Leveraging LLMs to Generate Threat Profiles

Shifra Lopulalan

Delft University of Technology

TU Delft

Leveraging LLMs to Generate Threat Profiles

by

Shifra Lopulalan

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday June 29, 2026 at 13:00.

Student number:	4564383	
Project duration:	November 14, 2025 – June 29, 2026	
Thesis committee:	Dr. ir. H. J. Griffioen,	TU Delft, supervisor
	Ir. Y. Song,	TU Delft, advisor
	Dr. M. A. Migut,	TU Delft

TU Delft

Preface

First of all, I would like to thank my supervisor, Harm Griffioen, for his support and guidance throughout this project. I would like to thank him for always supporting me, thinking along with me, and helping me through this finale hurdle before graduation. His encouragement and infectious positivity has been a great source of motivation throughout this journey and played an important role in being able to complete this thesis. I would also like to thank Yuqian Song for his guidance during this thesis. His feedback on my drafts and his tips and tricks during my thesis have been invaluable to me. He was there at the start of my academic journey at TU Delft, so I am grateful for his guidance to help me to the end. I would also like to extend my thanks to Gosia Migut for making time in her busy schedule to be a part of my thesis committee.

I would like to extend a special thank you to Pascal Oldenzeel and Selma Sturmans for allowing me to combine my thesis project with work. I appreciate the trust, support, and flexibility provided throughout this period. Thank you for this opportunity to learn more about the use of threat intelligence and LLMs within a practical context, which contributed to my academic, professional, and even personal development.

Finally, I would like thank my friends, family, and colleagues for their constant support and encouragement throughout this academic journey. Their belief in me and their help in staying motivated have been invaluable and were essential in reaching the end of this journey.

Shifra Lopulalan
Delft, June 2026

Abstract

The heavy reliance on digital infrastructure introduces many risks for organizations. Therefore, it is key to understand which threats are more relevant in a rapidly changing threat landscape. This is especially important for financial institutions, which are attractive targets for cyber attacks and operate under strict regulatory requirements. Threat profiles can help organizations understand who may attack them, why they may be targeted, how attacks may occur, and what the potential consequences could be. However, creating threat profiles is a labor intensive process that requires collecting, analyzing, interpreting, and prioritizing threat information.

This thesis investigates how Large Language Models (LLMs) can support the development of threat profiles for a financial institution, and how the threat profiles can be validated. Existing research has explored threat profiling, threat intelligence, LLMs, and prompt engineering separately, and recent work has shown that threat intelligence can enrich threat modeling. However, less attention has been given to how structured threat intelligence can be used by LLMs to generate threat profiles for organizations.

To address this gap, a threat-centric profiling framework was developed based on existing threat modeling frameworks. The framework consists of four components: Threat Actor, Motivation & Intent , Threat Events, and Consequences. This framework was used to guide an iterative prompt engineering process and to evaluate generated threat profiles using a structured evaluation rubric. A custom GPT was also developed and enriched with filtered and normalized threat intelligence from MISP. The generated profiles were validated through a technical validation using aggregated security monitoring data and an expert based validation with three cybersecurity experts.

The results show that LLMs can support several steps in the threat profiling process, including analyzing information, mapping it to a framework, prioritizing threats, and generating coherent threat profiles. Prompt design had the strongest effect on the quality of the generated profiles, while threat intelligence made the profiles more concrete and actionable. The validation showed that technical monitoring data alone is not sufficient to validate all threat profile components, as claims about threat actors, motivations, and consequences often require additional evidence or expert interpretation. Overall, LLMs can reduce manual effort and make threat profiling more manageable, but the generated profiles must still be reviewed and validated by analysts before being used for decision-making.

Contents

1 Introduction

Organizations increasingly rely on digital infrastructure to support their daily operations. While this enables efficiency and connectivity, it also introduces many risks. Since it is impossible for organizations to protect against all possible threats, they need a clear understanding which threats are most relevant to them. Since the threat landscape changes rapidly, it is important that organizations stay aware of the most prominent threats they may face. This is particularly relevant for the financial sector, where organizations are not only active targets for cyber attacks, but are also subject to strict supervision and regulatory requirements. Threat profiles can support these organizations by identifying and structuring the most relevant threats. They help organizations understand who may target them, why they may be targeted, how attacks occur, and what the possible consequences could be.

However, creating a threat profile is a labor intensive process that requires collecting, analyzing, interpreting, and prioritizing threat information. It also requires expert knowledge to correctly map the threats to the organization and to determine which threats are relevant. The rapid development of Large Language Models (LLMs) introduces opportunities to support analytical tasks and to reduce manual effort for certain tasks. Since LLMs are capable of processing and structuring large amounts of information, their application looks promising for the threat profiling process.

Existing literature shows promising results for the use of LLMs in threat modeling, which is relevant because threat modeling can be seen as the technical application of a threat profile by translating the threats into risks for an organization's systems or environments. Previous studies have shown that LLMs can support threat modeling by structuring information, identifying possible threats, and help analysts reason about attack scenarios [39, 40, 8]. This suggests that LLMs may also be useful in the earlier and broader threat profiling step.

Additionally, research shows that threat intelligence can improve threat modeling by providing more recent and contextual information about threats. By integrating threat intelligence, threat models can become less static and better reflect the rapidly changing threat landscape [28, 9]. However, less attention has been given to how threat intelligence can be used to support the creation of threat profiles. Since threat profiles provide the contextual foundation for threat models, this research investigates how LLMs can use threat intelligence to support the threat profiling process.

In this thesis, a threat profiling framework was created based on existing threat modeling frameworks. This framework was then used as the basis for the prompt engineering process and the evaluation of the generated outputs. The prompt engineering process was an iterative cycle that adjusted each prompt accordingly after the profile was evaluated. This evaluation rubric assesses whether the threat profiles were complete and cohesive, and whether the components were clearly described and mapped to each other. Based on the evaluation results, the prompt was adjusted to improve the quality of the generated threat profiles. Next, a custom GPT was created to include threat intelligence in the threat profiles. This was done to provide the model with domain specific information and to generate more concrete threat profiles. Finally, the generated threat profiles were validated in two ways. A technical validation was performed using security data from an organization to assess whether the profiles reflected real-world

activity. And an expert based validation was conducted with three experts, who evaluated the generated threat profiles using the same evaluation rubric to assess their quality and usefulness.

The primary objective of this research was to answer the following research question: *How can LLMs be used to support the development of threat profiles for a financial institution, and how can the generated profiles be validated?*". To address this research question, several sub-questions have been generated:

1. What input data and prompts are required for an LLM to generate a threat profile?
2. Which steps in the threat profiling process can be automated or supported using an LLM?
3. How can the generated threat profiles be validated to assess their quality and reliability?

This thesis shows that LLMs can partially automate the threat profiling process. When the LLM is provided with explicit instructions, a threat profiling framework, and relevant input data, it can generate a coherent threat profile and automate several steps on the process which reduces the manual effort required to create threat profiles. However, the generated profile should still be reviewed and validated by an expert before it is used. Meaning that LLMs should be used as an supportive tool for threat profiling rather than a replacement for cyber security analysts.

The main contributions of this thesis are as follows:

- Provides a structured approach for investigating how LLMs can be applied to threat profiling.
- Provides insight into the role of prompt design and threat intelligence in the quality of threat profiles.
- Presents a validation approach for generated threat profiles by combining technical validation with expert based validation.

The remainder of this thesis is structured as follows. Chapter 2 introduces background concepts to understand this research. Chapter 3 provides an overview of related work on threat profiling, threat intelligence, LLMs in cyber security, and prompt engineering. Chapter 4 describes the research methodology. Chapter 5 presents the results of the prompt engineering experiments. Chapter 6 presents the outcome of the technical and expert based validation. Chapter 7 discusses the findings, limitations, recommendations, and reflects on the research. Chapter 8 concludes the thesis. And finally, Chapter 9 acknowledges how AI has been used in this thesis.

2 Background

This thesis investigates how LLMs can be used to support the generation of threat profiles. To provide a proper understanding of the following chapters, this chapter introduces the foundational concepts used in this research. First, threat profiling will be explained, including its purpose, the process to generate a threat profile, and important concepts relevant to threat profiling. Next, threat intelligence is introduced as an information source that helps understand threats better. Finally, large language models and prompt engineering are explained, as these are the foundation when using an LLM to generate threat profiles.

2.1. Threat Profiling

A threat profile is a structured representation of potential threats targeting systems, networks, or organizations [38]. Typically, it contains information about relevant threat actors, their motivation, and objectives. However, there is no uniformly accepted standard for threat profiling, which leads to variations in structure, methodology and outcomes between profiles. These differences are not only due to the context of organizations, but also because of the framework and approaches used to develop the profile. Despite the lack of standard for threat profiling, there exist several established frameworks, such as MITRE ATT&CK [37], STRIDE [17], and NIST SP 800-30 [24]. These frameworks provide a structured approach to identify, analyze and categorize potential threats. For example, by offering standardized threat categories (e.g. tactics and techniques in MITRE ATT&CK [37] or spoofing and tempering in STRIDE [17]), and therefore creating a consistent vocabulary for threat profiling. Additionally, they often describe how threats can be systematically identified and analyzed, supporting a more replicable threat profiling process.

Threat profiles can be developed using different approaches, depending on the focus of the analysis or the intended application. Three main approaches are commonly used: threat-centric, asset-centric, and system-centric [5]. Threat-centric profiles focus on understanding the potential threats first by identifying relevant threat actors, their motivations, capabilities, and possible threat events. In contrast, asset-centric profiles start by identifying and prioritizing assets that could be affected by threats, after which potential threats are characterized and mapped to the underlying systems that support or store these assets. Finally, system-centric profiles take a more technical approach. They focus on the components and potential vulnerabilities of systems, and how they can be exploited by threats. Each approach provides a different perspective on potential threats and influences how they are identified and prioritized.

In practice, the value of threat profiling becomes particularly evident when considering the constraints organizations face. Due to limited time and resources, it is impossible for organizations to defend against all possible threats. Therefore, it is essential for them to identify and prioritize the most relevant threats and determine how to defend against them. Threat profiles support this process by providing an overview of active threats and their potential impact on an organization. They can support defense strategies, enable more effective risk assessments, and guide the selection of appropriate risk mitigation measures. Overall, threat profiles act as guide rails that help organizations prioritize threats and make informed security decisions.

2.1.1. Threat Landscape, Threat Profile, and Threat Model

It is important to understand a few key concepts to develop a clearer understanding of threat profiling and its place within the broader threat landscape and research context. This subsection clarifies what threat landscapes and threat models are, as well as how they relate to threat profiling. The relationship between these concepts is visually presented in Figure 2.1.

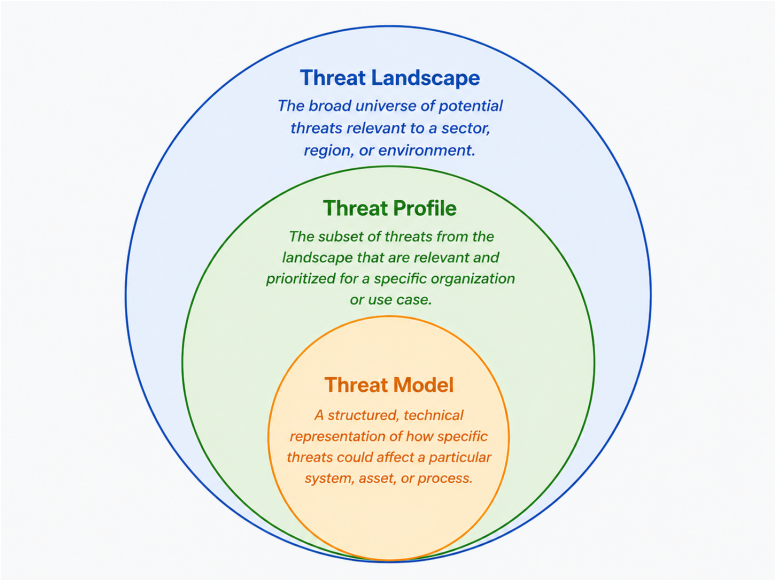

Figure 2.1: A visual representation of the relationship between threat landscapes, threat profiles and threat models.

A threat landscape provides a broad overview of all observed potential threats. Its scope is often wide and can be industry focused, region specific, or even global. Understanding which threats are active within such a broad scope helps create a general understanding of where individuals or organizations may be at risk. This can support organizations in preparing for potential risks, although, due to the broad scope, not all threats might be relevant to them. Threat models focus on identifying and analyzing threats to specific systems. This focus is more technical than that for threat landscapes and profiles, as it investigates what could potentially go wrong within a system when it is exposed to a threat. It can be seen as the technical application of a threat profile.

Threat profiles can be seen as a subset of a threat landscape, tailored to a specific context, such as an organization within the sector or industry covered by the threat landscape. They filter out the irrelevant information and leave only what is applicable to the specific context, essentially reducing noise of the threat landscape. Threat profiles are used to answer the question: who is trying to attack us, and how? They form the bridge between a threat landscape, which provides a very broad overview of threats, and a threat model, which zooms in on the technical aspects of those threats.

Threat profiles provide input for threat modeling by creating a contextual understanding of who might be trying to attack and how. Threat models then make this understanding tangible by their technical application to the respective environment. They provide insight into which systems are at risk, as well as possible attack paths within the system. Both threat profiling and threat modeling aim to identify and understand risks in order to support risk assessments and improve defensive strategies. Due to this

overlap, threat modeling frameworks are often reused for threat profiling. And due to existing threat profiling approaches mainly consisting of practical guides, templates, and vendor-specific methods, rather than standardized frameworks.

2.1.2. The Threat Profiling Process

The creation of a threat profile is a structured process in which threat data is collected, analyzed, and transformed into an overview of relevant threats. A visual representation of this process can be found in Figure 2.2. Threat data can be collected from different sources such as security reports, network traffic logs, and open-source threat intelligence. These sources provide information about malicious activity, known threat actors, possible attack methods, indicators of compromise, and possible impacts [15].

Once the data has been collected, it needs to be analyzed to identify the most relevant threat actors, their motivations, and the attack methods they use. Next, the analyzed data needs to be mapped to a threat profiling framework. For example, observations from network logs need to be mapped to the relevant threat event, and the actors need to be linked to the corresponding threat events that were spotted. Then, the motivation of the threat actor can be inferred based on the type of actor, their attack method and their targets. By analyzing the potential impact of a threat event, the consequences of the event can be determined [15]. These analyses require combining information from multiple sources. Expert analysis is often needed to determine whether the information is relevant and how it can be used, since the threat data can differ significantly. When there is a clear overview of the threats, they need to be prioritized. This can be done by considering the likelihood of a threat event and the potential impact it might have. Prioritization helps organizations focus on the most relevant risks and supports decision making for defensive strategies and risk assessments.

The threat profiling process requires a significant amount of manual analysis, interpretation, and correlation of data from different sources. Since the threat landscape evolves rapidly, this process needs to be repeated regularly to remain up to date on the latest threats[15]. As a result, threat profiling is not only labor intensive, but also requires the right expertise to interpret threat information correctly and translate it into a useful profile.

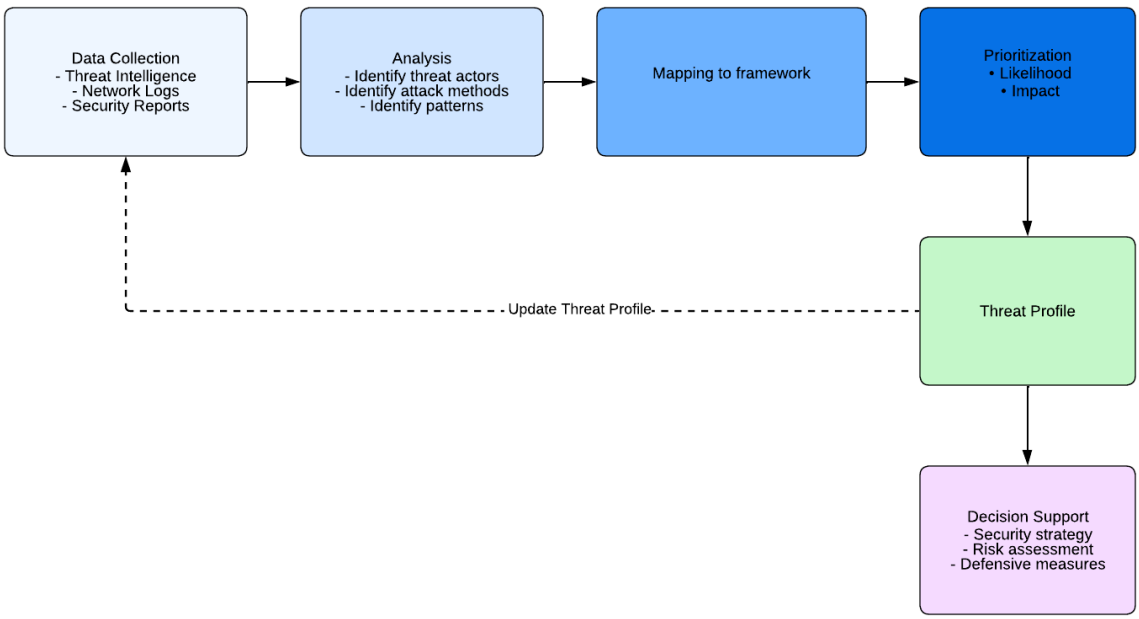

Figure 2.2: A visual representation of the threat profiling process. To create a threat profile, data needs to be collected, analyzed, interpreted, and prioritized. Once a threat profile is created, it can be used for decision making or to update it once the threat landscape has evolved.

