WiP: Where's Eve? Evaluating Student Threat Modeling Performance and Perceptions

Carson Powers, Nickolas Gravel, Maxwell Mitchell, and Daniel Votipka
Tufts University
{carson.powers, nickolas.gravel, maxwell.mitchell, daniel.votipka}@tufts.edu

Abstract

The process of identifying threats and developing mitigation strategies—referred to as Threat Modeling (TM)—is an important step in the early phases of secure system development. Despite being highly recommended and sometimes required by federal regulation, there has been limited work investigating developers' ability to perform this task. In particular, we focus on students to understand how well prepared they are upon entering the workforce and guide future work to improve education in this domain. To answer this question, we conducted preliminary semi-structured interviews asking students to complete a TM exercise while describing their thought process aloud. Our initial results indicate students struggle to identify technically detailed threats and that the concept of repudiation is particularly confusing to students. We conclude with recommendations to guide future work.

1 Introduction

The process of identifying threats, calculating the risk of each threat, and developing mitigation strategies is known as Threat Modeling (TM). TM is highly recommended for software development and related industries [11], it has been shown to support improved outcomes in enterprise settings [12], and, in some cases, it is required by federal regulation [5]. However, TM is not commonly taught in undergraduate computer science (CS) programs [4, 6]. Additionally, common CS curriculum guidelines and recommendations have only recently added TM [1], include TM in very limited contexts [9], or do not mention TM at all [7]. Therefore, it is not clear whether early-career professionals are prepared to effectively perform TM as needed. In fact, organizations frequently report having to give entering developers significant on-the-job training to familiarize them with TM [14]. Further, this on-the-job training is often not available, as many organizations do not prioritize security or have the expertise necessary to provide this training [3]. Therefore, we seek to understand students' ability to perform TM to determine how to best tailor education and tool support to meet their needs as they perform this essential task. Specifically, we consider the following research questions:

RQ1: Can students find threats in a system, and what are common misconceptions?

RQ2: Are there particular threat modeling concepts students struggle with?

Answering these questions will identify issues in students' processes, mental models, and knowledge-base and suggest possible tailored education or process aides (e.g., checklists, automated support agents).

To answer these questions, we conducted an initial study of upper-level undergraduate CS students' basic TM performance. This study consisted of five semi-structured interviews with undergraduate CS students currently attending Tufts University. The primary interview component was a TM exercise where participants were asked to identify threats in a mock system. We also asked students for their perception of the process – how easy or difficult the entire process felt, and whether they found any specific part difficult or confusing.

For the TM exercise, students used the STRIDE method. STRIDE is a TM approach originally developed at Microsoft [2], which focuses the developer's attention on potential categories of threats through the use of a helpful mnemonic. STRIDE is an acronym for **S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service, and **E**levation of privilege. We choose to use the STRIDE framework for this study because it is very

common in industry [10, 11, 13, 16], it forms the basis of other methodologies [16], it is more flexible and robust than other specialized frameworks [10, 17], and it is relatively straightforward and simple.

Our initial results highlighted the value of hands-on security experience and students' challenges to understanding difficult STRIDE concepts, such as repudiation (the ability for an attacker to deny involvement in a malicious operation). Drawing on our results, we suggest trends we plan to investigate as we continue this research.

2 Methods

To understand the challenges faced by students when threat modeling, we utilized semi-structured interviews. The interviews gauged participants' previous knowledge and exposure to TM, and evaluated their ability to create a TM for a simple system using a popular TM framework, i.e., STRIDE [11]. In this section, we discuss participant recruitment, outline our interview's structure, and describe our analysis method and study limitations. All study procedures were approved by the Tufts University's Institutional Review Board.

Study Recruitment We advertised the study with posters around the Tufts' CS department and at other places frequented by CS students around campus. We also posted advertisements for our study on discussion boards for CS courses. Interested students were asked to complete an initial consent form, which covered collection of initial screening information, a short screening survey to indicate their current degree year and any security courses completed at our institution or elsewhere. We restricted participation to students who had completed at least 2.5 years of the undergraduate curriculum. We chose this threshold as these are students who will soon enter the workforce and to ensure students had completed the necessary system development courses to have sufficient technical knowledge to understand our mock system.

2.1 Interview

Pre-exercise questions and training Qualifying student volunteers were invited to participate in a semi-structured interview. At the start of the interview, we asked participants to complete an additional consent form specific to the interview, which authorized to audio recording of the interview.

