (CMFs). CMFs provide a structured and methodical approach for organizing and optimizing SOCs information. By leveraging CMFs, SOCs can more effectively improve the overall security, leading to an improvement in its maturity and capabilities and an increased confidence of the stakeholders.

Given the important that information management has for improving detection and monitoring of cyber threats (two of the most important aspect of a SOC) and to create a better alignment with business goals, a special attention can be given to Use Case Frameworks (UCFs), a type of CMFs. Use cases (UCs) are used to provide a structured approach to security monitoring and detection. They offer a more clearer and structured approach to manage the SOCs detection and monitoring information. UCs are also tied with business drivers [20]. To have better control over UCs, UCFs can be used. Such frameworks would

allow more control over UCs and provide insight into identify how well an organization is capable of defending against cyber threats and what business needs are tied to the UCs.

Despite the promising potential benefits of integrating such frameworks into SOCs, research on their importance and implementation is very limited. Given this, additional investigation is required to address this pertinent challenge.

1.2 Problem Statement

As already understood, SOCs are an extremely important unit in many companies and provide a defense system against cyber threats [18]. Their role in combating the ever-evolving cyber threats is crucial in a world where the threat landscape continues to expand. However, without an effective and efficient SOC, a company can still be highly vulnerable to cyberattacks [21].

A survey conducted by Devo Technology in partnership with the Ponemon Institute underscores that issue, revealing that most respondents of the survey rate their SOC's effectiveness as low. Moreover, 49% believe that their operations are not adjusted to business needs [22]. This poor SOCs effectiveness and business misalignment makes difficult to gain senior leadership's commitment to providing adequate funding for investments in technology and staffing. Further, the SOC budget is inadequate to support the necessary staffing, resources, and investment in technology, leaving a company more vulnerable to cyber threats.

Ideally, with certain exceptions, a SOC should have a continuous improvement and strive to be highly capable and mature [23, 24]. This means having the skills and tools to effectively detect and respond to cyber threats, while also having an organized and structured approach in place to operate in a consistent manner. However, an increase in maturity does not necessarily mean an increase in capability, as the other way around is also true. A highly capable SOC can have low maturity, and, conversely, a highly mature SOC may be limited in its capabilities [20]. On one side, a SOC that possesses high capability but lacks maturity can, for instance, efficiently identify threats, yet it might not be able to do it systematically because of the lack of standardized processes and procedures. On the other hand, a SOC that is highly mature but has limited capabilities is well organized but may not be effective, failing to achieve its objectives of detecting and mitigating threats. Therefore, it is crucial for a SOC to achieve both a high level of maturity and capability.

Given that most SOCs handle vast amounts of data, building more mature and capable SOCs requires a detailed and systematic method for managing information. Adopting a more systematic and uniform approach to information management can improve data quality, enable more accurate threat detection and response, and provide other benefits. This, in turn, enhances the primary objective of a SOC of monitoring, detecting, and responding to cyber threats [20]. A structured approach also facilitates better alignment with the organization's business needs, which is essential for SOC stakeholders.

In this context, UCs provide a structured approach to security monitoring, detection, and business alignment. Given their characteristics, they should be given significant attention to enhance the SOC's ability to detect and respond to cyber threats, and to align the SOC with the business needs, thereby improving its maturity and capabilities. To organize the UCs, CMFs, with a focus on UCFs, could emerge as valuable solutions. However, there is a lack of detailed information about such frameworks and their practical implementation. Therefore, research is required to assess their effectiveness for SOCs.

In short, a reactive approach to managing SOCs without a clear focus on enhancing their maturity and capabilities can result in significant failures in preventing cybersecurity breaches across various industries. This lack of proactive measures can reduce stakeholder confidence, demotivate SOC teams, and even disrupt daily operations which, in some cases, can potentially compromise the safety of individuals. Even more, without a considerate level of maturity and capability, it is more difficult to demonstrate SOC stakeholders the effectiveness and efficiency of SOCs. Therefore, it is crucial to assess and strengthen the SOC maturity and capabilities, particularly in detection and monitoring phase, which plays an most important role in preventing real cyber threats. In this case, CMFs, with more emphasis in UCFs, could allow for this improvements, and thus will be studied.

1.3 Ethical Considerations

Ethics, in its broadest sense, is a system of accepted beliefs control behaviour [25]. In contemporary contexts, ethical considerations extend to various domains, including technology, notably cybersecurity. Cybersecurity ethics are crucial for safeguarding the integrity, functionality, and reliability of data-dependent systems and institutions. Such ethics in the cybersecurity field protect the rights and interests of individuals, organizations, and society while complying with legal and regulatory data protection and cybersecurity standards. Unethical misconduct can lead to legal action and regulatory penalties.

Cybersecurity professionals in SOCs deal with large amount of systems that collect and instrument sensitive private data. Therefore, having appropriate measures becomes even more important in such systems to prevent security breaches. Considering this thesis problem in addressing SOCs and its information management, this can raises privacy concerns issues due to its potential access. This could lead to data breaches and privacy violations. Therefore, during this thesis development, no sensitive content was disclosed that could impact the SOC. All the sensitive information needed in for the thesis was always informed to the SOC team. Also, artificial intelligence was used to rephrase information of this document. However, this process was conducted without exposing sensitive information from being leaked.

1.4 Objectives and Research Questions

This part of the document aims to investigate existing ways to assess capability and maturity and to improve both, using CMFs, particularly UCFs. To achieve this goal, four specific objectives (OB) were formulated to guide the research:

This section of the thesis explores current methods for assessing capability and maturity, as well as strategies for enhancing these areas, through CMFs, with a focus on UCFs. To achieve this, four key objectives (OB) have been defined to guide the research. These are be defined as follows:

- **OB1**: Conduct a research on existing ways to assess the maturity and capability of a SOC.
- **OB2**: Investigate and evaluate the most practical and suitable CMF(s), with an emphasis on UCF, to improve the maturity and capacity of SOC.
- **OB3**: Implement a Proof-of-Concept (PoC) to validate the selected CMF's effectiveness.

- **OB4**: Analyze and evaluate the impact of the selected solution.

To effectively guide the research in this work and ensure that the previous objectives are met, two Research Questions (RQs) were formulated:

- **RQ1**: What is the state-of-the-art approach to self-evaluating a SOC maturity and capability?
- **RQ2**: Which existing CMFs can enhance the management of SOCs content through UCs, with emphasis in monitoring, detection, and business alignment, leading to a SOC maturity and capability improvement?

1.5 Scientific Contributions

Completing the previously established objectives allows for the following scientific Contributions (C):

- C1 – A literature review of maturity and capability assessment of SOC assessment and CMFs.
- C2 – A functional CI tool for defining, measuring, analyzing, improving and controlling a SOC aspect.
- C3 – One case study and two SOC expert opinions on the real-world application of the tool.

1.6 Document Structure

This dissertation, for now, is divided into two chapters, which can be described as follows:

- Chapter 1 provides a comprehensive overview of the research context, including the motivation, challenges, objectives, and research questions that guide this thesis.
- Chapter 2 presents the state-of-the-art information obtained. It is divided into three sections. Both the first an second section present the adopted research methodology, the findings and discussion about the findings, for each of the RQs, respectively. The third section presents conclusion remarks of the chapter.
- Chapter 3 describes the proposed solution, by conceptualizing the tool, identifying its target audience, and decisions about the solution. The system architecture is also illustrated, along with defined requirements.
- Chapter 4 describes the practical implementation of the solution, providing a PoC for a SOC in production.
- Chapter 5 presents the a demonstration of the implemented solution, along with its results. A insurance case study and SOC professional opinions were also documented
- Chapter 6 concludes the work, by exploring current limitations of the work developed, assessing the objectives attained in the proposal of this thesis, answering the research questions and stating future work.

Chapter 2

State-of-the-art

This chapter presents the literature review that was performed to thoroughly investigate the formulated RQ. In the following sections, the research methodology adopted for each question is described, along with the respective findings is presented. A discussion about the finding for each RQ is also provided.

2.1 SOC Maturity and Capability Models

The concept of maturity models originated in 1973, when Richard Nolan observed a gap in planning and controlling IT resources [26]. This consequently led to an inefficient use of IT resources. With this, Nolan proposed a maturity model with the main objective of addressing these issues, which later served, for other investigations, as a starting point in the development and improvement of processes and maturity models.

Recognizing the growing importance of SOCs, recent studies and proof-of-concept projects merging maturity models and SOCs began to appear. Given the evolving cyberthreat landscape and limited research on SOC maturity models, this section aims to address RQ1: 'How can SOCs effectively self-assess their maturity and capabilities?

2.1.1 Research Methodology

To have a significant in-depth knowledge about the previously defined RQ1 question, a review of the literature was performed, partially adhering to Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA [1]). PRISMA serves is a guideline that allows authors to perform easier preparation and reporting in a transparent, replicable, and complete way by defining an evidence-based checklist with a minimum set of items [27].

The initial phase of researching RQ1 involved reviewing a broad selection of articles in Google Scholar[2] pertinent to the topic concerning SOC maturity and capability models. After having a deeper understanding of the topic, related keywords were put together in a query to obtain the relevant articles from known scientific databases. The Association for Computing Machinery Digital Library (ACM DL [3]) was used due to its extensive collection of articles and literature specifically related to computing and information technology. Its repository contains articles from the Institute of Electrical and Electronics Engineers (IEEE) Xplore [4] but not all of the articles, so IEEE Xplorer was also used to search for relevant articles. A

[1] http://www.prisma-statement.org/?AspxAutoDetectCookieSupport=1
[2] https://scholar.google.com/
[3] https://dl.acm.org/
[4] https://ieeexplore.ieee.org/Xplore/home.jsp

search in ScienceDirect [5] was also performed since it is a large and reputable database of scientific publications provided by Elsevier, an internationally recognized academic publishing company that includes millions of journals, articles, and books that can be used to search about engineering, science, technology, medicine, and others topics. It is important to note that the snowballing process of checking the references of the findings also led to additional publications that were not directly obtained from querying these databases.

After the research made, the obtained articles were then screened (reading title, abstract and conclusion), leading to a filter of the keywords and an update of the research query. Since RQ1 was created to understand the state-of-the-art of methods for assessing the maturity of SOCs, the final query was obtained and is presented in Figure 2.1.

```
("Security Operations Center" OR "Security Operation Center" OR "Security Operation Centers" OR
"Security Operations Centre" OR "Security Operation Centre" OR "Security Operation Centres")

AND

"capability maturity model"
```

Figure 2.1: Search Query

Inclusion and exclusion criteria were also created but with careful consideration, since from the beginning it was understood that there was a lack of research and published articles that could help answer RQ1. The criteria included both conference papers, journal articles, and articles in English that addressed methods to assess maturity and capability of SOCs. It were excluded duplicated works, and papers that did not have the full text available. The inclusion and exclusion criteria are summarized in Table 2.1

Table 2.1: Inclusion and Exclusion Criteria

Inclusion Criteria (IC)	**Exclusion Criteria (EC)**
IC1 - Peer-reviewed journal article or conference paper	**EC1** - Duplicate publications
IC2 - Available in English language	**EC2** - Full text not available
IC3 - Addresses methods to assess maturity and capability of SOCs	

2.1.2 Findings

In the contemporary digital landscape, organizations employ SOCs as a key strategy to defend their Information Technology (IT) assets [28], but the importance of continuous SOC improvement to protect organizations from cyber threats cannot be overstated. However, its evaluation and enhancement could present complex challenges. Problems such as staff fluctuations, inefficient communication, lack of collaboration among various teams in the organization [21], the use of a variety of technologies, and the provision of a wide range of services can all make it difficult to improve SOCs.

Having a clear understanding of the current state of the organization and having a well-defined vision for its future are crucial factors for a successful implementation of business

[5] https://www.sciencedirect.com/

processes in SOCs. Continuous planning and improvement are essential, and when describing the maturity level of a SOC, it would be prudent and beneficial to refer to existing established IT management frameworks. The fundamental frameworks relevant here are [29]:

- **Control Objectives for Information Technology (COBIT)** - is a highly regarded framework that builds on established industry standards and best practices. COBIT takes a business-driven approach to IT management. It helps organizations identify and address critical IT criteria that align directly with their business goals. Instead of dictating specific methods, COBIT emphasizes the importance of effective IT processes and outlines the key objectives organizations should strive for [29].
- **Information Technology Infrastructure Library (ITIL)** - is a framework that focuses on the management of IT processes, covering the planning, sourcing, designing, implementing, operating, supporting and improvement of IT services [29].
- **International Organization for Standardization (ISO/IEC 27001)** - is a international standard for information security. This framework provides assistance and guidance to security specialists in the implementation of information security within an organization and, as ITIL, it focuses on IT processes [29].

CoBIT and ITIL can be coupled with information security framework ISO/IEC 27001 [4, 29], to provide a comprehensive IT management system that improves information security, IT service levels, and compliance, while also increasing transparency and flexibility for the organization. These frameworks served as the basis for a model developed in 2013 and described in [4] . The authors recognized that there was no holistic model that addressed the services of SOCs, including processes, staffing, and technology. To address this gap, they created a model based on industry-accepted maturity models and the previously described frameworks (CoBIT, ITIL, and ISO/IEC 27001) to enable evaluation of both the capabilities and maturity of the SOC's services.

However, the model presents certain limitations. As noted in [30], although it offers a general overview of a SOC, it lacks in providing a complete analysis of SOC operations, alignment, and maturity. Additionally, SOCs evolved and are, at the moment, recognized as organizational entities comprising four primary services: people, processes, technology, and governance and compliance, as noted in [31]. This evolution leads to an outdated modal, not being able to evaluate all the current SOC services and for its maturity and capability.

To assess the overall maturity of SOCs, it's necessary to evaluate the maturity of each individual service it provides [32]. In order to perform such an evaluation, organizations should adopt well-defined and standardized strategies to analyze and refine not only their SOC's maturity but also its capabilities. Based on the presented flaws for the previous model and the lack of research and options on how to define and improve the maturity and capability of SOCs, Capability Maturity Model Integration (CMMI[6]) emerged, in studies [32, 33], as an option to establish a solid foundation for evaluating SOCs. CMMI is a capability maturity model for assessing organizational processes, focusing primarily on their end results. The objective of the CMMI model is to assess the maturity of an organization's processes and provide guidance on improving processes, with the goal of improving products [34].

Despite the consideration that the authors of [33] and [32] took in CMMI, both also stated that literature on specific CMMI for SOCs is extremely scarce and produces generic results for SOCs. Furthermore, [33] highlights that although some research has been done in the

[6] https://cmmiinstitute.com/

field of SOC models, these studies do not provide sufficient detail or a specific focus on capability and maturity. This absence of a well-established and specific model designed for evaluating the maturity and capability of SOCs led to the creating of the SOC-Capability Maturity Model (SOC-CMM), in 2016, and presented in [32]. Given that this model is the most frequently referenced in relevant research publications related to the assessment of the maturity and capability of SOCs, such as [31, 35, 30, 33], it will receive more attention when compared to other maturity and capability models, in this state-of-the-art chapter.

SOC-CMM consists of a model, implemented and accessible in an Excel[7] assessment tool, allowing SOCs to be evaluated according to their maturity levels and capability [32]. To design the SOC-CMM model, the author conducted a two systematic literature review methodology. One to understand the available information on SOCs and a second one to determine the available information on CMMs. Based on the research, the SOC-CMM model was created and is depicted in Figure 2.2

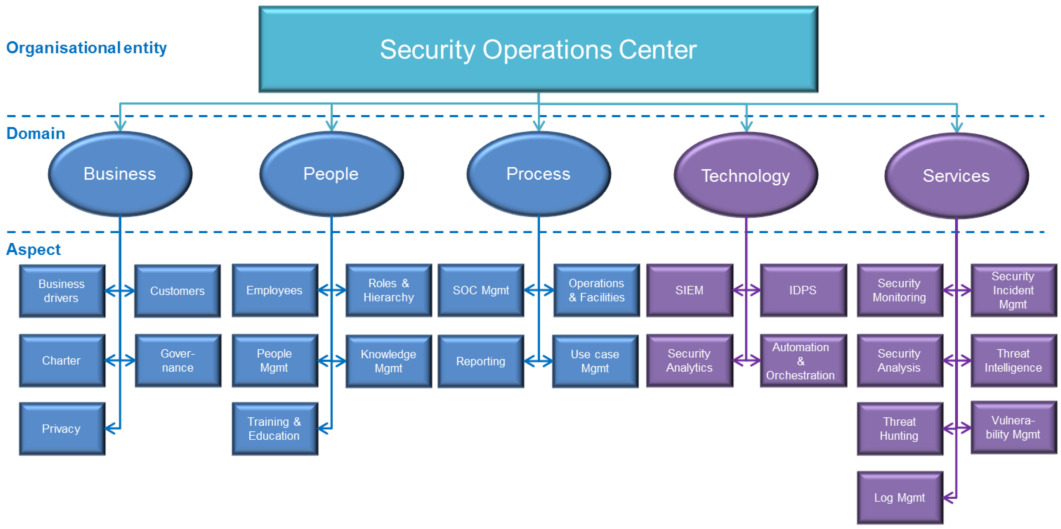


Figure 2.2: SOC-CMM Domains and Aspects [32]

The SOC-CMM divides SOCs into five domains (domains are logical groups of a SOC). The domains are Business, People, Process, Technology, and Services. Each domain has aspects, which are SOC functionalities or services, making a total of 25 aspects. The domains and aspects in blue are evaluated only for maturity. The aspects and domains in purple are evaluated for both maturity and capability.

To evaluate each aspect's maturity or capability, the author of SOC-CMM created specific questions. These questions are based on the CMMI model, with modifications in order for the question to be more adequate to SOCs. The general rule for the creation of these questions was:

- Has the aspect been formally identified? This is an indicator of maturity level 'Initial'.

- Has the aspect been formalised for repeated quality? This is an indicator of maturity level 'Managed'.

- Has the aspect been fully documented and formalised? This is an indicator of maturity level 'Defined'.

[7] https://www.microsoft.com/pt-pt/microsoft-365/excel

- Is the aspect being measured for process optimization goals? This is an indicator of maturity level 'Quantitatively managed'.

- Is the aspect being measured for organisational optimization goals? This is an indicator of maturity level 'Optimizing'.

For example, to assess the Business drivers aspect, the author of SOC-CMM used the following five questions:

- Have you identified the main business drivers?

- Have you documented the main business drivers?

- Do you use business drivers to in the decision making process?

- Do you regularly check if the current service catalogue is aligned with business drivers?

- Have the business drivers been validated with business stakeholders?

For each of the questions, a 5-point scale was used. The answers selected for the questions relate to the maturity levels. Table 2.3 shows a the maturity scoring mechanism.

Table 2.2: Maturity scoring mechanism

Answer	**Score**
Incomplete	0
Partially complete	1.25
Averagely complete	2.5
Mostly complete	3.75
Fully complete	5

Also, to offer greater flexibility in determining the appropriate maturity level, a weighting feature was also added to the SOC-CMM, in its Excel self-assessment tool. For weighting, the author used a similar mechanisms as the CREST [36] and Group Service Integration Maturity Model (OSSIM) [37] tools, which are both maturity assessment models. Basically, this mechanism provides a means to personalize the tool to the SOC under evaluation.

For each sub-criterion under evaluation, the weighing could be changed to influence the scoring of the criterion. By assigning different weights to various aspects, the SOC-CMM self-assessment tool can prioritize elements that are more critical to the organization's objectives. For example, an aspect may not be implemented in a particular organisation the element can be removed from scoring so that it does not negatively affect the score.

Table 2.3 shows the weighing mechanism.

Table 2.3: Maturity scoring mechanism

Importance	**Factor**
None (removes from scoring)	0
Low	0.5
Normal	1
High	2
Critical	4

For the capabilities in SOC-CMM, these are assessed as a percentage, with a full implementation of all capabilities resulting in a 100% score. To allow for more granular assessment, each capability is measured on a 5-point scale, enabling the organization to define its capability level with precision. For capabilities deemed irrelevant to the SOC, an additional option, 'Not Required,' is provided to exclude them from the overall capability score.

After answering all the question in the SOC-CMM assessment tool, a web-diagram as the one seen in Figure 2.3 is presented. This diagram presents all the SOC's defined domains, the respective aspects, the maturity and capability levels, and two different types of line. One blue line and one green line. The blue line indicates maturity levels, and the green line represents capability levels. Only the services and technology domains are evaluated for capability, which justifies the fact that no other SOC domains have a green line.

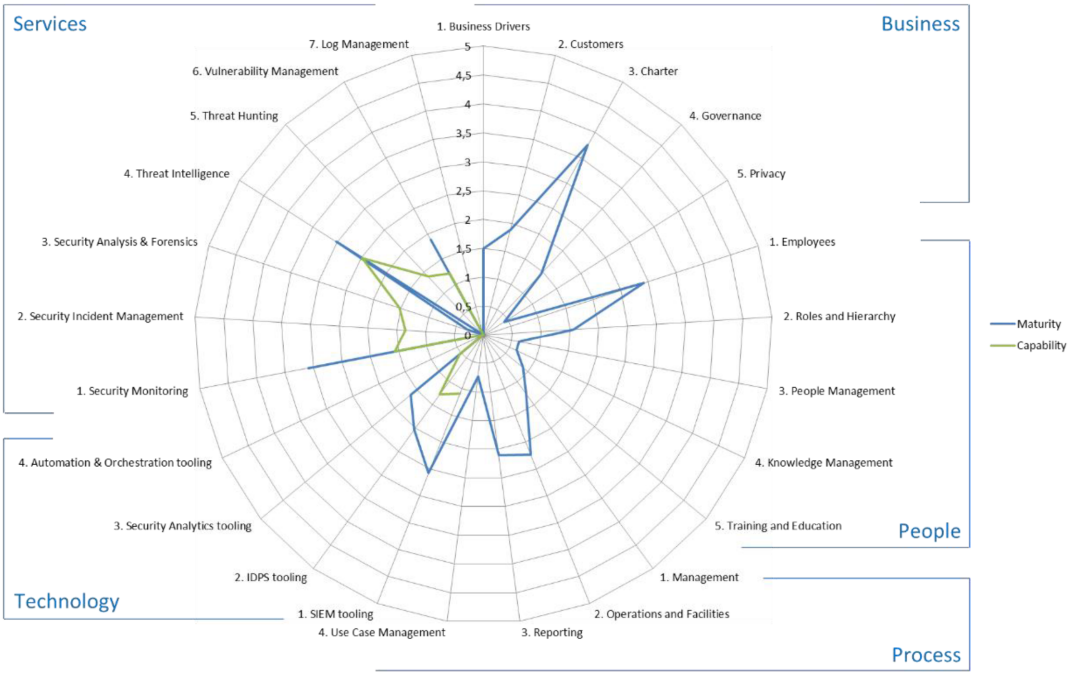


Figure 2.3: Maturity and Capability Scores Visualization [35]

A key strength of the SOC-CMM is its alignment with the National Institute of Standards and Technology (NIST) Cybersecurity Framework (NIST CSF) 2.0. This widely adopted framework incorporates best practices from COBIT, ISO/IEC 27001, and other NIST standards [32]. In essence, NIST CSF provides a standardized approach to cybersecurity risk management, allowing businesses of all sizes to implement effective controls. The SOC-CMM assessment tool reflects this alignment by including a mapping to NIST CSF 2.0. This mapping allows to visualize how the SOC-CMM results correspond to different NIST CSF categories, facilitating comparison with other widely used frameworks.

Although the SOC-CMM is a comprehensive model and takes into account both maturity and capability assessment, it also presents flaws. The author of SOC-CMM recognizes that each SOC is unique. Hence, designing a generic model or tool that can be applied to all SOCs is very difficult, and this model is unlikely to cover all SOC activities. Also, models like SOC-CMM lack guidance on how to actually improve maturity and capability. CMMs for SOCs generally focus on identifying areas for improvement (what to improve) without providing a clear roadmap for implementing these improvements (how to improve).

This absence of a systematic improvement methodology can difficult organizations to better protect against cyberattacks, making it difficult to improve their overall security posture [35].

To address the need for a consistent SOC improvement framework the authors of [35] proposed, in 2020, an approach of comparing different Continuous Improvement (CI) methods and applying it to SOCs.According to [38] CI is defined as 'a broad change program, planned, organized and systematic, and distinguished from project-based models of change'. The authors argue that for a sustainable protection against cybersecurity threats, cybersecurity teams must rely on CI and rapid adaptation to the ever-evolving cyber threat landscape.

In this study, different CI approaches are compared to understand the best fit for SOCs. Six CI methodologies were described and evaluated in by the authors:

- Plan, Do, Check, Act (**PDCA**) - This is a simple but effective methodology. However, it oversimplifies improvement process which makes it good for service-level continuous improvement but not the most adequate option for large-scale complex changes.
- Define, Measure, Analyze, Improve, and Control (**DMAIC**) - This is a data-driven methodology for enhancing existing processes and is integral to the Six Sigma approach, which systematically employs collected data and statistical analysis to assessment, measurement, and improve performance. This characteristics made the DMAIC methodology a good fit to be used for SOCs.
- Design for Six Sigma (**DFSS**) - The main objective of DFSS is designing right at the first time, therefore, this approach was suggested as a best fit for new products or processes but not SOCs because these, most of the times, are already active and providing ongoing services.
- **Project life cycle** (no analyses was given by the authors).
- **Radar** - Similar to DFSS, Radar is also a complex methodology as a longer term and resource-demanding process, also not making this methodology not the best fit for SOCs.

Based on the proposed CI methodologies, the authors proposed two CI methodologies. One of the CI methodologies considered was DMAIC. This choice was made because DMAIC is a data-driven model. In other words, the measurement and evaluation of the information is a crucial part in this methodology. Given that measurement, evaluation, and enhancement of the maturity and capacity are also one very important aspect for most SOCs, DMAIC model was considered a well-suited methodology for the study. However, DMAIC was used at the organization level, meaning that it involves maturity and capability assessment of all the SOC's services, in order to better determine which services as a priority to be improved. For the improvement of each service, given that is a specific task that must be evaluated and processed separately, a second methodology, PDCA cycle, was used for service-level improvements due to its simplicity and adaptability.

The DMAIC methodology of Six Sigma approach has been defined in five steps, namely problem definition (**D**), measurement of the problem (**M**), data analysis (**A**), improvement process (**I**) and controlling or monitoring process to prevent recurring problems (**C**) [39]. In the study, each phase of DMAIC was applied in the context of a SOC. Since PDCA was considered a service-level methodology, and the improvement of a service of a SOC happens

in the Improve phase of DMAIC, the authors of the study used PDCA in the Improve phase. The DMAIC phases and their respective proposed information for each phase are:

- **Define** - Scope and objectives of the organization; Stakeholders affected by the SOC; Budget for the improvement plan; Stakeholders expectations and metrics needed;
- **Measure** - Prioritization of stakeholders and metrics; What should be measure and how to measure; Gaps for improvement; Objectives of the CI process;
- **Analyse** -Services/components to be improved; Root causes of the problems in services/components; Cause–effect diagram for expected improvements; Financial requirements for the determined improvements;
- **Improve (PDCA)** - Brainstorm about possible solutions; Risk assessment of the potential solutions; Defining and implementing the best solutions; Re-evaluation of the impact of performed improvements;
- **Control** - Documentation of results documentation; Development/improvement and documentation of the standards; Alignment with the updates in the system.

To illustrate a practical application and a better understand how the proposed methodology works, the authors provided an example of a hypothetical scenario. Briefly:

1. In the first phase **(Define)**, the type of SOC model, geographic operation, relevant stakeholders and services were defined, along with expected maturity and capability levels for each SOC service, having in account the budget and expectations.
2. In the second phase **(Measure)**, an assessment of the SOC using the SOC-CMM self-assessment tool (the Excel) was used to assess the existing maturity and capability levels of the organization. The SOC-CMM web-diagram with maturity and capability scores were shown, similar to the one presented in Figure 2.3.
3. During the third phase **(Analyze)**, the results of the web-diagram were scrutinized, revealing that two aspects of the SOC (security incident management and security monitoring), which corresponded to Incident Handling & Response (IRH), exhibited the lowest results when compared to the desired maturity and capability levels established in phase 1.
4. The subsequent fourth phase **(Improve)** involved the application of the PDCA cycle. The used of this methodology allow for an iterative improvement of the IRH service.
5. In the last phase **(Control)**, updates were documented, relevant stakeholders were informed of the changes made, and required precautions were defined and implemented to make the updates permanent in the system.

At a high level, this scenario demonstrates the use of a CI approach to focus on the 'how' to improve maturity and capability for SOCs. The applied methodology could be viewed as an improvement to the Rob van Os' SOC-CMM, as it presents not only 'what' needs to be improved, but also 'how' to make those improvements.

In case of limitations, the authors of the study acknowledge that the effectiveness of the proposed methodology could not be fully assessed due to its difficulty in being applied to real-world situations. The implementation of this methodology in an actual SOC environment was difficult due to the vast amount of confidential data involved. This makes data collection

and publishing the methodology itself challenging. The lack of quantitative data and the sensitivity of the information hinder definitive conclusions about its effectiveness in practice.

Another limitation lies in the subjectivity involved in setting target maturity and capability levels for the methodology. While it considers factors like budget constraints, customer expectations, and Critical To Quality (CTQ) definitions, the final decision ultimately relies on the judgement of stakeholders. This introduces subjectivity into the process.

Finally, the study focused solely on the Incident Handling and Response (IHR) service of an SOC. This raises questions about the applicability of the service-oriented architecture of the methodology to the broader SOC environment. Further research is needed to determine if the methodology can be effectively adapted for other SOC services.

In 2021, based on the established SOC-CMM, the Security Operation Center - Assessment Model (SOC-AM) was also proposed as SOC assessment modal. However, when compared to SOC-CMM, SOC-AM focuses specifically on assessment of maturity and not capability. Additionally, the author of the SOC-AM references that SOC-CMM is very detailed, meaning that it requires an extended amount of time complete it and also could be bias towards certain types of SOC, given the detail that that question have. To address this, the SOC-AM was designed with a more streamlined approach, featuring a reduced number of questions and allowing for the assessment to be as independent as possible from the type of business environment where a SOC exists [33].

Like the creator of SOC-CMM, the SOC-AM work highlights the scarcity of academic research that supports the assessment of SOCs. This scarcity compromises the main goal of the SOC-AM author, which is creating a maturity model that allows to evaluate and compare different SOC implementations in different business environments.

Similarly to the CMMI for software development, SOC-AM utilizes a 0 to 5 grading scale to assess the maturity of a SOC across its entirety, individual domains, and specific aspects within those domains (0 - Nonexistent, 1 - Identified, 2 - Managed, 3 - Defined, 4 - Reviewed and Updated, and 5 - Continuously Optimized). These levels provide a structured and easy-to-understand assessment of a SOC's security posture.

To ensure its ease of use, the number of assessment questions for each aspect is limited to a maximum of three (with exception for the domain named implementation, which has six questions). This simplified approach allows to balance precision of the model and ease of use, and still provide a quick overview of areas requiring improvement. It allows for a quick overview of areas where improvements are needed without overwhelming users.