2.2. Threat Intelligence

Threat intelligence is the collection, processing, and analysis of threat data to understand threat actor's motives, targets, and attack methods [4]. Its purpose is to transform raw data into actionable insights that support threat detection, risk assessments, and defensive strategies. By using threat intelligence, organizations can shift from a reactive security posture towards a more proactive one. This is only possible when threat data is interpreted into the organization's context and translated into information that can support concrete security measures.

Raw threat data can originate from many different sources, such as firewalls, honeypots, log files, and scanners. On its own, this data has limited value, as it often consists of isolated observations without any further explanation. The value increases when this raw threat data is processed, analyzed, and interpreted, allowing individual observations to be placed within a broad threat landscape. This processed data, also known as actionable threat intelligence, provides information about active attack campaigns, adversary behaviors, or what malicious IP addresses and domains are being used by adversaries. Actionable threat intelligence can be used to support threat detection, alerting, and finding correlations. Being able to correlate malicious activities to certain types of events and threat actors will be useful to support claims in threat profiles.

2.2.1. Types and Sources of Threat Intelligence

Threat Intelligence has many different applications and can support security operations and different levels within an organization. There are three types of threat intelligence: strategic, operational, and tactical threat intelligence. Strategic threat intelligence provides an high-level view of the threat landscape. This type of intelligence is often used for risk assessments and to determine business implications of cyber threats. Operational threat intelligence explains how and why a threat is occurring. It provides contextual information about attack vectors, adversary behavior, and ongoing campaigns, helping organizations understand how a threat may affect their systems and business operations. Tactical threat intelligence supports threat detection and support. This type of intelligence focuses on identifying indicators of compromise and often contains detailed information about attacker's tools like the malware they use or vulnerabilities they exploit. Therefore, tactical threat intelligence is useful for developing and improving defensive mechanisms.

Threat intelligence can be collected from different sources, which can be divided into internal and external sources. Internal sources are generated by the organization itself, for example through system logs, firewall logs, threat alerts or incident reports. External sources originate outside the organization. Examples include open-source threat intelligence, commercial threat feeds, honeypots, and security advisories. Since no single source can capture all possible threat data, organizations often use a combination of internal and external sources to obtain a broad understanding of the potential threats targeting them.

2.2.2. Operational Use of Threat Intelligence

Threat intelligence truly becomes valuable when it is operationalized, it can be embedded into security workflows to automate actions and enrich data to assist decision making. Which often means integrating threat intelligence into existing security tools and supporting information sharing between organizations.

A common implementation of threat intelligence is within Security Information and Event Management Systems (SIEM). Here, incoming logs from infrastructure, for example, network traffic or system logs are analyzed against threat intelligence feeds through correlation rules, generating alerts when indicators, like IP addresses or file hashes, match malicious activity [18]. This implementation helps threat detection and makes incident investigations more effective. Another example is the integration of threat intelligence feeds into firewalls, where known malicious indicators can be used to automatically block suspicious traffic. This reduces exposure to known threats by proactively taking action before more harm can be done.

Threat intelligence is also shared between organizations through closed communities, allowing the community to benefit from observations made by individual members. Platforms such as Malware Information Sharing Platform (MISP) [22] support this process by facilitating the collection, storage, and correlation of threat data in a structured manner. Within MISP, information is organized into *events*, which can be understood as case files that group together all relevant data related to a specific threat event and will contain Indicators of Compromise (IoCs), contextual information, and associated metadata. The individual data points within an event, like the IoCs, are referred to as *attributes*. *Attributes* are stored in a structured format that allows for correlation and linkage to known threat actors or campaigns within the platform. Additionally, MISP also includes *galaxies*, which are structured knowledge bases that link events to broader concepts, such as threat actor groups, malware types, or techniques defined in frameworks like MITRE ATT&CK [37]. In this way, MISP supports the transformation of raw data into structured and contextualized threat intelligence.

2.2.3. Challenges and Relevance to This Research

Although threat intelligence can be very useful for organizations, there are several challenges that limit its use. One of the most frequent challenge is data overload. Generally, threat intelligence platforms and feeds contain large amounts of indicators, events, and metadata. While this gives an extensive view of the threat landscape, it makes it difficult to determine which information is relevant for an organization. Another challenge is the lack of context, threat intelligence feeds often provide lists of isolated indicators, like malicious IP addresses or domains. When they are used for analysis, these indicators provide limited value as they do not explain how they relate to a threat actor or attack pattern. As discussed earlier, the value of threat intelligence increases when it can be interpreted within a broader context. Without this, it remains difficult to use the threat intelligence for threat profiling or decision making. Additionally, threat intelligence may contain false positives or noise. Some indicators may no longer be malicious or only be applicable in specific situations, which is closely related to the challenge of timeliness. The threat landscape changes continuously, so indicators can quickly become outdated.

These challenges are especially relevant for organizations, since it is important to focus on the most relevant threats and to use available resources effectively. For this reason, this research focuses on how threat intelligence can be structured and used in a way that supports the creation of threat profiles. In this thesis, MISP is used to collect threat intelligence, because it provides threat data in a structured format. The custom GPT will process and interpret the threat intelligence from MISP, and use it to enrich the generated threat profiles with relevant and structured information.

2.3. Large Language Models

Large Language Models (LLMs) are deep learning models designed to understand and work with human language [19]. The models are trained on massive amounts of data using machine learning techniques. By training to predict the next item of a sequence, the LLM learns how to generate text. Which is why they are so good at generating and analyzing text.

The ability to understand and learn sequences is accomplished with the use of transformer architecture. This architecture is a type of neural network architecture that can handle sequential data effectively by looking at the relationship between sequence components [2]. Due to parallel computation, transformers can process long sequences without a significant increase in training and processing time. They are also easily scalable to train on and handle more data, which makes them perfect for LLMs.

Within cybersecurity, LLMs provide value through their ability to analyze large-scale unstructured data sources. Their pattern recognition and reasoning capabilities can help support threat detection and incident analysis. Because of their ability to generate explanations of cyber security matters, they can support security teams in understanding their organization's security risks better.

Although LLMs provide many advantages, it is equally important to understand their limitations so that appropriate measures can be taken during their use. The most frequently discussed limitation is hallucinations [14]. This occurs when the LLMs perceives nonexistent patterns and uses these to generate an output. As a result, this output will be highly inaccurate or does not exist. A further limitation is the sensitivity to the input quality, small changes in the input can affect the model's output, and might result in unreliable results.

There are several LLMs available for reasoning and analytical tasks, such as ChatGPT [26], Claude [3], and Gemini [10]. In this research, the LLM used is ChatGPT [26], developed by OpenAI. ChatGPT is a conversational artificial intelligence system, allowing users to interact with an LLM in a conversational way. Instead of requiring technical input or predefined commands, users can ask questions, provide instructions, or give context in written form. Based on this input, ChatGPT generates responses by predicting and producing relevant text. An advantage of ChatGPT is the possibility to create a custom GPT. A custom GPT is a version of ChatGPT that can be configured for a specific purpose by adding instructions, context, files, or additional actions through API connections. This allows the model to be tailored to a specific task or workflow, making its responses more consistent and relevant to the intended use. In this thesis, the custom GPT can be guided by the research context, terminology, and desired output structure.

In summary, LLMs provide strong language processing and reasoning capabilities but they remain sensitive to input formulation and inaccurate outputs. Their effectiveness therefore heavily relies on the quality of the input. Thus making prompt engineering a crucial aspect to leverage the full powers of LLMs.

2.4. Prompt Engineering

To ensure that the LLM gives the desired output, it is important to present it with the correct prompt. To achieve this, prompt engineering is necessary. This is the process of creating the input prompt to generate a specific and high quality output. A prompt will guide the behavior of the LLM, but it will not train the LLM. Prompt engineering is important because the LLM relies entirely on the input prompts. Therefore, the quality of the input will determine the quality of the generated output. Ambiguous prompts will lead to incorrect responses, while well defined prompts will result more accurate and precise outputs.

2.4.1. Prompting foundation

According to Shenoy et al. [36], the three fundamental rules of prompt engineering are as follows: Keep it simple, be specific, be concise. The request should be simple and well defined, if a task needs to be executed, it has to be specific and explicit. Finally, to prevent the LLM from diverging from the topic, ensure the prompt is concise. While keeping these fundamental rules in mind, a prompt should consist of four elements: context, instruction, input text, and output pointer. First of all, the context should be given so the LLM knows the necessary background information. Then, the instructions should be given so the LLM knows what it is expected to do or what actions should be taken. Next, the input text gives the LLM instructions on what sources it should use. And finally, the output pointer informs the LLM about the preferred style of the output.

2.4.2. Prompting Techniques and training methods

Once a prompt has been engineered using these principles, several prompting techniques can be applied to control how prompts are executed and how their outputs are reused. The simplest and most direct technique is Single Stage Prompting, in which a single prompt is sent to the LLM and a single output is returned. However, if the output is used as input for the next prompt, for example by asking a follow-up question about the output, the technique transitions to Multi-Stage Prompting. This technique is useful for iterative reasoning, but when there are too many stages, it can lead to error propagation. Another technique that can be used is Batch Prompting, in which a batch of prompts is sent to the LLM at once, rather than iteratively as in Multi-Stage Prompting.

Besides prompting techniques, there are also prompt-based training methods. These methods guide the LLM toward preferred behavior by either relying solely on its pre-trained knowledge or by including examples of the preferred output. When no examples are given, and the LLM relies entirely on its pre-trained knowledge, this approach is also known as zero-shot learning. Providing one or more examples leads to one-shot or few-shot learning, respectively.

Unfortunately, there are cases where no examples are available to include in the prompt, or the pre-trained knowledge of the LLM is not sufficient. When additional information is required, Retrieval-Augmented Generation (RAG) can be used to include external sources, like databases, APIs or company knowledge. By applying RAG, the pre-trained knowledge of the LLM is combined with external knowledge sources. This approach is particularly useful for domain-specific tasks, as information related to these tasks is often not publicly available. It is important to note that the model will not be trained with this external data, it is only used during output generation. Unlike few-shot learning, where examples are embedded directly into the prompt, RAG dynamically retrieves the information without increasing the number of in-prompt examples.

2.4.3. Prompting Quality

On top of selecting the appropriate prompting training methods and techniques, the quality of a prompt plays a crucial role in the effectiveness of LLM outputs. The prompt quality can be evaluated by looking at the consistency of the generated output, the accuracy of the task execution, and the robustness to small changes in wording. In other words, a high quality prompt should produce a stable output across multiple executions, correctly followed the instructions, and remain operative when minor changes in formulating the prompt are introduced.

As the previous paragraphs have shown, several factors influence the quality of an LLM's output. An appropriate prompting technique must be selected in combination with the correct prompt-based training method, and the prompt itself must be of high quality to avoid inconsistencies. For domain-specific tasks, additional guidance is often required, which can be provided through the use of Retrieval-Augmented Generation (RAG).

3

Related Work

This chapter reviews existing research on threat profiling and threat modeling, threat intelligence, the application of Large Language Models in cybersecurity, and prompt engineering for cybersecurity. The first section discusses literature on threat modeling, focusing on existing frameworks, the threat modeling process, and the automation of threat modeling. Next, the practical value of threat intelligence is examined, along with the challenges associated with its use. This is followed by a discussion of the use of LLMs in cybersecurity and an analysis of prompt engineering research within the cybersecurity domain. Based on this literature, the research gap is then presented. Topics that require further investigation are identified and used to motivate the research question of this thesis.

3.1. Threat Profiling

The existing research on threat profiling covers several aspects, including threat modeling frameworks, the threat profile process, and attempts to automate parts of the threat profiling process. Combining these aspects will show that threat profiling is a structured activity, and existing approaches differ substantially in scope, level of detail, and automation.

There are many different threat modeling frameworks available, each with its own terminology, structure, and purpose. Bodeau et al. [5] compare multiple threat modeling frameworks and develop a new framework by combining elements from existing approaches. Their work highlights that threat modeling lacks standardization, as frameworks differ in terminology, structure, and purpose. Similarly, Khalil et al. [16] show that threat modeling benefits from structured frameworks and emphasize that selecting an appropriate framework is an important part of the threat modeling and profiling process. Together, these studies show that threat modeling is a structured process, but that there is no universally accepted standard. However, both studies remain limited in scope for this research. Bodeau et al. [5] stay at a general framework level and do not provide organization-specific or automated methods for generating threat profiles, while Khalil et al. [16] focus specifically on industrial control systems. These studies are relevant to this research because they support the creation of a threat profiling framework by identifying and combining elements from existing frameworks, while also emphasizing the importance of a structured foundation for generating threat profiles.

Once a framework has been selected, the next step is to create the threat profile. However, existing research shows that this process is often labor intensive as threat data needs to be collected, interpreted, and organized into a useful threat profile. Xiong and Lagerström [41] analyzed 54 threat modeling articles and concluded that most threat modeling work remains to be done manually, which is a time consuming and error prone process. They also discovered that there was a lack of validation, and when it did occur it was through theoretical examples, or empirical case studies. Möller [23] shows that threat attack modeling requires identifying and structuring multiple elements of an attack, including the adversary, their motivation, resources, expertise, attack patterns, incidents, and possible attack paths.

This illustrates that building a meaningful threat model is a detailed process that goes beyond simply listing threats. While Rodríguez et al. [30] propose a method for attacker profiling using the MITRE ATT&CK framework to gain a better understanding of attacker behavior by connecting attacker tactics and techniques to each other. Applying this method requires collecting and centralizing attack events, labeling the events with MITRE ATT&CK tags, manually correcting missing labels when automated rules leave them blank, then organizing them into cases, constructing a process-mining event log, and, finally, discovering and evaluating an attack process model. This shows that even semi-automated threat profiling remains a labor-intensive process, because significant preparation, filtering, labeling, and expert interpretation are still required before meaningful profiles can be generated. These papers show that threat profiling is a labor intensive job, it requires extensive data collection and thorough analysis to gain an understanding of a threat actor or event. Since it is a labor intensive process, this work investigates the use of LLMs/ChatGPT to make threat modeling more accessible.

Sharma et al. [35] present a semi-automated method for threat agent profiling using network traffic data combined with threat intelligence to support the threat assessment. Their work shows that threat profiling can be supported by automation and does not need to be performed entirely manually. They also show that network traffic data and threat intelligence can be used together to enrich the analysis and provide more context to the profiles. Shahid et al. [34] propose an unsupervised clustering approach for profiling threat actors based contextual cyber attack information from cyber attack reports, and show that profiling can be supported without labeled data when contextual information is available. While both papers focus on different types of data, they both show what kind of data can be used to support the automation of threat profiling and how this can be done. Neither paper focuses on threat profiling for organizations. Instead, Sharma et al. [35] focus on profiles for threat agents, and Shahid et al. [34] focus on a global threat actor profiling. Regardless of that, these papers are relevant because they show that threat profiles can be enriched with threat intelligence or a knowledge base, and that profiling process can be partially automated.

