WHITEPAPER

Evolving Threat Modeling for Agility and Business Value

What is threat modeling and why it is important?

Threat modeling represents a plethora of different practices to analyze a system from a security perspective. There are many implementations of this practice, therefore it may be useful to introduce the definition we have used as a reference while writing this guide.

Threat modeling is a practice to perform design and assess the potential threats to a system in scope, evaluate the eventual risks both from a technical perspective and from the point of view of the business, and to identify what can be done to make those risks acceptable.

The paper focuses on threat modeling from a general perspective, without delving into a specific methodology. The considerations and recommendations collected here should therefore be applicable to most approaches.

The evolution of threat modeling

In the early days, threat modeling was much simpler and based on systems where threat vectors against the system were well-known. In such cases creating threat modeling diagrams manually was easier – we had controlled access to the few systems that were available. But in today's DevSecOps world, things look quite different.

We have highly distributed systems where the emphasis is largely on component aggregation rather than ground up coding. Execution and control flow are not always predictable through the system. It means we require the expertise of scarce security experts and architects to threat model effectively. Another typical factor is represented by the accelerated solution development lifecycle, typical of the Agile methodologies which are so prevalent nowadays: they require even more to focus the energies of the teams on satisfying the functional requirements rather than anything else.

Therefore, aspects like security tend to be considered as a cost, to be best covered by automated tools and processes, designed with the main intent of reducing impact on development.

Threat modeling faces the same issues. Therefore, it is crucial to improve the efficiency of this practice, if we want it to be relevant within a DevSecOps world.

Phases of threat modeling

Each threat modeling methodology defines its own set of phases. From our point of view, it is possible to identify seven distinct phases that are more or less explicitly present in all the most successful implementations.

Phase 1: Information gathering

At the start of the threat modeling process, we are interested in the context of our threat model. Stakeholders in this phase include developers, product managers, business analysts, security engineers, and security testers. Through a brainstorming exercise, they determine the scope, features, and use cases.

Phase 2: Identification of appropriate threats

The next phase is typically time-boxed and further decomposes the architecture, to identify critical assets. This implies that all threats will not be analyzed, just the most important ones based on the experience of individuals and the collective knowledge base. These assets are reviewed against threat identification categories. Attacks scenarios are created through real-life examples based on the relevant business context. The severity of the threats is determined, most frequently adopting an approach based on impact and probability. At this phase, there is some assessment of the cost involved to determine whether the identified threats are relevant.

Phase 3: Mitigations of threats

In the third phase, mitigations are proposed against the threats identified previously. They are categorized and prioritized based on all threats. The goal is not to produce a comprehensive list of mitigations for each threat, but to cover different types of security controls for achieving defense-in-depth. This will provide different choices for consideration. Each mitigation is associated with a threat to justify and contextualize it. The mitigations must be expressed with unambiguous actionable descriptions and clear verifiable criteria.

Phase 4: Assess mitigations

In this phase, mitigations are assessed against business and technical risk based on prioritized trade-offs. Recommendations are assessed for impact (both positive and negative) and then defined in a proposed roadmap.

Phase 5: Communicate results of threat model

In this phase, the mitigations are presented to the relevant stakeholders. Trade-off decisions are made based on budget, resource capacity, and organizational risk factors. Mitigations should be enforceable.

Phase 6: Update the threat model

Once approval has been given to undertake the mitigations, the threat model is regularly revised to reflect the latest update to the system in terms of risk and design.

Phase 7: Share knowledge and learn continuously

Learnings are fed back to improve threat modeling. Lessons learned are shared with internal and external communities.

Anti-patterns of threat modeling process

We have seen some good implementations of threat modeling, but many more have been affected by problems. This has allowed identifying some common anti-patterns.

1. Forcing tools to do what they were not intended for. We need to know what characteristics to look for depending on the maturity of the overall solution. Experts use different features than novices.

2. Trying to achieve perfection. Analysis tends to focus on completeness rather than knowing how much is "good enough."

3. Using cut and paste rather than thinking about the assumptions that went into a previous threat model.

4. Adopting a rigid threat modeling process for all projects without discriminating on scope or relevance.

5. Focusing on a diagram-based language. In the end, it is a tool; if it works for you, go for it. Otherwise use something else. Infrastructure teams, for example, might find tables work better (a list of objects and can relate to properties).

