Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Mai Hoang Tran

How the MITRE ATT&CK Framework can be used for Threat Modelling in the Cloud

Master's thesis in Computer Science
Supervisor: Daniela Soares Cruzes
Co-supervisor: Romina Druta
June 2023

NTNU
Norwegian University of
Science and Technology

Mai Hoang Tran

How the MITRE ATT&CK Framework can be used for Threat Modelling in the Cloud

Master's thesis in Computer Science
Supervisor: Daniela Soares Cruzes
Co-supervisor: Romina Druta
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

The primary objective of this thesis is to improve the security of cloud infrastructures by addressing vulnerabilities and weaknesses in an early stage, preferably in design phase. Specifically, the thesis proposes a method artefact as solution to address this issue, with the the intention of providing answers to answer following research questions:

- RQ1: How can the security findings from deployed cloud infrastructures improve the threat modelling process?
- RQ2: What is the effect of the proposed solution?

The artefact consist of a set of questions that have been derived from security post-deployment findings utilising a Cloud Native Application Protection Platform (CNAPP) tool. These questions can be applied during threat modelling process to identify potential weaknesses and implement security measures and strengthen security posture of the infrastructure early in the development cycle.

Design science framework is the methodology that has been used throughout the study. First step involves explicating the problem, followed by defining the requirements. Moreover, next stage is designing and developing the artefact. Then a demonstration is performed through a fictional case, and concluding with an evaluation and discussing the results. The evaluation consists of a survey that assesses the effectiveness the effectiveness of the artefact in solving the problem, usability, defined requirements, side-effects and comparison to other tools.

The survey results indicated a positive outlook on the artefact, with a majority of participants finding it helpful and perceiving an improvement in the threat modelling process, by gaining useful ideas for further brainstorming. However, it was also highlighted that the questions should be quality assured by a domain expert, in addition to lack cloud-specific terminology. Furthermore, it was acknowledged that the artefact requires more time to mature in order to be fully adapted and integrated into existing practices.

Sammendrag

Hovedmålet med masteroppgaven er å forbedre sikkerheten til skyinfrasttrukturer ved å identifisere sårbarheter og svakheter på et tidlig stadium, helst i designfasen. For å oppnå dette, foreslås en metode-artefakt som en potensiell løsning for å håndtere dette problem, med mål om å besvare følgende forskningsspørsmål:

- RQ1: Hvordan kan sikkerhetsfunn fra allerede implementerte skyinfrastrukturer forbedre trusselmodelleringsprosessen?
- RQ2: Hva er effekten av den foreslåtte løsningen?

Artefakten består av et sett med spørsmål som er utledet fra sikkerhetsfunn fra et CNAPP-verktøy. Disse spørsmålene kan brukes under trussemodelleringsprosessen for å identifisere potentiselle svakheter, implementere tiltak og styrke sikkerheten rundt infrastrukturen tidlig i utviklingsløpet.

Design science-rammeverket er metodikken som har blitt brukt gjennom hele studien. Første trinn innebærer å avklare problemet, etterfulgt av definere kravene. Deretter går man over til å designe og utvikle artefakten. Videre gjennomføres en demonstrasjon gjennom en fiktiv case, og avsluttes med en evaluering og diskusjon av resultatene. Evalueringen består av en undersøkelse som vurderer effektiviteteten av hvilken grad artefakten løser problemet, brukervennligheten, hvor godt de oppfyller kravene, mulige bivirkninger og sammenligning med andre lignende verktøy.

Resultatene fra undersøkelsen indikterte en positiv respons blant de som vurderte artefakten. Flertallet av deltakerne syntes den var nyttig og opplevde at den forbedret trusselmodelleringsprosessen ved at de fikk nyttige ideer for videre idémyldring. Imidlertid ble det også påpeket at spørsmålene ikke alltid var like tydelige, derfor burde det i kvalitetssikres av en fagekspert, i tillegg til manglende sky-spesifikk terminologi. Videre ble det erkjent at artefakten trenger mer tid for å modne før den eventuelt skal bli integrert i eksisterende metoder og praksis.

Acknowledgements

I would like to express my gratitude to my supervisor, Professor Daniela Soares Cruzes, and co-supervisor, Senior Infrastructure Engineer and Security Researcher Romina Druta for their guidance and invaluable feedback throughout this thesis.

I am also thankful to my friends for keeping me motivated but also providing occasional distractions when I needed them the most.

Lastly, I am grateful to my family for their unwavering support and encouragement throughout the years.

Table of Contents

List of Figures

List of Tables

1 Introduction

Cloud computing is an emerging technology that is rapidly growing in popularity, enabling businesses to speed up their operations, reducing costs and increase flexibility [1]. It can be observed a rising trend towards migrating to cloud infrastructures. An article from Gartner estimate world-wide end-users spending to a total $600 billion dollars in 2023 compared to $491 billion from last year [2]. However, the speed at which cloud adoption is increasing exceeds the pace at which security risks are being addressed. A report from IBM indicate the average cost of data breach to be approximately $4.35 million globally and more than twice in the United States. Close to one-half of all data breaches are exploited in the cloud [3].

The complexity of the cloud architecture with interconnected services and numerous capabilities that lack visibility [4], makes it difficult to identify security flaws. Netwrix's report from 2022 on *Cloud Data Security Report* addressed the biggest challenge for cloud adoption is integration with existing IT environment. In addition, they also report "unplanned expenses to fix security gaps" as mainly the biggest data breach consequences, experiencing an increase from 28% in 2020 to 49% in 2022 [1]. Therefore, the responsibility of improving the security posture in cloud infrastructure should be given more importance.

1.1 Motivation

Navigating the complex cloud landscape can be challenging, leading to making errors and mistakes. Organisations that used AWS as a cloud provider, suffered major data breaches impacting millions of users which turned to be misconfigured S3 buckets i.e. data storages [5]. This is not unexpected, considering an article from Gartner that states *99% of cloud security failures will be the customer's fault through 2025* [6]. Implementing security measures and maintain throughout the cloud infrastructure early on, helps to mitigate the attack surfaces and prevent such incidents from occurring.

DevSecOps, also commonly known as shift-left, *requires planning application and infrastructure security from the start* [7]. According to the principles of shift-left, planning is a crucial when securing the cloud environment. While leveraging automatic tools to scan for security vulnerabilities in deployed cloud workloads can be advantageous, it is ideally to conduct these scans prior to deployment, preferably in planning and design phase.

One of the key component in this life-cycle is compliance monitoring, which aids in identifying compliance violations and addressing them proactively. This thesis seeks to investigate a novel approach on how to plan and design a secure infrastructure using threat modelling.

Threat modelling is an approach that can handle threats and vulnerabilities at the planning or design stage, by enumerating through potential malicious scenarios and prioritise countermeasures accordingly. CNAPP is a type of automatic scanning tool that *address the full life-cycle protection requirements of cloud-native applications from development to production* [8]. Through leveraging these findings generated by CNAPP tools, it can be adopted in threat modelling processes to shift security to the left.

Despite the increasing concerns about cloud security, there has been little research focus on the topic of threat modelling in cloud, addressed by the systematic literature review

conducted from previous semester, see section A.1.

The motivation behind the pre-study was due to the lack of maturity in this field, which was implied from a previous SLR performed by Håkonsen & Ahmadi from 2021 [9]. Among the findings from the last semester's pre-study, were lack of cloud visibility, tool support and infeasible tools. In short, it was difficult to gain a clear overview of the cloud environment when performing threat modelling. Additionally, it was challenging for the practitioners to adopt to the suggested threat modelling techniques for cloud due to lack of tool support. Tool support serves the purpose that it can help to automate parts of the workflow, making it easier to adopt to a new procedure. Although, some of the techniques did incorporate tools, the guidelines were too abstract to comprehend, causing the tools to be considered infeasible.

One of the challenges identified during the pre-study for this master thesis is the integration of automatic tools into threat modelling, in a way that is applicable and practical for the practitioners. There exist cloud tools such as Cloud-Native Application Protect (CNAPP) tool, that identify and mitigate cloud risks by continuously monitoring cloud infrastructures and detecting threats and misconfigurations. This details of CNAPP tools are discussed in section 2.4. Furthermore Mitre ATT&CK framework was recognised as a great knowledge base and compliance framework. The findings indicated a good level of precision, indicating that it could be utilised to categorise the issues.

1.2 Objective and research questions

The goal of this thesis seeks to define an approach for threat modelling in cloud based on the security findings of automatic tools. This proposed "shift left" approach is to leverage the security findings detected by the tools after deployments, by integrating it into an effective threat modelling strategy. This helps to gain early visibility into potential security issues and address them proactively. By adopting this strategy, it becomes possible to identify and mitigate vulnerabilities at an earlier stage in the development life-cycle, potentially preventing them from occurring.

Figure 1 depicts the process of when the artefact can be applied in a threat modelling session. OWASP 4 question framework[1] is used as an example of any arbitrary threat modelling method.

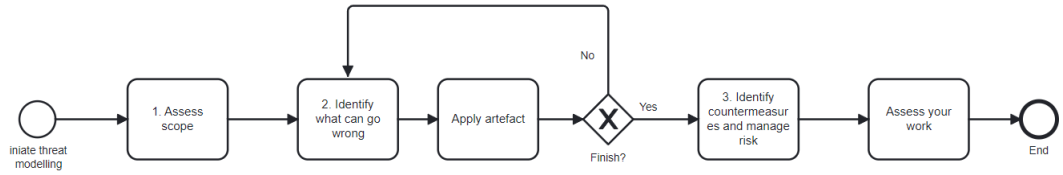


Figure 1: Modified OWASP 4 question framework on threat modelling

From the pre-study there is already addressed a gap in terms of lack of research on this topic, which this thesis aims to contribute to. Essentially, this thesis seeks to answer the question: *How can we prevent cloud vulnerabilities in an earlier stage?*.

The research questions can be summarised and dissected into:

[1] `https://owasp.org/www-community/Threat_Modeling`

1. RQ1: How can the security findings from deployed cloud infrastructures improve the threat modelling process?
   (a) Which questions can be asked in a threat modelling session?
2. RQ2: What is the effect of the proposed solution?
   (a) To what extent is it effective at solving the problem?
   (b) What is the usability?
   (c) How well does it satisfy the requirements?
   (d) How is it compared to similar approaches?
   (e) What side effects does it pose?

1.3 Contribution

More specifically, the contribution will be in terms of an artefact created to address cloud vulnerabilities earlier through threat modelling. The artefact consists of a set of questions derived from prior and frequent issues in the infrastructure detected by a CNAPP tool. The goal is to promote a discussion about spotting and identifying security flaws early on, and resulting the team to take proactive actions and implement necessary and preventive measurements.

1.4 Methodology outline

The Design Science framework written by Johanesson and Perjon [10] has been used as the methodology for creating the artefact questions. Evaluation on the artefact has been carried out through a voluntarily and anonymous survey on the artefact. The data collection on existing cloud infrastructures has been done in cooperation and agreement with the Norwegian software and information technology company Visma[2].

1.5 Thesis outline

The thesis has been structured into a background chapter that will present relevant context to general threat modelling, last semester's pre-study, threat modelling in cloud and CNAPP tools. Next chapter will elaborate on the methodology that have been used throughout the research and describe key activities that have been carried out. Moreover, the following chapter consists of how the artefact has been implemented and how it can be applied in threat modelling through a simulation. Subsequently, the next chapter present the evaluation results obtained through a feedback survey. Subsequently, the evaluation results will be discussed in this chapter. Finally, the thesis ends with its conclusion and implications for future research and work.

[2] `https://www.visma.com/company`

2 Background

This section will provide a sufficient context and background related to the research for this thesis. Firstly, introducing the threat modelling framework, subsequently followed by the pre-study from previous semester and threat modelling in cloud. Next section covers cloud computing and its technology. Finally, a small section is devoted to cloud tools that are used to secure the cloud assets, namely Cloud-Native Application Platform Protection (CNAPP).

2.1 Threat modelling

Threat modelling is a process containing multiple steps that can be incorporated into the development cycle, at an early stage, in order to identify potential threats and vulnerabilities or lack of security measurements. Adam Shostack, author of *Threat modelling: Designing for Security* frames it as a technique that breaks down the main activity into sub-goals that are achieved at each phase [11]. Achieving objectives in each step gradually help to reduce attack surfaces and resolve new arising challenges.

There are numerous ways to engage in threat modelling, either adopt parts of the techniques or completing the entire process. Some of the techniques that have been published are STRIDE [12], PASTA [13], Trike [14], OCTAVE [15] and OWASP threat modelling approach[3]. The motivation is to shift part of the responsibility for improving security posture when designing and implementing the system in order to make it more secure and reliable. This is accomplished by encouraging the team to consider security when developing and enumerating potential malicious scenarios.

2.2 Pre-study on Threat modelling in Cloud

Previous SLRs by Tuma et al. [16] and Håkonsen & Ahmadi [9] addressed the issues of the lack of cloud threat modelling techniques as one of the implications on future research. The motivation behind this study was therefore to investigate and understand the current practices and challenges in the cloud domain, with the aim to identifying areas for improvements on cloud threat modelling. To achieve this, a systematic literature review was performed last semester on *Threat modelling in Cloud*, see section A.1.

This systematic literature can be viewed as an extension of the SLRs of Tuma et al. [16] and Håkonsen & Ahmadi, adhering to their research strategy but only filtering on cloud domain. There was a total of six techniques from Tuma et al. [16] and Håkonsen & Ahmadi [9] from 2013-2021. Additionally, five new ones were identified in the pre-study between the span from 2021-2022, which indicate a promising trend for threat modelling in cloud. However, it is too early to conclude due to the small sample size as the field needs more time to mature.

Approximately half of the techniques were supported with tools. Although some techniques had incorporated tools into the techniques, it was still difficult for practitioners to adapt due to not precise guidelines that also required a bit of prior security knowledge. Additionally, the input representation for threat modelling, mainly in terms of model-based or textual description, did not have a structured or systematic way of representing

[3] https://owasp.org/www-community/Threat_Modeling_Process

the cloud environment. This made it sometimes difficult to gain a consistent and visibility of the cloud infrastructure. In essence, it was discovered that the techniques were still immature and needed further research before they were applicable in practice.

2.3 Threat modelling in cloud

Cloud computing technology has experienced a rapid growth over the past years. A report from Gartner forecast a 23.1% growth in cloud end-users, accumulating a value of $332.3 billion compared to $270 billion in 2020 [2]. Industries are still undergoing a transition towards cloud-based infrastructure, and the market is expected to expand even further in the foreseeing future. The cloud computing sector is continuously growing, resulting in more attack surfaces. This comes with the responsibility of improving the security posture of the cloud infrastructures. Most cloud breaches are rooted in the misconfigurations [6] which could have been prevented in an earlier stage. Therefore, threat modelling is a suitable framework in identifying emerging issues.

One of the challenges in cloud is the speed at which everything is happening that the development and deployment of software applications can happen almost instantly. The fast-paced continuous integration (CI) and delivery (CD) are accelerating so fast that security issues are often overlooked [17]. Younas et al. raise concerns about security threats as one of the obstacles in cloud computing [18]. Ghani et al. discuss the role and responsibilities for secure software development and indicate involving a security expert in agile development is considered as overhead [19].

Thus, the concept of *shift left* has been broadly discussed to integrate security earlier in the development cycle, with the development cycle being visualised moving clockwise. The implications are to implement security and policy earlier, preferably in design phase before they are shipped to the code. This approach is crucially needed, to keep up with the rapid software development [17]. To effectively address this, it is essential to incorporate more automation, security and network capabilities into the application in order to monitor, detect, orchestrate and automate the cloud infrastructure. This thesis is focused on exploring "shifting left" in terms of threat modelling in cloud, by combining security findings from automatic scanning tools.

Another challenge is the cloud complexity that is identified in the infrastructure with its interconnected services and resources. One of the reasons being the numerous capabilities it can provide across several interconnected services, increasing the complexity significantly. Automatic scanning tools are necessary as detecting this manually in the system can be a difficult and tedious task. More precisely, to achieve the goal of shifting security to the left in the development cycle, the thesis will make use of the results from CNAPP tools.

2.3.1 Cloud computing

The National Institute of Standards and Technology (NIST) has developed standards and guidelines for the cloud computing paradigm to ensure adequate security information for the industry [20]. According to their definition of cloud computing technology it should provide a broad pool of computational capabilities that should be configurable to control the resources, memory spaces etc., and additionally having a ubiquitous network access. Service and resource provisioning is swiftly executed and released for deployment with

little to close to no management and hassle. Furthermore, five essential characteristics can be identified as *on-demand self-service*, *broad network access*, *resource pooling*, *rapid elasticity* and *measured service* with the capabilities of:

- Scaling and paying resources accordingly to e.g. traffic and usage.
- Distribute resources and applications across the network to make it more accessible for the global users.
- Lower IT cost and effort to installing, configuring and manage the infrastructure with seamless deployment.

The service models consist of Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS), and the deployment models include private, community, public and hybrid cloud. Control and visibility depend on the selected service and deployment model which brings multiple layers of complexity into the mix. One of the limitations identified in the pre-study, given in section A.1, is the difficulty in gaining a clear overview of the entire cloud environment, considering various aspects and components involved. The following section will discuss more about how the responsibility has increased on the engineer's end and briefly about the cloud visibility issue.

2.3.2 Cloud Native

Cloud native technology is defined by the Cloud Native Computing Foundation (CNCF) to help organisations to adapt to cloud computing. This includes providing a wide range of techniques and services including, containers, service meshes, mircoservices, immutable infrastructure and declarative APIs. The goal is to enhance scalability, elasticity, resiliency and flexibility within a dynamic cloud environment [21]. These capabilities accelerate the development cycle and allow rapid deliveries. With all this leveraged, the responsibilities are also significantly increased on the engineer's end. The engineers need to have sufficient knowledge about each services to configure and secure it correctly. In the past, monolithic application was the norm but the emerging trend of cloud native design has replaced this traditional approach [22].

Figure 2 and Figure 3 depict the differences between the design. One can observe that the modules in a monolithic application are more tightly coupled together. Additionally, the communication flow between the clients and modules are more direct, and then the requests are sent to a database. In a cloud native application, all requests must pass through a single entry point, known as an API gateway, which then redirects them to their designated microservices. From there, the requests are placed on an event bus. The event bus continue to perform an action depending on how it was configured. This asynchronous event-driven architecture serves as the backbone of the cloud native technology that glue its interconnected microservices together.

While cloud computing brings many benefits, there is also a learning curve involved in understanding the underlying properties of cloud resources and how they interact. Since the resources are decoupled, it is necessary to secure each of them independently. This is important to consider when designing a secure cloud infrastructure.

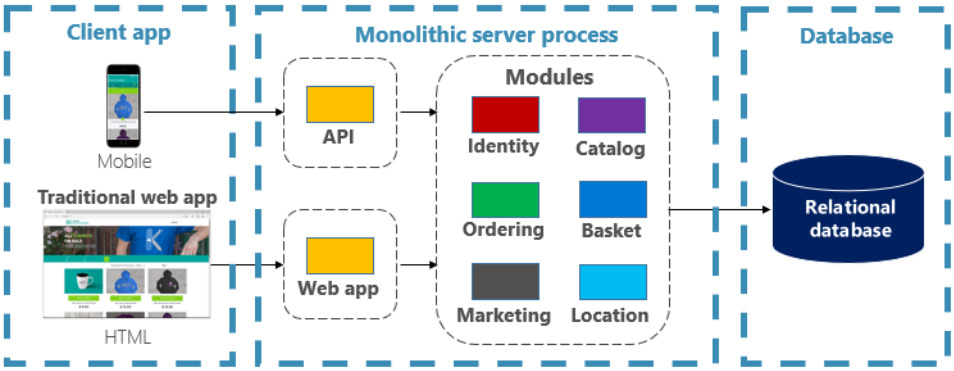


Figure 2: Monolithic application in traditional IT system design

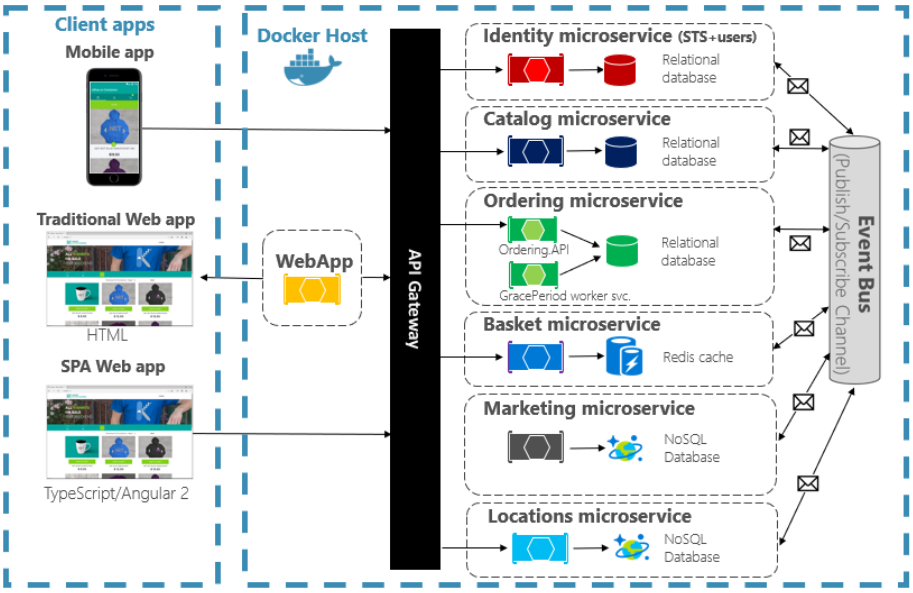


Figure 3: Cloud native design

2.4 Cloud-Native Application Protection Platform (CNAPP)

Cloud-Native Application Protection Platform (CNAPP) automates the process of identifying and remediating risk in the cloud, by continuously monitoring cloud infrastructure and detecting threats and misconfigurations in the cloud environment such as IaaS, SaaS and PaaS.

CNAPP tools ensures compliance by inspecting and comparing the cloud infrastructure to a set of best practices or industry standards. They are vital in enhancing cloud security posture management as part of shifting left, by detecting and correcting misconfigurations [23]. The key capabilities a CNAPP tool provides are [24]:

- Control
- Compliance assurance
- Monitoring
- Insight and visibility

Control: Predefined cloud security policies play a crucial role in ensuring that current services and resources remain compliant, even in dynamic changes in the cloud environment. These policies can be modified and adjusted to meet the specific needs of different management groups, subscriptions, or tenants, allowing for greater adaptability in maintaining compliance.

Compliance assurance: It can be integrated and configured into current system and existing platforms, allowing organisations to follow industry-derived compliance standards. This integration enables the automatic identification and investigation of threats, as well as recommended remediation steps for mitigation.

Monitoring: Continuous scanning of cloud resources is essential for obtaining an up-to-date overview of their current state and promptly identifying abnormal activities. To follow up on mitigation strategies, organisations can strengthen security access control measures by implementing customised configurations tailored to their specific resource requirements.