More recent studies have investigated the integration of threat intelligence into the threat modeling process. Podlesnik et al. [28] investigate how threat intelligence and threat modeling can be integrated to improve cyber resilience compared to stand alone implementations. Their analysis shows that there is a strong preference in practice for combining threat intelligence and threat modeling, but that there is a lack of standards to support this integration. To address this, Podlesnik et al. [28] propose an evaluation framework that can be used to assess the integration of threat intelligence and threat modeling. Erbas et al. [9] demonstrate how threat intelligence from MISP can be integrated with a threat modeling framework to analyze threats and simulate attacks. They argue that threat modeling frameworks are often static and do not adapt to newly emerging threats. By integrating threat intelligence, the threat model becomes more dynamic as it enabled real-time threat intelligence updates and demonstrate that is strengthens structured threat analysis. Altogether, these studies show that threat intelligence can enrich threat analysis by making threat models more responsive to changes in the threat landscape. They demonstrate the value of combining threat intelligence with structured threat analysis.

3.2. Threat Intelligence

Threat intelligence plays an important role in cybersecurity by helping organizations better understand relevant threats and vulnerabilities. Existing research shows both the practical value of threat intelligence and the challenges that must be addressed before it can be applied effectively within an organization.

Threat intelligence helps to improve cybersecurity of organizations by providing actionable information about threats and vulnerabilities. Saaed et al. [31] show that threat intelligence contributes to resiliency of organizations because it helps them understand the risks, anticipate threats, and improve preparedness. Abu et al. [32] highlight that threat intelligence supports proactive defense strategies by helping organizations identify new threats, weaknesses, and attack trends. Ahmad and Haripriya [1] examine how threat intelligence improves cybersecurity posture by providing actionable insights, integration into security planning, and support threat identification and response processes. Research shows that threat intelligence can be used to identify and understand threats better, which is a key component in threat profiling. Therefore, threat intelligence would be a useful data source when creating a threat profile.

While threat intelligence can be useful for organizations, there are some challenges that need to be addressed before threat intelligence can become actionable intelligence. In 2018, Abu et al. [32] noted that threat intelligence needs to be processed, analyzed, and requires context before it becomes actionable. Due to the amount of data and the inconsistent data types, this process would be quite time consuming, and requires skilled analysts to perform this task. Which made the application of threat intelligence in organizations a labor intensive task. In 2024, Ahmad and Haripriya [1] address this issue as well. They highlight data quality issues, data overload, and the need for automation as challenges. Due to the rapid developments in AI, they suggest utilizing AI to support analysts and help with the processing of threat intelligence. These challenges highlight the need for thorough filtering to avoid data overload, and support the use of AI to analyze and apply the threat intelligence.

Threat intelligence sharing platforms were introduced to tackle the previously mentioned challenges. Abu et al. [32] show that threat intelligence sharing platforms are important for making threat intelligence more usable, as these platforms support the sharing and exchange of threat information between organizations. Threat intelligence sharing platforms can help address challenges such as information overload and fragmented threat data, as these platforms often correlate threat events and add context, making it easier to analyze the threat intelligence. More specifically, Saeed et al. [31] show how a threat intelligence sharing platform can help transform large amounts of data into actionable threat intelligence. They discuss the INTIME framework, in which MISP is used as an integrated platform for storing and sharing threat related information across organizations. Together, these studies show that threat intelligence sharing platforms increase the practical value of threat intelligence by supporting collection, sharing, and structuring of threat data.

3.3. Large Language Models in Cybersecurity

In recent years, Large Language Models (LLMs) have gained attention for their potential use in cybersecurity. Existing work shows that LLMs can assist with tasks such as threat detection, explanation generation, or threat modeling. Their ability to process technical information and present it in a explainable way, makes them useful for analytical tasks. While the use of LLMs in cyber security looks promising, it also introduces several challenges such as hallucinations, bias, and dependence on domain specific context. This section discusses research that demonstrates the potential of LLMs in cybersecurity, while also highlighting the limitations associated with their use.

First, this section explores the use of LLMs in detection oriented cybersecurity, where they are used to identify attacks and explain security events. This is relevant to threat profiling, as it demonstrates the potential of LLMs to structure and explain technical threat information. Houssel et al. [12] investigate the performance of LLMs for Network Intrusion Detection. Their work shows that LLMs struggle to effectively detect threats, but that LLMs do provide clear explanations, making them a useful complementary tool in Network Intrusion Detection. The use of LLMs for DDoS attack detection has been investigated by Guastalla et al. [11]. They discovered that LLMs can detect DDoS attack with high accuracy, when given the right training and context. Together, these studies show that LLMs have potential for threat detection tasks, but also that their performance depends heavily on suitable training methods, relevant data, and sufficient domain specific context. Which is important for the generation of threat profiles, as it requires the LLM to interpret domain specific information and transform it into clear, structured descriptions.

The application of LLMs on threat modeling mainly focus on how LLMs can support, or automate, parts of the threat modeling process. Wimbauer et al. [39] demonstrate that LLMs can support threat modeling in a structured way when using explicit security frameworks and formal representations such as attack graphs. Similarly, Wu et al. [40] propose a domain specific framework for automated threat modeling in banking systems, and improve the threat modeling process by combining prompt optimization and fine-tuning. Elsharef et al. [8] develop an LLM to answer threat modeling questions, using Retrieval Augmented Generation (RAG) to incorporate design documents and vulnerability information. They demonstrate that the LLM can reduce manual effort, and that the use of RAG improves response quality. However, they do note important weaknesses such as hallucinations, bias, privacy risks, and the continued need for expert review. Together, these studies show that LLMs can support threat profiling through structured representations, prompt engineering, fine-tuning, and including external knowledge.

3.4. Prompt Engineering for Cybersecurity

The performance of LLMs depends strongly on how tasks are framed and contextualized, therefore prompt engineering is an important aspect of using LLMs reliably in cybersecurity.

Shenoy and Mbaziira [36] provide an in depth review of prompt engineering in the cybersecurity domain. They offer examples of prompts for cybersecurity tasks, and explain how a prompt should be structured. They show the importance of prompt design for LLM tasks, provide guidance on the prompt structure, and what elements are required to develop an effective prompt. Similarly, Huang et al. [13] provide a guide to prompt engineering techniques for cybersecurity operations where they explain the different prompt training techniques, and highlight the risks and biases that can occur in prompt engineering. According to Huang et al. [13] responsible prompt engineering is a continuous process that requires evaluation and refinement to leverage the most of the LLMs. Priescu et al. [29] focus on structured, scenario-based prompts within cybersecurity workflows. Their work discusses how prompt engineering can support the automation of cybersecurity tasks and highlights the importance of well structured prompts to achieve this. Altogether, these papers demonstrate the importance of prompt design, an iterative prompt engineering process, and the use of external knowledge when performing domain specific tasks.

3.5. Research Gap

The literature review shows that threat profiling, threat intelligence, large language models, and prompt engineering have all been studied within the cybersecurity domain. Research on threat profiling and threat modeling shows that numerous frameworks are available to support the identification and analysis of threats. However, there is no single standard for threat profiling, nor uniformly used terminology, and applying these frameworks is often a labor-intensive process. Studies on attacker profiling show that partial automation can support the analysis of attacker behavior. Research on threat intelligence shows that it can help organizations better understand relevant threats and support defensive security decisions. Recent work demonstrates that threat intelligence can be integrated into threat modeling, for example by using platforms such as MISP. Finally, studies on large language models and prompt engineering show that LLMs can support cybersecurity analysis and threat modeling, but that their performance depends strongly on structured prompts, domain-specific context, and the integration of external knowledge sources. Table 3.1 summarizes the papers discussed in this chapter, including their respective section and contribution to this research.

This literature review shows that there is a research gap at the intersection of threat profiling, threat intelligence, and large language models. These areas have all been explored separately, and recent studies have started to investigate the use of threat intelligence in threat modeling. However, there has been less of a focus on how threat intelligence can support the creation of threat profiles. Since threat profiling is an earlier and broader step than threat modeling, threat modeling research cannot be directly applied to the threat profiling processes. While the studies highlight the challenges of the use of threat intelligence, there is limited research available on how threat intelligence can be filtered and mapped etc. to overcome the challenges. Research on the use of large language models in cybersecurity shows that LLMs perform well in supporting analytical tasks and can reduce manual effort, especially when

prompts are structured, external knowledge is provided, and sufficient examples are given. However, current work mainly focuses on specific systems, attack types, or threat modeling tasks and does not address how an LLM can use structured threat intelligence to generate threat profiles for organizations.

This research addresses the aforementioned gaps by investigating how structured threat intelligence can be operationalized through a Large Language Model to support the generation of threat profiles for an organization within the financial sector. The steps taken to filter and process the threat intelligence are described, as well as how the LLM maps the threat intelligence to the different threat profiling components. This provides a clear development process for generating threat profiles.

Authors	Year	Category	Contributions / Findings
Houssel et al.	2024	Large Language Models	LLMs can interpret and explain network data, improving understanding of security events.
Guastalla et al.	2023	Large Language Models	LLMs can process threat intelligence and detect attacks like DDoS.
Wimbauer et al.	2025	Large Language Models	LLMs combined with structured security frameworks and attack graphs can generate threat hypotheses and quantify risk in Kubernetes environments.
Wu et al.	2025	Large Language Models	LLM-based threat modeling can be operationalized via dataset creation, prompt engineering, and fine-tuning.
Elsharef et al.	2024	Large Language Models	NLP, open-source LLMs, and RAG help answer threat-modeling questions from product documentation.
Shenoy and Mbaziira	2024	Prompt Engineering	Prompt engineering is key for adapting LLMs to cybersecurity tasks and analyst needs.
Huang et al.	2024	Prompt Engineering	Prompt design strongly impacts LLM performance; supports RAG and few-shot prompting.
Priescu et al.	2025	Prompt Engineering	Structured prompting improves detection, automation, CTI, and incident response.
Saeed et al.	2023	Threat Intelligence	Threat intelligence improves resilience by helping organizations understand risks and prepare for threats.
Ahmad and Haripriya	2024	Threat Intelligence	Provides actionable insights for identifying, preventing, and responding to threats.
Abu et al.	2018	Threat Intelligence	Threat intelligence is limited by ambiguity, practical challenges, and reliance on skilled analysts.
Bodeau et al.	2018	Threat Profiling	Structured approaches for identifying threats, assets, and vulnerabilities.
Sharma et al.	2021	Threat Profiling	Categorizes threat actors by behavior, capabilities, and intent.
Shahid et al.	2025	Threat Profiling	Machine learning supports profiling by identifying patterns in data.
Khalil et al.	2024	Threat Profiling	ICS modeling needs structured frameworks covering safety, security, and privacy.
Xiong and Lagerström	2019	Threat Profiling	Field lacks common ground and remains largely manual.
Rodríguez et al.	2024	Threat Profiling	Profiles attacker behavior by linking ATT&CK tactics using process mining.
Möller	2020	Threat Profiling	Uses attack models and scenarios to describe adversary actions and risks.
Erbas et al.	2025	Threat Profiling	MISP-based CTI enriches threat models with contextual, up-to-date data.
Podlesnik et al.	2025	Threat Profiling	Integrating threat intelligence with modeling improves cyber resilience.

Table 3.1: An overview of all papers used in the Related Work chapter, with the year of publication, their corresponding category, and their contribution to this thesis.

4

Methodology

Based on the research gap identified in Chapter 3, this chapter defines the research question of this thesis and provides the methodology used to answer it. It provides detailed explanations of the steps taken throughout the research, including the development of the threat profiling framework and evaluation rubric, the prompt engineering process, and the custom GPT development.

4.1. Research Question

The main research question of this study is: "*How can LLMs be used to support the development of threat profiles for a financial institution, and how can the generated profiles be validated?*". To address this research question, several sub-questions have been generated:

1. What input data and prompts are required for an LLM to generate a threat profile?
2. Which steps in the threat profiling process can be automated or supported using an LLM?
3. How can the generated threat profiles be validated to assess their quality and reliability?

The first sub-question investigates what prompts or instructions are necessary to generate a threat profile, and how input data can be used to support the threat profiling process. This includes how threat intelligence is filtering and mapped to the components of the threat profile. Next, the second sub-question investigates what steps in the threat profiling process can be automated using an LLM. Identifying where manual effort can be reduced in the threat process. Finally, the third sub-question investigates how the generated threat profiles can be validated to assess their quality and reliability. This includes a technical, and expert based validation. The technical validation will compare outputs to available threat data of an organization, while the expert based validation is performed by evaluating the outputs using a structured evaluation rubric.

4.2. Threat Profiling Framework

A threat-centered approach was chosen for the threat profiling framework to provide a coverage of a broad range of threats. By considering a broad set of threats, a larger portion of available threat intelligence can potentially be used. Since threat profiling can be a labor-intensive task, this research aims to let the LLM perform as much of the process as possible. Therefore, a threat-centric approach was considered to be the most suitable. Asset-centric and system-centric approaches would require substantially more manual input, as the chosen LLM cannot independently create a complete asset inventory or system inventory for an organization.

Because there is no universally adopted standard for threat profiles, there is no single framework on which this research can be directly based. Due to the overlap between threat profiling and threat modeling, multiple threat modeling frameworks from existing literature were analyzed to identify which components they include and which dimensions are used to measure those components. In this research,

a component is defined as an element that must be described to characterize a threat, such as a threat actor or threat event. Dimensions are defined as the attributed used to categorize each component, for example, a dimension of the Threat Event component is the event type. After the components and dimensions were mapped for each framework, the most common components were selected as the basis for the framework used in this research.

The resulting framework consists of four components: Threat Actor, Motivation & Intent, Threat Events, and Consequences. These components and their corresponding dimensions are shown in Table 4.1. Together, these components provide a structured description of a threat by capturing the adversary, their motivation, the attack itself, and the potential consequences for the organization. The four components included in the framework are Threat Actor, Motivation & Intent, Threat Events, and Consequences. The *Threat Actor* component describes who the adversary is. Its dimensions are attacker category, attack groups, and attack sophistication. In this research, attack group refers specifically to named threat actor groups. The *Motivation & Intent* component explains why the adversary acts and what they aim to achieve, using the dimensions goals and desired cyber effect. The *Threat Events* component captures the observable actions that form the attack and is described through event type, attack pattern, Indicators of Compromise (IoCs) or Tactics, Techniques and Procedures (TTPs), and likelihood. Finally, the *Consequences* component represents the effect of the attack on the organization and is characterized by the dimensions 'type of impact' and severity.

To ensure consistent application of the framework, categorical levels were predefined for several dimensions. For attack sophistication, three levels were defined: low, medium, high. Low sophistication refers to simple, single-step attacks, like scanning activity or a basic phishing attempt. Medium sophistication refers to multi-step or automated attacks, for example, a Distributed Denial-of-Service attacks. High sophistication refers to advanced, tailored, or stealthy attacks, such as the use of custom malware or multi-stage intrusions. The likelihood of a threat event was categorized as rare, occasional, or frequent. Rare refers to threat events that are unlikely to occur and are only expected under exceptional circumstances. Occasional refers to threat events that may occur from time to time but are not expected regularly. Frequent refers to threat events that are expected to occur regularly or repeatedly. Finally, the severity of consequences was also categorized into three levels. Low severity implies attack with limited impact on the organization, where business operations can continue as usual. Medium severity refers to attacks that cause a temporary disruption, for example, when services are unavailable for a limited period but can be restored without any major long-term effects. High severity refers to attacks that result in significant operational or financial damage, such as major financial losses or situations in which the infrastructure needs to be rebuild, affecting the business operations significantly.

4.3. Evaluation Rubric

A single threat profile cannot be considered the complete truth, as it is impossible to identify and map all possible threats. Therefore, no ground truth is available against which the generated output can be compared. As a result, evaluation metrics as accuracy, precision, or recall are not applicable, since they require comparison to a known complete truth. For this reason, this research uses a rubric-based evaluation approach. A rubric with scores from 0 to 5 was developed to evaluate the generated threat profiles. The criteria in the rubric were determined through the analysis of threat profiling frameworks and existing threat profiles. The evaluation criteria and corresponding scores are presented in Table 4.2.