6. Thinking of threat modeling only as a technical activity and ignoring risk, use/misuse cases, and abuse cases.

7. Thinking of a threat modeling diagram as the final goal. In fact, it is just a starting point.

8. Neglecting a feedback loop to update the threat model. The model should be living and maintained.

9. Neglecting to factor in business risk priorities during threat analysis.

10. Inability to identify a consistent risk definition to allow for comparisons between different systems.

11. Lack of diversity in POVs. By not involving a broad group of stakeholders, there is a reliance on personal bias and assumptions about components or libraries based on previous experience.

12. Reliance only on a checklist approach rather than combining with appropriate analysis.

13. Blind acceptance of unknown system components without taking the time to conduct an in-depth study of the gaps.

14. Focusing on development practices and pitfalls instead of limiting the scope to design.

15. Being fully dependent on abstract knowledge bases and knowledge bases. In fact, they do not provide business context and remain at an abstract level.

16. Focus on the parts instead of the whole or vice versa.

17. Making everything high priority for fear of having mitigations dismissed as unnecessary.

18. Blindly adopting best practice security guidance as mitigations without linking to threats.

Gaps with traditional data flow diagrammatic threat modeling

We identified seven key areas where traditional data flow diagram-based threat modeling fails to effectively deliver for today's DevSecOps world.

1. Speed of updating diagrams

Updating these diagrams is relatively slow since it is largely a manual task. Revisiting a diagram requires re-contextualizing the system and key participants in the original threat modeling effort may already be on other projects.

2. Lack of consistent Threat Modeling process

We have a lack of consistency because there is no global standard on how to create the right threat modeling diagram. It is left up to individual teams based on what they think is important and they bring this bias and insight into their discussions. Current standardization efforts are largely based on the diagram language representation rather than true semantic analysis around security.

3. Emphasis on the system rather than a holistic approach

Considering that Threat Modeling is predominantly carried out during the design phases, most of the time it is the Architect and/or Dev lead who will be heavily involved. Other domains like infrastructure, operations, and CI/CD are rarely considered. This leads to an incomplete vision of the system under assessment.

4. Lack of reusable models

There's a lack of reusability as many teams consider threat modeling to be a one-time activity due to the nuances of a particular system. This is typically managed through the adoption of templates. While templates can help, they typically require all the knowledge for a specific scenario. A byproduct of templates is to theoretically facilitate knowledge retention, but trying to codify this knowledge leads to more complex models. This complexity makes achieving consistency harder, and therefore it may lead to very different results based on who produces the Threat Model.

5. Focus on subsystems rather than the bigger picture

Many times, Threat Modeling results are not consistent because threat modeling diagramming is largely an art. Each team will produce something that looks different based on what they feel is important. This leads to creating Threat Models from a very narrow perspective of the overall system. As a result, the most important threats impacting the system may be lost. Therefore, the Threat Model leads to wasting a lot of effort in trying to mitigate threats of marginal importance.

6. Lack of measured impact

Many teams do not account for the value accrued from Threat Modeling activities. There is no comparison against other security activities, or by not doing it at all. For example, there is no feedback loop on how to effectively prioritize threats against quantitative measurements.

7. Knowledge bases generate a large number of generic threats

Knowledge bases are intended to provide generic mitigation. In the interest of speed, teams sometimes execute on this generic advice without taking into account the specifics of the solution and the business point of view, thereby not identifying threats relevant to the business.

Capturing risk

The reason for a threat model is to identify the security risks of a system. The focus of risk should be on the economics and based on meaningful data around frequency and impact. Unfortunately, people don't do a good job of risk scoring today. We want to use observable quantities to drive objectivity.

Where is threat modeling headed?

Traditional threat modeling relies on security experts and architect experts. This excludes a vast number of other stakeholders from the business and technical teams. We believe the next generation of threat modeling will be layered. It will allow both experts and non-experts to contribute to the threat model with relevant functionality limited to their perspective. Threat modeling will account for experience, where non-experts will be intentionally limited in scope, experts will perform deeper, quantitative risk analysis. The threat model will derive insights developed from other projects and be able to address the full risk lifecycle (identification, mitigation, residual risk) not just at the design phase.

Platform integrations will be a key part of the threat modeling experience. Many useful repositories will be leveraged, such as CWE, CVE, and CAPEC. There will be integrations with other security tools such as SAST, DAST, and risk management. Information viewed through the platform will be relevant to a specific role and meet the context and needs of an organization as needed. The system under analysis will be provided in some codified form and threats will be represented in a flexible, reusable manner.

Mitigations will be at the core of the threat modeling experience. Different tools in the pipeline will focus on different problems in order to paint a cohesive picture.

Threat modeling needs to fit into an Agile workflow. There has to be a process that is flexible enough to address a cross-functional team (developers, project managers, and architects) all having different approaches. This means there is no single way to do threat modeling. We need to allow for other approaches. It also implies the ability to go back and change previous analysis. As such, the threat model becomes a living document or model continually being updated based on multiple views on the truth based on each person's needs.