Insight and visibility: Provide deeper insight of the current state of resources and recommendations on improvements. Comprehensive analysis and scanning contribute to identification and connecting various risk factors, thereby reducing attack surfaces and adding an enhanced visibility of the cloud.

Based on the findings of the preliminary study, it was identified that there is insufficient visibility into the cloud environment and lack of tool support in threat modelling. The objective of this thesis is to address this gap by using a CNAPP tool to gather and analyse data, and subsequently integrate it into the threat modelling process. The utilisation of this tool will be regarded as a means to shift focus towards earlier stages in cloud threat modelling.

2.4.1 MITRE ATT&CK compliance

MITRE ATT&CK is a curated knowledge base designed to improve detection of malicious behaviour based on real-world observations from various cyber adversary attack stages and life cycles. It is a framework organised into tactics, techniques and procedures (TTPs) that can be used as a foundation for developing threat models and methodologies across domains [25].

Tactics denote the adversary's intended goal to perform the attack, while *techniques* reflect the high level details on how the attacker is achieving the tactic. *Procedures* describe the techniques in-depth with detailed specifics. The knowledge base is highly accessible to any individuals regardless of background. Furthermore, it updates frequently with the latest security events, improving the understanding of evolving attack techniques.

Based on pre-study findings, it was found techniques that used the MITRE framework had a higher level of precision compared to other techniques. It was determined that the framework holds significant potential and could be used as a compliance framework for threat modelling in the cloud. Based on this compliance, it was identified security issues of different systems that were using cloud services.

3 Research Methodology

This section outlines the research methodology conducted in this thesis. Design science was chosen as the framework to approach the research questions introduced in section 1. The research strategy consists of several main steps and the following subsections will provide a brief overview of each stage and how they fit into the methodology framework.

3.1 Design science

The core elements of Design Science encompass the *practice*, *people* and *problem*. In this context, a practice refers to a set of activities regularly performed by individuals with the goal to solve a practical problem. The procedure can be broken down into a sequence of stages that lead to the creation of an artefact. This artefact serves as a tool to assist the people engaged in the practice to solve the practical problem. That could be either improving existing solutions or innovate new ones. In essence, Design Science is a scientific study and the creation of artefacts that are developed and utilised by people to address practical problems of general interest [10].

Design Science takes into account the generalisability of solutions within the problem domain. This means that the outcomes derived from Design Science can be applied in a broader global context. Moreover, Design science emphasises on sharing the knowledge to a wider research community within its respective fields [10]. In addition to these distinctions, three additional requirements characterise the purposes of Design science:

- The use of rigorous research methods
- Relate the produced knowledge to existing knowledge base
- New results should be communicated to both practitioners and researchers

This is to highlight the importance of conducting research methodology in a systematic and rigorous manner, ensuring not only the reproducibility of the study but also be able to facilitate knowledge through iterative learning. Disseminating and sharing the results with both practitioners and researchers is equally vital for the field. Publications serves as a means to convey the findings. By making the research methodology transparent and accessible, other researchers can apply the same approach and acquire similar results. This promotes the reliability and validity of the research findings contributing to increase the knowledge within the scientific community.

The first requirement is covered and further detailed in the next subsection. Lastly, the results from the thesis will be presented and discussed in later sections, achieving the remaining requirements.

3.2 Design Science Research Strategy

The framework of Design science, published by Johanneson and Perjons [10] consist of five main activities; *explicate problem*, *define requirements*, *design and develop artefact* and finally *evaluate artefact*. These activities forms a rigorous research methodology, ensuring a systematic approach throughout the process. Design science is an iterative process,

allowing researchers to move between activities both backward or forward. Figure 4 demonstrates the steps for each activity. It is important to note that these activities do not need to be strictly sequential.

Moreover, it is possible to improve the artefact by iteratively performing the activities over a continuous period without starting from scratch in each iteration. For instance, if the problem explication and requirements are sufficient, there is no need to explicate them or define new ones again. Instead, the primary focus should be on developing the artefact until it meets the established requirements.

In this thesis, the primary focus will be on *Design and Develop Artefact*, *Demonstrate Artefact* and *Evaluate Artefact*, mainly due to time constraints and limitations of the study. In addition, the pre-study outlined in section 2.2 was conducted in the previous semester which greatly contributed to clarify the problem explication and determining the necessary requirements.

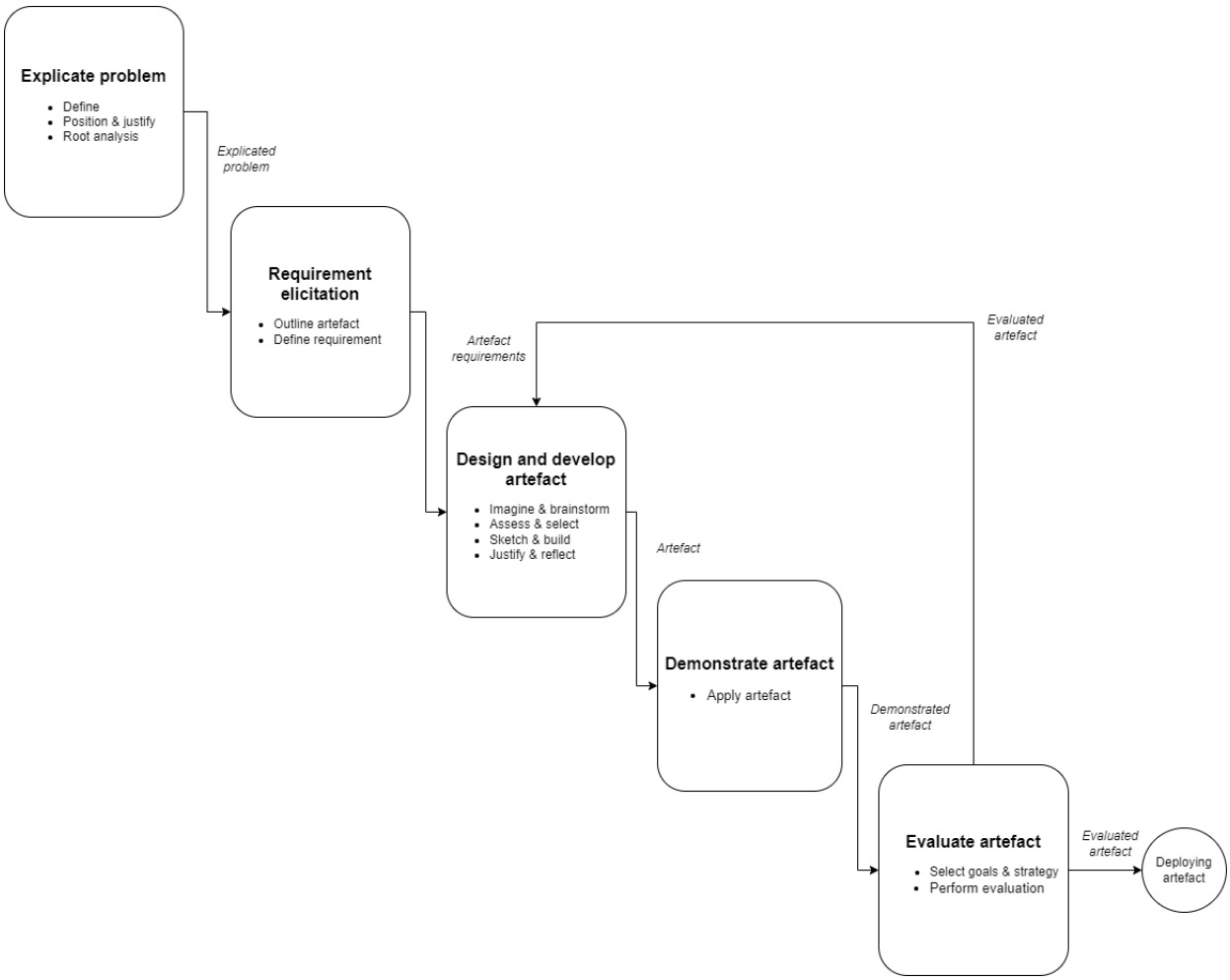


Figure 4: Outlined sub activities and tasks in Design science process

Explicate problem
A problem is, as stated by Johanesson and Perjons [10], a gap between the current and a desirable state. Thus, the goal is to clearly define what is lacking and the factors that are causing the gap. Therefore, it is important to provide context to help the readers understanding the problem from a new perspective. The problem should be presented in a way that is not too specific or niche, but rather of interest to a wider audience.

One important condition when investigating the problem is to conduct a *root cause analysis*. This involves to investigate into the underlying causes of the problem to gain an understanding of its scope, beyond the surface-level issues. By identifying and analysing the root causes, a deeper insight into the problem can be achieved.

The initial step of the framework is to define the problem statement as precisely as possible by justifying its importance and to investigate its underlying causes. This stage will lay the foundation for the subsequent activities and provide a dependent input for the next step. The main sub-activities within this step can be further broken down into:

1. Problem definition
2. Justifying the problem
3. Identifying its root causes

Defining the problem precisely will reduce the risk of being misinterpreted and establish a common view of the problem. However, abstracting too much can make it harder to grasp, and thus missing out on important aspects and details. To mitigate the risk, involving multiple stakeholders in the discussion is valuable. By including inputs and insights from diverse perspectives, helps to narrow down the problem definition. This approach helps to ensure a holistic understanding of the problem's scope and facilitates the identification of key elements.

Another aspect to consider is positioning and justifying the problem. This involves integrating the problem within its relevant context to help others understand why it is significant to address. One effective approach is to view the problem from a practical standpoint, taking into account various factors such as stakeholders, related activities and surrounding environment. By providing this contextual information, it becomes clearer why addressing the problem is essential and how it relates to the broader picture.

Finally, to gain a more detailed understanding of the core problem, instead of just looking at the undesirable current state, a root analysis can be performed to identify, analyse and group the root causes. For instance, a diagram can be used to visualise and present these root causes.

Requirement elicitation
The goal of defining requirements is to establish a blueprint of the artefact that will be created. This can be done through firstly determining the type of artefact, such as; *construct*, *model*, *method* or *instantiation*. Once the type is determined, the next step is to elicit the necessary requirements that are important to the stakeholders. A requirement can be viewed as a desirable property or characteristics of the artefact that the artefact possess. The requirements can range widely from functional, structural, environmental or non-functional.

Design and develop artefact
This activity focuses on the process of designing and developing the artefact, which the input from previous steps come to fruition. The steps are broken down into *Imagine and brainstorm*, *Assess and select*, *Sketch and build* and *Justify and reflect*, which contribute to design and create the artefact.

During *Imagine and brainstorm* phase, ideas are generated either individually or collaboratively in groups, using various methods such as brainstorming sessions, workshops or

interviews. The ideas are then forwarded to *Assess and select* phase to be assessed and selected for the next stage. This stage helps to narrow down the solution space and focus on the most viable options that align well with the defined requirements. Moreover, once the ideas are assessed and selected, *Sketch and build* phase begins. The artefact is sketched based on the selected ideas either by designing use case diagrams, user stories or storyboarding. The final sub-activity, *Justify and reflect*, centers around providing a design rationale. The design rationale helps to communicate the thought process and understanding of the artefact's design decisions.

Demonstrate artefact
The next step concerns how the artefact is applied and validated in a specific use case, addressing the explicated problem as described earlier. It is measured by how well it functions in the use cases in terms of descriptive knowledge, but also why it works through explanatory knowledge. The sub-activities consist of choosing and applying the use case which it can be demonstrated in.

Evaluate artefact
The main objective of the final step is to evaluate how well the artefact solves the problem and to which degree it satisfies the requirements. This is measured through six goals:

- Effective at solving the problem
- Requirements evaluation
- Usability
- Artefact comparison
- Side effect investigation
- Formative evaluation

3.2.1 Explicate problem

Understanding the expanding market for cloud computing, is an important factor to recognise why it is necessary to address the cloud issues. A market which is forecasted to reach nearly $600 billion dollar in 2023, compared to $491 billion dollar in 2022, making a growth of 21.7% [2]. This growth signifies that businesses are increasingly adopting cloud services, leading to a rise in cloud vulnerabilities and attack surfaces. Consequently, the demand for cybersecurity services are also expected to surge in order to meet these needs.

Problem definition
To mitigate potential risks, it is crucial to adopt to proactive approaches and embrace a security mindset early on in the development cycle. One such practice that embodies security from the start is threat modelling. By incorporating threat modelling into the development process, security flaws can be prevented *before* occurring rather than detecting and responding *after* it has occurred.

As described in section 2.2, a systematic literature review was conducted last semester on the topic *Threat modelling in cloud*, among the findings were lack of tool support and the feasibility of the tools. There exist tools in the industry that do scan through systems and will provide a wide coverage and overview of the vulnerabilities. Although, these applications are more targeted toward networks and web applications with minimal attention on cloud infrastructure.

Referring to section 2.4, CNAPP tools appear to be a promising outlook. However, when it comes to research publications on cloud tools and their effectiveness in identifying security issues, there is clear lack of published literature, especially in comparison to other domains [26] [27]. Consequently, the *undesirable state* can be understood as lack of research on CNAPP tools and *desirable state* involves exploring novel approaches to leverage these tools in a new context. This can help to address some of the implications of future work highlighted in the pre-study, motivating the proposal of an artefact in this thesis. The forthcoming chapters will provide a detailed exploration of this artefact and its role bridging the gap between the current and desired states.

Justifying importance
The problem is still not clearly defined yet, as it can still cover lots of different research areas related to CNAPP tools. Therefore, it is necessary to narrow down the scope further. The question that remains is *how* these tools can be effectively used to detect cloud issues. Automated scanning tools yield numerous security findings which can be difficult to detect prior to this. The clue is to "left shift" this to an earlier stage in the development cycle. Hence, the problem formulation can be refined to: Can we detect the cloud vulnerabilities at an *earlier stage*?. "Earlier stage" relates to the application of threat modelling in conjunction with the findings from a CNAPP tool that detects cloud vulnerabilities. Consequently, this will improve the maturity of Threat modelling in the cloud. Therefore, a suggested solution to the problem statement involves integrating CNAPP tools into the threat modelling process.

Root cause analysis
A root cause analysis inspired by Ishikawa fishbone diagram on Quality control was performed in a similar fashion, but applied on cloud computing [28]. The different categories can vary but the main ones used in the analysis include; *People*, *Process*, *Technology*, *Environment* and *Third party*. Some of the causes was a result of the pre-study and frequent meetings with supervisor, infrastructure engineer and architects at Visma, but also briefly peeking at main findings outputted from CNAPP tools.

An additional category, *Third party*, emphasises the perspective from the client side. When cloud vendors are affected by incidents it can also impact the clients as well, and therefore including cloud vendors as a third party actor. Additionally, insider threats within the team and compromised software supply from third-party agreements can potentially introduce cloud vulnerabilities, such as outdated software versions. However, this thesis will mainly focus on the root causes for *People*, *Process* and *Technology*, and partially *Environment* to reduce the number of attack surfaces. The Figure 5 serves as an overview of the very high-level root cause analysis.

People represents the stakeholders that are involved when detecting the cloud vulnerabilities. The software developers are responsible for creating and building the system, and are consequently also accountable for the vulnerabilities. The same applies for architects and infrastructure engineers which are respectively in charge of designing and deploying the system into production. Misconfigurations and oversight issues that snowball into exploitable vulnerabilities are plausible scenarios.

Process describes which policies, procedures or guidelines that are in place to mitigate the vulnerabilities. If an incident should occur, a good start is to investigate the process that is implemented and how it appeared in the system.

Third-party summarise which external vendors or partners that are associated with the issue. This can be used to filter out which relevant parties that the vulnerabilities origi-

nated from or contaminate assets that are potentially exposed.

Environment includes infrastructure, software, hardware implementations that are surrounding the applications and systems are running on. This is specifically more targeted towards the environment regarding cloud resources. In other words, this include which services are interconnected, data flow, their capabilities in order to map out possible outcomes for potential threat actors.

Technology details which level of abstraction the technology are applied to. It is roughly dissected into following categories; application, integration, network and infrastructure for simplicity. Separating this into different categories can help to identify which area and responsibilities the team can focus on. However, it is important to acknowledge that the cloud is complex, and categorisation alone cannot capture its entirety. The categorisation should take into account the inherent complexity of the cloud and avoid excluding other options or perspectives.

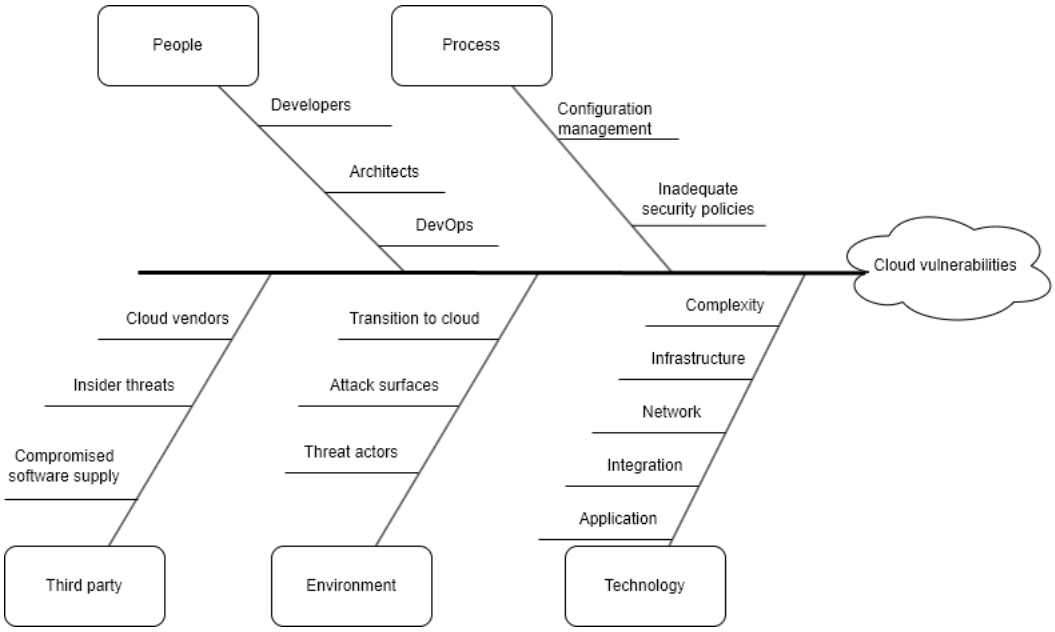


Figure 5: Identified root causes represented in a fishbone diagram

3.2.2 Requirement elicitation

The type of the artefact is identified to be a *method artefact* which is described as defining guidelines and processes to achieve goals [10]. The reason being the artefact is intended to present a set of questions to assist the team in a threat modelling process.

By allowing users to freely ask these questions, the artefact encourages open discussions and facilitates the identification of potential flaws during threat modelling sessions. Thus, the artefact can be justified as a contribution to defining new processes aimed at mitigating vulnerabilities at an early stage and addressing low-hanging fruits. Additionally, the artefact is particularly useful for the team during initial brainstorming as the questions can spark ideas used to discuss different topics and related issues.

Consequently, the next step in the research strategy is to define the requirements tailored around this type of artefact. The goal is to elicit the requirements that effectively address the explicated problem [10]. A crucial aspect, is to include the perspectives and inputs

of stakeholders who play a significant role in defining these requirements. As part of explicating the problem in section 3.2.1, the stakeholders were also involved in discussing the requirements through the regularly meetings. These meetings identified the desirable properties that the artefact should entail, and the requirements were elicited accordingly. This has resulted into functional and non-functional requirements that can be depicted in Table 1 and Table 2.

Functional requirements describe the relation between the input and output of the artefact, essentially how it should behave and which features it should entail.

Non-functional requirements are related to capabilities and constraints, describing the desired properties of the functional requirements.

Table 1: Functional requirements

ID	Requirement
FR1	Different types of visualizations should showcase the statistics accumulated by the tool.
FR1.1	Displaying failed checks relatively compared to the total amount.
FR1.2	Displaying failed checks frequency by cloud provider.
FR1.3	Displaying failed checks frequency by tactic.
FR1.4	Displaying failed checks frequency by sub-tactic.
FR1.5	Displaying the distribution of the severity by cloud provider.
FR1.6	Displaying the frequency of the failed checks by asset type.
FR2	Questions can be extracted from the processed dataset.
FR2.1	It is dictated by the most frequent failed checks sorted by each asset type.
FR2.2	The set of questions are extracted for each cloud provider.
FR3	The questions should assist in threat modelling sessions to address cloud vulnerabilities

The functional requirements primarily focus on presenting visualisations that depict various types of statistics, providing an overview of the findings identified in the dataset. These visualisations aim to offer deeper insights into the meaning behind the extracted questions, for the curious users.

Firstly, a relative comparison of failed checks can give users an understanding of the overall frequency of vulnerabilities. Another valuable insight is examining the distribution of vulnerabilities across different cloud providers. Additionally sorting the frequency based on the MITRE ATT&CK category and sub-category can help users comprehend the most prevalent types of vulnerabilities across projects, addressing the low-hanging fruits.

Considering the severity of identified vulnerabilities is also important. Sorting the failed checks by severity, specifically within each cloud provider, allows users to grasp the crit-

Table 2: Non-functional requirements

ID	Requirement
NFR1	The data should be pre-processed and cleaned in a format that easily and effecitvely can be used for further analysis
NFR2	The data should be validated and tested to ensure the correctness of the outputs.
NFR3	The artefact questions should be easily understandable
NFR4	It should efficiently process the dataset in a reasonable waiting time

icality of these vulnerabilities, as different cloud providers have their own cloud concept and does not necessary share the same terminologies. Filtering out severity levels labeled as "Low" ensures that only highly prioritised issues are included when extracting the questions. Subsequently, the extracted questions are grouped by asset type, sorted by the most frequent issues, and the top ones are selected. These selected questions from a set of questions specific to each cloud provider, which can be utilised during any threat modelling sessions.

By fulfilling these functional requirements, the artefact will enable users to gain valuable insights from the dataset and effectively support threat modelling activities.

3.3 Data collection methods

The data collecting and methods used for the research will be discussed in this chapter. The dataset used to develop the artefact is collected through the CNAPP tools. Specific choice of tool to use is explained in the following sub section. Evaluation and feedback on the artefact are gathered through survey questionnaires from volunteers in the industry.

3.3.1 Choice of CNAPP tool

Two CNAPP tools were tested and compared to determine their suitability for the thesis project and their alignment with Visma's preference. The findings from the tools was presented during an internal meeting with the relevant stakeholders represented from Visma. The tools used in this analysis have been anonymized to respectively *Tool 1* and *Tool 2* to ensure the confidentiality of the company's information.