The criteria in the rubric were selected to assess whether a component is present, how it is described, and how useful it is within the overall threat profile. The rubric includes six criteria: presence of the component, the level of detail, context relevance, mapping to other components, prioritization, and likelihood assessment. The first criteria determines whether the component is present in the output. If the component is missing, it will receive the lowest score, since its absence affects the quality of the threat profile. The rubric then considers how the component is described. Generic descriptions of the component receive a lower score, while concrete and detailed descriptions receive a higher score. In this study, a detailed description is defined as a description that goes beyond a general label, it includes ex-

Component	Description	Dimensions
Threat Actor	Who is the adversary	• Attacker category (e.g. cybercriminal, nation-state, insider) • Attack groups (if known, i.e. Lazarus Group, GhostSec) • Attack Sophistication (Low, Medium, High)
Motivation & Intent	Why the adversary acts, and what they aim to achieve	• Goals (e.g. financial gain, espionage, extortion) • Desired cyber effect (disruption, exfiltration, malware deployment etc.)
Threat Events	Observable actions that form the attack	• Event type (brute force login, phishing attempt, malware delivery) • Attack pattern • IOCs or TTPs (malicious IPs, hashes, ATT&CK techniques) • Likelihood (rare, occasional, frequent)
Consequences	What are the consequences of a successful attack	• Type of impact (Financial loss, data breach, service disruption) • Severity (low, medium, high)

Table 4.1: Threat profiling framework applied in this research. The table presents the four selected components of the framework and their corresponding dimensions: Threat Actor, Motivation & Intent, Threat Events, and Consequences. These components capture who the adversary is, why they act, how the attack manifests, and what impact it may have.

0	1	2	3	4	5
The component is not mentioned or is absent from the output.	Only generic labels are used to describe the component. No examples or further explanation are provided.	Only generic labels are used to describe the component; some descriptive explanations or examples are provided, but the information remains context-independent.	The component is described using concrete details or examples, but lacks prioritization, likelihood assessment, or explicit mapping to other components.	The component is described with concrete and relevant details and includes either a mapping to other components, prioritization, or a likelihood assessment.	The component is described with concrete and relevant details, includes a clear mapping to all other components, and provides prioritization and a likelihood assessment.

Table 4.2: The evaluation rubric used in this research. This rubric uses a scale from 0 to 5, and assesses whether a component is present, how it is described, and how useful it is within the overall threat profile.

amples, characteristics, or contextual information to make the description more meaningful. The rubric also evaluates whether the description is relevant to the context of the threat profile, in this research this means relevance to the financial sector. Again, general or context-independent descriptions will receive a lower score, and descriptions that are tailored to the specific sector or even organization will gain a higher score.

Additionally, the rubric considers whether the component is mapped to other components in the threat profile. Higher scores are assigned when the relationship between components is described clearly, as this improves the coherence of the overall profile. Finally, the rubric includes prioritization and likelihood assessment. When there is an indication of how urgent a threat is, the component will receive a higher score, while a lower score is assigned when this is missing. Similarly, when an indication is present of how likely the component will occur, a higher score will be rewarded. The prioritization and likelihood

assessment improve the practical value of the overall profile, since they support decision making in defensive strategies and risk assessments. Combining these six criteria, they provide a structured way to evaluate the generated threat profiles on a scale from 0 to 5.

4.4. Prompt Engineering and Testing Process

All prompts were tested using the same model: GPT-5.2 Thinking in standard thinking mode. This was the most recent thinking model available as of January 2026. While other model and options were also available, such as the Pro model, extended thinking mode, and deep research, these were restricted by limited monthly usage. Therefore, for the continuity of this research, it was decided to use the standard thinking mode for all experiments.

The prompt engineering process followed an iterative cycle, as shown in Figure 4.1. First, a simple prompt was created. This prompt was then run multiple times to evaluate the generated output and to assess whether the results were consistent. In this research, each prompt was run at least five times. Using the rubric shown in Table 4.2, each component was evaluated to determine which component needed improvement. Based on these findings, the prompt was adjusted accordingly. This process was repeated until the prompt produces outputs with sufficient quality and consistency.

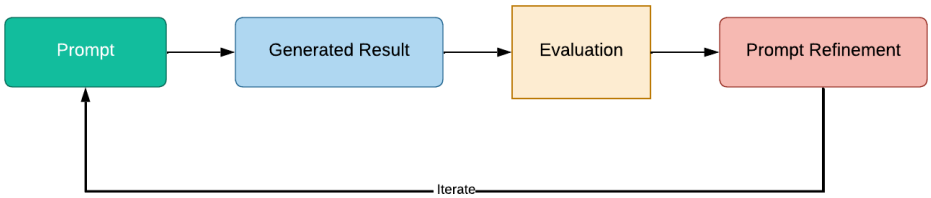

Figure 4.1: The prompt engineering process followed in this research. It is an iterative cycle that evaluates prompts and adjusts the prompt accordingly to get a better result.

4.5. Custom GPT

To develop the custom threat profiling GPT, a structured configuration was applied consisting of three components: instruction configuration, setting up a threat intelligence, and custom actions configuration. The instruction configuration defines the behavior of the GPT, it specifies the role of the GPT and provides guidelines so it behaves as desired. The threat intelligence set up includes a description of filters and fields used before it is sent to the GPT. And the actions configuration will determine what external sources the custom GPT can access and how they are integrated. Figure 4.2 provides an overview of the design and interaction between these components. Together, they form the foundation of the custom GPT developed for this study. The following subsections describe the design and implementation of each component in more detail.

4.5.1. Instruction Configuration

The instructions of a custom GPT define how the GPT behaves [27]. They specify what the GPT should do, what information it should use, how the information should be used, and what it should avoid. The instructions are applied to every conversation with the custom GPT, and are similar to sending a prompt to ChatGPT.

For this research, the best performing prompt from the initial experiments was used as foundation for the instruction configuration. Additionally, the instructions describe how the GPT should interact with the configured actions. Through these actions, the custom GPT receives threat intelligence from MISP. The instructions include a guide on how to call the actions, and when the retrieved data from the actions should be used. A mapping is included so the custom GPT knows what data to use for each component in the threat profiling framework. The Threat Actor Galaxy is mapped to the Threat Actor and Motivation & Intent components and the MISP events and attributes are used for the Threat Event component.

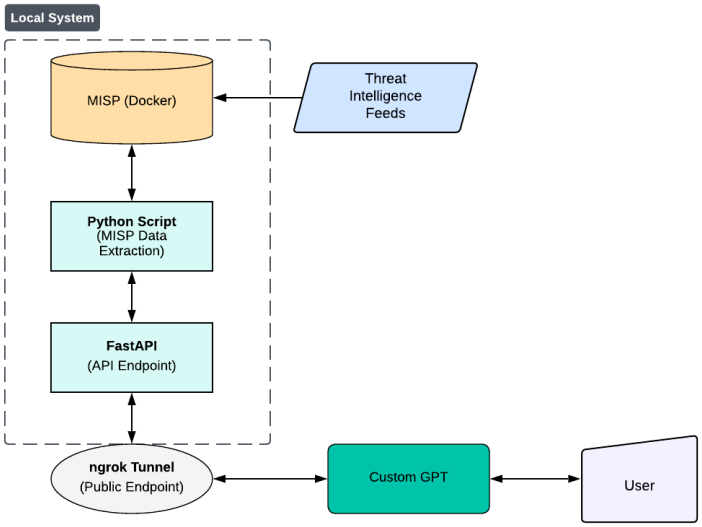

Figure 4.2: Overview of the custom GPT design used in this research.

4.5.2. Threat Intelligence setup

The threat intelligence used in this study, is collected using MISP which acts as a centralized repository for threat intelligence feeds. This setup allows multiple threat intelligence feeds to be accessed and aggregated through a single interface for further processing and analysis. During this study, MISP was deployed locally using a Docker container [7], to provide a reproducible environment for data collection and experimentation.

Initially, a single threat intelligence feed was integrated into MISP to test the functionalities of the custom GPT and its usefulness for threat profiling. The feed used for testing was the *CIRCL OSINT Feed* [6]. This open-source feed is provided by the same organization responsible for maintaining MISP and has the standard format used in MISP, so it was ready to use and did not require additional configurations. Open-source threat intelligence was used during this research to enable reproducibility in this setup/configuration.

To retrieve and process data from MISP, the REST API functionality was used through a custom Python script. Authentication to the MISP instance is handled by using an authentication key in combination with the MISP server URL, enabling secure access to the platform. The Python script performs API queries to extract the relevant data from MISP. Subsequently, the retrieved data is filtered and normalized to ensure usability and relevance for integration with the custom GPT.

Filtering of the data involved removing non-essential fields that were not relevant to the threat profiling framework to reduce noise in the data. In this study, non-essential fields are defined as fields that did not contribute to identifying or describing threat actors, motivations, threat events, or consequences. For example, the reference field containing a link to external sources was excluded as it was not needed for the threat profile itself. After filtering the data, normalization was applied to ensure consistent data structures. For example, fields such as *targeted-countries* were combined into a single field containing all relevant countries, rather than multiple separate entries. This made the data easier to process and map to the framework.

To enhance the Threat Actor component of the threat profiling framework, the MISP Galaxy ”Threat Actor” [21] was integrated. A galaxy within MISP is a structured knowledge base that describes broader threat intelligence concepts, such as threat actors, malware families, attack techniques, or sectors. The Threat Actor galaxy contains clusters that represent threat actors or adversary groups and provide structured contextual information such as a description of the adversary, aliases, targeted countries and targeted sectors.

The Threat Actor galaxy was retrieved using the endpoint *POST /galaxy_clusters/restSearch* and filtered on the galaxy type ”Threat Actor”. Because this research focuses on the financial sector, not all threat actors in the galaxy were relevant. Therefore, additional filtering was applied to create a data set with only relevant threat actors. Since the sector relevant information was not used consistently in the galaxy, all fields were analyzed to identify which contain sector relevant information. This ensured that all fields that could be used for sector filtering were included. The analysis resulted in applying filters on the *target-sector* and *cfr-target-category* fields. The following filters were used to select the threat actors relevant to the financial sector:

- **target-sector**: *Finance, Trade, Bank, Payment, Investment.*
- **cfr-target-category**: *Private sector, Information technology, Finance, Financial, Private Sector, Financial services, Information Technology.*

After filtering the threat actors based on the relevant sectors, the non-essential fields were removed to reduce noise in the dataset. The remaining fields and their description are listed in Appendix A.1. The resulting output was a JSON file containing the filtered threat actors and relevant contextual information, including their description, aliases, known targets, and their motivation. This enriched the Threat Actor component by providing detailed examples of adversary groups relevant to the financial sector.

To enrich the Threat Event component of the threat profile, MISP events were used as an additional source of threat intelligence. These events were retrieved using the endpoint *POST /events/restSearch*. To reduce noise and outdated information, only events from the past year were included. While the time frame is relatively broad for threat intelligence, it was selected to identify recurring patterns in threat events. The filtering for threat events was based on event tags. Only events containing tags related to the financial sector were used, specifically tags that contained *Finance*, *financial services*, or *Bank*. Relevant events may have been excluded if they were not tagged consistently or did not contain one of the selected tags. This was considered an acceptable trade-off, as including all available events would introduce noise and reduce the contextual relevance of the threat intelligence. By removing the non-essential fields, the output gave a clear image of when an event occurred, an event description and additional information like the threat actor or method that is shared in the tags or through the galaxies. An overview of all fields included in the threat event entries can be found in Appendix A.2.

Finally, to further enrich the Threat Event component, the attributes of MISP events were also collected. Attributes represent individual data points within an event, such as indicators, technical artifacts, or contextual information. These attributes can provide more detailed and technical information about threat events. The attributes were retrieved through the MISP API using the endpoint *POST /attributes/restSearch*. Since MISP attributes can contain a wide range of information, filtering was applied to retain only attributes relevant to the threat profiling framework. The selection was based on the MISP attribute categories and types defined in the MISP data model [20]. The following attribute categories were selected, the descriptions are provided by MISP [20]:

- **Attribution**: Identification of the group, organization, or country behind the attack.
- **Payload delivery**: Information about how the malware is delivered.
- **Payload type**: Information about the final payload(s).
- **Persistence mechanism**: Mechanisms used by the malware to start a boot.
- **Targeting data**: Internal Attack Targeting and Compromise Information.

These categories were selected because they provide information that can support the identification and description of threat events. Payload delivery and payload type can help describe attack patterns, while persistence mechanisms can support the mapping of observed activity to intrusion-related threat events. Targeting data helps determine whether the event is relevant to the financial sector. An overview of all the fields used to describe the attributes can be found in Appendix A.3.

4.5.3. Actions Configuration

There are two elements that enable the connection of the custom GPT with the MISP API: FastAPI [33] and ngrok [25]. FastAPI acts as the gateway between the GPT and the MISP API. It receives requests from the custom GPT and translates them into calls for the MISP API, and returns the processed results.

Because MISP and FastAPI run locally, they are not directly accessible from the internet. To resolve this, ngrok is used to create a tunnel for the requests between the GPT and FastAPI through a HTTPS endpoint. Allowing the custom GPT to send requests to the FastAPI gateway.

One of the restrictions of the actions of a custom GPT is that a request cannot take longer than 45 seconds or it will automatically time out. To prevent this timeout, the calls of the gateway are split in two. When the first request comes in, the job is started in the background. Then once the job is done, it will get the status 'done' and can be retrieved and used for the threat profile.

4.6. Validation

In this research, the validation of the threat profiles consists of two elements: technical validation and expert based validation. Together, these validations are used to determine whether the generated threat profiles are realistic, relevant, and useful in practice.

4.6.1. Technical validation

The technical validation investigates whether the generated threat profiles are reflected in observed security monitoring data. Because of the sensitivity of the underlying data, the validation was conducted and reported at an aggregated level, meaning individual alerts, events, systems or logs were not disclosed. The goal of the technical validation is to asses whether the threat profile reflects real world activity.

For the technical validation a time frame of three months was used. This time frame was selected to ensure the analysis is focused on recent activity and all data sources could be consulted to get as much context as possible. Informational and low relevance data was excluded to reduce noise and to focus on meaningful security data. To validate the threat profiles, the security data was first reviewed and categorized by data type to support consistent mapping to the threat profile claims later on during the validation process. For each experiment, one generated threat profile was selected for technical validation. The claims in these profiles were then separated according to the four components of the framework and mapped to the corresponding security data where possible.

Each component was validated using a different approach. Threat Event claims were directly mapped, Motivation & Intent and Threat actors were partially mapped using the security data, and partially through inferring from threat events. Finally, Consequences were validated using both technical evidence and reasoned inference. Some consequences could be linked to observed threat event patterns, while others required interpretation based on known impact relationships, regulatory expectations, or expert judgment.

To classify the technical validation results, four validation categories were used: supported, partially supported, not observed, and not assessable. These categories were applied to each claim in the generated threat profiles. See the definition of each of these categories below.

- **Supported:** Relevant activity patterns were observed in the available validation data.
- **Partially supported:** Weak, indirect, or benign-but-relevant indicators were observed.
- **Not observed:** The claim could be assessed with the available data sources, but no relevant activity was found during the selected time frame.
- **Not assessable:** The available validation data did not provide sufficient visibility to assess the claim.

To demonstrate how the validation categories were used, the following example is provided. Suppose that a threat profile contains phishing as one of the threat event claims. This claim would be supported when the available data contains relevant detections matching this threat event, so there would be confirmed phishing activity. It would be partially supported when related but low-confidence indicators are present, indicating that the activity resembles the threat event but it does not confirm that the event truly occurred. In this case, an email that looks like phishing was received, but it cannot be confirmed that it is a phishing mail. It would be not observed when the relevant data sources were checked but no matching activity was found, so no phishing attempts have been found in the data. It would be not

assessable when the available validation sources do not include email, web, or other data sources capable of detecting this type of activity.

4.6.2. Expert Based Validation

The expert based validation focuses on evaluating the quality and practical usefulness of the generated threat profiles. Three participants with experience in cybersecurity, either through academic research or professional practice, were asked to evaluate the generated threat profiles using the evaluation rubric defined in Table 4.2. The evaluation rubric was created to assess whether a component is present, how well it is described, and how useful it is within the overall threat profile. Therefore, this validation step can be seen as validating the usefulness of the profiles. Since the initial runs were solely evaluated by the researcher, this validation step helps reduce individual bias and provides a more balanced assessment of the threat profiles.

The expert based validation was analyzed using the mean and standard deviation of the expert scores. The mean score was used to represent the central score of the expert valuations, while the standard deviation was used to assess the level of agreement between the experts. A low standard deviation would indicate that the experts assigned similar scores, whereas a higher standard deviation indicates more variation in their evaluations.