A new threat modeling pratice

The response to the problems identified with traditional Threat Modeling is twofold:

1. Drive threat modeling use cases through security requirements. Usually, these are tied to common practices like OWASP Top 10 or from standards groups.

2. Create a feedback loop from SecOps back into the Dev teams. This allows for real-time monitoring of anomalies.

Threat Modeling is an important activity, and we need to do it. We just have to evolve our thinking to account for the complexity of our systems today. We need to consider a higher level of abstraction rather than focusing on just lower-level details. Large and complex single template approaches intended to capture all the knowledge tend to be repurposed without due consideration of overall architecture and risk. These templates tend to increase in complexity over time. Instead, we need to create a set of reusable templates that are specialized and focused on

a subset of architectural and risk constraints. Multiple templates can then be used to guide and direct DevSecOps teams for initiating basic use cases. Thereafter, the threat model evolves to become more specific to the solution. And finally, complementing this with SecOps creates the collaborative feedback loop that enables unseen threat vectors to be codified and quickly resolved to contain the vulnerabilities.

A maturity model

We recognize that to achieve our view for a modern threat modeling practice requires to improve over the current one in a number of ways.

This involves actions to increase the Quality of the produced Threat Model, to make it more useful as a tool to evaluate the Risk and identify relevant Mitigations to be implemented. But it also involves evolving Threat Modeling to be central for Security Risk Management, by making it the main part of a LEAN process fully integrated with the various tools and processes adopted by the Organization.

We have identified various categories of tools and processes that are good candidates for being integrated with such Threat Modeling practice, and for each of them we have identified four different maturity levels:

- The first level corresponds to the experience where there is no integration at all, and even Threat Modeling is performed without a specialized tool. Our reference experience for this maturity level is the whiteboard Threat Modeling practice. At this level, Threat Modeling is mostly something done by enthusiasts, with no standardization nor control over the process.

- The second level sees the introduction of rudimentary Threat Modeling tools. There are various freely available out of there, and most of them are not integrated or extensible. The process tends to be a little more standardized, but it is still in its infancy. The Business point of view is not considered: Threat Modeling is merely done from a technical perspective. Still, this level evidently provides additional value over the fully manual experience.

- The third level starts to see some integration capability and the process . At this level, Business is taken more into account and there is some management of the process, mostly with the intent of standardizing the experience. There is some integration, but it is limited.

- The fourth and last level is obviously where the highest maturity is achieved. At this level, the focus is on full control over the process, Continuous Improvement and strong integration with processes and tools in use by the Organization.

For each of those levels, we have identified some typical questions that may be asked to understand if the Organization is at that specific level, and typical answers we expect would be obtained. Those questions and answers should be considered as indicative and can be integrated, based on the specifics of the Organization.

Tool Category	Maturity Level 1 (No Threat Modeling Tool)	Maturity Level 2 (Threat Modeling Tools)	Maturity Level 3 (Threat Modeling Patterns)	Maturity Level 4 (Risk Driven Knowledge base)
Applications and Services Inventory	**Quality related audit questions:** Does the inventory contain key assets that are in your TM? Do you know the owner for each key asset in your TM? **Evidence:** Inventory and owner of assets in a spreadsheet format	**Quality related audit questions:** Do you have clear relationships across the stack, between key assets in the inventory? Can you output Infrastructure as Code, Security as Code, Compliance as Code from your inventory tool? **Evidence:** A representation or list of relationships that link assets together (ex. component diagram)	**Quality related audit questions:** Are the assets in your TM monitored against CWE, CVE codes? Are the key inventory assets in your TM integrated with an incident management process? Do you know the primary and secondary risk stakeholders of key inventory assets in your TM? **Evidence:** A list, from the tool, of prioritized assets based on risk Incident management process has information about assets, from the tool	**Quality related audit questions:** Do you regularly perform a risk assessment of the key inventory assets? Is the business included in the asset risk assessment? Is the asset risk assessment contextual? **Evidence:** List of assets that are prioritized and classified against business risk Process for updating an asset status throughout its lifecycle Report on usage of each asset across the organization
Application & infrastructure architecture system modeling	**Quality related audit questions:** Does the system model include all key systems (external interactors) in your TM? Is there some basic system classification used in your TM? Does you TM address the right system context? **Evidence:** Hand drawn or whiteboard representation of the system (including classification and context)	**Quality related audit questions:** Does your TM include tool representations (ex. secrets management, IAM)? Does your TM have system trust boundaries? Does your TM comply with common system criteria such as organizational policies and regulations? **Evidence:** A diagram or simple language that accurately represents static knowledge of the system, including trust boundaries and common criteria	**Quality related audit questions:** Does your TM make use of reusable, self contained system templates? Is your TM using multiple layers of abstraction across different system diagrams? Is the business system flow documented in your TM? **Evidence:** System details including reusable stacks for library, components and service relationships Layered diagram that includes top down representation of systems, dependencies, and business flows	**Quality related audit questions:** Do you associate multiple threat models for your system? Are you accounting for different contexts (infrastructure, application, data)? **Evidence:** Rich and dynamic report that provides contextualized views at multiple levels of detail including business and technical

