


# Who Are “We”?

## Power Centers in Threat Modeling

Adam Shostack   
shostack@uw.edu

University of Washington and Shostack + Associates

**Abstract.** I examine threat modeling techniques and questions of power dynamics in the systems in which they’re used. I compare techniques that can be used by system creators to those used by those who are not involved in creating the system. That second set of analysts might be scientists doing research, consumers comparing products, or those trying to analyze a new system being deployed by a government. Their access to information, skills and choices are different. I examine the impact of those difference on threat modeling methods.

### 1 Introduction

Threat modeling is a collection of techniques for proactive security analysis of systems. The consensus industry methods are based on Shostack’s Four Question Framework (“What are we working on, what can go wrong, what are we going to do about it, did we do a good job?” [?]) This paper builds on work by feminist scholars and activists to look at the influence of the intended users of industry methods. In other words, the use of ‘we’ in the framework was a choice that ignored power dynamics. I suggest a threat modeling approach designed to helping people analyze a system they were not involved in creating. (Terms like ‘customer’ or ‘user’ are not broad enough. Systems are often imposed, such as resume scanners, traffic cameras or border security.) For clarity, this paper avoids the convention of single author referring to themselves as ‘we.’ This draws heavily on themes of power dynamics from Ross Anderson’s work.

There are two main senses in which the term *threat model* is used. The earlier is ‘What’s your threat model?’ and ‘random oracle’, or ‘a network attacker,’ could be complete answers. The term was adopted into ‘a model of threats,’ in the sense of an abstraction of possible future harms (spoofing, tampering, etc) as applied to a system under development [?], and was deployed in informal practices such as whiteboard discussions about system security. These were adopted by [?], [?], [?] and others into increasingly structured methodologies. The first sense is answered by a few words, the second sense is often answered with a set of diagrams, lists of threats and mitigations and tables interlinking them.

I’ll refer to these approaches as ‘analyst’ threat modeling and ‘creator’ threat modeling, respectively. The first helps us understand the relevance of an attack or analysis, the second helps anticipate and thus prevent them. Interestingly,

the question ‘what are we working on’ can be applied in either, while the techniques for answering it change. Analysts start by identifying components, data flows, and scope from a purely observational perspective. Creators have access to documentation, source code, and decision makers.<sup>1</sup>

## 2 Critiques

Sets of scholars and practitioners sought to bring creator threat modeling techniques to the analyst perspective. These included those writing under an umbrella of feminist cybersecurity and others focused on the needs of activists. In doing so, they exposed biases and limits of the techniques. Others lacked either access to the developers, or technical knowledge of software creation or operations.

### 2.1 Survey of Critiques

Freed et al examine ‘interface-bound attackers,’ who cause harm while using products as intended[?].<sup>2</sup> Spammers, bullies, trolls, phishers and creators of deepfakes operate within system rules, yet Stamos notes these attacks caused most harm while he led security at Facebook [?].

Slupska et al attempted to threat model a smart lock, and in particular analyze it for issues of intimate partner violence (IPV) [?]. The project exposed first, that creator perspective is limited, and second, that the techniques of creator threat modeling don’t help an end user understand the problem. I’ll use this as an example, because it illustrates many challenges with creator threat modeling.

Creator techniques assume a trustworthy administrator. IPV perpetrators often take control of a user session, and monitor systems for changes. If Alice manages the lock, Bob (an abuser) may have her password or demand administrative access. Bob may be notified if Alice limits his access. If Bob is the admin and Alice uses physical access to the lock to reset it, Bob may be notified or asked to approve the change. So how should the lock company design an access control matrix? They might focus on an admin who can create accounts or change permissions, and users who lock or unlock the door. But the use case of two users with the administrative password is unusual for computer security, and our normal response of ‘set an acceptable policy’ may lead to a literal slap in the face. The complexity and effort of enumerating attacks may inhibit creators from investigating or recording them. If they are analyzed, the complexity of addressing them may be declared to be an ‘edge case’ or otherwise de-prioritized.

Additionally, creator threat modeling methods like STRIDE or kill chains don’t help Alice (as an analyst) discover or reason about these problems.

Space limits our ability to discuss a growing body of work including that by EFF [?], Levy [?], Loadenthal [?], Kazansky [?], and Sterling[?].

<sup>1</sup> A distinction that I failed to note in a recent corporate whitepaper [?].

<sup>2</sup> The author’s unpublished exploration of how to threat model such systems is at <https://github.com/adamshostack/conflictmodeling/>

## 2.2 Analysis

We can consider possible threat modelers in a space defined by technical knowledge and system knowledgem as shown in Figure 1.