Since the evaluation rubric scores on a scale of 0-5, the following interpretation was applied to the standard deviation:

- A standard deviation between 0 and 0.5 indicates low variance, meaning that the experts strongly agree.
- A standard deviation between 0.5 and 1 indicates moderate variance, meaning that the experts differ slightly in their evaluations.
- A standard deviation larger than 1 indicates high variance, meaning that the experts disagree substantially.

5 Results

This chapter presents the results of the prompt engineering experiments conducted using the developed threat profiling framework. The process of improving the threat profiles is based on a combination of understanding the functionalities of ChatGPT and the prompt engineering process to get the best possible threat profile. These results provide insight into how ChatGPT and the custom GPT can support the creation of threat profiles.

5.1. Zero-Shot Prompting

While there are threat landscape reports available for different sectors, which provide a broad overview of threats in the respective sector, threat profiles for organizations are not publicly available. Therefore, the initial tests will be performed by using zero-shot learning. This will create a general understanding of ChatGPT's knowledge on threat profiling, it guides the experiments to see where improvement is needed, and what data could be useful to add.

5.1.1. Model Selection

To determine the difference between the available ChatGPT models, the LLM has been tested using a simple prompt to get a sense of its basic knowledge about threat profiling, and to get a better understanding of how the different models process prompts and deliver outputs.

Table 5.1 shows the results of these initial experiments. Based on the evaluation rubric in Table 4.2, the Pro model performed best with this prompt. This model takes more time to think, so it can generate better responses. The generated threat profile has a threat actor first narrative, giving the relevant motivations, threat events and consequences for each individual threat actor. The Pro model searched the internet to improve its answer, by including relevant information from 177 sources that were found online. The Thinking model scores second best, and follows a similar approach by combining threat actors and their motivations, and combining threat events and their respective consequences, followed by explicitly mentioning the most important consequences. Just like the Pro model, the Thinking model searches the internet for relevant information. The main difference is that the Thinking model spends less time searching for sources, and only used 81 sources. The Instant model generated lists for each component, including very generic examples without any clarification. Finally, the Auto model gave the impression that it was a combination of the Instant and Thinking models, where some information about the components is quite limited while other parts include examples.

Overall, all generated threat profiles provide an overview of common threats for an organization. However, the profile style differs clearly between the models. Despite the differences in presentation, the profiles remain mostly descriptive, and lack contextual details about the organization or sector. As a result, the generated profiles offer limited practical value. They provide a broad overview of threats, but give limited information that helps organizations in their defensive strategy against threats.

	Auto	Instant	Thinking	Pro
Threat Actor	2	2	3	3
Motivation & Intent	1	1	3	3
Threat Events	3	2	3	4
Consequences	2	2	3	4

Table 5.1: Results of running the same prompt on different models. The Pro model scores best, while the Instant model has the lowest scores.

5.1.2. Establishing a Baseline for Zero-Shot Learning

The results from subsection 5.1.1 showed considerable variability in output formats. To encourage more consistency, an explicit output pointer was added to the prompt. Previous experiments showed that ChatGPT has a tendency to structure response as lists. Therefore, the concise bullet points were selected as the most natural output format. This format was not considered the final intended style for the threat profiles, instead it was used to test whether an output pointer could lead to more consistent results.

After testing the prompt several times, the output stayed consistent throughout all attempts. However, after evaluating the output, it was clear that this prompt creates worse threat profiles than previous experiments, the output was very generic and not context specific. Table 5.2 shows that the prompt scores lower for at least three of the four components when comparing with the scores from the Thinking or Pro model in Table 5.1, which is a significant decline in performance. Although the structure of the response became more stable, the content became more generic and less context specific. As a result, consistency in output format alone is not enough, since it may come at the cost of performance.

	Mean Score
Threat Actor	2
Motivation & Intent	2
Threat Events	2.6
Consequences	3

Table 5.2: The mean evaluation scores after including an output pointer in the prompt. While the output was more consistent, the scores were lower than previous experiments.

5.1.3. Framework Guided Zero-Shot Learning

To improve the quality of the generated threat profiles, the entire framework from Table 4.1 was included in the prompt. Adding the framework gives ChatGPT more guidance to what is expected to be in the threat profile, as the framework includes all dimensions and example values for the dimensions. The goal was to get a more structured output, since the framework provides guidance for the generation of threat profiles.

Testing the prompt across five runs showed a slight improvement in the Threat Actor score, while the scores of the other components did not improve, as shown in Table 5.3. The score for Motivation & Intent stayed the same, while Threat Events and Consequences decreased when comparing to Table 5.2. The improvement in the Threat Actor component was caused by profiles including more information about the threat actors, which confirms that ChatGPT used the threat profiling framework. For the Threat Event and Consequences components, more information was also provided, but ChatGPT failed to make it context specific, which led to lower scores.

The adjustment to the prompt did not result in better scores, they remain similar to the scores from Table 5.2. Although the inclusion of the framework ensured that more information was provided, this did not lead to better explanations in the generated threat profiles. The output lacked context-specific details needed to make the profile relevant to the financial sector, which makes the overall result insufficient and not directly applicable for organizations. A possible explanation is the requested output format. The prompt currently asks for concise bullet points, while a threat profile is better suited to a more report-like form in which components can be connected and their relationships are made explicit.

	Mean Score
Threat Actor	2.6
Motivation & Intent	2
Threat Events	2.4
Consequences	2.8

Table 5.3: The mean evaluation scores of the five prompt runs. The scores did not improve by adding the threat profiling framework to the prompt.

Therefore, simply adding the framework to the prompt is not sufficient to improve performance. The output and performance remained consistent across the runs, but the scores did not improve, suggesting that the output pointer may be the bottleneck.

5.1.4. Updated Zero-Shot Prompt

The previous experiments showed that the current prompt was not sufficient to generate a coherent threat profile, due to the limited output pointer. Therefore a different approach was used. The new prompt was divided into four parts: role, task, context, and output format. First, ChatGPT was assigned the role of a threat intelligence analyst specializing in the financial sector in the Netherlands, to guide the type of knowledge and perspective expected in the response. Secondly, the task was defined as generating a threat profile that describes threat actors, motivation & intent, threat events, and consequences. Next, the prompt included additional context by specifying the scope and constraints, like keeping it sector relevant and explicitly state when information is unknown, and the complete threat profile framework is given to establish was information is expected in the threat profile. Finally, the output format was divided into two parts. Part A contained the reference components that define the building blocks of the threat profile, while part B required a coherent threat profile, including an executive summary, threat landscape, threat actor profiles, common attack chains, and an impact and consequences summary. Dividing the output into two parts was expected to improve the quality of the generated threat profile, as it gives the model more guidance and encourages a more structured reasoning process when generating the second part of the threat profile.

	Mean Score
Threat Actor	4
Motivation & Intent	3
Threat Events	4
Consequences	4

Table 5.4: The mean evaluation scores of the five prompt runs after changing the prompt. All four components scored better than the previous experiments.

Table 5.4 shows that the mean score of all four components improved compared to the previous experiments. This emphasizes the importance of selecting the right output pointer in a prompt. It is important to clearly describe what you want, otherwise ChatGPT may not generate it as intended. These tests also show that a clear prompt contributes to more consistent outputs. The scores across the different runs are very similar, which indicates that the results are consistent as well.

Threat profiling is a domain-specific task, so improvements from this point on will probably be limited, as additional specialized data is required. Although the current prompt produces better and more consistent threat profiles, the output is still limited in how useful it is for organizations. To make the profiles more actionable in practice, threat intelligence data will be added, so the generated profiles better reflect the actual threat landscape, and relevant threats.

5.2. Custom GPT

The final experiments from the zero-shot prompting phase showed that ChatGPT can achieve strong overall scores when guided by a clear and structured prompt. However, the generated profiles still lacked clear prioritization and consistent likelihood assessments, and the data lacked actionable details to support practical use. To examine whether these limitations could be reduced, the next step was to develop a custom GPT, as this offers more flexibility and customization than the regular ChatGPT interface. This allows the combination of a fixed instruction set with additional threat intelligence data from MISP to enhance the generated threat profile.

5.2.1. Comparison to ChatGPT

Before the connection with MISP was implemented, a comparison was made between ChatGPT and the custom GPT. This comparison was done to better understand how both models used the provided data and to determine if any differences appeared when only a file was added. It was also used to assess whether the custom GPT needed further refinement to ensure that its evaluation scores were at least equal to those of ChatGPT. The experiment was performed by providing both models with an export of 100 threat events from MISP in a .json file, together with the prompt from Section 5.1.4. Only a small sample of unfiltered threat intelligence data was used to avoid introducing too much noise that could immediately affect the threat profile. At the same time, this setup provided an indication of how both ChatGPT and the custom GPT handled data, including how they dealt with information that was not directly relevant to the generated profile.

	Mean ChatGPT	**Mean Custom GPT**
Threat Actor	4	4
Motivation & Intent	3	3
Threat Events	4	4
Consequences	4	4

Table 5.5: The mean scores of both ChatGPT and the custom GPT runs when including 100 threat attribute samples. The GPTs score similar, showing that they work similar.

The mean evaluation scores of testing the prompts five time for each model, are shown in Table 5.5. These results show that both models perform similarly. This was further confirmed upon inspection of the generated outputs, which showed that the threat profiles were similar as well. The main difference between ChatGPT and the custom GPT is how the provided threat intelligence was used. ChatGPT generally used a single threat actor as an example, without assessing whether or not that actor was relevant to the context. Conversely, the custom GPT mainly used the threat intelligence in the second part of the threat profile, specifically in the *Actor Profile* and the *Common Attack Chains* sections. In two runs, the custom GPT explicitly stated that the provided information was not specifically relevant to the specified region, and hence would not used as an example. However, the underlying approach was still used to construct a possible threat event for the requested region.

These experiments indicate that there is no significant difference in performance between the custom GPT and ChatGPT. The instructions configured in the custom GPT appear to operate in a similar manner to prompts entered manually in ChatGPT, suggesting that no further refinement of the custom GPT instructions is required to achieve similar performance. Although both models produce similar outcomes, the main advantage of the custom GPT is its improved efficiency. The threat intelligence data does not need to be manually attached to each prompt, as it is stored in the GPT's knowledge base during configuration and can be accessed whenever a threat profile needs to be generated. And since the instructions include the elaborate prompt, the user can just use the prompt "create a threat profile for me". This reduces both time and manual effort from a user. A logical next step would be to build on this efficiency by adding an action that calls the MISP API, allowing threat intelligence data to be retrieved directly from MISP rather than being downloaded and uploaded manually.

5.2.2. Threat Intelligence Enrichment

To build on the efficiency advantage of the custom GPT, threat intelligence from MISP was used to enrich the generated threat profiles. This allows the model to include more relevant and recent examples, making the profiles more tangible. Because unfiltered threat intelligence can introduce substantial noise, the MISP data is filtered before the custom GPT receives it. The instructions were also updated to clearly indicate when the MISP data should be used, to ensure that the GPT analyzes and applies the data in the intended way. More specifically, threat actor information is used for the Threat Actor component, and when applicable, for the Motivation & Intent component, while threat attributes and threat events are used for the Threat Event component.

	Mean Score
Threat Actor	4
Motivation & Intent	3
Threat Events	4
Consequences	4

Table 5.6: The mean evaluation scores of the five prompt runs after enriching the threat profile with threat intelligence from MISP. The scores did not improve compared to the previous experiments.

The addition of more threat intelligence did not lead to an improvement in evaluation scores, which remained similar to those after the last prompt adjustment. This indicates that performance has stagnated, despite the generated threat profiles becoming richer and more specific. The added threat intelligence resulted in more concrete examples, such as named threat actor groups and malware families, making the profile more tangible, less generic, and more actionable for practical use. However, this did not result in higher scores. A likely explanation is that the instructions do not explicitly describe the criteria needed to achieve the highest possible scores, which causes not all components being mapped to all other components, inconsistent prioritization, and missing likelihood assessments. Compared to the previous experiment with the small threat intelligence sample, the custom GPT no longer indicated that part of the threat intelligence was irrelevant, which may suggest the filtering was effective and the data could be used more directly in the generated profile. This suggests that further improvement does not depend on adding more data alone, but on refining the prompt so the relationships between components, as well as prioritization and likelihood assessments are made explicit.

5.2.3. Score Maximization Adjustment

As stated in subsection 5.2.2, refining the prompt may improve the evaluation scores. To maximize the score, the prompt was adjusted to explicitly state that all components must be linked to each other, that a likelihood assessment must be included for all components, and a prioritization must be given for the most relevant items within each component. Because the prompt is explicitly tailored to the highest score in the evaluation rubric, it introduces a lot of bias. The instructions tell the custom GPT exactly what to do to achieve the highest possible score for each component. However, this does not mean that the custom GPT will automatically generate a perfect-scoring threat profile. Instead, it is given the right conditions to construct such a profile. To better understand the effects of this bias, a second, more neutral prompt was created without these explicit instructions. This allows the natural behavior of the custom GPT to be observed when it is not guided towards consistent mapping, prioritization, and likelihood assessments.

	Mean Neutral Prompt	**Mean Maximization Prompt**
Threat Actor	4	5
Motivation & Intent	3	4
Threat Events	4	5
Consequences	3	4

Table 5.7: The mean evaluation scores for both the score maximization experiment and the neutral prompt. The results indicate that score maximization improves the threat profile significantly, and that the results would not improve without the explicit score maximization instructions.

Table 5.7 shows that the custom GPT achieves the maximum score for the Threat Actor and Threat Event components, while Motivation & Intent and Consequences both improve by one point. These two components do not reach the maximum score because they lack stronger analytical descriptions. Compared to the neutral prompt, the maximizing prompt scores one point higher on all components, showing that structural requirements can significantly improve the evaluation scores. Nevertheless, Motivation & Intent and Consequences remain the lowest-scoring components, indicating that further improvement depends less on the output structure and more on the analytical depth of the generated output.

Experiment	Threat Actor	Motivation & Intent	Threat Events	Consequences
Zero-Shot Baseline	2	2	2.6	3
Framework Guided	2.6	2	2.4	2.8
Prompt Adjustment	4	3	4	4
Custom GPT	4	3	4	4
Score Maximization	5	4	5	4

Table 5.8: Summary of the mean scores across all experiments. The results show a clear improvement in all components as the prompt design becomes more structured and tailored.

Table 5.8 shows that the evaluation scores improved as the prompt design became more structured and specific. The Zero-Shot Baseline and Framework Guided experiments received relatively low scores across all components, indicating that the generated threat profiles were insufficiently cohesive. The Prompt Adjustment experiment caused an improvement in all evaluation scores. The custom GPT achieved similar scores, suggesting that the additional threat intelligence did not contribute to the cohesiveness of the threat profiles. But the threat intelligence did make the claims in the threat profiles more specific. Finally, the Score Maximization experiment achieved the highest scores overall, with maximum scores for Threat Actor and Threat Events, and an improvement for Motivation & Intent and Consequences. This indicates that explicitly instructing the model to include prioritization, likelihood assessments, and mappings between components improved the evaluation scores of the generated threat profile. However, Motivation & Intent remained one of the lower scoring components across all experiments, this suggests that this component is more difficult to describe with analytical depth for the LLM.

6

Validation

As stated in Chapter 4, the threat profile validation step is divided into two parts: a technical validation, and an expert based validation. In the technical validation, the goal is to determine if the generated threat profiles reflect the real world. And the expert based validation evaluates the quality and practical usefulness of the threat profiles.

6.1. Technical validation

Due to the sensitive nature of the security monitoring data, validation results are presented at an aggregated level. The validation focuses on the presence of patterns corresponding to the components of the threat framework. For the Threat Event component this will be directly through an analysis of the validation data. While Threat Actors and Motivation & Intent will be inferred from threat events, and consequences will be inferred from available evidence and knowledge. As explained in Section 4.6.1, each claim was classified into one of four categories: Supported, Partially Supported, Not Observed, and Not Assessable. To clarify how these categories were applied to claims, Figure 6.1 presents the corresponding decision tree.