Tool Category	Maturity Level 1 (No Threat Modeling Tool)	Maturity Level 2 (Threat Modeling Tools)	Maturity Level 3 (Threat Modeling Patterns)	Maturity Level 4 (Risk Driven Knowledge base)
Threat categorization such as CWE codes	**Quality related audit questions:** N/A **Evidence:** N/A	**Quality related audit questions:** Are you using threat categories to identify threats in your TM? Are you using the threat categories to identify mitigations in your TM? **Evidence:** List of threats per chosen threat category Map of mitigations to a threat category	**Quality related audit questions:** Are you using industry knowledge bases (CAPEC, CWE, ATT&CK) to inform your threat model? Do you have internal knowledge bases to construct your threat model? Are your threats across different systems associated to create new or different insights? **Evidence:** Report that shows links between threat modeling with industry and internal categories Report of new threats and insights for	**Quality related audit questions:** Do you have a structured approach to maintain and apply your internal TM knowledge base across its lifecycle? Are you constructing threats in your TM that are relevant to the business? **Evidence:** Report or dashboard on updated content in the knowledge base Report aligning CWEs to business risk and priorities

Tool Category	Maturity Level 1 (No Threat Modeling Tool)	Maturity Level 2 (Threat Modeling Tools)	Maturity Level 3 (Threat Modeling Patterns)	Maturity Level 4 (Risk Driven Knowledge base)
Mitigation categorization tool	**Quality related audit questions:** Have you identified one or more mitigations for each threat? **Evidence:** Something planned for each threat.	**Quality related audit questions:** Have you defined standard mitigations (maybe using a defensive framework like STRIDE, OCTAVE, etc. as guidance) against threat categories? **Evidence:** List of mitigations that are categorized based on the selected or chosen category.	**Quality related audit questions:** Are you extending the categories beyond preventative to include, for example, corrective and detective mitigations? **Evidence:** List of recommendations or possible roadmap to stakeholders that addresses the risk through various mitigation categories. Identification of mitigation against all identified threats through the various layers to achieve Defense in Depth	**Quality related audit questions:** Do you have a structured approach to maintain and apply your internal knowledge base (knowledge lifecycle)? Is the knowledge accessible to others for determining additional mitigations? **Evidence:** Report or dashboard on updated content in the knowledge base Dashboard that presents a hierarchical relationship of mitigations to threat categorization (not limited to technical perspectives but also business, compliance, risk) Result of "what if?" analysis against product or enterprise architecture services.
Static Threat Model Analysis Tool	**Quality related audit questions:** Are you using best practice guidelines (ex. OWASP Top 10) as part of the TM process? **Evidence:** List of problems constructed manually based on guidelines	**Quality related audit questions:** Is there an integrated process to automatically validate the TM and identify problems in the tool? Are you updating the TM as soon as errors are discovered? **Evidence:** Report of modeling errors identified	**Quality related audit questions:** Is this integrated into your automated DevOps pipeline? Is there a feedback loop that automatically create TM update tasks and activities? **Evidence:** Status report form CI server that shows result Results are shared with stakeholders for manual update in DevOps pipeline if needed	**Quality related audit questions:** Can you create your own policies or rules? Is there an automated feedback loop to business and technical teams so that it creates additional rules for your TM? Is there a centralized view around quality of all your TMs (dashboards)? **Evidence:** Library that shows proper usage of rules Customizable dashboard across multiple TMs and across the enterprise