**Social Mileu** Microsoft recognized that design choices were being made unknowingly by developers and wanted them to be able to perform analysis. To scale, we aimed at simpler processes. (There were several downsides to this, including perhaps insufficient

recognition of the quality tradeoffs between experts, and a focus on reviews and documents over skills and engagement.) These circumstances informed the creation of threat modeling methodologies appropriate for use by technical experts to analyze systems with which they were highly familiar, or where they had access to the developers or code.<sup>3</sup> Early versions of the Four Question Framework used ‘you,’ as in “What are you working on?”, and that was intentionally changed to ‘we’ to be more collaborative.<sup>4</sup>

This approach can be (and was!) contrasted with Anderson’s educational approach. Colleagues argued “We can’t require people to get a PhD in security,” or “read a 500 page book.”<sup>5</sup> Anderson expected people to think critically and well, Microsoft needed to provide a process or methodological set of steps they could follow. The focus on process was seen as a requirement for scaling, supported auditability, and was a response to a frequently expressed “just tell me what you want me to do.”

The approach can also be contrasted to the sorts of threat modeling done by spies, attackers, bug bounty participants, or even academics who start with limited knowledge of a system, but a great deal of technical knowledge, possibly including security knowledge. They may be willing to dedicate more time, or they may see a single bug as a sufficient result. (The ‘single bug’ goal can be contrasted with the need for creators to build a secure system.) Their technique choices and investment of energy will be shaped by those circumstances.

Microsoft’s approach was an implicit decision of which participants matter. The company put technical participants (and technical threats) first. The concerns of the people impacted was not a ‘use case’ that we discussed often. This move made perfect sense *to the company*, who refered to their products as ‘secure by design.’ This can be contrasted with the approaches required by the

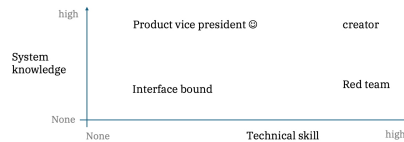


Fig. 1: A threat modeling space

<sup>3</sup> It is tempting to say ‘easy access,’ but that ignores the sometimes contentious inter-team relationships.

<sup>4</sup> Other important work included that of Kohnfelder and Garg [?] and Swiderski and Snyder[?]. A slightly fuller history is available at [?].

<sup>5</sup> Noting that the first edition of *Writing Secure Code* was 501 pages including introduction, and had a quote from Bill Gates, ‘Required reading at Microsoft’ on the front cover.

Food and Drug Administration, whose design-time requirements for medical device makers include a ‘multi-patient harm view.’ Here, the FDA is acting as a counter-balancing power center relative to device makers. Deeper consideration of power relationships could improve the benefits brought by threat modeling.

**Technical Knowledge** Microsoft’s software engineering roles (even program management) require deep technical knowledge. Their threat modeling methodologies thus assumed technically skilled participants.

**Knowledge of system** Threat modeling methodologies were developed for internal use by Microsoft product teams who were asked to engage with product security experts. Cost and effort of knowledge transfer less important because these experts would often embed for periods between weeks and years. Even so, those experts might not be briefed on features for many reasons. Those could include people doing feature work didn’t see security implications, or a desire to avoid security so an insecure feature could ship. Reviews were also conducted by highly skilled experts, and likely closer to what’s called product red teaming.

### 3 Threat Modeling ‘for the rest of us’

This section presents a simpler approach to threat modeling, designed for use by those with lower technical skill and less knowledge of a system. (The term is used for clarity, not as a judgement.) I’ve selected pronouns to be personal, even though foundational work to be done by advocates. The Framework is:

1. What have they delivered?
2. How will it hurt me?
3. Can I protect myself?
4. Should I even use it?

These questions are designed to be answerable, even if finding answers may require specialized skills. They aligned with the Four Question Framework to help experts remember them. Next, I explain each question and structured approaches.

#### 3.1 What have they delivered?

Understanding what a software package is has become more complex with the prevalence of web apps and associated back ends, compared to earlier models of software on floppy disks.

We might be able to use a simple model of ‘local’ and ‘cloud.’ People believe that data on their device is private and more secure, a belief created or reinforced by both intuition, and marketing like “Your fingerprint never leaves your device.” Questions that can be asked by those with low technical skill might include:<sup>6</sup>

<sup>6</sup> These questions should obviously be tested for usability.

- Does it work without internet access?
- Can I use it without creating an account or providing a working email address?
- What does the privacy policy tell me?

Analyzing privacy policies requires determination, and maybe skill, but can expose accessible lessons, like "We share data with our 1400 partners."

Those with more technical skill browser plugins like Noscript or tools like Wireshark, and going deeper, analyst methods start to resemble those used by security researchers, rising to enumerating libraries, using a debugger or even logic probes or electron microscopy to analyze a chip or device. Firmware and mobile apps can be downloaded and prised open, and freely available code even provides the permissions the library uses.[?]

### 3.2 How will it hurt me?

Creator-oriented threat modeling may draw on frameworks like STRIDE to structure an analysis, but that requires technical skills.[?]. A simpler set of threats, such as what does it learn and where does it send it may be helpful, but even local processing may be against the interests of a user. For example, does it show ads? Will it change function on update?