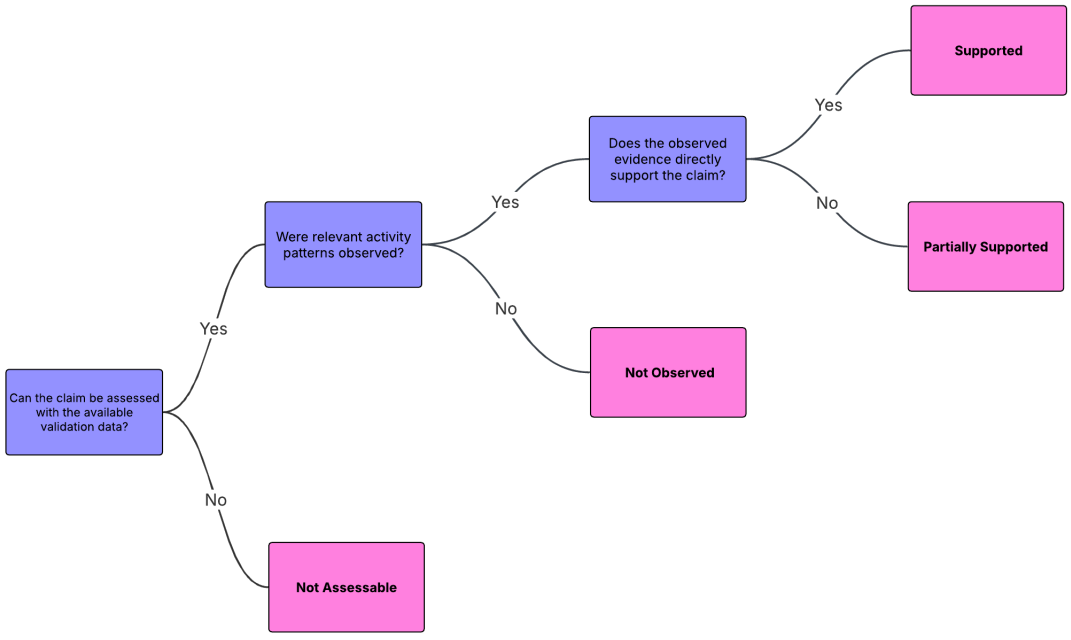

Figure 6.1: Decision Tree used to classify each claim during the technical validation.

The Zero-Shot Baseline experiment had eight supported claims and seven partially supported claims out of 30 total claims, as shown in Table 6.1. This means that half of the claims were at least partially supported by the available security data. Threat Actor, Threat Event, and Consequences each had four claims that were classified as not assessable. In total, 14 out of 30 claims were not assessable, while only one assessable claim was not observed.

Component	Not assessable	Not observed	Partially supported	Supported	Total
Threat Actor	4	0	0	3	7
Motivation & Intent	2	0	3	2	7
Threat Event	4	1	2	2	9
Consequences	4	0	2	1	9
Total	14	1	7	8	30

Table 6.1: Technical validation results for the Baseline Experiment. From the 30 claims, eight were directly supported by the data, and seven claims were only partially supported.

The technical validation results of the Framework Guided experiment are shown in Table 6.2. In this experiment, seven claims were supported and seven claims were partially supported. Which means that 14 out of 26 claims were at least partially supported. Again, Threat Actor, Threat Event, and Consequences are tied for most claims that were not assessable, while only two assessable claims were not observed in the data.

Component	Not assessable	Not observed	Partially supported	Supported	Total
Threat Actor	3	0	1	2	6
Motivation & Intent	1	0	2	2	5
Threat Event	3	1	3	2	9
Consequences	3	1	1	1	6
Total	10	2	7	7	26

Table 6.2: Technical validation results for the Framework Guided Experiment. Out of the 26 claims, 7 were fully supported and 7 were partially supported.

Table 6.3 shows the technical validation results of the Prompt Adjustment experiment. In total, seven claim were fully supported and nine were partially supported, meaning that close to half of the claims were at least partially supported. The consequences component had five claims classified as not assessable, which was the highest number among all components. In total, five assessable claims were not observed in the data, which is higher than in the previous experiments. And compared to the previous experiments, fewer claims were classified as not assessable. This suggests that the Prompt Adjustment experiment produced more claims that could be evaluated using the available validation data. However, this also resulted in more claims being classified as not observed, because they were not reflected in the data during the selected time frame.

Component	Not assessable	Not observed	Partially supported	Supported	Total
Threat Actor	2	0	1	2	5
Motivation & Intent	3	0	3	1	7
Threat Event	2	2	3	3	10
Consequences	5	3	2	1	11
Total	12	5	9	7	33

Table 6.3: Technical validation results for the Prompt Adjustment Experiment.

Table 6.4 presents the technical validation results of the Custom GPT experiment. In total, there were 36 claims to validate, eight claims were fully supported and eight partially supported. Meaning that 16 claims were at least partially supported by the data. This experiment had the highest number of not assessable claims, the Consequences component contained the largest number of not assessable claims. Additionally, five claims were classified as not observed, of which four were Threat Events classified as not observed. Compared to the Prompt Adjustment experiment, the custom GPT generated more claims that were not assessable. This suggests that the custom GPT generated more specific or detailed claims, making the components harder to validate technically.

Component	Not assessable	Not observed	Partially supported	Supported	Total
Threat Actor	2	0	2	2	6
Motivation & Intent	4	0	2	2	8
Threat Event	4	4	2	2	12
Consequences	5	1	2	2	10
Total	15	5	8	8	36

Table 6.4: Technical validation results for the Custom GPT Experiment. From the 36 claims made, 8 were fully supported and 8 were partially supported.

The results of the technical validation of the Score Maximization experiment are shown in Table 6.5. From the 33 claims that were made, seven are fully supported and nine partially supported by the validation data. This means that 16 out of 33 claim were at least partially supported. The Score Maximization experiment had fewer not assessable claims compared to the Custom GPT experiment. As in the other experiments, several not assessable claims were found in the Consequences component, showing that this component is difficult to validate using technical monitoring data alone. The results suggest that the Score Maximization experiment produced a profile with a similar level of support as the Prompt Adjustment experiment, but with stronger structure and more explicit component relationships.

Component	Not assessable	Not observed	Partially supported	Supported	Total
Threat Actor	2	0	2	1	6
Motivation & Intent	2	1	2	2	7
Threat Event	3	2	1	4	10
Consequences	5	1	4	0	10
Total	12	4	9	7	33

Table 6.5: Technical validation results for the Score Maximization Experiment.

Table 6.6 shows that the early zero-shot experiments achieved a higher support rate than some of the more elaborate prompts. The support rate is the percentage of supported and partially supported claims. This may be because the earlier outputs contained more general claims, which were easier to match to the available validation data. In contrast, the later experiments produced more detailed and specific claims, which made them harder to validate technically. Across all experiments, a large number of claims were classified as not assessable. This suggests that technical monitoring data alone is not sufficient to validate a complete threat profile, especially for claims related to actor attribution, motivation, business impact, or consequences. The results also show that optimizing a prompt for the evaluation rubric does not automatically make every generated claim technically verifiable. While prompt optimization improved structure and completeness, some claims still required evidence that was not available in the validation data.

Experiment	Not assessable	Not observed	Not supported	Partially supported	Supported	Total	Support rate
Baseline	14	1	0	7	8	30	50.00%
Framework Guided	10	2	0	7	7	26	53.85%
Prompt Adjustment	12	5	0	9	7	33	48.48%
Custom GPT	15	5	0	8	8	36	44.44%
Score Maximization	12	4	1	9	7	33	48.48%

Table 6.6: Comparison of technical validation results across experiments. The support rate was calculated as the percentage of supported and partially supported claims.

6.2. Expert based validation

For the expert based validation, three experts evaluated the generated threat profiles using the evaluation rubric from Table 4.2. For each experiment, the mean and standard deviation of the expert scores were calculated for each component. The expert scores were then compared to the baseline results presented in Sections 5.1 and 5.2. Experiments that were only conducted to better understand the behavior of ChatGPT and the custom GPT were left out of the expert based validation, as they were not specifically aimed at improving the quality of the threat profiles. Section 4.6.2 explained how the standard deviation values were interpreted. For clarity, Table 6.7 provides a summary of the applied interpretations.

Standard Deviation Range	Variance	Interpretation
0 – 0.5	Low	Experts strongly agree.
0.5 – 1.0	Moderate	Experts differ slightly.
> 1.0	High	Experts disagree substantially.

Table 6.7: Interpretation of standard deviation values used for the expert based validation.

The first experiment included in the expert based validation is the Zero-Shot Baseline experiment. In this experiment, the prompt instructed the model to generate a threat profile using concise bullet points as the output format. This format was selected because earlier observations showed that ChatGPT tends to use a similar structure when no explicit output format is provided. Therefore, this experiment functions as a baseline that is closest to ChatGPT's natural response style.

	Baseline	Expert 1	Expert 2	Expert 3	Expert Mean	Std. Dev.
Threat Actor	2	2	2	2	2	0
Motivation & Intent	2	2	2	2	2	0
Threat Events	2.6	2	2	3	2.33	0.58
Consequences	3	2	2	1	1.67	0.58

Table 6.8: Validation of the zero-shot baseline experiment. The experts uniformly agree on the scores for Threat Actor and Motivation & Intent, while there is some variation in judgment for the Threat Events and Consequences components.

The experts evaluations show uniform agreement for the Threat Actor and Motivation & Intent components, all experts assigned the same scores as the baseline evaluation, as can be seen in Table 6.8. For the Threat Events and Consequences components, the standard deviation is 0.58, indicating some variation in expert judgment. The expert mean for the Consequences components is lower than the baseline results. This suggests that the baseline evaluation may have assessed this component more positively than the experts did. Overall, the zero-shot baseline produces limited scores across all components, indicating that the quality and usefulness of the threat profiles is not sufficient.

The next experiment included in the expert based validation is the Framework Guided experiment. This experiment extended the prompt by including the threat profiling framework. The goal was to provide the model with clearer guidance on the expected components and dimensions of a threat profile.

	Baseline	Expert 1	Expert 2	Expert 3	Expert Mean	Std. Dev.
Threat Actor	2.6	2.8	3.4	4	3.4	0.60
Motivation & Intent	2	2	2.8	3	2.6	0.53
Threat Events	2.4	2.4	3	3	2.8	0.35
Consequences	2.8	2.4	2.8	2	2.4	0.40

Table 6.9: Expert validation results for the framework guided zero-shot experiment. There is slight variance for Threat Actor and Motivation & Intent, while there is a strong agreement for Threat Events and Consequences.

The expert evaluations show strong agreement for the Threat Events and Consequences components, with a standard deviation of 0.35 for Threat Events and 0.4 for Consequences. The expert scores differ slightly for Threat Actor and Motivation & Intent, the standard deviations remained close to 0.5 indicating only slight variance for these components. The expert mean for Threat Actor, Motivation & Intent, and Threat Events are higher than the baseline result, while the expert mean of Consequences is lower. This suggests that the baseline evaluation for this experiment may have been assessed more strict than the experts did. Ultimately, the scores remain relatively low, indicating that the threat profiles are not considered useful and complete.

The subsequent experiment included in the expert based validation is the Prompt Adjustment experiment. In this experiment, the prompt was redesigned to improve the threat profile quality. The output was clearly divided into two parts: first, an overview of reference components, and second, a coherent threat profile build from the components of these reference components.

	Baseline	Expert 1	Expert 2	Expert 3	Expert Mean	Std. Dev.
Threat Actor	4	4	4	5	4.33	0.58
Motivation & Intent	3	2	3	4	3	1
Threat Events	4	4	4	5	4.33	0.58
Consequences	4	4	4	4	4	0

Table 6.10: Expert validation results for the adjusted zero-shot prompt experiment. The results show improved scores across most components, with strong expert agreement on Consequences and disagreement on Motivation & Intent.

Table 6.10 shows that the adjusted prompt resulted in higher expert scores compared to the previous experiments. Threat Actor and Threat event received an expert mean of 4.33 and have a standard deviation of 0.58, indicating slight variation between expert evaluations. Consequences obtained an expert mean of 4, with standard deviation 0, showing the experts fully agreed on the evaluation of this component. The Motivation & Intent component received a mean score of 3 with a standard deviation of 1, indicating large variation in expert judgment. This suggests that experts interpreted the usefulness of this component differently. Compared to the baseline scores, the expert means are similar, indicating that the initial evaluation and expert validation are largely aligned for this experiment. Overall, the adjusted prompt clearly improved the generated threat profiles, although Motivation & Intent remains the weakest component.

The following experiment for expert based validation is the Custom GPT experiment. During this experiment, the custom GPT retrieves data from MISP to enrich the threat profiles with threat intelligence.

	Baseline	Expert 1	Expert 2	Expert 3	Expert Mean	Std. Dev.
Threat Actor	4	4	4	4	4	0
Motivation & Intent	3	3	3	4	3.33	0.58
Threat Events	4	4	4	4	4	0
Consequences	4	4	4	4	4	0

Table 6.11: Expert validation results for the Custom GPT experiment. The results show that Motivation & Intent is the only component with any variance.

The expert evaluations show strong agreement for the Threat Actor, Threat Events, and Consequences components. All three components received an expert mean score of 4 and a standard deviation of 0, as shown in Table 6.11. The only component with variation is Motivation & Intent , it received an expert mean score of 3.33 and a standard deviation of 0.58. Indicating again that Motivation & Intent remains the weakest component and is more difficult to evaluate consistently. Compared to the baseline evaluation, the expert scores are largely aligned. These results suggest that the threat profiles generated in the Custom GPT experiment are considered useful and mostly complete by the experts, although Motivation & Intent still requires improvement.

The final experiment included in the validation is the Score Maximization experiment. In this experiment, the prompt was adjusted in an attempt to get the maximum evaluation score by explicitly stating the requirements for each component.

	Baseline	Expert 1	Expert 2	Expert 3	Expert Mean	Std. Dev.
Threat Actor	5	5	5	5	5	0
Motivation & Intent	4	5	4.4	5	4.8	0.35
Threat Events	5	5	5	5	5	0
Consequences	4	5	5	5	5	0

Table 6.12: Expert validation results for the Score Maximization experiment. The results show near-maximum scores and strong expert agreement.

The results in table 6.12 show full agreement for Threat Actor, Threat Events, and Consequences. These components all received an expert mean score of 5 and a standard deviation of 0. The only component with variation is Motivation & Intent , which received an expert mean score of 4.8 and a standard deviation of 0.35. Indicating that experts still evaluated Motivation & Intent slightly differently, although the overall score is high. Compared to the baseline evaluation, the experts assessed the output more positively, particularly for Motivation & Intent and Consequences. The results indicate that the Score Maximization experiment produced the most complete and useful threat profiles.

Table 6.13 summarizes the expert mean scores across all experiments and follow a similar progression in performance as the baseline evaluation. This confirms that the improvements observed during the prompt engineering phase are also reflected in the expert evaluations.

Motivation & Intent remains the lowest scoring component across all experiments. Once the threat profiles became more elaborate due to prompt adjustments, Motivation & Intent had the highest standard deviation, compared to the other components. This indicates that this components is most difficult to evaluate and the experts interpret the results for this component differently. But despite these disagreements, they did agree on the evaluation of the other components, especially for the Score Maximization experiment. The maximum scores show that the threat profile is considered well described, and the components are considered useful. Concluding that the Score Maximization threat profiles are useful.

Experiment	Threat Actor	Motivation & Intent	Threat Events	Consequences
Zero-Shot Baseline	2	2	2.33	1.67
Framework Guided	3.4	2.6	2.8	2.6
Prompt Adjustment	4.33	3	4.33	4
Custom GPT	4	3.33	4	4
Score Maximization	5	4.8	5	5

Table 6.13: Summary of expert mean scores across all experiments. The results show a clear improvement in all components as the prompt design becomes more structured and tailored. The highest scores are achieved in the score maximization experiment, while Motivation & Intent is the lowest scoring component across all experiments.

7

Discussion

7.1. Interpretation of the results

The results suggest that ChatGPT can support the development of threat profiles, but that the quality of the generated threat profile strongly depends on the quality of the prompt. The score progression across the experiments support this claim, when the prompt described more clearly what was expected in the threat profile, the evaluation rubric scores increased. Using threat intelligence to enrich the threat profile helped make the profiles more concrete and tangible, but it did not directly result in higher evaluation scores. The technical validation became more difficult for the enriched profiles, because the generated claims became more specific and were therefore harder to confirm using the available validation data.

The LLM supports the threat profiling process by performing an analysis on all data, interpreting this data, map it to the provided framework, prioritize claims, and then generate the threat profile. This means that users mainly need to provide additional data if they want to include data that the LLM cannot access itself. After the threat profile has been generated, the user has to review and validate the output. A visual representation of this process is shown in Figure 7.1. While the LLM reduces the manual effort required for some labor-intensive steps, it also introduces a new step in the process: validating and reviewing the generated threat profile. This new step still requires expert knowledge, because the user must assess whether the claims in the threat profile are accurate and relevant.