To ensure all participants had some familiarity with TM prior to the start of the exercise, we used a short lesson to teach all participants generally how to use

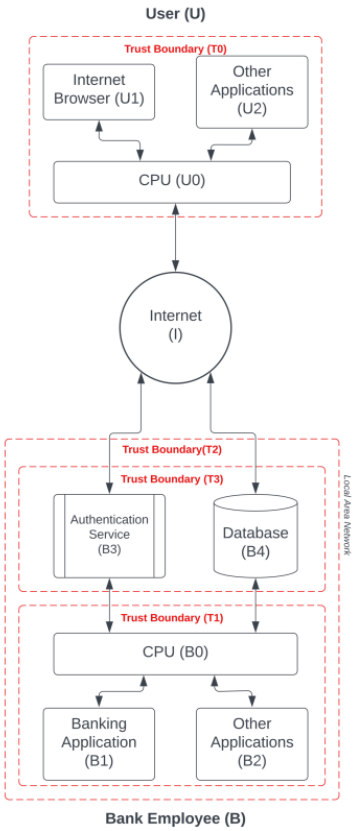

Figure 1: Exercise Mock System

the STRIDE TM process. Our STRIDE lessons first described the graphical elements of a STRIDE system model using an example data flow diagram from *TM: Designing for Security* [11]. Then, we described each threat in the STRIDE mnemonic, giving nontechnical examples for each, similar to the approach in Stevens et al. [12].

Threat modeling exercise The central interview component was a TM exercise. Participants were asked to identify threats in a mock system and suggest possible mitigations to these threats while being observed by a researcher. The model presented to participants is given in Figure 1.

The mock system is a simple banking service, with bank (B) and client (U) components that communicate over the Internet. In the bank component, a database service holds all client financial information (B4). Access to this service is mediated by an authentication service (B3) that users must first connect to and provide their credentials. Bank employees can access the database through a banking application (B1) installed on machine (B0)

located on the same local area network as the database service. The local machine also hosts other applications (B2) not related to the banking database. Clients access the banking database over the Internet through their Internet browser (U1) on their personal device (U0), which is expected to also host other unrelated applications (U2).

We chose a relatively simple mock system to ensure participants had sufficient development knowledge to easily understand the system's functionality. While this system is relatively simple, it exposes multiple attack vectors (e.g., malicious user, insider threat, man-in-the-middle) and could potentially have several different types of vulnerabilities (e.g., incorrect or insufficient authentication, faulty input parsing, lack of protection of data in transit). It is likely a more complex system would cause more issues for participants, so we expect responses in this exercise will offer an upper bound on threat modeling performance.

After showing participants the mock system and providing a high-level functionality description, we instructed participants to identify threats in the system and suggest mitigations. We also asked participants to think aloud. Because the system diagram is not an exhaustive definition of the system's functionality, we informed participants that they could make assumptions about the system (e.g., they could assume Internet traffic uses SSL/TLS encryption), but all assumptions would need to be explicitly stated.

To avoid biasing participant responses, the interviewer mostly quietly observed the participant during the exercise. However, if the participant only described a threat very vaguely (e.g., the database could be tampered with), then the interviewer asked the participant to provide more details if possible. In these situations, the interviewer did not provide any information beyond that already stated by the participant and simply asked for more details. Also, in cases where it appeared participants were implicitly making assumptions about the system, the interviewers probed the participant to make these assumptions explicit.

Post-exercise questions and debrief After completing the exercise, we asked the participant several questions about their experience. Specifically, we sought to determine whether participants perceived TM as useful and doable with a reasonable amount of effort and time (RQ1). We also asked participants whether they found specific parts of the STRIDE framework to be particularly challenging or easy (RQ2).

Because there was potential for psychological harm if a participant believed they performed poorly on the TM exercise, we conducted a debrief at the end of each interview. During this debrief, we discussed the set of potential threats in the mock system and reminded participants that their performance does not impact their compensation, and that they can withdraw from the study at any time for any reason.

2.2 Analysis

We used quantitative and qualitative measures of performance and participant perceptions of threat modeling.

Quantitative analysis To numerically compare participant performance, we created a point system to evaluate each TM's thoroughness. Each reasonable threat identified increases the model's score by one point. If the participant listed a threat that could not exist given the diagram, that threat was deemed unreasonable and did not gain a point–though we tracked invalid threats for later analysis. We also did not award points when the response was too vague to effectively determine a particular problem. Participants were also given an additional point for each STRIDE category they identified a threat in, for a possible 6-point bonus. Participants were not informed of this scoring system during the interview and were not provided the assigned score for their threat model to avoid motivating artificial behavior to game the scoring mechanism. Our scoring system is simply used as a method for quantifying participant TM output.

Qualitative analysis We performed a thematic analysis of the threats identified, as well as the strategies students developed. For example, did participants begin by considering points of human input (an attacker-focused approach) or start first by identifying the most important assets in the system (a "center of gravity" approach [12])?