The overall maturity score is calculated as the average of all domain scores. While this approach might introduce slight variations in precision, it offers a holistic metric that simplifies reporting and facilitates comparisons between different SOCs. Furthermore, SOC-AM recognizes the importance of flexibility to set priorities for specific aspects, meaning that users can adjust the importance of services, leading to a possible influence in the final score calculation. This prioritization feature allows users to tailor the assessment to their specific security needs and objectives.

Figure 2.4 illustrates the proposed SOC-AM model, which closely resembles the SOC-CMM already presented in Figure 2.2. However, comparing both figures, in the SOC-AM modal the 'Business' domain has been renamed to 'Operations', and its 'Privacy' aspect has been omitted. In the 'Process' domain, 'Use Case Management' was removed. Also 'Automation & Orchestration' aspect was removed from 'Technology' domain. Similarly, in the 'Service'

domain, the 'Threat Hunting' aspect has been removed. Additionally, a new domain named 'Implementation' was added, with the same aspects as the 'Technology' domain.

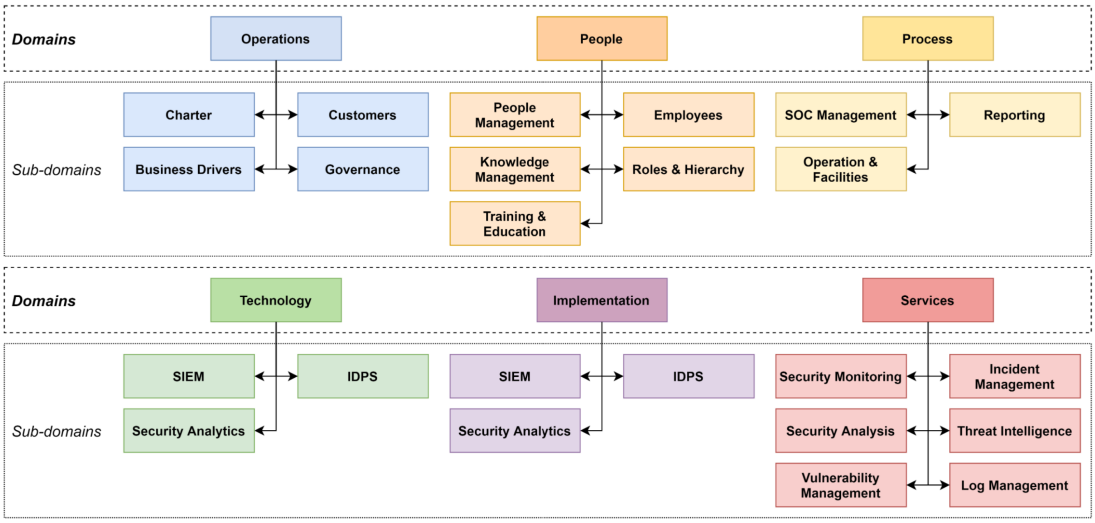


Figure 2.4: SOC-AM [33]

The model also has a self-assessment tool in Excel and, in discussions with SOC professionals, the tool went through an iterative process of receiving feedback and improving it with the given suggestions. The main changes implemented, when comparing with the initial Excel made, during the iteration process, were as follows:

- Two additional columns were added to each domain to allow assessors/auditors to leave notes that could be later reviewed to understand the train of thoughts that may explain a specific answer.
- Domains, sub-domains, and questions were assigned references of respective form XX, XX00, and XX00 11, where XX represents the first two letters of the domain, 00 the sub-domain number in a sequence, and 11 the question number, also in a sequence.
- Possibility to set emphasis on certain domains or subdomains when performing calculation of maturity.
- Option to set a target score to each question in order to conduct gap analysis.

After these improvements, the model was tested in two different SOCs for companies who needed feedback and guidance. The SOC-AM model, in general, was reviewed as very helpful, although the author stated that no fully review of was possible due to the lack of time and availability it took for SOC personal to provide that information.

The last model discussed in this literature review is the SOC-MF, presented in [30]. This model was created in 2022 and consists in a small-scale SOC-CMM, also with a self-assessment tool. Both model and tool were designed to guide IT auditors and streamline the comprehensive evaluation of SOC operations, alignment, and maturity.

In SOC-MF, the authors conducted a study of the available literature to develop a SOC CMM maturity model. All elements identified in the literature were subjected to a thorough evaluation for their applicability in real-world SOCs environments through a survey conducted among 16 participating organizations. The collected data was utilized to construct the SOC

capability maturity model, which includes 5 domains (Business, People, Process, Technology, Services) and 25 aspects/components.

For the determination of maturity levels, CMMI v2.0 was used. CMMI v2.0 is a more recent iteration of the CMMI, which was used for the SOC-CMM in [32]. The key distinction between the original CMMI framework and the enhanced version is that the second offers a more up-to-date, efficient, and performance-oriented approach to improve organizational processes, particularly in the context of agile development practices, where projects are divided into stages, with a focus on ongoing collaboration and CI. In addition to the SOC-MF model, the authors provide a table correlating SOC maturity level with capabilities, organizational characteristics, and risk characteristics.

After finishing the self-assessment tool referred to in the article, the scores will be displayed in a graph such as the one shown in Figure 2.5. The graph shows the different SOC domains (Services, Business, People, Processes, and Technology). Two different colors are also presented, a red for the maturity goal and a blue for the maturity scored achieved, and levels from 0 to 5. Although the desired maturity for each SOC domain is level 3, only one domain, the service, is assessed.

Figure 2.5: Capability Maturity Model [30]

For the assessment of the capability levels, despite the model's recent development, it is assessed using a table provided in the authors' work [30]. This can be identified as a problem. Given the rapidly evolving threat landscape, these key points could quickly become obsolete. The lack of content about a wide implementation of such model in real-word scenarios can also be understood as an issue by the fact that no review of the potential of such model is given.

2.1.3 Discussion

To better understand the importance and relevance of the previous presented models for this thesis and their use when creating a adequate solution for this thesis, these will be discussed in this subsection.

The first CMM for SOCs found leveraged general IT frameworks (CoBIT, ITIL, and ISO / IEC 27001) and other industry-accepted maturity models. However, this model had limitations. The model only provided a high-level overview of the SOC services, which was not sufficient to provide a more granular assessment. Furthermore, the services and functionalities originally considered by the model's authors became outdated over time. Thus, this model will not prioritized for use in solution development.

Due to scarce research on SOC capability and maturity models and lack of a widely adopted SOC-specific maturity model, CMMI was considered as a possible model to be used. However, similar to the mentioned IT management frameworks, it was not specifically designed to assess SOC services. Although it provides a structured approach to improvement processes in an organization, it is not a design for SOCs. Therefore, it is still not an ideal option to use in a solution, when assessing the SOC.

These models and frameworks provide. while valuable, lacked in assessing SOC-specific maturity and capabilities. This highlighted the need for a more accurate and up-to-date capability and maturity modal specific for SOCs maturity and capability. In this context, SOC-CMM [32] emerged. This is a widely referenced standard modal [40] and is cited in numerous publications, including [31, 35, 30, 33]. The SOC-CMM model adopts a holistic approach by dividing a SOC into five domains, assessing each in terms of maturity and, in some cases, capabilities. The model provides a user-friendly Excel-based evaluation tool and conveniently maps its results to the NIST Cybersecurity Framework (CSF) 2.0, allowing for easy comparison and demonstrably aligning the SOC's security posture with industry best practices. It has also existed for a number of years and is regularly updated, providing a very solid idea of how a model for SOCs can be developed and maintained. All these comprehensive features make the SOC-CMM a highly suitable choice to latter be used within this thesis solution.

For the SOC-CMM, despite its robustness and numerous positive aspects, limitations were presented. Assessment models like SOC-CMM are designed to identify strengths and weaknesses, thereby revealing what areas need to be improved. However, these models do not aim at providing guidance on how to make the improvements. The authors of [35] addressed this gap by proposing an integration of maturity and capability assessment models, such as the SOC-CMM, with two CI methodologies, the DMAIC and the PDCA.

Given the impact that integrating the CI approach with the CMMs can have in SOCs, this approach is important to have in consideration when designing this thesis solution. However, in the study, the authors used DMAIC at the organization level, meaning that a maturity and capability assessment, of all the SOC's aspects, using the SOC-CMM, was conducted in order to better determine which aspect of the SOC had priority to be improved. While the DMAIC methodology at an organization level appears to be a well-suited approach for the [35] study, given that this thesis mainly addresses content management of SOCs, with emphasis in UCs, using DMAIC at an organizational level, like the authors did, may not be necessary. This means that in case of using this CI approach a solution, all the DMAIC cycle will already be tailored to the specific problem adressed in this thesis.

Other models, such as SOC-AM, were also discussed. The SOC-AM was created based on SOC-CMM, although with more streamlined questions and to be independent from the type of business where the SOC is placed. However, the assessment questions, as far as they could be found, are only present in the paper describing the SOC-AM. So, it does not provide an assessment tool to download and use. The model was also designed to assess

the maturity and not the capabilities of SOCs. In addition to that, it also does not map the results to a standard framework, such as the NIST CSF 2.0. Therefore, the SOC-CMM model still remains as a priority modal to be used, given its benefits.

The last model described was the SOC-MF, largely based on SOC-CMM and using the most recent version of CMMI, the CMMI v2.0, for maturity levels. However, there is little reference to this model, and no tool was found to use this model.

After a evaluation of the available models, the SOC-CMM remained as the most optimal choice for assessing the maturity and capabilities of the SOC. The application of this model will not only provide insight into the current maturity and capability of the aspects and services offered by the SOC, but also facilitate a comparison with a well-established cyber-security framework, the NIST CSF 2.0. However, combining the SOC-CMM with the CI's methodologies could allow for constant improvement to SOC's maturity and capabilities.

2.2 Content Management Frameworks

SOCs play a critical role with the main objective in protecting an organizations data from being stolen by cybercriminals. To improve the effectiveness of SOCs, efficient information management strategies should be adopted. CMFs can be described as a tool to organize and manage SOC content. Using such frameworks could allow for more informed decision making, allowing SOC teams to prioritize threats and implement proactive measures to strengthen the overall security posture, leading to improvements in a SOC maturity and capability.

2.2.1 Research Methodology

For the RQ2, an attempt was made to apply the same research methodology presented in section 2.1.1. The IEEE Xplorer, ACM Digital Library, and ScienceDirect databases were used the first sources of information for CMFs. This initial search revealed that there is limited discussion on CMFs. However, these databases yielded limited results. To broaden the search, Google Scholar was employed using less specific keywords. Despite its expansive database, Google Scholar also provided minimal information on CMFs.

Given this challenge, a even more more broader research using Google search engine was made to gather information about existing CMFs for SOCs. This approach led to findings that will be discussed in the subsequent sections, in order to help in answering to the RQ2. It is important to note that although some of the findings were not directly related to CMFs, they were still saved and are used in this section of the state-of-the-art to provide additional context and better understand the CMFs presented. Some works were also found using the snowball process, by checking the references of the founded papers.

2.2.2 Findings

Through the previous research, it was identified a limited number of CMFs for SOCs. Since most of these CMF use UCs to structure information, the CMFs are considered Use Case Frameworks (UCF). However, before presenting the findings of the research, given the importance that UCs have in the CMFs found, and in this thesis, it is crucial to understand what UCs represent, and their purpose.

A UC is a description of a cyber threat [41, 20]. UCs provide a structured approach to security monitoring (the process of monitoring manifestations of cyber threats), and are also tied with business drivers to show how security monitoring reduces risk in the organization. Through UCs, is possible to create a clearer structure and overview of a SOC information, thus leading to a more maturity and capable SOC.

Scenarios for creating UCs typically derive from concerns that organizations wishes to proactively address. UC can go from high level description (the *modus operandi* of the cyber criminals) to the lowest level (concrete security events in a SOC infrastructure such as exploits, failed logins, etc) [20]. In order to identify and create appropriate UCs, organizations should have a topology map of their existing assets [42]. This approach will provide a clearer perspective of the assets, facilitating the development of UCs.

Despite the importance that UC have, the authors of [5] and [43] emphasize certain issues that can arise associated with UCs:

1. Relying on default UCs: Using pre-defined UCs, that already come with cybersecurity software, can generate too many false alarms, overwhelm SOC teams, and delay the detection of real threats.
2. Lack of clear definitions: Poorly defined UCs can lead to missed threats, false alarms, and ineffective threat management.
3. Insufficient testing: Not testing UCs thoroughly can result in a high number of false positives, wasting time and resources during threat investigations.
4. Neglected fine-tuning: Not optimizing UCs to minimize false alarms and identify genuine threats can lead to compliance issues and ineffective threat management.
5. Lack of communication: UCs that do not align with specific organizational requirements can lead to analysts wasting time chasing irrelevant alerts.
6. Static use case management: Without a defined approach to creating and updating UCs, it is difficult to adapt to emerging threats and add new information for detecting threats.
7. Poor performance monitoring: Not regularly evaluating the performance of use cases can lead to inefficient use cases that do not meet organizational needs.
8. No documentation of use cases: Not documenting use cases makes it challenging for SOC managers to ensure effective use case management, decision-making, and visibility within the SOC team.

To understand cyber attacks and and subsequently to develop UCs, the authors of [44], [45] and [46] emphasize that some CMFs tools can be used. The most common tools are described in [47]:

- **MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK)**[8]: ATT&CK, which has been designed in MITRE since 2013 [48], originated from a project to document and categorize adversary tactics, techniques and procedures (TTPs), after a compromise against Microsoft Windows systems, to improve detection of malicious behavior.

[8] `https://attack.mitre.org/`

MITRE ATT&CK can be defined as a globally accessible knowledge base of adversary tactics, techniques and sub-techniques based from real-world attacks observations [49]. The tactic part represents the high-level objectives that attackers aim to achieve. At present, there are 14 identified tactics. Each tactic is accompanied by a set of techniques, which outline the specific methods/procedures that attackers can use. These techniques are further categorized into sub-techniques, which represent more granular or specific implementations of a technique. Table 2.4 provides a overview of ATT&CK's tactics, a description, and the corresponding number of techniques.

Table 2.4: MITRE ATT&CK Tactics, Tactics description and Number of Techniques associated [50]

Tactics	**Tactic's description**	**Number of techniques**
Reconnaissance	Actively or passively gathering of data that might be used to attack a target.	10
Resource Development	Creating, purchasing, or compromising/stealing resources that can be used to support targeting.	8
Initial Access	Utilization of various entry vectors to establish the initial foothold required to attack targets in the network.	10
Execution	Running malicious code on a local or remote machine.	14
Persistence	Used to maintain access to the system without blocking technologies, such as avoiding changing credentials or using other disruptive methods that could block access.	20
Privilege Escalation	Secure a higher level of authority, such as root, than the one attackers currently have on the network or system.	14
Defense Evasion	Avoid detection.	43
Credential Access	Steal credentials, including passwords and IDs.	17
Discovery	Obtain information about internal networks and systems.	32
Lateral Movement	Connecting to or controlling a network remotely.	9
Collection	Collect information for a purpose.	17
Command and Control	Communicate with the systems within the victim's network.	17
Exfiltration	Steal data from victims' networks.	9
Impact	Influence business operations and security availability and integrity.	14

MITRE ATT&CK framework also provides detailed information on mitigations and detection methods that can be used to counter the attacker's techniques and sub-techniques. Mitigations represent security concepts and classes of technologies that can be used to prevent a technique or sub-technique from being successfully executed. For detections, MITRE ATT&CK specifies data sources, which represent the various subjects/topics of information that can be collected by sensors/logs. Data sources

also include data components, which identify specific properties/values of a data source relevant to detecting a given ATT&CK technique or sub-technique.

- **Lockheed Martin Cyber Kill Chain (CKC)** [9]: This framework was created to document the steps that attackers usually complete to achieve their objective. Lockheed Martin CKC is based on the proposition that stopping one step of the attack, thereby breaking its link, prevents all the attack chain. The model, as said in [49], specifies seven stages/steps that an adversary would follow to achieve their objectives: Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control, Action on objectives. By detecting in each stage an attack is, it is possible to have a more structured and improved defense.

  Although it shares similarities with MITRE ATT&CK, the CKC framework, on its own, does not map the vulnerabilities to its stages of attack. On the contrary, MITRE ATT&CK, which does not propose a specific order for stages of attack, allows for more descriptive documentation of adversarial techniques [51].

Both MITRE ATT&CK and CKC frameworks provide guidance to create and prioritizing UCs based on threats. However, they are not specifically CMFs and do not provide adequate management, tracking, and improvement of UCs. Additionally, they do not allow to prove to stakeholders of the SOC the efficiency in detection and mitigating cyber threats. For this, CMFs, with emphasis on UCF, can then be used.

During the research, the Management, Growth, Metrics, and Assessment (MaGMa), which uses UCs, stood out as the most complete and thorough framework to improve SOCs detection and monitoring of cyber threats while aligning these with business goals [20]. Given the UC focus of MaGMa, the framework also as the name of **MaGMa Use Case Framework** (UCF).

The MaGMa framework structures UCs into three distinct layers, seen in Figure 2.6.

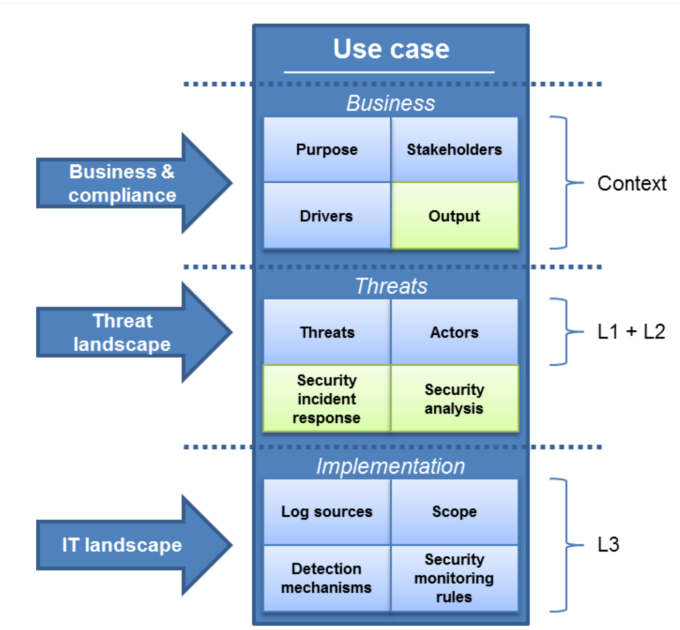


Figure 2.6: MaGMa UCF UC model [20]

[9] https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

The Figure presents the following layers:

1. **Business layer**: This is a strategic layer responsible to describe how the UCs are connected a organization's business needs and vice versa. In this layer, UCs should have the following information:
    - **Purpose**: This describes the relevance of the UC to the business.
    - **Driver**: This represents the main business drivers for the UC (avoiding financial loss, complying to external policies).
    - **Stakeholders**: The UC should be aligned with the stakeholders it impacts.
    - **Output**: The UC should have information regarding the output it provides (security events, security incidents, reports, etc.).
2. **Threat Layer:** This layer focuses on aligning UCs with the processes for managing cyber threats. This layer can be further divided into four key components:
    - **Threats**: This refers to the specific threats that UCs are designed to address. The MaGMA framework categorizes these threats as Level 1 (L1) and Level 2 (L2) UCs. L1 UCs are tailored for high-level threats (e.g., Exfiltration) and represent the steps of CKC (although UCs outside the CKC can be used). L2 UCs presented a more detailed description of UCs threats (e.g., Exfiltration by malware).
    - **Actors**: This identifies the individuals or groups responsible for, or capable of, carrying out the threats.
    - **Security Incident Response**: This component defines the set of actions and specific SOC team roles associated with responding to security incidents. It ensures a coordinated and efficient response to incidents.
    - **Security Analysis**: This provides guidance for SOC analysts in interpreting and making decisions about alerts that might indicate potential threats.
3. **Implementation Layer:** This layer focuses on documenting the technical aspects of implementing UCs. It has four key elements:
    - **Log Sources:** This identifies the specific logs that provide relevant information for detecting the UC's targeted threat.
    - **Scope:** This defines the specific activities or events that the UC should monitor for.
    - **Detection Mechanisms:** This specifies the technologies used to detect threats based on the defined scope and log sources (e.g., log analysis tools).
    - **Monitoring Rules:** These are the detection rules. It is the practical translations of UCs into actionable measures. In MaGMa UCF, the rules are based on the MITRE ATT&CK Matrix for Enterprise[10].

    All these elements of Implementation Layer make the L3 UCs.

[10] `https://attack.mitre.org/matrices/enterprise/`

Besides having structuring method UCs, the MaGMa framework also outlines a methodology to use the tool. The framework can be used in two scenarios. One for UCs that did not existed in a SOC infrastructure and one for UCs that already existed.

For UCs that are not yet implemented in the SOC, a top-down approach should be followed:

1. First, the strategical business layer should be outlined. This will provide the necessary context for the UCs. Outlining the business layer is done by speaking with business stakeholders and determining purpose, business drivers and optionally compliance drivers for the framework's UCs.
2. Then, L1 UCs should be created, based on threat landscape. In this layer the Lockheed Martin CKC can be used to create the UCs. Additional L1 UCs that are specific for the business where the framework is implemented should be added. For example: financial fraud (for the financial sector), Industrial control system sabotage (for energy and utilities sector).
3. Once all L1 UCs were added, L2 UCs can be put into built. An extensive list of 62 L2 UCs is already present in the tool. The list may contain UCs that are not applicable to the organization where the framework is implemented, or may not fully cover your security monitoring requirements. Therefore, it is important to carefully select the proper UCs and extend this list where required and useful. In this phase is also important to include security management stakeholders to ensure proper alignment with organizational risk & security management processes.
4. Lastly, the L3 UCs should be outlined and operationalized. To do it, all of the actual monitoring rules in the SOC should be outlined along with the applicable operational elements (log sources, scope and detection mechanisms) for each of the monitoring rules.

Since most organizations already have an existing security monitoring infrastructure with many UCs, a bottom-up approach should be used to implement the MaGMa UCF. In this case, a bottom-up approach should be used;

1. The first step involves identifying and documenting all detection rules currently running within the detection mechanisms of the SOC infrastructure, along with its respective their scopes, detection mechanism and log sources in the layer L3. This creates the L3 UCs.
2. Once the L3 UCs are documented, these should be mapped to existing UCs in the L2 layer. If a direct mapping isn't possible, a new L2 UC should be created.
3. The final step involves mapping the L2 UCs to the high-level business-oriented UCs (L1 layer), defined in the MaGMa framework. This process establishes a traceability path from L1 UCs to L3 UCs, and vice versa.

After using the MaGMa UCF to document UCs, the framework and its UCs also need continuous management, overtime. For this purpose, a life cycle management for UCs is used. The MaGMa UCF proposes a life cycle management as shown in Figure 2.7.

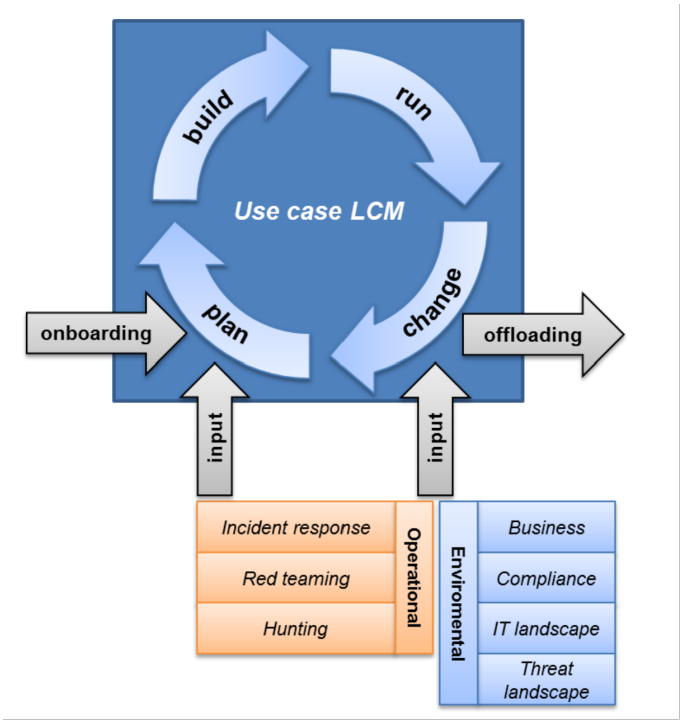


Figure 2.7: MaGMa Use case life cycle management

In this figure:

1. The first phase of the cycle involves planning and building UCs. For this part, the methodologies previous described for adding new UCs or documenting UCs that already existed in the SOC infrastructure should be made used. After documenting the L1, L2 an L3 UCs, the operationalization of the UC should be made, which is done in the next phase.
2. The second phase emphasizes that all operational aspects are implemented and running as part of daily security operations. This stage primarily relies on the documentation from the L3 UCs. The primary elements of this phase include:
   - Integration of Log Sources: All relevant log sources are connected to the security monitoring systems, ensuring they provide the required data.
   - UC Scope Definition and Execution: The scope of each UC is clearly defined and implemented.
   - Security Incident Response: Procedures for handling security incidents related to each UC is known and documented.
   - Roles and Responsibilities: Roles and responsibilities for each UC are formally known and documented.
   - Implementation of Monitoring Rules: Security monitoring rules are implemented, tested, and documented, effectively triggering security alerts as intended.
3. The third phase focuses on adapting the framework to accommodate changes. These changes might include environmental factors (organizational changes) like alterations

in the threat landscape, business adjustments, new or updated regulations, modifications in the IT infrastructure, or operational factors (security operations) such as insights gained from incident response or threat hunting activities that require a reformulation of UCs.

4. The fourth and last phase is the decommission of UCs. This happens when UCs are no longer required and should be removed.

Implementing the MaGMa UCF and having a UC life cycle management can greatly improve the maturity and capabilities of a SOC by organizing the SOC's content and aligning it with the organization's requirements. However, once MaGMa UC life cycle management is well established, the implementation of MaGMa itself should also progress toward a more mature and capable framework.

When considering improvements to the framework's capabilities, there are two main aspects to consider. The first one is the framework's completeness, which involves creating more UCs to increase the level of capability. However, it is important to note that while increasing the number of UCs is important, the thoroughness of each UC is equally essential. The second aspect is the completeness of the UC. This focuses on improving the practical application of a single UC. For example, this could involve increasing the number of log sources or the UC detection rules. Together, both of these aspects contribute to the overall improvement of the framework capability. When considering improvements to the MaGMa UCF maturity, a continuously assessment should be done to align all information gathered from the SOC's daily operations with business needs and improve the completeness of the framework.

To determine the effectiveness of the UC framework in delivering optimal security monitoring, MaGMa UCF also comes equipped with embedded metrics to provide valuable insights into this framework and the security monitoring of a SOC. These are the embedded metrics, control metrics, and output metrics.

The embedded metrics are used to provide steering information regarding the effectiveness of the use case framework. These are:

- **Effectiveness**: the value for this metric is manually set in the framework, for each UC, at the L3 layer. This type of metric is used to indicate the effectiveness of the detection mechanism. For example, a proxy inspecting traffic is much less effective if it is unable to inspect Hypertext transfer protocol secure (HTTPS) traffic.
- **Implementation**: this metric is also set manually at the L3 layer. It indicates how well a detection mechanism has been implemented. For example, the implementation level of an Intrusion Detection Systems (IDS) is much lower if the detection rules this tool are incomplete or have not been improved.
- **Coverage**: just like the previous two metrics, this value is set manually in the framework at the L3 layer. This type of metric allows to indicate the level in which this detection mechanism covers a UC. For example, a UC focused on firewall events has less coverage if not all traffic is routed through the connected firewall.
- **Weight**: this metric is the multiplication of the previous three metrics (effectiveness, implementation, and coverage). It indicates how well is the SOC overall security monitoring.

- **Potential**: this metric is determined by subtracting the Weight metric value from 1 (1 - Weigh metric). This indicates how much improvement can be gained by investing in the effectiveness, coverage, and implementation metric.

The control metrics offer governance details about the framework itself. This includes demonstrating, for instance, how the framework evolves over time, the current maturity level of UCs, and additional information. The control metrics are:

- **Changes to the framework**: This metrics provides information on the number of changes to the framework in a set period of time (for example: 1 quarter). This provides information on the stability of the framework.
- **Growth in number of UCs**: Growth in number of UCs is another change indicator, in this case a change in monitoring scope. This indicator can be used to quantify efforts in extending security monitoring capabilities to the SOC manager.
- **Growth in weight**: In the framework, weight is a calculated value, using the product of effectiveness, implementation and coverage as input (embedded metrics). A higher weight value means that security monitoring is becoming more effective.
- **Changes to potential**: This represents the growth potential of UCs in relation to its effectiveness.

The last type of metrics presented in the MaGMa whitepaper are the output metrics. These metrics are used to understand the efficiency of the SOC security monitoring. The output metrics are the following:

- **Number of alerts**: This metric shows, for each UC, the number of alerts generated by security monitoring systems. It highlights which UCs are frequently activated and which ones are rarely triggered.
- **Number of incidents**: This metric is similar to the previous one but instead of showing the number of alerts for each UC it shows the number of incidents. The main difference between alerts and incidents is that incidents are happen when a particular alert (or series of alerts) has occurred
- **False-positive ratio**: False-positives occur when a security alert is triggered, while there's no actual security incident. This metric provides an indication of the quality of the security monitoring system. For this metric, the calculation proposed in [20] is:

$$100 - 100 * (\frac{\text{number of alerts relation to incident}}{\text{total number of alerts}})$$

- **Number of false-negatives**: False-negatives occur when an actual incident has taken place that is within scope of one of the UCs, but was not detected by any L3 operational monitoring detection mechanisms.

The MaGMa UCF also provides high-level statistics such as:

- Average values for all the embedded metrics.
- The number of operational UCs per threat. The MaGMa UCF tool provides for the measurement of the number of L2 and L3 use cases per L1 use case. Thereby providing insight into the number of UCs assigned to each threat.

- The number of identified business and compliance drivers. This information can be used to show alignment of UCs with business and compliance.
- The number of unique detection mechanisms as well as the number of unique log sources.
- Effectiveness, implementation and coverage per L1 UC.
- Weight and potential per L1 UC.

In addition to these metrics, the same author of MaGMa UCF, along with other authors, created metrics that can be applied to CMFs, such as the MaGMa UCF, offering deeper insights into a SOC's performance, reveal weaknesses, and identify areas for growth. This would provide a clearer understanding, for example to the MaGMa UCF effectiveness and also the SOC's management of security monitoring,detection and UC management. These metrics are presented in [52].

The Table 2.5 presents the relevant metrics based on the context of this thesis. The authors of the metrics also classify each metric on four fields.

- Type: This is the type of metric used. This can be a budget metric, percentage metric, absolute number metric, timing metric, trend metric (increasing or decreasing), and Quality Key Performance Indicator (KPI) or Key Risk Indicator (KRI) metric
- Level: This is the implementation level of the metric. These can be level L1 (basic metrics that should be used in any SOC), level L2 (intermediate metrics that provide a higher level of detail for processes and services within the SOC), and level L3 (advanced metrics provide the highest level of detail).
- Target: The target determines what value the SOC should be aiming for.
- Differentiation: Differentiating within the metric. For example, for managed security service providers (MSSPs), which sell security services to other businesses, some of these metrics (especially in the services domain) are customer-facing, so differentiation per customer can make sense.