The experiments show that prompt engineering had stronger effect on the quality of the generated frameworks than including domain specific data. The initial prompts generated generic threat profiles, but each prompt adjustment yielded better evaluation scores. The final adjustment in the Score Maximization experiment emphasized the importance of explicitly describing what is expected from the LLM. By clearly describing what a threat profile should contain and how each component in the profile should be mapped to one another, the threat profiles received near maximum evaluation scores. This observation is supported by the findings of Huang et al. [13], Shenoy and Mbaziira [36], and Priescu et al. [29] that the prompt design is a key requirement for cybersecurity tasks. Adding threat intelligence to the threat profiles provided more details which made the profiles feel less generic. Although this improved the readability of the threat profile, this was not reflected in the evaluation as the rubric mainly focused on the creation of a cohesive threat profile.

The technical validation shows that more cohesive results do not necessarily lead to more verifiable results, while the expert validation scores increased as the threat profiles became more structured and cohesive. As the prompts included more data and produced more coherent outputs, the claims made in the threat profiles also became more specific. This made some claims harder to validate using the validation data. In one case, an activity observed in the validation data was not included in the generated threat profile, because the LLM focused on creating specific threat actor profiles and the activity could not be mapped to those profiles. This suggests a trade-off between actionability and verifiability, a generic threat profile may be easier to validate because it contains broader claims, while a more specific threat profile may be more useful to experts but harder to confirm using the validation data. Therefore, the validation method should match the type of claim being assessed.

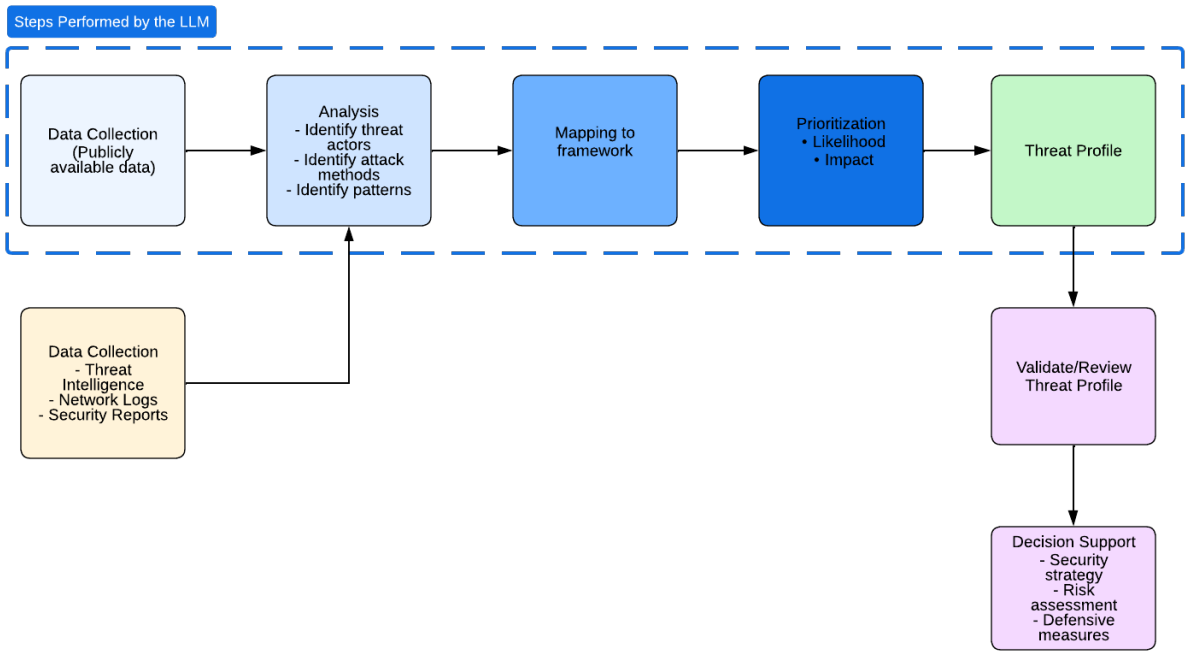

Figure 7.1: A visual representation of the threat profiling process when using an LLM.

7.2. Answers to sub-questions

In Section 4.1, three sub-question were defined to support answering the main research question. In this section, the findings of this research are summarized by answering each sub-question.

SQ1: What input data and prompts are required for an LLM to generate a threat profile?

The results show that the prompt needs to clearly describe what the generated threat profile should contain. In this research, this meant that the complete threat profiling framework had to be included in the prompt, together with all components and dimensions. The prompt also needed to explicitly describe how the components should be linked, and that prioritization and likelihood assessments should be included. The highest evaluation scores were achieved when the prompt clearly described these details.

The addition of threat intelligence was not required to generate a threat profile with high evaluation scores. The experiments show that a well structured prompt can still receive above average scores without extra data. The true value of the threat intelligence was not measured with the evaluation rubric. It helped provide more specific examples, such as threat actors, attack methods, and indicators. This makes the threat profile more concrete and helps organizations better understand the components of the threat profile.

SQ2: Which steps in the threat profiling process can be automated or supported using an LLM?

The LLM can support several steps in the threat profiling process. Data collection can be partially supported, since the LLM can search for publicly available information online. However, if internal sources such as network logs or internal threat intelligence are preferred then these should be provided by the user and they will have to create an API connection to let the LLM automatically retrieve this data.

Once the data is collected, the LLM can support the analysis of the data, the mapping of information to the threat profiling framework, the prioritization of threats, and the generation of the final threat profile. This means that several labor-intensive steps can be supported by the LLM. Once the threat profile has been generated, it is the responsibility of the user to review and validate the profile before it can be used within the organization.

SQ3: How can the generated threat profiles be validated to assess their quality and reliability?

The generated threat profiles can be validated by assessing whether the profiles are realistic and relevant, and by assessing the quality of the generated output. In this research, the relevance and realism of the threat profiles were validated using security monitoring data. This enabled to check whether generated claims occurred in the validation data. However, the validation also showed that more data sources are required to perform a complete technical validation.

The technical validation demonstrated that different components of a threat profile require different types of evidence, one validation source is not sufficient. Threat events can be validated using sources such as network traffic data, system logs, SOC data, and security portal alerts. Threat actors are more difficult to validate and may require incident reports, threat intelligence, or attribution information. Motivation and intent often require inference based on the type of actor, the attack method, and the target. Consequences may require additional context, such as business impact information, regulatory requirements, or previous incident consequences.

The quality of the generated threat profiles can be validated using an evaluation rubric. In this research, the rubric assessed whether the profiles were complete, structured, and cohesive. However, future evaluations should also include the applicability and usefulness of the profile for an organization. This would make it possible to assess not only whether the threat profile is well-structured, but also whether it is practically useful for decision-making.

7.3. Future Work

One of the main suggestions for future work is testing the different models and modes of ChatGPT. The Pro model, extended thinking mode, and deep research mode enable the LLM to do online research. This could reduce the information collection phase even further by letting the LLM perform part of this work. These modes are also more advanced and take more time to generate answers, which could result in better threat profiles. Another suggestion for future work is to compare different LLMs. For this research, a license to ChatGPT was provided, but there are many more LLMs available. By testing the same prompts in different LLMs, it could be determined which LLM is most suitable for this task.

This work focused on a threat profiling framework that was inspired by a combination of existing frameworks. Future work could investigate the quality of generated threat profiles when focusing on a specific industry standard. Finally, this work could be expanded by focusing on threat modeling. The threat profile provides a solid basis for a threat model. By including information about an organization's infrastructure and assets, the LLM could also start supporting the threat modeling process.

7.4. Reflection

This research shows that using LLMs for threat profiling requires a balance between providing the right data and creating the proper prompt for the task. The LLM should not be expected to know how to perform the task independently. Instead, the complete task should be described clearly. Therefore, before using the LLM, it is important to have a clear plan of what the output should contain, how the task should be performed, and what level of detail is expected. This paper offers a structured approach for using an LLM for a task that has not yet been widely explored. And it can also help in the planning stage of other underexplored domain specific tasks. Beyond the technical implementation, this research shows how human expertise, structured task design, and domain specific knowledge need to be combined to make LLMs useful in practice, regardless of the domain.

This reflects a broader trend within computer science, where LLMs are used to support complex analytical tasks. In this shift, the role of the user changes from manually performing every step, to designing the task, providing the right context, and validating the output. In this way, LLMs can help reduce the

manual effort required to complete a task. This can be especially useful in sectors where there is a staff shortage. While LLMs do not replace people, it does change their role. People become responsible for providing the right context to the LLM and reviewing the output, instead of doing all the work themselves.

7.5. Limitations

The most prominent limitation is that all experiments were based on the zero-shot training method. This was necessary because threat profiles are not publicly available, at least not to the extent known in this research. As a result, the LLM could not be provided with examples of what a threat profile should look like. This means that the model had no direct way of learning what was expected from the output. Therefore the only way to generate threat profiles was through zero-shot training and improving the profiles could only be done through the prompt engineering cycle, in which the prompt was adjusted based on the evaluation results. This made the quality of the generated threat profiles dependent on the quality of the prompt.

For the domain specific data, this research depended on open-source threat intelligence and MISP. MISP is a community based platform, which means that the available data in MISP depends on the contributions from the community. Since there are no strict standards or universally used definitions, fields and tags were used inconsistently across the data. This made filtering the threat intelligence more difficult, as all the data had to be analyzed to ensure that the selected filters did not exclude relevant values or fields. However, because tags and labels were not used consistently, relevant data may still have been missed if it was not properly tagged or submitted. Furthermore, the use of an open-source threat intelligence feed introduced a lot of noise. This was especially visible in the threat attributes, as the threat feed contained a large amount of scanner related information. As a result, this type of information may have outweighed other relevant attributes, which could have influenced the generated threat profiles.

Another limitation is related to the dependency of the LLM on the prompt design. In this research, the prompt structure was adjusted in several experiments. These changes has a significant effect on the generated threat profiles, showing that the output depends strongly on the prompt that is used. More elaborate prompts improved the structure and completeness of the threat profile, but they may also introduce bias. By providing exactly what information should be included, how the components should be mapped, and how the output should be structured, the LLM is guided toward generating the type of profile that is expected by the researcher. While this improved the evaluation scores, it also meant that the output partly reflected the assumptions embedded in the prompt rather than only the patterns present in the data. Another limitation regarding the LLM applies to the data it receives. When providing threat intelligence to the LLM, it may focus strongly on that data and shape the profile around it. Even when instructing to only use it for enrichment, as a result, patterns that are not reflected in the provided data may may be overlooked.

The evaluation rubric was created with the idea of generating a complete and cohesive profile. However, it did not take into account the applicability or organizational usefulness of the threat profile. This means that a profile could receive a high score when the components were clearly described and mapped, even if the practical value for an organization was low. Another limitation is that the initial evaluation was performed by a single person. Therefore, the scores reflect the interpretation of one evaluator and are biased towards their views. The expert based evaluation helped reduce this limitation, but it was only performed by three experts. While this provides more perspectives than the original evaluation, the mean and standard deviation are still sensitive to the individual scores given by each expert.

Finally, the technical validation was based on only one type of data. Consequently, many claims could not be assessed, because the available validation data did not provide enough evidence for all components of the threat profile. This strongly affected the technical validation results, as several claims were classified as *Not Assessable* rather than *Supported* or *Partially Supported*. A broader variety of data sources would therefore be needed to perform a more complete technical validation.

7.6. Recommendations

When using an LLM for threat profiling, it should be used as a supportive tool and not as the main decision maker in the threat profiling process. To ensure successful use of the LLM, planning is the most important step. A clearly defined framework should be available and translated into explicit instructions that that guide the LLM toward creating the preferred threat profile. During the planning stage, it should be determined how each component can be verified and which data sources are required for this validation. Moreover, the LLM should include an evidence label for each claim it makes, this makes it easier to review and validate the generated threat profiles. When providing threat intelligence to the LLM, the data should be thoroughly filtered and normalized to prevent events or attributes to be overrepresented.

Creating a custom GPT for this process, with automatic data retrieval, may take some time during the initial setup. Once the setup is complete, generating a threat profile can be done with a short prompt to activate the custom GPT. This makes the process easier in the long term and makes keeping threat profiles up to date more manageable.

To support the practical application of this approach, Appendix B provides a general prompt that can be used as a starting point for creating LLM-generated threat profiles.

8

Conclusion

This research shows that LLMs can support the development of threat profiles by performing several steps in the threat profiling process. When the LLM is provided with a clear prompt, a predefined framework, and relevant input data, it can analyze the available information, map it to the framework, prioritize threats, and generate a coherent threat profile. Threat intelligence is not strictly required to generate a threat profile, but it helps make the profile more concrete and easier to understand. The generated threat profiles can be validated through a combination of technical validation and expert-based validation. Technical validation can assess whether claims are reflected in available security data, while expert-based validation can assess the structure, completeness, and usefulness of the profile. However, different components require different types of evidence, meaning that one type of security monitoring data is not sufficient for a complete validation.

Overall, using an LLM can reduce the effort required to create threat profiles and make the process more manageable for organizations. Instead of manually performing the full threat profiling process, the user's role shifts toward setting up the LLM for success by defining the framework, providing relevant data, and writing clear instructions. Once the LLM has been set up correctly, the same configuration can be reused to generate new or updated threat profiles with less effort. This streamlines the threat profiling process and makes it easier to keep threat profiles up to date as the threat landscape changes. This research highlights a broader shift within computer science, while LLMs do not replace people, it does change their role. People become responsible for defining the task and reviewing the output, while the LLM performs the task for them.

9

Acknowledgement

In this thesis AI tools were used to support the research and writing process.

For literature search support, SciSpace (https://scispace.com/) and Elicit (https://elicit.com/) were used to help find relevant papers.

ChatGPT (https://chatgpt.com/) was used in several ways throughout this thesis. First, it was used as part of the research itself. ChatGPT and a custom GPT were used to generate threat profiles, which were then evaluated by the researcher. The generated outputs served as research material for analyzing how LLMs can be used in the threat profiling process. Moreover, ChatGPT was used as writing support, including formatting assistance, paraphrasing, clarification, and language support. Finally, ChatGPT was used to generate the cover image of this thesis.

While AI has been used in this research, all final decisions, interpretations, analyses, and conclusions remain the responsibility of the author.

References

[1] Md Ahmad and Haripriya V. “The Role of Threat Intelligence in Enhancing Cybersecurity Posture”. In: *International Journal of Innovative Research in Computer and Communication Engineering* 12 (Mar. 2024), pp. 1739–1746. DOI: 10.15680/IJIRCCE.2024.1203061.

[2] Amazon Web Services. *What Are Transformers in Artificial Intelligence?* Amazon Web Services. n.d. URL: https://aws.amazon.com/what-is/transformers-in-artificial-intelligence/ (visited on 01/05/2026).

[3] Anthropic. *Claude AI*. Anthropic. n.d. URL: https://claude.ai/ (visited on 01/05/2026).

[4] Kurt Baker. *What is Cyber Threat Intelligence? [Beginner’s Guide]*. Accessed: 2026-04-12. Mar. 2025. URL: https://www.crowdstrike.com/en-us/cybersecurity-101/threat-intelligence/.

[5] Deborah J. Bodeau, Catherine D. McCollum, and David B. Fox. *Cyber Threat Modeling: Survey, Assessment, and Representative Framework*. Tech. rep. MTR180151. The MITRE Corporation, Apr. 2018. URL: https://www.mitre.org/sites/default/files/publications/pr-18-1174-cyber-threat-modeling.pdf.

[6] CIRCL. *CIRCL OSINT Feed*. MISP OSINT feed. URL: https://www.circl.lu/doc/misp/feed-osint/.

[7] Docker, Inc. *Docker*. Docker. n.d. URL: https://www.docker.com/ (visited on 01/14/2026).

[8] Isra Elsharef, Zhen Zeng, and Zhongshu Gu. “Facilitating Threat Modeling by Leveraging Large Language Models”. In: *Proceedings of the Workshop on AI Systems with Confidential Computing (AISCC 2024)*. San Diego, CA, USA, Feb. 26, 2024. ISBN: 979-8-9894372-4-5. DOI: 10.14722/aiscc.2024.23016. URL: https://www.ndss-symposium.org/wp-content/uploads/aiscc2024-16-paper.pdf.

