THREAT MODELING CONNECT
GLOBAL MEETUP

Success and Metrics for Threat Modeling

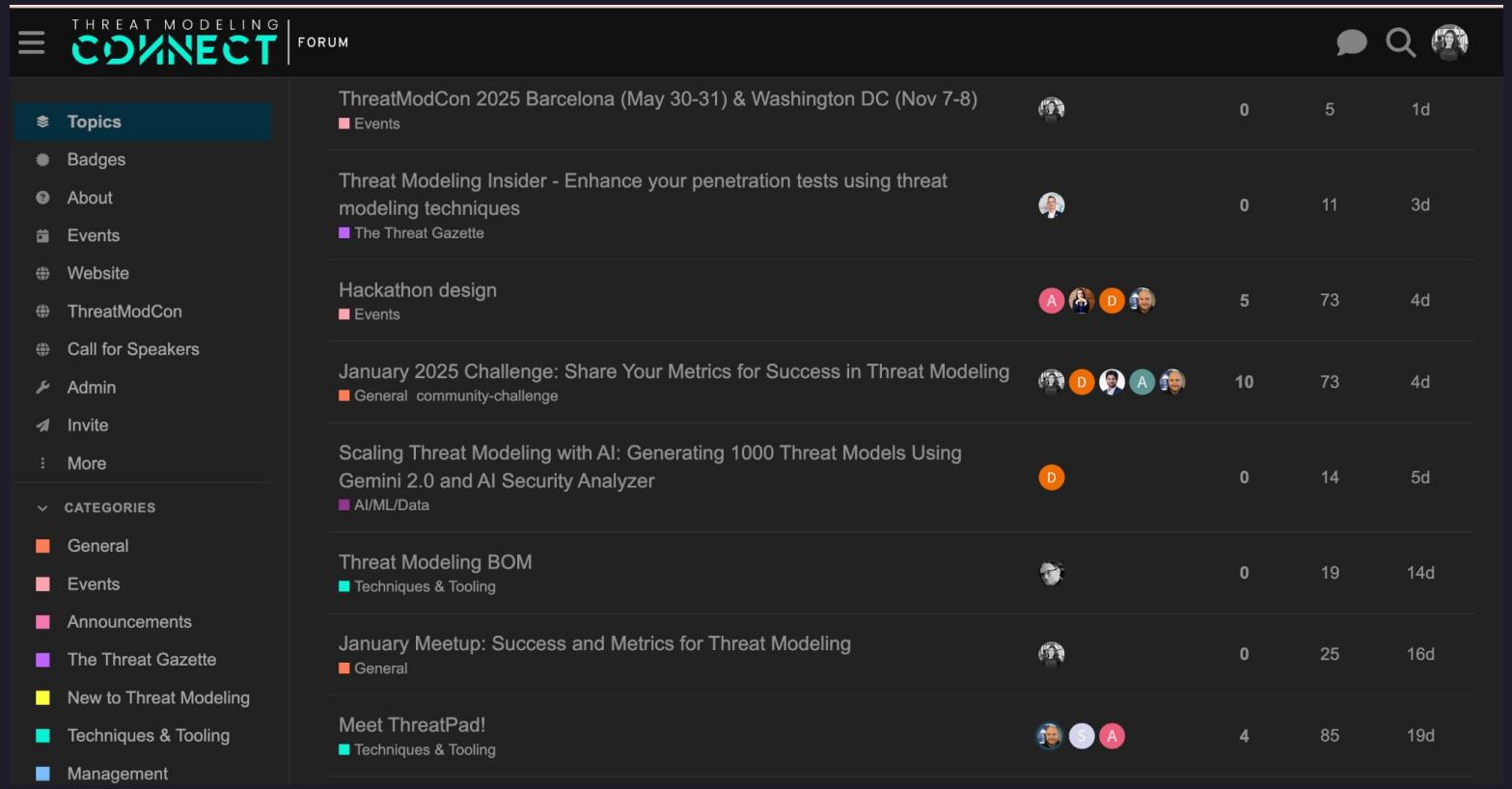
January 23, 2025
12pm - 1pm ET

THREAT MODELING CONNECT | POWERED BY IRIUSRISK

MEMBER NEWSLETTER

Get threat model examples, curated content, and event updates delivered to your inbox every month.

