A limited technical background is sufficient for attack-defense tree acceptability

Nathan Daniel Schiele
Leiden University

Olga Gadyatskaya
Leiden University

Abstract

Attack-defense trees (ADTs) are a prominent graphical threat modeling method that is highly recommended for analyzing and communicating security-related information. Despite this, existing empirical studies of attack trees have established their acceptability only for users with highly technical (computer science) backgrounds while raising questions about their suitability for threat modeling stakeholders with a limited technical background. Our research addresses this gap by investigating the impact of the users' technical background on ADT acceptability in an empirical study.

Our Method Evaluation Model-based study consisted of $n = 102$ participants (53 with a strong computer science background and 49 with a limited computer science background) who were asked to complete a series of ADT-related tasks. By analyzing their responses and comparing the results, we reveal that a very limited technical background is sufficient for ADT acceptability. This finding underscores attack trees' viability as a threat modeling method.

1 Introduction

Threat modeling has taken an increasingly prominent role in risk assessment and security-oriented design [2], especially in the area of secure software engineering [3, 28, 95]. *Attack-defense trees* (ADTs), a graphical component-based representation of attack scenarios, are a highly recommended model for analyzing attacks as well as communicating attack-related information to others in a succinct manner [2, 56]. ADTs have been long considered to be a suitable, versatile, and easy-to-use threat modeling approach [3, 32, 56, 60, 66, 68, 70, 74, 77]. However, as threat modeling process and results *need to be accessible to people with different backgrounds* [11, 18], in order to be effective as a threat modeling method, ADTs must be acceptable for all stakeholders in the software development process, including, among others, security analysts, software engineers, product owners, and managers [88]. For a model to be *acceptable*, stakeholders need to be able to use the model efficiently and effectively, as well as perceive the model to be useful and usable [54].

Thus far, there have been relatively few studies focusing on ADT acceptability. A few studies have directly compared attack trees with other threat models. For example, Opdahl and Sindre [58] and Karpati et al. [35] found that attack trees allowed for better analysis than misuse cases. Broccia et al. [8, 9] have recently demonstrated high comprehensibility and acceptability of ADTs for users with a technical background. Lallie et al. [48] compared fault trees (the precursor to attack trees) and attack graphs, a temporal state-based threat model [63], in a study with participants of different backgrounds. They found that those with a technical background strongly outperformed those without on both models [48].

As threat modeling is an important part of the secure software development lifecycle [51, 88] with a strong focus on collaboration [31, 88], it is crucial to understand whether such a popular and recommended method as ADTs is suitable for all involved stakeholders who might have a very limited technical background. To address this gap in the acceptability of ADTs, re-examine the findings from [48], and guide our research, we formulated the following research questions following the Method Evaluation Model (MEM) as described by Moody [54]:

RQ1 Is the actual effectiveness of ADTs affected by technical background?

RQ2 Are the perceived ease of use or perceived usefulness of ADTs affected by technical background?

RQ3 Is the intention to use ADTs affected by technical background?

RQ4 Does technical background impact how ADTs are drawn?

Our work aims to establish *whether the extent of the technical background affects the ADT acceptability* by conducting a study with student participants from different fields (53 computer science participants; 49 non-computer science participants with a very limited technical background) who

complete the same suite of tasks involving using and creating ADTs. Our study is the first to examine ADTs in this context, and, to the best of our knowledge, our study is the first to examine the creative aspect of using any threat model by having participants create an ADT for a scenario of their own choosing and comparing the resulting set of ADTs.

Our main findings are:

- *A limited technical (computer science) background is sufficient for the acceptability of ADTs*: Participants of different backgrounds did not show a significant difference in their usage or perceptions of ADTs.
- *A creative component in designing ADTs does not appear to be affected by the background*: All self-drawn ADTs fell within the same general limits (in terms of the number of nodes, depth, refinements, etc.), represent similar types of scenarios, and are of similar quality, regardless of the background of their authors.

Overall, our results strongly support ADTs as a threat modeling tool that is acceptable for threat modeling stakeholders, including those with a very limited technical (computer science) background. We share our study design, training materials, and the anonymized data from the participants in [64] to enable further research in this field.

The remainder of this paper is structured as follows. We first present the necessary background information on ADTs and threat modeling and summarize the relevant state-of-practice in threat modeling in Sec. 2. We then review the related work on empirical studies in Sec. 3. Sec. 4 presents the methodology of our study. It is followed by Sec. 5 presenting the study results and answering our four key research questions. We discuss the results in Sec. 6 and acknowledge the study limitations in Sec. 7. Sec. 8 concludes this paper.

2 Threat Modeling and Attack Trees

The notion of *threat modeling* (TM) refers to a process to identify relevant attacks or threats; it typically takes place in the context of software development or security risk management [83, 93]. In the context of software development, TM can refer to a requirements elicitation or design analysis technique [69]. Given the diversity of secure software development guidelines [42] and security risk management methods [23] – and the wide variety of organizational contexts and systems where threat modeling is applied – there are also many established TM approaches [6, 22, 68, 78, 83, 93]. These methods differ substantially in their focus and process to follow: e.g., STRIDE helps with discovering pertinent security issues during software development, LINDDUN is designed for privacy threats, TARA and Persona non Grata focus on identifying relevant attacker profiles, while PASTA and OCTAVE cover the whole security risk assessment process [6, 68].

Attack trees. *Attack trees* (sometimes called *threat trees*) were proposed by Bruce Schneier in 1999, inspired by the fault trees model [65]. According to Shevchenko et al. [68], attack trees are one of the oldest and most widely used threat modeling methods that help capture and dissect possible cyber, cyber-physical, or physical attack scenarios. Attack trees are labeled acyclic graphs (trees) in which every node label is either an attacker's goal or an attack component in service of that goal. Each node can have any number of child nodes with a defined relationship, otherwise referred to as a *refinement*, between those nodes. The OR relationship indicates that one child must be completed for the parent to evaluate as complete. The AND relationship indicates that all children must be completed for the parent to evaluate as complete. We note that each parent node can be refined in only one way (either AND or OR) or have no children at all: such nodes are called *leaf* nodes, and they represent simple attacker's actions that don't need to be further specified. This simple AND-OR tree model is very versatile and allows representing complex scenarios succinctly [53, 65, 90].

Mauw and Oostdijk defined the attack tree theory by proposing several semantics that can be used to represent attack trees formally [53]. Kordy et al. further expanded on this by introducing attack-defense trees (ADTs) [40]. ADTs allow each attack node to have a single countermeasure edge to a defense node, representing a defense to the attack goal or component it is attached to. These defense nodes are roots of their own defense subtree, with the same construction rules as attack trees, including being able to have countermeasure edges to attack nodes, representing an attack against the defense. Thus, attack trees are a particular case of ADTs that do not have any defense nodes. The ADT model allows representing complex attack-defense scenarios where defenders can deploy countermeasures against attacks, and attackers can try to circumvent these countermeasures [40, 41]. Moreover, considering countermeasures explicitly and collecting a library of best practices for mitigation are recommended in the TM literature [19, 31, 82], and ADTs can help with these objectives. Fig. 1 shows an example ADT from [75].

Attack trees and ADTs have been further expanded in other ways [27, 39, 90]. Our work focuses on ADTs *à la* Kordy et al. [40], i.e., without any additional attributes.

Attack trees usage in practice. Attack trees are quite popular, as evidenced by the fact that they are described in many textbooks (for example, Bishop [5], Stallings and Brown [72], van Oorschot [86], and Anderson [2]), authoritative references on threat modeling (Shostack [70], Shevchenko et al. [68], Bodeau et al. [6], and Tarandach and Coles [77]), adversarial modeling (CyBOK [74]), and advice from relevant government institutions and industry bodies (e.g., the UK NCSC [50, 56]), OWASP [59], or US NIST [57]).

It is frequently recommended to combine STRIDE-based threat modeling with attack trees for more in-depth analysis of critical data flows and threats [32, 70, 77]. This aligns

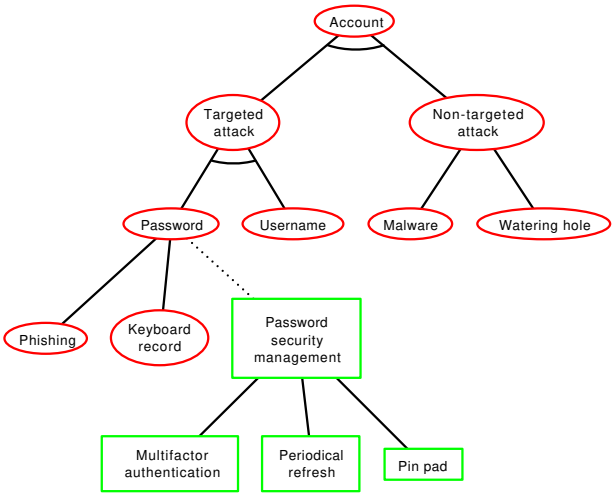

Figure 1: An example of an ADT (the second ADT in our small study) from [75].

well with the evidence-based recommendation to complement data flow diagram-based analysis of STRIDE with expressive attacker models by Van Landuyt and Joosen [85] and observations from practitioners that having a library of relevant threat scenarios improves the TM outcomes [19]. Schneier advises organizations to develop collections of attack trees to share knowledge and alleviate the need for in-depth security expertise [65]. LINDDUN implements this advice, featuring a dedicated privacy threat trees catalogue [17], which was appreciated as useful by participants in an empirical study evaluating LINDDUN [92]. Jamil et al. [30] report that attack trees are chosen as a method because they can help covering all possible attack entry points.

Despite the popularity, to the best of our knowledge, there are few established references that prescribe how to apply attack trees. Sonderen [71] designed a manual for producing attack trees. The manual aims to support a single person designing an attack tree for a given scenario (i.e., the context is not a TM exercise done as a team); it was refined and evaluated in both a qualitative study and a case study. Sonderen reports that careful handling of the levels of abstraction is the most important for a structurally solid attack tree. Schneier prescribes to develop an attack tree top-down, revise it over time, and share with one or more colleagues to improve the completeness of the model [65]. He also advises having a library of attack trees that could capture relevant attack scenarios and can be reused – and thus diminish the need to have security experts around.

Threat modeling best practices. The TM literature offers substantial insights into the practice of threat modeling. However, it is clear that there is still a gap in understanding how different human factors affect the TM process [81]. Stevens et al. [73] reported on their experience with introducing the Center of Gravity TM approach to New York City Cyber Command, highlighting the benefits of threat communication that were reported by the participants. Thompson et al. [79] interviewed and observed 12 medical device security experts to understand their TM practices. They find that the approaches to TM used by different experts vary, and it is important to support a free-flowing, natural approach to ideation (brainstorming).

Verreydt et al. [88] conducted an empirical study of TM methods applied in Dutch organizations within the secure software development process. They found that while the roles involved in the software product development (developers, architects, product owners, and the security team) were central to conducting the TM process itself (this is concurred by other works, e.g. [4, 14, 19, 69, 82]), the outcomes are often communicated to information security officers and managers. Moreover, one of the reasons that management is not involved directly during the TM activities is the belief that such sessions require a strong technical and/or security background [88]. Involving a business representative familiar with the key business objectives is also recommended by Ingalsbe et al. [29]. Considering security risk management practices, Brunner et al. [11] also report on the heterogeneity of roles being involved: CxOs, quality and compliance managers, software developers, security-related staff, and others.

To summarize, TM is a team-based activity that involves different stakeholders: developers, security experts, product owners, and managers. Given the multitude of roles involved, communication becomes very prominent. While TM can be an opportunity to raise awareness about security in managers and bring their attention to the importance of security [14, 88], difficulties in communication and conveying security messages across the teams are known to be a security "blocker" [88, 89]. Thus, it is important to establish whether such a prominent TM method like attack trees is amenable for all stakeholders, especially for people without a substantial technical background. A positive answer would help organizations to recommend that management and other stakeholders with a limited technical background participate in TM more actively, as well as use attack trees for communicating the TM results outside of the product development team.