3 Preliminary Results

This section describes themes identified in our initial evaluation from the TM exercise and the post-exercise questions. Note, these trends are drawn from a small sample and may not generalize. We report these trends to suggest directions for analysis in a larger-scale study.

3.1 Participants

There were five participants in our initial study pilot, covering the spectrum of security experience. Two participants had no prior security experience. Two participants had previously completed a computer security course:

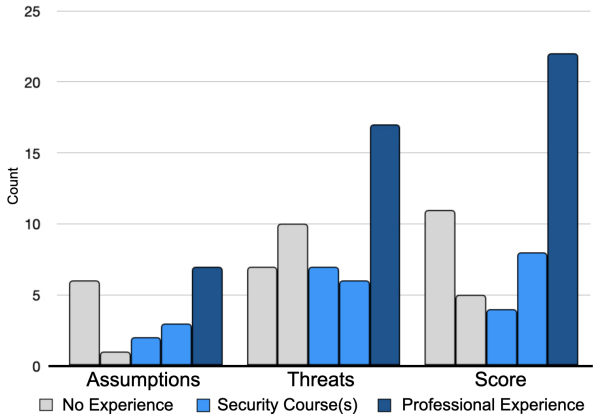

Figure 2: Participant performance on three metrics–Number of assumptions, number of threats, and final score–divided by prior security experience

one completed an introductory course which broadly covered a range of topics in security with hands-on technical assignments, and the other completed a non-technical course that covered cyber threats from a policy and legal perspective. The remaining participant had previously taken the same introductory security course and had professional experience in a security-related role as an intern. All our participants were in the second semester of their third year of their undergraduate degree.

3.2 Threat Modeling Performance (RQ1)

We first look at how well participants performed during the TM exercise by considering the number and quality of threats identified and assumptions made and contrasting results between participants with different levels of security experience. Figure 2 shows the number of assumptions and threats identified by each participant, as well as their final assessed score. Each bar represents a unique participant and bars are colored to indicate the participants' level of security experience.

Most participants identified few detailed threats On average, participants identified about nine threats during the TM exercise. However, in many cases, these threats were marked invalid. This was particularly true for P2 and P3 (see Figure 2), who identified more threats than the other student with similar experience (P1 and P4, respectively), but scored lower. These threats were often marked invalid because the participant could not determine any details of the threat, not because of a misconception about security. For example, one participant explained that a denial of service threat could happen at the gateway between the Internet and the authentication service. However, when probed for how this could happen, they could not suggest a possible method to be defended against. This indicates that students may be able to identify areas for potential security issues, but would need additional expert or automated support to determine more specific threats.

Practical experience improves threat modeling P5, the only participant with practical security experience, performed dramatically better than all other participants. They identified 7 more threats and received 11 more points than the every other participant. They were also the only participant that identified threats in every STRIDE category. This difference may be unique to P5, but further investigation is necessary to determine if this dramatic gap exists across the broader population and whether just a small amount (one summer) of hands-on experience can significantly improve TM performance.

Security courses did not improve performance Participants who took CS security courses and did not have professional experience (P3 and P4), on average, identified fewer threats and scored worse than participants with no experience (P1 and P2). Further, we observed that both groups focused on less-technical business logic vulnerabilities and did not consider vulnerabilities in specific technologies, such as buffer overflows and SQL injection. This result was particularly interesting for students who took security courses as these topics were covered in those courses. This result aligns with prior work that showed students did not consider potential technical threats during code implementation that they had been exposed to previously in lectures [15]. This result should be investigated to determine whether it holds with a larger number of students, and also with students exposed to these concepts in different courses.

Attacker-focus improved scores Though all non-internship students focused on nontechnical threats, some students clearly performed better than others. The higher-performing students appeared to use a more attack-focused approach than those with lower scores. "Attacker focus" in this context means that students considered threats from a malicious adversary more than accidents and unintentional errors. The difference in scores among students with near-identical technical knowledge supports Hamman et al.'s separation of the hacker mindset into technical, creative, and practical intelligence [8]. We consider these implications further when discussing future work.