👉 threatmodelingconnect.com/join-the-community



Join the TMC forum to connect with peers, ask questions, share insights.

👉 threatmodelingconnect.discourse.group


THREAT MODELING
CONNECT
HACKATHON
APRIL 1-23, 2025
👉 https://www.threatmodelingconnect.com/hackathon
IN PARTNERSHIP WITH
IriusRisk

Call for Papers for ThreatModCon Barcelona opens now through Feb 28th.

👉 threatmodcom.com

THREAT MODELING CONNECT
GLOBAL MEETUP

Success and Metrics for Threat Modeling

January 23, 2025
12pm - 1pm ET

THREAT MODELING CONNECT | POWERED BY IRIUSRISK

About the Community Meetup

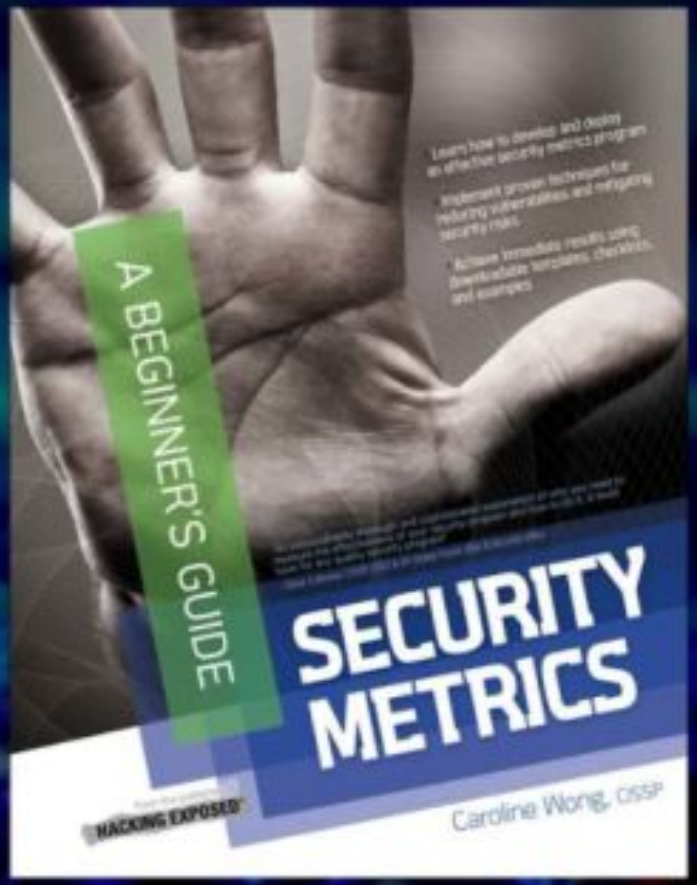
Our goal
Exchange real-world experience, share practical knowledge, validate ideas to improve our own practice.

Today's agenda (ET)
12:00 Welcome & intro
12:05 Presentation
12:25 Q&A
12:55 Closing and announcement

cwpro

CYBERSECURITY

CANON



Success and Metrics


The trouble with schools is – they always teach the wrong lessons

THREAT MODELING MANIFESTO

What is threat modeling?

Threat modeling is analyzing representations of a system to highlight concerns about security and privacy characteristics.

At the highest levels, when we threat model, we ask four key questions:

1. What are we working on?
2. What can go wrong?
3. What are we going to do about it?
4. Did we do a good enough job?

SDLC TOUCHPOINTS

SDLC TOUCHPOINTS: ARCHITECTURE ANALYSIS (AA)

Architecture analysis encompasses capturing software architecture in concise diagrams, applying lists of risks and threats, adopting a process for review (such as Microsoft Threat Modeling [STRIDE] or Architecture Risk Analysis [ARA]), building an assessment and remediation plan for the organization, and using a risk methodology to rank applications.