3 Related Work

Acceptability of attack trees. A common strategy for examining TM notations such as ADTs is a study designed to compare two or more notations against each other. Such studies split participants into several groups, and have them complete tasks designed to measure TM method efficacy, with the same tasks being performed using different methods. Opdahl and Sindre used this design to explore the effectiveness of attack trees compared to misuse cases, finding that attack trees are more effective, but the participants has similar perceptions of the two techniques [58]. This study has been replicated with industry practitioners by Karpati et al. who found simi-

lar effects, also showing that in the context of cybersecurity, students make a sufficient proxy for practitioners [35].

In Diallo et al. [20], two computer science master students applied Common Criteria, misuse cases, and attack trees to the same scenario, evaluating the methods' learnability, usability, analyzability, and clarity of output and finding advantages and disadvantages for each approach. They concluded that attack trees were easy to learn and use, provided a clear output, but were more difficult to analyze [20].

Broccia et al. applied the Method Evaluation Model (MEM) and used 25 human subjects (all with technical background) to examine attack defense tree acceptability [9][1]. This study was also recently replicated in another experiment with 49 subjects (computer engineering students) [8]. For their participants with a technical background, Broccia et al. found a good level of understandability and acceptability of ADTs [8, 9]. Yet, unlike our study, these studies did not examine participants with a very limited technical background.

To our knowledge, there has only been one previous study on the effect of technical background on attack tree effectiveness. Lallie et al. compared attack graphs to fault trees (considered as a variant of attack trees), finding attack graphs more effective [48]. Additionally, in the same study, they compared participants with a computer science background to those without one. Their findings did show that computer science participants were able to significantly outperform those without a computer science background using both models.

Studies of other security methods. Moving beyond attack trees, Katta et al. conducted an experiment with student participants to compare understanding, performance, and perception of misuse sequence diagrams and misuse case maps, finding that the models perform similarly [36]. Labunets et al. compared visual and textual risk assessment methods with student participants using a similar design, focusing on evaluating perception and effectiveness [44, 47]. They found that each type of method was effective in different tasks. De La Vara et al. conducted a study with students concerning Systems Process Engineering Metamodel-like diagrams, comparing this model to text descriptions; they found that the model was statistically significantly more effective in helping students understand the scenario [16]. Tondel et al. [80] examined the acceptability of Protection Poker in a study with computer science students and reported that the participants found it to be acceptable but perceived a limited impact on the security of the project. Wuyts et al. [92] empirically evaluated LINDDUN in a series of studies with students and a case study with experts, finding that the method helps to identify relevant privacy threats (correct), but many threats are also not discovered (incomplete). The participants perceived LINDDUN to be easy to use, but the method's efficiency was lower than expected [92].

A group of empirical studies focused on evaluating STRIDE-based threat modeling, as STRIDE is the most commonly used method [30, 79, 88]. For example, Bernsmed et al. [4] conducted a study with students to evaluate user acceptance and usage of two versions of a STRIDE-based threat modeling process. Scandariato et al. [62] evaluated STRIDE in a study with computer science students, concluding that STRIDE is relatively time-consuming (not very efficient), but it is perceived as easy to learn. The threats identified by the participants were largely correct, but many threats were not discovered (low completeness) [62]. Tuma and Scandariato [84] followed a similar design and compared time cost and effectiveness in threat elicitation of STRIDE per element and STRIDE per interaction in a controlled experiment with computer science master students, reporting that STRIDE per element provided better results. However, all these studies did not examine the effects of participants' background.

Examining the difference in backgrounds. The existing literature demonstrates that technical background can affect comprehension. Hogganvik and Stolen evaluated the background-affected comprehensibility of risk analysis terminology on professionals and students. They found a statistically significant difference in correct responses, concluding that background does affect comprehension [25]. Wu et al. conducted a study examining the ability of participants to understand security texts, finding that a significant percentage of security jargon is not comprehensible by those with a limited IT background [91]. Chen et al. [13] found that participants with IT background could understand explanations of Alexa skills privacy policies and related terms better than participants without an IT background.

To summarize, it appears that technical background seems to be an important prerequisite for comprehending many security-related concepts [25, 48, 91], and, in particular, users without a computer science background might be disadvantaged when using ADTs [48]. As threat modeling involves participants like managers with possibly a very limited technical background, we set out to examine in our study whether they would be disadvantaged when using ADTs compared to participants with more advanced technical backgrounds.

4 Methodology

This section outlines how we designed the study and collected data to address our research objective.

4.1 Choosing the methodology

Examining the aforementioned empirical studies, it is clear that there is no consensus or established guidelines on how to evaluate the acceptability of a threat modeling method such as ADTs, but several key dimensions and methods can be identified. *Comprehensibility* is important when users are presented with some security-relevant information (e.g., privacy policy or a warning) [13, 91], and it has been a point of attention in

[1] This research was performed concurrently with ours, and we had no knowledge of these works when designing and performing our study.

empirical studies of security methods [25,48]. Moreover, studies examining attack trees and other security risk assessment and threat modeling methods have looked at *effectiveness* in eliciting threats/requirements [46, 58, 84], and *acceptability* for the intended users [8, 9, 80]. Note that these objectives are not independent, for example, the comprehensibility of models (how well can users interpret them) produced can be considered as a part of the method's effectiveness [1, 36, 46], while the effectiveness can be assessed as a component of its acceptability [8, 9, 43, 73].

Two prominent frameworks have been used in the literature to assess the acceptability of a method by its intended users: the Technology Acceptance Model (TAM) [15] and the Method Evaluation Model (MEM) [54]. TAM focuses on the perceptions of the intended users and it prescribes to measure *perceived usefulness* (**PU**), *perceived ease of use* (**PEOU**), and the *intention to use* (**ITU**) [15]. MEM, depicted schematically Figure 2, extends TAM with components related to actual usage: in addition to the TAM constructs, it recommends measuring *actual effectiveness* (**AE**), *actual efficiency* – and, combined, these constructs will translate into *actual usage* [54]. These two frameworks have been used for evaluating tools and methods in a variety of fields, including cybersecurity. Among the previously mentioned studies, TAM has been applied in, for example, [4, 35, 36, 58, 80], while MEM was used in [8, 9, 43, 45, 73]. We wish to evaluate the suitability of ADTs as a threat modeling method through the lens of technical background, examining if performance and perceptions change based on the extent of the technical background of users. Since MEM examines perceptions as they relate to actual usage, we believe this framework is a suitable basis for our study design.

Moreover, threat modeling is a creative activity: teams frequently engage in brainstorming [11] and free-flowing creative thought needs to be facilitated [79]. Therefore, it is important to examine to what extent threat modeling stakeholders with a very limited technical background might be disadvantaged if they use ADTs for creatively expressing their ideas of relevant attacks. Therefore, to the MEM constructs **AU**, **PU** & **PEOU**, and **ITU** (which correspond, respectively, to our **RQ1**, **RQ2**, and **RQ3**) we add another dimension captured by our **RQ4**.

We detail how we use the MEM constructs and **RQ4** in our study context in the remainder of this section.

4.2 Study design

From the research questions **RQ1**–**RQ3** derived from the MEM components and our additional **RQ4** that examines differences in creative usage of ADTs depending on the background, we developed a **series of hypotheses** to specifically test the aspects of MEM and the creative usage component with the added context of technical background. The hypotheses are presented in Table 1. Figure 2 also shows the hypotheses positioned in their relevant MEM component. We have two hypotheses for each aspect we measure: a null hypothesis where we expect no difference between the two groups of participants, and an alternative hypothesis where we expect a difference. We start by testing for a difference as the previous study by Lallie et al. [48] observed an influence of technical background on successfully using attack trees.

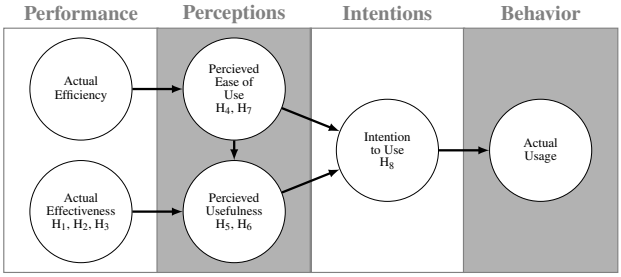

Figure 2: The Method Evaluation Model (MEM) [54] with our study hypotheses placed in context.

We measure the actual effectiveness (**AE**) of ADTs by looking at how well the participants can understand the provided ADT models (H_1), how effectively can they design ADTs (H_2) , and how many errors they make when designing these ADTs (H_3). We evaluate perceived usefulness (**PU**) by asking the participants to evaluate on the Likert scale how useful do they find ADTs, separately as a means of threat analysis (H_6) as well as a means of communication (H_5), as we want to see whether our participants would demonstrate different preferences depending on the background. We measure perceived ease of use (**PEOU**) by asking the participants to report on the Likert scale whether they find the provided ADT easy to understand (H_4) and if they find it easier to understand a given ADT compared to a textual description (H_7). Intention to use (**ITU**) is measured by asking the participants whether they would like to use ADTs in the future, on the Likert scale (H_8).

Finally, we studied the creative aspects of designing ADTs by examining how effectively the participants can design ADTs for the self-selected scenario ($H_{2\text{-}2}$) and by measuring, qualitatively and quantitatively, the differences in the ADT models designed for the self-selected scenario (H_9). Specific questions used to measure these aspects are listed in Table 1 (text of the referenced questions is available in Appendices A and B.). We provide more details about the measurements done per each hypothesis in the next section (Sec. 5).

Note that we do not measure the actual efficiency of using ADTs separately, because of the study constraints: it was given as a part of a homework assignment where participants worked at their own pace and according to their own schedule. However, we believe we are still able to evaluate ADTs within the scope of the MEM without assessing actual efficiency separately, as the ability of participants to *understand* ADTs by correctly interpreting existing models and creating new

ones after a short training translates into both effectiveness and efficiency (see Broccia et al. [9]).

Protocol design. In the context of this work, we consider the technical background to be a background in computer science-related subjects. Our study was designed to measure how the difference in background affects the measured components. Thus, we used a between-subjects design with two groups of students: one group with a strong computer science background, and another group with a very limited computer science background. Details about our participants are given further in Section 4.5. As common in such studies [9, 45, 48], our participants first received training on the studied method (ADTs). Both groups received the same lecture on ADTs given by the first author of this study, and afterward they participated in two identical study components: *a small study*, which was an automatically assessed online quiz, and *a large study* that involved a graded homework assignment.

The two researchers involved in the study have several years of experience in teaching ADTs to diverse audiences (university students in Bachelor and Master programs, with and without a computer science background). This experience was instrumental in identifying the right questions and tasks for measuring the different components of interest. The study questionnaires were not pre-tested with the target student population as this was part of graded coursework and students who had seen the questions would have an unfair advantage; instead, the questionnaires were developed by taking advantage of the researchers' experience in teaching ADTs. The ethical considerations of our study are discussed in detail in Section 9.

4.3 Study components

Small study. The small study contained 19 questions with each section starting with an image of an ADT with content and Likert questions for each ADT; each ADT was increasingly complex. This study focused on what information was received by looking at ADTs that were already created. All ADTs included in this assignment were taken from existing studies about ADTs [12, 38, 52, 75]. We selected such ADTs from the literature that a technical background would not be necessary to understand the attack scenario (i.e., without any specialist terms used for labels). All the questions and Likert statements can be found in Appendix A.

Students did not receive a grade for completing the small study, but they were able to see the correct answers to the questions for self-evaluation immediately after completing the quiz. They were encouraged to do the quiz for their own learning, to ensure they understand ADTs as a concept, and as preparation for the larger homework assignment on ADTs and the final exam where ADTs were among the test questions.

This assignment focused on checking whether students are able to read ADTs and interpret them in the context of the studied theory (comprehension of the models), as this was not a direct goal of the large study; although, as mentioned in Sec. 4.1, being able to interpret models correctly is necessary for the overall effectiveness of the method. Further, an important purpose of the small study was to establish if the provided training was adequate.

Large study. The large study was implemented as a take-home assignment, and students had four weeks to complete it at their own pace. This assignment was graded, contributing to the final course grade. Students were required to submit the assignment for the coursework, but they had to explicitly opt-in for participating in the study. We further discuss the ethical considerations of our study in Section 9.

