Threat modeling state of practice in Dutch organizations

Stef Verreydt, Koen Yskout, Laurens Sion,
and Wouter Joosen, *DistriNet, KU Leuven*

https://www.usenix.org/conference/soups2024/presentation/verreydt

This paper is included in the Proceedings of the
Twentieth Symposium on Usable Privacy and Security.

August 12–13, 2024 • Philadelphia, PA, USA

978-1-939133-42-7

Threat modeling state of practice in Dutch organizations

Stef Verreydt
DistriNet, KU Leuven
3001 Leuven, Belgium

Koen Yskout
DistriNet, KU Leuven
3001 Leuven, Belgium

Laurens Sion
DistriNet, KU Leuven
3001 Leuven, Belgium

Wouter Joosen
DistriNet, KU Leuven
3001 Leuven, Belgium

Abstract

Threat modeling is a key technique to apply a *security by design* mindset, allowing the systematic identification of security and privacy threats based on design-level abstractions of a system. Despite threat modeling being a best practice, there are few studies analyzing its application in practice. This paper investigates the state of practice on threat modeling in large Dutch organizations through semi-structured interviews.

Compared to related work, which mainly addresses the execution of threat modeling activities, our findings reveal multiple human and organizational factors which significantly impact the embedding of threat modeling within organizations. First, while threat modeling is appreciated for its ability to uncover threats, it is also recognized as an important activity for raising security awareness among developers. Second, leveraging developers' intrinsic motivation is considered more important than enforcing threat modeling as a compliance requirement. Third, organizations face numerous challenges related to threat modeling, such as managing the scope, obtaining relevant architectural documentation, scaling, and systematically following up on the results. Organizations can use these findings to assess their current threat modeling activities, and help inform decisions to start, extend, or reorient them. Furthermore, threat modeling facilitators and researchers may base future efforts on the challenges identified in this study.

1 Introduction

Many security-enhancing activities can be performed during software development, ranging from training and the security requirements specification over source code analysis to pentesting and incident response handling [14]. One of these activities, and the focus of this research, is *threat modeling*.

Threat modeling is widely promoted as a best practice for secure software development. For example, it plays a prominent role in Microsoft's Security Development Lifecycle [14], OWASP's Software Assurance Maturity Model [19], NIST's Secure Software Development Framework [16], and others. Moreover, insecure design, for which threat modeling is considered a key mitigation strategy, appears in the fourth place in the most recent (2021) edition of the OWASP Top 10 [17].

In the words of the 'Threat Modeling Manifesto' [3], and in alignment with the 'four questions' framework of Shostack [24], "*threat modeling is analyzing representations of a system to highlight concerns about security and privacy characteristics. [This involves] four key questions: 1) What are we working on? 2) What can go wrong? 3) What are we going to do about it? and 4) Did we do a good enough job?*". Numerous threat modeling approaches exist (e.g., STRIDE [24], PASTA [35], CVSS [12], attack trees [22]), as well as several supporting tools (e.g., Microsoft Threat Modeling Tool [15], IriusRisk [10], pytm [32], OWASP Threat Dragon [18]) to (partially) automate threat modeling analyses.

Current empirical research (Section 5) mostly focuses on how threat modeling is applied by practitioners, identifying best practices and challenges related to specific threat modeling methodologies, tools, and application domains. Few studies, however, investigate scheduling, stakeholder involvement, frequency, organization introduction, etc, yet such non-technical aspects affect the overall effectiveness of threat modeling. Through semi-structured interviews with practitioners from large Dutch organizations in critical sectors that are part of the target audience of the Dutch National Cyber Security Center (NCSC, the sponsor of this research), the goal of this research is therefore to provide qualitative insights into the state of practice on threat modeling, paying particular attention to the non-technical aspects of threat modeling. The results of this study can be used by other organizations to assess their current practices, and help inform decisions to start, extend,

USENIX Symposium on Usable Privacy and Security (SOUPS) 2024.
August 11–13, 2024, Philadelphia, PA, United States.

or reorient existing threat modeling programs.

The remainder of this paper is structured as follows. Section 2 explains the research methodology. Section 3 answers the research questions based on observations from the interviews. Section 4 discusses the implications of this study's results and the limitations of this study. Section 5 provides an overview of related work and relates our observations to those of similar studies. Finally, Section 6 concludes the paper.

Table 1: Overview of the organizations

Sector	Focus	Participants
Energy	OT Systems	1
Finance	Software development	4
Marine	IT Infrastructure	1
Public sector	Software development, advice	3
Transport	Software development	4

2 Methodology

2.1 Goal and scope

The insights in this paper originate from a set of interviews with practitioners from large, Dutch organizations, conducted between August 2022 and February 2023. The sponsor of this research (NCSC) is a government organization that provides security advice to large organizations in critical sectors, and the interviewees are employees of those large organizations. We focus primarily on organizations that have in-house software development teams, but also include organizations without such teams yet focusing on Information Technology (IT) and Operational Technology (OT) infrastructure, as well as one organization that has an advisory role.

Our assessment of the state of practice addresses four broad research questions: **RQ1**: How is threat modeling embedded in the organization? **RQ2**: Which organizational roles are involved in threat modeling activities? **RQ3**: How is threat modeling performed within the organization? **RQ4**: What are the experiences with threat modeling within the organization? These research questions were determined in collaboration with the sponsor, ensuring that the study addressed relevant and meaningful aspects of the subject matter. However, the subsequent research was conducted independently, safeguarding the objectivity and impartiality of the findings and conclusions. The sponsor was provided with a report of the study findings [29], on which this paper is based.

The goal of this research is to provide qualitative insights into the state of practice of threat modeling. Hence, this paper refrains from using precise numbers or percentages when discussing observations, as they would give a false impression of accuracy due to the limited number of interviewees. Future research may aim for a quantitative characterization of the state of practice through a larger set of interviewees.

2.2 Study design

The interview guide [28] was constructed based on the research questions (see Section 2.1) and lists the different topics to discuss during the interviews in the form of questions. The interviews themselves were performed using the technique of responsive interviewing [20], allowing the interviewers to delve into more detail when appropriate. This means that the questions from the interview guide were not asked literally nor sequentially; interviews took the form of a natural conversation, merely guided by the topics to discuss.

Ethical approval for this research was obtained from KU Leuven's ethical committee[1] before potential participants were contacted. All interviewed participants have signed an informed consent form. After an interview, each participant was offered a 20 euro gift voucher for their participation.

2.3 Recruitment process

Given the study's focus on organizations that are part of the target audience of the sponsor (NCSC), the sponsor provided a list of contacts at the relevant organizations to reach out to. As the goal of our research was to gain insight into the state of practice, we informed these contacts that potential interviewees should be directly involved in threat modeling. All contacts received from the sponsor agreed to an interview and/or provided other contacts within their organization; we unfortunately have no information on contacts that the sponsor approached and declined, or their reasons. Potential participants with the relevant expertise were provided with an informed consent form [27] and information sheet [26] which described, among others, the goal of the study, the methods used, information on voluntary participation and withdrawal, compensation, potential risks, confidentiality, data processing, and contact information of the researchers.