### 3.3 Can I protect myself, and Should I even use it?

The history of general-purpose computing is a history of modifying software to serve local needs, including security. Adblockers [?]. The trend towards restricted platforms (e.g., phones, IoT) limit user control while increasing protection against malware[?]. These restrictions complicate decisions about whether to use such systems.

More broadly, defending against trusted but untrustworthy software is challenging, even for experts. For less skilled users, it can become a Kafka-esque experience, with valid advice hard to separate from superstition.

## 4 Conclusion

The author regrets implying that threat modeling techniques are universal. Both people’s depth of technical skills and their involvement in the creation of a system influence how they may threat model.

## Acknowledgements

Julia Slupska and Leonie Tanczer helped me understand the problem they were grappling with. Josiah Dykstra, Jay Healey, Loren Kohnfelder and Kim Wuyts provided helpful feedback on drafts. Over decades, Ross Anderson’s writings have profoundly influenced my own. I mourn his loss and hope to contribute this small bit to the celebration of his legacy.

## References

1. Electronic Frontier Foundation, *Risk Assessment (Threat Modeling)*, June 24, 2019, <https://www.eff.org/files/2020/01/06/threatmodeling-onepager.pdf>
2. Farmer, W. R. and Venema, A., *Improving the Security of Your Site by Breaking Into It*, message to comp.security.unix, December 1993, <https://cyberwar.nl/d/1993-FarmerVenema-comp.security.unix-Improving-the-Security-of-Your-Site-by-Breaking-Into-It.pdf>
3. Freed, D., J. Palmer, D. Minchala, K. Levy, T. Ristenpart, and N. Dell, *A Stalker's Paradise: How Intimate Partner Abusers Exploit Technology*, in Proceedings of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing, 2018, pp. 163-177.
4. Howard, M., and D. LeBlanc, *Writing Secure Code*, 1st ed. Redmond, WA: Microsoft Press, 1999.
5. Kazansky, B. *It depends on your threat model: the anticipatory dimensions of resistance to data-driven surveillance*, Big Data and Society, January 29, 2021, DOI: 10.1177/2053951720985557
6. Kohnfelder, L., and P. Garg, *The threats to our products*, Microsoft Interface, Microsoft Corporation, vol. 33, Apr. 1999.
7. Levy, K. and Schneier, B., *Privacy Threats in Intimate Relationships*, Journal of Cybersecurity, Vol 6, Issue 1, 2020, available at: <https://academic.oup.com/cybersecurity/article/6/1/tyaa006/5849222>.
8. Loadenthal, M., *Risks, Dangers, and Threat Models: Evaluating Security Analysis for Conflict Practitioners*, August 2021, DOI:10.13140/RG.2.2.35515.95526
9. Schneier, B., *Attack Trees: Modeling Security Threats*, Dr. Dobb's Journal, December 1999.
10. Slupska, J. and L. M. Tanczer, *Threat Modeling Intimate Partner Violence: Tech Abuse as a Cybersecurity Challenge in the Internet of Things*, in The Emerald International Handbook of Technology Facilitated Violence and Abuse, 2021, pp. 663-688.
11. Shostack, A. *Understanding the Four-Question Framework for Threat Modeling*, corporate whitepaper, November, 2024, <https://shostack.org/whitepapers>
12. Shostack, A., *Threats: What Every Engineer Should Learn From Star Wars* (Wiley, 2023).
13. Shostack, A., *Threat Modeling: Designing for Security* (Wiley, 2014).
14. Sterling, L., *Practitioners of Civil Resistance: Assess Your Cybersecurity through Threat Modeling* March 22, 2018, International Center for Non-violent Conflict, [https://www.nonviolent-conflict.org/blog\\_post/practitioners-civil-resistance-assess-cybersecurity-threat-modeling/](https://www.nonviolent-conflict.org/blog_post/practitioners-civil-resistance-assess-cybersecurity-threat-modeling/)
15. Stamos, A., *Stepping Up Our Game: Re-focusing the Security Community on Defense and Making Security Work for Everyone*, YouTube, 2017. Available: <https://www.youtube.com/watch?v=YJOMTAREFtY>
16. Swiderski, F., and W. Snyder, *Threat Modeling*, Microsoft Press, 1st ed., 2004.
17. Xin, J. *app-third-party-library*, GitHub, 2024. Available: <https://github.com/xinjin95/app-third-party-library> [Accessed: 22 Nov 2024].
18. Zeunert, M. *Chrome Extension Statistics: Data From 2024*, blog post August 29, 2024, <https://www.debugbear.com/blog/chrome-extension-statistics>
19. Zittrain, J. L. *The Future of the Internet: And How to Stop It*. New Haven: Yale University Press, 2008.