This study consisted of three parts, with students creating attack trees in each part, under different conditions (from a set of components, from a given textual description, and for a self-selected scenario). Here, we aim to assess the more creative aspects of producing ADTs, which is the major motivation behind **RQ4**. To our knowledge, this is also unique among TM studies, as to the best of our knowledge, nobody has yet examined creative aspects of threat model design. The list of questions from this study is available in Appendix B.

4.4 Data analysis

Ultimately, since we start from the results by Lallie et al. [48], we wish to find if there is a statistically significant difference between two independent treatment groups (those with a technical background and those with a very limited technical background). Much of our data is gathered through Likert questions, which result in ordinal data that cannot be normally distributed [87], and for the remaining continuous data, we used the Shapiro-Wilk test to find that this data is not normally distributed [24]. Our data also does not have equal variance according to Levene's test [49]. Thus, we opt for the non-parametric Brunner-Munzel (BM) test [10] that is robust in the unequal variance case [21, 33]. As suggested by Labunets [44], when we do not find a statistically significant difference according to the BM test, we use the non-parametric Two One-Sided t-tests (TOST) to check for equivalence [67].

We correct for multiple tests using the Holm-Bonferroni (HB) correction method [26] and adopt a significance threshold of α=0.05, as is common practice in similar studies [9, 44]. In the remainder, we report the corrected p values (denoted for short as $p*m$).

4.5 Participants

Participants in our study were undergraduate students at Leiden University (The Netherlands). The LT (Limited Technical) students were predominately 3rd (final) year Bachelor students completing majors related to law, governance, and policy studies. The LT students were all a part of a minor focused

Table 1: Hypotheses to be investigated by our research and the research questions they contribute to. The null hypothesis in each case proposes no difference, while the alternative hypothesis proposes a difference between the Limited Technical (LT) and Highly Technical (HT) groups. In the measurement questions, `SS-` refers to questions from the small study, and `LS-` refers to questions from the large study. Question text can be found in Appendices A and B.

Null ID	Alt ID	RQ	MEM	(Alternative) Hypothesis Text	Measurement Questions
H^0_1	H^A_1	**RQ1**	**AE**	Difference in check-for-understanding questions between the LT and HT students	`SS-Q2, Q3, Q7, Q8, Q9, Q12, Q13, Q14, Q18`
H^0_2	H^A_2	**RQ1& RQ4**	**AE**	Difference in being able to successfully create ADTs	
$H^0_{2\text{-}1}$	$H^A_{2\text{-}1}$	**RQ1**	**AE**	Difference in the successful creation of an ADT from a text description between the LT and HT students.	`LS-ADT2`
$H^0_{2\text{-}2}$	$H^A_{2\text{-}2}$	**RQ1 & RQ4**	**AE**	Difference in the successful creation of an ADT from a self-selected scenario between the LT and HT students.	`LS-ADT3`
H^0_3	H^A_3	**RQ1 & RQ4**	**AE**	Difference in the number of errors made in ADT construction between the LT and HT students.	`LS-ADT1, LS-ADT2, LS-ADT3`
$H^0_{3\text{-}1}$	$H^A_{3\text{-}1}$	**RQ1 & RQ4**	**AE**	Difference in the number of multiple parent nodes used between the LT and HT students.	
$H^0_{3\text{-}2}$	$H^A_{3\text{-}2}$	**RQ1 & RQ4**	**AE**	Difference in the number of multiple refinement used between the LT and HT students.	
$H^0_{3\text{-}3}$	$H^A_{3\text{-}3}$	**RQ1 & RQ4**	**AE**	Difference in the number of multiple countermeasure nodes used between the LT and HT students.	
$H^0_{3\text{-}4}$	$H^A_{3\text{-}4}$	**RQ1 & RQ4**	**AE**	Difference in the number of single child nodes used between the LT and HT students.	
H^0_4	H^A_4	**RQ2**	**PEOU**	Difference in the self-assessment between LT and HT students.	`LS-ADT1-L1, SS-Q5, Q10, Q15, Q19`
H^0_5	H^A_5	**RQ2**	**PU**	Difference in the perception of usability of ADTs as a communication tool between the LT and HT students.	`LS-ADT3-L3`
H^0_6	H^A_6	**RQ2**	**PU**	Difference in the perception of usability of ADTs as an analysis tool between the LT and HT students.	`LS-ADT1-L5, LS-ADT2-L2, LS-ADT3-L1`
H^0_7	H^A_7	**RQ2**	**PEOU**	Difference in the comparison of ADTs to a written description of attacks between the LT and HT students.	`LS-ADT2-L1, LS-ADT3-L2, SS-Q6, Q11, Q16, Q20`
H^0_8	H^A_8	**RQ3**	**ITU**	Difference in the intention of students to use ADTs in the future between LT and HT students.	`LS-ADT3-W3, LS-ADT3-W5`
H^0_9	H^A_9	**RQ4**	N/A	Difference in the freely created ADTs of the HT and LT students.	`LS-ADT3`

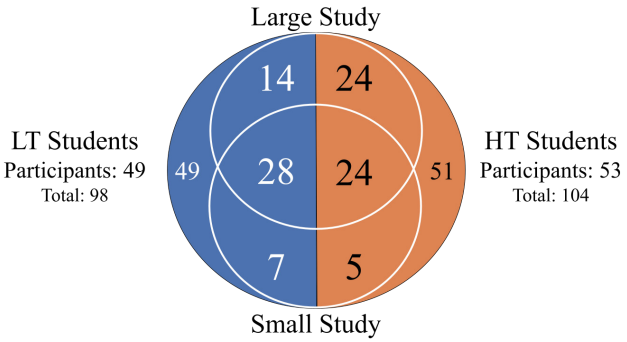

Figure 3: Distribution of participants across the treatment groups and studies.

Table 2: Comparison of the final course grades (out of 10) for participants and non-participants. `SS` stands for small study.

Type	Participant		Non-participant		BM Test		TOST	Effect Size
	n	mean grade	n	mean grade	statistic	$p*m$	$p*m$	Cohen's d
LT (all)	49	7.58	48	6.90	-2.05	1.0	1.0	0.70
HT (all)	53	7.52	48	6.39	-2.12	1.0	0.79	0.30
LT (SS)	35	7.64	63	7.23	-1.678	1.0	1.0	0.41
HT (SS)	29	7.54	72	6.68	-1.193	1.0	1.0	0.36
LT	49	7.58			1.384	1.0	**0.037**	0.11
HT	53	7.52						

on cyber security and governance. The HT (Highly Technical) students were predominately 2nd year Bachelor students within the Computer Science Department. Both groups of students were taking a major-appropriate Introduction to Cyber Security course, within which we ran our study.

We consider that the LT students have a very limited technical background and the HT students have a highly technical (computer science) background. This was confirmed with an optional demographic question asking participants how much programming experience they had. The LT participants had an average of 2.5 months of programming experience, which was the result of the LT students simultaneously taking a basic Python programming course (a component of the aforementioned minor)[2]; in contrast, the HT students had an average of 3 years of programming experience. Additionally, according to their curriculum description, the HT students had two years of dedicated study in computer science, including courses on computer architecture, databases, linear algebra, algorithms, etc. These courses are not taken by the LT students.

Figure 3 provides the participant distribution between treatment groups in each experiment. There were a total of 49 LT (out of 98 taking the course) and 53 HT (out of 196 taking the course[3]) consenting participants across the two studies. As the study was done in the educational context, we consider all submitted answers valid, even if part(s) of the questions were not answered. We reviewed all submissions and did not find evidence of invalid answers (e.g., participants who submitted intentionally wrong answers or answered randomly). Table 2 shows a comparison of the final course grades (composed, in addition to the large study assignment, of an exam and several other assignment grades) of students in both treatment groups demonstrating that these groups are comparable to each other. While the grade analysis implies that stronger students self-selected to participate in the study, especially the optional small study, we can conclude that this is not different per students' background and study program.

4.6 Training

As we mentioned, most of the empirical studies into the acceptability of security modeling methods provide training on the method as part of the study (see, e.g., [45, Table 2.4], or

[2]This course is designed for students with zero programming experience. By the end of the course, students are expected to be able to write small (less than 30 lines) Python scripts that may integrate self-defined or imported functions and use objects.

[3]In this group, it was possible to choose another assignment instead of ADTs, and 104 out of 196 students submitted the ADT assignment.

previous studies of attack trees [8, 9, 48].) As our training, we gave a 90 min. lecture on threat modeling more broadly and ADTs in particular to both groups of students. The lecture covered an overview of threat modeling and a detailed introduction to ADTs with several examples. It also included an interactive component where students created their own ADTs, which were presented to the class as a whole with any issues or improvements discussed. A short description of the lecture and the slides are available in the provided data artifact [64]. To ensure that both groups of students received a similar level of training, the slide deck and the lecturer were the same for both groups.

The lecture to the LT students was given in October 2022, and the lecture to the HT students was given in February 2023. Both lectures were given in person without streaming or a recording being made. Attendance was encouraged but not required in both courses. There was an optional demographic question before the large study, which was answered by 29 participants in each treatment group (59% of LT and 54% of HT). Of these, 26 participants in each group indicated they attended the training lecture. The percentage range of training attendance for LT is 53% - 93% and the range for HT is 49% - 87%. Students had access to the detailed lecture slides while working on the study components at home.

5 Study Results

In this section, we present the study results per our main research questions **RQ1–RQ4**.

5.1 RQ1: Effect of the background on AE

5.1.1 H_1: Understanding ADT concepts

As mentioned in Section 4.6, both LT and HT students received the same training in the form of a lecture. The lecture covered ADTs as a whole and delved into specific important concepts such as the types of nodes and refinements, levels of abstraction (LoA), and attack vectors. These concepts were addressed in detail during the lecture and practiced by students in small groups. We then tested the understanding of these concepts in the small study.

In Table 3 we see the aggregated responses to questions covering five chosen concepts related to ADTs. For each concept, Table 3 presents the number of questions asked about each concept. We see the number of respondents from both groups as well as the average percentage of correct answers for each population and concept. Finally, we can see the statistics and $p*m$ values (with HB correction) from the BM test. We can see that on four out of five concepts LT students scored, on average, somewhat worse than the HT students. However, only one of five topics (leaf nodes) has a statistically significant difference ($p*m < 0.05$) between the two populations according to the BM test, and all topics show statistically significant results for equivalence according to TOST. We provide a visualization of this comparison in Figure 4.

H_1: We find evidence of equivalence between LT and HT students on understanding ADT concepts.

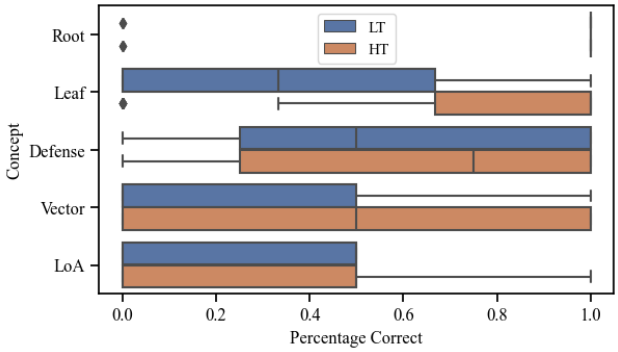

Figure 4: Comparison of the average scores across check-for-understanding questions.

5.1.2 H_2: Successfully creating ADTs

$H_{2\text{-}1}$: Creating ADTs from a written description. The second task of the large study was to create an ADT from a written description of an attack scenario. The written scenario was the result of reading out an existing ADT chosen by the research team into text. The students were tasked with reconstructing the original ADT from the text alone and were not told of the existence of the original ADT. They were specifically instructed to only include information from the scenario and to not introduce new information. From this task, we have 89 submitted ADTs (one participant did not submit an ADT for this task) that are all nearly identical, as they are drawn from the same source material. Because of how the task was designed, we consider that this task has a correct answer. As such, we can compare the ADTs created by students to the original ADT to find where the participants deviated.

Fifty-seven (57) ADTs (64%) were identical to the original ADT according to the seven metrics we chose to measure the similarity of ADTs[4]: the numbers of attack and defense nodes (we also separately count the number of attack and defense leaf nodes), the number of OR and AND refinements, and the number of levels of abstraction in a tree. Of those identical ADTs, 26 (61.9%) were provided by LT students, and 31 (64.6%) were provided by HT students. The complete set of results is found in Table 5.