In total, 13 participants from 7 organizations agreed to participate, resulting in 10 interviews (in three interviews, two participants were interviewed at the same time). Each organization has thousands of employees, and all participants have a role dedicated to security. General characteristics of the participants and organizations are provided in Table 1.

2.4 Data collection process

Two interviews were conducted at the participant's offices; the others were conducted online (through a video call). The interviews were performed by the authors (KY, LS, and SV). One researcher took the lead during the interview. For early interviews, other researchers observed and took notes for consistency with later ones. Each interview lasted approximately

[1] KU Leuven Social and Societal Ethics Committee (SMEC), case ID G-2021-4578-R2

one hour, except for joint interviews (approx. 90 minutes). All interviews were in Dutch, except one (in English).

After a brief introduction and repeating the agreements on confidentiality and data protection, the remainder of the interview was recorded (using a microphone for on-site interviews, and the built-in Teams functionality for online interviews). During or after the interview, some participants also briefly showed reports of threat models on which they have worked.

Afterwards, the recordings were transcribed using automated transcription, the results of which were subsequently checked and corrected using the original recording. Automated transcription was initially performed using the built-in functionality of Microsoft Word; for later interviews, a fully offline implementation of Whisper [8] was used.

The interview transcripts were subsequently anonymized manually, by replacing or scrubbing all information that would enable identification of the participant or the organization. All participants received a copy of the anonymized transcript of their interview, and had the opportunity to add remarks, provide corrections, or highlight potentially identifying information. Two participants explicitly confirmed that the information in the transcript was still accurate; one other participant clarified changes in the organization that occurred after the interview. All copies of recordings, non-anonymized transcripts, and notes are destroyed at completion of the research study.

2.5 Analysis procedure

The analysis of the research data involved a systematic coding process of the anonymized transcripts. To facilitate organized coding, a software package for qualitative data analysis (ATLAS.ti) was employed. The coding process used a mix of bottom-up and top-down codes, allowing themes and patterns to emerge from the raw data [5]. Initial (top-down) codes were generated in alignment with the research questions (for example, related to demographics, process/execution, etc.), augmented with the researchers' recollections from reading through and anonymizing the transcripts. Throughout the coding process, these codes were complemented with (bottom-up) codes that capture significant other concepts, ideas, or phrases relevant to the research questions. After coding, codes with similar meanings or concepts were grouped into higher-level codes. The complete codebook can be found online [25].

Based on the coded transcripts, recurring themes and challenges were identified, for which quotations were collected. For interviews that were conducted in Dutch, the quotations used in this paper were translated to English. These quotations served as evidence to support the findings and conclusions drawn from the analysis. A single researcher (SV) was in charge of coding,[2] identifying themes among the coded transcripts was done by multiple researchers (KY, LS, and SV).

3 Results

This section answers the research questions based on the data collected through the interviews.

3.1 Embedding of threat modeling activities

The first research question concerns the embedding of threat modeling in organizations. This is split up into three subquestions regarding (1) the definition and perceived benefits, (2) the organizational motivation, and (3) using the results.

RQ1.1. Why do organizations threat model?

All participants agree that threat modeling is an important analysis activity in the development process. Participants frequently mention using threat modeling to analyze and map threats, vulnerabilities, or risks; combined with thinking about potential countermeasures, although some participants note the limited support in this regard. Less frequently mentioned aspects of threat modeling include the importance of considering particular threat actors (*"know your enemy"*), explicitly thinking about key assets (*"what do we want to protect?"*), abuse cases (*"next to the use cases, to also define abuse cases [...] and think what could go wrong in the flow"*), and the supporting role of threat modeling in subsequent activities such as pentesting (*"it is also an excellent basis for [a pentest]"*).

Threat modeling is, however, not always explicitly labeled or systematically executed, and several participants mention security practices being performed in their organization which closely resemble threat modeling without being labeled as such (*"[security practices happen] a lot, but not structurally and not under the umbrella of threat modeling"*).

The main benefits organizations perceive are twofold. First, threat modeling is employed to gain understanding and insight into an application's security concerns (*"to develop more secure products"*). Second, it is also a useful technique to raise the overall security awareness of teams (*"they learn to think about threats"*), and to give the teams a way to talk about security (*"a way for them to discuss information security in a practical way within their team"*). A third goal mentioned by one of the participants is to use threat models as a communication tool for security with non-technical people (*"so that [non-IT] people also get a good understanding of how certain things can occur"*).

RQ1.2. How are stakeholders motivated to threat model?

There is a strong focus on promoting threat modeling internally as a technique for the development teams to apply, rather than mandating threat modeling through organizational policies. Awareness measures range from simply mentioning the technique (*"tell them once, let it simmer"*) to organizing internal workshops (*"so that people at least understand what threat modeling is and why we do it"*).

[2] While we agree that multiple coders would improve the reliability, this is not a crucial aspect of a qualitative surveys according to the ACM SIGSOFT Empirical Standards for Software Engineering [1].

Most participants stress that development teams should internally recognize the usefulness of threat modeling. This ensures that the motivation comes from within the team (intrinsic motivation) rather than being imposed (extrinsic motivation) (*"The initiative to do [threat modeling] should come from the developers. [...] the moment you start forcing threat modeling, people naturally lose enthusiasm and do it because they have to and not because they see the usefulness and necessity of it."*). In some cases, threat modeling is explicitly required for certain types of applications (e.g., depending on the sensitivity or the business impact). In general, however, it is rarely imposed, as doing so would result in it becoming a checkbox activity (*"once you start having these compliance requirements [. . .] they will just not write stuff down anymore. So, the question is, what is the impact of that going to be?"*).

The relevance and usefulness of threat modeling are already appreciated by teams in several organizations (*"threat modeling is also well received, generally, by the teams"*).

RQ1.3. How are the threat modeling results used?

Follow-up is mostly an ad-hoc activity for which the responsibility usually lies with the team itself (with the exception of some severe issues where the security team actively follows up). How to monitor and follow up on the results more systematically is a recurring challenge (*"That varies depending on the team, and also on the priorities of the product owner [...]"*). This will be explored in more detail in Section 3.4.

One activity that does frequently occur is pentesting, which allows to verify the implementation of mitigations and tends to resurface issues that were not resolved by the teams. Having access to a threat model was mentioned to simplify the pentest process. There is also an opportunity here for positive feedback. Analyses that do not uncover any findings often result in minimal reports, and stakeholders may think that they waste time and resources without really gaining any value. The observation that the team properly implemented the right mitigations is, however, something that can also be actively communicated to them as positive feedback (*"[as a pentester,] it's not really accepted yet that you just go back to a customer, and say, 'gee, you guys just did a great job'."*).

Summary

While there is no consensus on the definition of threat modeling and what these activities specifically entail, all participants recognize and agree on the importance of threat modeling. The obvious benefit perceived by participants is the identification of security threats, as this is the primary reason to perform threat modeling. An important secondary benefit recognized by many participants is raising security awareness among the development teams. Intrinsic motivation of the development teams to perform such analyses was considered an important aspect by many participants, stressing the desire to have the teams want to perform such activities rather than a mandatory assessment that would be perceived as checkbox compliance exercise. In some organizations, threat modeling is required for critical applications. Using the threat modeling results and especially the more systematic use and follow-up of the results is more of a challenge for organizations.