Table 2.5: SOC metrics [52]

Type	Metric	Description	Level	Target	Differentiation
Use Case Management					
Percentage	Percentage of use cases reviewed	The percentage of use cases reviewed within a defined period (e.g. last 30 days). This measures the execution of use case life cycle management process.	L2	10%	N/A

Continued on next page

Table 2.5 – continued from previous page

Type	Metric	Description	Level	Target	Differentiation
Absolute number	Number of use cases without alerts	The number of use cases that have not triggered any alerts during the reporting period (e.g. last 30 days). This may be an indicator that the use case may need to be tuned or that there is a problem with the log feed for that use case.	L3	N/A	N/A
Percentage & Trend (increasing)	MITRE ATT&CK Coverage (Techniques)	The percentage of MITRE ATT&CK techniques covered by use cases. This metric helps assess the SOC's preparedness against known attacker tactics and techniques.	L2	Trend: increasing	N/A
Detection engineering					
Percentage & Trend (decreasing)	Percentage of benign positives	The percentage of alerts that fired correctly but were not a security incident. Provides insight into the quality of the rule set.	L1	Trend: decreasing	N/A
Absolute number	Number of rule tunings applied	The number of rules that were tuned to increase their performance. This measures the quantitative output (and workload) of the detection engineering / false-positive reduction process.	L2	N/A	N/A
Absolute number	Number of detections moved to production	The total number of detections moved to production.	L2	Two per week	N/A
Automation engineering					

Continued on next page

Table 2.5 – continued from previous page

Type	Metric	Description	Level	Target	Differentiation
Absolute number & Trend (increasing)	Number of automated playbooks available	The total number of automated playbooks available to SOC analysts to support their analysis tasks.	L1	Trend: increasing	N/A
Absolute number & Trend (increasing)	Number of executed automated playbooks	The total number of automated playbooks executed within the defined period.	L2	Trend: increasing	N/A
Security monitoring					
Absolute number & Trend (increasing)	Monitoring coverage	The number of relevant assets connected to the security monitoring service.	L1	Trend: increasing	Per log source type
Absolute number & Trend (decreasing)	Percentage of alerts without playbook	This measures how many alerts were generated that do not have an associated playbook. These are candidates for playbooks and possible automation.	L1	Trend: decreasing	N/A
Security incident management					
Absolute number & Trend (decreasing)	Number of incidents	The total number of incidents that were identified and handled. While it is not always possible to reduce the number of incidents, monitoring the incident trend is important as it is an indication of the security posture.	L1	Trend: decreasing	Per category Per severity Per kill-chain stage
Absolute number & Trend (decreasing)	Number of affected assets	The total number of assets involved in incidents. Measures the scope of the incidents.	L2	Trend: decreasing	Per vector Per category

Continued on next page

Table 2.5 – continued from previous page

Type	Metric	Description	Level	Target	Differentiation
Percentage	Incidents closed without closure reason	The percentage of incidents that were not closed with a proper closure reason. This measures the effectiveness of the incident closure procedure.	L1	0%	N/A

To still enhance content management in a SOC, the authors of MaGMa UCF authors also created a CMF but specifically tailored for threat hunting. Threat hunting, recognized as a SOC service as depicted in Figure 2.2 within the SOC-CMM domains and aspects, involves actively searching for cyber threats. This practice helps identify anomalies and suspicious activities within an organization's network and systems. By engaging in threat hunting, SOCs can significantly reduce their vulnerability to cyberattacks and protect their key assets. Nevertheless, without a structured method for conducting threat hunting and documenting the results for improved information management, SOCs find it challenging to advance the maturity and effectiveness of this service.

The threat hunting framework, called **MaGMa for threat hunting**, was created to allow SOC teams to define a methodology to perform threat hunting, document their results and have metrics to improve the threat hunting process [53]. Having such a framework for threat hunting may potentially lead to improvement in other SOC services such as security monitoring and incident response. These improvements can then be translated into an increase in capability and maturity of this service and possibly others, and thus an improvement of SOCs.

To provide a structured methodology for thread hunting and better document the results in MaGMa for thread hunting framework, a methodology named Targeted Hunting integrating Threat Intelligence (TaHiTI) is used, as described in [53]. This methodology consists in structured way to combine threat hunting and threat intelligence. This is explained by the fact that threat intelligence, which consists of data collected from various sources about ongoing or potential threats to an organization, is used as a resource for conducting threat hunting investigations and offers additional context throughout the investigation to enhance the hunt.

For the TaHiTI methodology, three elements were taken into account when comparing threat hunting with threat intelligence.

- **Intelligence as a starting point for hunting** - threat intelligence provides information which may lead to threat hunting activities. For example, information about a new type of ransomware that could affect a company's business was used to start a threat hunting .
- **Intelligence for contextualizing and driving the hunt** - context and information from threat intelligence may lead to extending the scope of the hunt, adding new data to the hunt, refining the hunting hypothesis or generating ideas for subsequent hunts. For example, during a threat hunting, a suspicious file is found. Threat intelligence can be used to understand more about the file.

- **Hunting to generate intelligence** - Hunting investigations may uncover unknown information that later can be used to enhance the threat intelligence process. For example, during a threat hunting, a new type of ransomware was found. This information was used to inform other SOCs.

Taking this into account, three phases, seen in Figure 2.8, are used in the TaHiTI methodology:

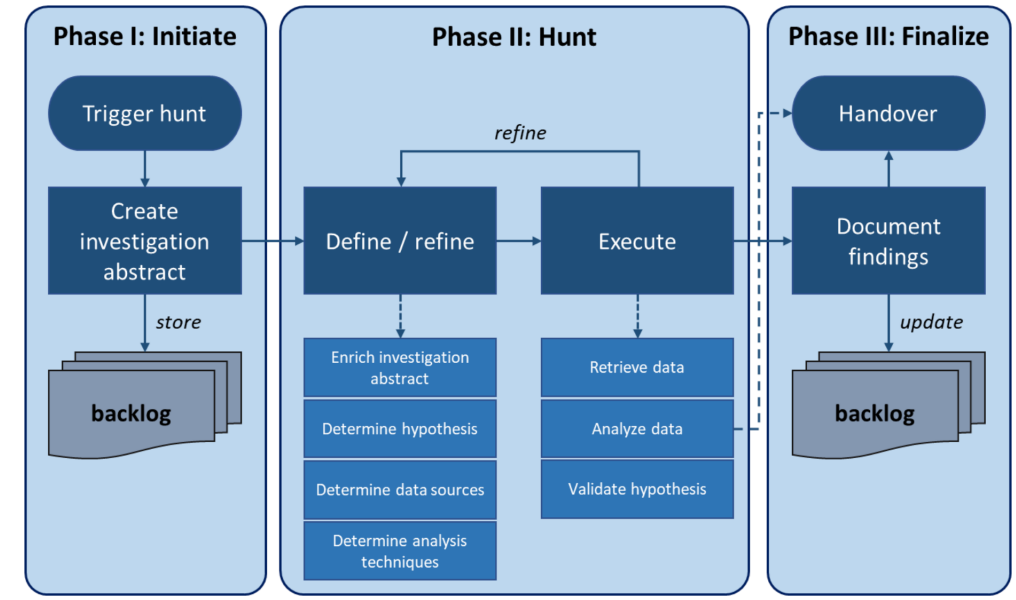


Figure 2.8: TaHiTI process

1. Phase 1 (**Initiate**) - The hunting process begins with an initial trigger. This trigger is converted into a summary that encapsulates the essential details of a potential threat in a clear and organized manner. This summary is then placed in a repository, known as the hunting backlog, where collected potential threats are stored for potential future analysis.
2. Phase 2 (**Hunt**) - This marks the stage where the real investigation occurs. This phase contains two key activities. First, the define/refine activity shapes the hunt from a vague idea into a specific investigation by determining data sources, analysis methods, and, most importantly, formulating a hypothesis to steer the hunt. Then, the execute activity commences, initiating the actual hunt. During this stage, all relevant data is gathered and examined, and the hypothesis is tested. At the conclusion of the investigation, the hypothesis will either be confirmed or refuted.
3. Phase 3 (**Finalize**) - This final phase is the documentation of the results with additional recommendations. After that, the activity is handover to other SOC processes. Potential processes that can receive input from the hunting investigation are the security incident response, security monitoring, threat intelligence, vulnerability management, and others, which may benefit and increase its processes capability and maturity and thus the SOCs capability and maturity.

In order to assess and improve the management of the SOCs thread hunting process, metrics are embedded in the MaGMa for thread hunting framework. These are:

- **The dwell time of the findings**: this metric is the amount of time an attacker was present in a victim network before being are detected.
- **Incident response**: this metric represents the number of incidents triggered during a threat hunting process.
- **Security monitoring**: this metric is the number of added and updated UCs after threat hunting activities.
- **Threat intelligence**: this metric is the new threat intelligence data or mechanisms created during the threat hunting process.
- **Security recommendations**: these are the new preventative measures recommended in the threat hunting reports.
- **Vulnerability management**: this metric is the number of vulnerabilities or misconfigurations uncovered.

Although MaGMa for Thread Hunting can be used separately from other CMFs, combining such a framework with the MaGMa UCF allows more easily integrated threat hunting and security monitoring and detection processes, through UC management [53].

The third CMF framework explored in this chapter is the SimPle and EffectivE Detection (**SPEED**) **UCF**. This framework, similar to previous described CMFs, aims at enhancing SOC information management via UCs. The primary goal of SPEED UCF is to offer a systematic approach for identifying, classifying, and documenting UC.

The creators of the SPEED framework pointed out several issues with cybersecurity systems, mainly with security information and event management (SIEM), which is a cybersecurity tool used to collect, aggregate, and analyze volumes of data across an enterprise. The identified issues are the following:

1. **No room for threat modeling**: Some UCFs do not leave room for threat modelling on specific threat actors and external threat intelligence sources.
2. **No distinction between external and internal threats**: Some UCFs does not make a distinction between internal threats vs external threats.
3. **Out-of-the-box SIEM use cases are split-up while covering same scope**: Some SIEMs come with out-of-the-box UCs that are distributed across different rule content packages. These packages, despite being separated, often share the same detection scope and can be grouped together under a single UC.
4. **No naming conventions or pre-determined directory structure**: Most SIEMs, both with their out-of-the-box and custom-added detection rules, lack a consistent naming convention across all rules.
5. **Most use case approaches have an Attack-centric or quantitative detection bias**: Many SIEM vendors try to structure UCs within a UCF that mainly emphasizes a single constrained threat model (such as the kill chain or MITRE ATT&CK Framework). This approach can miss essential categories like self-monitoring, local anomaly detection (unrelated to attacks), and the differences between quantitative and qualitative threat modeling.
6. **UCs are hard to align with a Security orchestration, automation, and response (SOAR) playbooks platform without a framework**: SIEM systems collect security

data, while SOAR tools automate responses to security threats. Rules should be clear and connected to specific actions (playbooks) that SOAR can understand.

In response to these problems, the SPEED UCF was developed. Its creation involved consideration of various factors:

1. The main focus is on UCs that cover most of the risks in the threat landscape.
2. Keep the framework design simple and straightforward.
3. Create UCs based on threat modeling (identify and enumerate threats in enterprise assets, and create countermeasures) and use existing detection controls.
4. Have a customized approach to allow the framework to be adapted to specific needs (project requirements and stakeholder needs) and incorporating input from various sources.
5. Have in account external threat and internal threats.
6. Have a defense-in-depth view of the internal infrastructure.
7. Deal with external threats using quantitative (based on the specific threat profile of an attacker) and qualitative (based on data of the attack) threat modeling. For this, the MITRE ATT&CK framework is used.

All this was reflected in the final model used by SPEED UCF, seen in Figure 2.9.

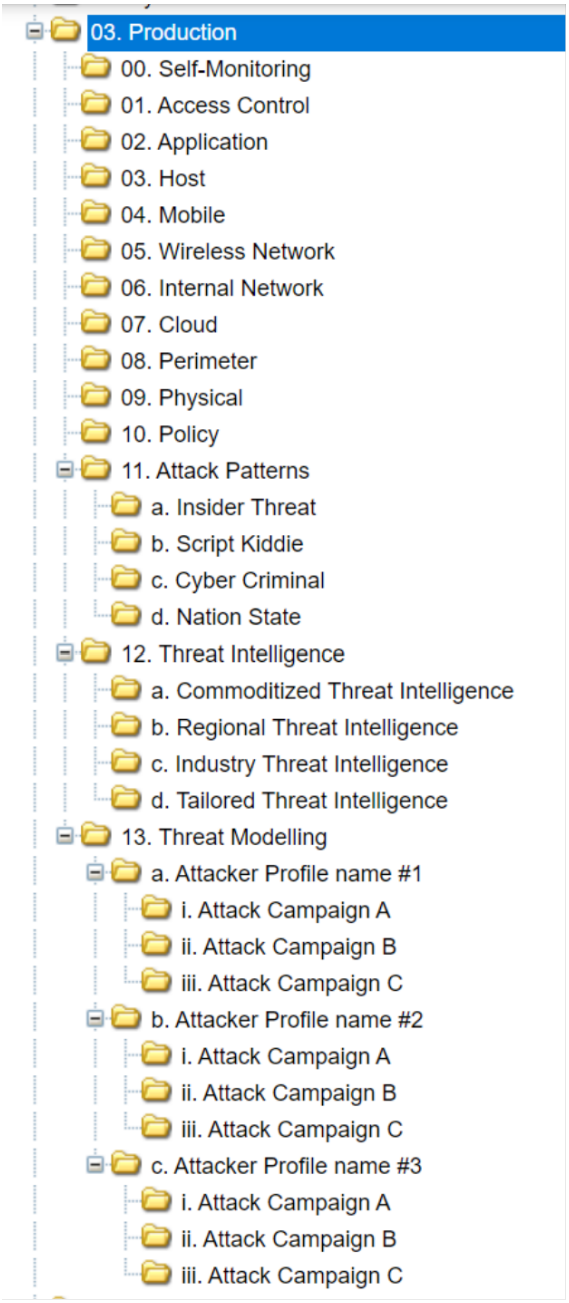


Figure 2.9: SPEED Use Case framework implementation example

The SPEED framework categorizes UC into internal threats using an onion model of infrastructure (Access Control, Application, Host, Mobile, Wireless Network, Internal Network, Cloud, Perimeter, Physical, and Policy). It then separates UCs in quantitative and qualitative threat modeling. Quantitatively, it uses attack patterns (insider threat, script kiddie, cyber criminal, nation state) and threat intelligence (commoditized, regional, industry-specific, tailored). Qualitatively, it profiles an attacker with the corresponding attack campaign.

The last CMF presented is the Detect Tactics, Techniques & Combat Threats (**DeTT&CT**) framework. DeTT&CT leverages the MITRE ATT&CK to manage, assess and compare the quality of a SOC data log sources, visibility and detection coverage, and also to understand threat actor's behaviors [54], leading to a more resilient SOC against cyberattacks. To

employ this framework a Python tool (DeTT&CT CLI), and the DeTT&CT editor can be used. Scoring tables are also provided to have a standardised way of scoring the data quality, visibility and detections coverege. To administrate all the information, DeTT&CT uses YAML Ain't Markup Language (YAML) files.

Data log source are the raw logs or events generated by systems, security appliances, network devices, etc [55]. Having this will allow SOC teams to know if certain cyber threats can be detected or if new detections should be created. These data sources are added and can be administrated in DeTT&CT Editor, which further allows the download of a YAML containing all the information added and administrated. The following information can be recorded in the YAML file [55]:

- The date when the data source was registered in DeTT&CT.
- The data when the data source was connected to the SOC security data lake (repository that stores vast amounts of raw data related to security).
- In which product(s) the data resides (environment, system, service, etc.).
- The type of system(s) the data source applies to (e.g., Windows servers).
- A flag to indicate if the data source can be used in data analytics.
- Data quality to score the quality of data.

The YAML file can then be converted into a JSON file, using the DeTT&CT CLI, so that the JSON file can be used in ATT&CK Navigator (a tool that allows to create a heatmap of defenses against various tactics, techniques and sub-techniques of of MITRE ATT&CK). Using the DETT&CT JSON file in ATT&CK Navigator allows to have a good comparison of the data sources that the SOC has against the data sources of each technique and sub-technique of MITRE ATT&CK, as depicted in Figure 2.10.

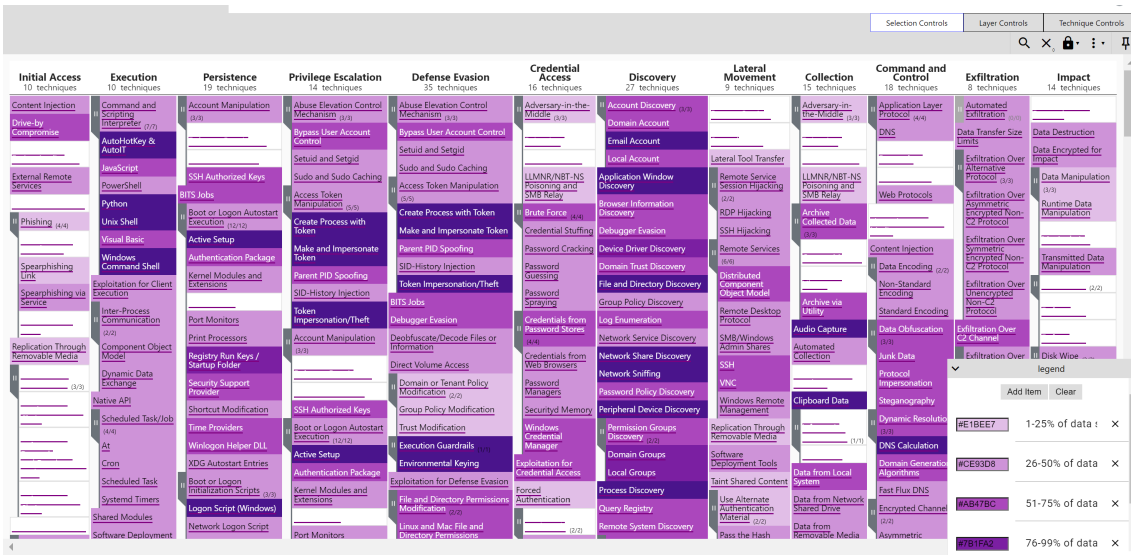


Figure 2.10: Data Sources in ATT&CK Navigator [54]

The generated heatmap presented in the Figure 2.10 shows, in colors, the percentage of data sources available, for each technique and/or sub-technique. The percentage of available data sources, for each technique and/or sub-technique, is determined by diving the number

of data sources added in the DeTT&CT editor, by the number of existing data sources in the MITRE ATT&CK framework. This value is then multiplied by 100 to obtain the percentage.

Each color represents an interval of available data sources. If the available data sources are between 1-25%, then color #E1BEE7 is used. If the available data sources are between 26-50%, then color used is #CE93D8. If it is between 51-75%, then the color used is #AB47BC. The color with value #7B1FA2 is used if the available data sources are between 76-99%. Lastly, if the available data sources is 100% then the color has the #4A148C value.

Visibility coverage within DeTT&CT shows if quality data sources are enough to capture evidence for ATT&CK techniques and sub-techniques. Visibility is necessary to respond to security incidents, carry out hunting investigations, and build detections **??**. Within DeTT&CT, for example in the DeTT&CT editor, it is possible to score the visibility coverage of a SOC per MITRE ATT&CK technique and sub-technique. The visibility scores are based on existing information from the data sources YAML file download, but manual scoring of visibility is required. The visibility coverage information can then be also downloaded as a YAML file and also converted in a JSON file to be used in ATT&CK Navigator.

In DeTT&CT, after having the previous data sources and visibility coverage, a good understanding of detection, the level of detection and the lack of detection that a SOC has should be made. For this it exists the detection coverage, which can also be scored and visualized on the ATT&CK Matrix. Administrating and scoring detection is a manual exercise. This administration is done can be done in the DeTT&CT editor, using the same YAML file as used for visibility coverage.

Additionally, the DeTT&CT framework allows to compare techniques used by threat actors with against the previous mentioned visibility and detection coverage, also by using a JSON file in ATT&CK Matrix to produce a heatmap, to allow SOCs teams to uncover to uncover possible gaps in the SOC and improve.

2.2.3 Discussion

In order to understand CMFs suitability to latter be used in the proposed solution for this thesis, these will be discussed, based on their characteristics.

Among the identified CMFs, the first one that will be discussed is the MaGMa UCF. This framework stands out due to its comprehensive approach to content management, security monitoring, and UC structuring. Several key strengths of this framework can be considered, such as:

- **Structured UC life cycle**: MaGMa UCF methodology facilitates the entire UC life-cycle, encompassing planning, building, operationalization, maintenance, and decommissioning of UCs.
- **Leveraging established frameworks**: This framework integrates well-known frameworks for understanding and defending against cyberattacks. This includes the Cyber Kill Chain, providing a generic overview of typical attack phases of a cyberattack, and the MITRE ATT&CK framework, helping to understand attacker behavior.

- **Business-Implementation Correlation**: The MaGMa framework enables alignment between business needs and the practical implementation of UCs, ensuring that security measures directly address business goals, and vice versa.
- **UC Metrics**: Each UC has built-in metrics, providing valuable insights into the framework's effectiveness. Control metric and output metrics are also presented in [20] to allow to better understand the framework and UCs evolution over time.

However, potential areas for improvement for the MaGMa framework can be considered. Most of the MaGMa UCF embedded metrics are manually entered, which might presents challenges like:

- **Time-Consuming and Inefficiency**: Manually analyzing and recording metrics can significantly decrease productivity. It forces SOC teams to spend valuable time on manual metrics entry, diverting them from their core responsibilities.
- **Error-Prone**: Manual metrics entry is susceptible to human errors, including typos, interpretation mistakes, and accidental omissions. These errors can lead to inaccurate metrics and potentially misleading conclusions.
- **Inconsistent metrics**: If data collection and recording of metrics are performed by different individuals, inconsistencies may arise in how metrics are measured.
- **Difficult to scale**: As the volume of data required for collection increases, the manual collection and management of metrics becomes increasingly challenging.

These challenges highlight the possibility for improvement of the manual metrics of MaGMa framework, which can be achieved through the development of standardized and/or automated metrics. Changing to such approach could not only ensure more consistent in reduced manual effort, more data consistency, minimized human error (if implemented correctly) [56], but also improve the overall process of the SOC which, in the context of the MaGMa framework metrics, could lead to an improvement in security monitoring and detection through UC management. As noted in [20], to quantify the embedded metrics, qualitative categories (e.g., none, low, medium, high, full) can be used. However, given the benefits of automating metrics, this should also be investigated. Also, base on the metrics provided in Table 2.5, increasing the number of metrics that can lead to a better insight into the SOC information management, this leading to its improvement, should also be explored.

The second CMF discussed for this relevance and use in this thesis is the MaGMa for Threat Hunting. This framework has several key strengths that can be considered, such as:

- **Supports the TaHiTI methodology**: This methodology allows for a standardized approach to threat hunting, ensuring consistency and facilitating the threat hunting process.
- **Embedded UC Metrics**: The framework provides several metrics to determine the efficiency and effectiveness of a threat hunting process and show its added value to the organization.
- **Combination with the MaGMa UCF**: Integrating the MaGMa for threat hunting into an environment where MaGMa UCF is already implemented would enhance the coordination between threat hunting and security monitoring, detection, and business

alignment. Furthermore, MaGMa UCF provides the means for reviewing and identifying gaps in security monitoring. These gaps are candidates for hunting investigations, where the MaGMa for threat hunting can be used.

This framework shares similarities with MaGMa UCF in design and comprehensiveness. Given this fact, the MaGMa for threat hunting framework also presents the same problem regarding its embedded metrics. These are also manually entered by the user, which introduces subjectivity and potential errors.

Despite the improvements that could be applied to MaGMa for threat hunting, one of the most important aspects for this thesis is to improve a SOC maturity and capability mainly for in detection and monitoring of cyber threats, rather than threat hunting. Taking this into account, and while it is recognized the potential value that MaGMa for threat hunting could bring to the solution to be developed and including its ability to augment SOC maturity and capability, the integration of this framework would diverge from the project's defined objectives. Such integration would require allocating resources and efforts towards adapting this solution's thesis to fit this threat hunting approach, extending the project timeline and scope, most probably beyond feasibility.

The third CMF framework discussed is SPEED. This framework is similar to MaGMa UCF in its objectives. This means that both frameworks provide structure methods for better organizing information of SOCs, mainly in the context of security monitoring, detection, and business alignment, through UCs. However, MaGMa presents to be a more detailed and mature framework, with not only a UC structure and alignment with MITRE ATT&CK framework, which SPEED UCF also has, but provides several metrics to evaluate UC management, a methodology to add UCs the MaGMa framework, either for UCs that already existed in the SOC infrastructure or not, and a UC life cycle, going from planning and building UCs to their removal. Although SPEED UCF can be a value tool in certain SOCs, for this thesis the most adequate and complete framework that most aligns with the thesis problem and objectives is MaGMa. However, SPEED UCF maps playbooks ((tool used by cybersecurity professionals to identify and respond to security issues) to UCs, which MaGMa UCF does not do. Considering the integration of playbooks in UC structure of MaGMa show be taken in account, allowing for a better information management of SOC.

The fourth and last CMF discussed for its characteristics and possible flaws is the DeTT&CT framework. This framework, has been understood, can be a powerful framework to allow SOCs team to better understand their current detections, visibility and detection coverage, and thus improve overall prevention and detection of cyber threats. For this framework, a problem similar to MaGMa UCF and MaGMa for threat hunting also exists. In the DeTT&CT, visibility and detection coverage is a manual task that has to be done by the SOC team. Having to do such task manual leads to having manual visibility metric and detection metric, which has the already presented problems of having manual metrics.

2.3 Chapter Remarks

In this chapter, a review of the literature was performed in order to answer the research questions RQ1 and RQ2. RQ1 aimed to answer methods to assess capability maturity models for SOCs. RQ2 aimed to answer about which existing frameworks can enhance the management of SOC content through UCs, with an emphasis on monitoring, detection, and business alignment, leading to a SOC maturity and capability improvement.

The research of RQ1 led to the finding that early SOC maturity and capability models relied on standardized frameworks for a broad assessment. However, these types of frameworks were not specifically created for SOC, and although they could be used, they only provided a general overview of a SOC maturity and capability. Recognizing the absence of an in-depth model to evaluate the maturity and capability of SOC, the author of [32] presented the SOC-CMM model, a comprehensive model that provides a comprehensive assessment of SOC's maturity and capability. This model served as the starting point for subsequent studies on existing ways to assess SOCs, such as the SOC-AM, presented in [33] and the SOC-MF discussed in [30]. A CI approach was also proposed in [35] that allowed not only to assess the maturity and capability of SOC, but also to improve it.

Research on RQ2 led to the finding that there is a limited number of CMFs using UCs to improve SOC content management. MaGMa for Threat Hunting can improve SOC capabilities, but is tailored to threat hunting. The RQ2, given the problem and objectives of this thesis, places more emphasis on CMFs that through UCs allow for improvement in SOC monitoring and detection of cyber threats and in business alignment. The SPEED UCF and MaGMa UCF provide those functionalities, however MaGMa UCF stood out as a more comprehensive, holistic, and adequate framework, based on the thesis problem and objectives.

Chapter 3

Design

This chapter outlines the solution's design, influenced by Chapter 2. Initially, the tool's concept and objectives are clarified, followed by an overview of the Software Engineering process, including software requirements and architectural design. Finally, each component of the tool is detailed.

3.1 Conceptualisation

The proposed solution of this thesis is intended to allow SOCs professionals to have a continuous improvement of its aspects (as already mentioned, aspects, in this case, are SOC's functionalities or services), by defining, measuring, analyzing, improving and controlling the defined aspect. Based on the insights obtained from the previous chapter, the solution described here employs two CI methodologies. The first one is DMAIC, which is used at an aspect level. This means that each DMAIC cycle addresses a specific aspect of a SOC. The second methodology, PDCA, is used for planning, implementing and evaluate a solution for the aspect defined in the DMAIC cycle. PDCA is introduces in the Improve phase of DMAIC. To specify, assess, and obtain results of the assessment, the SOC-CMM is used. However, given the thesis's focus on UC management of SOCs, the solution is tailored to this specific aspect of a SOC.

The application follows a sequential approach, and is is divided in five different phases:

1. **Define**: Is identified the problem to be solved.
2. **Measure**: Is conducted a maturity and/or capability assessment based on the defined problem.
3. **Analyze**: Presents the results from the Measure phase.
4. **Improve**: Utilizes the PDCA cycle, subdiving the Improve phase into four sub-phases. This CI methodology is used where to to plan, implement, and evaluate improvements for the SOC aspect defined.
   (a) **Plan**: This phase is for developing a plan to achieve the desired results.
   (b) **Do & Act**: This phase is for implementing the plan. In here, the MaGMa framework, its metrics, the DeTT&CT, and metrics from Table **??** are used. In this Do phase it also also combined with the Act phase because, Act is where improvements of the implemented occurs. Given this, the improvements can be made in the Do phase.

This phase is divided into eight tabs (tabs represent the same phase but different content being displayed):

- One **Overview** Tab: This tab displays an overview of metric values.

- Three **UC Organization** Tabs: This tab is for organizing UCs from, going from their business level to implementation level.

- Three **Business and Compliance Drivers** Tabs: These tabs are for business drivers, internal policies, and external regulators, which are assign to UCs.

- One **Dashboard** Tab: Provides various metrics.

(c) **Check phase**: Evaluates improvements and concerns from the Do phase.

5. **Control phase**: Represents the Check phase of DMAIC, ensuring control over the defined problem.

By continuously monitoring and evaluating progress, the solution ensures a sustained improvement in SOC aspects. The objective pretended with this tool is not only to enhance the SOC detection, monitoring, and business alignment via UCs, but also continuously improving its effectiveness and efficiency, thereby increasing its maturity and capabilities.

3.2 Software Engineering

Software engineering refers to the development, design, testing, implementation and maintenance of software. It allows to produce and manage good software systems and to build better software solutions [57]. The following subsections present phases of software engineering used to design the solution.

3.2.1 Requirements Engineering

Requirements engineering is a critical phase in software development, acting as the foundation for designing systems that satisfy user needs, stakeholder expectations, and the broader operational context. This process involves the identification, documentation, and management of the software's essential functionalities and constraints.

The success of a project heavily relies on effective requirements engineering. Poorly defined requirements can result in project delays and cost overruns, whereas well-executed processes can ensure that the software not only meets but exceeds user expectations.

Non-functional requirements define the expected characteristics of a software system, beyond its specific functionalities. These characteristics are crucial to ensure the overall quality and user experience of the system. They encompass aspects like performance, security, reliability, and usability. The FURPS+ model provides a system for classifying non-functional requirements. This acronym stands for:

- **Functionality**: This covers constraints on the overall capabilities of the system.
- **Usability**: This focuses on the user interface and how easily users can interact with the system.
- **Reliability**: This addresses the system's ability to perform consistently and recover from failures.

- **Performance**: This defines how well the system handles workload and responds to user actions.
- **Supportability**: This outlines the ease of maintaining, troubleshooting, and evolving the system.
- **+:** This encompasses additional non-functional requirements specific to the project.

