Let's Retire the Term 'Threat Modeling'

By Donald A. McKeown, Member, ISSA New England Chapter

I am an enthusiastic proponent of threat modeling. There is great value in systematically analyzing product security from a threat-based perspective during the software design stage and implementing fixes, or mitigating, eliminating, or accepting risk. There is also value in doing this analysis in post-design stages. However, I believe the term threat modeling is problematic and is one factor that inhibits teams from performing it.

One definition of threat modeling is "a family of activities for improving security by identifying objectives and vulnerabilities, and then defining countermeasures to prevent, or mitigate the effects of, threats to the system [1]." Another definition is similar but also includes understanding "the threat landscape, the most likely attacks, their methodology, motive, and target system [2]." Adam Shostak discusses the practical value of using a model to represent underlying detail so that you can find problems early and anticipate relevant threats [3].

Regardless of the definition you prefer, the idea is to approach design and analysis of the security of information systems from a threat-centric point of view. It helps generate requirements in addition to those from the business, organizational standards, compliance, legal/regulatory, and the industry best practices. While such requirements are important, and often mandatory, it is difficult for these to account for all threats in the real world. The threat environment is changing rapidly, all system designs vary, and in the real world it is extremely difficult to maintain lists of requirements that can adequately protect a system in today's rapidly shifting threat landscape. Threat modeling is a critically important security practice that helps close the gap between predefined requirements and everchanging threats.

While threat modeling can be applied to systems later in the Software Development Life Cycle (SDLC), the greatest value lies in performing threat modeling during the design phase of a system. This is because the earlier security is built into the design, the more cost-effective it is to do so. As code defects are detected later in the software development lifecycle, the cost to remediate them rises exponentially [4].

Therefore, it is most cost-effective to motivate architects and engineers to consider application security in a structured, threat-centric manner while products and features are being designed. In my experience, most if not all teams take security seriously and build security in as well as they can. However, it is a challenge to balance security with the immense pressure to quickly produce products and features that will generate revenue.

Besides the pressure to create money-making features, the term threat modeling is another factor that inhibits teams from performing it. Threat modeling is not an easy term for non-security professionals to understand. For those outside of information security, it is nonintuitive. I'm pretty certain many engineers' initial response to the term threat modeling is something like "Model threats? What the heck are you talking about?" The product may be designed and perhaps some of it is coded in a development environment. But oh no, the team realizes or is reminded that it must do that threat modeling thing or whatever it is called! Because the process uses an obscure term and is often bolted on rather than integrated nicely into the SDLC, it can feel like a separate, required exercise that can seem more like a compliance checkbox exercise than a valuable opportunity to build better products for customers. Moreover, for many teams, it is built into their DNA to think about building features and products with reliability and availability. But security can sometimes feel like an afterthought. At times, teams will only take a strong focus on security if they are told to.

Perhaps we should do away with the term threat modeling. How about if we call it security requirements generation and bake it into the design phase? When a product or feature idea is conceived, a product team will deliver requirements to the engineering team. There are the functional requirements, many of which will directly generate revenue. There are security requirements that can come from multiple sources such as organizational standards, legal, regulatory, and customer contracts. Perhaps the definition of done for a design should include security requirements derived from a threat-based analysis. So why not call the output of the threat modeling process "security requirements" and consider it as part of the requirement gathering stage of the SDLC?

As mentioned, a threat modeling methodology has value when applied in post-design stages to existing systems. Once again, simply do not use the term threat modeling and bake it into the process of security reviews. There is tremendous value in doing a security review based on a data flow diagram with potential threats and treatment decisions documented. The review moves faster and demonstrates that the teams performed a systematic analysis.

If you take a broad view of threat modeling that includes analyzing the real-world threat environment (referenced above [2]) rather than only applying a threat framework such as STRIDE, then separate the analysis of the threat environment into a separate threat assessment. There are many approaches [5] and sources of data for doing this [6] [7]. It is helpful to understand the real-world prevalence of various types of threat actors, the types of attacks they typically launch, and their effectiveness because it is not practical to build systems that are secure against all known threats. Teams need to make tradeoffs, and understanding the dominant, most

continued on page 12

Open Forum (continued)

continued from page 7

potentially impactful threats in the real world can help with prioritization. Once the threat assessment is completed, it can be utilized for helping prioritize design choices, to help understand the most serious threats during a security review, or as an input into a risk assessment.

Ideally, dropping the term threat modeling and renaming it so that it aligns with existing, well-understood processes in the SDLC could be part of top-down driven organizational cultural shift to improve security [8]. If such a dramatic transformation is not practical, another approach could be to implement changes as part of a process improvement initiative. Integrating threat modeling practices into requirements gathering and security reviews will likely improve process efficiency. Another approach is to make the change as part of rolling out the annual security programs.

In short, let us retire the term threat modeling and replace it with the terms 'security requirements generation' and 'security reviews.' Let us tightly integrate threat modeling practices into design requirements gathering and security review processes. It will make the process more understandable, help speed adoption, and improve process efficiency. Most importantly, it will help improve the security of software in a very cost-effective manner because it will help ensure that security is baked into the software design.

About the Author

Don McKeown is currently an Information Security Manager at Wolters Kluwer Health. He earned an MBA with Distinction from Bentley University and holds the CISSP, CRISC, and GIAC Security Leadership (GSLC) certifications. He is available at don@donmckeown.net.

References

1. OWASP, "Threat Modeling," [Online]. Available: https://owasp.org/www-community/Threat_Modeling. [Accessed 28 June 2021].
2. EC-Council, "Cyber Threat Modeling," [Online]. Available: https://www.eccouncil.org/threat-modeling/. [Accessed 28 June 2021].
3. A. Shostak, Threat Modeling Designing for Security, Hoboken, NJ: Wiley, 2014.
4. M. Dawson, D. Burrell, E. Rahim and S. & Brewster, "Integrating Software Assurance into the Software Development Life Cycle (SDLC)," Journal of Information Systems Technology and Planning, vol. 3, pp. 49-53, 2010.
5. J. Wynn, J. Whitmore, G. Upton, L. Spriggs, D. McKinnon, R. McInnes, R. Graubart and L. Clausen, "Threat Assessment & Remediation Analysis (TARA)," Mitre, October 2011. [Online]. Available: https://www.mitre.org/sites/default/files/pdf/11_4982.pdf. [Accessed 29 June 2021].
6. D. A. McKeown, "Understanding threats to Protected Health Data (PHI) using the Verizon DBIR and HHS breach data," 12 September 2019. [Online]. Available: https://donmckeown.net/understanding_threats_to_protected_health_data.htm.
7. D. A. McKeown, "Analysis of PHI Breach Data Indicates Different Control Recommendations than 2020 Verizon DBIR," 30 May 2020. [Online]. Available: https://donmckeown.net/analysis_of_phi_breach_data_indicates_different_control_recommendations_than_2020_verizon_dbir.html.
8. D. McKeown, "Building a Risk-based Information Security Program," ISSA Journal, pp. 14-21, 2019.