3.2 Involved organizational roles

The second research question concerns the involved stakeholders, specifically (1) during threat modeling, (2) introducing threat modeling, (3) the goal of management and operations, and (4) the involvement of third parties.

RQ2.1. Who is involved in threat modeling activities?

Promoting threat modeling and making development teams aware of its benefits is mostly done by dedicated security teams. In general, the development teams themselves are responsible to start threat modeling, but the security team may also suggest or mandate threat modeling, especially for high-risk applications (as was described in Section 3.1).

The main stakeholders involved in the threat modeling activities are the development team, the product owners, and an architect, supported by a facilitator from the security team. To a lesser extent, testers, information security officers (ISOs), and operations are involved. The lesser involvement of these other roles is usually the consequence of their limited availability. Two participants mentioned that involving incident response people can be particularly useful, enabling the additional insight into which types of security concerns are relevant and actively abused in incidents; however, their involvement is rare (*"[...] they don't have the capacity [to attend threat modeling sessions]"*, *"we share our threat models with [incident response] [...] but I think it would be better if they just join threat model sessions."*).

RQ2.2. How is threat modeling introduced?

For most organizations, threat modeling has been introduced fairly recently (i.e. in the past 5–6 years) by the security team. Most people that take up an active role in introducing threat modeling to an organization had prior experience with pentesting (*"We noticed that information security officers found it difficult to start up threat modeling activities, and because pentesters are more involved in the [development] activities, we noticed that they could do so more easily"*). In general, participants did not mention a specific trigger to start up threat modeling activities other than having heard about the technique and its benefits. Structured approaches to start up threat modeling activities were also not mentioned: in general, the security team gets familiar with threat modeling through literature (e.g., Shostack [24]) and gradually learns to apply existing threat modeling approaches (*"we gradually learned [to threat model] together"*).

One exception is that, in one of the interviewed organizations, external expertise was consciously attracted to introduce threat modeling into the organization (*"I really followed [hired expert] around for 3 months, almost like a shadow, and that helped a lot too"*), which enabled overcoming organizational challenges and habits (*"[the expert] does not have the bias of the organization and its processes"*).

RQ2.3. What is the role of management and operations?

Information security officers and management positions are often only involved in the communication of the results. However, it is often difficult to communicate these results and clarify the usefulness of threat modeling. Being able to demonstrate a clear business impact and having success stories can help to communicate the results (*"We share successful [threat modeling] stories from time to time, so that [management] sees the added value."*). Management positions are rarely involved during threat modeling sessions. Furthermore, follow-up by management is lacking, and challenging in general (Section 3.4). Operations, including members of the Security Operations Center (SOC), are also rarely involved, except when applications are bought from third parties and need to be integrated. In such a case, operations are the main stakeholder.

RQ2.4. Are any external parties involved?

In all interviewed organizations, threat modeling is performed in-house, with support from the security team. In a single case, however, external expertise was consciously attracted to introduce threat modeling into the organization, which enabled overcoming organizational challenges and habits.

When software is acquired rather than developed in-house, it may be necessary to involve the provider of the application when making a threat model of the integration. Similarly, when software is hosted externally, the host may need to be involved in the threat modeling process. Not all third parties, however, provide equally detailed security documentation (*"Then you depend on, on the one hand, [third parties] being able to provide information, and on the other hand also the level of maturity on security of those kinds of companies."*). Mitigating security threats which require help from the external party is therefore mentioned to be challenging.

Summary

The main stakeholders during a threat modeling session are the development team, the product owner, and an architect, usually supported by a facilitator from the security team to provide expertise. Testers, ISOs, and operations are usually not involved. Management roles are often only involved in the communication of the results. In many cases, the introduction of threat modeling was triggered by prior (pentesting) experience of a security team member. One organization hired external expertise for this particular purpose, which was well-received. The security team then further propagates threat modeling within the organization.

3.3 Threat modeling process

The third research question concerns the threat modeling process, including (1) the trigger, (2) teaching threat modeling, (3) inputs and models, (4) threat elicitation, and (5) output and follow-up.

RQ3.1. When are threat modeling activities triggered?

As described in Section 3.1, while threat modeling may be mandated for high-risk applications, organizations foster intrinsic motivation, and threat modeling activities are therefore also mostly triggered by development teams that want to investigate the security of their application. This usually involves reaching out to the security team for training or support, or for confirmation or feedback on their threat models.

There is an overwhelming consensus that threat modeling is a continuous effort and thus requires periodic re-assessments. Implementations vary from development teams reaching out for feedback on their models to the security team frequently checking in with developers to do a re-assessment if necessary. In practice, such reassessments, and follow-up in general, depends on the willingness of development teams and the priorities of the product owner, and overall is challenging. When prompted about tool support, participants recognize the opportunity for tooling and automation such as integration in CI/CD pipelines to trigger reassessments if changes may introduce new threats, but none of the organizations do so at the moment, mostly due to the lack of tool support. In general, threat modeling therefore remains mostly a one-time activity, and models are infrequently revisited or updated.

The usefulness of early threat modeling is recognized, but this is in practice not always done. One of the reasons for this is the backlog of high-risk applications which require a threat model, leaving less room for the security teams to support early-stage threat modeling sessions (*"[...] we're actually catching up now first, which means you're mostly threat modeling on applications that are already live"*). Even so, there are several instances were threat modeling was applied very early in the development lifecycle, in tender processes and procurement, leading to valuable feedback and concrete security requirements. For example, in one specific case mentioned by participants, threat modeling during procurement later prevented a specific ransomware attack.

RQ3.2. How is threat modeling taught?

In general, the interviewees indicated that a threat modeling session usually starts with an introduction to threat modeling, which varies from a couple of slides (*"a few slides, two or* *three, to shortly explain the methodology "*) to more lengthy ones (*"we first gave an introduction of about 40-45 minutes"*).

Providing separate learning materials or organizing workshops before the actual threat modeling session is also prevalent. While generally perceived as useful, participants indicate that separate learning materials do not suffice to teach teams to threat model independently (*"I don't see teams picking it up and doing this completely independently any time soon"*), and teams may not always go through them (*"I don't think they go through the materials we are sharing with them"*).

Besides introducing the methodology and basics of threat modeling, the following aspects are usually covered during training. First, teaching teams to think about what can go wrong was mentioned several times (*"worst-case thinking really needs to be taught"*). Second, while teams may be more comfortable with a well-defined method, several participants note that the exact methodology in general is of little importance, and that thinking about security at all is more important than following strict guidelines (*"the most important thing is to start [threat modeling]. You can't really do something wrong"*). Third, teams may lack security expertise, so some examples or prevalent threats may also be illustrated. This lack of security expertise was also mentioned as the main reason why teams are not confident to independently start threat modeling (i.e., without the presence of a facilitator or security expert), as for example described by *"My impression is also that they are perfectly capable of doing it themselves if they have seen it once. That last 5% is indeed 'what do we [as security experts] see?'. And they can't do that themselves."*

RQ3.3. What kind of inputs (models) are used?