TOP 10

[AA1.1: 99] PERFORM SECURITY FEATURE REVIEW.

Security-aware reviewers identify application security features, review these features against application security requirements and runtime parameters, and determine if each feature can adequately perform its intended function—usually collectively referred to as threat modeling. The goal is to quickly identify missing security features and requirements, or bad deployment configuration (authentication, access control, use of cryptography, etc.), and address them. For example, threat modeling would identify both a system that was subject to escalation of privilege attacks because of broken access control as well as a mobile application that incorrectly puts PII in local storage. Use of the firm's secure-by-design components often streamlines this process (see [SFD2.1]). Many modern applications are no longer simply "3-tier" but instead involve components architected to interact across a variety of tiers—browser/endpoint, embedded, web, microservices, orchestration engines, deployment pipelines, third-party SaaS, etc. Some of these environments might provide robust security feature sets, whereas others might have key capability gaps that require careful analysis, so organizations should consider the applicability and correct use of security features across all tiers that constitute the architecture and operational environment.

[AA1.2: 56] PERFORM DESIGN REVIEW FOR HIGH-RISK APPLICATIONS.

Perform a design review to determine whether the security features and deployment configuration are resistant to attack in an attempt to break the design. The goal is to extend the more formulaic approach of a security feature review (see [AA1.1]) to model application behavior in the context of real-world attackers and attacks. Reviewers must have some experience beyond simple threat modeling to include performing detailed design reviews and breaking the design under consideration. Rather than security feature guidance, a design review should produce a set of flaws and a plan to mitigate them. An organization can use consultants to do this work, but it should participate actively. A review focused only on whether a software project has performed the right process steps won't generate useful results about flaws. Note that a sufficiently robust design review process can't be executed at CI/CD speed, so organizations should focus on a few high-risk applications to start (see [AA1.4]).

[AA1.4: 55] USE A RISK METHODOLOGY TO RANK APPLICATIONS.

Use a defined risk methodology to collect information about each application in order to assign a risk classification and associated prioritization. It is important to use this information in prioritizing what applications or projects are in scope for testing, including security feature and design reviews. Information collection can be implemented via questionnaire or similar method, whether manual or automated. Information needed for classification might include, "Which programming languages is the application written in?" or "Who uses the application?" or "Is the application's deployment software-orchestrated?" Typically, a qualified member of the application team provides the information, but the process should be short enough to take only a few minutes. The SSG can then use the answers to categorize the application as, e.g., high, medium, or low risk. Because a risk questionnaire can be easy to game, it's important to put into place some spot-checking for validity and accuracy—an overreliance on self-reporting can render this activity useless.

[AA2.1: 37] PERFORM ARCHITECTURE ANALYSIS USING A DEFINED PROCESS.

Define and use a process for AA that extends the design review (see [AA1.2]) to also document business risk in addition to technical flaws. The goal is to identify application design flaws as well as the associated risk (e.g., impact of exploitation), such as through frequency or probability analysis, to more completely inform stakeholder risk management efforts. The AA process includes a standardized approach for thinking about attacks, vulnerabilities, and various security properties. The process is defined well enough that people outside the SSG can carry it out. It's important to document both the architecture under review and any security flaws uncovered, as well as risk information that people can understand and use. Microsoft Threat Modeling, Versprite PASTA, and Black Duck ARA are examples of such a process, although these will likely need to be tailored to a given environment. In some cases, performing AA and documenting business risk is done by different teams working together in a single process. Uncalibrated or ad hoc AA approaches don't count as a defined process.

[AA2.2: 38] STANDARDIZE ARCHITECTURAL DESCRIPTIONS.

Threat modeling, design review, or AA processes use an agreed upon format (e.g., diagramming language and icons, not simply a text description) to describe architecture, including a means for representing data flow. Standardizing architecture descriptions between those who generate the models and those who analyze and annotate them makes analysis more tractable and scalable. High-level network diagrams, data flow, and authorization flows are always useful, but the model should also go into detail about how the software itself is structured. A standard architecture description can be enhanced to provide an explicit picture of information assets that require protection, including useful metadata. Standardized icons that are consistently used in diagrams, templates, and dry-erase board squiggles are especially useful, too.