For this project, the table3.1 presents the non-functional requirements, according to the FURPS+ model.

Table 3.1: Non-functional requirements of the application, according to the FURPS+ model

FURPS+ Category	**Non-functional requirement**
Usability (U)	The application should have intuitive navigation, display of information and provide tooltips and guidance; The application should maintain a consistent look across all phase and functions to minimize user confusion;
Reliability (R)	The application should gracefully handle unexpected errors or interruptions without crashing or losing data.
Performance (P)	Information should load within an acceptable time under normal load conditions.
Supportability (S)	Comprehensive and up-to-date documentation should be provided for users and developers; Easy configuration of different settings and parameters without requiring code changes should exist;
Security (+)	Implementation of mechanisms for user authentication and role-based access control;

Functional requirements, in contrast to the previous requirements, delineate the precise features and functionalities expected from a software system, detailing its intended behavior and capabilities. These requirements are important because they serve as the architectural design for the software's core functions, directing its development, and ensuring its alignment with desired outcomes and user demands. The following represent the functional requirements:

- The tool should allow for a continuous improvement of SOCs.
- The tool should allow to measure the maturity and/or capability of a SOC.
- The tool should map the maturity and/or capability result to NIST CSF 2.0.
- The tool should have a structured method for organizing UCs.
- The tool should be able to map UCs from their business needs and compliance drivers to their implementation level UCs, and vice-versa.
- The tool should allow users to add, change and delete UCs, business drivers and compliance drivers.

- The tool should be able to monitor the maturity and/or capability of a SOC, over time.
- The tool should have relevant metrics, with focus on UC management, security monitoring and detection.
- The tool should allow a mapping of UCs to the MITRE ATT&CK and CKC framework.
- The tool should always be align and up-to-date with the MITRE ATT&CK information.

3.2.2 Architectural Design

After completing the project requirement analysis, the next step is to develop an appropriate solution for the project. Based on the previous requirements and the research finding and discussed in Chapter 2, the suggested architecture for this solution is illustrated, in Figure 3.1.

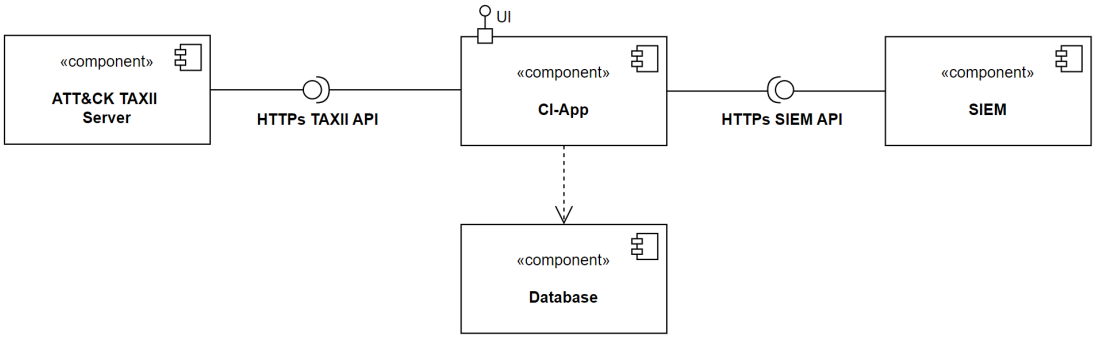


Figure 3.1: Component Diagram of the proposed solution

As can be seen in Figure 3.1, the diagram has four components. Two of them (WebApp and Database) are the proposed solution. The other two (ATT&CK Trusted Automated Exchange of Intelligence Information server and SIEM) are external components used by the WebApp.

Regarding each component:

1. **CI-App**: This is the core of the solution. It consists of a web application connected with a database, a SIEM and the ATT&CK TAXII server. These last two components provide an Application Programming Interface (API), so that CI-App can retrieve the necessary data for its operations. The application also provides a User Interface (UI) to allow users to interact with CI-App. The WebApp communicates with the APIs in a secure way, using Hypertext Transfer Protocol secure (HTTPs).
2. **Database**: This component is responsible for storing and sending information to the CI-App.
3. **ATT&CK TAXII server**: The information in the MITRE ATT&CK framework is accessible through a TAXII server, provided by MITRE itself. TAXII is an application protocol for exchanging Cyber Threat Inteligence (CTI). CTI is a type of data used used to understand a threat actor's motives, targets, and attack behaviors, allowing for a better defense against cyber threats. TAXII exchanges CTI over HTTPS. The

ATT&CK TAXII server provides an API to access to access the MITRE ATT&CK information.

4. **SIEM** : The SIEM provides, via an API, information to the CI-App.

3.3 Proposed Solution

After having the high-level description and design of the proposed solution, along with its requirements, this section will describe in more detail how the CI-App was design, the choices made for designing the application and how it interacts with the rest of the components.

3.3.1 Login

The initial component of the proposed solution is the login. This acts as a normal login, allowing existing users to log in, new users to create an account, and has a password reset mechanism.

The inclusion of login allows the application to have a security layer, by restricting access to authorized users only, aligning with the established functional requirements. Users, by having an account, enable role-based permissions, ensuring granular control over information access. This allows for different levels of visibility and editing privileges, going from limited access to information to full read and write capabilities.

3.3.2 Main Information

After the user logs in, the CI-App presents information about the use of DMAIC throughout the application. This allows users to better understand what information each DMAIC phase contains.

3.3.3 Define

Next, the Define phase is presented. This phase represents the initial stage of the DMAIC methodology, where the user will define the aspect of the SOC (e.g., UC management) to be improved. This phase has the following information:

- **Problem**: This is the problem that is trying to be solved during the DMAIC. In this proposed solution, defining the problem always starts by aligning it with one service and one aspect of a SOC. For example, if the problem is related to UC management, then the Process service and Use Case Management aspect should be specified. Once a service and an aspect are established, the problem is detailed in accordance with the specified aspect.
- **Objectives**: In this part, the target maturity and/or capability, for the selected SOC service and aspect, is documented. If the chosen aspect belongs to the Technology or Services domains both maturity and capability targets can be evaluated. For aspects outside these domains, only the maturity target will be assessed, as capability measurement does not apply. This goes in accordance with the SOC-CMM.
- **Stakeholder**: This section is where it is identified the relevant stakeholders involved in the DMAIC cycle. This can include SOC managers, SOC analysts, the Chief Information Security Officer, and others.

- **SOC Type**: This section is the corresponding type of SOC that is implemented. This can be a hybrid SOC (both in-house and outsourced security professionals), outsourced SOC (is run by a third party, external to the organisation), among other options.
- **Success & Failure Measurements**: This is used to understand how the enhancements implemented throughout the DMAIC cycle will be positive or not. It will ensure that the work performed in the DMAIC is towards the same goals and provide more data-driven results.
- **Budget**: This represents the budget for the DMAIC cycle.

3.3.4 Measure

Following the Define phase, where the relevant project information is documented, the application progresses to the Measure phase. In this stage, the chosen aspect on the Define phase is assessed for maturity and/or capability. For the assessment, the same method and information as in the SOC-CMM Excel tool is used. The questions are presented to be answered. Based on the answer, a guidance phrase is displayed. In addition, remark phrases and comments are also part of each question.

Based on this assessment, a maturity and/or capability result is obtained. This result allows to establish a baseline of the SOC's defined problem and its current maturity and/or capability. To view the result obtained, the next phase, the Analyze phase, should be selected.

3.3.5 Analyze

The Analyze phase presents the results from the previous phase (Measure phase). The information in this phase is the following:

- **Maturity Result**: This is the result of maturity based on the answers from the previous phase, calculated using the method as the SOC-CMM Excel tool.
- **Maturity Target**: This is the maturity target value defined by the user in the Define phase. This value allows to compare the maturity wanted by the SOC against the actual maturity.
- **Capability Result**: If the chosen domain in the Define phase is also measured for capability, then this value also appears in this phase. This value is calculated the same way as the Maturity Result, but only for the questions related to the capability.
- **Capability Target**: This is the capability target value defined by the user in the Define phase, in case it applies to the selected domain.
- **NIST CSF 2.0**: This value represents the mapping of the maturity and/or capability results to the NIST CSF 2.0. This value is also calculated the same way as in the SOC-CMM.

If the maturity and/or capability levels obtained during the Measure phase are below the target values established in the Define phase, it may be pertinent to review the question with the specific responses that resulted in these lower scores. By comparing actual scores with target values, stakeholders can better understand the potential gaps in the SOC. This analysis helps identify root causes and pinpoint areas that require targeted improvement efforts. Such insights are essential for the SOC to achieve its desired level of maturity and capability.

3.3.6 Improve

The Improve phase leverages data gathered from the previous phase to develop and implement solutions that can enhance the defined problem of the SOC. This phase adheres to the PDCA cycle to plan, execute the plan, evaluate the solution's effectiveness, and make necessary adjustments to standardize and stabilize the service.

The PDCA cycle is also iterative, allowing for CI. Once a solution is standardized and stabilized, the cycle can begin repeated, depending on the results. This iterative approach ensures that the solution of PDCA is continually refined and optimized.

Plan

The Plan phase leverages the insights and conclusions obtained from the Analyze phase to create a detailed and strategic plan for enhancement. The Plan phase has the following information:

- **Guidance and Remark Phrases**: For every question in the Measure assessment where the answer values fall short of the predetermined maturity and/or capability set during the Define phase, the corresponding guidance and remark phrases of the Measure phase are presented. For instance, if the maturity and/or capability target was established at 4, then the guidance and remark phrases specifically tailored for questions with answers beneath this threshold must be displayed.
- **Planning**: This represents the plan used throughout the PDCA cycle.
- **Technological Solutions**: This is were the technologies solutions, based on the define plan, are proposed.
- **KPI's**: KPI's that can or will be used to understand the efficiency of plan are documented in this part of the phase.
- **Additional Comments**: Addiction also exist can also be written, to show information that was not adequate for the other parts of the phase but still should be documented.

Do & Act

Following the Plan phase, the Do phase of the PDCA cycle involves the practical implementation of the planned solution. As already mentioned in the conceptualization of the tool, since the Act phase of PDCA involves fixing issues and refining the solution, both the Do and Act phases can be combined into a single section. However, the Act phase is only done after the Check phase.

As this thesis solution is about improving detection, monitoring and business mainly through UCs, this part of the solution leverages the MaGMa UCF and the DeTT&CT frameworks. The solution uses the MaGMa method for structuring UCs, the information that UCs have, along the metrics it provides. The DeTT&CT framework is used to understand the data sources covered by the SOCs.

For the proposed solution in this phase of the PDCA, it starts out with an Overview tab. This tab has metrics and graphs. The metrics are:

- **Average Effectiveness** : This metric is the average value of the Effectiveness metric. It represents the average value of the detection mechanisms of a SOC.

- **Average Implementation**: This metric is the average value of the Implementation metric. It represents the average value of how well the detection mechanism of a SOC have been implemented.
- **Average Coverage**: This represents the average value of the Coverage metric. It represents the average value that the SOC detection mechanism covers.
- **Average Detection**: This represents the average value of the Detection metric. It represents the average value for the SOC detection effectiveness.
- **Average Weight**: This metric is the average value of the Weight metric. It represents the average value for SOC overall security monitoring.
- **Average Potential**: This metric is the average value of the Weight metric.
- **Business Drivers**: This represents the number of different business drivers that exist within the solution. It represents how much improvement can be made to UCs.
- **Compliance Drivers** : This represents the number of different internal security policies and external regulators documented in the solution.
- **L1 Count** : This value is the number of existing UCs in the L1 layer.
- **L2 Count** : This value is the number of existing UCs in the L2 layer.
- **L3 Count** : This value is the number of existing UCs in the L3 layer.
- **Detection Technologies** : This value is the number of different Detection Technologies documented.

Three graphs are also part of this Overview phase. One graph illustrates the effectiveness, implementation, and coverage values for each L1 UC. In this graph, the detection metric value for each L1 UC will also be added. A second graph shows the weight and potential values for each L1 UC. The third graph displays the number of L2 and L3 UCs mapped to the L1 UCs.

After the Overview tab, the solution features two sets of three closely related tabs: business drivers, internal policies, external regulatory (first group), and L1, L2, L3 tabs (second group). Each tab allows for adding, removing, and changing information, to be detailed later.

Starting with the business drivers, internal policies, and external regulatory ones. The business drivers tab records the business driver ID, its name, and a description. The internal policies tab records the internal policy ID, its name, and a description. The external regulators tab records the external regulator ID, its name, and a description. These tabs are designed to be connected to the UCs. ensuring alignment with the organization's specific business needs.

The next tabs are the L1, L2, and L3 tabs. These are used to structure UCs, going from the UCs business needs, which is represented in the L1 tab, to the information about their implementation, in L3, and vice versa. The information for the UCs in each tab is divided in groups to make the information more clear.

In tab L1, the UCs have the following information:

- Regular information for L1 UCs:

- **UC Name**: This is the name for the L1 UC. The name is also used to map UCs in tab L2 and L3 to UCs in this L1 tab.
- **UC Description**: This is the description of the UC. It allows to better understand the UC.
- **Threat Category**: This represents the type of threat that a UC is covering from a high-level perspective. The CKC is used here.

- Information about L1 UCs mapped to other tabs:

- **Number of L2 UC related**: This represents the number of UCs that exist in the L2 tab, that are mapped to each UCs this L1 tab. This mainly helps to have a clear and understandable way to communicate the security strategy to stakeholders and ensures that L2 UCs are justified and linked to L1 UCs.
- **Number of L3 UC related**: This number is similar to Number of L2 UC related, but represents the number of UCs in tab L3 that are mapped to each UC in tab L1.

- L1 UCs business mapping:

- **Purpose**: This is the purpose of a UC. It represents the UC relevance for the business.
- **Stakeholders**: This is used to document the stakeholders of a UC.

- L1 UCs metrics:

- **Effectiveness metric**: This metric is the effectiveness value for each L1 UC.
- **Implementation metric**: This metric is the implementation value for each L1 UC.
- **Coverage metric**: This metric is the coverage value for each L1 UC.
- **Detection metric**: This metric is the detection value for each L1 UC.
- **Weight metric**: This metric is the weight value for each L1 UC.
- **Potential metric**: This metric is the potential value for each L1 UC.

In tab L2, the UCs have the following information:

- Regular information for L2 UCs:

- **UC Name**: This is the name of the L2 UC.
- **UC Description**: This is the description for the L2 UC.

- Information about L2 UCs mapped to other tabs:

- **L1 UC Name**: This is the L1 UC mapped to the L2 UC.
- **Number of L3 UC related**: This metric is the number of UCs in tab L3 that are mapped to each UC in tab L2.

- L2 UCs cyber actors:

- **Actors**: This represents the actors that perform the type of attack that the L2 UC covers.

- L2 UCs business mapping:

- **Business Drivers**: This is used to assign a business driver, from the business driver tab, to the L2 UC.
- **Internal Policy**: This is used to assign an internal policy, from the internal policies tab, to the L2 UC.
- **External Regulators**: This is used to assign the external regulators, from the external regulators tab, to the L2 UC.

- L2 UCs metrics:

- **Effectiveness metric**: This metric is the effectiveness value for each L2 UC.
- **Implementation metric**: This metric is the implementation value for each L2 UC.
- **Coverage metric**: This metric is the coverage value for each L2 UC.
- **Detection metric**: This metric is the detection value for each L2 UC.
- **Weight metric**: This metric is the weight value for each L2 UC.
- **Potential metric**: This metric is the potential value for each L2 UC.

The last tab for organizing UCs, the L3 tab, which focuses on the implementation of UCs, as the following information:

- Regular information for L3 UCs:

- **UC Name**: This is the name of a L3 UC.
- **UC Description**: This is the description for a L3 UC.

- Information about L3 UCs mapped to other tabs:

- **L1 UC Name**: This is the L1 UC mapped to a L3 UC.
- **L2 UC Name**: This is the L2 UC mapped to a L3 UC.

- Information provided by the ATT&CK TAXII server:

- **Tactic**: This is the MITRE ATT&CK tactic associated with a L3 UC.
- **Technique or Sub-technique**: This is the MITRE ATT&CK technique or sub-technique associated with a L3 UC. This gives more insight into attack method of the UC.
- **Technique or Sub-technique ID**: This is the MITRE ATT&CK technique or sub-technique ID associated with a L3 UC. This standardizes attack classification into UCs.
- **Platforms**: This is the MITRE ATT&CK platforms associated with a L3 UC. This allows to indicates the potential affected systems of a L3 UC.

- Information about L3 UCs automation:

- **Playbooks**: This is the SIEM playbooks associated with a L3 UC.

- SIEM information:

- **SIEM Rules**: This is the SIEM rules associated with a L3 UC. Facilitates precise threat identification for L3 UCs

- **Data sources and platforms mapping to Sentinel Rules**: This represents the existing data sources (detections) for the existing platform in the SOC (e.g., Linux servers) and the number of detection rules that exist for each. It provides insight into the detection gaps of the SOC.

- L3 UCs metrics. The last two metrics (Number of alerts and Number of incidents) are the same two output metrics of MaGMa UCF:

- **Effectiveness metric**: This metric is the effectiveness value for each L3 UC.
- **Implementation metric**: This metric is the implementation value for each L3 UC.
- **Coverage metric**: This metric is the coverage value for each L3 UC.
- **Detection metric**: This metric is the detection value for each L3 UC.
- **Weight metric**: This metric is the weight value for each L3 UC.
- **Potential metric**: This metric is the potential value for each L3 UC.
- **Number of alerts**: This metric is the number of security alerts, generated in the SIEM, for the L3 UC, in a certain period of time (e.g., one month). It reflects alert management workload.
- **Number of incidents**: This metric is the number of security incidents, generated in the SIEM, for the L3 UC, in a certain period of time. Indicates threat activity level.

The last tab in this Do & Act phase is the Dashboard tab. This tab consists of several metrics, providing more insight about a SOC information, mainly in UC management. The metrics in this tab are the following:

- Dashboard control metrics (these are the same control metric of MaGMa UCF):

- **Changes to the framework** : This metric is the number of changes that occurred in the L1, L2 or L3 tab, in a set period of time. This means that each time a UC is changed, the metric is updated. The metric allows to understand the stability of the solution, for this Do & Act phase.
- **Growth in the number of UCs** : This metric represents changes in the number of UCs, over time. It is an indicator of change in monitoring scope.
- **Growth in weight**: This metric measures changes in the weight metric over time. An increased weight value indicates enhanced security monitoring.
- **Changes to potential** : This metric is similar to the weight metric but focuses on the potential metric. In case the metric is decreasing, that means that security monitoring is becoming more effective. In case the value is increasing, then security monitoring becomes less effective.

- Dashboard UC Management metrics (these are the UC Management metrics in Table 2.5):

- **Percentage of UCs reviewed** : This metric indicates the percentage of UCs reviewed in a defined period of time (e.g., last three months). It Measures the execution of UC life cycle management process.
- **Number of UCs without alerts**: This metric shows, as an absolute number, the L3 UCs that had no security alerts, in a define period of time. This indicates that a UC may need to be tuned.

- **MITRE ATT&CK coverage** : This metric is the coverage of UCs against the organisations MITRE ATT&CK. This allows SOC to prioritize their security efforts.

- Dashboard Detection engineering metrics (these are the Detection engineering metrics in Table 2.5):

- **Percentage of benign positives**: This metric shows, for each UC, and over time, the percentage of alerts that triggered correctly but were not a security incident. It measures the accuracy of the SOC detection systems.
- **Number of rule tunings applied**: This metric would show, the number of rules that were tuned in the SIEM, in a defined period of time. This metric allows to better understand the rate at which the SIEM detection rules are modified to improve detection accuracy or reduce false positives.
- **Number of detections moved to production**: This metric would show the number of SIEM detection rules that were moved to production, in a period of time defined. This metric emphasizes the efficiency and responsiveness of security teams in deploying SIEM detection rules to enhance threat monitoring.

- Dashboard Automation engineering metrics (these are the Automation engineering metrics in Table 2.5):

- **Number of automated playbooks available**: This metric represents, over time, the available playbooks that exist in the SOC. An increase in this metric indicates more efficiency in the SOC automation.
- **Number of executed automated playbooks**: This metric represent, over time, the number of playbooks executed by the SOC team over a period of time. This metric provides insights into the efficiency, effectiveness, and overall performance of a SOC automation efforts.

- Dashboard Security monitoring metrics (these are the Security monitoring metrics in Table 2.5):

- **Monitoring coverage**: This metric would show the number of relevant assets connected to the security monitoring service. It helps SOC teams better understand and manage their security monitoring scope.
- **Percentage of alerts without playbook**: This metric represent the alerts that do not have associated playbooks. A high percentage for this metric can demonstrate, to SOCs, inefficiencies and inconsistencies in response of security incidents.

- Dashboard Security incident management metrics (these are the Security incident management in Table 2.5):

- **Number of incidents**: This value represents the number of security incidents in the SIEM, in a defined period of time. It helps assess the overall security posture and identify trends in threat activity.
- **Number of affected assets**: This metric would display the number of affected assets related to the security incidents that exist, over a certain period of time. This allows the SOC teams to better measures the impact of security incidents in the SOC protected assets.

- **Incidents closed without closure reason**: This value would be calculated by dividing the incidents that do not have a closed reason by the number of incidents. This highlights potential process gaps in SOC. Both values would come from the SIEM.

- Other metrics. This metrics are not part of Table 2.5 but were also added to this thesis solution since they can lead to an improvement in SOC content management. These are the following:

- **Total number of detection rules**: This value simply shows the number of rules in the SIEM. Its value is supposed to increase over time and reflects detection capability scope of a SOC.
- **Percentage of detection rules reviewed**: This metric displays the number of rules changed over a defined period of time. It validates detection rule's effectiveness and relevance. The SIEM should have this information, in order for the metric to be calculated.
- **Incidents not analyzed**: This metric represents is the security incidents not analyzed in the SIEM. Given insight into the efficiency of incidents handle by SOC team.

Check

The Check phase is the last phase used for the PDCA cycle. In this phase the solution from the Do phase is evaluated, based on the objectives set in the Plan phase. The Check phase involves evaluating the effectiveness of the implemented plan, identifying any deviations or problems, and gathering insights for improvement.

For this solution, it is proposed that this phase has some questions to understand the importance of the solution used in the Do phase, along with possible issues that appear. For these, the questions are presented:

- **Did the implementation achieve the planned objectives?** If the answer is yes, then no comment is necessary. If the answer is no, comments on why the implementation did not achieve the planned objectives should be made.
- **Were there any deviations from the planned actions?** If the answer is no, then no comment is necessary. If the answer is yes, comments on why there were deviations from the planned actions should be made.
- **Did any issues or problems arise during implementation?** If the answer is no, then no comment is necessary. If the answer is yes, comments on what issues appeared during the implementation should be made.
- **Was the solution effective in solving the problem?** If the answer is yes, then no comment is necessary. If the answer is no, comments on why the implementation was not effective in solving the described problem should be made.
- **Are there areas where the process could be improved?** If the answer is no, then no comment is necessary. If the answer is yes, comments on what could be improve in the next PDCA cycle should be made.

After the Check phase, comes that Act phase. The Act phase is used to standardize and stabilize the solution in the Do phase, presented in the Do phase, in case issues were described and documented in this Check phase. In case not, then the solution advances to the Control phase.

3.3.7 Control

The Control phase is the fifth and final phase representing the DMAIC cycle. The main objective in this phase is to guarantee that the maturity and/or capabilities improvements implemented during the DMAIC process are maintained in the long term, preventing a return to former ineffective methods and, if regression occurs, to understand the reasons and timings behind it.

To efficiently oversee the maturity and/or capabilities changes, the phase has a control mechanism that allows to track the maturity and/or capability, over time, of the defined aspect of the SOC, documented in the Define phase.

As part of this solution, users can document the budget expenses incurred during the DMAIC cycle such as costs associated with databases, software, and other essential resources, which are then compared to the original budget set out in the Define phase. Also, Improvements, suggestions, and additional comments are proposed to be documented to that these can be used if the DMAIC cycle needs to be iterated again.

After completion of this phase and the conclusion of the DMAIC cycle, the solution should be documented and the lessons learned throughout the process should be presented to management personnel to share insights and knowledge gained. Finally, the project is formally closed, and the enhanced process is handed over to the process owner for continued management.

3.4 Resume

In the initial section of this chapter, the conceptualization of solution was introduced, demonstrating the objective of the tool and providing a high-level description of it is organized. Then, the focus shifted towards analyzing the solution from a software engineering perspective. Within this chapter, the requirements, encompassing both functional and non-functional aspects, were provided, along with a design of the necessary components for the solution. This initial exploration was imperative to establishing the intended direction for this tool. Subsequently, the chapter transitioned to the detailed description of the propose solution.

Chapter 4

Implementation

Building upon the previous chapter, where the theoretical solution of this thesis was proposed, this chapter presents the practical implementation of the proposed solution for the UC management problem, using a web application, to allow for a more easier implementation of all the proposed aspects of the solution. It represents a implementation of a proof of concept (PoC) into a real-world scenario, in this case in an insurance company.

This chapter begins with an overview of the software tools chosen for implementation. Subsequently, a detailed description of the deployment process for the proposed solution is presented. Finally, the practical implementation specifics are described.

It is important to note that the PoC implemented leverages an existing and deployed infrastructure. While the core principles of the proposed solution are maintained, certain adaptations of the implementation were necessary to accommodate the specific needs of the company and the environment where it was deployed. Also, some implementation options depended on available resources, SOC stakeholders' needs, among other variables.

4.1 Technologies

In this subsection, a brief analysis on technologies to understand the tools that were used throughout this project is presented. Most of the selected one were defined by the SOC team where the POC was implemented, so although other tools could be used, these were not taken in consideration.

4.1.1 Microsoft Sentinel

Microsoft Sentinel, formerly known as Azure Sentinel, is a cloud-based security platform that combines SIEM and Security Orchestration, Automation, and Response (SOAR) functionalities. This solution allows cybersecurity teams with the capabilities for collecting security data, detecting threats, investigating incidents, and automating responses.

The Microsoft Sentinel's SIEM main capabilities provide centralized log collection and querying, enabling advanced correlation and anomaly detection. This translates to proactive threat identification through the creation of alerts and incidents based on security findings.

The SOAR functionality automates incident response workflows triggered by specific cybersecurity events. It also orchestrates various security processes, improving the overall efficiency of a SIEM.

The use of Microsoft Sentinel is due to the fact that the SOC infrastructure where the solution was deployed leverages this technology.

4.1.2 Jupyter Notebook

A Jupyter Notebook is a tool that enables the creation of documents. Such documents can include both programming code in various languages (e.g., Python) as well as text elements like paragraphs, equations, figures, and links. Each Jupyter Notebook comes with a kernel, which is considered a computation engine responsible for executing the code within the document. For instance, the IPython kernel runs Python code in the Jupyter notebook.

The Jupyter Notebook was selected primarily because Microsoft offers an easy method of communication between Microsoft Sentinel and the Jupyter Notebook, facilitating straightforward data extraction from Microsoft Sentinel to the Jupyter Notebook within its cloud portal.

4.1.3 Python

Python is a popular computer programming language often used to build websites and software, automate tasks, and conduct data analysis [58]. It was invented in the late 1980s and has become one of the most popular programming languages in the world, in recent years.

This programming language has several benefits such as having clear and straightforward syntax, making it easier to learn and read compared to many other programming languages. It has an extensive standard library, and numerous third-party packages allow python to be adapted for various tasks. This programming language also has a large and active community, providing ample resources for learning and troubleshooting problems.

In this thesis context, Python will be more tailored to web development. Python offers several frameworks for web development. Commonly used ones include Django and Flask, being the last on, Flask, the one used as a based for the framework used for the construction of the thesis tool, which is Plotly Dash, described next.

4.1.4 Plotly Dash

Dash is a Python framework created by Plotly for interactive web applications. It is designed to facilitate the creation of interactive web-based dashboards and data visualizations. Dash is written on the top of Flask, Plotly.js and React.js. This framework is open-source, and an application built using this framework can be viewed on a web browser. Dash integrates seamlessly with Plotly's graphing library and provides a range of features to support complex, multi-page applications. These features were important for implementation of this thesis tool.

4.1.5 MySQL

MySQL is an open-source relational database management system (RDBMS). This RDBMS uses Structured Query Language (SQL) which is a programming language used to retrieve, update, delete, and otherwise manipulate data in relational databases. A relational database organizes data into one or more data tables in which data may be related to each other.

MySQL was chosen because it is open source and offers good performance, scalability, and ease of use.

4.1.6 DeTT&CT

As previously mentioned, DeTT&CT facilitates SOC teams in evaluating how well their data logging sources align with the MITRE ATT&CK framework. To use DeTT&CT, this one was installed installed in the same VM where the implemented PoC resides. To install DeTT&CT, the following commands were using in the VM termial:

```
git clone https://github.com/rabobank-cdc/DeTTECT.git
cd DeTTECT
pip install -r requirements.txt
python3 dettect.py e
```

Listing 4.1: DeTT&CT's installion.

git clone https://github.com/rabobank-cdc/DeTTECT.git: This command clones the DeTTECT repository from GitHub to the machine machine. It creates a directory named DeTTECT containing all the files from the repository.

cd DeTTECT: This command changes the current directory to the newly cloned DeTTECT directory.

pip install -r requirements.txt: This command installs all the Python dependencies listed in the requirements.txt file. These dependencies are necessary for the project to run correctly.

python3 dettect.py editor: This command runs the dettect.py script with the argument editor, to open the editor in the browser. DeTT&CT editor can also be access online here.

4.2 Deployment Architecture

Deployment diagrams illustrate how the conceptual architecture of software is translated into the physical infrastructure where the software is deployed. These diagrams map software components to hardware nodes and depict communication protocols between them, providing a clear visualization of how the system's software components interact and operate within their execution environment across multiple nodes [59].

To better understand how the solution was deployed in the SOC environment, a deployment diagram is presented in Figure 4.1. The diagram provided aligns with the component diagram, presented in Chapter 3, in Figure 3.1.

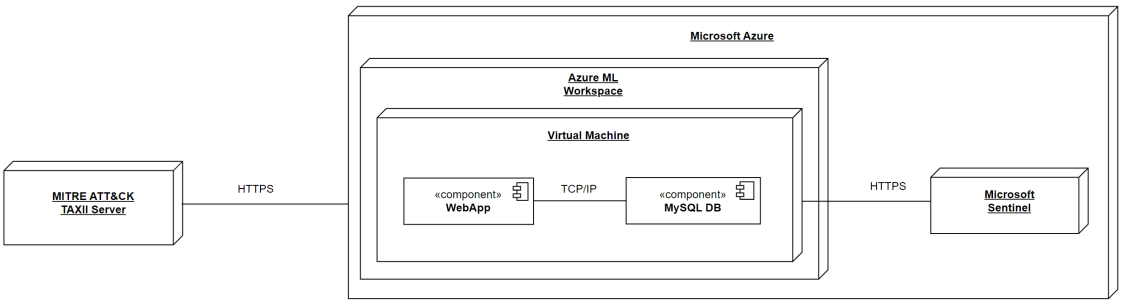


Figure 4.1: Deployment Diagram

In the component diagram, the WebApp, which as previously mentioned, is a web application, represents the implementation of the proposed solution. The WebApp resides within a virtual

machine (VM) set up in an Azure Machine Learning (AML) workspace (virtual environment to collaborate with other workers). In the same VM, the MySQL database is also installed and both communicate via the Transmission Control Protocol/Internet Protocol (TCP/IP). All these components are part of the Microsoft Azure where the Microsoft Sentinel is also present and where the WebApp is able to retrive information through the HTTPs protocol.