Tool Category	Maturity Level 1 (No Threat Modeling Tool)	Maturity Level 2 (Threat Modeling Tools)	Maturity Level 3 (Threat Modeling Patterns)	Maturity Level 4 (Risk Driven Knowledge base)
Dynamic Threat Model Analysis Tool (Simulated data flow)	**Quality related audit questions:** Did you have someone playing the threat agent and another playing the defender? **Evidence:** Whiteboard of potential attack scenarios (including abuse cases) linked with effective mitigations against those attacks	**Quality related audit questions:** Have we identified key users (including malicious insiders and attacks to privileged roles) with various privileges and tested whether the mitigations prevent a breach? Have you considered known attack scenarios from the past and contextualized to your systems? Have you tested all documented assumptions about external systems and services that interact with the system in question? **Evidence:** Report, diagram, or metadata of potential attack scenarios linked with effective mitigations against those attacks Mitigations manually grouped by effectiveness and relationships (complementary vs alternative) between them	**Quality related audit questions:** Have we simulated the most relevant environment and scenarios aligned according with organizational risk (from knowledge base or public repository)? Are the simulation rules easily understandable and manageable? Have you considered the most important attacker profiles as aligned with organizational risk for your scenarios? **Evidence:** List of simulation rules that account for the environment and scenarios Report of organized and categorized rule sets (CRUD operations against simulation rules) Report on list of attacker profiles Mapping of high priority attack trees (implying specific attacker profiles) against simulation rules Formalized and structured report (e.g. attack trees) of simulated scenarios and optionally mapping to industry recognized categories (e.g. MITRE ATT&CK) List of attack scenarios and mitigations that include multiple weaknesses	**Quality related audit questions:** Do the simulation rules map to business priorities and specific scenarios (such as regulatory compliance, architecture, functional cases, etc)? Do you learn and expand the simulation rules and scenarios based on the attacks you are seeing? Are you continuously improving based on known KPIs (accuracy of % of false positives, calculated vs real severity, coverage of threats)? **Evidence:** Simulation rules to acceptable business risk policies are mapped and stored in the platform Results of simulations against the business priorities and specific scenarios Validate the impact of recommended mitigations and identify residual risk. The platform learns from existing behaviors by the adoption of Machine Learning algorithms to improve capability of simulation models. Metadata or comment on rules showing the origin of the rule itself, what has been changed and why

Tool Category	Maturity Level 1 (No Threat Modeling Tool)	Maturity Level 2 (Threat Modeling Tools)	Maturity Level 3 (Threat Modeling Patterns)	Maturity Level 4 (Risk Driven Knowledge base)
Root Cause Analysis Tool	**Quality related audit questions:** Have you applied an informal RCA approach to a breach? Have you changed anything in your system as a result? **Evidence:** Initial RCA findings or incident report Fishbone diagram	**Quality related audit questions:** Do we have a process for RCA? Are you using Threat Models to inform RCA? Are you using out of the box RCA templates? **Evidence:** Some sort of document that explains RCA	**Quality related audit questions:** Do we have a formal and repeatable process for RCA? Have you categorized the RCA output into people/process/ technology? What corrective actions were taken? Did the RCA lead to an updated TM which helped to re- prioritize the business risk? **Evidence:** Incident analysis report Past potential risks and issues that occurred Updated threat model and mitigations based on RCA	**Quality related audit questions:** Is the RCA process continually improved and tied to KPIs? Is the RCA process linked to the knowledge base so that feedback can improve the system security? Is there a repeatable process to include multiple teams for updating their assets (ex. knowledge base, threat model, test cases, etc.) based on RCA? **Evidence:** Report that links the RCA with the learnings for improvement (ex. misuse cases, configuration, etc.) KPIs related to RCA impact (ex. # of findings, other qualitative metrics, etc.) Recommended mitigations and how risk profile changes
Dashboard/reporting system	**Quality related audit questions:**	**Quality related audit questions:** Correlation heat maps to prioritize security risk	**Quality related audit questions:** Specialized reports for different roles to understand risk Pareto analysis for risk	**Quality related audit questions:** Does it integrate with a risk assessment framework? **Evidence:** Stakeholders Business Impact