A comparison analysis between the tools was conducted and performed to reach a conclusion. The requirements from Table 1 and Table 2 were used as baselines to compare the results outputted from both tools. The main differences can be summarised into following bullet points:

- Total amount of controls
- Different categorisations, formatting and data clustering

- Different ways of enforcing the controls

Tool 1 had a significant more amount of controls than Tool 2, leading to them having a greater amount of findings compared to Tool 2. A comparison of the controls is depicted in Figure 6. The size of the datasets from both tools differed widely, a scan between a period of approximately 2.5 weeks yielded Tool 1; 48 255 failed checks and 3 869 failed checks by Tool 2 as can be seen in Figure 7. It is important to note that the comparison is relative *within* each tool and not *between* the tools, as this can be misleading due to the juxtaposed figures.

One observation is that Tool 2 is relatively detecting more failed checks than Tool 1 which will be soon elaborated. Initially, one might assume that Tool 2 has higher accuracy compared to Tool 1. However, upon closer examination of the raw data, it was observed that the findings were tagged with multiple tactics, indicating a less strict identification of the results as failed compliance. This increased the likelihood classifying a finding as a failed check due to its association with multiple tactics. Such clustering resulted in a less granular dataset from Tool 2, requiring additional steps for processing and refinement. This ultimately gave Tool 1 the edge.

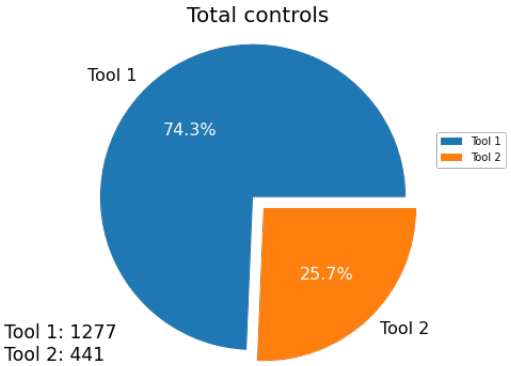


Figure 6: Relative failed check comparison between the tools

Another notable difference was that Tool 2 categorised Kubernetes as a cloud provider, which can be explained that Kubernetes is a platform that runs cloud resources on different cloud providers. This distinguishing had a minimal impact in overall, but it did rule in favour to Tool 1 as it was not a representative of the cloud vendors used in the Visma projects, which consisted of AWS, Azure and GCP.

Additionally, working with the data from Tool 2 proved to be more tedious as it required decoupling and further processing steps for data analysis. This made Tool 2 a less favourable tool compared to Tool 1. Ultimately, the decision to choose Tool 1 was influenced by their provision of more granular data in an easier-to-process format. Furthermore, Tool 1 offered better usability and features, providing more valuable insights. While there were other arguments and justifications that influenced the decision, they were not directly related to the data quality from the tools.

3.3.2 User testing

The purpose of user testing is to engage the end-users to evaluate the artefact against the goals described above. In this context, the target user is identified as any participants

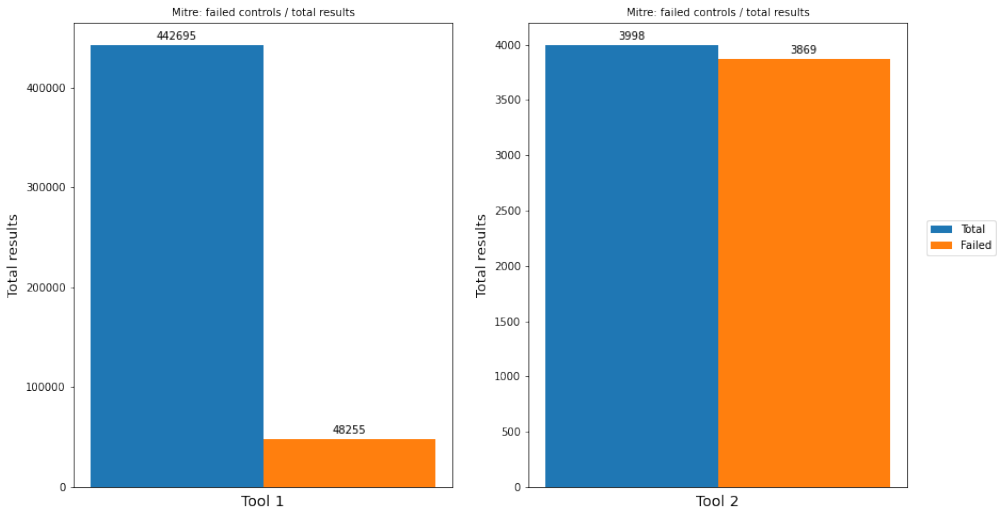


Figure 7: Relative failed check comparison

that are building their systems running in cloud. This could include a range from project managers, designers, architects, infrastructure engineers, security engineers, and more.

The evaluation of the artefact will consist of a usability test, carried out as an *ex ante evaluation*, meaning it will be evaluated without being used or fully deployed. This is due to practicality and feasibility, such as finding a suitable projects or systems to apply threat modelling on it. Testing it in a design phase or early development is generally more optimal, but time consuming to complete.

Additionally, it addresses challenges related to accessing confidential information from organisations and within the team's projects, as they typically disclose only minimal details as possible. The process of reaching an agreement with organisations can be time-consuming. However, due to the limited time frame, conducting real threat modelling scenarios was deemed infeasible. Therefore, the user testing will primarily consist of participants reviewing the artefact questions and completing a survey at the end.

What is usability?
Joseph Dumas and Janice Redish, the authors of *A practical guide to usability testing*, defines usability as *People who use the product can do so quickly and easily to accomplish their own tasks* [29]. In this thesis, the product is considered as the generated artefact and its main goal is to identify cloud vulnerabilities at an earlier stage as stated in section 1. The evaluation is to measure how well it is able to achieve its purpose, according to the users. Furthermore, the usefulness of a product is determined by four key points [29]:

- Usability means focusing on users
- People use products to be productive
- Users are busy people trying to accomplish tasks
- Users decide when a product is easy to use

Focusing on users

In order to evaluate the artefact, it has to be tested and understood by the users. Therefore it is important that the testers represent the actual users. To gain feedback from the users of the artefact, a survey has been created to recruit interested readers in an internal channel within Visma to seek out potential users.

People use products to be productive
An intuitive measurement for people to determine the usability is described in the book as *in terms of the time it takes to do what they want, the number of steps they go through, and the success they have in predicting the right action to take* [29]. To put it another way, the users should be able to quickly and simply learn how to utilise it within a reasonable amount of time. Therefore the users should feel an increase of productivity when applying it. Thus, it is important to understand the user's performance goals. How can we help them to perform their tasks quicker, or possibly automate it?

Users are busy people trying to accomplish tasks
Additionally, the artefact has to solve the user's daily tasks effectively, as it should help the users with their work. However, along with people's concern with productivity, the time to adopt should not be too long that discourage people for learning it. Thus, the emphasis should be on make the user spend as less time possible learning the artefact.

Users decide when a product is easy to use
Only the users themselves determine how usable the product is, not the designers nor the developers of the product. The factors depend on how much the users are willing to put the effort and time to gain the benefit of it. The learning curve should not be steep, but a balance between learning and utilise its functionality. Particularly, it should be *consistent, predictable and easy to use* [29].

A survey was created to inform potential users on the artefact prototype. A brief and sufficient background information was provided on threat modelling, cloud issues, MITRE ATT&CK framework, the CNAPP tool and the project itself to inform the readers on the necessary details. It is important to note that the survey was done voluntarily. This is to attract users that are most enthusiastic to use it, particularly early adopters. In that way, the hopes is to have an accurate representation of the users that are willing to try. It is favourable to attract the early adopter that can give valuable and motivated feedback in hopes that it can improve the artefact further.

The survey consisted of two sets of questions. The first set aimed to gather demographic information about the participants, including their professions, experience, security knowledge, familiarity with threat modelling, and cloud provider preference. This information helps in obtaining a representative sample of testers and provides insights into their backgrounds. Additionally, the data from the feedback can be correlated with these meta metrics and analyse the evaluation in more depth. The metadata questions are depicted in Table 3.

The final set of questions consisted of a questionnaire about the artefact and how it should fulfil the evaluation goals described in section 3.2. The main survey questions were designed to follow a modified 5-point scale, ranging between *Disagree, partially disagree, neutral, partially agree and agree*, originally a 7-point scale suggested by Ajzen and Fishbein [30]. This was to narrow down the scale in a way that it did not feel too granular and made it easier for the user to select an option. Thus, the questions were arranged neatly in a tabular format to provide a clean overview and being easily presentable, making it easier for the user to complete the survey. Questions regarding the artefact itself are presented in Table 4.

Table 3: Metadata questions for artefact survey

ID	Question	Answer types
1	What is your profession?	Developer, Architect, tester, Infrastructure engineer, Designer, Other
2	Your experience	1 year, 2-3 years, 4-5 years, 6-10 years, 10 years
3	Rate yours security knowledge from low to high	1-5
4	How often do you use threat modelling?	Never, Once - in design phase, usually the beginning of a project, Frequently - as part of the development cycle, Other
5	Which cloud provide are you using?	AWS, Azure, GCP

Table 4: Question about artefact for survey

ID	Question	Answer type
Q1	Have you previously used a similar artefact or tool for threat modelling?	Disagree, partially disagree, neutral, partially agree, agree
Q2	Do you find the questions helpful for threat modelling?	Disagree, partially disagree, neutral, partially agree, agree
Q3	Did it spark ideas that you can use for further brainstorming?	Disagree, partially disagree, neutral, partially agree, agree
Q4	Was it easy to understand the questions?	Disagree, partially disagree, neutral, partially agree, agree
Q5	Do you think it can help to improve threat modelling?	Disagree, partially disagree, neutral, partially agree, agree
Q6	Do you think the artefact is mature to be adapted to threat modelling?	Disagree, partially disagree, neutral, partially agree, agree
Q7	Would you have used it yourself in an actual threat modelling session?	Disagree, partially disagree, neutral, partially agree, agree
Q8	Do you have any other ideas to improve threat modelling in the cloud?	Free text

3.3.3 Case study

Since conducting the artefact evaluation in a real case scenario would be time-consuming and pose challenges related to information disclosure, a compromise can be achieved through a simulated case scenario. The case study will therefore consist of two stages: firstly, a stimulated case scenario, and secondly, the evaluation of artefact through a voluntarily and anonymous survey. A detailed description of the simulated application of the artefact can be found in section 4.3, followed by the evaluation process in section 5.

The simulation is to give context on how the procedure is conducted and how the artefact can be integrated to the phases of threat modelling. Then it is easier to have an in depth understanding on which steps in threat modelling activity the artefact can improve on. Accordingly, how teams can use the artefact to better their their work flow and the usability can be studied closer. The data generated for this evaluation is performed through the *Artefact survey*, outlined in section 3.3.2. Responses and analysis on the evaluation form after reviewing the artefact questions is given in section 5.

3.4 Data analysis

Data analysis in this thesis aims to derive information from the artefact survey, which the results are presented in section 5. Other data generated has been extracted from the CNAPP tool to analyse the vulnerabilities. There exist two main types of data; qualitative and quantitative data analysis which have been accumulated in this thesis. The following sub sections briefly discuss these distinctions.

3.4.1 Quantitative Data Analysis

The questionnaire in the artefact survey generate quantitative data that will be used for the data analysis. There exist four main types quantitative that can shortly be summarised:

- *Nominal data*: used to group categories on common characteristics. Only used to count the frequency.
- *Ordinal data*: is essentially categories that can be ordered, for instance as a scale from a range for "Agree" and "Disagree" and there exist nuances between them. However, the distance between the categories is not necessary equal.
- *Interval data*: is the same as ordinal data, but the distance is always the same.
- *Ratio data*: share the similarity with interval data, but there is a true zero which is used as a reference to compare against.

Generated data from this thesis contains nominal and ordinal data derived from both datasets by the CNAPP tool and the artefact survey. These types of data are only used for simple statistical purposes, e.g. looking at mean, median, mode and frequency.

3.4.2 Qualitative Data Analysis

Qualitative data is data that does not contain any numeric value, for example text, sound, photos, images and videos. Contrary to quantitative data that is more concerned around measuring, qualitative is more descriptive. The purpose is to derive patterns and characteristics for the analysis.

In this thesis, qualitative data is considered the open-ended questions from the survey that participants answered stimully. Otherwise, datasets scanned by the tool contained also qualitative data represented by the columns that did not have numeric values. Patterns and other findings were derived from these volumes of data to compare the different CNAPP tools and support the decision regarding which tool to use.

3.5 Ethics

The overaching principle for research ethics is that the ends do not justify the means in the pursuit of knowledge [10]. This emphasises that researchers bears the responsibility and held accountable to behave ethically and ensure fair treatment of all individuals involved in the study. In the case of the artefact survey, the risk of disclosing user information was carefully considered, leading to the decision to conduct the survey anonymously and not save any personal information, including email accounts. Participants were informed about the purpose of the thesis and voluntarily engaged in the survey, in compliance with Article 32 of the personal data Act [31].

Regarding intellectual property and disclosing confidential information, these were the main concerns during the collaboration with Visma. To protect Visma's rights, a non-disclosure agreement was signed, granting access to their tools, infrastructure repositories, scanning tools, and related data. Prior to the thesis, internal teams at Visma voluntarily agreed to participate by granting permissions to scan their infrastructure.

It is important to note that the raw and final results of the thesis have been obfuscated, ensuring that they cannot be traced back, as crucial information has been omitted. No passwords or secrets were stored in the raw data, and the names, team and project were anonymised. Furthermore, after the implementation was completed, access to the repositories was revoked, and scanning or access to the infrastructure was no longer possible.

4 Implementation

In this chapter, it will be provided a detailed explanation of the implementation of the artefact. *Design and develop* will describe the different design drafts and how the development progressed. Furthermore, *Demonstrate artefact* will present a simulated case scenarios to showcase how the artefact is applied in practice.

4.1 Design and development

The initial phase of creating the artefact involved explicating the core problem and defining the requirements. These steps were already elaborated in detail in section 3.2.1 and section 3.2.2. Moving forward, brainstorming sessions were initiated during weekly meetings to generate new ideas and potential solutions. Next section presents the ideas that were discussed during the meetings.

4.1.1 Brainstorming ideas

This section will explore the ideas that were initially considered during the early meetings but were ultimately set aside due to time constraints. These ideas were considered as extended features on the base artefact and the implementation was deemed as a lower priority. However, it is worth nothing that these features have potential to further improve the artefact and make it more effective as a preventive tool. An elaborated discussion of these ideas is given in section 6.2 about future work.

Radar chart
Other ideas that were discussed but were not selected, was implementing a dynamic and interactive radar chart with multiple variables representing the tactics. A distinction between the proposed radar chart and other type of graphs, are the interactivity and dynamicity. Interactivity allows user to click on the different tactics to obtain more detailed information about each finding, and based on this the questions can be extracted. A dynamic graph will provide visual representations of the different statistics defined in FR1, given in Table 1. An example is showing the frequency of the aggregated failed checks over a specific time period. Dynamic imply that it will automatically update the graphic with recent scans and results, ensuring that it will stay up to date with current cloud infrastructure state.

Another feature is to compare the findings within a project against different projects or the entire organisation's projects, providing insights into the security levels or identifying irregularities and interesting patterns that warrant further investigation. Furthermore, this could be extended to establish thresholds on normal and alarming counts of failed checks. These thresholds would be represented by different colours, such as red to indicate the need for immediate inspection of a specific cloud resource or project, while other colours would signify other actions. Although, the specifics of what actions that needs to be taken is not defined but the gist of it, is to prompt appropriate actions based on these indications.

Best Practice Compliance
It was discussed to test another compliance, using the tool's Best Practice compliance in additional to the Mitre framework. This would provide a more targeted and detailed

overview of the cloud architecture and its resources, allowing for a better understanding of which resources that are affected. The categories were organised around essential cloud resources, providing a clear visualisation of the architecture. Tailoring the controls around these resources would ensure better visibility and highlight potential vulnerabilities specific to them. It will inform the engineer when employing a certain service from a given category by being familiarised with the vulnerabilities associated with the categories.

Comparing pre and post deployment scans
Another suggestion that was briefly brought up was to compare pre-deployment code, specifically the Infrastructure as Code (IaC) scans, with the findings obtained after deployment. The aim was to detect any potential misconfigurations that could have been prevented in IaC.

The aim was to detect any potential misconfigurations that could have been prevented in the IaC phase. However, this idea faced challenges in terms of data format compatibility, making it more difficult to implement in practice. While potential workarounds could have been explored, other considerations took higher priority in the project. Therefore, further exploration of this idea was not pursued within the scope of this thesis.

However, this proved to be more challenging in terms of data format compatibility. There was also an uncertainty regarding the potential benefits versus the effort to carry on with this idea. As a result, it was decided not to pursue further, as perceived challenges and uncertainties outweighed the potential benefits within the scope of this thesis.

4.2 Technical tools

This section outlines the tools used to implement the artefact. These decisions were justified based on factors such as familiarity and flexibility to process the data. Specifically, the choice of data analysis tool and CNAPP tool were carefully considered. The goal was to select tools that would effectively support he development and analysis of the artefact.

4.2.1 Jupyter notebook

The data analysis tool chosen for this project was Jupyter Notebook[4] with Python[5] as the programming language. Python was selected as the preferred programming language due to the availability of numerous open-source libraries specifically designed for data science projects. Jupyter Notebook is convenient framework for organising and performing data manipulation tasks.

Using programming languages like Python offered greater flexibility in terms of data processing and manipulation compared to tools like Excel[6]. A downside, was the time-consuming nature of coding certain features compared to utilising built-in features in Excel. Despite that, the flexibility and extensibility outweighed the drawbacks, providing greater control and customisation.

[4]https://jupyter.org/
[5]https://www.python.org/
[6]https://www.microsoft.com/en-us/microsoft-365/excel

4.2.2 CNAPP tool

As already described in section 3.3.1, Tool 1 was selected as the CNAPP tool which would extract the datasets from. These datasets were further refined and processed in Jupyter and used to develop the artefact.

4.2.3 Processing results

This section provides an overview of the steps involved in processing the raw dataset from the CNAPP tool. Starting from selecting the MITRE ATT&CK framework as the compliance, scanning the infrastructure, and finally extracting the questions. Figure 8 presents a high-level overview of processing the security findings.

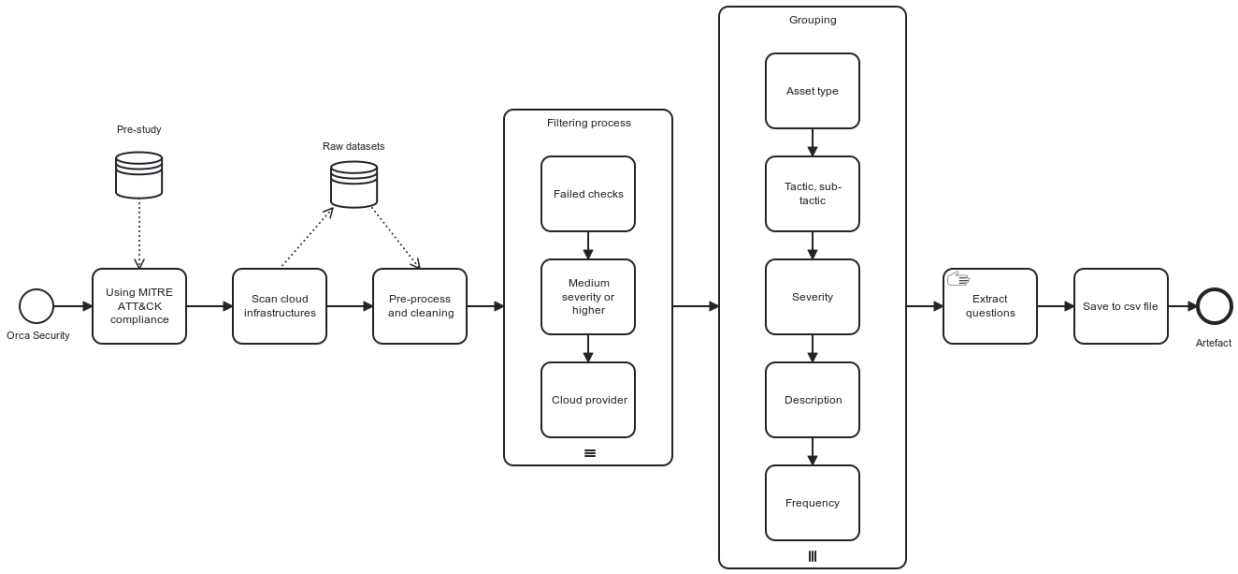


Figure 8: Overview of processing the results

The next step involved manually rephrasing the data into questions. Additionally, "Threat" and "Asset category" are added to the columns. This was done manually by deriving the values from the tactics, using the description of tactics from the official page[7]. The results are thereafter stored in a csv format as a final result.

The initial phase starts at the CNAPP tool by firstly choosing the Mitre ATT&CK framework as the compliance with the appropriate security controls. The MITRE ATT&CK framework was chosen based on the findings from the pre-study, referred in section 2.2.

Following the selection of the compliance, scanning was performed on totally 27 Visma infrastructure projects using the CNAPP tool. Permissions was obtained from the respective teams responsible for the project to conduct the scanning. The output of the scans resulted in a raw dataset containing both passed and failed checks, which could be further analysed. Subsequently, downloading the dataset was not fully automated, but there is potential for automating this step in the future.

Next step involved pre-processing the dataset, which entailed cleaning and refined it to ensure an organised and structured data format. Then comes the filtering the dataset.

[7] `https://attack.mitre.org/tactics/enterprise/`

First, the dataset was filtered based on failed checks and cloud provider, and then those with a medium severity or higher were given priority.

The data was thereafter grouped according to different properties, such as "asset type," "tactic," "sub-tactic," "severity," "description of why it failed" and the "frequency". For each asset type, the data was sorted based on frequency, providing a ranking for each asset.

As a result, the highest-ranked failed checks were chosen to be manually rephrased into questions. In this thesis, the top three failed checks were selected, although the number is arbitrary. Furthermore, two additional columns, "Threat" and "Asset category", were supplementary added to the set of questions. These new columns serve the purpose of addressing the threat posed by each question and providing a convenient grouping of the asset types. The values for these columns were derived manually by referring to the tactics and using the description of tactics from the official MITRE ATT&CK page[8].

Finally, the questions were stored in a CSV format, as the final output of the process. This lightweight workflow highlights the various and partially automated stages in processing the security findings.

While some tasks of the process are automated, there were certain stages that required manual intervention. These manual steps included tasks such as downloading the findings from the CNAPP tool and uploading it to Jupyter, as well as extracting the questions and deriving values for "Threat" and "Asset category" columns. However, it is important to note that some of these manual steps have potential to be automated in the future, although this aspect was not specifically addressed within the scope of this thesis.

The objective of this approach is to establish a reproducible guideline that can serve as a foundation for future research work. The aim of the artefact is to facilitate automation, allowing for the extraction of new questions. These can in turn be validated against existing questions. The cycle of extracting and validating new questions can be performed at regular intervals, such as every six months, although the specific timeframe can be determined by the team.