The diagram also includes the MITRE ATT&CK TAXII server, which the WebApp uses to retrieve data also by HTTPS.

4.3 Accessing the Application

Deploying a Dash app within a VM of AML was a straightforward process. However, accessing the deployed application needs customization in the VM. As the app operates on the localhost of the AML compute instance, direct access from the AML Studio is not feasible. To overcome this limitation, establishing an secure shell (SSH) tunnel provides a reliable method for connecting to the AML localhost and subsequently viewing the Dash app in a web browser [60].

In order to use the SSH tunnel approach, when the computer instance was created, the SSH configuration access was enabled and a pair of keys were generated (public and private key). With SSH enabled and with the keys generated it was then possible to access the application running on the virtual machine of the AML workspace using the following command in a command-line interface (CLI):

```
ssh -i \path-to-private-key\private-key.pem -L <local-machine-port>:
    localhost:<application-port> azureuser@<remote-machine-public-IP> -p
    <remote-machine-port>
```

Listing 4.2: Command for accessing the application in a web browser.

To better understand this command, each component is explained:

- **ssh**: This is the command to use SSH itself
- **-i path-to-private-key\pramlsockey01.pem**: This specifies the path to the private key file.
- **-L <local-machine-port>:localhost:<application-port>**: This sets up a port forwarding rule. It means that any traffic on <local-machine-port>, which is port of the local machine, will be forwarded to <application-port>, which is the port where the application is running in the VM of AML workspace.
- **azureuser@<remote-machine-public-IP>**: This specifies the username and the public IP address of the VM in AML workspace.
- **-p <remote-machine-port>**: This specifies the port number of the VM in AML workspace.

After running the command with the information to the application and the machine where the application was running, it was only needed to open a browser and navigate to localhost:<local-machine-port>.

4.4 Implementation

This section describes the practical implementation of the solution. The application was implemented as a web application, being each phase of the DMAIC and PDCA a page. In the Do and Act phase of the PDCA, each tab presented the necessary information and when one tab is selected, the other existing tabs hide their information.

It is important to mentioned that, some adaptations when implementing the PoC, compared to the solution proposed. These deviations are documented throughout this chapter. Also, although no software test method (e.g.,unit tests) were performed, every time a change was made in the code, the affected components within the application were always manually test to ensure that the application worked correctly. The test were mainly not done mainly because the Jupiter Notebook is not suited for such software testing, but also because of lack of time during the development of the application. However, it is acknowledgeable the importance of such testings in software development.

The initial component of the proposed solution is the login page. However, given that the MyApp solution is in a VM within the Microsoft Azure ML workspace, Microsoft provides access management to that workspace and to the VM. This means that only authorized individuals of the SOC can access these resources. Given this and the fact that the SOC stakeholders of the SOC did not need a login page, one was not implemented.

4.4.1 Main Page

Since the login page was not implemented, the main page was the first page implemented and presented to the user. The layout of this page can be seen in Figure 4.2.

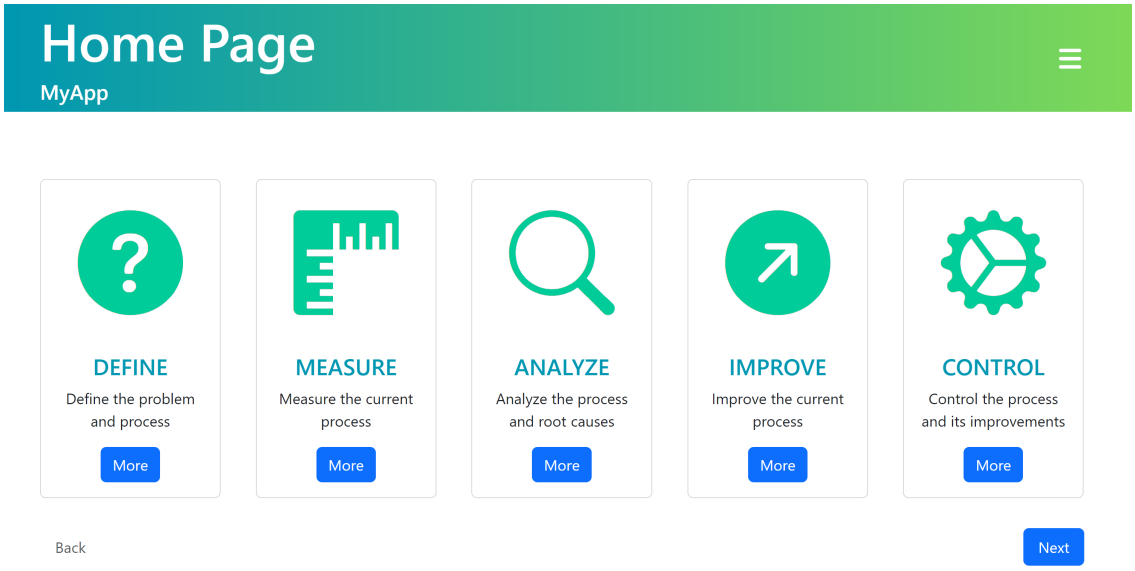


Figure 4.2: Implementation for the Home page.

The user is presented with five containers, each corresponding to each phase of the DMAIC methodology. Each container has a button showing 'More'. The button can be clicked. When a button is clicked, the application uses the ID of the button to present a pop-up showing more information about the phase selected.

Other three buttons are also present. One button on the bottom left of the page named 'Back', one button on the bottom right named 'Next', and one on the top right. The buttons Back and Next are used to go previous or next pages of the solution. In this case, the Back button is in gray given that this is the first page MyApp. The button on top right displays a sidebar in case users want to navigate to different pages without needing to click the Next and Back buttons.

4.4.2 Define

The Define page was the second page implemented. This one, as shown in Figure 4.3, can capture all the necessary information for this page.

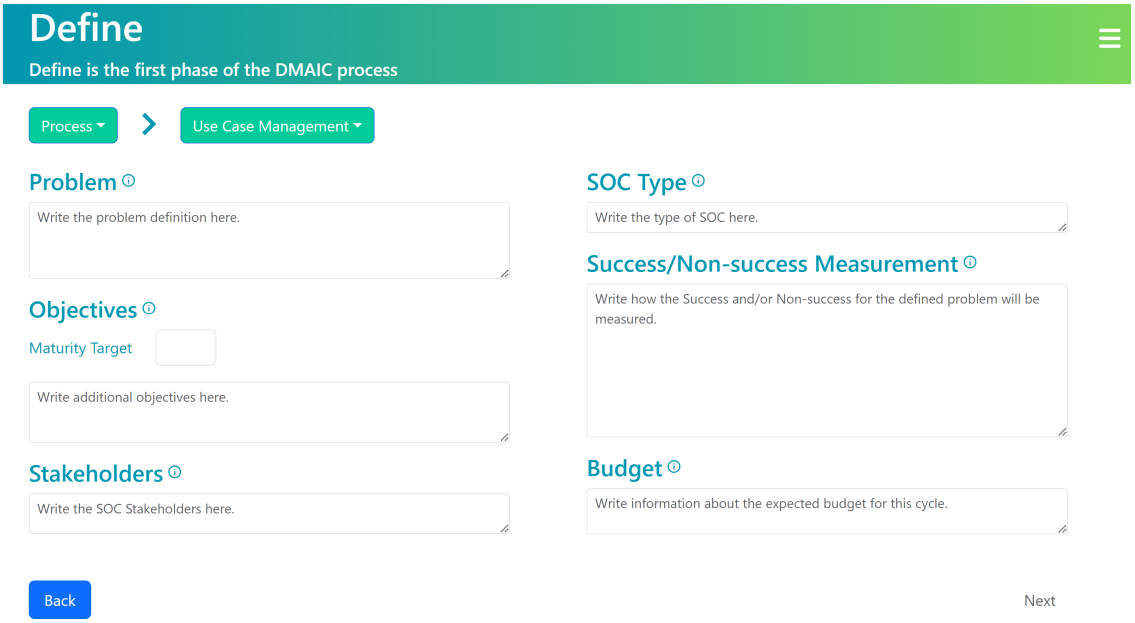


Figure 4.3: Implementation for the Define page.

Two dropdowns were implemented, one for a SOC domain (the left dropdown) and one for a SOC aspect (right dropdown), but since this PoC was design to improve the UC management aspect of the SOC, only the Process domain and the Use Case Management can be selected.

Below the dropdowns is presented the Problem label with a text area, allowing the user to write the problem for the DMAIC cycle. Next to the Problem label is a icon with a question mark, allowing users to better understand what should be placed in the Problem textarea.

After the Problem textarea is presented the Objectives section. In this section is where the maturity and/or capability target are defined, along with the objectives for the DMAIC cycle. Since the UC Management aspect is the one selected then only maturity is evaluated, so only Maturity Target with a box is displayed. The box is used to define the maturity value wanted by the SOC team, in this case for UC Management. The maturity value can only be between one and five. In case a user tries to put a a value inferior to one or superior to five the box color turns to red, prompting the user to select a valid maturity value. In case the user does not understand why the box is red, the question mark icon can be used. This

icon displays information about what the Objectives textarea should have and the interval of valid values for Maturity Target.

In the page it was also implemented textareas for documenting relevant SOC stakeholders, the type of SOC where the solution is implemented, along with the success and non-success measurement and provided budget for this DMAIC cycle.

To maintain data availability, information is stored in a JSON file. On the first entry, a JSON file specific to the page is created, and because each textarea, dropdown and box is identified by an ID, its ID and the associated value and is saved in the JSON file. For subsequent changes, the JSON file is automatically updated, based on the textarea, dropdown and box changed.

Once all the text-areas are filled, the Next button can be clicked so that a user can proceed to the Measure page.

4.4.3 Measure

After the Define page, where the relevant information for the DMAIC cycle can be documented, comes the Measure page. Once the user is on this page, the domain and aspect selected in the previous page (Define page) is retrieved from the JSON file of the Define page. In case the domain and aspect has no value, then a warning message is displayed to one, as seen in Figure 4.4. The same is also shown in the following pages (Analyze, Improve and Control).

Figure 4.4: Warning message to select a domain and aspect in the Measure page.

If both both the domain and aspect have values, then the page shown in Figure **??** is presented.

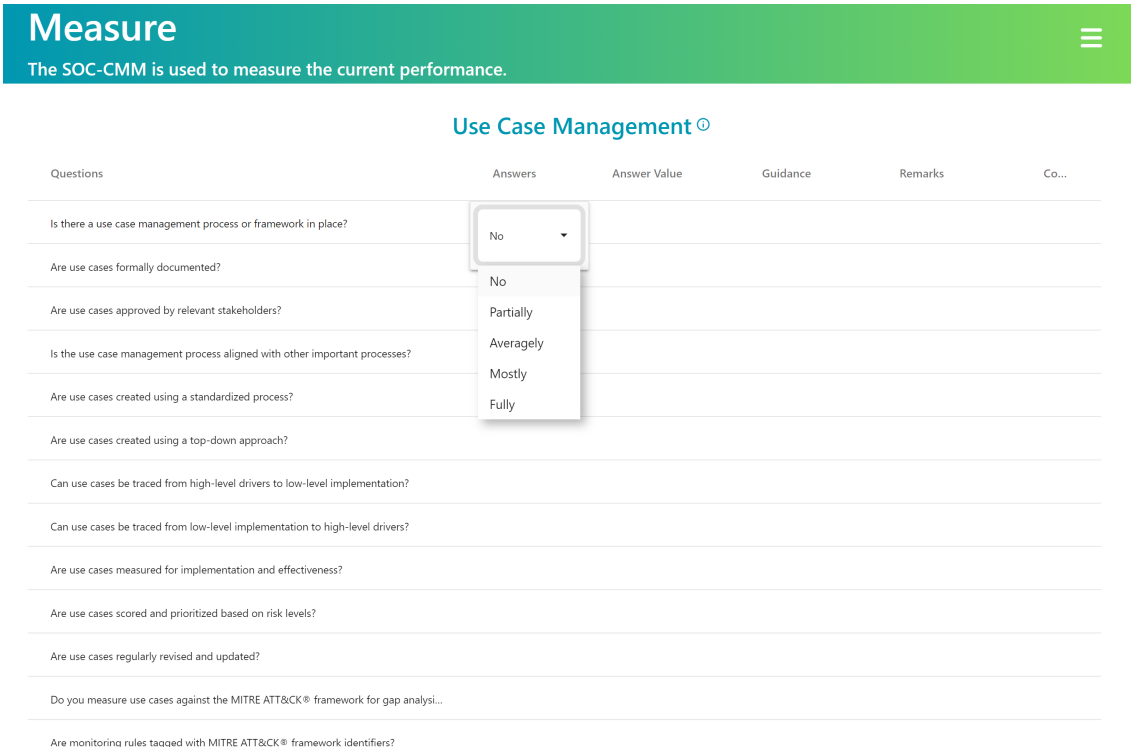


Figure 4.5: Implementation for the Measure page.

In this page, a table is displayed. This table contains the questions, answers, answer values, guidance, remarks, and comments, to measure the maturity. The questions for this table were manually added to the database (although an attempt to use the information from SOC-CMM Excel was tried).

To respond to a question, the user clicks on a cell from the Answers column, which triggers a dropdown menu with possible answers for that question. After the user selects an answer, the database is updated with the provided response, based on the ID of the row. The values in columns 'Answer Value' and 'Guidance', change according to the chosen answer and are also updated in the database.

In this table, given that it takes advantages of the Plotly Dash framework, it is possible to filter and order the rows based on different needs. Once all the questions are answered, it is possible to advance to the next page, the Analyze page.

4.4.4 Analyse

The Analyse page shows the maturity assessment results based on the question answered in the previous page (Measure page). For this implementation, the Analyse page can be seen in Figure 4.6.

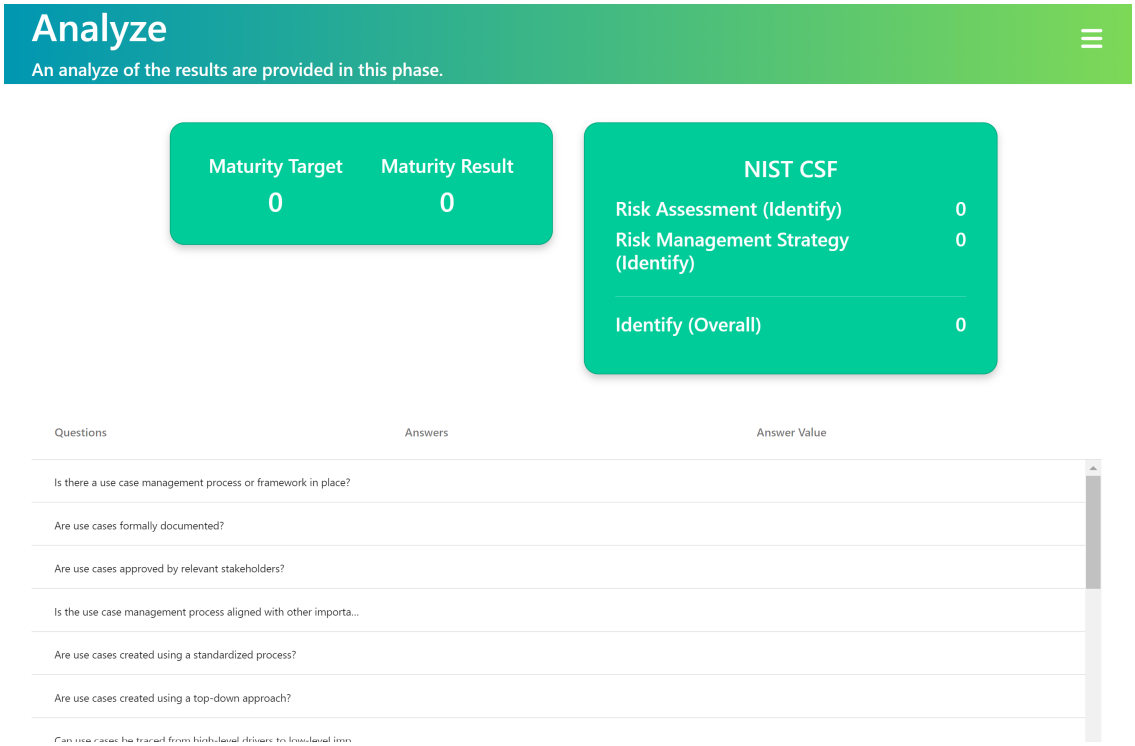


Figure 4.6: Implementation for the Analyze page.

The green container on the left displays two values. The first value is the maturity target, defined in the the Define page, and its value is retrieved from the Define page's JSON file. The second value, the Maturity Result, is calculated based on the answers from the assessment conducted on the Measure page. Maturity Result is calculated using the following equation:

$$(\frac{\text{total score - min score}}{\text{max score - min score}}) * 5$$

total score : represents the sum of the answer's values (e.g., If there are five questions, and and all the questions were answered with Mostly, which as a answer value of 4, the total score would be 5 * 4 = 20).

min score : represents the sum of the minimum possible values that the answers can have, for all the questions (e.g., If there are five questions, and given that the minimum answer value is 1, which is the most of the time the answer No, the minimum score would be 1 * 5 = 5).

max score : represents the sum of the maximum possible values that the answers can have, for all the questions. (e.g., If there are five questions, and given that the maximum answer value is 5, which is the most of the time the answer Fully, the maximum score would be 5 * 5 = 25).

Given that the results range between a minimum value of one and a maximum value of five, the score is then multiplied by five to fit within this interval. Using the formula presented to calculate the maturity result, the result would be $\frac{20-5}{25-5} * 5 = 3.75\%$

The container on the right shows the mapping of the results of the assessment to the NIST CSF 2.0. The answers of the Use Case Management asoect, according to the SOC-CMM, are mapped to the Risk Assessment and Risk Management Strategy categories under the "Identify" function of the NIST CSF. As a result, the calculation for each category and function is performed and displayed. For its calculation the following equation is performed:

NIST CSF 2.0: This value represents the mapping of the maturity and/or capability results to the NIST CSF 2.0. This value is calculated the same way as in the SOC-CMM, with the following equation:

$$(\frac{\text{subcategory maturity total - subcategory maturity min}}{\text{subcategory maturity max - subcategory maturity min}}) * 100$$

NIST CSF 2.0 is organized into functions, categories, and subcategories, as illustrated in A.1. Since no figures with subcategories for NIST CSF 2.0 were found, Figure A.2 presents subcategories for the first version of NIST CSF. This image allows to have a understanding of the subcategories.

Most questions in the SOC-CMM are associated with functions, categories, and subcategories. For example, a question might be linked to one function: Identity function; two categories: Risk Assessment (RA) and Risk Management (RM); and three subcategories ID.RA-1 and ID.RA-2 and RM-1. This is then used to map the results from the Measure phase to the NIST CSF 2.0.

For the calculation of this metric:

subcategory maturity min: Is the sum of the minimum possible value for an answer, for each appearance of a subcategory.

subcategory maturity total: Is the the sum of all answer values, for each appearance of a subcategory.

subcategory maturity max: Is the sum of the maximum answer values, for each appearance of a subcategory.

To better understand this calculation, an example is presented:

A question from the Measure page is mapped to NIST CSF 2.0 for ID.RA-1 and ID.RA-2 subcategories.

ID.RA-1: Appears mapped to one question with an answer value of 3, which represents "Averagely." This indicates that one question is assigned to ID.RA-1. ID.RA-2: Appears mapped to two questions with an answer value of 4, which represents "Mostly." This indicates that two questions are mapped to ID.RA-2.

The minimum value an answer can have = 1 (most of the time is represented by "No") and the maximum value an answer can have = 5 (most of the times is represented by "Fully");

Subcategory Maturity Min for ID.RA: Is calculated as the sum of the minimum possible values for each appearance of a subcategory (1 * minimum value for ID.RA-1) + (2 * minimum value for ID.RA-2) = (1 * 1) + (2 * 1) = 3;

Subcategory Maturity Total for ID.RA: Is calculated as the sum of the actual answer values for each appearance of a subcategory (1 * answer value for ID.RA-1) + (2 * answer value for ID.RA-2) = (1 * 3) + (2 * 4) = 11;

Subcategory Maturity Max for ID.RA: Is calculated as the sum of the maximum possible values for each appearance of a subcategory (1 * maximum value for ID.RA-1) + (2 * maximum value for ID.RA-2) = (1 * 5) + (2 * 5) = 15;

Applying the previous presented equation:

$$(\frac{\text{subcategory maturity total - subcategory maturity min}}{\text{subcategory maturity max - subcategory maturity min}}) * 100$$

the result would be $(\frac{11-3}{15-3}) * 100$ = 66.67%. Once this value is calculated, it is further divided by 100% and multiplied by five (the same way as calculating the maturity result), so that the result is on a scale of zero to five, resulting at the final maturity level for the ID.RA category of 3.37.

The calculation of Identity (the NIST 2.0 function) is summing all the categories values that the Identity function has, which are (ID.RA, ID.AM, ID.IM) and divide by three, which is the number of categories of the Identify function.

Additionally, a similar table as the one displayed in Measure page is presented but, in this case, it only lists the questions, responses, and the values for each response that fall below the defined maturity target maturity level. This would help the user to better understand what question fall bellow the maturity wanted.

4.4.5 Improve

This page of the DMAIC cycle represents the practical application of the PDCA methodology. This page is sub-divided in three other pages. The first one is the Plan page, the second is the Do & Act page and the last one is the Check page.

Plan

The implementation for the Plan page of PDCA can be seen in Figure 4.7.

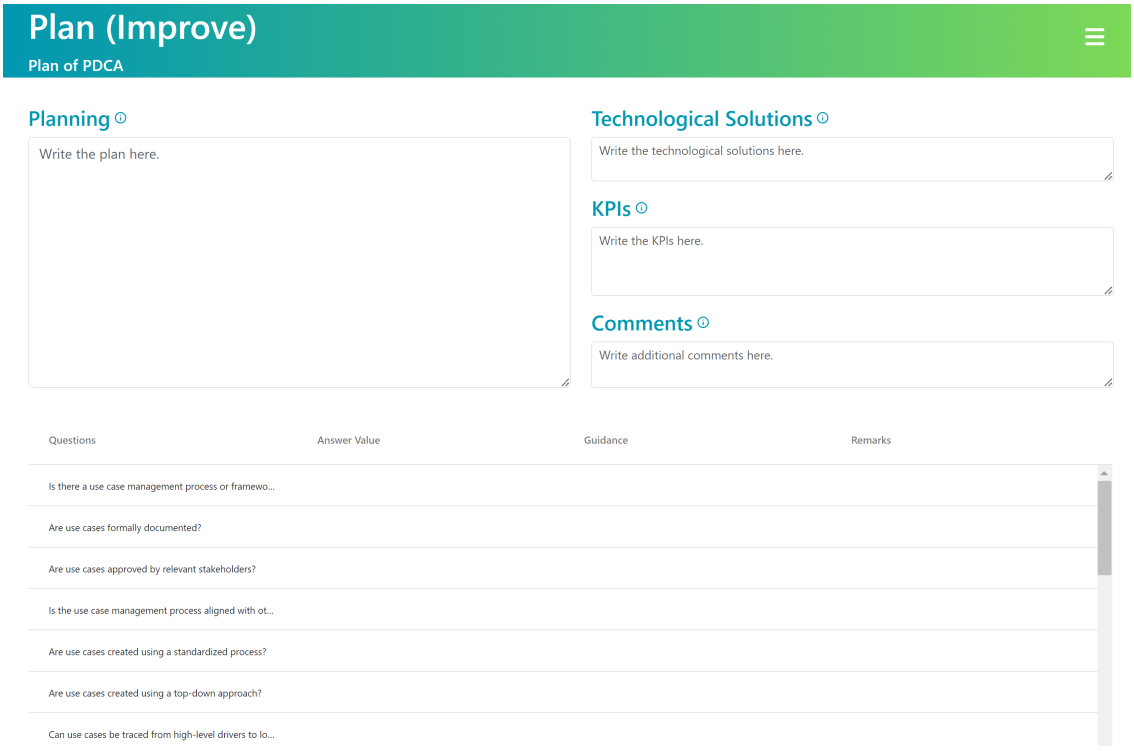


Figure 4.7: Implementation for the Plan page.

In this page, as it can be seen, the plan for this PDCA cycle along with the technologies that can/will be used, and important KPIs to have in account when implementing the plan can be written in the respective textareas. It is also possible to write additional comments. All this information is also stored in a JSON file specific for this page (similar to the Define page but each page have their own JSON file). Similar to the Define page, each component has an ID. When the information of the JSON file is load, the information for each ID is retrieved is displayed. The same with when information is changed. The ID of the component is used to update, in the JSON file, the field that was changed.

Bellow the information written, a table is shown with the questions and the answer values under the define maturity target. The table has also two other columns, the Guidance and the Remarks column, which display the same phrases as in the Measure page, for each of the questions below the defined maturity target.

To present this information, a fetch is made from the database and through hiding columns and a through filtering only the rows with answer values bellow the define maturity target, it was possible to create the presented table. The table was used to help to better plan how the maturity can be improved, for the defined problem.

Do & Act

This page is the implementation of the Do phase of the PDCA cycle. In this page is also where the Act phase will take place, which follows what was proposed in the solution by combining both phases. However, as also mentioned, the Act phase is only done after the Check phase.

The Do & Act page is divided in tabs. Each tab has the specific content proposed in the solution. One tab is for the Overview information, three tabs are for the L1, L2 and L3 layers (using tables), another tab acts a Dashboard, where relevant metrics are implemented, as the rest of the tabs are where information of business drivers, compliance drivers and external regulators information is present (also thought tables).

For the first tab, the Overview tab, its layout can be seen in Figure 4.8.

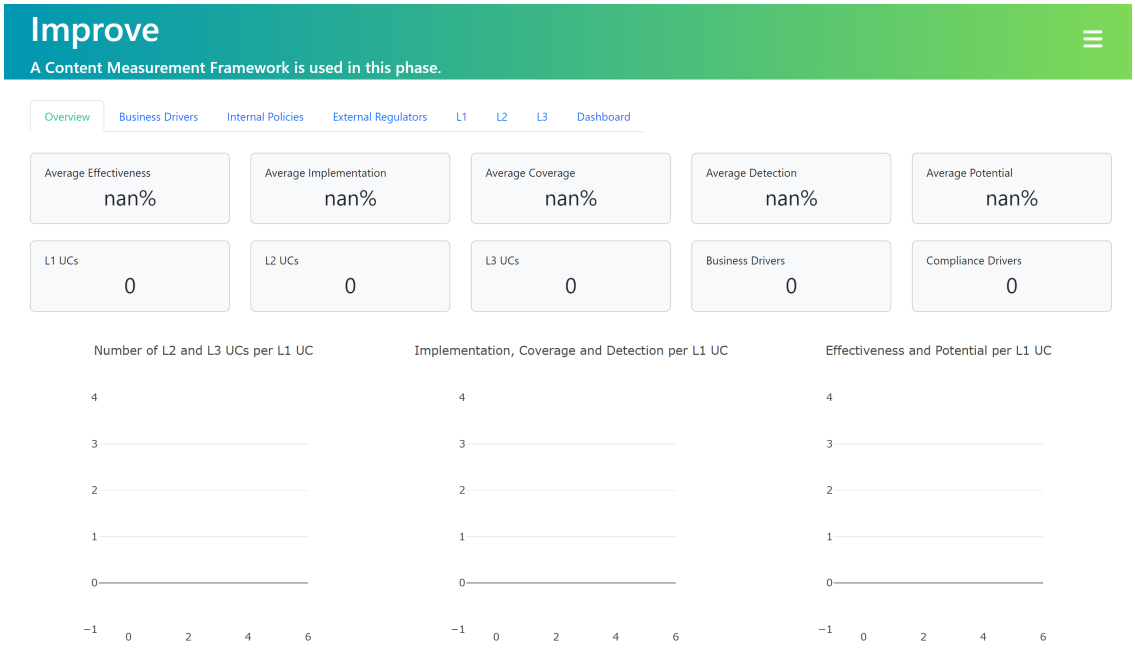


Figure 4.8: Implementation for the Overview page.

It can be seen that the Weight metric, from the proposed solution was not included in the implementation of the solution. In discussions with the SOCs team, because the Effectiveness metric presented significant challenges for automated calculation, two alternatives are proposed for this solution: the metric stays manual or it is removed. However, because it was also observed that the definition of Effectiveness can be the ability to produce desired output [61], it was decided to keep the Effectiveness use it to be the product of the implementation, coverage, and detection metrics. Since the Weight metric also is the product of implementation, coverage, and detection metrics, this one was removed.

In order to have the metrics up-to-date, all the values for these metrics are calculated when the application is first run. The metrics are calculated for all L3 UCs and then assigned to the L2 and L1 layer. Since the Overview page average metrics use the value from the L1 UCs metrics, the values will also be up-to-date.

The average values for Effectiveness, Implementation, Coverage, Detection, and Potential shown on this page are calculated by retrieving all the individual values for each metric from the L1 UCs. Then, the mean of the values associated with each metric is calculated.

The number of UCs for layers L1, L2, and L3, along with the number of business drivers and compliance drivers (the number of internal policies + the number of external regulators) are presented below the average metrics. All the values are calculated by counting the number of rows that exists in the tables of each tab (L1, L2, L3, Business Drivers, Internal Policies,

External Regulators) for the specific tab. For example, the number of L1 UCs is calculated by determining the number of rows that exist in the table in the tab L1.

The page also presents three proposed graphs. The graph on the left displays the number of UCs per L1 UC, illustrating how many L2 and L3 UCs are mapped to each L1 UC. To implement this graph, the columns UC_Name, L2_UC_Related, and L3_UC_Related were extracted. The X-axis represents the data from the UC_Name column, while the Y-axis shows the number of L2 UCs and L3 UCs related to each L1 UC using the L2_UC_Related and L3_UC_Related columns, respectively. The second graph shows the value, as percentage, for the implementation, coverage and detection metrics that each L1 UC has. The last graph is similar to the second one but for the effectiveness and potential metrics. In the proposed solution and in the MaGMa UCF, this last graph would have been for the Weight and Potential metric, but this first one, as already referred, was removed and changed to Effectiveness. The same implementation logic applied in the first graph is applied in both graphs. The columns necessary from the table in the L1 tab are used to create the graphs.

Each time the Overview tab is selected, the average for the metrics, the number of L1, L2, L3, business drivers, compliance drivers, and the graphs are calculate and displayed.

Following the Overview tab, three distinct tabs were implemented : Business Drivers, Internal Policies, and External Regulators. While these tabs share a common structure, each provides essential information about specific aspects of the business. To ensure clarity and organization, each tab was allocated its own space within the database. Each tab features two columns: Name and Description, allowing for detailed categorization and explanation. To avoid redundancy, only the Business Drivers tab is visually presented in Figure 4.9. The other tabs, while functionally similar.