The first step of a threat modeling session is usually to create a model of the application or system being analyzed. Overall, the diagrams created or used in the context of threat modeling can take various forms, ranging from re-used architectural documentation to whiteboard diagrams. There is a balance between diagram quality conventions and the effort for teams to adhere to them because of the overhead they introduce. As a result, tool support for creating diagrams is mostly limited to drawing tools like Threat Dragon [18], but in some cases more elaborate modeling support like Microsoft's Threat Modeling Tool [13] is also used. In terms of model types, data flow diagrams (DFD) were most commonly used for software systems. One exception is the interviewed organization focused on operational technology (OT), which used a map of the network layout as the primary model.

A broadly recognized benefit of threat modeling is that it forces the explicit consideration of architectural documentation which can be either non-existent or, more frequently, outdated. Threat modeling therefore provides an incentive to revise and update this documentation. In some cases, the security teams construct initial diagrams to bootstrap the threat modeling activities, based on the inputs of the development teams. An important concern for the creation of the diagrams is the scope of the analysis to ensure a focused discussion.

RQ3.4. How are threats elicited?

STRIDE (a mnemonic for *spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege*, which can be used to guide threat modeling exercises) is most frequently mentioned as the main driver for threat elicitation. Threat elicitation is not necessarily performed systematically (e.g., using the STRIDE threat mapping table as described in Shostack [24]). Indeed, organizations prefer flexibility, giving development teams freedom in how to do the analysis. Other approaches such as PASTA [35] are used when the need arises (*"we chose* STRIDE *at the time mainly because it's very easy to explain and very accessible"*).

Besides the system model, inputs that are frequently leveraged during threat analysis are the ingress points in the system, attack vectors, types of adversaries, and attack scenarios (*"[...] which threats, and which attackers do we think are interesting?"*). Organizations want to reuse any such organization-specific knowledge across multiple analysis activities.

Finally, participants perceive the value to be mainly in the process rather than in the quality of a threat model. That is, it is more important to do the analysis than to have a detailed threat model (*"Going through the process is perhaps the most fruitful."*). There are also generally no strict criteria on when analyses are finished. Usually, sessions end naturally when no new threats arise or when all elements have been covered.

RQ3.5. How are results reported and tracked over time?

In general, threat modeling results in a report containing the system model, identified threats, present mitigations and recommendations to resolve unmitigated threats. In some cases, richer descriptions are made using attack scenarios. To reduce the number of issues to tackle, threats can be prioritized (*"[...] a summary of the relevant risks, at the basis of which recommendations are made"*).

Overall, organizations want to limit the reporting overhead as writing everything out in textual reports requires substantial effort with limited returns (*"writing takes a lot of time, and I don't know if it's always worth the effort."*). In some cases, presentations of the results are used to limit such overhead. The execution of the threat modeling process itself is considered more important than the reporting. While tool support is considered, linking the findings to business risks remains a challenge and requires manual effort.

Follow-up is mostly an ad-hoc activity for which the responsibility usually lies with the team itself (with the exception of some severe issues where the security team actively follows up). How to monitor and follow up on the results more systematically is a recurring challenge (*"That varies depending on the team, and also on the priorities of the product owner*

[...]"). This will be explored in more detail in Section 3.4. While going through the process to create security awareness is, in some cases, the main goal, some participants expressed a wish for more frequent and standardized follow-up, but strict policies may not be favorable and result in compliance-like checkbox activities.

Summary

Threat modeling sessions are mostly triggered by development teams wanting to examine the security of their system or application. Participants agree that threat models should be started early on in the development lifecycle and require periodical reassessments, but this is not common practice as security teams are currently prioritizing a backlog of high-risk, operational systems. While dedicated training sessions are both commonplace and essential for instilling the proper mindset, enabling a team to independently execute threat modeling can be challenging. Software models used during threat modeling take various forms ranging from free-form whiteboard drawings to structured notations like data flow diagrams. (Up to date) architectural models are not always available for re-use, so (re)constructing them becomes an important part of threat modeling. Concerning the use of models, pragmatism prevails over conforming to standardized notations. A pragmatic use of the STRIDE acronym is the most common approach for identifying threats during threat modeling. In this context, taking action and moving forward is considered more valuable than achieving a perfect threat model or prioritization of threats. In most organizations, no strict follow-up processes for the results of threat modeling are in place.

3.4 Threat modeling experiences

The fourth and final research question concerns experiences with threat modeling, including (1) positive experiences, (2) challenges and (3) causes of difficulties.

RQ4.1. What are positive threat modeling experiences?

A major success experience consists of teams becoming increasingly aware of security and the advantages of threat modeling. In some cases, these insights directly prevented concrete attacks (i.e., ransomware attacks). Furthermore, threat modeling is mentioned to decrease the effort required to develop pentests. Participants also indicate that teams are starting to threat model earlier in the development lifecycle, and do so more periodically, which has a positive impact on the complexity and duration of threat modeling sessions. Threat modeling during the design phase, although not prevalent, was also indicated to be beneficiary, leading to concrete security requirements which can be taken into account throughout the remainder of the development lifecycle. Finally, involving external parties to introduce teams and organizations to threat modeling was also indicated to be beneficial.

RQ4.2. What are threat modeling challenges?

Threat modeling challenges described by participants relate to (1) planning, (2) training materials, (3) modeling, (4) threat elicitation and prioritization, (5) follow-up, (6) tool support, (7) involving management, (8) demonstrating effectiveness, and (9) intra-organizational differences. A comprehensive overview is provided in what follows.

Planning. Scoping threat modeling activities is crucial to manage their size and complexity. Starting too early may lead to an ill-defined scope, starting too late to a too large scope. Several participants described difficulties finding the right time to start or revisit a threat model. Furthermore, mitigating issues, especially design issues, may be difficult or even impossible when applications are already fully implemented or deployed (*"what can you still do, right?"*).

Security teams themselves may also experience difficulties to plan a session if teams request it close to their deadline (*"Not all teams are aware of our schedule as [the security team][...]"*). A more general challenge is that security teams simply may not have the resources to provide threat modeling support to all teams (*"we simply don't have the capacity for that yet, because we just have so many development teams."*).

Regarding the duration of threat modeling activities, teams may lose interest if a session takes too long, especially if it is dominated by one or a few people, or gets too technical, and teams may be reluctant to start threat modeling a large system or application due to the amount of time that must be invested (*"You have to keep the focus time short, right? [...] Otherwise the team gets bored or there's no time left."*).

In general, participants described that the best way to tackle planning-related challenges would be to make threat modeling a part of the default workflow of the teams, as they themselves know best when threat modeling would be opportune.

Training materials. One participant mentions that creating worked examples for threat modeling is challenging, both because it is time-consuming (*"they tend to be very time intensive to actually create"*), and because teams tend to focus on the specific material covered in the examples, which may hinder them from finding other issues (*"[...] the only thing they're going to be doing is regurgitating the exact same thing that you told them during the training, at which point, yeah, you can also just give them a checklist"*). Another participant mentions the lack of real-world experiences on how to introduce threat modeling to an organization (*"you rarely hear about, well, I did it this way, and you need this, and you need these contacts, and you need to arrange it this way."*).