Figure 5 summarizes the 32 answers that deviated from the correct ADT on at least one of the seven metrics. For example, if a student had one extra attack node, the figure would represent this answer as +1 in the "# atk nodes" category.

[4]To the best of our knowledge there is no established metric to measure distance or similarity between attack trees.

Table 3: Check for understanding.

Description	Questions	LT		HT		BM test		TOST	Effect Size
		n	% Correct	n	% Correct	statistic	$p*m$	$p*m$	Cohen's d
Root nodes	1	35	91.43	28	89.28	-0.28	1.0	**1.98e-17**	0.07
Leaf nodes	3	49	40.13	53	70.13	3.73	**0.025**		0.75
Defense nodes	4	49	51.19	53	61.13	1.25	1.0	**2.65e-19**	0.27
Attack vectors	2	35	34.29	28	44.64	0.84	1.0	**6.19e-10**	0.24
Levels of abstraction	2	34	26.47	27	37.04	1.36	1.0	**8.81e-17**	0.38

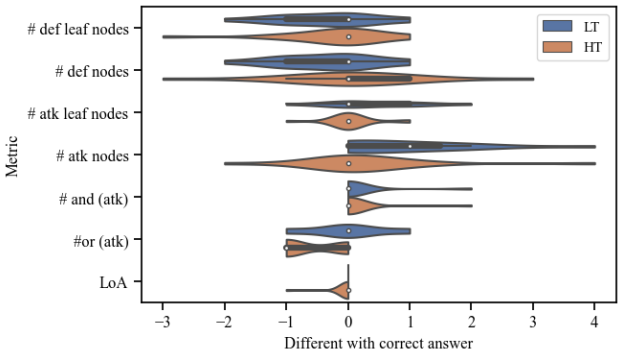

Figure 5: Comparison on creating ADTs from a written description.

s

Table 4: Qualitative analysis of self-drawn ADTs.

Quality	Largely Correct		Neither		Largely Incorrect		BM Test		TOST	Effect Size
	LT	HT	LT	HT	LT	HT	statistic	$p*m$	$p*m$	Cohen's d
Cohesive	21	22	15	19	4	6	0.46	1	**8.21e-07**	0.1
Clear	26	35	11	8	3	4	-0.91	1	**2.00e-07**	0.15
Concise	24	24	16	20	0	3	1.20	1	**6.96e-08**	0.3
Complete	30	28	9	15	1	4	1.67	1	**1.07e-06**	0.35

This figure shows that most students tended to make errors on only a few metrics, and produced results similar enough to the correct ADT. We see that only one HT student and no LT students made any errors regarding levels of abstraction (LoA); this could indicate that it is relatively easy for human participants to infer the different LoA from a textual description, and this holds for both participants with and without technical background.

$H_{2\text{-}2}$: Creating ADTs for a self-selected scenario. It is important to assess whether the participants are able to produce high-quality ADTs to represent a diverse set of attack scenarios. In total, there were 88 ADTs (two participants did not submit an ADT for this task) drawn for the task where students had to model their own scenarios.

We qualitatively evaluated the ADTs designed for self-selected scenarios (we call them *self-drawn ADTs*) based on four criteria: how meaningful are the refinements (*cohesiveness*), how clear are the labels (*clarity*), how relevant are the suggested attack components and whether there are any excessive steps (*conciseness*), and how complete are the scenarios (*completeness*). These qualities were selected to represent together a quality evaluation of the designed models.

Table 5: Results for hypotheses $H_{2\text{-}1}$, H_3, and H_9.

Hypothesis	Component		BM Test		TOST	Effect Size
			statistic	$p*m$	$p*m$	Cohen's d
$H_{2\text{-}1}$		ADT2 defense leaf nodes	0.656	1.0	**1.44e-11**	0.07
		ADT2 defense nodes	1.276	1.0	**4.88e-07**	0.23
		ADT2 attack leaf nodes	-1.435	1.0	**1.73e-16**	0.33
		ADT2 attack nodes	-1.727	1.0	**3.53e-04**	0.31
		ADT2 AND (attack)	-0.111	1.0	**3.77e-25**	0.02
		ADT2 OR (attack)	-2.496	**.969**	**5.15e-13**	0.52
		ADT2 levels of abstraction	-1.000	1.0	**5.57e-59**	0.2
H_3	$H_{3\text{-}1}$	ADT1 multi-parent nodes	0.32	1.0	**3.70e-13**	0.17
		ADT3 multi-parent nodes	-0.64	1.0	**9.87e-17**	0.21
	$H_{3\text{-}2}$	ADT1 multi refinement	-0.53	1.0	**8.23e-15**	0.03
		ADT3 multi refinement	-0.13	1.0	**3.54e-26**	0.04
	$H_{3\text{-}3}$	ADT1 multi countermeasure	1.76	1.0	**4.59e-12**	0.38
		ADT2 multi countermeasure	0.46	1.0	**9.86e-22**	0.02
		ADT3 multi countermeasure	-0.96	1.0	**0.035**	0.1
	$H_{3\text{-}4}$	ADT1 single child (attack)	-4.68	**8.66e-04**		0.79
		ADT2 single child (attack)	1.50	1.0	**1.01e-10**	0.19
		ADT3 single child (attack)	-2.66	0.641	1.0	0.47
H_9		ADT3 defense leaf nodes	-0.35	1.0	1.0	0.10
		ADT3 defense nodes	-0.31	1.0	1.0	0.13
		ADT3 attack leaf nodes	0.83	1.0	1.0	0.36
		ADT3 attack nodes	0.19	1.0	1.0	0.24
		ADT3 AND (attack)	1.24	1.0	1.0	0.34
		ADT3 OR (attack)	0.40	1.0	1.0	0.23
		ADT3 levels of abstraction	-0.53	1.0	0.105	0.10
		ADT3 and:or ratio	0.19	1.0	**1.95e-03**	0.11

The evaluation was done by two researchers experienced in attack trees and cybersecurity. First, the researchers designed together a rubric to evaluate ADTs based on these four criteria. The rubric was adjusted and calibrated in two iterations, when the researchers would first independently evaluate a set of randomly selected ADTs from both LT and HT participants and then jointly discuss the results. In the second iteration, the two researchers independently assessed all considered trees in the same way (reaching an agreement). This final rubric used to evaluate the ADTs according to these criteria is available in the provided data artifact [64]. The principal researcher then evaluated the whole set of ADTs based on the final rubric. The results of the evaluation according to this rubric can be found in Table 4, which shows that there is statistically significant equivalence between the groups on all four criteria.

H_2: We find no significant evidence of a difference between LT and HT students on effectively creating ADTs.

5.1.3 H_3: Common errors when designing ADTs

Another metric we used to compare the two populations of students is the common mistakes they made while creating ADTs. After manually checking all 180 received ADT images, we identified four common types of mistakes described below.

$H_{3\text{-}1}$: Multi-parent nodes. These describe nodes that have more than one parent. ADT construction rules (syntax) allow only a single parent for every node [40]. For each node that had more than one parent, we counted that node as an error. If a node had more than two parents, the node was still counted only once.

$H_{3\text{-}2}$: Multi-refinement nodes These are nodes that have children with multiple refinement relationships. ADT construction rules allow for one refinement per node, in our case either AND or OR [40]. Some students would have two child nodes in an AND relationship, and then a third or fourth child node that was not included in the AND. This was expressed by the AND arc not extending to the connecting edge of these other children. It was clear to us, also based on the node labels, that some children were in an AND relationship, while the remainder was in an OR relationship. We counted each node with multiple refinements regardless of the number of children that node had.

$H_{3\text{-}3}$: Multi-countermeasure attack nodes. These are attack nodes that have multiple countermeasures. ADT construction rules only allow for one countermeasure child per node [40]. If multiple countermeasures are possible, there should first be an intermediate defense node with the single countermeasure edge, and then the multiple countermeasures can be added to the intermediate node in either AND or OR relationship. We counted each time an attack node had more than one countermeasure, regardless of the number of countermeasures attached to that node.

$H_{3\text{-}4}$: Single-child nodes. These are nodes that had only one child node. This type of error is unlike the previous three in that it is not a semantic error. Semantically, there is no issue with having a single child, with multiple semantic representations of ADTs allowing a single child node [40, 53]. A single child node can be shown to be equivalent in both AND and OR refinements, thus technically we can admit attack trees with such refinements as valid. The primary reason for single-child nodes to be included in this section is students were explicitly instructed to avoid using single-child nodes, as the syntactic ADT definition requires that each refined node has at least two children of the same type in either AND or OR relationship, and if only one child is needed, it can be absorbed in the parent node itself. We acknowledge that this argument is flawed for practical reasons, as single child nodes may be necessary to cognitively help the analysts to consider different sub-scenarios and keep the levels of abstraction of a tree consistent across different branches. However, levels of abstraction and the cognitive needs of the analysts were not a focus of our research, while the use of ADTs in a syntactically correct manner was a focus; thus, we have elected to consider single child nodes as an error.

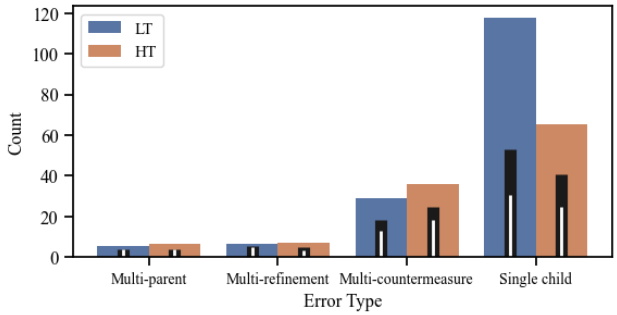

Figure 6: Comparison of of the amount of semantic errors made by LT and HT students.

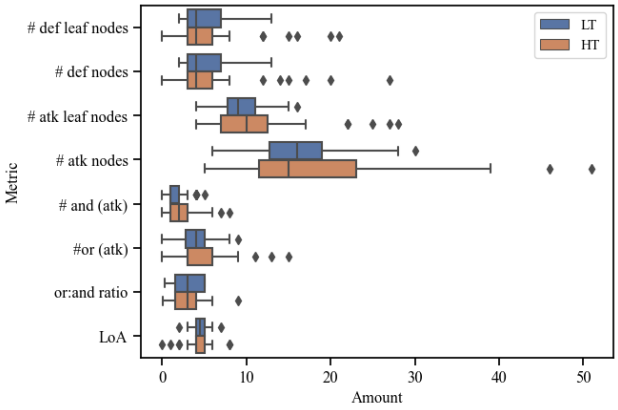

Figure 7: Comparison of LT and HT students' self-drawn ADTs on quantitative metrics.

.

Analysis of common errors. Figure 6 shows the total number of errors present in the ADTs of both LT and HT students. The colored bars show the total error count; if a student made an error three times on the same ADT, then this would be counted three times in the total error count. By contrast, the small black bar inside each colored bar shows the total number of ADTs that have errors in them (the large study consisted of three separate ADTs). The small white bar within the black bar shows the total number of students who made these errors. If the height of the colored and black bars is similar, it indicates that the number of errors present per ADT is closer to 1. If the height of the white and black bar is similar, this indicates that students only made this mistake on one of their three ADTs; a significant height difference here indicates that some students made this mistake on more than one ADT.

In Figure 6, we see that multi-refinement and multi-countermeasure errors are made very infrequently at very similar rates between LT and HT students. For the single-child error ($H_{3\text{-}4}$), we see that a similar number of students

made these errors across similar numbers of ADTs; however, LT students made this error nearly twice as many times as HT students (this difference is statistically significant in ADT1 according to the BM test). The results of our testing can be found in Table 5. Across the other errors $H_{3\text{-}1}$, $H_{3\text{-}2}$, and $H_{3\text{-}3}$, there is no statistically significant difference between LT and HT students, but there is statistically significant equivalence according to TOST.

H_3: We find a significant difference between the groups with respect to single-child nodes. We see evidence of groups' equivalence for all other types of errors. Overall, we find little evidence of a difference and significant evidence of equivalence between LT and HT on common errors.