The intention is to routinely scan the infrastructure, adapting and addressing identified issues for new systems. This aligns with the "shift left" approach, where security considerations are integrated early in the development process.

4.3 Demonstrate artefact

This section will demonstrate how the artefact is integrated into cloud threat modelling, by first presenting a description of the artefact itself, followed by reasons why the artefact particularly is suitable for threat modelling, with a specific focus on addressing the cloud issues in the infrastructure.

4.3.1 Description of artefact

Summarised, the questions extracted from the dataset were derived from FR2 given in Table 1, and were categorised based on type of cloud asset, frequency and filtered on

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

severity of at least medium or higher. These questions were manually modified and formulated to better align a threat modelling session, adopted to a more generalised phrasing. This was necessary to adjust since the original text only described the issue and therefore needed to be reformulated to a question. A short version of the questions for AWS can be seen in Table 5, the detailed version is given in the Appendix A.

Although the questions were attempted to be adjusted to threat modelling questions, it could sometimes be difficult to understand the implication of the questions. The user might not understand why to ask them or is not familiar with the context. An additional "Threat" column was added to indicate what type of threat the questions were mapped to. This was an attempt to put it in a context for a better understanding of the question.

The goal of the questions is to motivate the team to engage in a discussion about detecting possible solutions to secure the infrastructure. It is encouraged to apply these questions in the beginning of the session to serve as an initial focal point for discussion. They can rapidly assist in narrowing down the crucial cloud resources that need to be protected.

In addition, the artefact is designed to be automated to a certain extent, allowing for periodic scanning and performing analysis. This enables the validation and evaluation of the questions, that can be used in an empirical assessment of the artefact's performance. By comparing the results of the artefact against actual findings, its effectiveness can be measured and improvements can be made.

4.3.2 Adaption to cloud threat modelling

As introduced in section 1, the objective is to integrate the artefact questions into threat modelling. This section will elaborate on how the artefact questions are adopted into a threat modelling, effectively addressing FR3 specified in Table 1.

Identified root causes from problem explication
"People" and "Process" were identified as root causes in the fisbone diagram presented in Figure 5. The artefact aims to address these root causes by integrating the artefact questions with the threat modelling process. As a method artefact, it provides guidelines and processes to achieve specific goals. In this case, the artefact introduces supplementary questions that are adopted in the threat modelling process, making it a "new" approach. The primary goal is to detect cloud vulnerabilities and address issues in the cloud environment. It follows a "shift left" approach, by emphasising the importance of conducting threat modelling activities early in the development phase. The questions incorporated in the artefact are rooted in previous security issues, allowing them to be addressed at an earlier stage. This proactive approach enables the team to identify and mitigate potential vulnerabilities effectively, leading to an improved security posture.

Catalyst for brainstorming regarding
One of the challenges with threat modelling, is knowing what to focus on, as there can be numerous potential risks with varying probabilities. It can be time-consuming and exhaustive to try cover all possible scenarios. Therefore, identifying key focal points can be time saving and prioritise the important assets to protect in the infrastructure. However, it is not recommended to solely relying on the provided questions. Instead, the artefact serves as a tool to initiate discussions and facilitate the exploration of potential vulnerabilities. By applying the questions, the team can engage in discussions and collectively spotting and resolving security considerations.

Alignment with cloud threat modelling guideline
The report on "Cloud Threat Modelling" published by CSA, differentiate between cloud and non-cloud threat modelling by accounting for the cloud infrastructure and its interconnectivity. This is reflected through asking questions related to the infrastructure. The report provides examples of questions as following [32]:

- Can I trust cloud services and infrastructure with company X, in a multi-tenant fashion?
- Is it safe to move key business and financial processes to SaaS from our premises?
- Can the cloud offer sufficient privacy and confidentiality controls for sensitive and regulated data?

Thus, the artefact questions aim to serve a similar purpose for the team during a threat modelling session, specifically tailored to address the cloud assets. These questions assist in understanding the threats, assets and security controls associated with cloud systems. In turn, this allows the team to make informed decision about their cloud infrastructure and services. By identifying and discussing potential attack surfaces, the team can proactively implement mitigation measures and security controls early on, in order to protect assets and data, aligning with the concept of "shift left" approach.

Security controls and cloud visibility
As mentioned in section 2.2, lack of cloud visibility was a concern due to its complexity. The security controls embedded in the MITRE compliance utilised by the CNAPP tool, are represented as a graph whereas the nodes correspond to the assets and the edges are inbound and outbound connections to other nodes. This provides a visualisation of the complex cloud environment and helps in overseeing underlying threats. Specifically the MITRE ATT&CK compliance will query suspicious link between the connections and identify failed control checks.

AWS		
Asset type	**Question**	**Counts**
VM	Does any EC2 instances face the internet with broad S3 access?	25
	Are there any EC2 instances that allow public ingress access on SSH port 22?	24
	Are there any EC2 instances with admin privileges?	19
AWSUser	Are there any inactive users?	87
	Are there any IAM users with admin privileges?	82
	Does the users have MFA enabled?	10
AwsSqsQueue	Is the SQS queue publicly accessible?	2
AwsSnsTopic	Is the SNS Topic publicly accessible?	2
AwsRoute53HostedZone	Does the Route53 Alias Record point to invalid resource?	15
AwsKmsKey	Does the master key have cross-account access?	43

	Does the key have public access?	6
	Is the CMK (customer master key) exposed?	6
AwsIamRole	Is there any unused role attached to a policy?	804
	Is there any cross-accounts access without external ID or MFA enabled?	85
	Is there any IAM role with admin privileges?	83
AwsIamManagedPolicy	How is the privileged policy manged? Any unecessary policy attachments?	34
	How is the privileged policy manged? Any policy versions that should be removed?	27
	How is the privileged policy manged?	23
AwsIamInstanceProfile	Any instance profiles with admin privileges?	10
	Any privileged instance profiles with assume roles?	1
	Any privileged instance profiles with pass roles?	1
AwsIamGroup	Any IAM group with admin privileges?	13
	Any privileged groups with unecessary policy attachments?	8
	Should any users from a privileged group be removed?	6
AwsEksCluster	Is the Kubernetes API server publicy accessible?	1
AwsEcsContainerInstance	Is there any AMI for ECS-related instances that are outdated?	72
AwsEc2Elbv2	Any Elastic Load Balancer that are public accessible?	51
AwsEc2Elb	Does the ELB have inbound rules in their security groups?	4
AwsDmsReplicationInstance	Is the Database Migration service publicly accessible?	1
AwsCertificate	Is the ACM certificate expired?	30
AwsAsg	Is the EC2 instance configured with public IP addresses?	8
AWS S3 Bucket	Does the S3 Bucket enforce HTTPS?	745
	Is the S3 buckets accessible to unmonitored accounts?	21
	Is the S3 buckets public accessible via bucket policies?	12
AWS Lambda Function	Does the environment variables expose secrets?	68
	How should it handle outdated Lambda function?	54
	Is the Lambda Function exposed publicly?	3

Table 5: Shortened version of extracted questions for AWS

4.4 Simulation case

In the following sub-sections, a simulation of applying the artefact in a threat modelling session is illustrated. Due to time and resource constraints, a simplified use case scenario has been developed specifically for this scope, instead of performing on a real system. Although it is not an actual application, it serves the intended purpose of demonstrating the artefact in a cloud threat modelling.

4.4.1 Case description

A description of a fictional web application is provided to give context for the relevant threat modelling activity.

Given a web application focused on fashion e-commerce, the team has been assigned the task of designing and developing this application for a medium-sized company. On average, the application is expected to cater to approximately 1,000 users daily. However, during campaigns and effective promotions, the user traffic can surge to nearly 20,000 daily users interacting with the web servers. Consequently, the team needs to consider the flexibility of cloud resources to efficiently handle peak traffic. Additionally, the company is in a growth phase, with aspirations of rapid expansion in the coming years, it is crucial to incorporate scalability in the design phase.

4.4.2 Core Threat Modelling activities

Cloud Security Alliance (CSA)[9] presents seven core threat modelling activities in their report about Cloud Threat Modelling [32]. This simulation will follow these referenced steps to demonstrate how the artefact can be effectively applied. While there exist different techniques, it is important to note that fundamental essence of the threat modelling process is the same. The specific order and details may vary, but the core principles and objectives remains untouched.

1. Identify threat modelling security objectives
2. Set the scope of the assessment
3. System/application decomposition
4. Identify and rate the potential threats
5. Identify weaknesses and gaps in the system and design components
6. Design and prioritise mitigations and controls
7. Communicate and create call to action

4.4.3 Identify security objective

The first and foremost step is to prioritise critical aspects such as confidentiality, integrity, availability, privacy, and more [32]. In the case of this web application, certain security objectives will be prioritised.

[9] `https://cloudsecurityalliance.org/`

Ensuring privacy is essential for the application, as it involves handling customer data and personal credentials. Protecting these assets are crucial to maintain customer trust and comply with privacy regulations.

Additionally, ensuring the integrity of the web application is vital. Preventing unauthorised access and malicious alterations to the website, protects the data integrity and maintain the trustworthiness of the e-commerce platform.

Another key security objective is ensuring availability. As an e-commerce web application, it is essential to handle both low and peak levels of traffic in order to provide seamless user experience. Any disruptions or downtime can result in lost customers and missed sales opportunities, in addition to unsatisfied customers. Therefore, implementing robust measures to ensure high availability is important for this business application.

4.4.4 Determine scope

This step focus on defining the scope of the application by selecting the level of detail for this use case. The thesis topic revolves around securing the cloud environment, and therefore, the scope is set to the cloud infrastructure. This entails protecting the cloud resources, and identifying the user groups that interact with these assets. While it is important to discuss the specific technology aspect in terms of application security, it falls outside the scope for this case scenario. For simplicity, AWS is chosen as the cloud provider since it had the most listed questions and therefore cover a wider range of assets.

During this step, infrastructure engineers can utilise the artefact to identify and select cloud components for the web application. They can accomplish this by reviewing the provided list in Table 5, sorted on "Asset type" or "Asset category". By doing so, they can identify relevant assets that need to be deployed alongside the web application to ensure its functionality and security.

Another aspect to consider is the management of "user groups" that have permission to access the cloud resources of the web application. This is to restrict unauthorised access to the cloud environment, aligning with the principle of least privilege. It is important to determine how different privileges should be handled within the infrastructure. In this regard, sorting the list depicted in Table 5 based on "Identity and Management resources" can be helpful finding the relevant assets to include and determining which questions that can provide useful insights. Assets such as "AwsIamRole," "AwsIAmManagedPolicy," "AwsIamGroup," and "AwsIamInstanceProfile" should be addressed within the defined scope. Applying the artefact questions can assist in considering how authentication and authorization should be handled, as well as determining the appropriate permissions to be granted to different types of users.

To address the privacy objective mentioned in section 4.4.3, it is necessary to establish an infrastructure that focuses on encryption, data protection and network security. If the team is uncertain about how to approach these aspects, utilising the "Asset category" column can be beneficial, by sorting the list based on "Network Resources", the team can easily identify and select relevant resources. This approach proves particularly helpful when the team has limited knowledge in certain fields, and serves as a catalyst for discussions and further exploration.

Therefore, assets such as "AwsCertificate" and "AwsKmsKey" can be considered as part of the cloud environment to address privacy concerns and strengthen data protection

and network security. The cloud resources used in the infrastructure for this case are as follows:

- DNS service
- API gateway
- Load balancer
- Event Queue
- Serverless Function
- Cache
- Database or data storage component
- Network resources
- Identify and Management resources

4.4.5 Application decomposition

This step involves breaking down the system into smaller components and establishing connections between them. Essentially, it is about identifying the trust boundaries, inputs and outputs, and mapping the data flow.

Now that the scope has been defined as the cloud infrastructure. The relevant components have been picked out, the next task is to connect these and map out the flow. From the perspective of an end-user, this can be visualised in Figure 9, which illustrates the inter-connection of the components within the infrastructure.

By mapping out the flow and connections between the components, the team gains a clearer understanding of how data and information move within the system. This step is essential to identifying any potential vulnerabilities or security gaps in the data flow.

From Figure 9, the user is interacting with the web page through the internet. The API gateway service is responsible for authenticating and validating the user. Furthermore, Route 53, a domain name system, is used to perform an IP address lookup and redirect the user to the correct destination. When a request is made, it passes through the API gateway, which acts as an entry point for the web application. From there, the request is sent to a load balancer, which distributes network traffic evenly across different servers hosting the web application itself. This mechanism helps to ensure the availability objective. Depending on the user's action, known as an event, it may be placed in a queue, which triggers a lambda function to perform a specific task. This task could involve updating or writing data to a database, or reading data from the database. In the case of reading data, there may be a temporary data retrieval from a cache for quicker access.

Additionally, an auto scaling group has been added to cover the entire web application infrastructure. This is done to accommodate the company's anticipated rapid growth. It is important to ask security-related questions about this cloud resource as a failed auto scaling setup can become a vulnerable single point of failure.

By considering the security implications of each component and resource within the cloud infrastructure, the team can ensure a robust and resilient system that can handle

user interactions, maintain data integrity, and support the scalability needs of the web application.

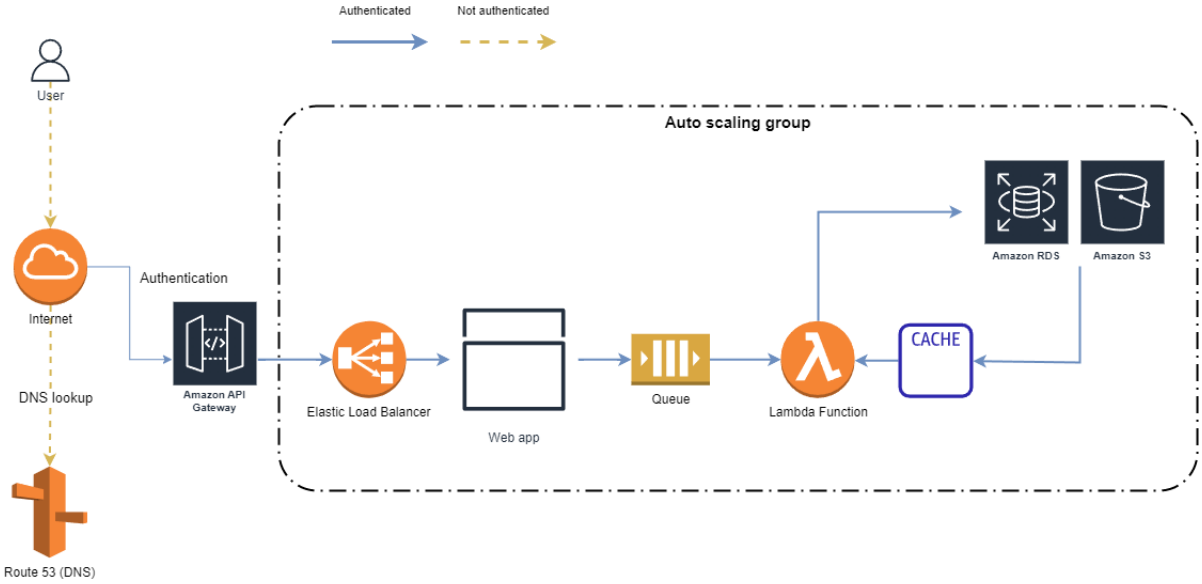


Figure 9: A diagram displaying the cloud infrastructure for the web application

4.4.6 Identify and enumerate potential threats

This step involve identifying threats, type of attacks and potential threats [32]. While frameworks like DREAD[10] is recommended, the artefact questions can be applied to enumerate potential attack surfaces at the infrastructure level in the cloud. In addition to the artefact, other frameworks can also be utilised to achieve the same goal. It is important to highlight that the artefact questions are not meant to be used exclusively for the threats enumeration. It is recommended to consider other frameworks and methodologies in addition to the artefact. These different strategies contribute to a more comprehensive assessment and enhance the overall threat modelling process.

Here are some example questions that can be used related to the cloud resources depicted in Figure 9, using Table 5. The questions have been adapted and modified to suit the specific context, by framing the questions such that it fits into a threat modelling session.

- Identity & Management Resources:
  - How should the admin privileges be handled? How restrictive should it be?
  - Do we have in place MFA enabled for accessing our infrastructure?
  - How should inactive users be handled? How long should they be granted access, initially to avoid inactive users?
  - How is the privileged policy managed? Who should have access?
- AwsRoute53HostedZone: Does the Route53 Alias record point to an invalid resources? Does the resource exist in the cloud environment?
- AwsEc2Elb: Should the load balancer be public accessible? Who should have access to this? How should the traffic to the load balancer handled?

[10] https://en.wikipedia.org/wiki/DREAD_(risk_assessment_model)

- AwsSqsQueue: Should the queue be public accessible? Who should have access to this?
- AWSS3Bucker: Is the traffic to and from storage encrypted? Is the storage publicly accessible? Does it has to be public?
- AwsCertificate: Do you use ACM in order to manage your certificates? If not, how do you prevent your certificates to get expired?
- AwsKmsKey: Does the master key have cross-account access? Does it have public access? How to prevent the CMK (customer master key) to be exposed?

The remaining steps of the core activities are beyond the scope of this simulation and have been omitted. The primary objective was to to demonstrate the application of the artefact in a threat modelling session, detailed on an infrastructure level.

5 Evaluation

The last and fifth step of the design science framework is to evaluate the artefact. Firstly, presenting the approach for data generation, followed by a quick introduction to the evaluation strategy and goals. Finally the evaluation results are presented.

5.1 Data generation

The data for the evaluation has been collected through a voluntary survey conducted using a Google questionnaire form. The form did not contain confidential data but only general questions. Participants were recruited both internally and externally at Visma, resulting in a total of 24 participants.

5.1.1 User tests

The questions were evaluated by participants who voluntarily and anonymously reviewed them. Each participant could freely choose the questions based on their preference for a specific cloud provider. While it would have been ideal to provide a case description similar to the one in section 4.4.1 to offer an example of how the artefact could be used. As a possible result, that could influence the responses toward on how well it addresses the use case. While the intention of the survey is to evaluate the questions' effectiveness for threat modelling in a cloud context. Therefore, there was a risk that the responses could become subjective and biased, making it difficult to draw objective conclusions about the artefact's overall performance.

5.2 Evaluation description

The fifth step on the design science framework is to select the evaluation goals and strategy before proceeding with evaluating the artefact itself. In this regard, each question from the artefact survey is directly linked to the evaluation goals, serving as a metric for assessment. The linkage between the evaluation goals and the corresponding survey questions is presented in Table 6 .

Table 6: Evaluation goals linked to survey questions

Evaluation goal	**Data generation**
Effectively solving the problem	Q3
Usability	Q2, Q4
Evaluate requirements	Q2, Q4, Q5, Q6
Compare similar artefacts	Q1
Investigate side-effects	Q7
Formative evaluation	Q8

5.2.1 To what extent does it effective solve the problem?

Summarised, the artefact consists of a set of questions extracted from the the CNAPP tool, which are based on previous issues found in scanned cloud infrastructures. The primary objective of the artefact is to assist the threat modelling session by identifying potential issues and preventing them from occurring, thereby securing the infrastructure during the design phase. By leveraging post-deployment findings, the artefact serves as a foundation for shaping proactive questions that can be asked to prevent similar issues before the deployment.

The problem the artefact is trying to solve, is to secure the infrastructure during the design phase. This is challenging to evaluate, as there does not exist any quantitative measures that quantify this goal. However, to gauge the extent to which the artefact assisted threat modelling, the question "Did it spark ideas that you can use for further brainstorming?" was used to evaluate the perceived usefulness in addressing this goal. This question aims to capture the essence of the evaluation goal by determining if the artefact generated ideas that could contribute to identifying potential vulnerabilities and initiate valuable discussions among participants that can lead to secure the cloud.

5.2.2 Usability

Evaluating the usability of the artefact is linked to the questions "Was it easy to understand the questions?" and "Do you find the questions helpful for threat modelling?", as it addresses some of the key points listed in section 3.3.2. These questions aim to capture the user's perception of usability and whether it succeeds to help accomplish their tasks. While the responses are subjective, the questions are intended to provide an overview of the ease of use and the clearness in the question formulations. Furthermore, the open-ended questions also contribute to additional feedback on the usability of the artefact.

5.2.3 Evaluate requirements

The artefact should be evaluated against the requirements depicted in Table 1 and Table 2. However, some of the requirements are not applicable to be directly evaluated by the users. The related requirements that are tested in this survey are:

- Functional: The questions should assist in threat modelling sessions to address cloud vulnerabilities
- Non-functional: The artefact questions should be easily understandable.

It is important to note that evaluation of the non-functional requirement is already being covered through the usability evaluation. This is due to the overlap between the evaluation metrics. However, the evaluation results of this goal will be discussed in light of both requirements and referring to the usability evaluation results. Thus, questions addressing the requirements are identified as:

- Was it easy to understand the questions?
- Do you find the questions helpful for threat modelling?

- Do you think it can help to improve threat modelling?
- Do you think the artefact is mature to be adapted to threat modelling?

5.2.4 Compare to similar artefacts

The evaluation consider similar artefacts or approaches used for threat modelling in the same way. This aspect is addressed through the question "Have you previously used a similar artefact or tool for threat modelling?". This question aims to gather information about the user's experience with similar tools.

5.2.5 Investigate side-effects

Examining the potential side effects of the artefact, the evaluation goal aims to identify unintended or undesirable consequences that may arise from using it. This aligns with the concept of investigating side effects in the design science framework [10]. To assess this, the question "Would you have used it yourself in an actual threat modelling session?" has been included. This questions helps to quantify how many perceive the current feasibility of the tool among participants. Furthermore, the open-ended questions allow participants to raise their opinions and provide, which can be analysed in context of this evaluation goal.

5.2.6 Formative evaluation

In formative evaluation, the purpose is to assess and gather feedback in order to improve the enhance and refine the artefact. This is for the developers to identify potential areas. In this case, the evaluation goal is addressed through the open-ended questions that suggest any potential areas for improvements. Additionally, formative evaluation has also been gathered through regularly and internal meetings with continuously feedback.

In formative evaluation, the purpose is to assess and gather feedback in order to enhance and refine the artefact [10]. In the context of this evaluation, open-ended questions were used to elicit feedback that could suggest ares for improvements. Additionally, formative evaluation was also gathered through regular internal meetings, where ongoing feedback and discussions took place to iteratively improve the artefact.

5.3 Evaluation results

This section discuss briefly the results from the evaluation survey associated with their goals. The findings of the evaluation and artefact are elaborated in more detail in section 6.

5.3.1 Participants demographics

A short outline of the demographics will be presented revolving the participants' profession, experience, security knowledge, familiarity with threat modelling and cloud

provider preferences to give a high-level understanding of their backgrounds and perspectives.