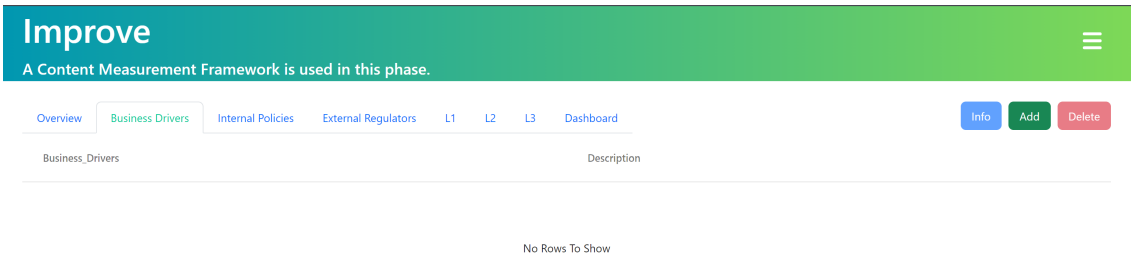


Figure 4.9: Implemented page for Business layer tab.

Finishing the previous business related tab, the following tab is L1. This tab can be seen in Figure 4.10 and presents a L1 UC as an example.

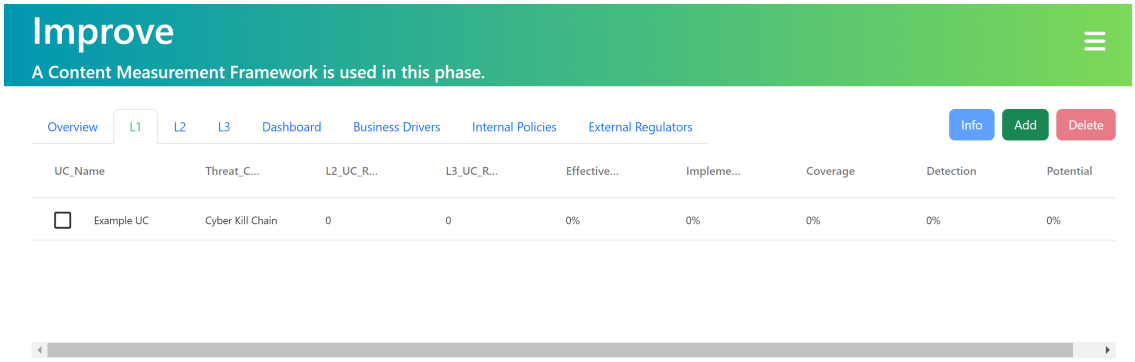


Figure 4.10: Implemented page for the L1 tab.

This table was manually created in a database, similar to how it was done in the next tables for the L2 and L3 tabs. However, since the L1 UCs are mapped to the L2 and L3 UCs and vice versa, the tables were created using a primary key (the IDs of each row) but also using a foreign key.

In this implementation, the table from the L2 tab, which will have the L2 UCs, has a column linking the L2 UCs to the L1 UCs. This column is used as a foreign key referencing the UC_Name column in the L1 tab table. Similarly, in the L3 tab's table, a column mapping the L3 UCs to the L2 UCs is used as a foreign key to the UC_Name column in the L2 table. If it is tried to delete a L1 UC and the UC has L3 and L2 values mapped, it cannot be deleted unless there are no L2 and L3 UCs mapped to the L1 UC. Likewise, if a UC is deleted from the L2 table, if any L3 UC is linked to it, the L2 UC cannot be deleted due to the foreign key constraints. The same rules apply when adding UCs. A new L2 UC must be linked to an L1 UC, and a new L3 UC must be linked to both an L1 and an L2 UC. The implementation of the foreign keys ensured data integrity between the layers, meaning that it guarantees the existence of a value in the primary table.

When clicking on the L1 tab, the information for this layer is retrieved from the database and displayed. This table displays the UCs name, their threat category, the number of L2 and L3 UCs related to each L1 UC, and the its respective metrics (Implementation, Coverage, Effectiveness, Detection and Potential). For demonstration purposes, an example of a L1 UC is provided.

This layer allows to add, remove, change and view more information for UCs. Adding, removing and viewing more information about a UC is done in the buttons on the top right of the page. Whenever a UC is added, a new row is displayed in the table. After this, information for the UC can be added. Only after information is added the row that this one is saved in the database. In case the name for the UC already exists, it is shown an error to the user to change the name for the UC and the UC is not be saved.

To remove UCs, it can be done by selecting one or more rows, in the square box on column UC_Name. After selecting the UC(s), the butted with name Delete can be clicked. This will remove the row(s) from the table and also delete them from the database. However, given to the implementation of table and the use of foreign keys, to delete the UC in the L1 layer, the L1 UC can not have L2 and L3 UCs mapped to it. In case it has L2 and/or L3 UCs mapped, an error is displayed to the uses, prompting to delete the associated UCs before deleting the L1 UC. In case no UC is selected, then the delete button can not be clicked.

To view additional details about a UC, the button Info is used. This feature was implemented not only in this L1 tab but also in the following L2 and L3 tabs, given that on non-function requisite is to have a intuitive navigation and display of information, reducing the volume of displayed information in each table, thereby enhancing user experience and simplifying the interface.

For this implementation detail, certain columns were concealed and its information was moved to a pop-up. To see this hidden information, one UC at a time can selected and then the button Info button can be clicked. This would open a the pop-up, as the one presented in Figure 4.11. In case more then one UC is selected, the Info button can not be clicked.

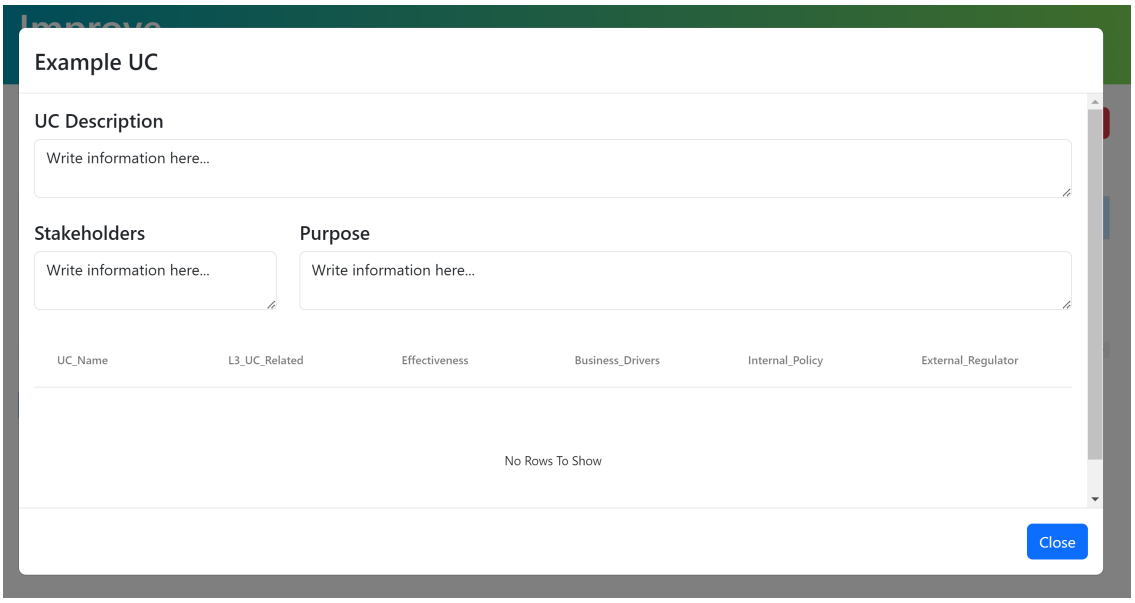


Figure 4.11: Implemented pop-up for the L1 tab.

In this pop-up, the name of the selected UC, a description about the UC, its purpose, and the associated stakeholders can be seen. All this information is retrieved from the database, based on the ID of selected UC , and can be changed (the ID of UCs are not presented in the layout). When any information in a component of the pop-up is changed, each component's unique ID allows for precise updates. Based on the selected UC ID and the changed component's ID, the database can be updated accordingly. The UC ID identifies the specific row, while the component ID pinpoints the exact column to be updated. For example, if the description of the UC is changed, the textarea for the description as an ID associated with this it and updates the UC_Description column, for the selected UC ID, in the database.

Additionally, the pop-up displays a table of L2 UCs associated with the selected L1 UC, along with some of the L2 UCs information. This data is retrieved from the database and shows the L2 UCs mapped to the current L1 UC. The table offers filtering and sorting capabilities, but its values are read-only. This information is supplementary and not part of the L1 tab, only providing additional context for the selected L1 UC

The following tab pertains to tab L2, which adheres to a similar implementation logic as layer L1. This tab is displayed in Figure 4.12 and presents a table with information including the UC name, the mapping of L1 UCs and the count of L3 UCs associated with L2 UCs, also

for a UC used as an example. Additionally, it shows effectiveness, implementation, coverage, potential and the new detection metric, as was done in layer L1. In the proposed solution of previous Chapter 3, this tab has the the Actors information, representing the associated types of attackers. However, this column was removed since the SOC team did not wanted this information.

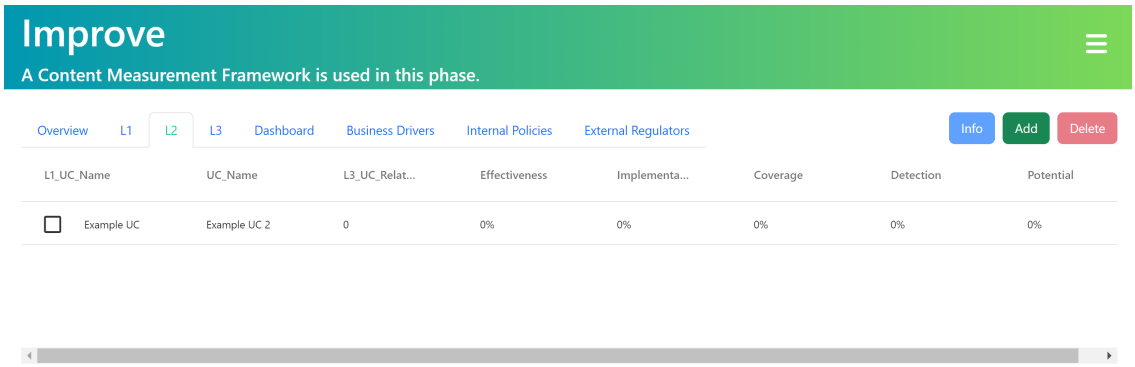


Figure 4.12: Implemented page for the L2 tab.

Each L2 UC, similar to the L1 UCs, has an associated pop-up containing detailed information. By selecting one UC and then clicking the Info button, a pop-up shows with additional information about the selected UC, as seen in Figure 4.13. The pop-up shows information regarding the selected UC's name, a description for the UC and the business driver, internal policy, and external regulator associated with it. The latter three fields are editable when clicking on a dropdown menu, revealing a list of options with the values from the business driver, internal policy, and external regulator tables, present in the Business Drivers, Internal policy and External Regulators tab, respectively.

In addition, the pop-up presents a table listing the L3 UCs associated with the chosen L2 UC, along with some of the L3 UCs information. The information was retrieved from the L3 table in the database, relative to the table for the L3 UC and filtered to only show the L3 UCs that are mapped to the selected L2 UC of the pop-up.

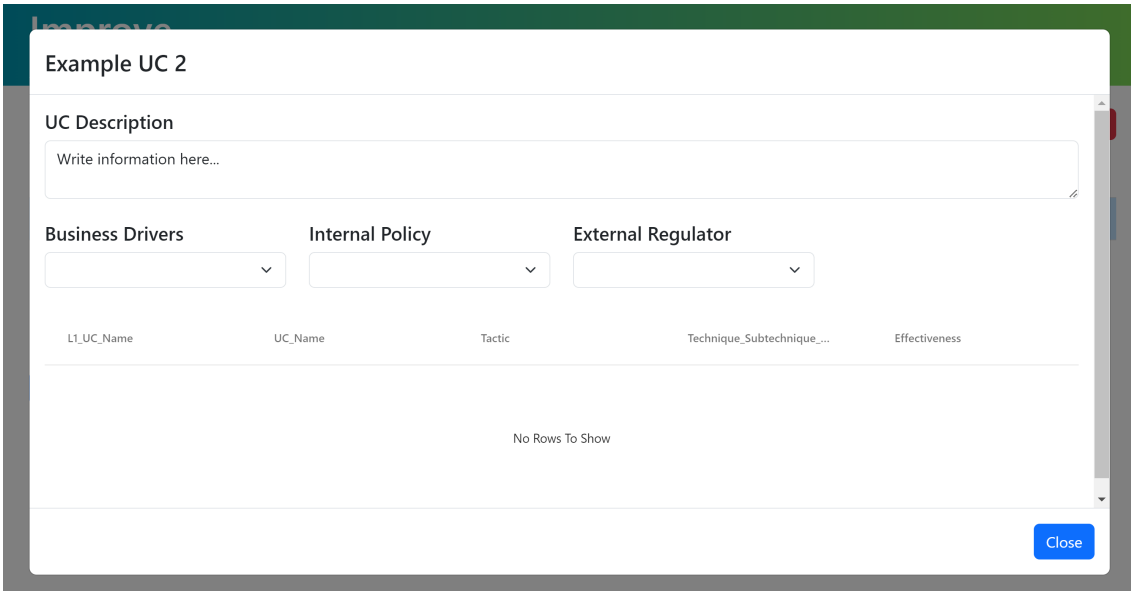


Figure 4.13: Implemented pop-up for the L2 tab.

The final layer, L3, focuses on the implementation aspects of UCs. As also happens in the previous tab L1 and L2, when the L3 tab is clicked, the information in the table for the L3 UCs if retrieved from the database and presented, as can be seen in Figure 4.14. Before describing this tab, it is very importance to take in account the one L3 UCs can have one or more detection rules. Understanding this is a core principal for understanding how this tab works.

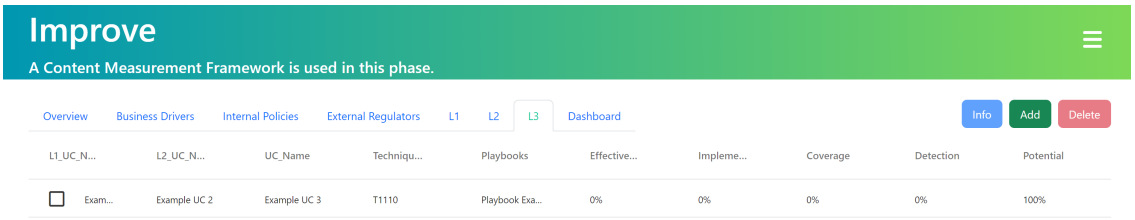


Figure 4.14: Implemented page for the L3 tab.

This table displays the mapping of L3 UCs to L1 and L2 UCs, the name of the L3 UCs, the MITRE ATT&CK technique/sub-technique associated with the L3 UCs, the L3 UCs playbooks (use taken in account the SPEED UCF improvement of Chapter 2), and the effectiveness, implementation, coverage, detection and potential metrics.

In this layer is where the metrics initially obtain their value, enabling the preceding layers (L2 and L1) to obtain metric values as well, since the UCs from L3 are mapped to the L2 layer, and the L2 UCs are further mapped to the L1 layer. In case a metric value for a L3 UC changes, the changes are also made to L2 and L1 UCs, accordingly.

Once a L3 UC is linked to a technique or sub-technique ID, the application retrieves the required information from the ATT&CK TAXII server and Microsoft Sentinel. Relative to Microsoft Sentinel, when a technique or sub-technique ID is associated with a L3 UC, the application uses the Microsoft Sentinel API to extract its detection rules. In Microsoft

Sentinel, the detection rules are mapped to techniques and sub-techniques. Based on this information of the detection rules, the application only filters for the detection rules that have the same technique sub-technique ID specified for the L3 UC. After that, the implementation, coverage, detection, effectiveness, and potential metric are calculated.

Starting with the **implementation** metric, this one is automatically calculated by dividing and using the following values, for each detection rule of an L3 UC:

- **0%** - Detection rule not implemented. For this case, it indicates that a UC does not have any detection rules.
- **25%** - Detection rule created but inactive. The detection rule exists but is inactive, hence not generating any security alerts or security incidents (formed by a one or more security alerts).
- **50%** - Detection rule active but not generating security incidents. The detection rule is able to generate security alerts but not security incidents.
- **75%** - Detection rule can generate both security alerts and incidents but without a structural handle. The rule is capable of generating both alerts and security incidents yet lack a structured approach for effective management and resolution of the generated security incidents.
- **100%** - Detection rule generating alerts and security incidents with systematic management. The rule can produce both alerts and security incidents, and there is an organized method for categorizing, prioritizing, and assigns these incidents. This means that incidents are managed with a dedicated individual or team for responding and handling to security incidents. Example of this can be using a IT Service Management (ITSM) platform (platform where an organization designs, builds, operates, and maintains information technology services). The incidents generated are sent to the ITSM for proper management.

The implementation metric value for a UC would be the mean value of the rules. The implementation of the metric follows the following formula:

$$\frac{\sum(\text{Detection value of each rule})}{\text{Total Number of detection rules}} * 100$$

For example, if a L3 UC has three rules and two of those rules have an implementation value of 25% and the other rule has an implementation value of 75%, this means that the implementation metric value for the UC would be $\frac{25\% + 25\% + 75\%}{3} * 100\% = 41{,}67\%$.

To obtain implementation metric, after adding a technique or sub-technique to a UC, the application calls the Microsoft Sentinel API - Alert Rules, receiving data in JSON format. Then it checks if there are detection rules. In case there aren't, then the implementation value would automatically be 0%. In case there are, each rule is iterated. If the rule has the key 'enable' set to false, then the rule exists but is inactive (not generating any security alerts), so the value of the rule is 25%. If the rule has the key 'enable' set to true but then the key 'createIncident' set to false, then the value for that rule is 50% (it is triggering alerts, but security incidents are not being generated). If the rule has the key 'enable' set to true and the key 'createIncident' also set to true, then it checked if the rule is integrated with the ITSM service. For this, the Microsoft Sentinel API - Automation Rules is called, and the application checks if the detection rule is present in the 'propertyValues' key of the

response. If not, then the metric is not integrated with the ITSM platform and the value of the metric would be 75%. In case it does, the rule value would be 100%. After this is done with all rules of the UC, the mean value is calculated for all the rules, and the value is displayed in the table and added to the the database.

To calculate the **detection** metric, this solution utilizes the DeTT&CT framework. The Data Components (detections) that the SOC should have for the existing platforms (e.g., Azure AD, Linux, Windows) in the SOC environment were added to the DeTT&CT framework editor. This information was then downloaded as a YAML file and converted to a JSON file using the DeTT&CT CLI. The JSON file, which lists all required detections for each platform of the SOC, was placed in the same directory as the application files. Next, the Data Components and platforms that the SOC is actually monitoring were added to each detection rule in Microsoft SIEM, using the same notation format as DeTT&CT. With this information, and after adding the technique or sub-technique to the UC, the detection rules of Microsoft Sentinel for the added technique or sub-technique were retrieved, along with their platform and data component information. The platforms and data components in the DeTT&CT JSON file were also retrieved for the given technique or sub-technique. A comparison was then made between the detections for the platforms that exist in Microsoft Sentinel and the detections for the platforms that should exist in the SIEM, as listed in the JSON file. Based on this comparison, the detection metric value is obtained as the percentage of data components for each platform that exist in Microsoft Sentinel but do not exist in DeTT&CT JSON file. This detection metric is then calculated, displayed in a table, and saved in the database.

Given that this metric can be difficult to understand, an example of how it works is presented. When a UC is associated with a technique/sub-technique, the following information is fetched:

- Information in the DeTT&CT JSON file, created by the SOC team, for the selected technique/sub-technique of the L3 UC:

- **Data Component**: File Creation; **Platform**: Windows, Linux
- **Data Component**: Network Traffic Content; **Platform**: Windows, Linux, macOS
- **Data Component**: Process Creation; **Platform**: Linux, macOS

- Information in the SOC detection rules for the selected, technique/sub-technique of the L3 UC:

- **Data Component**: File Creation; **Platform**: Windows
- **Data Component**: Network Traffic Content; **Platform**: Windows, Linux, macOS
- **Data Component**: Process Creation; **Platform**: Linux

Calculation:

- Total Data Component-Platform Pairs in the JSON file:
  - File Creation: 2 pairs (Windows, Linux)
  - Network Traffic Content: 3 pair (Windows, Linux, macOS)
  - Process Creation: 2 pairs (Linux, macOS)
  - **Total**: 2 + 3 + 2 = 7 pairs

- Total Data Component-Platform Pairs in the Sentinel detection rules:
  - File Creation: 1 pair (Windows)
  - Network Traffic Content: 3 pairs (Windows, Linux, macOS)
  - Process Creation: 1 pair (Linux)
  - **Total**: 1 + 3 + 1 = 5 pairs

The metric value would be:

$$\frac{\text{Covered Data by the SOC Component-Platform Pairs}}{\text{Total Data Component-Platform Pairs}} * 100$$

Changing by the values of the provided example, the value for the metric would then be $\frac{5}{7} * 100\% = 71.42\%$.

The coverage metric is also automatically calculated. Initially, the coverage calculation was intended to be as follows: Each L3 UC would have one or more detection rules. Each detection rule has one or more data components. Each data component has associated platforms (e.g., Windows, Linux).First, the coverage of each data component would be calculated. This is done by summing the number of platforms that the data component is monitoring and dividing by the total number of existing platforms. This calculation is performed for each data component of each detection rule.After calculating this coverage, the mean coverage of each data component for each rule is determined, resulting in the coverage of each detection rule. Finally, the coverage for the UC is the mean of the coverage of the detection rules.

To better understand this metric calculation, and example is presented for a L3 with 2 detection rules. The first detection rule has one detection and the second detection rule has two detections:

First Rule:

-First data component

- **Detection:** Only for Windows platforms.
- **Total Windows machines to monitor:** 100.
- **Windows machines actually monitored:** 50.
- **Coverage Calculation:** $\frac{50}{100} * 100 = 50\%$

Second Rule:

- First data component: For both Windows and Linux platforms.

- **Total machines to monitor:** 100 Windows + 200 Linux = 300.
- **Machines actually monitored:** 80 Windows + 150 Linux = 230.
- **Coverage Calculation:** $\frac{230}{300} * 100\% = 76.67\%$

- Second data component: Only for Windows platforms.

- **Total Windows machines to monitor:** 100.

- **Windows machines actually monitored:** 100.
- **Coverage Calculation:** $\frac{100}{100} * 100\% = 100.00\%$

Overall Coverage for Second Rule: $\frac{76.67\% + 100\%}{2} = 88.34\%$.

UC Coverage Metric: $\frac{88.34\% + 50\%}{2} = 69.17\%$.

However, it can be acknowledged that knowing the number of machine that each data component is monitoring can be very difficult in most SOCs, as it also happened in the SOC where this PoC was implemented. Given this, to still have a automated coverage, a second method for its calculating is proposed.

The second proposed coverage metric is similar to the first one, but instead of a SOC team needing to know the platforms that the detection components are monitoring, in each detection rule, the platforms monitored can be changed to the platforms that sent alerts in a defined period of time. This defined period of time is given by the SOC team as represent a value in each is normal for the platforms of a SOC to send information. If a platform does not send information after the specified period of time, then it is not being covered.

For the implementation of this metric, first it is retrieved the platforms covered by each data components associated with each detection rule. Then, for each platform of each data component of each detection rule, two queries are made to Microsoft Sentinel. One for retrieving the number of platforms that sent alerts in the last seventy-two hours (the value defined by the SOC team) and one for retrieving the number of excising platforms in the SOC, for the platform that is being iterated. Then, for each platform is is calculated the coverage by diving the number of platforms the sent alerts by the number of platforms that exist in the SOC. This is then done for all the platforms of each data component. Then the mean value of each platform for each data component is obtained, making the coverage for each data component. Then, the mean value for each data component is done, obtaining the coverage for each detection rule. Then the mean value of each detection rule is obtained, making the coverage for the L3 UCs.

Also, to better understand this metric, and example is presented?

First Rule:

-First data component

- **Detection:** Only for Windows platforms.
- **Total Windows machines to monitor:** 100.
- **Number of Windows machines that sent alerts in the last 72 hours:** 50.
- **Coverage Calculation:** $\frac{50}{100} * 100 = 50\%$

Second Rule:

- First data component: For both Windows and Linux platforms.

- **Total machines to monitor:** 100 Windows + 200 Linux = 300.
- **Number of Machines that sent alerts in the last 72 hours:** 80 Windows + 150 Linux = 230.

- **Coverage Calculation:** $\frac{230}{300} * 100\% = 76.67\%$

- Second data component: Only for Windows platforms.

- **Total Windows machines to monitor:** 100.
- **Number of Windows machines that sent alerts in the last 72 hours:** 100.
- **Coverage Calculation:** $\frac{100}{100} * 100\% = 100.00\%$

Overall Coverage for Second Rule: $\frac{76.67\% + 100\%}{2} = 88.34\%$.

UC Coverage Metric: $\frac{88.34\% + 50\%}{2} = 69.17\%$.

This information can be easier to retrieve from the SOC detection mechanisms and thus to calculate this coverage metric value

After having the three previous metrics calculated, the **effectiveness** metric is obtained by multiplication of the three metrics. The **potential** metric is then displayed by doing (1 - effectiveness metric value) * 100. Both of these metrics are also saved in the database and displayed in the table.

Following the calculation and display of the previous metrics, since there exists a mapping of the L3 UC to the L2 and L1 UC, the values of the metrics obtained for the added L3 UC are mapped to the L2 and L1 UC layers. This also happens when a UC is deleted. When a L3 UC is deleted, an update on the metric values for the L2 and L1 UCs is performed.

In this layer, additional details about a UC can also be viewed by selecting a UC and then clicking the Info button. A pop-up will appear similar to the one illustrated in Figures 4.15 and 4.16.

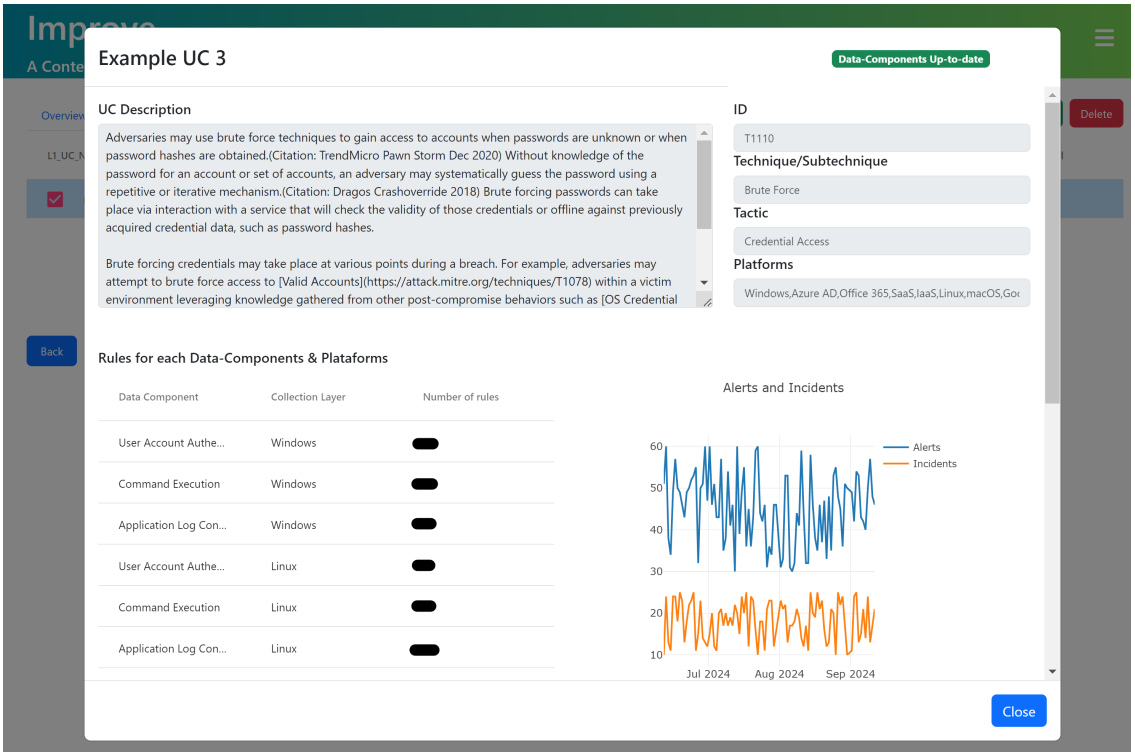


Figure 4.15: Implemented pop-up for L3 UCs (upper part).

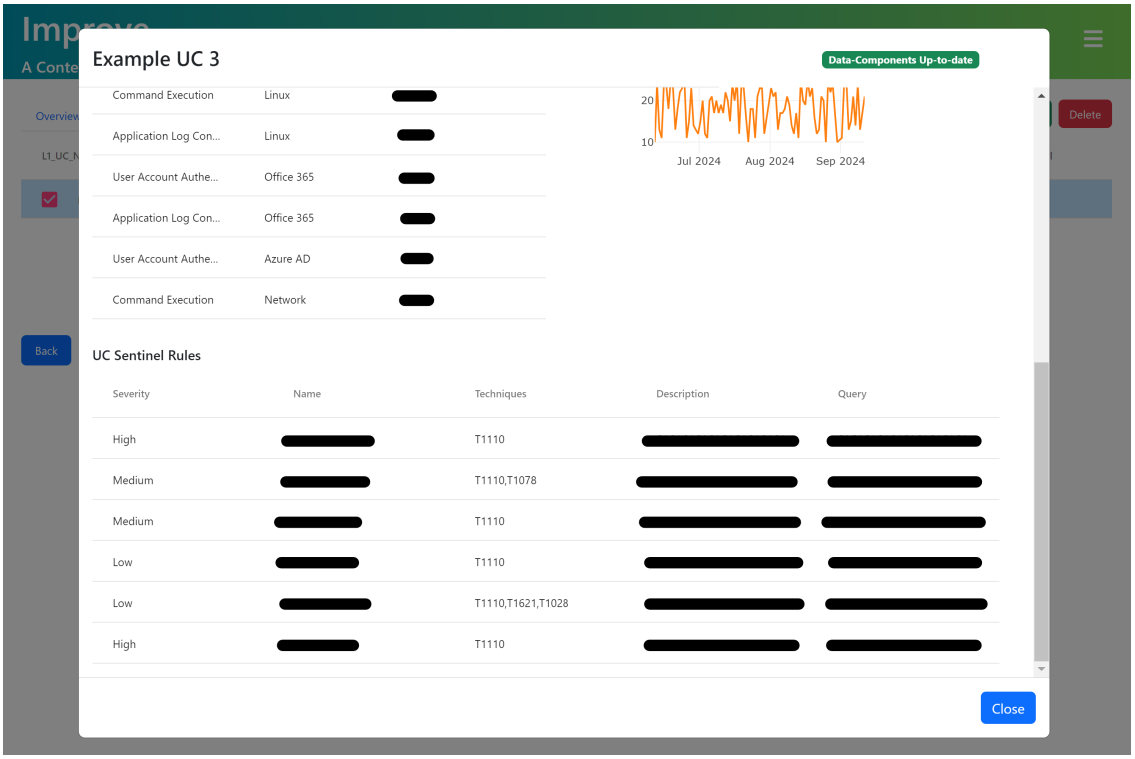


Figure 4.16: Implemented pop-up for L3 UCs (bottom part).