[AA2.4: 40] HAVE SSG LEAD DESIGN REVIEW EFFORTS.

The SSG takes a lead role in performing design review (see [AA1.2]) to uncover flaws. Breaking down an architecture is enough of an art that the SSG, or other reviewers outside the application team, must be proficient, and proficiency requires practice. This practice might then enable, e.g., champions to take the day-to-day lead while the SSG maintains leadership around knowledge and process. The SSG can't be successful on its own either—it will likely need help from architects or implementers to understand the design. With a clear design in hand, the SSG might be able to carry out a detailed review with a minimum of interaction with the project team. Approaches to design review evolve over time, so don't expect to set a process and use it forever. Outsourcing design review might be necessary, but it's also an opportunity to participate and learn.

[AA3.1: 20] HAVE ENGINEERING TEAMS LEAD AA PROCESS.

Engineering teams lead AA to uncover technical flaws and document business risk. This effort requires a well-understood and well-documented process (see [AA2.1]). But even with a good process, consistency is difficult to attain because breaking architecture requires experience, so provide architects with SSG or outside expertise in an advisory capacity. Engineering teams performing AA might normally have responsibilities such as development, DevOps, cloud security, operations security, security architecture, or a variety of similar roles. The process is more useful if the AA team is different from the design team.

[AA3.2: 8] DRIVE ANALYSIS RESULTS INTO STANDARD DESIGN PATTERNS.

Failures identified during threat modeling, design review, or AA are fed back to security and engineering teams so that similar mistakes can be prevented in the future through improved design patterns, whether local to a team or formally approved for everyone (see [SFD3.1]). This typically requires a root-cause analysis process that determines the origin of security flaws, searches for what should have prevented the flaw, and makes the necessary improvements in documented security design patterns. Note that security design patterns can interact in surprising ways that break security, so apply analysis processes even when vetted design patterns are in standard use. For cloud services, providers have learned a lot about how their platforms and services fail to resist attack and have codified this experience into patterns for secure use. Organizations that heavily rely on these services might base their application-layer patterns on those building blocks provided by the cloud service provider (for example, AWS CloudFormation and Azure Blueprints) when making their own.

[AA3.3: 18] MAKE THE SSG AVAILABLE AS AN AA RESOURCE OR MENTOR.

To build organizational AA capability, the SSG advertises experts as resources or mentors for teams using the AA process (see [AA2.1]). This effort might enable, e.g., security champions, site reliability engineers, DevSecOps engineers, and others to take the lead while the SSG offers advice. As one example, mentors help tailor AA process inputs (such as design or attack patterns) to make them more actionable for specific technology stacks. This reusable guidance helps protect the team's time so they can focus on the problems that require creative solutions rather than enumerating known bad habits. While the SSG might answer AA questions during office hours (see [T2.12]), they will often assign a mentor to work with a team, perhaps comprising both security-aware engineers and risk analysts, for the duration of the analysis. In the case of high-risk software, the SSG should play a more active mentorship role in applying the AA process.

OWASP SAMM

Model | Design | Threat Assessment

The Threat Assessment (TA) practice focuses on identifying and understanding of project-level risks based on the functionality of the software being developed and characteristics of the runtime environment. From details about threats and likely attacks against each project, the organization as a whole operates more effectively through better decisions about prioritization of initiatives for security. Additionally, decisions for risk acceptance are more informed, therefore better aligned to the business.

By starting with simple threat models and building application risk profiles, an organization improves over time. Ultimately, a sophisticated organization would maintain this information in a way that is tightly coupled to the compensating factors and pass-through risks from external entities. This provides greater breadth of understanding for potential downstream impacts from security issues, tradeoffs, or flaws, while keeping a close watch on the organization's current performance against known threats.