Conclusions on the actual effectiveness of ADTs. We can conclude that, while we observed a statistically significant difference between the treatment groups for the two types of errors we considered, the majority of the other tested components of the actual effectiveness show the absence of a statistically significant difference between groups' performances. On some measured components, like the quality of self-drawn ADTs, the two treatment groups show statistically significant equivalent behavior. Overall, while both groups show the same lack of understanding of some aspects of ADTs, both groups have demonstrated sufficient mastery of the topic at a similar rate, allowing us to conclude that the actual effectiveness of ADTs is high for both groups.

RQ1: Actual effectiveness of ADTs is high for both groups and does not appear to be affected by technical background.

5.2 RQ2: Effect of the background on PU and PEOU

5.2.1 Perceived Ease of Use (PEOU)

H_4: Self-assessment of understanding. Alongside the check-for-understanding questions we discussed in Section 5.1.1, we asked students if they found a given ADT easy to understand. For the small study, we asked if the provided ADT was easy to understand, and for the large study, we asked if the structure of ADTs was easy to understand. These questions were all in service of the same goal: assessing how students perceived their own understanding of ADTs.

In general, LT and HT students both assessed their understanding similarly (see Figure 8). With the small study questions (labeled `SS-Q#`), the students reported a steady decrease in their confidence in understanding. This is to be expected since, as we describe in Sec. 4.3, there were four ADTs with increasing complexity. The same question was asked about each ADT, and students were less confident with more complex trees.

In Table 6, we can see that none of the understanding Likert questions shows any statistically significant difference between the groups according to the BM test (and some of the questions demonstrate significant equivalence of the groups).

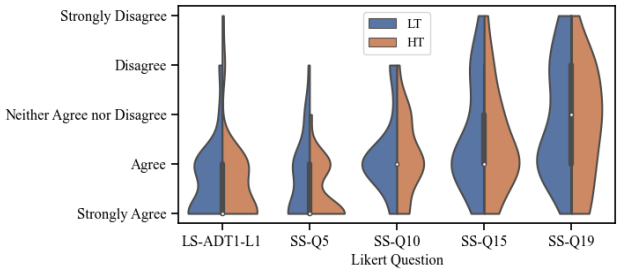

Figure 8: Comparison of LT and HT students on responses to questions self-assessing their understanding of ADTs. The ADTs used in the questions are referenced in Appendices A and B.

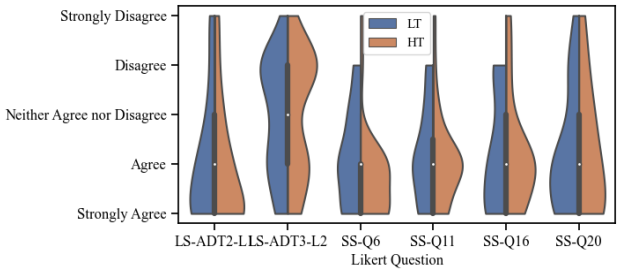

Figure 9: Responses to questions concerning the preference of ADTs to a written description.

H_4: We find evidence of equivalence between LT and HT students on self-assessment of understanding.

H_7: Written description preference. We asked students across every ADT model in the small study and across the final two ADTs in the large study if they prefer ADTs to a written description of an attack scenario. In all questions save one, there was no written description provided; students were asked if their preference was for an ADT that was either presented or to an ADT they had drawn, without an alternative written text about the scenario present (there is one exception to this: the task on building an ADT in the large study where students converted a textual attack scenario description to an ADT). The responses for both LT and HT students were similar: Table 6 shows that there is a statistically significant equivalence between LT and HT for questions in the written description category. This is also demonstrated by Figure 9.

H_7: We find evidence of equivalence between LT and HT students on preference of ADTs to a written description.

Table 6: Table showing the statistics and analysis of answers to Likert questions per hypothesis and treatment group.

Hypothesis	Question[5]	Str. Agree		Agree		Neither		Disagree		Str. Disagree		Average		BM test		TOST	Effect Size
		LT	HT	LT	HT	LT	HT	LT	HT	LT	HT	LT	HT	statistic	$p*m$	$p*m$	Cohen's d
Understanding (H_4)	LS-ADT1-L1	21	26	17	19	2	0	2	2	0	1	1.64	1.6	-0.46	1.0	**1.92e-05**	0.05
	SS-Q5	17	17	13	10	4	1	1	0	0	0	1.69	1.43	-1.25	1.0	**4.30e-03**	0.37
	SS-Q10	4	6	24	14	2	7	5	1	0	0	2.23	2.11	-0.28	1.0	**3.26e-03**	0.15
	SS-Q15	4	4	17	13	4	6	7	3	2	1	2.59	2.41	-0.50	1.0	0.162	0.17
	SS-Q19	2	5	14	6	5	8	8	7	4	1	2.94	2.74	-0.49	1.0	0.395	0.17
Communication (H_5)	LS-ADT3-L3	24	28	11	14	1	3	2	3	2	0	1.68	1.6	0.06	1.0	**1.14e-03**	0.08
Analysis (H_6)	LS-ADT1-L5	19	18	12	17	5	5	5	5	1	2	1.98	2.06	0.45	1.0	**0.013**	0.07
	LS-ADT2-L2	28	13	8	22	3	3	0	5	3	5	1.62	2.31	3.60	**0.041**		0.57
	LS-ADT3-L1	21	16	16	18	2	5	2	5	1	4	1.71	2.23	2.14	1.0	1.0	0.46
Written description (H_7)	LS-ADT2-L1	18	24	12	12	5	6	3	3	4	3	2.12	1.94	-0.68	1.0	0.101	0.14
	LS-ADT3-L2	6	7	11	14	7	3	16	21	2	3	2.93	2.98	0.26	1.0	**0.018**	0.04
	SS-Q6	11	13	13	13	7	0	4	1	0	1	2.11	1.71	-1.89	1.0	0.595	0.41
	SS-Q11	10	7	14	16	6	3	5	1	0	1	2.17	2.04	-0.51	1.0	**0.034**	0.14
	SS-Q16	12	7	10	14	4	2	8	2	0	2	2.24	2.19	-0.05	1.0	0.092	0.04
	SS-Q20	11	10	12	7	3	5	5	3	2	2	2.24	2.26	0.01	0.994	0.146	0.01

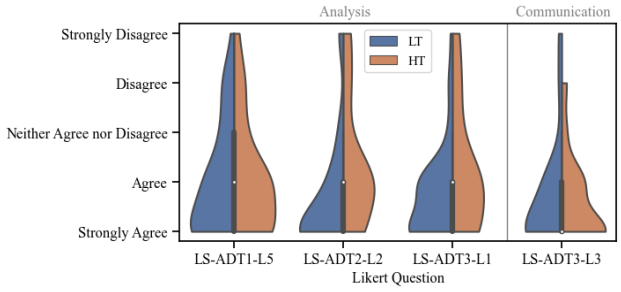

Figure 10: Replies concerning ADTs as a means of analysis and communication.

5.2.2 Perceived Usefulness (PU)

H_5&H_6: ADTs as a means of analysis and communication. We asked three questions about how students perceived ADTs as a means of analysis and one question about how they perceived ADTs as a means of communication. The data shape of responses can be seen in Figure 10.

We have more detailed information in Table 6, where we see strong equivalence between LT and HT students when considering ADTs as a means of communication. Both groups overwhelmingly agree that ADTs are useful as a tool for communicating attack scenarios. We see more agreement than disagreement about ADTs as a means of analysis, however, it is not as strong as the agreement we see for ADTs as a means of communication. Additionally, we see a statistically significant difference on two of the three questions concerning ADTs as a means of analysis. On these two questions, the LT students agreed more than the HT students that ADTs are a useful tool for analysis, with moderate effect sizes (see the Cohen's d values in Table 6).

H_5&H_6: LT and HT students equally perceive ADTs to be useful as a means of communication, but we find some evidence of a difference in their perceptions of ADTs as a means of analysis.

Conclusions on perceptions of ADTs. Overall, we find that the treatment groups largely perceived ADTs to be useful and easy to use (thus, the perceived efficacy is high). **PEOU** is statistically significantly equivalent in both groups, while **PU**, while similar, is not equivalent, and is significantly diverging on one measured aspect (ADTs perceived as a useful means of analysis when designing a model from a textual description).

The only aspect for which we have found a statistically significant difference between the populations revolved around the Likert question concerning ADTs as a means of analysis. One interpretation of this result could be that LT students were introduced to a novel means of organizing information (in the tree structure), which would aid in analysis. In contrast, HT students should have seen tree structures in their previous coursework, which would lead to ADTs not introducing a new means of organizing information. This hypothesis would need further study in order to be tested.

RQ2: We find little evidence that the perceived efficacy of ADTs is affected by technical background. The only hypothesis H_6 for which we have observed a statistically significant difference affects the perception of ADTs in a specific context only, as a means of analysis. The perceived efficacy of ADTs is high for both groups.

5.3 RQ3: Effect of the background on ITU

H_8: Intention to use. We asked two open questions relevant to this hypothesis: `LS-ADT3-W3` asked the participants if they believe ADTs have a place in the cybersecurity field, and if so, where, while `LS-ADT3-W5` asked the students if they would like to see ADTs again. To analyze these questions, we applied a simple coding. If students responded in the

affirmative, we applied a value of 1 to the code "Yes". If the student replied in the negative, we applied a value of 0, and if the student replied in a manner that was open to interpretation, we applied a value of 0.5. We followed this structure for the other codes. The "Communication" code refers to a response describing the utility of ADTs as a means of communication and the "Analysis" code refers to a response describing the utility of ADTs as a means of analysis. These codes are not mutually exclusive, as many responses were coded as neither or both. In this way, we obtain a quantitative evaluation of a qualitative question. The coding guidelines were developed by the two researchers together, and several randomly selected answers from each category were evaluated independently to verify that the assessment aligns. After the establishment of the guidelines, the coding was done by a single coder (the first author of this work).

Table 7 contains the LT and HT averages of these codes. We can see that there is a statistically significant equivalence between the responses. Additionally, we see that both LT and HT students strongly agreed that ADTs have a place in the cybersecurity industry, and fairly strongly agreed that they would like to see ADTs again in the future.

H_8: We find evidence of equivalence between the treatment groups on intention to use ADTs.

Conclusions on intention to use ADTs.

RQ3: The intention to use ADTs is high for both groups and is not affected by technical background.

5.4 RQ4: Effect of the background on creative aspects of ADT design

While the equivalence of two ADTs can be assessed based on a chosen semantics [53], to the best of our knowledge, ADT comparison and metrics of distance between two ADTs have not yet been investigated in the literature. Thus, we opted to compare the self-drawn ADTs based on several quantitative and qualitative metrics.

H_9: Self-drawn ADT comparison. The third task in the large study required the participants to design an ADT for their scenario of choice. As we mentioned in Sec. 4.3, we intentionally did not give any indication of the acceptable size for the tree, as we wanted to assess what differences, if any, would appear between ADTs drawn by LT and HT students when there are no priming restrictions, thereby evaluating the creative component.

We quantitatively assessed the ADTs on 8 metrics: the total number of attack and defense nodes, the number of attack and defense leaf nodes, the number of OR and AND refinements, the ratio of OR to AND refinements, and the levels of abstraction. For these criteria, we define a leaf node as any node that does not have children of the same type. Thus, a node that only has a countermeasure edge would also be defined as a leaf node. We define levels of abstraction to be the greatest depth in the tree, not including countermeasures.

Table 7: Coded responses to written questions concerning the future use of ADTs.

Question	Code	Average		BM Test		TOST	Effect Size
		LT	HT	statistic	$p*m$	$p*m$	Cohen's d
LS-ADT3-W3	Yes	0.92	0.96	0.90	1.0	**3.99e-40**	0.23
LS-ADT3-W3	Communication	0.61	0.52	-0.91	1.0	**3.57e-13**	0.19
LS-ADT3-W3	Analysis	0.44	0.45	0.12	1.0	**3.98e-15**	0.02
LS-ADT3-W5	Yes	0.85	0.73	-1.91	1.0	**1.37e-17**	0.31

We compared LT and HT students' answers on these eight metrics using the BM test and found that there is no statistically significant difference between the ADTs drawn by LT and HT students on any metric. The results of our testing can be found in Table 5. Overall, we find the ADTs drawn by these two groups of students to be remarkably similar (though not equivalent in a statistically significant way).