Modeling. Architectural documentation is seldomly available or up-to-date (*"the documentation we get is almost never up-to-date"*), which hinders the creation of models and diagrams (*"the fact that we have to spend the beginning of a* *session on getting the model correct, or as correct as possible is, in my view, a bit of a waste of time"*). An underlying problem is that a single comprehensive overview is usually not available (*"there is no single record, with* the *truth, not even on a conceptual level"*). Tackling this issue by involving multiple architects was also mentioned not to be favorable by one participant, as this may lead to lengthy discussions (*"we prefer to have only the architect who is most involved there."*).

Regarding the types of diagrams used, one participant describes that data flow diagrams may not be ideal for more specific and technical types of analyses (*"For more the protocol related things, for example, this is where it kind of, kind of breaks down [...] because you really want to look at much more specific and technical issues."*).

Threat elicitation and prioritization. Participants prefer to choose a methodology and stick to it to avoid losing time on discussions (*"If you aren't careful, you will have a lot of discussions about the form before you actually get started."*). Specifically for STRIDE, one participant mentioned that it does not scale well, as even for smaller applications, the amount of threats may rise rapidly (*"as the number of flows in and out of an application increases, the amount of time you have to spend on it increases exponentially"*). As a result, applying STRIDE during more agile workflows was indicated to be cumbersome (*"in an agile sprint or something like that,* STRIDE *is quite a cumbersome method"*).

Regarding risk estimation, it requires both security expertise and domain knowledge, and guidelines on how to do so are lacking in general (*"First, we don't provide a clear framework, how to do that themselves, and second, even if we had some way to evaluate the risk, they would still be guessing it, it's not going to be accurate enough."*).

Other related challenges include not thinking about the attacker (*"knowing who you're up against... I notice that a lot of people don't talk about that"*), approaching a threat model too much from a pentest point of view, which may lead teams to get stuck on the details (*"[sometimes] we treat the threat model a little too much as a starting document for our pentest, rather than a standalone thing"*), communication (*"Totally different sides of an organizations are suddenly going to be collaborating [...] Purely on language alone, you have to be very careful with that."*), and supply chain management (*"[...] yes, we are fine, but what about our suppliers?"*).

Follow-up. In general, systematic follow-up on the outcome of threat modeling sessions is lacking. Security may not be a priority of the team or product owner, which may lead to threat modeling outputs being ignored (*"our product owner doesn't think that's exciting enough right now"*). This is especially the case when threat modeling is mandated by some policy (*"they just want a list, and ticked off, and then you've done well"*). Participants do agree that this is not due to the lack of security interest, but rather because teams have limited time (*"It's not that they don't want to do security, but they have so many other things to think about besides security."*).

Following up was mentioned to be difficult for multiple reasons. First, acting on the threat modeling results may require the help of external people, for example for externally hosted applications. In such cases, it may take time to get this on the agenda of the external entity (*"To solve an issue [with an external host] would involve creating a ticket, and most likely lengthy email conversations, phone calls, ... "*). Furthermore, as mentioned in Section 3.4, threat modeling sessions are planned late in the development cycle in some cases, which limits the changes that can be made to an application (*"[...] and then we find out that there are actually quite insurmountable problems in the software"*).

Participants also described that follow-up is challenging if it involves other teams or stakeholders within the organization. For example, there is a risk of interfering with previously made (design) decisions, potentially taken by other teams (*"[...] they all take separate, siloed actions and don't take into account what preceded it, or too late."*). This is especially relevant when there is a business incentive to deploy as soon as possible. In such cases, deciding what to do or how to process the output of a threat modeling session (if at all) may become tedious and time-consuming (*"that generates a lot of discussion"*). Furthermore, even if teams want to take into account the threat modeling outcomes, interpreting the results was indicated to be challenging by the majority of the participants (*"It might be a problem with other teams interpreting threat models, one team interpreting a threat model [differently from] another team."*). Standardizing the outputs may be one way to tackle this challenge, but too much standardization may deter teams from threat modeling at all (*"[...] then you do get some interchangeability of [threat modeling results], without immediately killing the whole enthusiasm by putting it in a straitjacket, because that's not the goal either."*). Another challenge related to system models is a lack of diagram conventions, which inhibits the use and interpretation of threat modeling documents by other teams. Finally, one participant describes the risk of assuming that other stakeholders will take care of an issue (*"Assuming that another team does something [... is] more a problem than having the same circles, squares, arrows and whatnot."*).

Tool support. Tool support (e.g., Microsoft's Threat Modeling Tool [13] and Threat Dragon [18]) was indicated not to be user friendly (*"I find that it lacks some things in terms of usability"*). Microsoft's Threat Modeling Tool specifically was mentioned to require a lot of detailed inputs in order to get to useful output (*"you really have to fill out a lot to get useful information"*; *"you also don't want to tire the team with all those details, like, what* TLS *version are you using, and stuff like that"*). Interpreting the output of threat modeling tools was also indicated to be challenging, mainly because it requires security expertise (*"at the very least you want to pre-*

vent [the teams] from, yes, not having the knowledge and, yes, then simply disregarding [the output]"). For these reasons, except to draw simple diagrams, using threat modeling tools during a session was generally avoided.

One participant mentioned that, to make threat modeling tools a part of the general workflow of teams, they should be simplified (*"a simple implementation so that teams can start using it at all"*). Another issue mentioned by one of the participants is that threat modeling tools do not allow to model business logic well (*"it's not really very easy yet to include business logic"*). Finally, while participants indicated that integrating threat modeling tools in a CI/CD pipeline could be beneficial, none of them do so at the moment (*"I don't see how you could integrate threat modeling specifically into your* CI/CD *pipeline."*). One participant described the idea to automatically create tickets for threats, but due to the number of threats that are identified by threat modeling tools, this could also be challenging (*"[to] have ten thousand tickets automatically open... That's not going to be nice."*).

Involving management. (Risk) management may not always be aware of the added value of threat modeling, which makes getting support, time, and resources for threat modeling challenging (*"Getting resources to do it from the higher-ups, that always requires work."*). Ideally, according to one participant, management should not push or mandate threat modeling, but support teams wanting to do it (*"I would hate to have to push that from a leadership role. [...] But management, according to me, does play a role in accepting it, seeing the added value of it and being able to translate that back to their stakeholders."*).

Second, involving management during threat modeling sessions could provide useful insights, but is challenging for two reasons. First, management may not be aware of the benefits of them being present and may think that threat modeling sessions require a strong technical and/or security background (*"They are very quickly afraid that it really becomes a very technical session."*). Second, management simply may not have the time to join threat modeling sessions (*"[...] we have a single* ISO *right now. [...] Yeah, that's too few."*).

Finally, management does not follow up on the results of threats modeling sessions according to several participants (*"that just doesn't always happen or, at least, not consistently"*). Even if management would like to follow up, they may not always be able to correctly interpret threat modeling reports, because they are not always involved or familiar with the context (*"You need to be able to interpret a report."*). This lack of follow-up could result in a lack of oversight across applications and an organization in general (*"that leads to lack of oversight, where you can miss things"*).