Upon initiating the pop-up, due to the association of the UC with a technique/sub-technique ID, the data corresponding to the UC is fetched from the MITRE TAXII server through its

API. The retrieved data is converted to JSON format (to facilitate better handling of information) and is subsequently filtered to identify the associated technique/sub-technique ID linked to the UC. If no technique/sub-technique ID is located, an error message is presented to the user, indicating the non-existence of the technique/sub-technique ID.

On the top right side of the pop-up, the 'Data-Components Up-to-Date' badge, in green, indicates that the data components defined in the JSON file (created leveraging the DeTT&CT framework), for selected UC, are the same as the Data Components documented in the MITRE ATT&CK framework If a mismatch occurs, the badge turns red and displays 'Data-Components Outdated', For this a simple call use of the MITRE TAXII' API was made to return the data comports that have the same technique/sub-technique as the UC. Next, the data components from the JSON file that are associated with the same technique/sub-technique of the UC are also retrieved. Then, a comparison is made. If the values are not the same, then the SOC team should update the JSON file to ensure detections remain consistent with MITRE ATT&CK.

Below the badge the UC, additional information retrieved form the MITRE TAXII server. The ID, which is the technique/sub-technique ID associated with the UC was already provided by the user, so this information was not fetched from the MITRE TAXII. However, the UC description (which is the same description in the MITRE ATT&CK description for the UC), the technique/sub-technique ID, the technique/sub-technique name, the related tactic of the technique/sub-technique, and the platforms associated with the technique/sub-technique, representing the systems on which an adversary might operate, were retrieved and are all displayed. To allocate this data to specific components of the pop-up, each component is identified by a unique ID. Based on this ID, the corresponding information is appropriately positioned within the right component. For instance, for the Tactic component, which has a specified ID, the tactic related to the technique/sub-technique is retrieved once the API fetches the technique/sub-technique ID and is then placed within the Tactic component.

Below the previous information, on the left side, is a table representing the rules that exist in the Microsoft Sentinel, for the detections (Data Components) in each platform (Platforms). The columns Data Components and Collection Layer are obtained based on the DeTT&CT JSON file and on the information in the SIEM detection rules. In the JSON file, each technique/subtechnique has the data components that should exist for each platform, based on the SOC infrastructure. The SIEM detection rules have the data components that exist exist for each platform in SOC. This information is then used to create these two columns. In this case (as far as can be seen in the image), for the UC with technique ID T1110, the Microsoft Sentinel should have at least one detection rules for User Account Authentication for Windows and Linux, Command Execution for Windows and Linux, and Application Log Content for Windows and Linux. The column on the right, named 'Number of Rules', shows the number of rules that exists in the Microsoft Sentinel for these detections and platforms. However, the number of rules had to be obfuscated for privacy reasons, since it would demonstrate the detections for the platforms the the SOC is not monitoring.

In the pop-up, a line graph that displays that number of alerts and incidents that occurred in the last ninety days (this value was wanted by the SOC team). This graph represents the implementation of two of the proposed metric of Number of alerts and Number of incidents output metrics. For this, a query using the QueryProvider from the Microsoft Threat Intelligence Center Python (MSTICPymsticpy) package, was made to the Microsoft Sentinel, allowing to obtain a table for security alerts and one table security incidents. In one column of the tables is the number of alerts/incidents (depending on the table) and

in the other column column a timestamp of last ninety days, grouped by one-day intervals, that have the same technique/subtechnique as the UC of the pop-up. After retrieving this information, two columns were return, Based on the information received, the graphs were created.

The output metrics were presented in the solution chapter as being specific to each UC and were implemented accordingly. However, two of the output metrics, the 'False-positive ratio' and the 'Number of false-negatives' metrics were not implemented. Although these metrics are part of the proposed solution, they were not included because, in some SOCs (including the SOC where the PoC was implemented), teams are responsible for confirming whether a triggered security alert that escalated into a security incident was indeed a false-positive or false-negative. In the SOC infrastructure where this solution was implemented, these metrics could theoretically be calculated based on information available in the ITSM platform integrated with the Microsoft Sentinel. However, due to the current state of the ITSM integration with Microsoft Sentinel, it is not possible to obtain the necessary information to calculate these metrics. Further developments is required to obtain the information needed to calculate the metrics.

The last component of the pop-up is a table with the detection rules that belong to the selected UC. To display this table, the Sentinel detection rules that have the same technique/sub-technique ID as the UC being of the pop-up are retrieved from the Microsoft Sentinel also using the Microsoft Sentinel API - Alert Rules. After that, since the detection rules have a large amount of data, only the columns that had most interest for the SOC team are displayed.

The last tab, dashboard, is where the all the metrics proposed in Chapter3. For the proposed metrics, the SOC team decided that some were not worth of implementation given that the information that the metrics would present already could be deducted based on the information that is present in Microsoft Sentinel and information from metrics that were implemented. The metrics not needed in the dashboard are:

- **Number of use cases without alerts**: This metric was not needed to be implemented by the SOC team because, in the L3 UCs pop-up, the number of alerts that a UC has can be seen in the graph. In case no value is displayed in the graph, that would mean that the UC would count as a UC that does not have alerts.
- **MITRE ATT&CK coverage**: This metric was not implemented because Microsoft Sentinel already presents these values in a heatmap.
- **Changes to potential**: This metric was not needed to be implemented because it it is equal to 1 minus the 'effectiveness' metric (which, in this implementation, is identical to the weight metric in the MaGMa UCF). By having the Growth in weight metric (which is this implementation case should be called Growth in effectiveness), the Changes to potential metric would be the inverse of the Growth in effectiveness, making it unnecessary to track separately.

Also, taking into account the size of the implementation and time constraints, only a certain number of metrics could be implemented. For this, it was discussed with the SOC team what metrics could provide a bigger value to the SOC and could also be implemented having in account the existing time for the implementation. Given this, the following metrics were chosen and implemented, as it can also be seen in Figure 4.17.

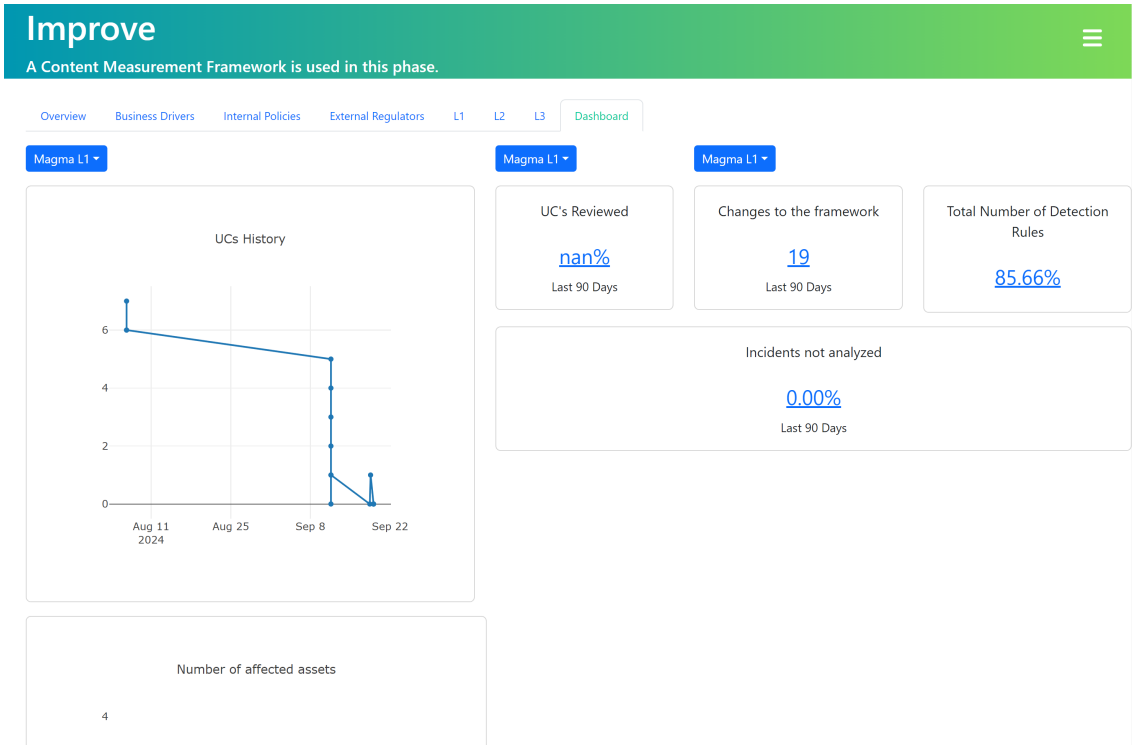


Figure 4.17: Implemented Dashboard tad

The previous Figure presents the following metrics:

- **Growth in number of UCs**: This metric is depicted by the initial graph on the left side of the Figure. As shown, the graph illustrates, for the chosen layer in the above dropdown (L1, L2, L3), the number of UCs starting from August (although it is designed to display data for the last 90 days, but it only shows data from August because its when the UCs were initially added). To implement this metric, each time a UC was added or removed, information about the name of the layer, the date of the addition or removal, and the number of UCs of the layer where the UC was added or removed was sent and recorded in the database (along with some additional data used in the Percentage of UCs Reviewed and Percentage of UCs reviewed metrics). With this, when a user selects a layer in the dropdown, all the data with more then ninety days is deleted. After that, data to use in this metric calculation is retrieved from the database and filtered accordingly to the selected layer. The graph then plots the dates from the database on the X axis and the number of UCs on the Y axis.
- **Percentage of UCs reviewed**: The UCs Reviewed card displays the percentage of UCs that have been modified within the past 90 days for the selected layer. To calculate this metric, each time a value of a UC is changed, then the date of the change, the action that occurred to the UC (Changed), and the layer of the UC is sent to the database (along with some additional information for the other metrics) are sent and saved in the database. When a layer in the dropdown is selected, the application first deletes all the data with more then 90 days. Then it retrieves data from the database, filtering for Changed actions within the specified layer. If a UC has multiple change records, only the earliest instance is considered. The percentage is then determined by dividing the count of modified UCs by the total number of UCs, in the selected

layer. It is also possible to click on the percentage value to display, in a pop-up, the UCs that were reviewed, their IDs, and the earliest time they were reviewed.

- **Changes to the framework**: This metric is similar to the Percentage of UCs Reviewed and Growth in number of UCs metric, as both use the same database table. However, instead of showing the percentage of UCs reviewed in the past 90 days, this metric provides the absolute number of changes that occurred in the selected layer from the dropdown. Each time a UC is added, deleted, or modified, the action is recorded in the database along with the UC's id and other details (which are not necessary for this metric). To calculate the metric value, a filter is applied based on the selected layer, and the number of resulting rows represents the number of changes made. Additionally, the metric value is clickable. Clicking on it opens a pop-up displaying the UC ID, the number of changes made to each UC, and the UC name.
- **Total number of detection rules**: To obtain the value of this metric, a simple call the Microsoft Sentinel API - Alert Rules was made. It returned a JSON value that was converted in a table. Each row has a detection rule. The number of rows was counted, obtaining the number of detection rules available in the Microsoft Sentinel.
- **Number of affected assets**: To determine this metric, a query was made to Microsoft Sentinel. For this query, the assets affected by each security incident over the past 90 days (a period specified by the SOC team) were retrieved and counted, with the data grouped by one-day intervals over the last 90 days. The Microsoft Sentinel provided two columns. The first column lists the last 90 days, segmented by each day. The second column displays the daily count of affected assets. The X axis indicates the time over the past 90 days, and the Y axis indicates the number of affected assets.
- **Incidents not analyzed**: Within the specific SOC where the solution was deployed, incidents with a status named New represent incidents not analyzed. To quantify not analyzed incidents, two Microsoft Sentinel queries were executed. One to count incidents with the New status and another to count the total number of incidents. The metric representing the percentage of not analyzed incidents was calculated by dividing the first one by the second one, multiplied by 100%. When percentage value is clicked, a pop-up displaying the name of the incidents not analyzed in the last 90 days is presented.

Check

The Check page layout, as shown in Figure **??**, features a progress bar that dynamically updates based on input. For each answered question, the progress bar advances by 20%, reflecting the 5 question structure.

Below the progress bar, five yes/no questions presented as radio buttons are presented. Upon answering a question, the application calculates the overall progress and updates the progress bar accordingly. For questions that require additional context, a textarea becomes visible to provide comments.

To maintain data integrity, a JSON file stores the answers and associated comments. When the page loads, the progress bar's initial value is determined by the number of previously answered questions, and existing answers and comments are displayed.

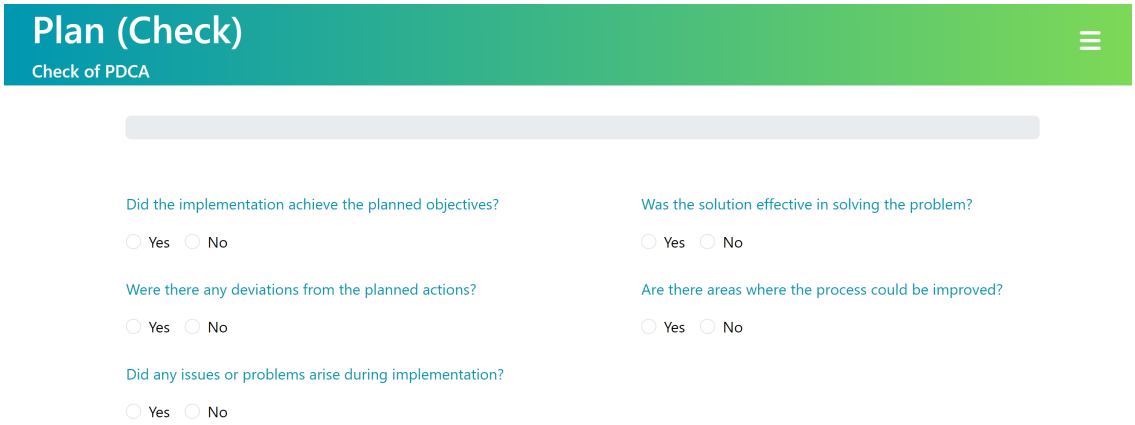


Figure 4.18: Implementation for the Check page.

4.4.6 Control

The Control phase is the fifth and last phase of a DMAIC process. The main activity in the Control phase is to control the improved process (Use Case Management). In the context of the implementation, in order to monitor and control the process, a graph is presented that displays the maturity value over time, which can be seen in Figure 4.19. The graph shows that in May 2024 a variation of the maturity for Use Case Management occurred, dropping for a value bigger then four to a value between zero and 1. In the next month, the maturity stabilized with a value of two.

In this page the expenses over the entire cycle are documented, for example databases, software, among others. Suggestion and improvements for the next DMAIC cycle are also written, along with additional comments. It is important to note that, as previously said, some information does not correspond to real information.

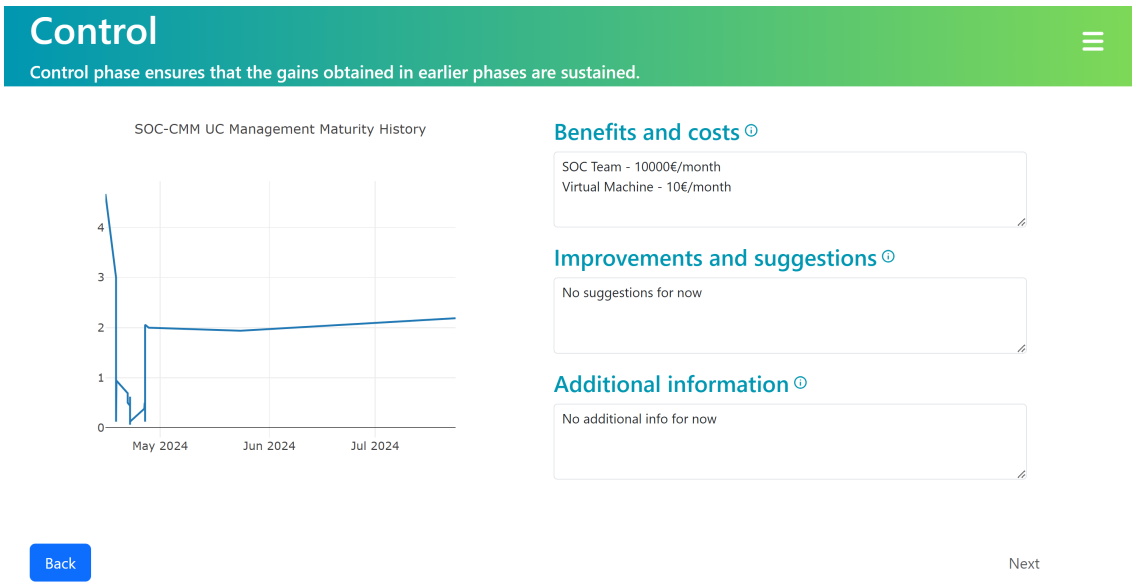


Figure 4.19: Implementation for the Control page.

4.5 Resume

In the initial section of this chapter, a PoC was implemented and documented, primarily as a web application. First, the technologies used for the implementation were introduced with brief descriptions. Following this, a deployment diagram was conceptualized and presented to illustrate the relationships between the software and hardware components of the PoC. Next, the method for accessing the application was detailed. Finally, a detailed explanation of the sequential implementation of the PoC was provided.

Chapter 5

Demonstration

To demonstrate the suitability and validity of the proposed solution, this chapter demonstrates how the implemented PoC, described in the previous Chapter 3 was performed. In this Chapter, results given by the implemented solution should be documented but given the sensitive information that the information can show, these could not be displayed.

5.1 Demonstration in Insurers

Firstly, in the Define page, the Use Case Management aspect of the SOC within the Process domain were defined. This means that throughout the DMAIC cycle, the Use Case Management will prioritize for its improvement. Next, for the problem statement, it was define the following problem: Difficulty in organizing SOC information, leading to misalignment between detection mechanisms, monitoring approach, and business objectives. After the problem defined, and in discussions with the SOC team, the maturity target and objectives were defined for Use Case Management. The maturity target was aimed at its highest maturity value (5) to ensure optimal Use Case Management. The objectives defined were the following:

- Align security monitoring with business needs and Use Cases.
- Enhance proactive threat identification and mitigation capabilities.
- Establish metrics to measure the effectiveness of threat identification and mitigation.
- Continuously monitor and improve monitoring processes, use case effectiveness, and security monitoring.
- Stakeholders: SOC Manager, SOC Analysts, and CISO.

Regarding the type of SOC and the budget, this was a straightforward answer, as the SOC is an hybrid SOC and no budget was given (although the money spent was reported to the SOC Manager). After describing the type of SOC, the success measurement were:

- Positive feedback from SOC stakeholders.
- Improvement in the defined aspect and overall SOC maturity.
- Ability to connect business needs to security monitoring.
- Increased visibility of SOC security monitoring.

The non-success measurement was:

- Resistance from SOC staff to adopt the solution.

- Failure to meet objectives within the allocated budget.
- Budget: While no specific budget was provided for the DMAIC cycle, we will report budget expenditures to the SOC Manager.

All this information can be seen in Figure 5.1.

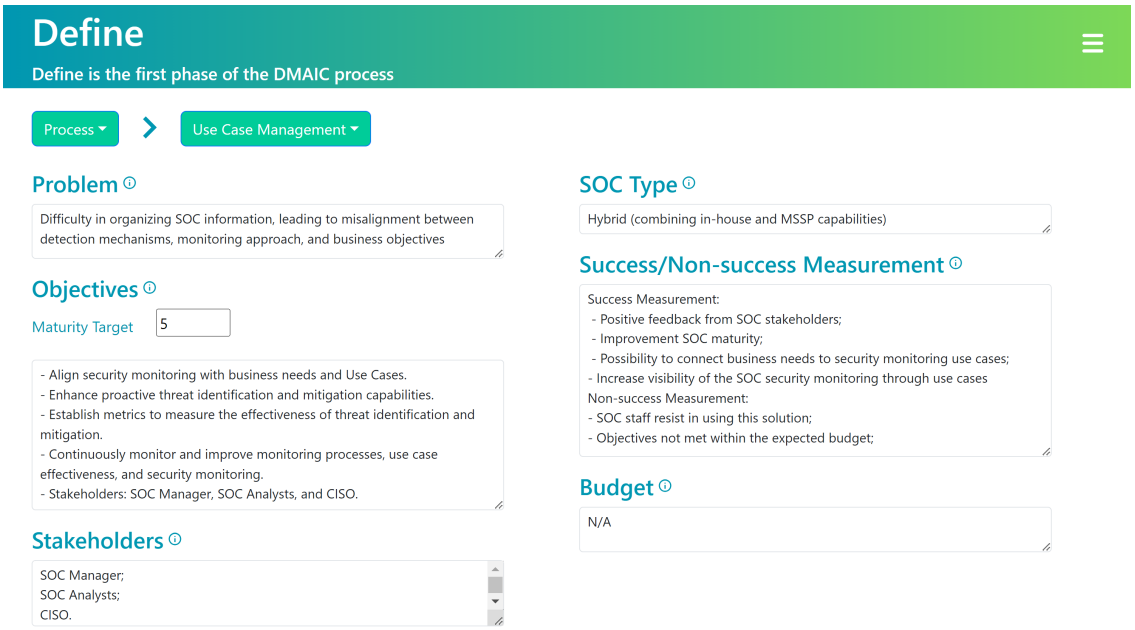
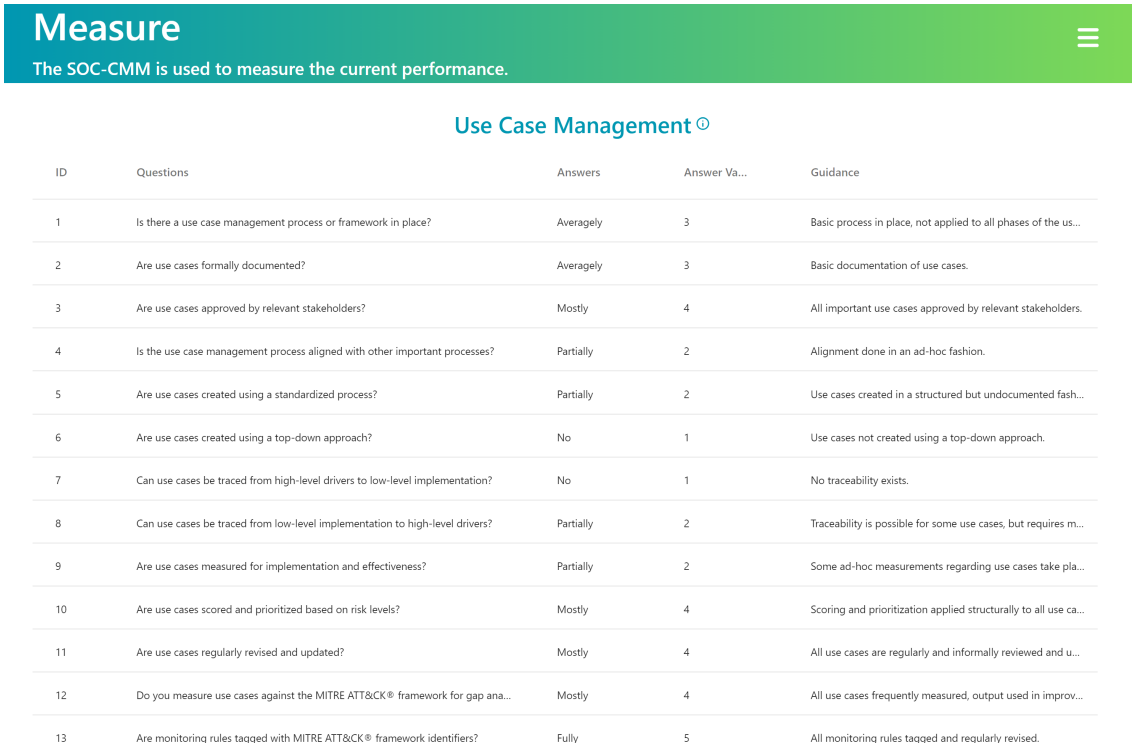


Figure 5.1: Demonstration for Define page.

With the Define page documented, the following page, the Measure page, was used. In here, all the maturity questions related to the Use Case Management were answer. For this, documentation and discussions with the SOC stakeholders were made in order to correctly answer the questions and have a correct value of maturity. The questions answered can be seen in Figure 5.2



ID	Questions	Answers	Answer Va...	Guidance
1	Is there a use case management process or framework in place?	Averagely	3	Basic process in place, not applied to all phases of the us...
2	Are use cases formally documented?	Averagely	3	Basic documentation of use cases.
3	Are use cases approved by relevant stakeholders?	Mostly	4	All important use cases approved by relevant stakeholders.
4	Is the use case management process aligned with other important processes?	Partially	2	Alignment done in an ad-hoc fashion.
5	Are use cases created using a standardized process?	Partially	2	Use cases created in a structured but undocumented fash...
6	Are use cases created using a top-down approach?	No	1	Use cases not created using a top-down approach.
7	Can use cases be traced from high-level drivers to low-level implementation?	No	1	No traceability exists.
8	Can use cases be traced from low-level implementation to high-level drivers?	Partially	2	Traceability is possible for some use cases, but requires m...
9	Are use cases measured for implementation and effectiveness?	Partially	2	Some ad-hoc measurements regarding use cases take pla...
10	Are use cases scored and prioritized based on risk levels?	Mostly	4	Scoring and prioritization applied structurally to all use ca...
11	Are use cases regularly revised and updated?	Mostly	4	All use cases are regularly and informally reviewed and u...
12	Do you measure use cases against the MITRE ATT&CK® framework for gap ana...	Mostly	4	All use cases frequently measured, output used in improv...
13	Are monitoring rules tagged with MITRE ATT&CK® framework identifiers?	Fully	5	All monitoring rules tagged and regularly revised.

Figure 5.2: Demonstration for Measure page.

After having all question answered and reviewed, the results of assessment made in this Measure page, are presented in the next page, the Analyze page.

In the Analyze page, the results obtained can be seen in Figure 5.3, Based on the maturity result obtained (1.88), the Risk Assessment (1.25), Risk Management Strategy (1.38) and Risk Identify (Overall) (0.53), when comparing with the maturity target (5), it was evident that a maturity improvement is needed for the Use Case Management aspect. Given this, the table below these results was then used to better understand where the SOC lacks in this aspect of the SOC, and the next phase (Improve phase) was use to plan and improve the maturity value.

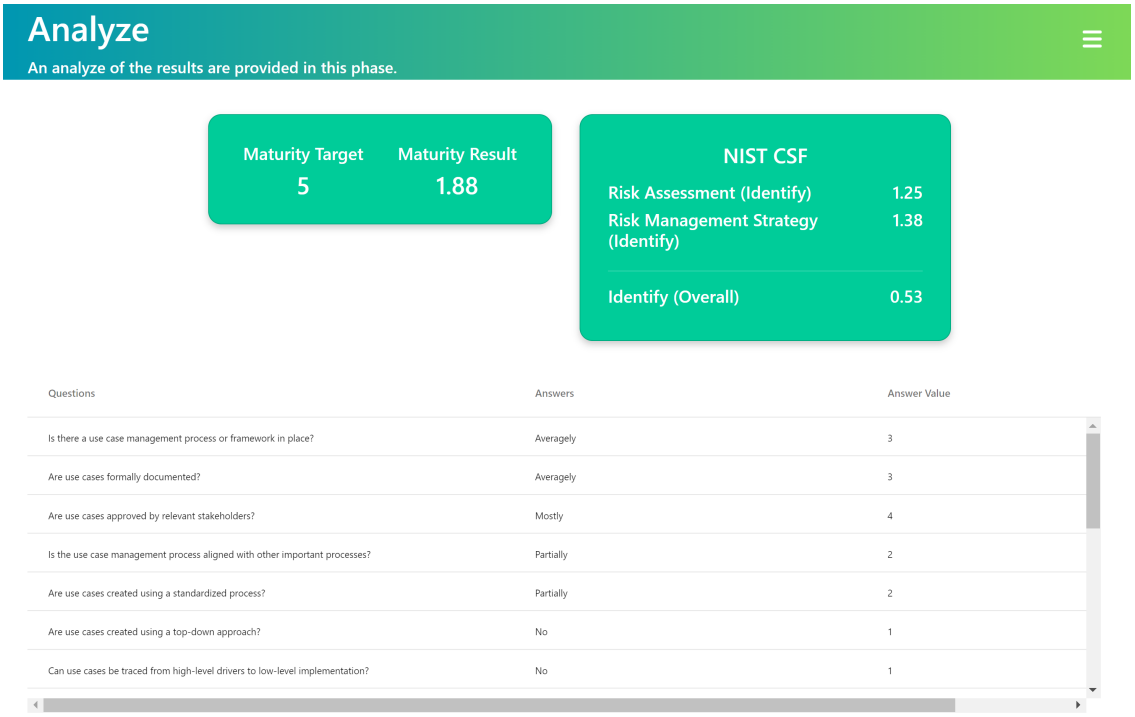


Figure 5.3: Demonstration for Analyze page.

Inside the Improve page (where the PDCA is implemented), the subpage Plan was used to document the plan for this PDCA cycle, technologies that can/will be used, and important KPIs. It is also possible to write additional comments, although none were made.

For the Plan, the following information was documented:

- Conduct an thorough audit of existing business drivers, internal policies and external regulators for the SOC. - Document implementation information of UCs. - Tie UCs to MITRE ATT&CT and Microsoft Sentinel. - Map UCs from their implementation to their business requisites. - Rank UCs from on their business needs to their implementation details. - Encourage collaboration between security analysts and incident responders when creating UCs. - Collect data on these metrics to assess UCs performance. - Keep a continuous review and update UCs.

Additional Considerations: - Training SOC analysts on the solution. - Regularly test UCs to ensure they effectively detect and respond to security threats. - Explore automation opportunities for repetitive tasks within UCs to improve efficiency.

1. Conduct an thorough audit of existing business drivers, internal policies and external regulators for the SOC.
2. Document implementation information of UCs.
3. Tie UCs to MITRE ATT&CT and Microsoft Sentinel.
4. Map UCs from their implementation to their business requisites.
5. Rank UCs from on their business needs to their implementation details.
6. Encourage collaboration between security analysts and incident responders when creating UCs.

7. Collect data on these metrics to assess UCs performance.
8. Keep a continuous review and update UCs.

Additional Considerations:

1. Training SOC analysts on the solution.
2. Regularly test UCs to ensure they effectively detect and respond to security threats.
3. Explore automation opportunities for repetitive tasks within UCs to improve efficiency.