We qualitatively evaluated the trees using two methods. Besides the quality evaluation results reported in Sec. 5.1.2 that show that both groups designed ADTs with equivalent quality, we processed the labels of the root nodes, taking the main verb from each label (when present) and standardizing these (for example, "steal" and "rob" were considered equivalent in meaning). In Figure 11, we can see the prevalence of verbs across the two groups for all verbs that were present in at least two ADTs. While there are some differences in the verbs, as with the quality analysis, overall, the verbs used in the root nodes are similar between the groups.

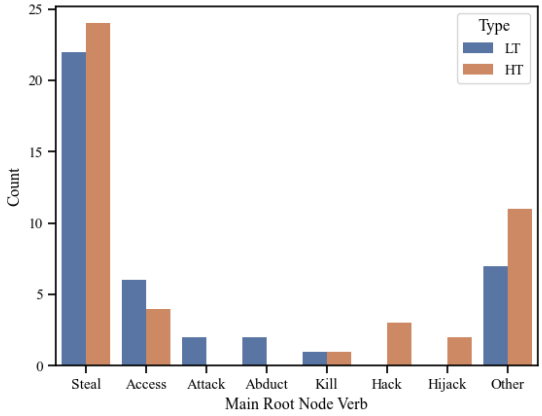

Figure 11: Comparison of main verbs in the root nodes of the self-drawn ADTs.

H_9: We find no evidence of difference between the treatment groups on self-drawn ADTs.

Conclusions on creative expression with ADTs. If ADTs were understood and used differently, we would expect to

see a statistically significant difference in the ADTs created by the two groups on some qualitative or quantitative metric. As we cannot see a significant and material difference, this supports our conclusion that the technical background does not impact how ADTs are created.

RQ4: The creative component in creating ADTs is not affected by technical background.

6 Discussion

Our results show that both participants with a highly technical background and a very limited technical background find ADTs acceptable. Moreover, they find it acceptable in an equal way: for most of the concepts we measured, both treatment groups have shown equivalent behavior and perceptions. They also use ADTs creatively in a similar way, designing models of very similar size and quality. These findings confirm the belief in the security community that attack trees are accessible and easy to learn [32].

Our research sought to establish if the technical background is a potential factor in the adoption of ADTs and, specifically, if the participants with a very limited technical background would be disadvantaged in using ADTs. The cyber security industry consists of people with widely varying backgrounds [2]. In particular, TM is done by people with diverse skillsets and objectives [69, 88]. If a technical background were to impact the acceptability of ADTs, then this could be a reason for not recommending them to be used.

Lallie et al. found that there was a difference between participants with and without a computer science background when using both fault trees and attack graphs in a similar study design to ours [48]. This result indicated that TM stakeholders who do not possess a highly technical background (e.g., managers) might potentially be disadvantaged if the team uses attack trees for threat modeling. However, it is reasonable to expect that people involved in TM, even managers, might possess at least a limited technical background as they are exposed to software development and/or IT security risk management activities. Our study concludes that ADTs are highly acceptable for such TM stakeholders and do not disadvantage them compared to threat modelers with a highly technical background.

We believe the difference in the results between our study and [48] to be due to two major differences in the study design and methodology. First, we intentionally used ADT examples that are equally accessible to all participants, attempting to remove any specifically technical jargon from our study questions. For the small study, all of our examples were pulled from papers on ADTs and we specifically looked for ADTs without complex technical labels, i.e., accessible to people with diverse backgrounds. This approach was inspired by Lallie et al. [48] who used fault trees from previous works. However, two of the fault trees they used are arguably difficult to understand to a layperson, using terms such as "`sshd_bof(1,2)`", which might be more accessible to someone with a computer science background. As such, their finding that those with a computer science background can use these models more effectively may speak more to the comprehensibility of the language used in their study. While subsequent studies in the attack trees context are required to test this, previous research has shown that technical language does affect comprehension: e.g., Bravo-Lillo et al. [7] have shown that technical terms complicate comprehension of security warnings for non-expert users, compared to security experts. In the TM context, Ingalsbe et al. [29] explicitly mention that the vocabulary of threat modeling is IT-biased, impeding communication with internal business customers, while Verreydt et al. [88] also acknowledge the TM challenges related to communication and the used language.

One important conclusion that we can draw from our study is that short training is sufficient for making ADTs equally acceptable for users with high and limited technical backgrounds. Threat modeling method training is an established practice in organizations [88], and it can be recommended to improve the outcomes and facilitate the process [14, 73]. To help implement training on ADTs in organizations, we share the slides of our training lecture along with a detailed description in our supplementary data material [64].

Another relevant observation that we can make from the analysis of the related literature (Sec. 3) is that there are no established protocols for empirical studies of TM methods. While the studies frequently follow reputable frameworks like TAM and MEM, the operationalization of the frameworks' constructs differs a lot. One of the reasons behind this might be the diversity of TM methods themselves. Still, it would be useful to systematize the experiences reported so far and develop guidelines for executing such studies.

7 Limitations

Our study has several limitations that we acknowledge in this section.

Study design. One of the most significant limitations of our research was the lack of standardization of testing. Unlike Opdahl and Sindre [58], where students completed assignments in a testing facility, our study consisted of students completing assignments at home with a month to complete the tasks. As such, we cannot exclude external factors from having an effect, and we could not measure data related to actual efficiency in the MEM. However, given that both populations of students were given the same conditions (training, access to resources, and time), we believe that our study design is sufficient to examine the possible effects of technical background on ADT acceptability. Additionally, this is in line with other threat model evaluation studies, such as [9, 48].

As an established practice in this type of study (see Section 4.6, we provided training on the method to our partic-

ipants. It might be the case that the training eclipsed any innate differences between the groups. However, if this is the case, it would suggest that relatively short training is a viable means to ensure that ADTs are accessible to stakeholders with varying technical backgrounds.

Attack trees are amenable to represent physical, cyber-physical, and purely cyber scenarios [68]. The first attack tree outlined by Schneier in [65, Fig. 1] represents a physical attack to open up a safe, while an attack tree from Mauw and Oostdijk captures a free lunch scenario [53, Fig. 1]. We aim to evaluate the acceptability of ADTs outside of a domain-specific context (cyber) and our ADTs were constructed in such a way that domain knowledge is not necessary to understand them. As mentioned previously, it is recommended in the TM literature to be considerate of the used terminology to improve conveyance [29]. However, in practice, some modeled attacks can be highly complex and require advanced security expertise. We welcome future studies that will measure the effect of the technical terms used in ADT models on the acceptability of the method for users with varying technical backgrounds.

Participants. Our sample size of 102 participants in total is quite substantial and consistent with the sample sizes of similar studies evaluating threat models, which have 87 [34], 63 [48, 58], 49 [8], 42 [36], 28 [47], and 25 [9] participants. Still, our sample might be biased, as the participants come from the same university and the majority of them have the same country of origin.

Another limitation of the sample is that students may not be representative of industry practitioners as a whole. Using students as study participants for threat model evaluation is standard practice with such studies [34, 36, 47, 48, 58, 62]. A study by Karpati et al. consisting of interviews with industry practitioners was able to confirm the results found in a previous study using student participants [35], which lends itself to the idea that generally student participants can speak to the acceptability of threat models. These results were reinforced by, for example, Naiakshina et al. [55], Salman et al. [61], Svahnberg et al. [76] and Yakdan et al. [94] who found that within the cyber security and software engineering contexts, treatment effects on computer science students hold for professionals. Based on these results, we believe that our sample of students is reflective of practitioners.

It might be that our participants self-selected for cybersecurity-related studies, and thus, they might be more geared toward cybersecurity than the general population. This would make them more representative of a cybersecurity practitioner (who is also geared towards security) than the general population. Threat modelers will likely receive hands-on experience and training on security-related topics, and some of them might be interested in security, but not all participants in threat modeling are necessarily geared towards security [69, 88]. Future studies should aim to examine this link with personal preferences.

A component of our study (the large study) was graded. This might have biased the students's answers, especially regarding their perceptions, if they wanted to please the graders. We tried to mitigate this by repeatedly informing the participants that perception questions were not evaluated as a component of their grades. Additionally, our core interest is in finding differences between the two groups. If one group perceived ADTs substantially differently than the other group, we would likely still see the effect in the data. We note that some participants did report low perceptions of ADTs, and both groups did this at relatively similar rates.

Finally, as participation in our study was voluntary, it is possible that our students self-selected, and only students who had a high level of understanding of ADTs elected to participate in the study. This is confirmed by the grade difference between the participants and non-participants as shown in Table 2. However, we can see that the final grades between the two treatment groups are equivalent. This implies that stronger students were self-selecting in similar proportions in both cohorts, and thus there was no difference between the two groups. We welcome future studies with more diverse population samples, preferably from industry practitioners that will independently examine the effect of technical (computer science) background on attack tree acceptability, especially for participants without technical background.

8 Conclusions

ADTs are a valuable threat modeling method, recognized for its accessibility [32, 68]. We investigated whether ADTs are acceptable for users with a very limited technical background using MEM [54]. Overall, we find sufficient evidence to support that ADTs are equally highly acceptable for users with a very limited technical background and users with a substantial technical background. Moreover, attack trees designed by these two types of users show similar patterns in terms of the size of the trees, types of attacks modeled, and quality of the trees. We conclude that ADTs are suitable as a threat modeling method for diverse groups of stakeholders. Further studies should look into measuring the exact effects of the technical terms on attack tree acceptability, making such models more accessible to practitioners without a technical background, and assessing different training regimens.

Acknowledgements

We thank Kate Labunets, the anonymous reviewers, and our shepherd for their helpful feedback on this paper.

This research has been partially supported by the Dutch Research Council (NWO) under the project "Cyber Security by Integrated Design (C-SIDe)" (NWA.1215.18.008).

9 Ethical Considerations

Several important ethical considerations are relevant to this research. We now outline how we considered them during the study design and execution. The Science Ethics Review Board at Leiden University reviewed and approved our study.

Our study involved human participants, and, moreover, these participants were students taking a course taught by the authors of this paper. This introduces ethical concerns due to the dual role of the authors being both in the research team and responsible for the education of the students in the course. We have done our utmost to ensure that the students were not pressured to participate in this study and that they did not perceive being pressured or nudged to participate. Below we discuss the multiple safeguards in this regard that we introduced.

Students were informed of the study objectives and design and then were asked to fill in and sign an informed consent form. In this form, they could choose to provide consent for their responses to the assignment to be included in the study and for the data they submit to be used for research purposes in an anonymized format. The consent forms were collected blindly for the teachers.

We made it clear to the students that the assignment was a mandatory, graded course component, but participation in the study was entirely optional and would not affect their grades. Students were informed that teaching assistants would grade their submissions according to defined grading rubrics, and teaching assistants had no knowledge of who had elected to participate in the study. The grading rubric did not account for study participation in any way; thus, the grade was not influenced by (non-)participation. Finally, students were told that they could withdraw their consent at any time. We informed students that we would not collect responses until one month after final grades were submitted (and were no longer able to be modified). For any students who were still concerned, we offered the protocol of initially providing consent to participate, and withdrawing said consent after final grades were submitted. Withdrawing consent required filling out an online form, which we provided with the intention of making it as easy and straightforward as possible for students who no longer wished for their responses to be included.

We further provided resources when presenting the research to students, in the participation consent form, and in the introduction to the assignment that students could reach out to if they were concerned about any negative effects resulting from the study. These resources included the contact information for the relevant Ethical Review Board, the university ombudsman's office, and a student counselor. To our knowledge, no students reached out to these resources with questions or concerns about the study.

The assignments were submitted via the university's learning management software (LMS), which is a standard and accepted practice for course assignments. For the students who opted to participate in the study, once the data processing started, students were assigned a "participant number" which was stored in a password-protected reference list on the first author's university-issue computer. The participant number was used to anonymize the data for analysis. All other study data was pulled directly off of the LMS into a spreadsheet for further processing. The data collected and analyzed for the study did not contain any personal information.