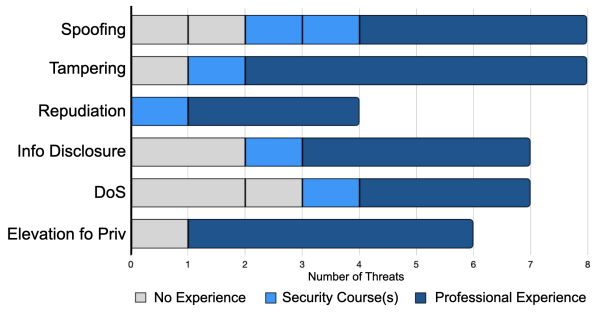

Figure 3: Number of threats identified in each STRIDE category, divided by participant prior security experience

3.3 STRIDE Category Challenges (RQ2)

Next, we investigated whether any STRIDE category was particularly challenging for participants. Figure 3 shows the number of threat identified for each participant, with each row indicating a different STRIDE category.

Spoofing and DoS were reported by most participants Spoofing threats were reported by all five participants and Denial of Service (DoS) threats were reported by four of five. Participants reported being the most comfortable identifying these types of threats and found these terms to be the most intuitive.

Repudiation was the most challenging Repudiation was the least represented threat category. Only three participants reported repudiation threats, and one was deemed invalid as it was based on a security misconception. Four of five participants also reported that repudiation was the most difficult category of STRIDE. Participants reported that the term itself was confusing, i.e., they did not understand the definition of "repudiation," and it was also difficult to know where repudiation threats might exist.

4 Discussion and Conclusion

After asking five undergraduate CS students to identify threats in a mock system, we observed that hands-on professional experience can have dramatic benefits, that introductory security courses did not improve threat modeling performance, and that students are particularly confused by repudiation threats. Therefore, in our future work, we plan to further investigate these trends through two specific changes: 1) recruitment of early-career professionals along with students to measure the impact of professional experience, and 2) recruit students from multiple institutions to compare several security introduction courses. Making these three changes will allow us to determine whether our results generalize and more clearly identify the causes for variation in TM performance.

References

[1] NICE Framework and the NSA Knowledge Units (KUs).

[2] The threats to our products. Technical report, Microsoft Interface, 1999.

[3] Hala Assal and Sonia Chiasson. Security in the software development lifecycle. In *Symposium on Usable Privacy and Security*, pages 281–296, Baltimore, MD, 2018. USENIX Association.

[4] Borka Jerman Blažič. The cybersecurity labour shortage in Europe: Moving to a new concept for education and training. *Technology in Society*, 67:101769, 2021.

[5] Deb Bodeau, David B Fox, and Catherine D McCollum. Cyber Threat Modeling: Survey, Assessment, and Representative Framework. Technical report, Homeland Security Systems Engineering and Development Institute (HSSEDI), April 2018.

[6] Sam Chung, Yun-Tse Wu, Teresa Escrig, and Barbara Endicott-Popovsky. Toward Software Assurance: Infusing a Threat Modeling Methodology into Beginning Level Programming Courses. *CCSC: Southwestern Conference*, page 11, 2014.

[7] ABET Computing Accreditation Commission. *Criteria for Accrediting Computing Programs*. ABET, Baltimore, MD, USA, December 2020.

[8] Seth T. Hamman and Kenneth M. Hopkinson. Teaching Adversarial Thinking for Cybersecurity. *Journal of The Colloquium for Information System Security Education*, 4(1), October 2016.

[9] Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. Association for Computing Machinery, New York, NY, USA, January 2013.

[10] Rafiullah Khan, Kieran McLaughlin, David Laverty, and Sakir Sezer. STRIDE-based threat modeling for cyber-physical systems. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6, September 2017.

[11] Adam Shostack. *Threat Modeling: Designing for Security*. John Wiley & Sons, Indianapolis, IN, 2014.

[12] Rock Stevens, Daniel Votipka, Elissa M Redmiles, and Michelle L Mazurek. The Battle for New York: A Case Study of Applied Digital Threat Modeling at the Enterprise Level. page 18, Baltimore, MD, August 2018. USENIX Association.

[13] Frank Swiderski and Window Snyder. *Threat modeling*. Microsoft Press, 2004.

[14] Alex Vieane, Gregory Funke, Robert Gutzwiller, Vincent Mancuso, Ben Sawyer, and Christopher Wickens. Addressing Human Factors Gaps in Cyber Defense. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 60(1):770–773, September 2016.

[15] Daniel Votipka, Kelsey R Fulton, James Parker, Matthew Hou, Michelle L Mazurek, and Michael Hicks. Understanding security mistakes developers make: Qualitative analysis from Build It, Break It, Fix It. page 19.

[16] Kim Wuyts, Riccardo Scandariato, and Wouter Joosen. Empirical evaluation of a privacy-focused threat modeling methodology. *Journal of Systems and Software*, 96:122–138, October 2014.

[17] Wenjun Xiong, Emeline Legrand, Oscar Åberg, and Robert Lagerström. Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix. *Software and Systems Modeling*, 21(1):157–177, February 2022.