[9] Muhammed Erbas et al. “Integrating Cyber Threat Intelligence into Threat Modeling for Autonomous Ships Using PASTA and MISP”. In: *2025 IEEE International Conference on Cyber Security and Resilience (CSR)*. 2025, pp. 133–139. DOI: 10.1109/CSR64739.2025.11130108.

[10] Google. *Gemini*. Google. n.d. URL: https://gemini.google.com/app (visited on 01/05/2026).

[11] Michael Guastalla et al. “Application of Large Language Models to DDoS Attack Detection”. In: *EAI International Conference on Security and Privacy in Cyber-Physical Systems and Smart Vehicles (SmartSP)*. 2023. URL: https://anrg.usc.edu/www/papers/LLM_Cybersecurity_SmartSP.pdf.

[12] Paul R. B. Houssel et al. “Towards Explainable Network Intrusion Detection using Large Language Models”. In: *2024 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*. IEEE, Dec. 2024, pp. 67–72. DOI: 10.1109/bdcat63179.2024.00021. URL: http://dx.doi.org/10.1109/BDCAT63179.2024.00021.

[13] Ken Huang et al. “Utilizing Prompt Engineering to Operationalize Cybersecurity”. In: *Generative AI Security: Theories and Practices*. Ed. by Ken Huang et al. Cham: Springer Nature Switzerland, 2024, pp. 271–303. ISBN: 978-3-031-54252-7. DOI: 10.1007/978-3-031-54252-7_9. URL: https://doi.org/10.1007/978-3-031-54252-7_9.

[14] IBM. *What Are AI Hallucinations?* IBM Think. 2026. URL: https://www.ibm.com/think/topics/ai-hallucinations (visited on 01/05/2026).

[15] Michael Irwin. *Creating a Threat Profile for an Organization*. GIAC Gold Certification Paper. SANS Institute, 2014. URL: https://www.giac.org/paper/gcih/1772/creating-threat-profile-organization/110995.

[16] Shaymaa Mamdouh Khalil, Hayretdin Bahsi, and Tarmo Korõtko. “Threat modeling of industrial control systems: A systematic literature review”. In: *Computers Security* 136 (2024), p. 103543. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2023.103543. URL: https://www.sciencedirect.com/science/article/pii/S0167404823004534.

[17] Microsoft. *Microsoft Threat Modeling Tool threats*. https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats. Last updated: 2022-08-25. Accessed: 2026-04-08.

[18] Microsoft. *What is SIEM?* https://www.microsoft.com/en-us/security/business/security-101/what-is-siem. Accessed: 2026-04-08. 2026.

[19] Microsoft Azure. *What are Large Language Models (LLMs)?* Microsoft. n.d. URL: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-are-large-language-models-llms (visited on 01/05/2026).

[20] MISP Project. *MISP Data Models*. Accessed: 2026-01-24. 2026. URL: https://www.misp-project.org/datamodels/.

[21] MISP Project. *MISP Galaxy: Threat Actor*. Accessed: 2026-10-24. MISP Project. n.d. URL: https://misp-galaxy.org/threat-actor/.

[22] MISP Project. *MISP: Open Source Threat Intelligence Platform and Sharing System*. https://www.misp-project.org/. Accessed: 2026-01-07. 2026.

[23] Dietmar P. F. Möller. “Attack Models and Scenarios”. In: *Cybersecurity in Digital Transformation: Scope and Applications*. Cham: Springer International Publishing, 2020, pp. 89–98. ISBN: 978-3-030-60570-4. DOI: 10.1007/978-3-030-60570-4_6. URL: https://doi.org/10.1007/978-3-030-60570-4_6.

[24] National Institute of Standards and Technology. *NIST SP 800-30 Rev. 1: Guide for Conducting Risk Assessments*. Accessed: 2026-04-08. 2012. URL: https://csrc.nist.gov/pubs/sp/800/30/r1/final.

[25] ngrok, Inc. *ngrok*. ngrok. n.d. URL: https://ngrok.com/ (visited on 01/14/2026).

[26] OpenAI. *ChatGPT Overview*. https://chatgpt.com/overview/. Accessed: 2026-01-06. 2026.

[27] OpenAI. *Creating and editing GPTs*. Accessed: 2026-01-10. 2026. URL: https://help.openai.com/en/articles/8554397-creating-and-editing-gpts.

[28] Luka Podlesnik, Igor Bernik, and Anže Mihelič. “Integrating CTI and threat modeling for cyber resilience: An AHP assessment”. In: *PLOS ONE* 20.11 (Nov. 2025), pp. 1–16. DOI: 10.1371/journal.pone.0335154. URL: https://doi.org/10.1371/journal.pone.0335154.

[29] Iustin Priescu et al. “Prompt Engineering in Cybersecurity – Achieving Technological Edge”. In: *Land Forces Academy Review* 30 (June 2025), pp. 291–302. DOI: 10.2478/raft-2025-0028.

[30] Marcelo Rodríguez, Gustavo Betarte, and Daniel Calegari. “A process mining-based method for attacker profiling using the MITRE ATTCK taxonomy”. In: *Journal of Internet Services and Applications* 15.1 (Aug. 2024), pp. 212–232. DOI: 10.5753/jisa.2024.3902.

[31] S. Saeed et al. “A Systematic Literature Review on Cyber Threat Intelligence for Organizational Cybersecurity Resilience”. In: *Sensors* 23.16 (2023), p. 7273. DOI: 10.3390/s23167273.

[32] Md Sahrom Abu et al. “Cyber Threat Intelligence – Issue and Challenges”. In: *Indonesian Journal of Electrical Engineering and Computer Science* 10 (Apr. 2018), pp. 371–379. DOI: 10.11591/ijeecs.v10.i1.pp371-379.

[33] Sebastián Ramírez. *FastAPI*. FastAPI. n.d. URL: https://fastapi.tiangolo.com/ (visited on 01/14/2026).

[34] Sawera Shahid, Umara Noor, and Zahid Rashid. *An Unsupervised Learning Approach For A Reliable Profiling Of Cyber Threat Actors Reported Globally Based On Complete Contextual Information Of Cyber Attacks*. 2025. arXiv: 2509.11683 [cs.CR]. URL: https://arxiv.org/abs/2509.11683.

[35] Gaurav Sharma et al. “Analysis and Implementation of Threat Agents Profiles in Semi-Automated Manner for a Network Traffic in Real-Time Information Environment”. In: *Electronics* 10.15 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10151849. URL: https://www.mdpi.com/2079-9292/10/15/1849.

[36] Neethu Shenoy and Alex V Mbaziira. “An Extended Review: LLM Prompt Engineering in Cyber Defense”. In: *2024 International Conference on Electrical, Computer and Energy Technologies (ICECET*. 2024, pp. 1–6. DOI: 10.1109/ICECET61485.2024.10698605.

[37] The MITRE Corporation. *MITRE ATT&CK*. https://attack.mitre.org/. Accessed: 2026-04-08.

[38] *Threat Profile - an overview*. https://www.sciencedirect.com/topics/computer-science/threat-profile. Accessed: 2026-04-01.

[39] Anna Wimbauer et al. “ThreatCompute: Leveraging LLMs for Automated Threat Modeling of Cloud-Native Applications”. In: *Proceedings of the 2025 Cloud Computing Security Workshop*. CCSW ’25. Association for Computing Machinery, 2025, pp. 14–27. ISBN: 9798400719011. DOI: 10.1145/3733812.3765533. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3733812.3765533.

[40] Tingmin Wu et al. *ThreatModeling-LLM: Automating Threat Modeling using Large Language Models for Banking System*. 2025. arXiv: 2411.17058 [cs.CR]. URL: https://arxiv.org/abs/2411.17058.

[41] Wenjun Xiong and Robert Lagerström. “Threat modeling – A systematic literature review”. In: *Computers Security* 84 (2019), pp. 53–69. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2019.03.010. URL: https://www.sciencedirect.com/science/article/pii/S0167404818307478.

A

Threat Intelligence Fields

A.1. Threat Actor fields

Below is a list of the fields used for the Threat Actor galaxy, and includes a definition of what each field means.

- **description:** A description of the threat actor.
- **country:** The country associated with the threat actor.
- **synonyms:** Alternative names, aliases, or labels used to refer to the threat actor.
- **motivations:** The known or assessed motivations of the threat actor.
- **targeted-sector:** A sector or industry reportedly targeted by the threat actor.
- **targeted-sectors:** A list of sectors or industries reportedly targeted by the threat actor.
- **targeted-countries:** A list of countries or regions reportedly targeted by the threat actor.
- **attribution-confidence:** A numerical indication of confidence in the attribution of the threat actor, on a scale from 0 to 100.
- **cfr-suspected-state-sponsor:** The suspected state sponsor of the threat actor according to the Council on Foreign Relations Cyber Operations Tracker.
- **cfr-suspected-victims:** The suspected victims or victim countries associated with the threat actor according to the Council on Foreign Relations Cyber Operations Tracker.
- **cfr-target-category:** The category of targets associated with the threat actor according to the Council on Foreign Relations Cyber Operations Tracker.
- **cfr-type-of-incident:** The type of cyber incident associated with the threat actor according to the Council on Foreign Relations Cyber Operations Tracker.
- **goals:** The high level objectives of the threat actor, describing what the actor is trying to achieve.
- **resource-level:** The general level of resources available to the threat actor
- **primary-motivation:** The main reason or purpose behind the threat actor's activity.
- **secondary-motivation:** Additional reasons or purposes behind the threat actor's activity.

A.2. Threat Event fields

Below is a list of the fields used in the threat events, and includes a short description of each field.

- **date:** The date on which the event occurred.
- **info:** A summary of the event.
- **tags:** Labels used to classify or contextualise the event.
- **galaxies:** Structured contextual information attached to the event.

A.3. Threat Attribution fields

Below is a list of the fields used in the attributes, and includes a short description of each field.

- **category:** The classification of the attribute.
- **type:** The data type of the attribute.
- **value:** The value of the attribute.

B

General Prompt

The following prompt provides a general starting point for creating LLM-generated threat profiles. It should be adapted to the specific threat intelligence framework, available data sources, and validation requirements of the organization.

```
1
2  You are a threat intelligence analyst specializing in the financial sector.
3  Generate a threat profile for a financial institution.
4
5  Goal:
6  Produce an output that scores as highly as possible on these four components:•
7          Threat Actors•
8          Motivation & Intent•
9          Threat Events•
10         Consequences
11 Use only widely observed, sector-relevant patterns. If a detail is unknown, write
       ""Unknown. Do not invent named groups unless they are widely known exemplars.
12 Scoring requirements:
13 For every component, provide:•
14         concrete, sector-relevant detail•
15         explicit prioritization•
16         explicit likelihood assessment•
17         explicit linkage to the other components through bracketed IDs in the
               narrative
18 Use this exact structure.
19 PART A - Reference Components
20 A1) Threat Actors (TA)
21 Provide -36 entries, ordered from highest priority to lowest priority.
22 Format:
23 [TA#] Actor label - Category | Sophistication: Low/Medium/High | Priority: Primary/
       Secondary/Tertiary | Likelihood: Rare/Occasional/Frequent | Notable group(s):
       ... / Unknown
24 A2) Motivation & Intent (M)
25 Provide -48 entries, ordered from highest priority to lowest priority.
26 Format:
27 [M#] Goal: ... | Desired cyber effect: ... | Priority: Primary/Secondary/Tertiary
       | Likelihood: Rare/Occasional/Frequent
28 A3) Threat Events (TE)
29 Provide -612 entries, ordered from highest priority to lowest priority.
30 Format:
31 [TE#] Event type: ... | TTPs/IOCs (high-level only): ... | Risk: Low/Medium/High |
        Priority: Primary/Secondary/Tertiary | Likelihood: Rare/Occasional/Frequent
32 A4) Consequences (C)
33 Provide -612 entries, ordered from highest priority to lowest priority.
```

```
Format:
[C#] Impact type: ... | Severity: Low/Medium/High | Priority: Primary/Secondary/Tertiary | Likelihood: Rare/Occasional/Frequent
Rules for Part A:•
        Keep entries compact and scannable.•
        Define items once only.•
        Every entry must be concrete enough to stand alone.•
        Do not use mapping tables.
PART B - Coherent Threat Profile
Use bracketed IDs from Part A as citations only.
B1) Executive Summary
Write ~58 bullets.
Each bullet must include at least 3 component types.
At least half of the bullets must include all 4 component types: [TA#][M#][TE#][C#].
Explicitly identify:•
        top 2 threat actors•
        top 2 motivations•
        top 3 threat events•
        top 3 consequences
B2) Threat Landscape
Write ~24 short paragraphs explaining why the financial institutions are targeted and what common targeting patterns look like.
Cite relevant items.
Make clear how sector characteristics influence both likely events and likely consequences.
B3) Actor Profiles
For each [TA#], write one short paragraph covering:•
        who they are and capability [TA#]•
        what they want [M#]•
        how they operate [TE#]•
        what happens if they succeed [C#]
Rule:
Each paragraph must cite at least one item from all four component types.
B4) Common Attack Chains
Provide ~46 concise prose bullets.
Each bullet must include at least:•
        one actor [TA#]•
        one motivation [M#]•
        one event [TE#]•
        one consequence [C#]
B5) Impact & Consequence Summary
Group consequences by theme:•
        financial•
        operational•
        customer harm•
        regulatory•
        strategic
Cite [C#] items.
Distinguish common consequences from less frequent but high-severity consequences.
Global rules:•
        In Part B, use IDs only as bracketed citations.•
        Avoid redundancy: Part A defines, Part B cites.•
        Keep the output professional, clear, and readable.•
        Motivations must be operationally meaningful, not just abstract labels.•
        Consequences must be specific financial-sector impact patterns, not just generic categories.•
        Do not imply priority or likelihood; state them explicitly.
Final self-check before answering:•
        every Part A section is ordered by priority•
        every Part A entry includes explicit priority•
```

```
        every Part A entry includes explicit likelihood•
        every Threat Actor in Part B is linked to at least one Motivation, one
            Threat Event, and one Consequence•
        every attack chain links actor, motivation, event, and consequence•
        motivations are concrete and sector-relevant•
        consequences are concrete and sector-relevant•
        the response is optimized for the highest possible score on all four
            rubric components

------
PHASE 1 - Start enrichment only
- Prefer the multi-job actions.
- If the user requests a subset of enrichment types, call:
  start_jobs_jobs_start_post
  with a JSON body like:
  { "jobs": ["actors"] }
  or
  { "jobs": ["actors", "events"] }
  or any valid subset of:
  ["actors", "events", "attributes"]
- If the user requests all enrichment types, call:
  start_all_jobs_jobs_start_all_post
- Save the returned job IDs by type.
- Show the started job IDs clearly.
- Then ask:"
  MISP enrichment jobs started. Do you want me to compute the threat profile now,
      or should I check later? Reply: ‘compute ’now or ‘check ’later".
- Stop there. Do not generate the report in this phase.

PHASE 2 - Compute only on explicit request
Only when the user replies "compute "now:

1. First call:
   check_jobs_status_jobs_status_post
   using the saved job IDs from this conversation.

2. If none of the selected jobs are complete:
- Do not keep polling in the same response.
- Report the current status.
- Ask the user to reply "compute "now again when they want a re-check.
- Do not generate the report yet.

3. If one or more jobs are complete:
- Generate the report.
- Retrieve results one job type at a time, only when needed for the section being
    written.
- Do NOT request all result types in one get_jobs_results call unless only one job
     type was started.

SECTION-BY-SECTION RESULT RETRIEVAL RULE
When writing the report, call get_jobs_results_jobs_results_post separately for
    each completed job type:

- For Threat Actors and Motivation/Intent sections:
  call get_jobs_results_jobs_results_post with only:
  { "job_ids": { "actors": "<actors_job_id>" } }
  if the actors job is complete.

- For Threat Events and attack-chain sections:
  call get_jobs_results_jobs_results_post with only:
  { "job_ids": { "events": "<events_job_id>" } }
```

```
  if the events job is complete.

- For attributes-based enrichment:
  call get_jobs_results_jobs_results_post with only:
  { "job_ids": { "attributes": "<attributes_job_id>" } }
  if the attributes job is complete.

- Never request actors, events, and attributes results together in one results
    call unless there is a clear reason and the response will stay small.
```