All this information can be seen in Figure 5.4

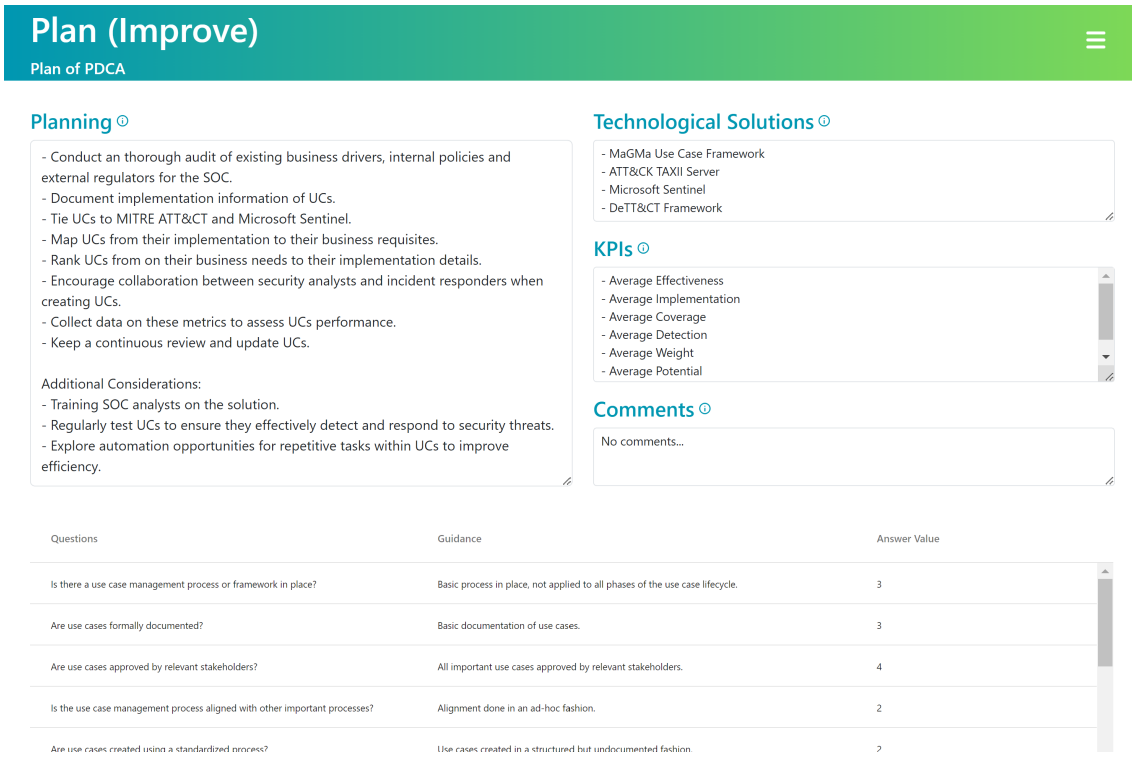


Figure 5.4: Demonstration for Plan page.

Following the Plan page, the defined plan was implemented in the Do & Act page, where business and compliance drivers were added, along with L1, L2, and L3 UCs, in their specific tab,. The MaGMa UCF methodology, as detailed in Chapter 2, was applied. However, a challenge arose when adding UCs to the L3 tab. Microsoft Sentinel's latest API for retrieving detection rules lacks sub-technique values, which are essential for L3 UCs to connect to the ATT&CK TAXII server and calculate metrics. This ultimately affects L2, L1, and Overview tab UCs. The SOC team raised a ticket with Microsoft to address this issue. After two months, Microsoft confirmed that the current API is intentionally omitting sub-techniques but an an older version of the API provides the sub-techniques information. This led to delays when adding UCs, resulting in incomplete metric values and and an incomplete information in the solution implemented.

To add UCs, since the solution was implemented in SOC already in production, this firstly involved collaborating with business stakeholders to identify key business drivers, internal policies, and external regulatory requirements. These factors were then documented to

establish the purpose and context for UC implementation. Next, L3 UCs were added, with their necessary information. After having the L3 UCs added, L2 UCs were then created and the L3 UCs were mapped to the L2 UCs. Finally, the L1 UCs were created, based on the CKC, and then the L2 UCs were mapped to L1 UCs. The results of the framework, duo to company privacy policies, could not be displayed.

After the implementation of the UCs, the next page, the Check page, was used to document related issues and possible improvement based on the Do phase. In this phase, all the questions were answered, but for question Did any issues or problems arise during implementation? it was documented that, because of the related Microsoft API problem, the documentation of UCs took longer then expect. Also, for the question Are there areas where the process could be improved?, it was commented that more automatic metrics could be added.

5.2 Insurers Case Study

To demonstrate the suitability and validity of the proposed solution and its implementation, this section presents one case study for the insurance company. In this case study, it will analyzed the maturity value for Use Case Management before implementing the proposed solution, which in the previous section, the value obtained was 1.88, and the maturity value after after the implementation of the solution.

For this, after completing the demonstration of the PoC, the Measure page was used to answer the question again. The new results were 3.88 for Maturity Result, representing a value increase of 1.87 in the maturity of Use Case Management. Also the Risk Assessment changed from 1.25 to 3.75, the Risk Management Strategy from 1.38 to 2.88 and and Risk Identify (Overall) from 0.53 to 1.33, as cam be seen in Figure **??**

Figure 5.5: Maturity results after implementing the PoC.

These results changes were because the answers for the following questions were changed:

- Is there a use case management process or framework in place? The maturity value for this question changed from 3 to 4 because a formal process covering all UC was in place.
- Are use cases formally documented? The maturity value for this question changed from 3 to 5 because the UCs are documented in the solution and maintained.

- Is the use case management process aligned with other important processes? The maturity value for this question changed from 2 to 4 because the UC management was started be align with other SOC processes.
- Are use cases created using a standardized process? The maturity value for this question changed from 2 to 3 because UCs are mostly documented structured and documented way.
- Are use cases created using a top-down approach? The maturity value for this question changed from 1 to 3 because UCs are created in a top-down approach but only with SOC context.
- Can use cases be traced from high-level drivers to low-level implementation? The maturity value for this question changed from 1 to 4 because the UC are can be traced from high-level drivers to low-level implementation but are not validated by the most important stakeholders.
- Can use cases be traced from low-level implementation to high-level drivers? The maturity value for this question changed from 1 to 4 because the UC are can be traced from low-level implementation to high-level drivers but are also not validated by the most important stakeholders.
- Are use cases measured for implementation and effectiveness? The maturity value for this question changed from 2 to 5 because these metrics are applied to UCs, used to guide risk-based use case growth.
- Have you created a MITRE ATT&CK risk profile for your organization? The maturity value for this question changed from 1 to 5 because the solution allows to create a MITRE ATT&CK provide, and is validated and regularly maintained.
- Have you prioritized MITRE ATT&CK techniques for relevance? The maturity value for this question changed from 2 to 4 because techniques are now prioritized and validated, but not regularly maintained.
- Is threat intelligence used for the creation and updates of use cases? The maturity value for this question changed from 2 to 4 because threat intelligence is starting to be used for creating and updating UCs, although no formal process is in place
- Do you determine and document visibility requirements for each use case? The maturity value for this question changed from 2 to 5 because there is visibility for UCs, fully documented and reviewed.
- Do you measure visibility status for your use cases for gap analysis purposes? The maturity value for this question changed from 2 to 4 because visibility status is measured but not always an improvement is made.
- Do you map data source visibility to the MITRE ATT&CK framework? The maturity value for this question changed from 2 to 5 because data sources are continuously mapped to ATT&CK and the output used in improvement.

Although the solution implemented led to an improvement in the UC Management aspect of the SOC, since the target maturity is 5, and the maturity obtained was 3.88, then the SOC should continuously improve this service.

5.3 Interview

In order to understand the impact that the developed solution and PoC have for the SOC where the solution was implemented, a brief questionnaire (given the to time constraints) as send to two SOC professionals.

5.3.1 Subjects

The initial phase the selection of interviewees. Based on the SOC where the PoC was implemented, only two users two more users can access the environment where the application was deployed, so only there were only to participates for the interview process. To maintain their privacy, their names and personal information are not disclosed in this report. However, their professional profiles demonstrate a high level of expertise and experience in the field of cybersecurity. These are:

- Subject 1 (S1)
  - Cybersecurity consultant and SOC Manager
  - Managed 3 SOCs
  - Has more then 5 years of experience in SOCs.
- Subject 2 (S2)
  - SOC Analyst
  - Has managed core technologies of SOC such as Security Information and Event Management, Security Orchestration, Automation and Response (SOAR) and Managed Detection and Response.

5.3.2 Method

As a methodology, two questions were formulated and sent to S1 and S2 participates to understand their about the solution developed. This allows have concise answers to document. Since both S1 and S2 were familiar with the solution developed, the questions were answered in few hours. The questions were:

- Q1 - How can the implemented solution help in the daily life of SOC professionals?
- Q2 - How important is having a continuous improvement application for SOCs?

5.3.3 Results

The responses from participants were analyzed and categorized as either supportive or developmental. The answers, although not using the exact same worlds, where the following:

S1

- Answer for Q1 - The solution developed enables a comprehensive assessment of the alignment between current detection capabilities and the intended detection strategy. By categorizing detection types (data components) and covered platforms, it highlights gaps between the SOC's desired detection state and underdeveloped techniques. Additionally, the solution facilitates the identification of redundant detection rules and unnecessary use cases, streamlining the overall detection process.

- Answer for Q2 - Given the complexity of implementation of a SOC along with the constant change of an organization's environment (technological environment, threat landscape, business/conformity requirements), the platform used for management and operationalization of the detection strategy should incorporate continuous improvement principles. This is achieved by implementing established frameworks and knowledge bases with DMAIC + PDCA methodologies in mind, integrating with real-time visibility over the operationalized detections and incident response procedures.

S2

- Answer for Q1 - It can help to have a clear and more global view of the SOC, and better understand the existing gaps and improvements that can be made.
- Answer for Q2 - Technology is always improving, so having an application that promotes continuous improvement allows for a change in operations, taking into account the new threats and needs of the SOC that the SOC should monitor.

5.4 Resume

This chapter encompassed a comprehensive examination of PoC demonstration the results produced. First, the aspect of the SOC, along with relevant information was documented to understand the aspect of SOC will be improved, throughout the application. Then, an assessment of the SOC aspect defined was made, and the results from the assessment prompt the need for its maturity improvement. Based on that, a plan was made, using the technologies implemented in the Do page, and implemented, to improve the problem of the defined SOC aspect. Then, is was Checked if the during the plan made where any issues, which did not, so no need to fix any problems in the solution. A case study and interview was also made to understand the impact of the solution in the SOC and for professional SOC workers.

Chapter 6

Conclusion and Future Work

This chapter provides the main conclusions of this thesis, highlighting the accomplished objectives. The limitations of the proposed solution and possible improvements are also described, indicating research topics to be explored in the future.

6.1 Conclusion

SOCs are vital for cybersecurity, offering real-time monitoring and fast response to threats. However, a SOC is just the first step. Continuous improvement, refining processes, updating tools, and adapting to new threats is crucial for long-term security and stakeholder trust. Despite, many SOCs lack this focus, hindering their effectiveness. SOCs should strive for continuous improvement and a increase in its maturity and capability.

This thesis presented research, on maturity and capability modals for SOC and CMF. Based on the research, a solution and a PoC for an application was design and implemented for allowing SOCs professionals to have a continuous improvement of the SOC aspects, in this thesis case mainly for detection, monitoring and business alignment through UC management. The solution aims to continuously improve SOC functionalities using DMAIC and PDCA methodologies.

Based on the results obtained, the tool was capable of improving the Use Case Management aspect (although it did not the maturity target defined by the SOC). However, questions were made to the SOC where the tool was implemented and a positive feedback was given for the use of the tool in the daily operations of the SOC.

6.2 Objectives Achieved

Table 6.1: Objectives fulfilled

Objective	Description	Achieved
OB1	Conduct a research on existing ways to assess the maturity and capability of a SOC.	✓
OB2	Investigate and evaluate the most practical and suitable CMF(s), with an emphasis on UCF, to improve the maturity and capacity of SOC.	✓
OB3	Implement a proof-of-concept (POC) to validate the selected CMF's effectiveness.	✓
OB4	Analyze and evaluate the impact of the selected solution.	✓

6.3 Research Questions Answered

In Section 1.4, four research questions were raised, and answers can already be provided based on the achieved objectives. The main conclusions that can be drawn from them are the following:

6.3.1 RQ1 - What is the state-of-the-art approach to self-evaluating a SOC maturity and capability?

It was observed that exist a limited number of models specifically tailored to measure the SOC maturity and capability. Based the available models, the SOC-CMM stood out as the most compressive and up-to-date modal, allowing to evaluate different SOC's services and aspects. The modal also maps the assessment results to NIST CSF 2.0, a widely used and well-known cybersecurity framework. However, it is meaningful to mentioned that in [35], a continuous improvement approach for SOCs, using the SOC-CMM, was presented allowing for a continuous measurement of the maturity and capability of SOCs services and aspects, and improve the necessary ones.

6.3.2 RQ2 - Which existing CMFs can enhance the management of SOCs content through UCs, with emphasis in monitoring, detection, and business alignment, leading to a SOC maturity and capability improvement?

Four CMFs were identified and discussed as potential enhancers of SOCs' maturity and capabilities. Among these, three CMFs are specifically designed to improve the management of SOC information through UCs. The first framework found, MaGMa for Threat Hunting focuses on threat hunting, providing a structured approach to organizing UCs and a methodology for SOC teams to conduct threat hunting effectively. The second one, SPEED UCF, is a framework that offers a systematic approach for identifying, classifying, and documenting UCs, enhancing the overall management of SOC information. The third framework, which is MaGMa UCF, is similar to MaGMa for Threat Hunting since it provides a structured method for organizing UCs, from addressing business needs to practical implementation. It also includes a UC life cycle, a methodology for using the framework, and several metrics to evaluate its effectiveness. The fourth framework and last framework founded, DeTT&CT,

although not specifically designed for UCs, helps SOCs better understand their current detections, visibility, and detection coverage, thereby improving the overall prevention and detection of cyber threats.

6.4 Limitations and Future Work

Despite the importance that the CMF can have to improve continuous improve SOCs (although it was limited to Use Case Management), and in detection, monitoring and better business alignment, several improvements can be made.

Firstly, the tool can be extended to the other aspects and services of SOCs. By extending the tool's capabilities to address the diverse functions of a SOC, organizations can achieve more holistic risk mitigation, optimize operational efficiency, enhance compliance posture, and foster a culture of continuous improvement. This approach enables a unified approach to security, proactive identification and addressing of vulnerabilities, demonstration of adherence to industry standards and regulations, and data-driven decision-making for security management, among many other advantages.

Secondly, the framework could be fully integrated with the DeTT&CT framework by incorporating heatmaps into the Do&Act page. Heatmaps offer a more intuitive way to visualize information, allowing SOCs to easily identify key insights. Since the DeTT&CT framework is already a part of the proposed solution, utilizing it to better visualize detection and coverage gaps would significantly enhance the SOC's capabilities.

Thirdly, the solution provides, mostly in its dashboard tab, a limited number of metric when compared to the ones presented in the solution proposed. Complementing the implementation with the those metrics would allow the SOC (where the PoC was implemented) to better organize and have a clear visualization of the information managed in the SOC, most likely leading to an improve monitoring and detection of threats.

Fourthly, it was discussed, during the development of the thesis solution, to automated the answers of the SOC-CMM based on the information that exists in the solution. This would a allow for more up-to-date and easier maturity and capability measurement of the SOC.

Lastly, the used software tools, although most were encouraged to be used by the insurance company where the PoC was implemented, are now acknowledge that are not the applicable tools to develop such solution. Changing the tool would be adequate, in this case.

Bibliography

[1] Harjinder Singh Lallie et al. "Cyber security in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic". In: *Computers & security* 105 (2021), p. 102248.

[2] Andreea Bendovschi. "Cyber-attacks–trends, patterns and security countermeasures". In: *Procedia Economics and Finance* 28 (2015), pp. 24–31.

[3] Arif Ali Mughal. "Building and Securing the Modern Security Operations Center (SOC)". In: *International Journal of Business Intelligence and Big Data Analytics* 5.1 (2022), pp. 1–15.

[4] Pierre Jacobs, Alapan Arnab, and Barry Irwin. "Classification of Security Operation Centers". In: *2013 Information Security for South Africa*. 2013, pp. 1–7. doi: 10.1109/ISSA.2013.6641054.

[5] Cyril Onwubiko and Karim Ouazzane. "Challenges towards building an effective cyber security operations centre". In: *arXiv preprint arXiv:2202.03691* (2022).

[6] Josephine Wolff. "How Is Technology Changing the World, and How Should the World Change Technology?" In: *Global Perspectives* 2.1 (2021), p. 27353. url: https://doi.org/10.1525/gp.2021.27353.

[7] Ömer Aslan et al. "A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions". In: *Electronics* 12.6 (2023), p. 1333.

[8] Scott Monteith et al. "Increasing cybercrime since the pandemic: Concerns for psychiatry". In: *Current psychiatry reports* 23 (2021), pp. 1–9.

[9] IBM. *Cost of a Data Breach Report 2023*. https://www.ibm.com/reports/data-breach. [Accessed 23-12-2023]. 2023.

[10] Calif. Sausalito. *Cybercrime To Cost The World 8 Trillion Annually In 2023*. https://cybersecurityventures.com/cybercrime-to-cost-the-world-8-trillion-annually-in-2023/. Accessed: (12/11/2023). 2022.

[11] Shield Services. *The Six Costliest Cyberattacks in History — linkedin.com*. https://www.linkedin.com/pulse/six-costliest-cyberattacks-history-shieldsupport. [Accessed 23-12-2023]. 2023.

[12] Ellis Stewart. *Top 10 Most Expensive Cyber Attacks in History*. https://em360tech.com/top-10/expensive-cyber-attacks. [Accessed 23-12-2023]. 2023.

[13] vodafone. *Cyberattack on Vodafone Portugal*. https://www.vodafone.pt/en/press-releases/2022/2/cyberattack-on-vodafone-portugal.html. [Accessed 03-01-2024]. 2022.

[14] The National Cyber Security Centre. *MOVEit vulnerability and data extortion incident*. https://www.ncsc.gov.uk/information/moveit-vulnerability. [Accessed 23-12-2023]. 2023.

[15] Risto Vaarandi and Sten Mäses. "How to Build a SOC on a Budget". In: *2022 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE. 2022, pp. 171–177.

[16] Manfred Vielberth et al. "Security operations center: A systematic study and open challenges". In: *IEEE Access* 8 (2020), pp. 227756–227779.

[17] ISACA. "The Evolution of Security Operations and Strategies for Building an Effective SOC". In: *ISACA JOURNAL* (2021).

[18] Otto Lindström. "Next generation security operations center". In: (2018).

[19] Hiep Nguyen Duc. *SOC Maturity Model*. https://eforensicsmag.com/soc-maturity-model/. [Accessed 03-01-2024]. 2023.

[20] Rob van Os et al. *Magma: A Framework and tool for use case management*. Nov. 2017. url: https://www.betaalvereniging.nl/wp-content/uploads/FI-ISAC-use-case-framework-verkorte-versie.pdf.

[21] Faris Bugra Kokulu et al. "Matched and Mismatched SOCs: A Qualitative Study on Security Operations Center Issues". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. CCS '19. London, United Kingdom: Association for Computing Machinery, 2019, pp. 1955–1970. isbn: 9781450367479. doi: 10.1145/3319535.3354239. url: https://doi.org/10.1145/3319535.3354239.

[22] PonemonInstitute. *Ponemon Institute and Devo Technology Study Reveals 65% of Cybersecurity Analysts Consider Quitting Due to Burnout, Lack of Visibility*. https://www.businesswire.com/news/home/20190729005244/en/Ponemon-Institute-Devo-Technology-Study-Reveals-65. [Accessed 23-12-2023]. 2019.

[23] Hewlett-Packard. *State of security operations*. Tech. rep. HP, 2014.

[24] Desiree Sacher. "Fingerpointing False Positives: How to Better Integrate Continuous Improvement into Security Monitoring". In: *Digital Threats* 1.1 (Mar. 2020). doi: 10.1145/3370084. url: https://doi.org/10.1145/3370084.

[25] Cambridge University Press. *ethic*. 2024. url: https://dictionary.cambridge.org/dictionary/english/ethic.

[26] Richard L. Nolan. "Managing the Computer Resource: A Stage Hypothesis". In: *Commun. ACM* 16 (July 1973), pp. 399–405. url: https://doi.org/10.1145/362280.362284.

[27] David Moher et al. "Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement". In: *Systematic reviews* 4 (2015), pp. 1–9. doi: 10.1186/2046-4053-4-1.

[28] Matthijs Vos. "Capability Maturity Measurement of a Security Operations Center through Analysis Detection". MA thesis. University of Twente, 2022.

[29] Pierre Conrad Jacobs. "Towards a framework for building security operation centers". PhD thesis. Rhodes University, 2014.

[30] Yassine Maleh Issam Taqafi and Karim Ouazzane. "A MATURITY CAPABILITY FRAMEWORK FOR SECURITY OPERATION CENTER". In: *EDPACS* 67.3 (2023), pp. 21–38. doi: 10.1080/07366981.2023.2159047.

[31] Daniel Schlette, Manfred Vielberth, and Günther Pernul. "CTI-SOC2M2 – The Quest for Mature, Intelligence-Driven Security Operations and Incident Response Capabilities". In: *Comput. Secur.* 111.C (Dec. 2021). issn: 0167-4048. doi: 10.1016/j.cose.2021.102482. url: https://doi.org/10.1016/j.cose.2021.102482.

[32] Rob Van Os. "SOC-CMM : Designing and Evaluating a Tool for Measurement of Capability Maturity in Security Operations Centers". In: 2016. url: https://ltu.diva-portal.org/smash/get/diva2:1033727/FULLTEXT02.pdf.

[33] Arthur Revaclier. *SOC-AM: An Accessible Maturity Model for Security Operation Centers*. 2021.

[34] Microsoft. *Background to Capability Maturity Model Integration (CMMI)*. 2023. url: https://learn.microsoft.com/en-us/azure/devops/boards/work-items/guidance/cmmi/guidance-background-to-cmmi?view=azure-devops.

[35] Cengiz Acarturk, Murat Ulubay, and Efe Erdur. "Continuous Improvement on Maturity and Capability of Security Operation Centres". In: 15.1 (Dec. 2020), pp. 59–75. doi: 10.1049/ise2.12005. url: https://doi.org/10.1049/ise2.12005.

[36] CREST. *THE CREST CYBER SECURITY INCIDENT RESPONSE MATURITY ASSESSMENT TOOL*. 2014. url: https://www.crest-approved.org/wp-content/uploads/2022/04/CSIR-Maturity-assessment-tool_Info1.pdf.

[37] The Open Group. *The Open Group Service Integration Maturity Model (OSIMM), Version 2*. 2011. url: https://www.opengroup.org/soa/source-book/osimmv2/index.htm.

[38] Per Lindberg and Anders Berger. "Continuous improvement: design, organisation and management". In: *International Journal of Technology Management* 14.1 (1997), pp. 86–101.

[39] Mohammad Aazadnia and Mehdi Fasanghari. "Improving the information technology service management with six sigma". In: *International journal of computer science and network security* 8.3 (2008), pp. 144–150.

[40] van Os. *SOC-CMM - Measuring capability maturity in security operations centers*. 2016. url: https://www.soc-cmm.com/introduction/.

[41] Bushra A Alahmadi, Louise Axon, and Ivan Martinovic. "99% False Positives: A Qualitative Study of {SOC} Analysts' Perspectives on Security Alarms". In: *31st USENIX Security Symposium (USENIX Security 22)*. 2022, pp. 2783–2800.

[42] Cyril Onwubiko and Karim Ouazzane. "Cyber Onboarding is 'Broken'". In: *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. 2019, pp. 1–13. doi: 10.1109/CyberSecPODS.2019.8885237.

[43] Tom Mulders. "Discerning Wheat from Chaff in SOCs: A Model to Identify 'Non-Interesting'Events in Security Operation Centers". In: (2022).

[44] Joonas Forsberg. *Measuring the technical performance of a security operations center*. 2022.

[45] Antonio Villalón-Huerta, Hector Marco Gisbert, and Ismael Ripoll-Ripoll. "SOC Critical Path: A Defensive Kill Chain Model". In: *IEEE Access* 10 (2022), pp. 13570–13581. doi: 10.1109/ACCESS.2022.3145029.

[46] Tsviatko Bikov et al. "Threat Hunting as Cyber Security Baseline in the Next-Generation Security Operations Center". In: *2021 29th Telecommunications Forum (TELFOR)*. IEEE. 2021, pp. 1–4.

[47] Matt Tatam et al. "A review of threat modelling approaches for APT-style attacks". In: *Heliyon* 7.1 (2021), e05969. issn: 2405-8440. doi: https://doi.org/10.1016/j.heliyon.2021.e05969. url: https://www.sciencedirect.com/science/article/pii/S2405844021000748.

[48] Seok Bin Son et al. "Introduction to MITRE ATT&CK: Concepts and Use Cases". In: *2023 International Conference on Information Networking (ICOIN)*. IEEE. 2023, pp. 158–161.

[49] R. Rehman. *Cybersecurity Arm Wrestling: Winning the Perpetual Fight Against Crime by Building a Modern Security Operations Center (SOC)*. Amazon Digital Services LLC - Kdp, 2021. isbn: 9798733168166. url: https://books.google.pt/books?id=MFBkzgEACAAJ.

[50] MITRE ATT&CK. "Mitre att&ck". In: *URL: https://attack. mitre. org* (2021).

[51] Stylianos Karagiannis et al. "A-DEMO: ATT&CK Documentation, Emulation and Mitigation Operations: Deploying and Documenting Realistic Cyberattack Scenarios-A Rootkit Case Study". In: *Proceedings of the 25th Pan-Hellenic Conference on Informatics*. 2021, pp. 328–333.

[52] *SOC-CMM Metrics 101*. https://www.soc-cmm.com/products/metrics/. Mar. 2024.

[53] Rob van Os et al. *Tahiti: A threat hunting methodology*. https://www.betaalvereniging.nl/wp-content/uploads/TaHiTI-Threat-Hunting-Methodology-whitepaper.pdf. [Accessed 30-12-2023]. 2018.

[54] Marcus Bakker and Ruben Bouman. *Detect Tactics, Techniques & Combat Threats*. 2019. url: https://github.com/rabobank-cdc/DeTTECT.

[55] NVISO. *DeTT&CT : Mapping detection to MITRE ATT&CK*. 2022. url: https://blog.nviso.eu/2022/03/09/dettct-mapping-detection-to-mitre-attck/.

[56] globema. *Automated data processing: key benefits and use cases*. 2023. url: https://www.globema.com/automated-data-processing-key-benefits-and-use-cases/.

[57] Cersei Page. *Software Engineering*. New York, NY, USA: Larsen and Keller Education, 2017. isbn: 1635492629.

[58] Coursera Staff. *What Is Python Used For? A Beginner's Guide*. 2024. url: https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python.

[59] GeeksforGeeks. *Deployment Diagram in unified modeling language(uml)*. 2024. url: https://www.geeksforgeeks.org/deployment-diagram-unified-modeling-languageuml/.

[60] Kelly Vehent. *How to use Dash and Shiny on Azure Machine Learning*. [Online; accessed 10-September-2024]. 2023. url: https://www.linkedin.com/pulse/how-use-dash-shiny-azure-machine-learning-kelly-vehent/.

[61] Wikipedia contributors. *Effectiveness — Wikipedia, The Free Encyclopedia*. 2024. url: https://en.wikipedia.org/w/index.php?title=Effectiveness&oldid=1240173831.

Appendix A

Appendix A

Function	Category	Category Identifier
Govern (GV)	Organizational Context	GV.OC
	Risk Management Strategy	GV.RM
	Roles, Responsibilities, and Authorities	GV.RR
	Policy	GV.PO
	Oversight	GV.OV
	Cybersecurity Supply Chain Risk Management	GV.SC
Identify (ID)	Asset Management	ID.AM
	Risk Assessment	ID.RA
	Improvement	ID.IM
Protect (PR)	Identity Management, Authentication, and Access Control	PR.AA
	Awareness and Training	PR.AT
	Data Security	PR.DS
	Platform Security	PR.PS
	Technology Infrastructure Resilience	PR.IR
Detect (DE)	Continuous Monitoring	DE.CM
	Adverse Event Analysis	DE.AE
Respond (RS)	Incident Management	RS.MA
	Incident Analysis	RS.AN
	Incident Response Reporting and Communication	RS.CO
	Incident Mitigation	RS.MI
Recover (RC)	Incident Recovery Plan Execution	RC.RP
	Incident Recovery Communication	RC.CO

Figure A.1: NIST CSF 2.0

Function/ Code	Category/ Code	Sub-Category/ Code
Identity (ID)	Asset Management [ID.AM]	ID.AM-1, ID.AM-2, ID.AM-3, ID.AM-4, ID.AM-5, ID.AM-6
	Business Environment [ID.BE]	ID.BE-1, ID.BE-2, ID.BE-3, ID.BE-4, ID.BE-5
	Governance [ID.GV]	ID.GV-1, ID.GV-2, ID.GV-3, ID.GV-4
	Risk Assessment [ID.RA]	ID.RA-1, ID.RA-2, ID.RA-3, ID.RA-4, ID.RA-5, ID.RA-6
	Risk Management [ID.RM]	ID.RM-1, ID.RM-2, ID.RM-3
Protect (PR)	Access Control [PR.AC]	RR.AC-1, PR.AC-2, PR.AC-3, PR.AC-4, PR.AC-5
	Awareness and Training [PR.AT]	RR.AT-1, PR.AT-2, PR.AT-3, PR.AT-4, PR.AT-5
	Data Security [PR.DS]	RR.DS-1, PR.DS-2, PR.DS-3, PR.DS-4, PR.DS-5, PR.DS-6, PR.DS-7
	Information Protection Processes and Information Procedures [PR.IP]	RR.IP-1, PR.IP-2, PR.IP-3, PR.IP-4, PR.IP-5, PR.IP-6, PR.IP-7, PR.IP-8, PR.IP-9, PR.IP-10, PR.IP-11, PR.IP-12
	Maintenance [PR.MA]	RR.MA-1, PR.MA-2
	Protective Technology [PR.PT]	RR.PT-1, PR.PT-2, PR.PT-3, PR.PT-4,
Detect (DE)	Anomalies and Events [DE.AE]	DE.AE-1, DE.AE-2, DE.AE-3, DE.AE-4, DE.AE-5
	Security Continuous Monitoring [DE.CM]	DE.CM-1, DE.CM-2, DE.CM-3, DE.CM-4, DE.CM-5, DE.CM-6, DE.CM-7 DE.CM-8
	Detection Processes [DE.DP]	DE.DP-1, DE.DP-2, DE.DP-3, DE.DP-4, DE.DP-5
Respond (RS)	Response Planning [RS.RP]	RS.RP-1
	Communications [RS.CO]	RS.CO-1, RS.CO-2, RS.CO-3, RS.CO-4, RS.CO-5
	Analysis [RS.AN]	RS.AN-1, RS.AN-2, RS.AN-3, RS.AN-4
	Mitigation [RS.MI]	RS.MI-1, RS.MI-2, RS.MI-3
	Improvements [RS.IM]	RS.IM-1, RS.IM-2
Recover (RC)	Recovery Planning [RC.RP]	RC.RP-1
	Improvements [RC,IM]	RC.IM-1, RC.IM-2
	Communications [RC.CO]	RC.CO-1, RC.CO-2, RC.CO-3

Figure A.2: NIST CSF Functions, Categories and Subcategories