Demonstrating effectiveness. Measuring the effectiveness of threat modeling, and security in general, is indicated to be challenging (*"evaluating whether threat modeling helps to achieve security is very hard, because you can't really measure security"*, *"it's an article of faith and we are part of the threat modeling church"*). However, in order to create awareness and motivate teams to do threat modeling, being able to communicate its added value may be crucial (*"What is the added value of threat modeling, right? And I think, making that clear and communicating unambiguously [and] empirically backed up [...] will be decisive."*). One participant mentions that the results of a pentest could be a starting point to evaluate a threat model, for example to identify issues that were missed during the threat modeling session (*"[...] does the pentest show up stuff that wasn't in a threat model or assumption that were incorrect?"*). Evaluating the artifacts created and used during a threat modeling session itself is also indicated to be challenging (*"Looking at the artifacts themselves [...] that's also an area that's still a bit open."*).

Intra-organizational differences. While our interviews only include one participant with a focus on OT (including for example industrial control systems), an important source of difficulties for that participant stems from the inherent (cultural) differences between the IT and OT domain. Mitigating certain threats or creating more secure systems may involve enforcing policies (for example related to patching), also on the OT side, even though IT policies don't always translate well to an OT context (*"*IT *organization as I know them are often quite bold and understand little of the* OT*, yet they feel we must comply with their policies."*). Understanding the differences between IT and OT, and effective communication between both sides, is therefore seen as an important but challenging aspect of security in general (*"embrace the fact that our worlds are different"*).

RQ4.3. What are the causes of the experienced challenges?

Challenges concerning motivation, timing, and follow-up are mainly caused by product owners, information (security) officers, and other management roles not being aware of the benefits of threat modeling. A root cause for this is that demonstrating the effectiveness of threat modeling is challenging. Teaching teams how to do threat modeling is furthermore complicated by a lack of a security mindset and knowledge within the team. Finally, the limited use of software tools for threat modeling is due to the required effort that outweighs the perceived benefits.

Summary

Development teams in the interviewed organizations are becoming increasingly aware of threat modeling and its advantages, and teams are starting to threat model earlier in the development lifecycle, and more periodically, which has a positive effect on the complexity and duration of threat modeling sessions. Other positive threat modeling experiences mentioned by participants include the prevention of concrete attacks, the use of threat modeling results when pentesting, and involving external parties to help introduce threat modeling to an organization.

Threat modeling related challenges faced by organizations include (1) finding the right time to start a threat model and finding a time slot that fits all stakeholders, (2) dealing with the overall lack of security expertise when introducing teams to threat modeling, (3) overhead during threat modeling sessions related to, among others, the lack of architectural documentation, discussing and deciding on the methodology, risk estimation, and long technical discussions, (4) the lack of follow-up, adequate tool support, and management involvement, (5) demonstrating the effectiveness of threat modeling, and (6) different (security) cultures between different parts of the organization, and IT and OT in particular. A lack of (1) threat modeling awareness at the level of product owners and management roles, (2) security knowledge among development teams, and (3) adequate tool support have been mentioned by organizations as potential causes of these challenges.

4 Discussion

This section discusses the main implications of our observations for practitioners, potential directions for future research, and the limitations and threats to validity of our study.

4.1 Advice for practitioners

Based on this study's findings, the main advice for organizations is to consider and incentivize thinking about security in any shape or form, rather than mandating threat modeling and imposing strict requirements on the methodology. Indeed, one of the major perceived benefits by participants is that it increases security awareness among the development teams. When evaluating their threat modeling practices, it is therefore important for organizations to recognize that there is no one-size-fits-all threat modeling approach that has worked for every organization, and that even within a single organization, different teams or applications may require a different approach. In this regard, forcing the use of a specific tool with the hopes of it leading to an efficient and fruitful threat modeling process should be avoided. Indeed, most of the interviewed organizations tried to use or considered using tool support to (partially) automate threat analysis, to support the creation of software models, or for more systematic follow-up, but adequate tool support seems to be lacking. Especially for organizations that are yet to start or just introduced activities related to threat modeling, it seems that successful instantiations of threat modeling spring from giving some space and flexibility to the development and security teams to see if, where, and how threat modeling can provide value, and gradually building upon and expanding this expertise.

In an ideal scenario, threat modeling is done early in the development lifecycle, as mitigating discovered threats in large, existing systems that are already operational may not be straightforward. Furthermore, threat modeling should ideally be repeated when changes are made (e.g., new features, or changes to the architecture). However, several organizations have highlighted difficulties with planning and finding the right time to threat model. Making threat modeling a part of the default workflow of development teams may alleviate such challenges, yet care must be taken that it does not become a checkbox activity. The fact that threat modeling allows gaining and maintaining a mutual understanding of an application and its architecture can also be promoted to incentivize teams to periodically apply threat modeling.

Product owners and management roles in general need to be aware of the potential benefits of threat modeling and allow for the necessary time for development teams to learn and apply this skill. Therefore, besides incentivizing development teams, awareness campaigns aimed at management roles could be fruitful. Such raised awareness may also contribute to better follow-up of threat modeling results which, in many of the organizations, appears to be limited and ad-hoc. Besides following up on threat models, actually involving management roles during threat modeling sessions was indicated by participants to be valuable, yet care must be taken that such sessions then do not become too technical.

Finally, organizations could use the research questions of this study as a starting point to evaluate their own threat modeling practices.

4.2 Directions for future work

The findings of this study reveal potential directions for future research regarding threat modeling. First, in order to further convince management roles of the benefits of threat modeling, the effectiveness and return on investment of threat modeling could be investigated, be it in terms of finding threats, raising security awareness (and thus preventing future threats), or supporting subsequent security activities like pentests.

Second, participants recognize the potential benefit of using tool support to automatically trigger re-assessments of threat models when significant changes are made to a system, but currently available threat modeling tools do not offer such capabilities. Future research and development efforts could aim to improve tool support and allow such integration in a CI/CD pipeline. Furthermore, the usability of threat modeling tools should be investigated, as participants agree that currently available tools require too much effort and, as a result, are not fit to be integrated in agile development processes.

It should, however, be noted that the described usability issues with current threat modeling tools may not necessarily be encountered by organizations that have heavily automated their threat modeling activities, and which may appreciate more detailed modeling capabilities and outputs. Still, since the interviewed organizations utilized little tool support, it would be interesting to see how threat modeling tools could be refined to support such organizations, for example by guiding development teams through a threat modeling session without the presence of a facilitator of the security team, which was mentioned to be difficult mostly due to the lack of security knowledge among developers.

Finally, rather than such 'user friendly' tools (and frameworks in general) not being available, another issue could be that such tools exist, but practitioners simply do not know about them, or do not know how to use them. A similar phenomenon was investigated by Canedo et al. [4], who describe that privacy requirements elicitation tools and techniques used and studied in literature do not align with the ones used in practice, partially due to the lack of dissemination and training materials. Participants in our study also described that their choice of using STRIDE over other methodologies is partly due to there being more training materials available for STRIDE. Therefore, future work could investigate practitioner needs in terms of training materials, and how novel threat modeling techniques could be better disseminated.

4.3 Threats to validity

This study is based on only a few organizations (13 participants in 7 organizations), where often only one person from each organization was interviewed. Although this person was always well-placed and had a comprehensive view on threat modeling in the organization (i.e., a member of the organization-wide security team), they may not be fully aware of all threat modeling initiatives.