Profession
As observed, architects and infrastructure engineers dominate the survey, which aligns well with the emphasis is on securing the infrastructure. Targeting these professions is crucial as they play a key role in decision-making, particularly during the design phase. It is essential to gather their opinions and insights to ensure the artefact addresses their needs effectively. Additionally, including responses from professionals with strong emphasis on security is valuable, as their perspectives provide important security insights.

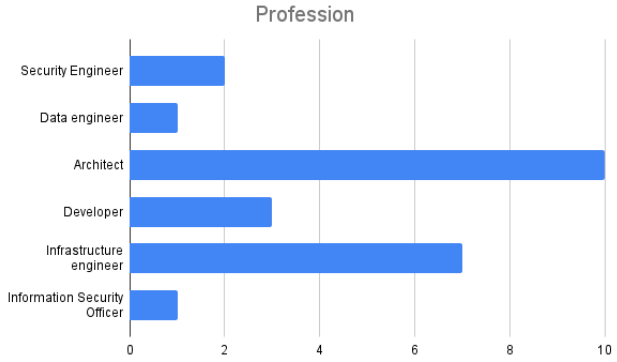


Figure 10: Profession demographics

Experience
Figure 11 presents the distribution of the participant's experience. The architects stick out as the most experienced group with 6 architects with over 10 years experience, while developers and infrastructure engineers are followed after.

It is evident that the participants in this survey are highly experienced and competent, with over one third of them having more than 10 years experience, The next largest groups are those with 6-10 years and 4-5 years experience, which are equal in count. This experienced pool of participants brings valuable responses to the evaluation process.

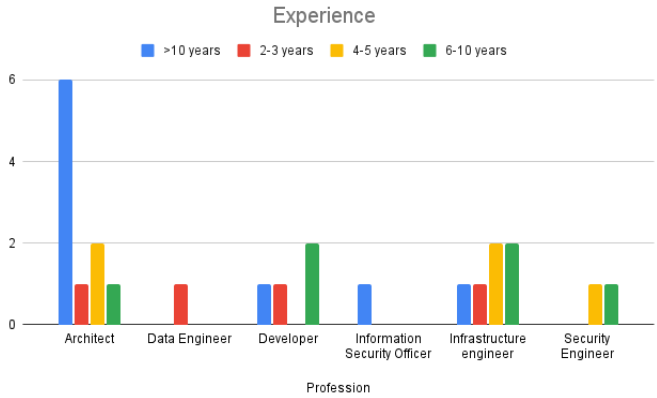


Figure 11: Experience demographics

Security knowledge
Figure 12 depicts the participant's security knowledge among those that volunteered. It can be observed that participants possess a high level of security knowledge, with the

majority falling into the range 3-4. Architects appears as the group that rate their security highest with majority of the participants rating it at 4.

Overall, the security knowledge seems to be solid. This indicates that the participants have a good understanding of security concepts and are well suited to evaluate the artefact accordingly.

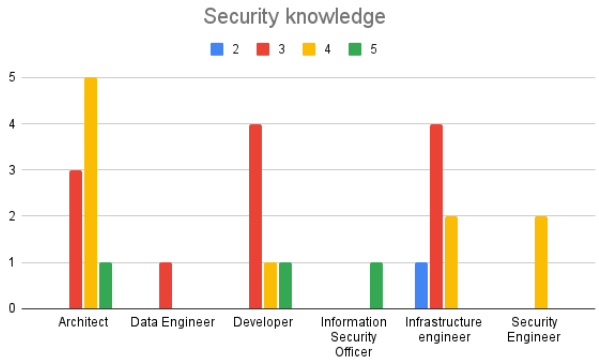


Figure 12: Security knowledge demographics

Familiarity with threat modelling
Figure 13 provides an overview of how often the participants are engaged in threat modelling practices themselves. Architects is the profession that had the most experience with threat modelling as they often perform frequently. Surprisingly, over half of the participants indicated that they either perform it only once or never at all. The distribution is somewhat divided, but the significant percentage with limited or no threat modelling experience is notable.

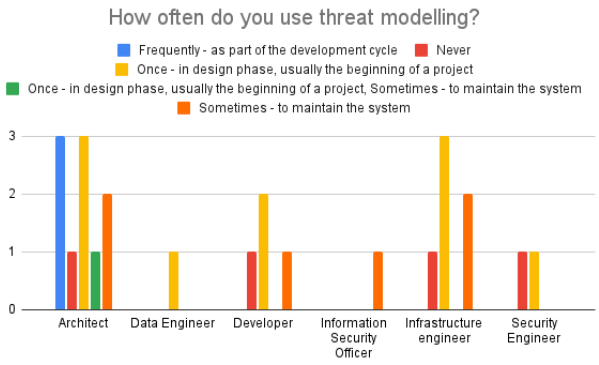


Figure 13: Familiarity with Threat modelling demographics

Cloud provider preference
Figure 14 shows the most popular cloud providers among the participants. It can be observed that Azure is the most popular choice, followed by AWS and GCP. Notably, one participant mentioned "Governmental data center" as an alternative option. It is worth noting that the majority of the questions in the artefact were derived from AWS, while the questions regarding Azure and GCP were more limited.

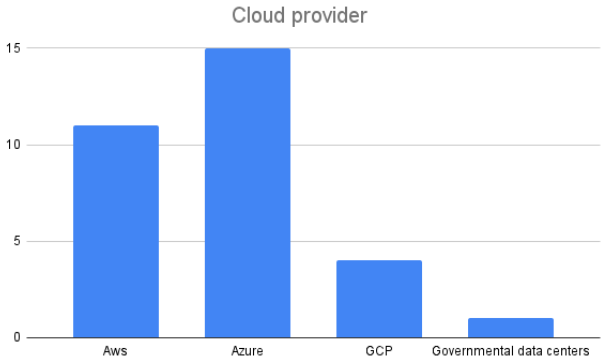


Figure 14: Cloud provider demographics

5.3.2 To what extent does it effective solve the problem?

The question "Did it spark ideas that you can use for further brainstorming?" was asked to assess to which extent it solves the problem. A majority of partially agree and agreee indicated that they gained some ideas they believed could be further used to brainstorm during threat modelling, but a small minority either disagreed or felt neutral about this. Figure 15 highlights the results of this question. It is notice that the architects solely expressed that they benefited from the questions the most. From the demographics, it was observed that the architects were the most experienced, rated their security knowledge highly and performed threat modelling often. This could be one of many factors on that explains why they gained ideas from the artefact.

While the goal is to address cloud vulnerabilities, it is a positive outcome that the participants is able to derive ideas from the questions. Potentially, these ideas can be further explored that lead to interesting discussions and develop solutions to mitigate such issues. The finding that a majority gained some ideas from the questions is promising and suggest that the artefact does partially solve the problem to a certain degree.

However, there was a comment expressing concerns about the questions making them feel constrained and limited in their thinking. They perceived the questions as checkboxes that have to be ticked off, rather than a tool to spark open-ended discussions and brainstorm ideas. This gap will be discussed further in section 6.2, where potential improvements and areas will be addressed.

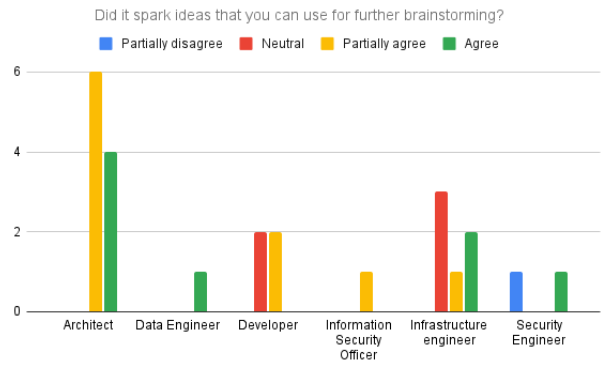


Figure 15: Distribution of Q3 in artefact survey

5.3.3 Usability

Figure 16 shows that the majority of architects and infrastructure engineers found the questions easy to understand the most. Small groups like information security and security engineer also agreed on this. This could be linked to previous demographics results such as experienced and security knowledge. For infrastructure engineers, the questions revolve around cloud services which are within their expertise, and thus may explains why they are able to understand the questions quicker.

Almost two-thirds of the participants agreed or partially agreed that they found the questions to be easy to understand. However, despite the simplicity of the questions, some participants noted that the did not immediately resonate with the wording, possibly due of the overgeneralised cloud specific terms. In some cases, the questions were too general and didn't fit to every situation. For example, the use of "primitive role" in the artefact did not correspond with the terminology used in GCP, where "basic IAM role" or "Authentication key management for service accounts" are more commonly used. Other participants also pointed out flaws in the phrasings and provided suggestions for improvement. These observations and feedback will be discussed in more detail in section 6, focusing on the maturity and potential of the question.

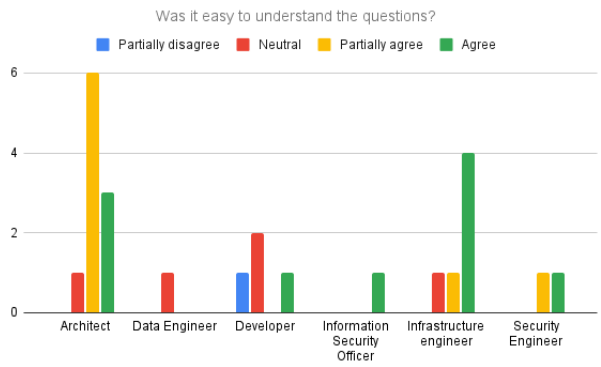


Figure 16: Distribution of Q4 in artefact survey

Figure 17 depicts that a large majority is in favour of agreeing that the questions are indeed helpful for threat modelling, with only one participant expressing a neutral opinion. This align well with the intention of assisting the users to accomplish their tasks. Despite that the questions can lack clarity and context, the overall responses indicate a positive trend in the right direction. To further evaluate usability, a more in-depth analysis and quantification of user feedback could be conducted.

5.3.4 Evaluate requirements

The previous section already covered the assessment of the non-functional requirement "The artefact questions should be easily understandable". The findings will be correlated with the evaluation of FR3, which aims to "Assist threat modelling to address vulnerabilities". Although the wording of the functional requirement may sound identical to the evaluation goal, there is a distinction on the implications between them. The requirement is inherently stricter and focuses on the long-term integration of the artefact into threat modelling practices, while the evaluation goal is more concerned with the short-term affect of the current state. It is important to recognise this difference, since the phras-

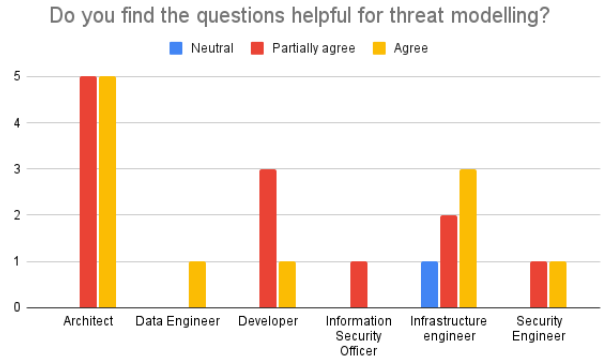


Figure 17: Distribution of Q2 in artefact survey

ing is similar. In this case, question 5 and 6 (Q5 and Q6) clarifies this by addressing the adaptability to threat modelling.

Figure 19 shows a more diverse distribution of responses regarding the maturity of adopting the artefact for threat modelling. While many expressed "agree" or "partially agree", a significant number remained undecided. In light of the results from Figure 18, the participants indicated that the artefact helped to improve threat modelling, there is still room for improvement in terms of maturity. The findings suggest that the artefact has the potential to be a valuable tool for threat modelling.

The same trend can be observed in terms of the usability of the artefact's questions. Participants generally found the questions understandable and helpful, but there is considerable potential for further improvement to bridge the maturity gap. Suggestions and feedback have been given that will be presented in section 5.3.7.

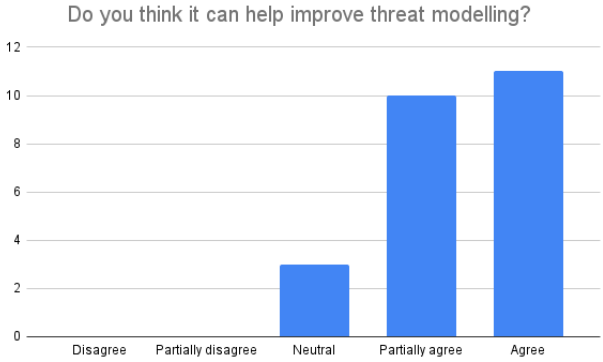


Figure 18: Distribution of Q5 in artefact survey

5.3.5 Compare to similar artefacts

Figure 20 depicts a varied level of familiarity with similar tools. There is approximately an even count between participants having and not having prior experience, and participants that feel neutral. This might also suggest the artefact may be relatively new to the users. It is important to emphasise the guidelines for how to apply the artefact, and give it more context to use it. As highlighted by participants who mentioned that they have previously used similar and various tools that was too immature to use.

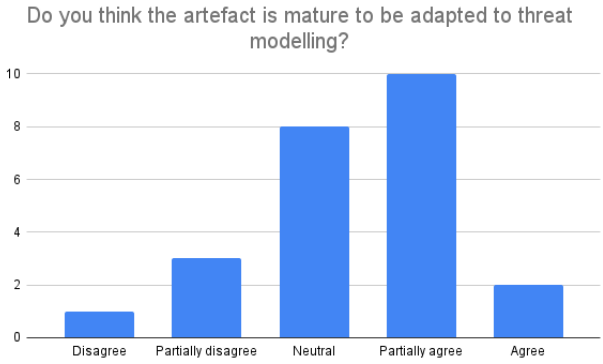


Figure 19: Distribution of Q6 in artefact survey

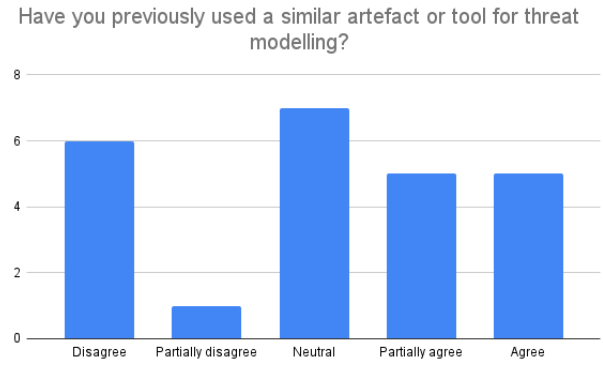


Figure 20: Distribution of Q1 in artefact survey

5.3.6 Investigate side-effects

Figure 21 outlines participant's willingness to use the artefact in actual threat modelling sessions. Approximately half of them expressed their interest in giving it a try, while eight remained neutral on the matter. While the majority are open to using the artefact, the findings also imply that it is not necessary straightforward to implement it immediately and there are still room for improvements. As mentioned earlier, there are concerns that the artefact questions may restrict users to a fixed structure, resembling a checklist, rather than encouraging a freeform exercise.

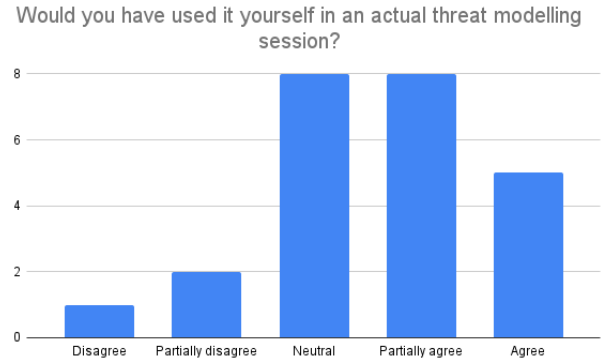


Figure 21: Distribution of Q7 in artefact survey

5.3.7 Formative evaluation

The formative evaluation is about gathering information that can help to improve the artefact. In this case, this was given by the participants through suggestions and recommendations in the survey.

Rephrase questions
The questions were sometimes a bit unclear and did not make sense. It was suggested to use more cloud-specific terms, taking into account that each cloud provider has its own cloud concepts and terminology. By incorporating provider-specific language, can increase the understanding quicker and reduce the cognitive load.

Additional data sources
It was expressed that there were limited questions for Azure, but the same can be applied for GCP. Microsoft Defender for Cloud[11] was proposed as a recognised CNAPP tool that could be leveraged, by extracting the set of questions in similar manner as was done with a CNAPP tool.

Other techniques
Other ideas suggested to focus on incorporating other techniques such as STRIDE into the artefact. This could involve mapping the artefact questions to each of the STRIDE objective, enabling users to identify the security objectives and implications more easily, and consequently the usability as well. Another suggestion was to explore the AWS Security Pillar Framework[12]. Users can benefit from mapping the questions with established best practices and principles.

Lack of other aspects
A lacking aspect of the artefact was the inclusion of certain topics, such as data encryption, application-level security, application integration, and secure data transfer from external sources. To further enhance the artefact, it is important to broaden its scope to cover a wider and comprehensive range of security considerations. It is worth noting that the selection of questions focused on the top three most frequent issues, which means some questions related to encryption may not have been included. Furthermore, the current state of the artefact is in a prototype development phase. As such, it is expected that it needs to undergo several iterations and refinements. The identified shortcomings can addressed in the subsequent iterations to minimise these gaps.

[11] `https://learn.microsoft.com/en-us/azure/defender-for-cloud/defender-for-cloud-introduction`
[12] `https://wa.aws.amazon.com/wellarchitected/2020-07-02T19-33-23/wat.pillar.security.en.html`

6 Discussion

This section will discuss and answer the research questions introduced in section 1. Furthermore, it explores the implications for the practice and future research, and additionally the threats to validity of this thesis.

6.1 Research questions

6.1.1 RQ1: How can the security findings from deployed cloud infrastructures improve the threat modelling process

This research question can be answered by the artefact that has been presented in this thesis. The artefact consists of a set of questions that can be used during threat modelling to tackle cloud vulnerabilities. It leverages a CNAPP tool that automatically scans cloud infrastructures using the MITRE ATT&CK compliance framework to categorise issues that have occurred post deployment. These undetected issues serve as a basis for deriving the set of questions that form the questionnaire. The questions are manually extracted, requiring domain knowledge to ensure they accurately capture the essence of the issues and easily understandable by others.

When integrated into threat modelling as demonstrated in section 4.4, these questions become a catalyst for initiating discussions related to potential threats and attack surfaces. Thus, it spark ideas and insights that can lead to taking appropriate security measures and countermeasures. Overall, the artefact can be a valuable tool for identifying the addressing compliance issues within cloud environments. By incorporating it into the threat modelling process and leveraging the post-deployment findings, it can address these concerns earlier in the development cycle. This approach aligns well with the shift-left approach that implement security early on and improve the cloud security posture.

6.1.2 RQ2: What is the effect of the proposed solution?

Since the artefact is a method artefact, which is described as defining guidelines and processes to achieve goals, it may be challenging to assess its properties at a detailed implementation level. The evaluation survey was designed with the intention of providing a high-level overview of the artefact's effectiveness, usability, requirements and side-effects. It was kept in a simplistic format, making it easier to complete to encourage as many as possible to participate. As a consequence, the obtained results were only at a high-level and lack specific details and nuances. It should be emphasised that the survey's high level findings allow for a general assessment of the artefact's potential and areas for improvements. These findings can then used for further iterations and refinements of the method artefact. The results will be discussed in more detail in the following paragraphs.

The first evaluation goal focused on assessing the effectiveness of the artefact in solving the explicated problem. The majority of participants either agreed or partially agreed that they found benefit in the ideas generated by reviewing the artefact questions, which can be used for further brainstorming. However, the granularity of this result is limited as it does not provide information on how useful these ideas are in practice. There was no specific measure of the usefulness of the ideas, as the primary focus was on determining

whether the artefact was helpful or not, rather than quantifying the degree of helpfulness. Nevertheless, the overall findings suggest a promising outlook for the artefact. To gain a deeper understanding of its problem-solving capability and to evaluate the attribute more comprehensively, future research could explore studying the practical application and impact of these questions in more depth.

Additionally, it is worth mentioning that participants did not apply the questions to a specific case scenario. Furthermore, it can be observed from the survey that some practitioners are new to threat modelling which they found the questions interesting and useful for securing their infrastructure. Others are not using threat modelling often, which can also be something that can be addressed. Incorporating a real threat modelling session with e.g. fictional system design in the next evaluation could provide a more contextualised assessment and a granular measurement. It was notable that the architects was the most experiences in terms of threat modelling and general work experience, and did well on understanding the questions and found it helpful. They were in overall the group that received the most positive feedback from. It is also an indicator that having experience, security and threat modelling knowledge were essential to benefit from the artefact questions. Therefore, future research on this can continue on how to make it more accessible for less experienced users.

Approximately two-thirds of the participants found the questions to be understandable, indicating a reasonable level of clarity. Additionally, the majority of participants agreed that the questions were helpful in terms of usability. However, it was significant that the majority of the experienced participants found the questions easy to understand, beside the infrastructure engineers (both experienced and inexperienced). This could be that the questions are related to their field of expertise regarding cloud services, making it easier for them to understand. Some participants also pointed out that certain implications were lacking in the questions, suggesting to incorporate more cloud-specific terminology to improve clarity due to the variation in terminology, configurations and services among different cloud providers. Ideally, the questions in the artefact should have been extracted and evaluated by a domain expert for each cloud provider to ensure the relevance and accuracy. However, due to time constraints, this aspect was only done for Azure and not AWS and GCP. In addition, the questions did not cover all cases because the scanned projects did not use the entire range of cloud services. Secondly, the scope was limited to only the top three most recurring issues, thus neglecting less frequent issues related to other cloud services. For future iterations, it is recommended to include domain expert for each cloud platform during the extraction process to improve the overall quality of the questions. Moreover, it can be considered to scan other infrastructure with other types of assets to include extend the range of questions.

It was a mixed response regarding the adaptability of the artefact, indicating that it needs further development before it can be fully integrated into threat modelling practices. This highlights the importance of improving the questions and conducting additional tests with a larger target audience to gather more feedback for further refinements. Some suggestions from the survey include integration the artefact with additional sources such as Microsoft Defender for Cloud to assess different types of assets and threats. This expansion could allow for a broader coverage of different asset categories. While the tools is specialised specific in Azure Cloud technology, there is potential to generalise the knowledge and apply it to other cloud providers as well.

Another suggestion is to incorporate techniques that make the artefact more familiar for users. For example, mapping the questions to well-known frameworks like STRIDE can

provide users with a familiar structure during threat modelling session. A combination of STRIDE and Mitre framework can be further explored.

When comparing whether the participants had previously used similar artefacts or tools, the responses were divided, indicating that the artefact may appear novel among the participants. This suggests the need to explore innovative solutions that can further improve upon the artefact. Automation can be a step into that direction, helping to simplifying complex layers and making it more convenient to use. However, automating the extraction process may currently be less prioritised due to the required maturity in this field. The process of forming the questions relies on expert knowledge and has to be performed manually. Nevertheless, there is potential for automation to become more feasible in the future.