Tool Category	Maturity Level 1 (No Threat Modeling Tool)	Maturity Level 2 (Threat Modeling Tools)	Maturity Level 3 (Threat Modeling Patterns)	Maturity Level 4 (Risk Driven Knowledge base)
Intelligence Knowledge base that contain information about threats and mitigations	**Quality related audit questions:** Are you complying with existing security policies? Do you have a reference checklist to guide your effort? Is your recommendation aligned with system security policies relevant to the high level architecture? Are you using some type of industry knowledge base as a reference? **Evidence:** Recommendations for prioritized mitigations	**Quality related audit questions:** How do you know whether your recommendations address all your threats and mitigations? Have you used a checklist from an existing knowledge base? (CIS) **Evidence:** Formal recommendations for prioritized mitigations	**Quality related audit questions:** Can we identify a credible story about how each attack could happen and be mitigated? **Evidence:** Report (attack trees, textual representation, or mindmaps) specific to the context of the application(s) under consideration. Relationships (conjunction/ disjunction) between mitigations in the knowledge base	**Quality related audit questions:** Is the organizational knowledge base up to date and relevant? How often is this updated? Is there an automated way of generating insights and recommendations? **Evidence:** Best recommended mitigations against threats
Threat severity calculator	**Quality related audit questions:** **Evidence:** List of qualitative (H/M/L) assignment based on personal experience	**Quality related audit questions:** Have you collaborated with stakeholders from other domains (architects, product managers, product owners)? Have you clearly defined what H/M/L means? **Evidence:** List of prioritized recommendations based on some form of calibration (eg. STRIDE, Bug Bars) List of calibrated severities	**Quality related audit questions:** **Evidence:** Sampling for quantitative analysis	**Quality related audit questions:** **Evidence:** Full quantitative analysis (eg. FAIR) Knowledge base that contains all relevant information to perform a quantitative analysis Automated generation of insights on severities Integration with business impact value streams across the operational lifecycle

Tool Category	Maturity Level 1 (No Threat Modeling Tool)	Maturity Level 2 (Threat Modeling Tools)	Maturity Level 3 (Threat Modeling Patterns)	Maturity Level 4 (Risk Driven Knowledge base)
Issue Tracker	**Quality related audit questions:** **Evidence:** Manual insertion of prioritized mitigations into system requirements	**Quality related audit questions:** **Evidence:** Mitigations are manually mapped to existing functionality in the issue tracker (cases, tasks, stories, epics) Mitigations are manually mapped to new functionality in the issue tracker (cases, tasks, stories, epics)	**Quality related audit questions:** **Evidence:** Mitigations are prioritized based on business and product team needs Maintain relationship between mitigation and threat Upon completion of a case, the threat model is updated manually	**Quality related audit questions:** **Evidence:** Realtime bidirectional view between issue tracker and threat model Reporting in issue tracker on various cases that map to risk categories Richness of dynamic metadata (complexity, risk, probability, cost) enables filtering of cases in issue tracker based on different stakeholders
Event and Monitoring Management and Incident Response (MITRE ATT&CK, STIX, TAXII)	**Quality related audit questions:** Are you examining the out of the box monitoring reports? **Evidence:** Annotated report from monitoring system	**Quality related audit questions:** Is the team identifying risk in production? Are the monitoring reports contextualized for business needs? Are you communicating event tracking and response based on the threat model (ex. SQL injection)? **Evidence:** OpSec report based on organizational security policies	**Quality related audit questions:** Are you integrating with a SIEM that extracts security incidents and attacks based on best practices? Are you correlating high risk security events and contextual details and patterns from the SIEM on a regular basis? Are you updating your threat model based on input from event tracking? **Evidence:** Manual updated of threat model with business risk attributes appended through observing attack patterns and security insights on the system. Revised list of prioritized actionable mitigations	**Quality related audit questions:** Are you using production system predictive analytics to feed into the knowledge base? Are you making use of up to date predictive analysis engines? Active Threat Monitoring **Evidence:** Updated stakeholder threat model with contextual (business, security, compliance) recommendations based on current trends Risk information is dynamically fed into risk assessment processes based on patterns observed in the production systems.

Tool Category	Maturity Level 1 (No Threat Modeling Tool)	Maturity Level 2 (Threat Modeling Tools)	Maturity Level 3 (Threat Modeling Patterns)	Maturity Level 4 (Risk Driven Knowledge base)
OSINT tools (Shodan, Searchcode)	**Quality related audit questions:** Are you examining the out of the box monitoring reports? **Evidence:** Annotated report from OSINT tools	**Quality related audit questions:** Is the team identifying risk in production based on OSINT reports? Are you integrating feeds from OSINT tools in the threat model (ex. common patterns)? **Evidence:** Threat model that uses data from OSINT tools (could be manual and annotated)	**Quality related audit questions:** Do you have a process in place (ex. PSIRT) to integrate output from OSINT tools into your threat model? Are you integrating with OSINT tools that extract security incidents and attacks based on best practices? Are you correlating high risk security events and contextual details and patterns from the OSINT tools on a regular basis? **Evidence:** Automated annotation of an updated link to OSINT output in the threat model Trend report on attack scenarios that might increase risk to the business Revised list of prioritized actionable mitigations.	**Quality related audit questions:** Are you sharing high risk security events and contextual details and patterns from the OSINT tools on a regular basis with all stakeholders? Are you using OSINT for predictive analytics to feed into the knowledge base? **Evidence:** Updated stakeholder threat model with contextual (business, security, compliance) recommendations based on current trends Risk information is dynamically fed into risk assessment processes based on patterns observed in the production systems.