This study is also subject to several selection biases. First, it is performed on target organizations of the NCSC, which typically are large organizations in critical sectors with a dedicated security team; software development is not their main activity. The results are thus not necessarily representative for other (smaller or commercial) organizations. Furthermore, regarding self-selection bias, the organizations already implement some form of threat modeling and are willing to openly talk about it, and contacts were provided by the sponsor of this research. Moreover, most of the interviewees are threat modeling 'advocates', appreciating its value, and actively pushing its use. This study does not include (nor encountered) any organizations that have tried and abandoned threat modeling, or where no threat modeling program is being developed.

Interviews being the only research method used, there is a possibility for respondent or social desirability bias (e.g., idealized, or exaggerated versions). Some interviewees showed threat modeling reports of projects in which they participated to illustrate what was said, which partially tackles this bias regarding the findings related to process and outcomes. Furthermore, based on the numerous challenges and negative experiences listed by participants, it is unlikely that an idealized version was presented. Moreover, with a limited interview duration of one hour (or 90 minutes if two participants were interviewed simultaneously) and the use of a responsive interviewing style, not all topics listed in the interview guide were explored in equal depth in each interview. Potential interviewer bias was reduced by formulating neutral, open-ended questions in the interview guide.

Finally, this study focuses on activities under the name of 'threat modeling'. Other organizations may perform similar activities under a different name (e.g., a security design review, security risk assessment, or the creation of abuser stories). A broader study that focuses on all design-level security activities would be needed for a more complete picture.

The main observed success factors (e.g., fostering intrinsic motivation and pragmatism) and challenges (lack of architectural documentation, follow-up, etc.) are shared by all interviewed organizations. Later interviews revealed no major new or contradictory observations. While this is not a grounded theory study, this indicates a certain level of data saturation. It should be noted that all contacts received from the sponsor (or other contacts provided by them) were interviewed, and that data collected stopped due to the contacts being exhausted, not because data saturation was reached. Still, we are confident that the observations described in this paper will, in general, also apply to other large organizations which are not primarily software development organizations, but have in-house software development teams, and apply some form of threat modeling. Further research is needed on the applications of threat modeling in other types of organizations, notably those focused on OT, as our study only included one participant of that sector.

5 Related Work

This section summarizes studies similar to this one, and highlights findings which differ from our observations.

Several practitioners have described their experiences and lessons learned from applying threat modeling within their organization. For example, Shostack [23] describes the threat modeling approach used by Microsoft, Ingalsbe et al. [9] describe their experiences at Ford, and Dhillon [7] elaborates on threat modeling at EMC Corporation (now Dell EMC).

Additionally, several empirical studies investigated specific threat modeling techniques. First, Stevens et al. [31] introduced a specific threat modeling framework to New York City Cyber Command and report the adoption and efficacy of threat modeling practices. Their participants stated they perform threat modeling in their daily efforts [31], observing analogous awareness benefits as observed in our interviews. Second, Soares Cruzes et al. [30] performed a case study on the adoption of STRIDE in a company comprising five agile development projects and identify challenges similar to the ones observed during our interviews. Third, Bernsmed et al. [2] bundle the results from four different studies on threat modeling as applied in agile projects, focused specifically on the use of data flow diagrams, STRIDE, and Microsoft's Threat Modeling Tool [15]. Related to the overall organization of threat modeling activities, their observations also include that developers are the main stakeholders, and that there is a need for better integration of threat modeling activities in the development pipeline. Fourth, Weir et al. [36] propose a design for so-called security interventions, which are similar to threat modeling sessions, and evaluate their effectiveness in terms of increased security engagement from product managers and the ability for developers to produce threat assessments. Finally, Trentinaglia et al. [33] describe experiences and lessons learned through conducting threat modeling workshops with practitioners in multiple domains.

As already mentioned, the above-mentioned studies [2, 30, 31, 33, 36] consider specific threat modeling approaches, and mostly focus on the application of the approach. In contrast, our study is not limited to specific approaches, and considers, besides the execution of threat modeling activities, organizational and human-centered aspects including motivation, planning, and stakeholder involvement.

Jamil et al. [11] consider a similar, broad perspective on the organization of threat modeling activities in practice, specifically for cyber-physical systems, through interviews with security experts from several different domains. Contrary to our observations, which mainly relate to IT rather than OT, they describe that the security team executes threat modeling activities separately, using input from other stakeholders (e.g., developers and architects), and that the developers themselves are not actively involved during the process. This may be attributed to IT people not being familiar with the physical aspects of cyber-physical systems, which Jamil et al. [11] describe to be difficult, similar to our findings (Section 3.4). However, if developers are not actively involved during threat modeling activities, the benefit of increased security awareness among developers, which was observed to be one of the main goals of threat modeling by our study as well as related work (e.g., [31, 33]), will not be attained.

A final category of related work is papers which evaluate the effectiveness of threat modeling techniques through experiments in more controlled settings. For example, Scandariato et al. [21] summarize the results of several empirical studies related to threat modeling, including evaluations of STRIDE with respect to the amount and validity of threats found, and a comparison between visual and textual approaches, Tuma et al. [34] evaluate two variants of STRIDE in terms of the number of high-priority threats identified, and de Gramatica et al. [6] investigate if the use of catalogs of threats and mitigations has an effect on the actual and perceived usefulness of security risk assessment methods. While participants in our study indicate that such evaluations could prove useful, for example to convince management of the benefits of threat modeling (Section 3.4), they do not investigate the actual adoption and organization of threat modeling in practice.

6 Conclusion

This paper described the results of a qualitative interview study into the threat modeling state of practice within 7 large Dutch organizations. In terms of organizing threat modeling activities, organizations tend to foster an intrinsic interest in threat modeling rather than putting strict policies in place. The goals for threat modeling are to find and mitigate security threats, but also to raise the overall security awareness among developers. Following up on threat modeling results is indicated to be challenging.

The main stakeholders of threat modeling activities are the development team, an architect, and a facilitator from the security team. Testers and operations are usually not involved, even though their input may be valuable. When software is acquired and integrated rather than developed in-house, however, operations are usually the main stakeholder, and input from vendor may be needed to ensure a secure integration.

In general, a threat modeling session starts with a facilitator from the security team who provides an introduction of threat modeling, including an overview of the methodology (usually based on STRIDE). Then, a model of the system is constructed, the form of which ranges from whiteboard drawings to structured notations like data flow diagrams. Constructing a model may be time-consuming if architectural documentation is lacking. This model is subsequently analyzed, typically in a pragmatic manner. After the session, the facilitator creates a report which is distributed to the stakeholders. Follow-up is mostly ad-hoc, except when critical issues are identified. In general, this is a one-time activity, although participants agree that there should be periodic reassessments.

Positive experiences include the prevention of concrete attacks (albeit seldomly), and (much more commonly) increased developer security awareness. Challenges relate to, among others, planning, training, model creation, risk estimation, and follow-up. These are (at least partially) associated with product owners and management roles not being aware of the benefits of threat modeling, as well as the security team lacking the capacity to assist all the development teams.

Organizations can use these results to help inform decisions to start or extend their threat modeling efforts. Furthermore, threat modeling facilitators and researchers may base future efforts on the challenges identified in this study.