Participants were not provided compensation for their participation in the study. As the assignment was a mandatory course component, it would have been inappropriate to compensate students for completing it. We designed our study following the Menlo Report's guidelines for ethical research [37] and we strived to carefully balance the benefits of the study against potential harms. The assignment itself is useful for students as it helps them learn about important concepts within cybersecurity and develops their analysis skills. We also believe that our students benefited from the study because they experienced the scientific process in the computer science domain. Moreover, the findings from this study allow us to further improve our research-based teaching, which will benefit future generations of students. It is important for the community that teachers can confidently teach attack trees to students without a substantial computer science background. Our personal experience told us that attack trees are accessible to such audiences, but only via doing a properly designed study can we be confident about this.

We believe that the potential harm to our students, on the other hand, is limited, because we actively emphasized that non-participation does not entail any consequences for the course and we placed multiple safeguards to protect the students. Participation in the study did not entail any extra effort for the students (because they would still be doing the work as a course assignment).

Our Ethics Review Board agreed with this risk-benefit analysis and approved our study (ref. 2022-016).

10 Open Science

The full set of anonymized, qualitative, perception data is shared alongside this work in our supplementary data material [64]. This includes all of the values used to calculate the results presented in this paper, as well as additional elements of data that were ultimately excluded. The data are shared in a `.csv` format. To enable verification, we provide the code we used to analyze the data and generate the results presented in this paper. This code is provided as a Jupyter notebook.

Further, we provide the dataset of ADTs generated by participants. All ADTs are provided as `.png` files, with trees without structural errors provided as `.xml` files in the ADTool schema. Finally, we also share the slides used in the training of the study alongside a summary of the training and indicative time amounts spent on each part of the training. All these materials are available in [64].

References

[1] S. Abrahão, E. Insfran, J. A. Carsí, and M. Genero. Evaluating Requirements Methods based on User Perceptions: A Family of Experiments. *Information Sciences*, 181(16):3356–3378, 2011.

[2] R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems. 3rd Edition*. John Wiley & Sons, November 2020.

[3] A. Apvrille and M. Pourzandi. Secure Software Development by Example. *IEEE Security & Privacy*, 3(4):10–17, 2005.

[4] K. Bernsmed, D. S. Cruzes, M. G. Jaatun, and M. Iovan. Adopting threat modelling in agile software development projects. *Journal of Systems and Software*, 183:111090, 2022.

[5] M. Bishop. Computer security: Art and science. 2nd edition, 2019.

[6] D. J. Bodeau, C. D. McCollum, and D. B. Fox. Cyber threat modeling: Survey, assessment, and representative framework, https://www.mitre.org/sites/default/files/2021-11/prs-18-1174-ngci-cyber-threat-modeling.pdf, 2018.

[7] C. Bravo-Lillo, L. F. Cranor, J. Downs, and S. Komanduri. Bridging the Gap in Computer Security Warnings: A Mental Model Approach. *IEEE Security & Privacy*, 9(2):18–26, 2010.

[8] G. Broccia, M. H. ter Beek, A. L. Lafuente, P. Spoletini, A. Fantechi, and A. Ferrari. Evaluating the understandability and user acceptance of Attack-Defense Trees: Original experiment and replication. *Information and Software Technology*, 178:107624, 2025.

[9] G. Broccia, M. H. ter Beek, A. Lluch Lafuente, P. Spoletini, and A. Ferrari. Assessing the Understandability and Acceptance of Attack-Defense Trees for Modelling Security Requirements. In *Requirements Engineering: Foundation for Software Quality*, pages 39–56. Springer.

[10] E. Brunner and U. Munzel. The Nonparametric Behrens-Fisher Problem: Asymptotic Theory and a Small-Sample Approximation. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 42(1):17–25, 2000.

[11] M. Brunner, C. Sauerwein, M. Felderer, and R. Breu. Risk Management Practices in Information Security: Exploring the Status Quo in the DACH Region. *Computers & Security*, 92:101776, 2020.

[12] A. Buldas, O. Gadyatskaya, A. Lenin, S. Mauw, and R. Trujillo-Rasua. Attribute Evaluation on Attack Trees with Incomplete Information. *Computers & Security*, 88:101630, January 2020.

[13] B. Chen, T. Wu, Y. Zhang, M. B. Chhetri, and G. Bai. Investigating Users' Understanding of Privacy Policies of Virtual Personal Assistant Applications. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, pages 65–79, 2023.

[14] D. S. Cruzes, M. G. Jaatun, K. Bernsmed, and I. A. Tøndel. Challenges and Experiences with Applying Microsoft Threat Modeling in Agile Development Projects. In *2018 25th Australasian Software Engineering Conference (ASWEC)*, pages 111–120. IEEE, 2018.

[15] F. D. Davis. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3):319–340, 1989.

[16] J. L. De La Vara, B. Marín, C. Ayora, and G. Giachetti. An Empirical Evaluation of the Use of Models to Improve the Understanding of Safety Compliance Needs. 126:106351.

[17] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16(1):3–32, 2011.

[18] J. Dev, B. Rashidi, and V. Garg. Models of Applied Privacy (MAP): A Persona Based Approach to Threat Modeling. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2023.

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

[20] M. H. Diallo, J. Romero-Mariona, S. E. Sim, T. A. Alspaugh, and D. J. Richardson. A Comparative Evaluation of Three Approaches to Specifying Security Requirements. In *12th Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 1–10, 2006.

[21] M. W. Fagerland and L. Sandvik. The Wilcoxon–Mann–Whitney test under scrutiny. *Statistics in medicine*, 28(10):1487–1497, 2009.

[22] D. Granata and M. Rak. Systematic analysis of automated threat modelling techniques: Comparison of open-source tools. *Software Quality Journal*, 32(1):125–161, 2024.

[23] D. Gritzalis, G. Iseppi, A. Mylonas, and V. Stavrou. Exiting the Risk Assessment maze: A meta-survey. *ACM Computing Surveys (CSUR)*, 51(1):1–30, 2018.

[24] Z. Hanusz, J. Tarasinska, and W. Zielinski. Shapiro–Wilk Test with Known Mean. *REVSTAT-Statistical Journal*, 14(1):89–100, 2016.

[25] I. Hogganvik and K. Stolen. Risk Analysis Terminology for IT-systems: Does it match intuition? In *2005 International Symposium on Empirical Software Engineering, 2005.*

[26] S. Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[27] J. B. Hong, D. Kim, C.-J. Chung, and D. Huang. A survey on the usability and practical applications of Graphical Security Models. *Comput. Sci. Rev.*, 2017.

[28] M. Howard and S. Lipner. Inside the Windows Security Push. *IEEE Security & Privacy*, 1(1):57–61, 2003.

[29] J. A. Ingalsbe, L. Kunimatsu, T. Baeten, and N. R. Mead. Threat Modeling: Diving into the Deep End. *IEEE Software*, 25(1):28–34, 2008.

[30] A.-M. Jamil, L. Ben Othmane, and A. Valani. Threat Modeling of Cyber-Physical Systems in Practice. In *International Conference on Risks and Security of Internet and Systems*, pages 3–19. Springer, 2021.

[31] A. Jawad, H. Assal, and J. Jaskolka. "I'm Getting Information that I Can Act on Now": Exploring the Level of Actionable

Information in Tool-generated Threat Reports. In *Proceedings of the 2024 European Symposium on Usable Security*, pages 172–186, 2024.

[32] John. P Mello. Lessons in threat modeling: How attack trees can deliver appsec by design, https://www.reversinglabs.com/blog/lessons-in-threat-modeling-how-attack-trees-can-secure-your-software-design, 2024.

[33] J. D. Karch. Psychologists Should Use Brunner-Munzel's Instead of Mann-Whitney's U Test as the Default Nonparametric Procedure. *Advances in Methods and Practices in Psychological Science*, 4(2), 2021.

[34] P. Karpati, A. L. Opdahl, and G. Sindre. Experimental Comparison of Misuse Case Maps with Misuse Cases and System Architecture Diagrams for Eliciting Security Vulnerabilities and Mitigations. *Journal of Systems and Software*, 104:90–111, June 2015.

[35] P. Karpati, Y. Redda, A. L. Opdahl, and G. Sindre. Comparing attack trees and misuse cases in an industrial setting. *Information and Software Technology*, 56(3):294–308, March 2014.

[36] V. Katta, P. Karpati, A. L. Opdahl, C. Raspotnig, and G. Sindre. Comparing Two Techniques for Intrusion Visualization. In *The Practice of Enterprise Modeling*, LNBIP, pages 1–15. Springer, 2010.

[37] E. Kenneally and D. Dittrich. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. 2012.

[38] B. Kordy, S. Mauw, S. Radomirovic, and P. Schweitzer. Attack–Defense Trees. *Journal of Logic and Computation*, 24(1):55–87, 2014.

[39] B. Kordy, P. Kordy, S. Mauw, and P. Schweitzer. ADTool: Security Analysis with Attack–Defense Trees. In *Quantitative Evaluation of Systems*, LNCS, pages 173–176. Springer, 2013.

[40] B. Kordy, S. Mauw, S. Radomirović, and P. Schweitzer. Foundations of Attack–Defense Trees. In *Formal Aspects of Security and Trust*, Lecture Notes in Computer Science, pages 80–95, Berlin, Heidelberg, 2011. Springer.

[41] B. Kordy, L. Piètre-Cambacédès, and P. Schweitzer. DAG-Based Attack and Defense Modeling: Don't Miss the Forest for the Attack Trees. *Computer Science Review*, 13:1–38, 2014.

[42] A. Kudriavtseva and O. Gadyatskaya. Secure Software Development Methodologies: A Multivocal Literature Review. *arXiv preprint arXiv:2211.16987*, 2022.

[43] K. Labunets, F. Massacci, F. Paci, M. Ragosta, B. Solhaug, K. Stølen, and A. Tedeschi. A First Empirical Evaluation Framework for Security Risk Assessment Methods in the ATM Domain. *SIDs 2014 - Proceedings of the SESAR Innovation Days*, 2014.

[44] K. Labunets. No Search Allowed: What Risk Modeling Notation to Choose? In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '18, pages 1–10. ACM.

[45] K. Labunets. *Security Risk Assessment Methods: An Evaluation Framework and Theoretical Model of the Criteria Behind Methods' Success*. PhD thesis, University of Trento, 2016.

[46] K. Labunets, F. Massacci, F. Paci, S. Marczak, and F. M. de Oliveira. Model Comprehension for Security Risk Assessment: An Empirical Comparison of Tabular vs. Graphical Representations. *Empirical Software Engineering*, 22:3017–3056, 2017.

[47] K. Labunets, F. Massacci, F. Paci, and L. M. S. Tran. An Experimental Comparison of Two Risk-Based Security Methods. In *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 163–172, 2013.

[48] H. S. Lallie, K. Debattista, and J. Bal. An Empirical Evaluation of the Effectiveness of Attack Graphs and Fault Trees in Cyber-Attack Perception. *IEEE Transactions on Information Forensics and Security*, 13(5):1110–1122, 2017.

[49] H. Levene. Robust Tests for Equality of Variances. *Contributions to probability and statistics*, pages 278–292, 1960.

[50] I. Levy. The future of telecoms in the UK, https://www.ncsc.gov.uk/blog-post/the-future-of-telecoms-in-the-uk, 2020.

[51] S. Lipner and M. Howard. Inside the Windows Security Push: A Twenty-Year Retrospective. *IEEE Security & Privacy*, 21(2):24–31, 2023.

[52] S. Mauw. RFID Communication Block. https://satoss.uni.lu/projects/atrees/trees/block_communication.pdf.

[53] S. Mauw and M. Oostdijk. Foundations of Attack Trees. In *Information Security and Cryptology - ICISC 2005*, LNCS. Springer, 2006.

[54] D. Moody. The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods. *ECIS 2003 Proceedings*, January 2003.

[55] A. Naiakshina, A. Danilova, E. Gerlitz, and M. Smith. On Conducting Security Developer Studies with CS Students: Examining a Password-Storage Study with CS Students, Freelancers, and Company Developers. In *Proc. of the CHI Conference on Human Factors in Computing Systems*. ACM, April 2020.