Tool Category	Maturity Level 1 (No Threat Modeling Tool)	Maturity Level 2 (Threat Modeling Tools)	Maturity Level 3 (Threat Modeling Patterns)	Maturity Level 4 (Risk Driven Knowledge base)
Risk management system (risk identification, mitigation approval, threshold, ..)	**Quality related audit questions:** Are you using the outcomes from threat modeling to understand the risks and act upon them? Are you involving the system owners in assessing risk? **Evidence:** Identification of the risks based on your own experience (informal, Excel)	**Quality related audit questions:** Is the threat model loosely based on some standards related to risk management and risk assessment? Are you analyzing assets based on business sensitivity (security, privacy, legal, revenue, etc)? **Evidence:** Loose association of threat model elements with internal or external standards (OWASP Top 10, SANS)? Asset identification and importance to the business	**Quality related audit questions:** Are you using a standardized risk management process? Are you referring to standards and regulations to inform various domain specific risk factors for analysis (compliance, regulations, etc)? **Evidence:** A repeatable process with standardized outcomes Compliance or regulatory report with specific requirements being met	**Quality related audit questions:** Are you using an automated risk management process? Are all stakeholders able to interact and provide feedback with a single, unified, view of risk profile for the solution? Is the risk management process continually improving? **Evidence:** Realtime, integrated report of the business risk exposure for a given system "What if?" analysis against different recommendations to help reduce risk Report on formal approvals obtained. Existence of evolving KPIs around risk factors for continual improvement. Standardized way to represent risk in an unbiased way (ex. quantitative risk assessment) Continuous compliance trend report or diagrams of risk assessment (quantitative)

Tool Category	Maturity Level 1 (No Threat Modeling Tool)	Maturity Level 2 (Threat Modeling Tools)	Maturity Level 3 (Threat Modeling Patterns)	Maturity Level 4 (Risk Driven Knowledge base)
Threat Modeling Value Stream Alignment	**Quality related audit questions:** Are your business stakeholders (Product Manager level) informed? **Evidence:** Story in the form of informal requirements (aligned to business value) from a conversation	**Quality related audit questions:** Is the recommendation contextual to the business goals and priorities? Are the impacts of mitigations aligned to the business goals and priorities? Are you using a repeatable process (ex. minimum security baseline of questions)? **Evidence:** Story in a structured way based on personal experience Story in the form of high level list of threats and mitigations	**Quality related audit questions:** Is there a formalized alignment between business goals and the impact of threats and proposed mitigations? Can you filter the requirements based on frameworks, standards, or risk? Is the threat model regularly updated through feedback during the value streams? Can you provide a report on the security posture against a given standard or framework? **Evidence:** Story in the form of a traceability matrix back to a standard or framework Story in the form of mitigations are prioritized based on business and product team risk needs Story in the form of a risk relationship between mitigation and threat Output is aligned with value stream tools	**Quality related audit questions:** Are the mitigations mapped to business priorities and context (process, impact)? Is there a process to review all mitigations against business priorities? What are the processes to ensure that security teams are aligned with the business? Is there a continuous improvement process to measure the quality, effectiveness and efficiency of the TM service? Can you produce a report that shows who has approved the current activities on the Threat Model? Does the Threat Model adhere to internal and external policies? **Evidence:** Quantified security risk attribute produced for each requirement Business prioritization attribute produced for each requirement Non technical business report that allows dynamic analysis based on risk, probability, severity, and monetary considerations Audit report of all changes and approvals in the Threat Model.