Maturity level		Stream A Application Risk Profile	Stream B Threat Modeling
1	Best-effort identification of high-level threats to the organization and individual projects.	A basic assessment of the application risk is performed to understand likelihood and impact of an attack.	Perform best-effort, risk-based threat modeling using brainstorming and existing diagrams with simple threat checklists.
2	Standardization and enterprise-wide analysis of software-related threats within the organization.	Understand the risk for all applications in the organization by centralizing the risk profile inventory for stakeholders.	Standardize threat modeling training, processes, and tools to scale across the organization.
3	Proactive improvement of threat coverage throughout the organization.	Periodically review application risk profiles at regular intervals to ensure accuracy and reflect current state.	Continuously optimization and automation of your threat modeling methodology.

SAMM Threat Modeling: Maturity Level 1

Question

Do you identify and manage architectural design flaws with threat modeling?

Quality criteria

You perform threat modeling for high-risk applications
You use simple threat checklists, such as STRIDE
You persist the outcome of a threat model for later use

Answers

No
Yes, some of them
Yes, at least half of them
Yes, most or all of them

SAMM Threat Modeling: Maturity Level 2

Question

Do you use a standard methodology, aligned with your application risk levels?

Quality criteria

You train your architects, security champions, and other stakeholders on how to do practical threat modeling
Your threat modeling methodology includes at least diagramming, threat identification, design flaw mitigations, and how to validate your threat model artifacts
Changes in the application or business context trigger a review of the relevant threat models
You capture the threat modeling artifacts with tools used by your application teams

Answers

No
Yes, for some applications
Yes, for at least half of the applications
Yes, for most or all of the applications

SAMM Threat Modeling: Maturity Level 3

Question

Do you regularly review and update the threat modeling methodology for your applications?

Quality criteria

The threat model methodology considers historical feedback for improvement
You regularly (e.g., yearly) review the existing threat models to verify that no new threats are relevant for your applications
You automate parts of your threat modeling process with threat modeling tools

Answers

No
Yes, but review is ad-hoc
Yes, we review it at regular times
Yes, we review it at least annually

Risk Management Objectives

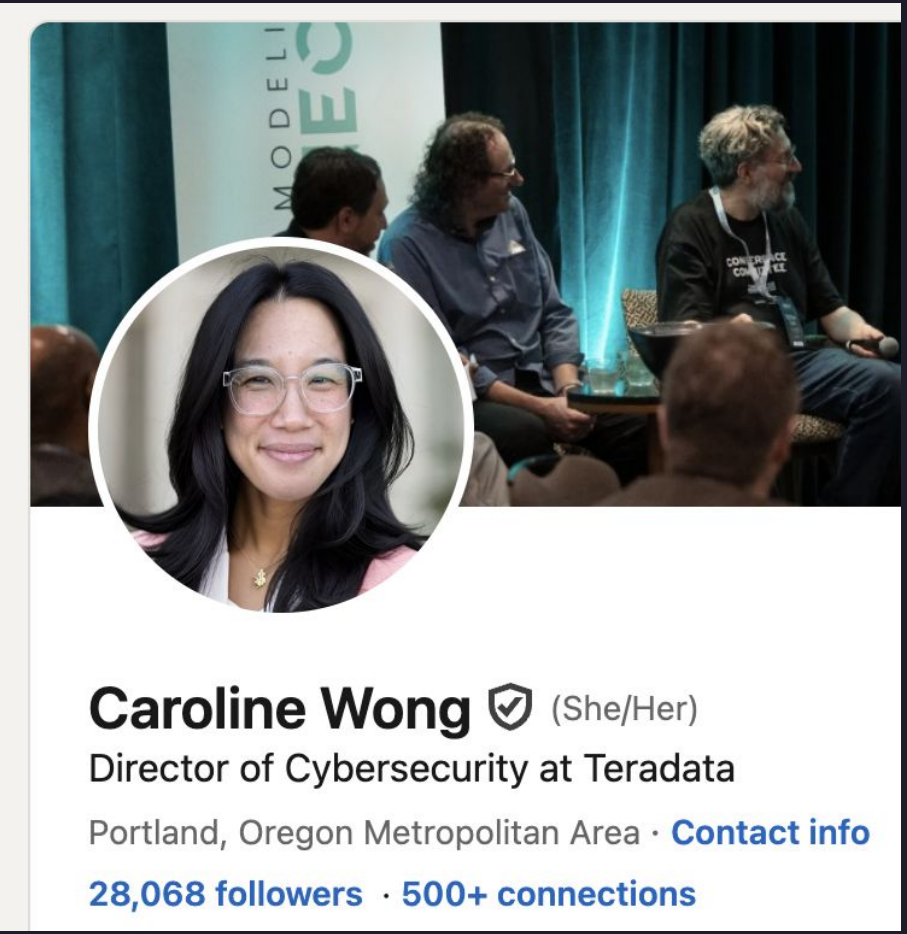
1. Use trust/security/privacy as a competitive differentiator
2. Comply with a regulatory requirement, contractual obligation, or industry standard
3. Achieve a defensible level of “due care”
4. Achieve a comparable level of trust/security/privacy as peers and/or competition