Approximately half of the participants expressed interest in using the artefact in an actual threat modelling session, while the other half stayed neutral or disagreed. Although the survey did not capture the specific reasons, it is important to address the issue and provide support to users who are hesitant about adopting the artefact. One suggestion to address this is to provide better documentation or use case scenarios similar to the simulation case presented in section 4.4. This documentation would serve as a guide to help users learn and understand the new approach. The aim of the documentation would be to present users with a range of use cases, then the users can adhere and adopt to their own needs. It can help them overcome hesitation and provide a practical starting point for utilising the artefact, but also necessary guidance to incorporate it into their practices.

6.2 Implication for research

As identified in the pre-study referred in section 2.2, no comparable approach or technique incorporated a CNAPP tool in the context of cloud threat modelling was found. The evaluation results from participants indicate that the tool holds promise for further development, but there is a notable gap in research within the field of cloud security. Participants expressed appreciation for the level of detail in the questions provided. Future work is encouraged to delve deeper into the adoption and integration of techniques specific to the cloud domain. The research could greatly improve the artefact and inspire new development of other similar techniques and approaches.

Another implication is the need for more granular evaluation that can assess the specifics details and nuances of the responses. This would provide a deeper understanding of the implications of the questions and allow for a more comprehensive analysis. In addition, it would be valuable to apply the artefact in a real or fictional case scenario for the participants to evaluate its effectiveness and feasibility to a larger extent.

Future research can focus on developing the interactive and dynamic radar chart described in section 4.1.1 as an extension of the current artefact. This idea was discussed during regular meetings with both supervisor and co-supervisor. The implementation of an interactive radar chart could improve the usability of the tool, making it more engaging provided with more details and visuals. Figure 22 shows the radar chart with the Mitre tactics. When clicking on the e.g. "Impact" tactic, another radar chart appears as depicted in Figure 23. The issues are grouped by different categories with different "weights", signifying the importance of the category to provide easier navigation and decision-making. This extension has the potential to improve the overall functionality and user experience of the artefact.

Figure 22: Radar chart showing grouped issues according to categories

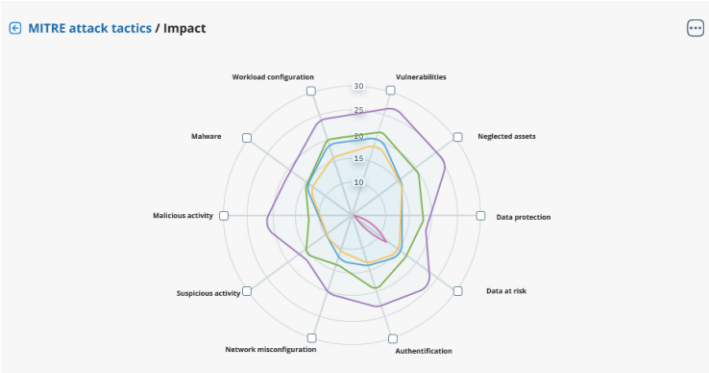


Figure 23: Radar chart showing related issues within the category Impact

6.3 Implications for practice

The artefact can be improved by periodically extracting questions based on the discovered security findings from the tool. By regularly monitoring the system, a broader range of attack surfaces can be included in the artefact. This should be a coordinated work between experts in cloud and security for each cloud platform. This is to understand how the questions can be formulated clearly and applicable for its context. In additionally, reviewing and evaluating the questions are equally important for the maturity of the artefact based on continuous feedback from the users.

Secondly, practitioners are encouraged to explore and adapt the artefact to their own techniques and methodologies. This empowers practitioners to find new ways and solutions that align with their specific needs, ultimately improving the effectiveness of threat modelling process.

6.4 Limitations and threats to validity

This section will address the limitations and threats to validity this study has been a subject to. Therefore, it is important to consider the findings within the context of these limitations.

The main limitation of this study arises from the constraints of time and resources. The duration of this thesis was limited to a single semester, which significantly restricted the time available for implementing and evaluating the artefact. Furthermore, due to the

relatively short time span, it was challenging to acquire a deep understanding of specific subject matter e.g. when phrasing the questions. Although the pre-study provided a foundation, gaining expertise in the field requires more time and experience. As a result, the questions presented in the artefact may not fully encompass all security perspectives relevant to the cloud domain. Ideally, expert inputs could been included to refine the questions. However this would been too time-consuming and was not feasible since the scan resulted in a large amount of data that could not have been extracted and evaluated with the given time frame.

Moreover, the evaluation process could have been more extensive and detailed. However, due to the limited time remaining, it was not possible to create a comprehensive survey to thoroughly analyse the evaluation results. These limitations should be taken into consideration when interpreting the findings and understanding the scope of this study.

One of the threat to validity in this study is the findings identified by CNAPP tool, that may be false positive. Not all alerts triggered by the tool does necessary indicate true positive findings, as some of them may not represent actual sensitive issues. This is common complaint related to Cloud Security Posture Management (CSPM) tools, which is a part of CNAPP [33]. Manually examining all findings to determine true positives would be an impractical task. However, further investigation of this issue would be an interesting idea, although it was not feasible within the constrains of time and available people to discuss with the practitioners on this topic. Therefore,

Although the questions potentially are influenced by false positive findings, their purpose in the artefact is not prediction or statistical inference. Instead, they serve as a means to generate ideas and facilitate discussions. Thus, the presence of false positives does not directly impact the artefact's objective. Nonetheless, it is essential to recognise that false positives and negatives are limitations in this study. However, but the findings were too large in quantity and infeasible to address in this study.

7 Conclusion

In this thesis, an artefact has been introduced to fill the gap in research regarding cloud threat modelling techniques. The aim was to address the following research questions:

- RQ1: How can the security findings from deployed cloud infrastructures improve the threat modelling process?
- RQ2: What is the effect of the proposed solution?

To answer these questions, the artefact was designed to extract questions derived from security findings outputted from a CNAPP tool that scanned the cloud services hosted on multiple cloud platforms: AWS, Azure and GCP. By identifying and addressing potential vulnerabilities and weaknesses before deploying it, security measurements can be implemented to ensure a secure cloud environment.

The effect of the artefact was evaluated through a simple survey that assessed on a high-level; its ability to address the problem, usability, defined requirements, side-effects, and a comparison to similar tools. According to the feedback from participants, the artefact was found to be helpful and improved the threat modelling process by providing useful and valuable ideas for further brainstorming and discussions.

However, some expressed scepticism about integrating the artefact into their current processes. This could be due to the participants being new to threat modelling or only using it a few times through development processes. Cloud threat modelling is a novel approach that is still deemed immature and needs further research. The respondents who provided feedback on threat modelling are primarily using it on an application level and not for the cloud. Regardless, the participants appreciated the level of detail provided by the questions. Participants also recommended helpful suggestions on improvements which have been summarised and elaborated section 6.

In conclusion, cloud threat modelling is still something new, the results from the feedback survey showing that there are practitioners with little knowledge about cloud threat modelling. Training and awareness sessions could be held to help them to gain more knowledge on this topic as well. The artefact presented in this study serves as a foundation for future investigations, promoting a shift-left approach to implementing cloud security in the development cycle.

References

[1] Netwrix. *2022 Cloud Data Security Report*. https://www.netwrix.com/2022_cloud_data_security_report.html. (Accessed on 05/10/2023). 2022.

[2] Gartner. *Gartner Forecasts Worldwide Public Cloud End-User Spending to Reach Nearly $600 Billion in 2023*. https://www.gartner.com/en/newsroom/press-releases/2023-04-19-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-600-billion-in-2023. (Accessed on 05/06/2023). Apr. 2023.

[3] IBM. *Cost of a Data breach*. https://www.ibm.com/reports/data-breach. (Accessed on 10/10/2022). 2022.

[4] Mohamed Almorsy, John Grundy, and Ingo Müller. "An analysis of the cloud computing security problem". In: *arXiv preprint arXiv:1609.01107* (2016).

[5] Eric Kedrosky. *Worst AWS Data Breaches of 2021*. https://sonraisecurity.com/blog/worst-aws-data-breaches-of-2021/. (Accessed on 10/10/2022). Dec. 2021.

[6] Security Boulevard. *Most Cloud Breaches are Due to Misconfigurations*. https://securityboulevard.com/2019/04/most-cloud-breaches-are-due-to-misconfigurations-2/. (Accessed on 05/02/2022). 2019.

[7] Microfocus. *What is DevSecOps*. https://www.microfocus.com/en-us/what-is/devsecops. (Accessed on 06/07/2023).

[8] Gartner. *Market Guide for Cloud-Native Application Protection Platforms*. https://www.gartner.com/doc/reprints?id=1-2CX3G4KL&ct=230315&st=sb. (Accessed on 06/07/2023).

[9] K. H. Håkonsen and V. Ahmadi. *Threat analysis in agile*. Department of Computer Science, Norwegian University of Science and Technology. Unpublished, 2021.

[10] Paul Johannesson and Erik Perjons. *An introduction to design science*. Vol. 10. Springer, 2014.

[11] Adam Shostack. "Experiences Threat Modeling at Microsoft." In: *MODSEC@ MoDELS* 2008 (2008), p. 35.

[12] Loren Kohnfelder and Praerit Garg. "The threats to our products". In: *Microsoft Interface, Microsoft Corporation* 33 (1999).

[13] Tony UcedaVelez and Marco M Morana. *Risk Centric Threat Modeling: process for attack simulation and threat analysis*. John Wiley & Sons, 2015.

[14] Paul Saitta, Brenda Larcom, and Michael Eddington. "Trike v. 1 methodology document [draft]". In: *URL: http://dymaxion. org/trike/Trike v1 Methodology Documentdraft. pdf* (2005).

[15] Christopher Alberts et al. *Introduction to the OCTAVE Approach*. Tech. rep. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 2003.

[16] Katja Tuma, Gül Calikli, and Riccardo Scandariato. "Threat analysis of software systems: A systematic literature review". In: *Journal of Systems and Software* 144 (2018), pp. 275–294.

[17] Forbes. *The 'Shift Left' Is A Growing Theme For Cloud Cybersecurity In 2022*. https://www.forbes.com/sites/rscottraynovich/2022/01/13/the-shift-left-is-a-growing-theme-for-cloud-cybersecurity-in-2022/?sh=6a412b367ff1. (Accessed on 05/15/2023). 2022.

[18] Muhammad Younas et al. "Agile development in the cloud computing environment: A systematic review". In: *Information and Software Technology* 103 (2018), pp. 142–158.

[19] Imran Ghani, Nor Izzaty, and Adila Firdaus. "Role-based extreme programming (XP) for secure software development". In: *Special Issue–Agile Symposium*. 2013.

[20] P. Mell and T. Grance. "The NIST Definition of Cloud Computing". In: *National Institute of Standards and Technology, U.S. Department of Commerce, Maryland, MD, USA* (Sept. 2011), SP 800–145. DOI: 10.6028/NIST.SP.800-145.

[21] Oracle. *What is Cloud Native?* https://www.oracle.com/in/cloud/cloud-native/what-is-cloud-native/. (Accessed on 03/27/2023).

[22] AWS. *What is Cloud Native?* https://aws.amazon.com/what-is/cloud-native/. (Accessed on 03/27/2023).

[23] Skyhigh Security. *Secure Cloud-Native Applications & Infrastructure for Your DevOps Team*. https://www.skyhighsecurity.com/en-us/products/cloud-native-application-protection-platform.html. (Accessed on 05/16/2023).

[24] Yuri Diogenes and Erdal Ozkaya. *Cybersecurity–Attack and Defense Strategies: Counter modern threats and employ state-of-the-art tools and techniques to protect your organization against cybercriminals*. Packt Publishing Ltd, 2019.

[25] The MITRE Croporation. *MITRE ATT&CK: Design and Philosophy*. https://www.mitre.org/sites/default/files/2021-11/prs-19-01075-28-mitre-attack-design-and-philosophy. (Accessed on 03/28/2023). 2018.

[26] Kyriakos Kritikos et al. "A survey on vulnerability assessment tools and databases for cloud-based web applications". In: *Array* 3 (2019), p. 100011.

[27] Kennedy A Torkura and Christoph Meinel. "Towards cloud-aware vulnerability assessments". In: *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE. 2015, pp. 746–751.

[28] Kaoru Ishikawa and John Howard Loftus. *Introduction to quality control*. Vol. 98. Springer, 1990.

[29] Joseph S Dumas and Janice Redish. *A practical guide to usability testing*. Intellect books, 1999.

[30] Icek Ajzen and Martin Fishbein. "Attitude-behavior relations: A theoretical analysis and review of empirical research." In: *Psychological bulletin* 84.5 (1977), p. 888.

[31] GDPR. *Art. 32 GDPRSecurity of processing*. https://gdpr-info.eu/art-32-gdpr/. (Accessed on 05/12/2023).

[32] CSA. *Cloud Threat modelling*. https://cloudsecurityalliance.org/artifacts/cloud-threat-modeling/. (Accessed on 05/20/2023). 2021.

[33] Zeus Cloud. *CSPM False Positives*. https://www.zeuscloud.io/post/cspm-false-positives. (Accessed on 05/12/2023).

Appendices

A Link to full set of questionnaires for each cloud provider

Due to the size of the tables, a link is provided to view the full details of the artefact questions.

`https://docs.google.com/spreadsheets/d/1Lx5MM-USC8l-s010Z-tTh5XBM4MuMdnxgvMPFArr9WE/edit?usp=sharing`

A.1 Pre-study: Threat modelling in Cloud

Pre-study project from autumn 2022.

NTNU
Kunnskap for en bedre verden

DEPARTMENT OF COMPUTER SCIENCE

TDT4501 - DATATEKNOLOGI, FORDYPNINGSPROSJEKT

Threat modelling in cloud

Author:
Mai Hoang Tran

Supervisor:
Daniela Soares Cruzes

Abstract

Cloud computing is an emerging technology that several businesses are continually adopting to the platform by migrating their services to run in the cloud. It can be quite challenging due to its complicated architecture and the multiple services it can provide. Unfortunately, breaches occur and frequently it is on the client's end due to e.g. misconfigurations and poor security practices. This can be prevented with iterative threat modelling with early focus on implementing security from the beginning.

This study will conduct a systematic literature review on how threat modelling in the cloud, using the same research methodology from Tuma et al. and Håkonsen & Ahmadi but with a focal point on the cloud and not a general threat modelling approach. The SLR concluded with a five techniques from 2021-present and six techniques from the combined sets from both of the authors. The results were that input representation for threat analysis need improvement on characterising a good input as it needs a sufficient representation of the cloud environment. Otherwise, the techniques appear to be similar to a general approach to threat modelling, the difference being having an emphasis on the cloud environment which are accounted for in the inputs.

Additionally, the cloud threat modelling techniques are not fully adopted for the practitioners as it is difficult to use and integrate into practice and lack empirical testing and evidence on how it works. There is not any present stopping condition and there are very few techniques that address specific cloud threat at all.

Table of Contents

1 Introduction

The advantages the cloud computing offer have greatly outperformed traditionally self-hosted servers that depended on having a team to manage the services, which increase the cost of hosting the servers and having difficulties with scaling the infrastructure. The cloud paradigm introduces the benefits of self-provisioning of resources and a pay-as-you-go model, which mean the customer could seamlessly scale the resources based on incoming traffic and only being charged for the usage. This allow rapid development processes and leveraged smaller teams to move forward quicker, without having to worry about issues about the deployment.

This rapid adoption of cloud computing can also be reflected in the report evaluation provided by Gartner. The end-user spending on public cloud services only in 2022 is estimated to a growth of 20.4%, a total of $\approx$ \$495 billion dollars, compared to $\approx$ \$410 billion last year. This is predicted to steadily increase to an astounding $\approx$ \$600 billion in 2023 [1]. However, the adoption to cloud has also raised security concern regarding the complicated cloud model which introduces new dimensions related to the architecture, multi-tenancy, elasticity and layers dependency [2]. In 2021 multiple AWS clients suffered notable data breaches due to misconfigurations with impact on millions of users [3]. An average total cost of a data breach incident is estimated to $\approx$ \$4.35 million globally and over doubled in the United States. Nearly half of this are exploited in the cloud [4].

Therefore, continuously detecting these vulnerabilities in an early phase and evaluating the risks before major incidents occur will significantly decrease the costs associated with this. Threat modelling is an effective methodology to find security flaws within various of software systems [5]. The technique is aiming to identify essential assets, understanding the related threats and countering with the proper mitigations.

Although threat modelling is a widely adopted technique, there has been limited research on threat modelling in a cloud environment. The objective of this study is to conduct a systematic literature review on how the various threat modelling techniques in cloud are addressed in the literature. This paper will be an extension to Håkonsen & Ahmadi [6] systematic literature review on *Threat modeling in Agile* which is also based on Tuma et al. literature review on *Threat modeling* [7], but with a shifted focus in the context of cloud.

2 Background

The following section will present the referenced systematic review from Tuma et al. [7] and Håkonsen & Ahmadi [6], in addition to related theory surrounding threat modelling and cloud computing.

2.1 Related work

A systematic literature review on threat modelling has already been done and published by Tuma et al. in 2018 [7]. Their review concluded a lack of maturity in the techniques in terms of quality assurance of outcomes, validation, tool support and a clear definition of done on when the procedure should be finished. Subsequently another study by Håkonsen & Ahmadi [6] dated from 2021, extended their review upon this, but their research were more skewed toward the applicability of adopting the techniques to agile environment. An important finding of their results was a lack of integration to agile development processes. There was also an implication of future work on cloud domain which this study will have a focal point on. Thus, this study can be considered as an extension of these literature reviews with a primary subject on the cloud, by also including the primary studies from these studies that are relevant to the domain.

2.2 Threat modelling

No system is perfectly secure, consequently there will always exist attackers attempting to exploit the vulnerabilities. Therefore the risks should continuously be evaluated and strive to mitigate and counter them. It is the best interest of securing the software product by incorporating security early as possible in the design phase, as security breaches often severely damage the reputation of the business and cause major financial consequences which in many cases can be irreversible when occurred. Therefore, security should be implemented and integrated into the development practices from the beginning, and not added later [8].

Adam Shostack, the author of *Threat modelling: Designing for Security* describes threat modelling as a framework composed with steps that accomplish sub goals rather than performing a single activity, with the main objective to reduce the exposure of attack surfaces and mitigate vulnerabilities. It is about abstracting the bigger and detailed picture to catch the surfacing issues preventing them to result into bigger problems. He further compares it to a version control by emphasising on the essence of using it when building software. Instead of being a niche skill set, every professionals within the field should have basic experience with treat modelling. The technique is not developed immediately but acquired through several processes and iterations to strengthen the knowledge base [9].

2.3 Cloud computing

NIST defines cloud computing as a service providing a shared pool of large and configurable computational capabilities, resources, memory space and access to ubiquitous and convenient network. The services and resources should be rapidly provisioned and released to its clients with minimal management effort. It can be further recognised by its five essential characteristics, three service models, and four deployment models [10].

2.3.1 Characteristics

On-demand self-service - The client can without any human intervention access and self-provisioning the necessary computing resources as they desire to.

Broad network access - The network is consistently available and accessible for multiple devices.

Resource pooling - Using a multi-tenant model, the computing resources are pooled to dynamically delegate the resources depending on the consumer's demand.

Rapid elasticity - Gives the impression to the consumer of unlimited resources by allowing automatically scale the services quickly to meet the demand.

Measured service - The resource usage can be measured and determined to provide transparency between the cloud provider and the client. This is allowing the client to pay-peruse or charge-per-use basis on the utilised services.

2.3.2 Service models

Providing different type of flexibility depending on the service models.

Software as a service (SaaS) - Providing application processes to the consumer which can be accessible through e.g. web browsers or a program interface. The consumer does not have any control of the management of the underlying infrastructure or configuration, only limited use of specific application configurations.

Platform as a service (PaaS) - Providing an environment for the consumer to deploy application processes, enabling the consumer to develop, test or run their applications on the cloud infrastructure. The consumer is in control of the deployed application and the configuration settings in the hosted environment.

Infrastructure as a service (IaaS) - Providing physical computing infrastructure i.e. processing, storage, network and other computing resources to the consumer to run and deploy arbitrary software. The consumer does not manage the underlying infrastructure but do have control over operating systems, storage and deployed applications.

2.3.3 Deployment models

Control and visibility depend on the various deployment models.

Private cloud - Exclusively for a single organisation. Although it can be owned, managed and operated by the organisation itself, third party or a mixture.

Community Cloud - Exclusively for a specific community of consumers with shared goals in terms of e.g. mission, security, requirements, policy, compliance considerations. Can be operated in similar fashion as private clouds.

Public Cloud - Open for the general public.

Hybrid cloud - A composition of two or more cloud models, combining both capabilities.

3 Research method

The main objective of this paper is to review how threat modelling for cloud-based systems is addressed in the literature. The research methodology is based on conducting a systematic literature review following the steps as described in Kitchenham et al. [11]. This study will as mentioned earlier be an extension of Håkonsen & Ahmadi [6] recent study on Threat modelling in Agile which also was based on the systematic literature review by Tuma et al. [7]. Thus, the relevant papers from both studies are being included in the literature review, in addition to newer primary studies ranging from 2021-present.

The study of Håkonsen & Ahmadi [6] showed a lack of maturity for threat modelling in the cloud. Latifa et al. [12] conducted a similar SLR with emphasis on risks in the perspective of both the cloud service providers and clients. While this study will attempt to focus more on eliciting the techniques applied in cloud based on their characterisations, ease of adoptions, validation and addressed cloud threats.

3.1 Research questions

For this study, the research questions presented in Tuma et al. [7] and Håkonsen & Ahmadi will be reused, this time the focus being on characterising, adopting and validating the technique. Additionally adding another question whether the technique address one of the issues presented in the Cloud Star Alliance (CSA) Top cloud threats[1].

3.1.1 RQ1 - What are the main characteristics of the identified techniques?

The motivation behind this question is to identify the different aspects of the threat modelling techniques. This has been done through breaking it down to four categories, namely *applicability*, *input*, *procedure* and *outcome* which again are divided into their respective subcategories as more detailed in Table 1 and elaborated in section 3.4.1. The research question were originally proposed in Tuma et al. [7] and slightly modified in Håkonsen and & Ahmadi [6].

Applicability: captures which stages of the development process is applicable to. Varying from eliciting requirements, to reviewing the architecture or code of the system.

Input: identifies what kind of information is required to carry out the analysis in terms of the type and its representation.

Procedure: determines what activities are necessary for the analysis process. To reduce expert involvement, a knowledge base (KB) can be incorporated into the technique, and the level of precision indicate the quality of the performed analysis. Furthermore, it can evaluate the security objectives (i.e. CIA triad and accountability) and whether risk assessment is included in the analysis. Finally, a stopping condition can be observed to be present or not.

Outcome: aims to recognise what information has been gained from the analysis procedure. It can be assessed similarly to input but also accounting for assurance of quality and the degree of granularity.