[56] National Cyber Security Centre. Risk management. Using attack trees to understand cyber security risk, https://www.ncsc.gov.uk/collection/risk-management/using-attack-trees-to-understand-cyber-security-risk, 2023.

[57] NIST. NIST SP800-30r1 Guide for conducting risk assessments, https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf, 2012.

[58] A. L. Opdahl and G. Sindre. Comparing attack trees and misuse cases in an industrial setting. *Information and Software Technology*, 51(5):916–932, May 2009.

[59] OWASP. OWASP Threat modeling project https://owasp.org/www-project-threat-model/, 2024.

[60] V. Saini, Q. Duan, and V. Paruchuri. Threat Modeling Using Attack Trees. *Journal of Computing Sciences in Colleges*, 23(4):124–131, 2008.

[61] I. Salman, A. T. Misirli, and N. Juristo. Are Students Representatives of Professionals in Software Engineering Experiments? In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 1, May 2015.

[62] R. Scandariato, K. Wuyts, and W. Joosen. A descriptive study of Microsoft's threat modeling technique. *Requirements Engineering*, 20:163–180, 2015.

[63] N. D. Schiele and O. Gadyatskaya. A Novel Approach for Attack Tree to Attack Graph Transformation. In *International Conference on Risks and Security of Internet and Systems (CRISIS 2021)*, volume 13204 of *LNCS*, pages 74–90. Springer, 2021.

[64] N. D. Schiele and O. Gadyatskaya. A limited technical background is sufficient for attack-defense tree acceptability: Dataset https://doi.org/10.5281/zenodo.14717342, 2025.

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

[66] B. Schneier. *Secrets and Lies: Digital Security in a Networked World*. John Wiley & Sons, 2000.

[67] D. Schuirmann. On hypothesis-testing to determine if the mean of a normal-distribution is contained in a known interval. In *Biometrics*, volume 37, pages 617–617, 1981.

[68] N. Shevchenko, T. A. Chick, P. O'Riordan, T. P. Scanlon, and C. Woody. Threat Modeling: A Summary of Available Methods. Technical report, Carnegie Mellon University Software Engineering Institute Pittsburgh United States, 2018.

[69] A. Shostack. Experiences Threat Modeling at Microsoft. *MODSEC Workshop at MODELS*, 2008.

[70] A. Shostack. *Threat Modeling: Designing for Security*. John Wiley & Sons, 2014.

[71] T. Sonderen. A Manual for Attack Trees. Master's thesis, University of Twente, 2019.

[72] W. Stallings and L. Brown. *Computer security: Principles and practice. 4th Edition*. Pearson, 2018.

[73] R. Stevens, D. Votipka, E. M. Redmiles, C. Ahern, P. Sweeney, and M. L. Mazurek. The Battle for New York: A Case Study of Applied Digital Threat Modeling at the Enterprise Level. In *27th USENIX Security Symposium*, pages 621–637, 2018.

[74] G. Stringhini. Adversarial behaviours knowledge area. Version 1.0.1. In *Cyber Security Body of Knowledge (CyBOK)*. 2021.

[75] W. Sun, L. Lv, Y. Su, and X. A. Wang. Cyber-Attack Risks Analysis Based on Attack-Defense Trees. In *Advances in Internetworking, Data & Web Technologies*, LNDECT, pages 667–678, Cham, 2018.

[76] M. Svahnberg, A. Aurum, and C. Wohlin. Using Students as Subjects – An Empirical Evaluation. In *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '08, New York, NY, USA, October 2008. ACM.

[77] I. Tarandach and M. J. Coles. *Threat Modeling*. O'Reilly Media, Inc., 2020.

[78] M. Tatam, B. Shanmugam, S. Azam, and K. Kannoorpatti. A review of threat modelling approaches for APT-style attacks. *Heliyon*, 7(1), 2021.

[79] R. E. Thompson, M. McLaughlin, C. Powers, and D. Votipka. "There are rabbit holes I want to go down that I'm not allowed to go down": An Investigation of Security Expert Threat Modeling Practices for Medical Devices. In *33rd USENIX Security Symposium*, pages 4909–4926, 2024.

[80] I. A. Tøndel, M. G. Jaatun, D. Cruzes, and T. D. Oyetoyan. Understanding challenges to adoption of the Protection Poker software security game. In *Computer Security: ESORICS 2018 International Workshops, CyberICPS 2018 and SECPRE*, pages 153–172. Springer, 2019.

[81] A.-D. Tran, K. Yskout, and W. Joosen. Threat Modeling: A Rough Diamond or Fool's Gold? In *European Conference on Software Architecture*, pages 120–129. Springer, 2023.

[82] R. Trentinaglia, S. Merschjohann, M. Fockel, and H. Eikerling. Eliciting Security Requirements – An Experience Report. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 351–365. Springer, 2023.

[83] K. Tuma, G. Calikli, and R. Scandariato. Threat Analysis of Software Systems: A Systematic Literature Review. *Journal of Systems and Software*, 144:275–294, 2018.

[84] K. Tuma and R. Scandariato. Two Architectural Threat Analysis Techniques Compared. In *Proceedings of the 12th European Conference on Software Architecture (ECSA)*, pages 347–363. Springer, 2018.

[85] D. Van Landuyt and W. Joosen. A descriptive study of assumptions in STRIDE security threat modeling. *Software and Systems Modeling*, pages 1–18, 2022.

[86] P. C. van Oorschot. *Computer security and the internet: Tools and jewels. 1st Edition*. Springer, 2020.

[87] B. Verhulst and M. C. Neale. Best Practices for Binary and Ordinal Data Analyses. *Behavior Genetics*, 51(3):204–214, 2021.

[88] S. Verreydt, K. Yskout, L. Sion, and W. Joosen. Threat modeling state of practice in Dutch organizations. In *Twentieth Symposium on Usable Privacy and Security (SOUPS 2024)*, pages 473–486, 2024.

[89] C. Weir, I. Becker, and L. Blair. Incorporating software security: using developer workshops to engage product managers. *Empirical Software Engineering*, 28(2):21, 2023.

[90] W. Wideł, M. Audinot, B. Fila, and S. Pinchinat. Beyond 2014: Formal Methods for Attack Tree–based Security Modeling. *ACM Computing Surveys (CSUR)*, 52(4):1–36, 2019.

[91] T. Wu, R. Zhang, W. Ma, S. Wen, X. Xia, C. Paris, S. Nepal, and Y. Xiang. What risk? I don't understand. An Empirical Study on Users' Understanding of the Terms Used in Security Texts. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pages 248–262. ACM.

[92] K. Wuyts, R. Scandariato, and W. Joosen. Empirical evaluation of a privacy-focused threat modeling methodology. *Journal of Systems and Software*, 96:122–138, 2014.

[93] W. Xiong and R. Lagerström. Threat modeling – A systematic literature review. *Computers & Security*, 84:53–69, 2019.

[94] K. Yakdan, S. Dechand, E. Gerhards-Padilla, and M. Smith. Helping Johnny to Analyze Malware: A Usability-Optimized Decompiler and Malware Analysis User Study. In *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016.

[95] K. Yskout, T. Heyman, D. Van Landuyt, L. Sion, K. Wuyts, and W. Joosen. Threat modeling: from infancy to maturity. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results*, pages 9–12, 2020.

A Small Study

ADT 1. The ADT for the following questions was created by Buldas et al.İt can be found on page four labeled as Figure 1 [12].

`SS-Q2`: How many leaf nodes are in this ADT?

`SS-Q3`: How many root nodes are in this ADT?

`SS-Q4`: How many different attack vectors are represented by this ADT?

`SS-Q5`: The attack tree is easy to understand

`SS-Q6`: I prefer this attack tree to a written description of this attack

ADT 2. The ADT for the following questions is shown in Figure 1.

`SS-Q7`: How many attack leaf nodes are in this ADT?

`SS-Q8`: How many different attack vectors are represented by this ADT?

`SS-Q9`: How many attack vectors do not have a defense?

`SS-Q10`: The attack tree is easy to understand

`SS-Q11`: I prefer this attack tree to a written description of this attack

ADT 3. The ADT for the following questions was created by Mauw and Oostdijk [52].

`SS-Q12`: How many attack vectors do not have a defense?

`SS-Q13`: How many different attack vectors are represented by this ADT?

`SS-Q14`: How many levels of abstraction are present in this ADT?

`SS-Q15`: The attack tree is easy to understand

`SS-Q16`: I prefer this attack tree to a written description of this attack

ADT 4. The ADT for the following questions was created by Kordy et al. can be found on page 58 of that work labeled Figure 1 [38].

`SS-Q17`: Is the overall goal kept? Why or why not?

`SS-Q18`: How many levels of abstraction are present in this ADT?

`SS-Q19`: The attack tree is easy to understand

`SS-Q20`: I prefer this attack tree to a written description of this attack

B Large Study

ADT 1: Assembling ADTs

The following attack **leaf** nodes are provided. The overall goal of this scenario (and thus the root node of the tree) is **Rob bank**. Assemble an attack-defense tree using these leaf nodes. Do not add any additional leaf nodes. You may add any intermediary nodes you wish.

Attack leaf nodes: Hire Outright; Promise part of the stolen money; Threaten insiders; Buy tools; Steal tools; Gain Access; Walk through front door; Locate start of tunnel; Find direction to tunnel.

Defense leaf nodes: Personnel Risk Management; Check employee financial situation.

Likert Questions.

`LS-ADT1-L1`: I find the structure of attack tree easy to understand

`LS-ADT1-L2`: Given all the nodes of an attack tree, it is easy for me to assemble the tree

`LS-ADT1-L3`: Given only the leaf nodes of an attack tree, it is easy for me to assemble the tree.

`LS-ADT1-L4`: I would rather define my own intermediary nodes

`LS-ADT1-L5`: The process of assembling the attack tree helped me better understand the attack scenario.

Short Response Questions.

`LS-ADT1-W1`: What did you find most difficult about this task? Why?

`LS-ADT1-W2`: How did you go about solving this task? What was your methodology?

ADT 2: Building ADTs

The following text scenario is provided for you. Please create a complete attack defense tree **of this scenario**. **Do not add extra information that is not in the scenario**. Try to encapsulate the entire scenario with an attack-defense tree (don't leave any aspect of the attack scenario out).

Scenario: The goal is to open a safe. To open the safe, an attacker can pick the lock, learn the combination, cut open the safe, or install the safe improperly so that he can easily open it later. Some models of safes are such that they cannot be picked, so if this model is used, then an attacker is unable to pick the lock. There are also auditing services to check if safes and other security technology is installed correctly. To learn the combination, the attacker either has to find the combination written down or get the combination from the safe owner. If the password is such that the safe owner can remember it, then the safe owner would not need to write it down.

Likert Questions.

`LS-ADT2-L1`: I prefer reading attack trees to text descriptions of attacks.

`LS-ADT2-L2`: The process of building the attack tree helped me better understand the attack scenario.

Short Response Questions.

`LS-ADT2-W1`: What did you find most difficult about this task? Why?

`LS-ADT2-W2`: How did you go about building the ADT? What was your methodology?

`LS-ADT2-W3`: What was the first node you added to your tree?

ADT 3: Creating ADTs

Construct an attack defense tree of a scenario of your choice. Your tree should be complete (covers all reasonable attack scenarios) and reasonably large.

Likert Questions.

`LS-ADT3-L1`: The process of creating the attack tree helped me better understand the attack scenario I selected

`LS-ADT3-L2`: I feel I could have achieved the same understanding by writing a text description of the attack.

`LS-ADT3-L3`: The ADT I created would help me communicate my threat scenario.

Short Response Questions.

`LS-ADT3-W1`: What did you find easy about using ADTs?

`LS-ADT3-W2`: What did you find difficult about using ADT?

`LS-ADT3-W3`: Do you think ADTs have a place in the cybersecurity industry? If so, where? If not, why not?

`LS-ADT3-W4`: What aspects, if any, do you think are missing from ADTs?

`LS-ADT3-W5`: Do you hope to encounter ADTs in the future?