Acknowledgments

This research is partially funded by the Research Fund KU Leuven, the Cybersecurity Research Program Flanders, and the Dutch National Cyber Security Centre.

References

[1] ACM SIGSOFT. Empirical Standards for Software Engineering: Qualitative Surveys (Interview Studies). https://www2.sigsoft.org/EmpiricalStandards/docs/standards?standard=QualitativeSurveys.

[2] Karin Bernsmed, Daniela Soares Cruzes, Martin Gilje Jaatun, and Monica Iovan. Adopting threat modelling in agile software development projects. *Journal of Systems and Software*, 183:111090, January 2022.

[3] Zoe Braiterman, Adam Shostack, Jonathan Marcil, Stephen de Vries, Irene Michlin, Kim Wuyts, Robert Hurlbut, Brook S.E. Schoenfield, Fraser Scott, Matthew Coles, Chris Romeo, Alyssa Miller, Izar Tarandach, Avi Douglen, and Marc French. Threat modeling manifesto. https://www.threatmodelingmanifesto.org/, 2020.

[4] Edna Dias Canedo, Ian Nery Bandeira, Angelica Toffano Seidel Calazans, Pedro Henrique Teixeira Costa, Emille Catarine Rodrigues Cançado, and Rodrigo Bonifácio. Privacy requirements elicitation: a systematic literature review and perception analysis of it practitioners. *Requirements Engineering*, 28(2):177–194, Jun 2023.

[5] Victoria Clarke and Virginia Braun. Thematic analysis. *The Journal of Positive Psychology*, 12(3):297–298, 2017.

[6] Martina de Gramatica, Katsiaryna Labunets, Fabio Massacci, Federica Paci, and Alessandra Tedeschi. The role of catalogues of threats and security controls in security risk assessment: An empirical study with atm professionals. In Samuel A. Fricker and Kurt Schneider, editors, *Requirements Engineering: Foundation for Software Quality*, pages 98–114, Cham, 2015. Springer International Publishing.

[7] Danny Dhillon. Developer-Driven Threat Modeling - Lessons Learned in the Trenches. *IEEE Security & Privacy*, 9(4):41–47, 2011.

[8] Georgi Gerganov. whisper.cpp. https://github.com/ggerganov/whisper.cpp, 2023.

[9] Jeffrey A. Ingalsbe, Louis Kunimatsu, Tim Baeten, and Nancy R. Mead. Threat modeling: Diving into the deep end. *IEEE Software*, 25(1):28–34, 2008.

[10] IriusRisk. IriusRisk, 2021. https://www.iriusrisk.com/.

[11] Ameerah-Muhsinah Jamil, Lotfi Ben Othmane, and Altaz Valani. Threat modeling of cyber-physical systems in practice. In Bo Luo, Mohamed Mosbah, Frédéric Cuppens, Lotfi Ben Othmane, Nora Cuppens, and Slim Kallel, editors, *Risks and Security of Internet and Systems*, pages 3–19, Cham, 2022. Springer International Publishing.

[12] Peter Mell, Karen Scarfone, and Sasha Romanosky. Common Vulnerability Scoring System. *IEEE Security Privacy*, 4(6):85–89, November 2006. Conference Name: IEEE Security Privacy.

[13] Microsoft. Threat Modeling Tool, 2023. https://aka.ms/tmt/.

[14] Microsoft. What are the microsoft sdl practices? https://www.microsoft.com/en-us/securityengineering/sdl/practices, 2023.

[15] Microsoft Corporation. Microsoft threat modeling tool.

[16] NIST. Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities (SP 800-218. https://csrc.nist.gov/Projects/ssdf, February 2022.

[17] OWASP. OWASP Top 10 - 2021. https://owasp.org/Top10/, 2021.

[18] OWASP. Threat Dragon, 2021. https://owasp.org/www-project-threat-dragon/.

[19] OWASP. Software assurance maturity model. https://owaspsamm.org/, 2022. Version 2.0.3.

[20] Herbert J. Rubin and Irene S. Rubin. *Qualitative Interviewing: The Art of Hearing Data*. Sage, 2011.

[21] Riccardo Scandariato, Federica Paci, Le Minh Sang Tran, Katsiaryna Labunets, Koen Yskout, Fabio Massacci, and Wouter Joosen. *Empirical Assessment of Security Requirements and Architecture: Lessons Learned*, pages 35–64. Springer International Publishing, Cham, 2014.

[22] Bruce Schneier. Attack trees. *Dr. Dobb's journal*, 24(12):21–29, 1999.

[23] Adam Shostack. Experiences threat modeling at microsoft. In Jon Whittle, Jan Jürjens, Bashar Nuseibeh, and Glen Dobson, editors, *Proceedings of the Workshop on Modeling Security (MODSEC08), Toulouse, France, September 28*, volume 413 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008. https://ceur-ws.org/Vol-413/paper12.pdf.

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

[25] Laurens Sion, Stef Verreydt, and Koen Yskout. Codebook. https://figshare.com/s/7dcdefa2cf15ee2e01a0, 2023.

[26] Laurens Sion, Stef Verreydt, and Koen Yskout. Information sheet. https://figshare.com/s/b9d3e0f6a821591bba1e, 2023.

[27] Laurens Sion, Stef Verreydt, and Koen Yskout. Informed consent firorm. https://figshare.com/s/3036fb6087838e9770b8, 2023.

[28] Laurens Sion, Stef Verreydt, and Koen Yskout. Interview guide. https://figshare.com/s/4768b946e59ea933cff1, 2023.

[29] Laurens Sion, Stef Verreydt, and Koen Yskout. Threat modeling in nederlandse organisaties. https://www.ncsc.nl/documenten/publicaties/2024/mei/7/index, 2023.

[30] Daniela Soares Cruzes, Martin Gilje Jaatun, Karin Bernsmed, and Inger Anne Tøndel. Challenges and experiences with applying microsoft threat modeling in agile development projects. In *2018 25th Australasian Software Engineering Conference (ASWEC)*, pages 111–120, 2018.

[31] Rock Stevens, Daniel Votipka, Elissa M. Redmiles, Colin Ahern, Patrick Sweeney, and Michelle L. Mazurek. The battle for new york: A case study of applied digital threat modeling at the enterprise level. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 621–637, Baltimore, MD, August 2018. USENIX Association.

[32] Tarandach, Izar. Pytm, 2020. https://github.com/izar/pytm.

[33] Roman Trentinaglia, Sven Merschjohann, Markus Fockel, and Hendrik Eikerling. Eliciting security requirements – an experience report. In Alessio Ferrari and Birgit Penzenstadler, editors, *Requirements Engineering: Foundation for Software Quality*, pages 351–365, Cham, 2023. Springer Nature Switzerland.

[34] Katja Tuma, Christian Sandberg, Urban Thorsson, Mathias Widman, Thomas Herpel, and Riccardo Scandariato. Finding security threats that matter: Two industrial case studies. *Journal of Systems and Software*, 179:111003, 2021.

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

[36] Charles Weir, Ingolf Becker, and Lynne Blair. Incorporating software security: using developer workshops to engage product managers. *Empirical Software Engineering*, 28(2):21, Dec 2022.