3.1.2 RQ2 - What is the ease of adoption for the technique

The second research question aims to address the difficulty to adopt the selected techniques in practice. Tool support could accelerate the time adopting the technique and thus being beneficial for the practitioners. Guidance of execution helps determine how well the steps are detailed and what type of documentation is available. Finally, the target audience indicate which type

[1] https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-pandemic-eleven/

of practitioners the technique is intended for. Table 3 maps the level of competence between the target audience, originally imported from Tuma et al. [7], as well as the research question.

3.1.3 RQ3 - What evidence exist that the threat modelling technique work?

The third research question's goal is to identify in what degree the technique has been verified in the range of case studies, experiments and illustrations. This question has also been reused from the study of Tuma et al. [7] but was dropped in Håkonsen & Ahmadi [6].

3.1.4 RQ4 - What are the cloud threats addressed in the literature?

The last research question's purpose is to investigate whether of some of the technique specifically addressed any of the cloud threats. The catalogue of threats used in this study is issued by CSA[2] for the reason being the most updated and thoroughly detailed as opposed to OWASP[3] that provided a draft currently a work in progress.

3.2 Search strategy

The modified search strategy from Håkonsen & Ahmadi was repeated for the purpose of keeping it systematic, as papers from the previous SLR will be extracted and analysed. However, a notable difference is that the search was only used at a single digital library, namely Web Of Science[4]. The reason being that without it, using multiple libraries would mostly result in duplicates. Since the query was not adjusted for a specific domain like cloud, it would result in a tedious, time consuming and manual filtering process with little benefit to gain.

3.2.1 Snowballing

To compensate for the few papers found, the snowballing technique published by Wohlin et al. [13] was applied to find additional papers. The initial set consisted of the combined papers from Tuma et al. [7] and Håkonsen & Ahmadi and was used as a foundation to discover additionally papers. Furthermore, the papers identified by snowballing were selected through filtering based on the inclusion and exclusion criteria.

3.3 Inclusion and exclusion criteria

As this study is based on the other SLRs, the inclusion and exclusion criteria is naturally inherited with a tweak for assessing cloud related papers, as well as adjusting the year range to 2021 and until present.

3.3.1 Inclusion

- Primary studies.
- Published between 2021 until present.
- Studies (i.e.) that address methodologies, methods or techniques for identifying, prioritising and analysing security threats to a system deployed in cloud.
- Studies related to security of cloud-related system.

[2]https://cloudsecurityalliance.org/
[3]https://owasp.org/www-pdf-archive/OWASP_Cloud_Top_10.pdf
[4]https://webofscience.com

3.3.2 Exclusion

- Studies written in any language other the English language.
- Short publications and posters (< 5 pages).
- Publications that were unavailable through the search engine.
- Studies that focus on concrete mitigation strategies, security solutions, taxonomies of security threats and security analysis of systems.
- Studies that focus on anomaly detection and intrusion detection systems.
- Publications about safety-hazard analysis and detection methods and studies investigating the relationships between safety and security requirements.

3.4 Data extraction

The data extraction template was used during the review process to extract the relevant data to answer the research questions, showcased in Table 1, Table 2 and Table 4. The tables for research questions one and two are identical to Håkonsen & Ahmadi [6] modified template while the template for research question three is similar to Tuma et al. [7] with the distinction of removing *Domain* and *Validator* as subcategories since domain is implicit cloud and most of the papers did not explicit state any validators. Also, Tuma et al. also appear not to address the validator subcategory in their original SLR [7] as well. Additionally Table 3 was supplemented to to answer sub question *Target Audience* in RQ2. The template for the last research question was added to explore whether the techniques addressed any specific cloud threats.

Table 1: Data extraction template for RQ1 used in Tuma et al. [7]

	Characterization	
Applicability	Level of abstraction	Requirement level Architectural level Design level Implementation level
Input	Type	Goals Requirements Attacker behaviour Security assumptions Architectural design Code Assets Other
	Representation	Textual description Model-based Other
Procedure	Knowledge-based	No Yes
	Level of precision	None Based on examples Based on templates Semi-automated Based on a formal framework (Very precise) Automated
	Security objectives	Confidentiality Integrity Availability Accountability Not applicable
	Risk	Not considered Internal part of technique Externally considered

	Stopping condition	Present Not present
Outcomes	Type	Mitigations Threats Security requirements
	Representation	Structured text Model-based Other
	Assurance of quality	Explicit Present Not present
	Granularity	High level Low level

3.4.1 RQ1 - Characterisations

Applicability - Contain four levels of abstraction that describe to which extent the analysis can begin with. *Requirement level* denotes the lowest level of them all which does not require much but the system requirements to start the process. While on an architectural level needs more knowledge of the system and its relations to other sub parts in order to continue. Design level could be expressed on a more refined architectural level (e.g. having design patterns) and implementation level implies that the procedure can be done on the fully implemented system with its source code.

Input - The input is categorised into type: *goals, requirements, attacker behaviour, security assumptions, architectural design, code, assets and other* and how they are represented: *textual description, model-based or other*. Goals could be any high-level objective of the system or anti-goal (i.e. malicious objective from the perspective of an attacker to exploit). Requirements represent the raw system requirements elicited through the early phase. Architectural design captures the relations between components of the system such as the data flow. Attacker behaviour describe the activities (e.g. historical data) the attacker carries out. Security assumptions are any assumptions made to describe the environment of the system related to security concerns. Code could be the source code or infrastructure as code needed for the analysis. Other is a category available if any of the other categories did not fit.

Procedure - A knowledge base was considered to be any external source of information incorporated into the technique to help the practitioners during the procedure.

The level of precision ranged from *none, based on examples, based on templates, semi-automated, based on a formal framework and automated*. Having no precision implied no structure or any guidelines on how to proceed with the technique and indicating low level of precision. Furthermore, the procedure can be supported by providing examples but does not give any precise or detailed guidelines, or a template to fill in which raises the level of precision. The procedure could also adhere to a formal framework with systematic and precise steps of performing the technique. Also, the procedure could also be partially (semi-automated) or fully automated.

The security objectives are evaluated accordingly to the CIA triad and accountability, and to which extend risk assessment is included, either as part of the technique or outside, or simply not considered at all. Finally, a stopping condition denotes whether the technique had a definition of done when the procedure was finished.

Outcomes - The type of the outcome can result in *mitigations, threats* or *security requirements*. Mitigations can be defined as any countermeasures made to reduce the risk of the threats. Threats are identified as anything that can harm and exploit the system. Security requirements are functional requirements that ensure the security objectives of the system.

3.4.2 RQ2 - Ease of adoption

Table 2 are divided into four categories with each attempting to answer the question for adopting the technique in terms of *tool support*, *guidance for execution*, *documentation* and *target audience*.

If the technique was supported by a tool, it could either be a complete and well tested tool or a prototype tool that are still in development stage and need further testing. The guidance of execution indicate how nuanced the steps are defined. Fine grained steps are very clear and well described and coarse-grained steps provide more a general guideline without any further elaboration beyond the procedure.

Documentation for the technique could be found available as a *publication*, *tutorial*, *presentation*, *tool documentation* and *demonstration*. Lastly, the target audience represent the minimum knowledge the technique is required for. Further explanations of the differences between the groups are elaborated in Table 3.

Table 2: Data extraction template for RQ2 used in Tuma et al. [7]

	Ease of adoption
Tool support	None Prototype tool Tool
Guidance for execution	Coarse-grained steps Fine grained steps No structure
Documentation	Publication Tutorial Presentations Tool documentation Demonstration
Target audience	Engineer Security trained engineer Security expert Researcher

Table 3: Description of target audience used in the data extraction process for RQ2, originally used in Tuma et al. [7].

Target audience	**Level**	**Major tasks**	**Exemplary title**
Engineer	L1	Tool support, low-level implementation, testing, and maintenance	Junior Software Developer,Acceptance tester, Junior Security Engineer, Software Assurance Technician
Security trained engineer	L2-L3	Requirements fundamentals and analysis, architectural design, implementation, risk analysis and assessment	Security Analyst, Release Engineer, Information Assurance Analyst, Maintenance Engineer, Senior Software Developer, Software Architect
Security expert	L4-L5	Assurance assessment, assurance management, risk management across the SDL, advancing in the field by developing, modifying, and creating methods, practices, and principles at the organizational level or higher	Project Manager, Senior Software Architect, Chief Information Assurance Engineer, Chief Software Engineer
Researcher	-	Remain in touch with the current research and publish own research in the discipline of security in software engineering	PhD student, Post Doctoral candidate, Assistant Professor, Senior lecturer, etc.

3.4.3 RQ3 - Validation

The possible alternatives for validation are *case study*, *experiment* or *illustration*. Case study is a loosely and broadly used term but for the scope of this study, is described as a reasonably detailed

examples applied on the technique. Experiments look at the empirical data for verification and illustrations are more lightweight examples slightly less detailed than case studies.

3.4.4 RQ4 - Addressed cloud threats

If the technique specifically address any cloud threat issued in CSA Top cloud threats, it can be marked in the *yes* column and further elaborate which issue it addresses in section 5.

Table 4: RQ3 and RQ4 related to the data extaction template. [7]

	Validation (RQ3)
Tool support	Case study Experiment Illustration
	Cloud threats (RQ4)
Addressed threats	Yes No

3.5 Quality assurance

Quality assurance was conducted through weekly meetings with the supervisor to establish goals and objectives for the following week during the entire process from planning, filtering and extracting the papers. During the filtering phase, promising but questionable and unsure primary studies were discussed with supervisor to review the relevance of them. Furthermore, the papers were also skimmed and evaluated through multiple iterations to ensure whether it truly satisfied the criteria.

The extracting phase was first conducted by reviewing the papers from the Tuma et al. [7] and Håkonsen & Ahmadi [6] to ensure if our understanding of the data extraction template were aligned. Thus, it was used as a control check to ensure correctness for the data extraction between the the older and newer primary studies.

4 Results

The results consist of extracted data from the papers filtering process and papers from both Tuma et al. [7] and Håkonsen & Ahmadi [6]. This is highlighted by the dashed lines in the table to differentiate the set of papers. The main findings will be presented and summarised in this section, using the data extraction templates in Table 1, Table 2 and Table 4 as a baseline. Furthermore, the results will be discussed in section 5.

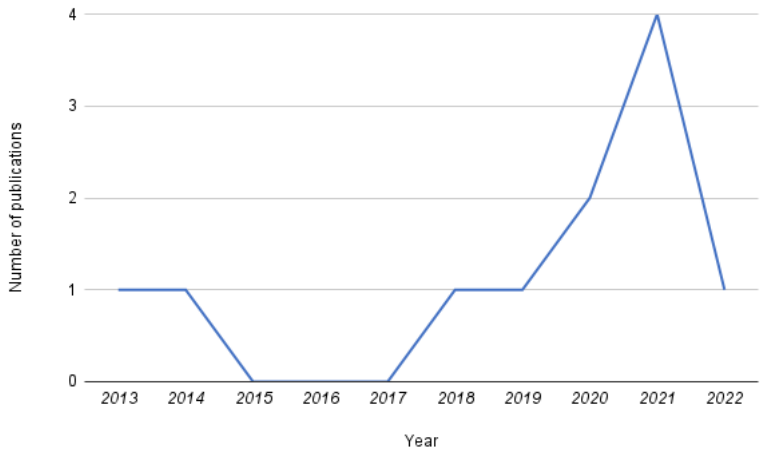


Figure 1: Number of publications per year

Figure 1 gives an overview of the number of publications per year from this SLR. No papers related to cloud were found between 2015-2017, which doesn't necessary imply that no research about cloud threat modelling were done but perhaps more difficult to find through the search strategy. From 2017 and onward, the trend steadily increases by each year with except for 2022 with a minor drop. This is due to the paper gathering were done in September and thus does not represent the entire year.

Table 5 provides a quick overview of the selected techniques with a short summary on how they work. As seen, the techniques vary from using the conventional graphs in terms of simple node graphs [14], misuse patterns [15] and use cases [16], attack trees [17] to more complex and auto generated models [18], [19]. Classifying and listing the threats in tables is also present [20].

Table 5: Overview of techniques

Methodology	Ref	Technique
Granata et al.	[14]	Mapping the cloud assets and their relations in a graph. Using a proposed taxonomy for threat selection.
Xiong et al.	[18]	Creating a domain specific language (DSL) based on Meta Attack Language (MAL) to simulate attacks. Using MITRE ATT&CK as KB.
Elahi et al.	[20]	Mapping intelligent mobile applications to attack vectors. Threat vectors are categorized in terms of impact assessment.
Mondal et al.	[17]	Attack and Attack-defence trees of Serverless computing.
Gilliam van der Merwe et al.	[19]	Knowledge graph representation of technqiues and targets specific related to APT, using MITRE ATT&CK Cloud matrix as KB.
Hong et al.	[21]	Exhaustive list of attack categories mapped to cloud assets, countermeasurs, STRIDE and OWASP categories
SSDE	[22]	Automatic threat selection from own Threat catalogue (constructed from (OWASP top 10, OAUTH, SSL threat model, CSA top threats))

Mouratidis et al.	[23]	Create goal cloud model, automatic elicitation of threat, risk, vulnerability, mitigation, asset, and actors. Uses external and internal KB (such as NVD)
Jouini et al.	[24]	Create 4+1 view model, decompose into multiple dimensions, then brainstorm threats
Beckers et al.	[16]	MUC
Encina et al.	[15]	Misuse patterns

4.1 Results from data extraction RQ1

Table 6: Input and applicability characteristics

		Applicability Abstraction				**Input Type**								**Representation**		
Methodology	**Ref**	Requirements	Architectural	Design	Implementation	Assets/Cloud resources	Requirements	Attacker behavior	Security assumptions	Architectural design	Infrastructure as code	Goals	Other	Textual description	Model-based	Other
Granata et al.	[14]		•			•				•				•	•	
Xiong et al.	[18]		•			•		•	•						•	
Elahi et al.	[20]	•				•	•							•		
Mondal et al.	[17]	•						•				•			•	
Gilliam van der Merwe et al.	[19]	•							•				•		•	
Hong et al.	[21]		•							•					•	
SSDE	[22]		•						•	•	•			•	•	•
Mouratidis et al.	[23]	•							•			•		•	•	
Jouini et al.	[24]		•							•					•	
Beckers et al.	[16]	•								•					•	
Encina et al.	[15]	•						•						•		

4.1.1 Applicability

As showcased in Table 6, it can be observed that it is almost evenly split between applying the techniques at an architectural or requirement level.

Beckers et al. [16] follow a requirement engineering approach eliciting the requirements and patterns necessary before starting the analysis. Mouratidis et al. [23] obtain the inputs about goals and components on a requirement level before proceeding further. The same also apply for for Elahi et al. [20], Mondal et al. [17], Gilliam van der Merwe et al. [19] and Encina et al. [15] that need to acquire information and raw data about the application for the analysis.

For the remaining techniques, an architectural understanding of the system is essential. Xiong et al. [18] need a structural view in order to model it. Granata et al. [14] rely on MACM (Multi-application Cloud Composition Model) which shows the relations between the cloud components as input for the technique. Jouni et al. [24], SSDE [22] and Hong et al. [21] assess the architecture before proceeding further.

4.1.2 Input

The most popular type of input is *Architectural design* with nearly half $\frac{5}{11} = 45\%$ of the possible inputs. This was mostly related to having a "blueprint" in terms of e.g. graph-based model to visualise how the components was related, or at least having an understanding of the underlying structure. Out of the input representation, model-based comes first with over 72% (out of 11

possible) while textual description estimated 55%. Only one is considered as *Other* which was source code from code reviews found in SSDE [22].

A close runner-up is *Security assumptions* which in the papers were extracted through the knowledge base [19], [18], [23]. While in SSDE [22], it were described as security objectives related to the cloud components.

The next types are split between *Attacker behaviour* and *Assets*. In Encina et al. [15] and Mondal et al. [17] analyse the misuse patterns and scenarios from the perspective of an attacker while Xiong et al. [18] draw the assumptions from the knowledge base. Assets are recognised as cloud assets or components which are prevalent in creating a model in Xiong et al. [18] and Granata et al. [14] or mapping the inputs with threat conditions in Elahi et al. [20].

The following type is *Goals* which in Mouratidis et al. [23] represents capabilities of the associated cloud asset. Conversely, in Mondal et al. [17] describe what the attacker wants to achieve.

Finally, the remaining that have been reported once in the methodologies, *Requirements*, *Code* and *Other*. Elahi et al. [20] elicit raw data requirements from the application with little to no refinement. Repeated code reviews are included in the process as stated in SSDE [22] which makes code as an input. *Other* refer to the data set[5] in Gilliam van der Merwe et al. [19] that classify adversarial tactics and techniques used as a knowledge base.

4.1.3 Procedure

The majority have incorporated a knowledge base into the technique, while two require expert knowledge when performing the technique. Examples of KBs found in the SLR are a list of security controls according to NIST[6], OWASP Risk rating methodology [14], MITRE Enterprise ATT&CK Matrix [18], MITRE ATT&CK Cloud Matrix [19], CSA cloud threats [16], NIST national vulnerability database[7] [23], Threat Catalogue, CVE[8] [22] and OWASP attack categorising [21]. A full overview can be seen in Table 12.

The level of precision primarily consisted of using examples which involve motivating the steps of applying the technique through examples given in the article. This could also be supplemented by using templates, framework or tool to automate the process in various degree.

It was reported that five techniques could be semi-automated, e.g. Mouratidis et al. [23] propose a model that continuously could be enhanced with new security knowledge that aid the process to be automated. Gilliam van der Merwe et al. [19] utilise a formal concept analysis (FCA) that also could be used to be partially automated. Granata et al. [14] aim to automate threat identification according to assets and protocols and risk ranking, but do require some manual intervention. The technique highlighted in Xiong et al. [18] generate attack graphs automatically but require to create a domain specific language to represent the cloud environment. SSDE [22] provide a template as part of the technique which also have the assessment phase semi-automated. Beckers et al. [16] also include templates to support for eliciting the requirements. Lastly, two of the techniques were reported to have any level of precision.

Security objectives as confidentiality, integrity and availability were found to always being accounted for in the techniques. Slightly less than half ($\approx$ 45%) did not cover accountability. However, it varied on how accountability was evaluated, from optionally including it in misuse activities [15] and keeping traceability of data through privacy patterns [16]. Other examples were having accountability assessed through measuring impact in risk assessment [22], [14].

There is a mix result between assessing risk externally, internally or not considered at all but with a small majority of five incorporating risk as an internal part of the technique. Four does not consider risk in the technique while the remaining two consider risk assessment externally. Risk

[5]https://attack.mitre.org/matrices/enterprise/cloud/
[6]https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final
[7]https://nvd.nist.gov/
[8]https://www.cve.org/

Table 7: Procedure characterstics

		KB		Precision						Objectives					Risk			Stopping condition	
Methodology	**Ref**	Yes	No	None	Based on examples	Based on templates	Semi-automated	Very precise	Automated	Confidentiality	Integrity	Availability	Accountability	Not applicable	Not considered	Internal part of technique	Externally considered	Present	Not Present
Granata et al.	[14]	•			•		•			•	•	•	•			•			•
Xiong et al.	[18]	•			•		•			•	•	•			•				•
Elahi et al.	[20]		•		•					•	•	•				•			•
Mondal et al	[17]		•	•						•	•	•					•		•
Gilliam van der Merwe et al.	[19]	•					•			•	•	•	•		•				•
Hong et al.	[21]	•			•					•	•	•	•		•				•
SSDE	[22]	•				•	•			•	•	•	•			•			•
Mouratidis et al.	[23]	•			•		•			•	•	•				•			•
Jouini et al.	[24]		•		•					•	•	•				•			•
Beckers et al.	[16]	•				•				•	•	•	•				•		•
Encina et al.	[15]		•	•						•	•	•	•		•				•

in Beckers et al. [16] was only used as a metric to assess an asset to whether it required an extra control, which an external risk analysis would be performed. In Mondal et al. [17] assumed that a statistical analysis of probability of the attacks will be carried outside the technique. SSDE [22] and Granata et al. [14] are both using OWASP Risk rating methodology as part of risk analysis while other examples is to evaluate the impact of threats associated with targeted assets (e.g. in terms of security objectives) as described in Mouratidis et al. [23] and Jouni et al. [24]. Lastly, Elahi et al. [20] calculate the risk by using predefined equations for severity and probability.

None of the technique had an explicit stopping condition for when the the process is finished, which is not surprising as previous results in Tuma et al. [7] and Håkonsen and Ahmadi [6] indicated similar outcome which also appear to be the case in this study as well.

4.1.4 Outcomes

The results in Table 8 show the most reported outcome type were threats followed by mitigations and then lastly security requirements. Furthermore, all of the techniques represented the outcomes in terms of structured text and only a single technique also had model-based. Only one of the listed technique did not have threats as it outcome type, which was SSDE [22], the security requirements and countermeasures were defined in the risk analysis for each component in the application. Mouratidis et al. [23] follows various cloud analysis techniques which include threat analysis for identifying threats, transparency analysis to elicit security requirements and security mitigation analysis for mitigating the threats. The results of the analyses are then stored in the different models. The security requirements in Beckers et al. [16] are acquired through using the threat as a basis to extract the necessary requirement. The threats are gathered from CSA top cloud threats and OWASP are further mapped in relation with security and privacy goals. Other types of threats used attack categorisation from OWASP and mapped to STRIDE and its cloud component. Hong et al. [21] list the threats as tables. Gilliam van der Merwe et al. [19] show the connections between Advanced Persistent Threat (APT) groups and their cyberattack techniques, similarly in Xiong et al. [18] show the attack paths and mitigations could be displayed as defence graphs. Attack tree represented the possible attack scenarios and conversely attack defence tree represented countermeasures for mitigations, which was found in Mondal et al. [17].

Explicit step of verification of the outcomes were absent in reviewed techniques. Only seven out of eleven had an informal way of verifying the outcomes. All of the techniques also had a high level of granularity, e.g. misuse diagrams, graphs and tables. None had any low-level specific and detailed scenario as its outcome.