4. Prevent the same cybersecurity problems from happening over and over again

5. Reduce the probability that malicious attackers can stop critical systems from functioning

6. Require fixes for security bugs for which well known attacks exist

Watch my 40 min course on Security Metrics, at no cost!

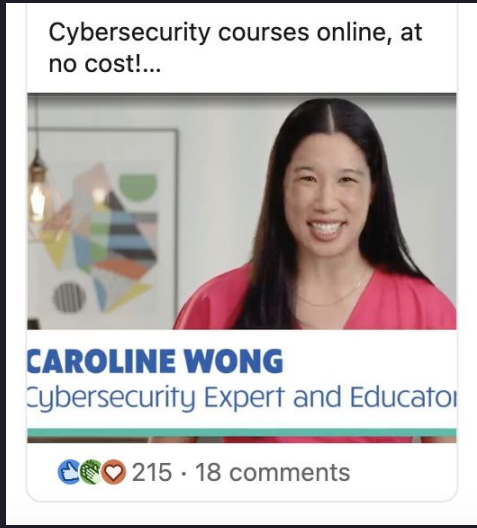


https://www.linkedin.com/in/carolinewmwong/

Featured → 4th post

In comments:

Security Metrics Insights, also known as "How to Ask for Security Budget ... And Get It!"



Zoom group photo

Q&A

Question:

Would a system complexity measure be helpful to normalize and compare all of the other metrics? And if so, are there recommended ways to (easily) measure complexity?

Question:

What's the benchmark for setting the success matrix for threat modeling programs?

Question:

How do you promote the qualitative value of threat models in a world that focuses on quantitative measures?

Question:

Depends on the proposed metrics, but I'm always interested to see how metrics handle the 'Cobra Effect'

Question:

How do you measure or assess the adoption of Threat Modeling across Development Lifecycle?

Question:

Assuming that a dev team does a fantastic job in securely designing a software system and no findings are identified in threat modeling, how can we measure that the threat model review was successful / effective?

Activity/progress metrics are not applicable in my case (e.g. how many hours we spent in threat modeling).

Question

How do you identify the word success for Threat Modeling?

Question:

Any suggestions for measuring the output of threat modeling against the actual implemented security controls?

Additional Questions

Take home messages from Caroline

To our Speaker of the Month - **Caroline Wong**

Slides & Recording

Available on the TMC forum next Monday
threatmodelingconnect.com (Click the "**Forum**" button)

Upcoming events

- **TMC Tokyo Meetup** (Jan 29th)
- **Global Meetup "It's Game Time: Elevation of Privilege & Byte Club"** (Mar 26th)
- **Threat Modeling Hackathon:** Registration opens on Feb 3
- **ThreatModCon 2025 Barcelona** (May 30-31) & **Washington, D.C** (Nov 7-8)

lu.ma/threatmodelingconnect

Call for Papers

ThreatModCon 2025 Barcelona: Open nows through Feb 28th
threatmodcon.com