Table 8: Outcome characterstics

		Type			Representation			Quality assurance			Granularity	
Methodology	**Ref**	Mitigations	Threats	Security requirements	Structured text	Model-based	Other	Explicit	Present	Not present	High-level	Low-level
Granata et al.	[14]	•	•		•				•		•	
Xiong et al.	[18]	•	•		•				•		•	
Elahi et al.	[20]	•	•		•					•	•	
Mondal et al	[17]	•	•		•					•	•	
Gilliam van der Merwe et al.	[19]		•		•				•		•	
Hong et al.	[21]		•		•				•		•	
SSDE	[22]	•		•	•				•		•	
Mouratidis et al.	[23]	•	•	•	•	•			•		•	
Jouini et al.	[24]		•		•					•	•	
Beckers et al.	[16]		•	•	•				•		•	
Encina et al.	[15]	•	•		•					•	•	

4.2 Result from data extraction RQ2

Over half of the identified techniques were reported to be supported by tools during the procedure. Examples of tools used were open source project such as Apparatus Software[9] used in Mouratidis et al. [23] which provide a graphical interface to create visual models and perform semi-automated security analysis. Beckers et al. [16] use UML4PF framework and problem-based privacy analysis (ProPAn) method with their tool support to respectively create diagrams and identifying privacy threats to generate the graphs. The SSDE [22] approach suggests tools as CAMEL[10] to model the cloud applications. Gilliam van der Merwe et al. [19] propose Lattice Miner[11], an open source project for representing the lattice graphs and Protégé ontology editor[12], another open source project to build ontologies which also conforms to the web semantics standards. Additionally Neo4j[13], also an open source project was used in Granata et al. [14] as a graph database and SecuriCAD is used simulate attacks in Xiong et al. [18].

Among the eleven techniques, only one had no structure in guidance for execution. Encina et al. [15] did not provide any detailed guidelines in how to conduct the technique. The remaining did provide more detailed execution steps but were more general guidelines and not fine-grained and explicit guidelines. All of the techniques are found through publications and four of them also had documentation of the supported tool. Many of these were as stated earlier, open source projects.

Engineer was found to be the most frequent main target of audience among the identified techniques in Table 2. They are qualified to construct low-level implementation such as attack and defence graphs as reported in Mondal et al. [17] and misuse patterns described in Encina et al. [15]. Additionally, the steps detailed in Encina et al. [20] are manageable for an engineer to perform. They are also supported by semi-automated tool during the process as stated in [22] with one of the main objective to reduce involvement of security experts. Likewise, Mouratidis et al. [23] propose tools to be assist non experts. The next common target audience are security trained engineers which require more analytical skills. It was identified in the techniques that architectural knowledge of the application was needed and performing risk analysis as addressed in Granata et al. [14] and Xiong et al. [18]. Beckers et al. [16] also require an essential understanding of security and privacy to elicit the requirements and map the patterns. The least common groups of audience were security experts and researcher. In Gilliam van der Merwe et al. [19], an security expert is needed to derive the patterns and signatures of APT from the data, while in Hong et al. [21] need a security expert due to address the attack scenarios. Finally, a researcher was

[9] https://github.com/NOMNUDS/apparatus
[10] https://paasage.ercim.eu/training-materials/camel-modelling-tutorial
[11] https://github.com/LarimUQO/lattice-miner
[12] https://protege.stanford.edu/
[13] https://neo4j.com/

Table 9: Ease of adoption

		Tool support			Execution			Documentation					Target audience			
Methodology	**Ref**	None	Prototype	Tool	Coarse-grained	Fine-grained	No structure	Publication	Tutorial	Presentations	Tool docum.	Demonstration	Engineer	Sec trained Engineer	Security expert	Researcher
Granata et al.	[14]			•	•			•			•			•		
Xiong et al.	[18]			•	•			•						•		
Elahi et al.	[20]	•			•			•					•			
Mondal et al	[17]	•			•			•					•			
Gilliam van der Merwe et al.	[19]			•	•			•			•				•	
Hong et al.	[21]	•			•			•							•	
SSDE	[22]			•	•			•			•		•			
Mouratidis et al.	[23]			•	•			•			•		•			
Jouini et al.	[24]	•			•			•								•
Beckers et al.	[16]			•	•			•						•		
Encina et al.	[15]	•					•	•					•			

Table 10: Type of validation

		Validation		
Methodology	**Ref**	Case study	Experiment	Illustration
Granata et al.	[14]		•	
Xiong et al.	[18]		•	
Elahi et al.	[20]	•		•
Mondal et al.	[17]	•		
Gilliam van der Merwe et al.	[19]			•
Hong et al.	[21]	•		
SSDE	[22]	•		
Mouratidis et al.	[23]	•		
Jouini et al.	[24]			•
Beckers et al.	[16]			•
Encina et al.	[15]	•		

necessary to perform the technique in [24] due to the complex decomposition of the dimensions and multidimensional assessment.

4.3 Results from data extraction RQ3

The main validation type was case study as shown in Table 10. Examples of case studies varied from theoretical examples of a specific scenarios as found in [17], [21], [23], [15]. Additionally, Elahi et al. [20] included research workshop sessions provided with feedback for improvement and SSDE [22] verified the technique through some real world applications as case studies. The second next type was illustration which were applied to four of the techniques. [19], [20], [24] and [16] verified it through giving lightweight examples that are less detailed than case studies. Lastly, experiments were used for two of the techniques as validation. Granata et al. [14] compared the outcome of the technique with the results of Microsoft Threat modelling tool and Xiong et al. [18] simulate the attacks on the generated model to compute amount of steps required to attack.

4.4 Results from data extraction RQ4

Only one out of eleven techniques specifically address at least one of the threats listed in CSA top threats to cloud computing. The technique showcased in Gilliam van der Merwe et al. [19] is specialised toward finding patterns and signatures related to Advanced Persisent Threat and thus address the security issue 10: *Organised Crime, Hackers & APT* in the recently published CSA report.

Table 11: Addressed cloud threats

Methodology	Ref	Addressed specific cloud issues: Yes	Addressed specific cloud issues: No
Granata et al.	[14]		•
Xiong et al.	[18]		•
Elahi et al.	[20]		•
Mondal et al.	[17]		•
Gilliam van der Merwe et al.	[19]	•	
Hong et al.	[21]		•
SSDE	[22]		•
Mouratidis et al.	[23]		•
Jouini et al.	[24]		•
Beckers et al.	[16]		•
Encina et al.	[15]		•

5 Discussion

A general discussion about the findings in systematic literature review will be elaborated in this section, as well as future work and threats to validity in this study.

5.1 General discussion about the literature

Four main points about the findings in the literature will be discussed addressed by the research questions.

5.1.1 Cloud threat modelling technique

Input representation From the results, it is noticeable that a majority of the inputs are represented as models. This can be challenging as there is not always the models guarantee a sufficient coverage of the entire system as the cloud architecture, as known for its complexity serving multiple purposes with its interconnected services, thus lack of visibility is one of the main problem when modelling a cloud environment. None of the reviewed techniques had any verification on how well the cloud was accounted for, which future techniques could consider to include.

Knowledge base It can be observed that techniques with incorporated knowledge base had among the highest level of precision (i.e. semi-automated and templates) with the exception of Hong et al. [21]. This does not directly imply the quality of the technique, especially drawing conclusions from a relative small sample of data without supported empirical evidence. It does however pose an interesting area to investigate the correlation and the quality of the knowledge bases used. A table overview of the knowledge base used by the different techniques are displayed in Table 12. Among the present knowledge base, MITRE&CK Cloud matrix stands out in terms of being more specific regarding detection and mitigation in the cloud, while the others contain general guidelines and exhaustive lists of threats and attacks. The techniques that utilised the cloud matrix as a KB was also the only one that addressed specific cloud threats which will be elaborated more in section 5.1.4.

Stopping condition Håkonsen & Ahmadi [6] reported a majority of the reviewed techniques with no stopping condition. Tuma et al. [7] also discussed the definition of done that is absence during threat modelling procedures. The results shown in section 4.1.3 reflect a similar trend about the lack of a present stopping condition in the procedure. Establishing the definition of done depends on the team which has proved to be challenging [25]. Therefore, having guidelines on how to determine a definition of done is highly valued and beneficial for the entire team. An idea could potentially be combining MITRE&CK Cloud matrix to establish a stopping condition as it provide preventive mitigation controls. This could be elicited to create measurable questions e.g. checklist.

Overall, there was not any other findings that differentiated a cloud technique versus a general threat modelling approach with respect to the characterisations (applicability, input, procedure and outcome). It was identified that the cloud environment should be clearly defined in inputs and the remaining activities were similar to a general threat modelling approach with the exception of using a specific cloud knowledge base.

Table 12: Overview of the knowledge bases used

Methodology	Ref	KB
Granata et al.	[14]	NIST framework, OWASP Risk rating framework.
Xiong et al.	[18]	MITRE ATT&CK Enterprise matrix.
Elahi et al.	[20]	x
Mondal et al.	[17]	x
Gilliam van der Merwe et al.	[19]	MITRE ATT&CK Cloud matrix.

Hong et al.	[21]	OWASP attack categorizaion.
SSDE	[22]	Threat catalogue, CVE.
Mouratidis et al.	[23]	NIST national vulnerability database.
Jouini et al.	[24]	x
Beckers et al.	[16]	CSA Cloud threats
Encina et al.	[15]	x

5.1.2 Difficult adaption to the techniques

Although most techniques are supported with tools, there were still a significant amount without any tool support which could cause a steep learning curve for the practitioners making it less prioritised to use. Additionally, precise guidelines for the procedure could potentially be further developed to be more specific, helping to rule out any difficulties that may occur. The target audience contain largely of engineers but for a complex domain as cloud, minimising the need of a deep security knowledge could intensive other developers to use it frequently. Overall, there is still areas of improvements that will ease the process to adopt such as tool support for more automation, clear cut steps and more security beginner friendly target group which in return could result in more maturity in this field of study.

5.1.3 Lack of sufficient validation

Many of the cloud techniques are verified through case studies and illustrations which often were found to be simple toy examples. Experiments were carried out with simple measurements that does not necessarily reflect real scenarios. This show that the techniques are only prototypes that are not necessary applicable in practice. Thus, further research and empirical testing is needed in order to improve the maturity in the field.

5.1.4 Addressed cloud threats

There was only one technique that specifically addressed the cloud threats, i.e. Gilliam van der Merwe et al. [19] on identifying advanced persistent threats (APT) linked to security issue ten *Organised Crime, Hackers & APT* in CSA's enumeration of cloud threats. The lack of specific addressed cloud technique could be due to threat modelling being a generalised technique aiming to identify all type of vulnerabilities and threats. Nevertheless, it is still worthy to search for specific techniques as well and analyse their compatibility with others although this was not in scope for the project. However, it was discovered that the knowledge base used i.e. MITRE ATT&CK Cloud matrix was the significant element that differentiated with the other techniques. An interest topic is to look into how to take advantageous of this knowledge base in a greater degree, exploring which aspects of the cloud it cover and not. In that way, it can be combined with the general threat modelling approach with its beneficial features and balance out the shortcomings of other techniques.

5.2 Future work

As lightly discussed, there are multiple areas for improvements but the main ones that were identified in this study were explicit stopping condition, guidelines for sufficient inputs, MITRE ATT&CK Cloud Matrix, feasible technique and extensive validation.

5.2.1 Explicit stopping condition

Evidently, it is proved to be difficult introducing a stopping condition which each team has to decide. It was found that MITRE ATT&CK Cloud matrix was an interesting knowledge base that could benefit mapping a stopping condition with its cloud specific guidelines. Future research could address this topic.

5.2.2 Guidelines for sufficient inputs

Having requirements or guidelines on what defines a sufficient input is a crucial part of threat modelling as it will affect the following process. This can result in a detailed overview of which security aspects of the cloud needed to be considered in the inputs. Therefore, future research could investigate on finding these solutions.

5.2.3 Feasibility and extensive evaluation

Another aspect of threat modelling is its practitioners, who are the only source that can provide feedback and empirical data on how the techniques work in practice. Therefore having a feasible technique will intensives a larger group of audience to use it. To achieve this, there is a necessity to make it feasible in terms of supported tool and automated flow with well described guidelines, and not requiring too advanced security knowledge. As discussed in section 5.1.1 and section 5.1.4, MITRE ATT&CK could potentially be utilised to automate the threat modelling, making it more easier to adopt to.

5.3 Threats to validity

The review was based on previous studies from Tuma et al. [7] and Håkonsen & Ahmadi [6] using their general search strategy to find the papers which naturally cover a wider and general scope, limiting the probability of finding specific papers related to cloud. It resulted in a total of four papers from 2021-present compared to six papers from 2000-2021 with 2013 being the earliest paper about cloud. Additionally, using the snowballing method resulted in finding another paper, adding up to a total of five papers from 2021 and onward. The small sample of papers made it difficult to analyse and infer the results due to the observations being biased due to its size. Therefore, modifying the search query to adjust explicitly for the cloud domain could have been done to increase the sample size. However, the decision was made in consultation with the supervisor to adhere to their search strategy in order to keep it systematic throughout the entire process. Consequently, this pose a threat to validity in this study.

6 Conclusion

A systematic literature review have been conducted on how the literature is addressing threat modelling in cloud. It was supported with research questions to answer this, categorised into characteristics on the techniques, validation and adoption of techniques, additionally to whether it address any specific cloud threats. A total of five papers were extracted from the year 2021-present, combined with six papers from Tuma et al. [7] and Håkonsen & Ahmadi [6] spanning from 2013-2021.

Overall, there were not many characteristics that distinguished the threat modelling techniques in cloud versus a general approach for any software system. The main key is to elicit good inputs with consideration of a cloud environment during the start phase as it will greatly impact the rest of the procedure. Having a good input is essential, as the cloud environment should be defined during this step. MITRE ATT&CK appeared to perhaps include relevant and cloud specific attributes that could potentially be utilised in some degree of automating during the procedure. Furthermore, there is lack of validation i.e. of how the techniques work in practice as many techniques only serve as a proof of concept, and thus require to be supported by empirical evidence before proceeding.

Further research could examine the possibility to automate the tools to help the practitioner during threat modelling. Utilising the MITRE ATT&CK Cloud Matrix could also benefit as it address specifically the cloud domain and its tactics and adversarial techniques. Defining and characterising a good representation of an input of the cloud would help the practitioners to move forward with a solid understanding of its interconnected cloud and leading to make a better judgement on the attack surfaces and countermeasures.

References

[1] Gartner. *Gartner Forecasts Worldwide Public Cloud End-User Spending to Reach Nearly $500 Billion in 2022*. https://www.gartner.com/en/newsroom/press-releases/2022-04-19-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-500-billion-in-2022. (Accessed on 10/09/2022). Apr. 2022.

[2] Mohamed Almorsy, John Grundy, and Ingo Müller. "An analysis of the cloud computing security problem". In: *arXiv preprint arXiv:1609.01107* (2016).

[3] Eric Kedrosky. *Worst AWS Data Breaches of 2021*. https://sonraisecurity.com/blog/worst-aws-data-breaches-of-2021/. (Accessed on 10/10/2022). Dec. 2021.

[4] IBM. *Cost of a Data breach*. https://www.ibm.com/reports/data-breach. (Accessed on 10/10/2022). 2022.

[5] Adam Shostack. "Experiences Threat Modeling at Microsoft." In: *MODSEC@ MoDELS* 2008 (2008), p. 35.

[6] K. H. Håkonsen and V. Ahmadi. *Threat analysis in agile*. Department of Computer Science, Norwegian University of Science and Technology. Unpublished, 2021.

[7] Katja Tuma, Gül Calikli, and Riccardo Scandariato. "Threat analysis of software systems: A systematic literature review". In: *Journal of Systems and Software* 144 (2018), pp. 275–294.

[8] Suvda Myagmar, Adam J Lee, and William Yurcik. "Threat modeling as a basis for security requirements". In: *Symposium on requirements engineering for information security (SREIS)*. Vol. 2005. Citeseer. 2005, pp. 1–8.

[9] Adam Shostack. *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

[10] P. Mell and T. Grance. "The NIST Definition of Cloud Computing". In: *National Institute of Standards and Technology, U.S. Department of Commerce, Maryland, MD, USA* (Sept. 2011), SP 800–145. DOI: 10.6028/NIST.SP.800-145.

[11] Barbara Kitchenham et al. "Systematic literature reviews in software engineering–a systematic literature review". In: *Information and software technology* 51.1 (2009), pp. 7–15.

[12] Rabia Latif et al. "Cloud computing risk assessment: a systematic literature review". In: *Future information technology* (2014), pp. 285–295.

[13] Claes Wohlin. "Guidelines for snowballing in systematic literature studies and a replication in software engineering". In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. 2014, pp. 1–10.

[14] Daniele Granata and Massimiliano Rak. "Design and Development of a Technique for the Automation of the Risk Analysis Process in IT Security." In: *CLOSER*. 2021, pp. 87–98.

[15] C Oscar Encina, Eduardo B Fernandez, and A Raúl Monge. "Threat analysis and misuse patterns of federated inter-cloud systems". In: *Proceedings of the 19th European Conference on Pattern Languages of Programs*. ACM. 2014, p. 13.

[16] Kristian Beckers et al. "A pattern-based method for establishing a cloud-specific information security management system". In: *Requirements Engineering* 18.4 (2013), pp. 343–395.

[17] Subrota Kumar Mondal et al. "Kubernetes in IT administration and serverless computing: An empirical study and research challenges". In: *The Journal of Supercomputing* 78.2 (Feb. 2022), pp. 2937–2987. ISSN: 0920-8542. DOI: 10.1007/s11227-021-03982-3.

[18] Wenjun Xiong et al. "Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix". In: *Software and Systems Modeling* 21.1 (Feb. 2022), pp. 157–177. ISSN: 1619-1366. DOI: 10.1007/s10270-021-00898-7.

[19] W. van der Merwe Gilliam van der Merwe C. Muller and D. Blaauw. "Identifying Adversaries' Signatures Using Knowledge Representations of Cyberattack Techniques on Cloud Infrastructure". In: *Proceedings of the 17th International conference on Cyber Warfare and Security*. ICCWS. 2022, pp. 333–339.

[20] Haroon Elahi et al. "On the Characterization and Risk Assessment of AI-Powered Mobile Cloud Applications". In: *Computer Standards & Interfaces* 78 (Oct. 2021), p. 103538. ISSN: 0920-5489. DOI: 10.1016/j.csi.2021.103538.

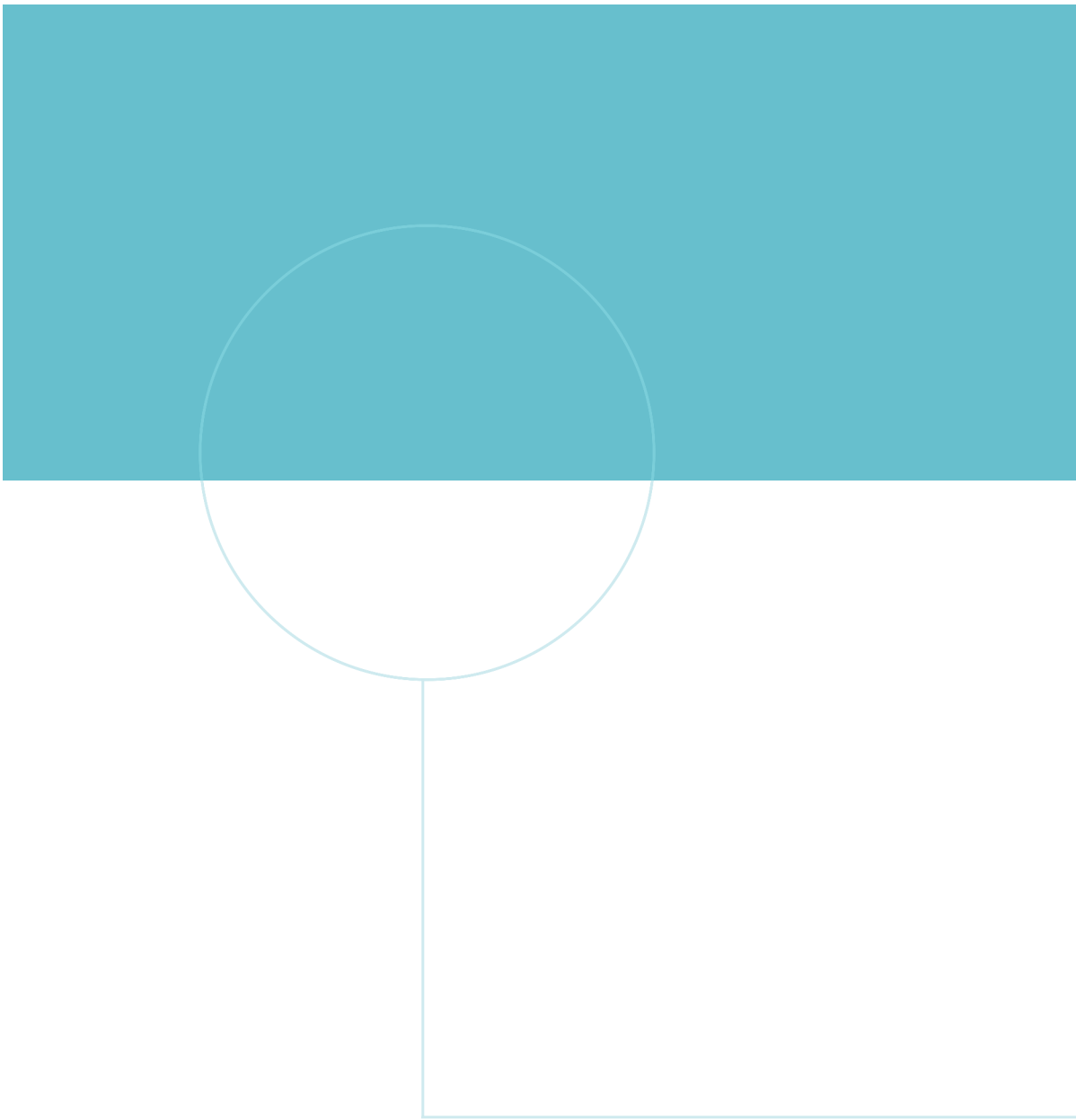
[21] Jin B. Hong et al. “Systematic identification of threats in the cloud: A survey”. In: *COMPUTER NETWORKS* 150 (Feb. 2019), pp. 46–69. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2018.12.009.

[22] Valentina Casola et al. “A novel Security-by-Design methodology: Modeling and assessing security by SLAs with a quantitative approach”. In: *JOURNAL OF SYSTEMS AND SOFTWARE* 163 (May 2020). ISSN: 0164-1212. DOI: 10.1016/j.jss.2020.110537.

[23] Haralambos Mouratidis, Shaun Shei, and Aidan Delaney. “A security requirements modelling language for cloud computing environments”. In: *SOFTWARE AND SYSTEMS MODELING* 19.2 (Mar. 2020), pp. 271–295. ISSN: 1619-1366. DOI: 10.1007/s10270-019-00747-8.

[24] Mouna Jouini, Latifa Ben Arfa Rabai, and Ridha Khedri. “A quantitative assessment of security risks based on a multifaceted classification approach”. In: *INTERNATIONAL JOURNAL OF INFORMATION SECURITY* 20.4 (Aug. 2021), pp. 493–510. ISSN: 1615-5262. DOI: 10.1007/s10207-020-00515-6.

[25] Daniela Soares Cruzes et al. “Challenges and experiences with applying microsoft threat modeling in agile development projects”. In: *2018 25th Australasian Software Engineering Conference (ASWEC)*. IEEE. 2018, pp. 111–120.



NTNU

Norwegian University of
Science and Technology