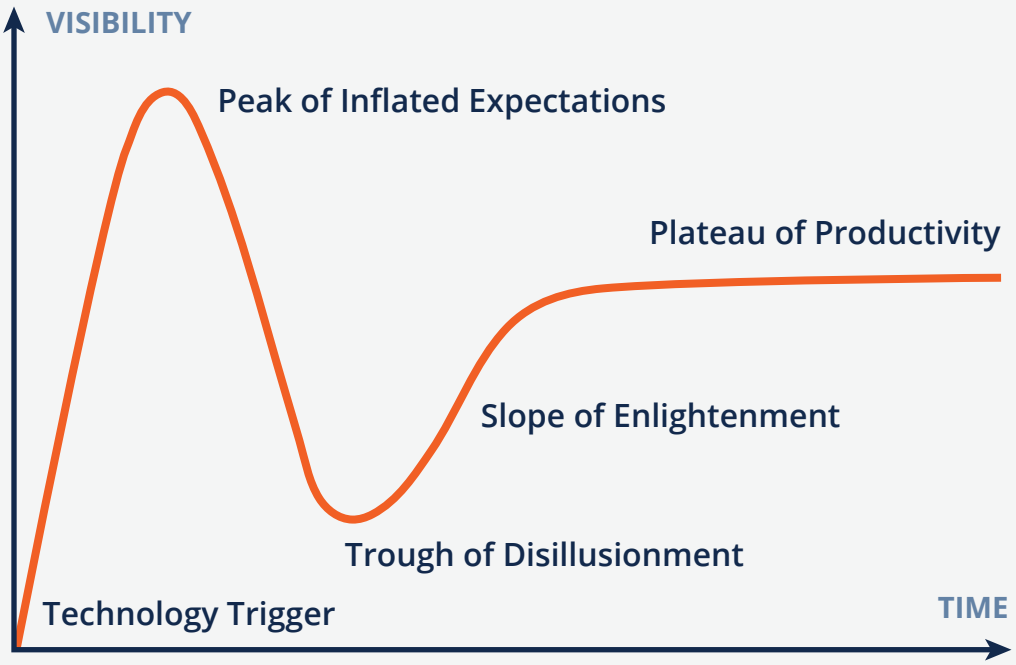
Conclusion

Threat modeling has a lot of potential. We can see an increase of interest for this practice, lately. But threat modeling also has important gaps that need to be addressed. The risk is to get into a Peak of Inflated Expectations, which would be followed very soon by a Trough of Disillusionment.

We are seeing this already: organizations having embraced threat modeling a while back are starting to feel the push from business, because the adopted process is seen as an expensive exercise with very limited outcomes. In some cases, it is causing projects to be blocked from being delivered to production for weeks or even months. As a result, organizations tend to see threat modeling as a bureaucratic task, a necessary evil with no practical value.

As it is, threat modeling is not empowering business, but is blocking it.

The best implementations are already providing a lot of value to customers, by helping to identify significant risks and important activities to be done to make those risks acceptable in a timely manner.



Source: The Gartner Hype Cycle

The problem is that traditional approaches are no longer scalable or accurate in an agile, cloud-based, microservices world. We need to include multiple stakeholders in the process to gain from the diversity of experience offered by both business and technical people. We also need to evolve our approach from a tool perspective into a shared platform perspective where integration with other tools and datasets is possible.

To fulfill its potential, threat modeling must evolve. We need to understand that an integrated experience is essential to make threat modeling useful for the business. We need to make it an integral part of risk management and to ensure that the organization implements a virtuous cycle to continuously improve the practice.

Threat modeling is too important to be left in the hands of individual threat modelers.

Next Steps

1. Determine your current state:

Assess prior threat models, understand the business objectives, assess the level required for the practice, all these will help us determine the maturity level of your practice. Obtaining the maturity level will help us ask the right audit questions and obtain the required outcome.

2. Determine the right future state for you:

Select the right tools and integrations based on costing and feasibility with current tools, and optimizing effectiveness with current processes. Emphasis should be on optimizing for quality of threat modeling. You don't need to implement the highest level of maturity, focus on what makes the most sense.

3. Implement the roadmap:

Prioritize the activities to achieve quick wins and then gradually expand into elements for your future state. Look for additional opportunities to further increase and contextualize your threat modeling practice by onboarding other services surrounding the tools described in this document.

Readers are invited to explore other thoughts in the broader Threat Modeling domain:

- **Threat Modeling Manifesto**
- **Schoenfield, B. S. E. (2015). Securing Systems: Applied Security Architecture and Threat Models (1st ed.). CRC Press.**
- **Shostack, A. (2014). Threat Modeling: Designing for Security (1st ed.). Wiley.**
- **Tarandach, I., & Coles, M. J. (2020). Threat Modeling: A Practical Guide for Development Teams (1st ed.). O'Reilly Media.**
- **UcedaVelez, T., & Morana, M. M. (2015). Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis (1st ed.). Wiley.**
- **Appsec Development: Keeping it all together at scale**
- **Threat Modeling Security Fundamentals**
- **Building Secure Software: It Takes a Champion**
- **Michael Howard Teaches Threat Modeling**
- **Threat Modeling a Retail Environment**
- **Being a better Threat Modeler**
- **OWASP Threat Model Cookbook**
- **Threats Manager Studio**
- **Threat Modeling: A Survey of Available Methods**

Authors:

Simone Curzi

Security Expert, Author of Threats Manager Studio

Simone Curzi is a Principal Consultant from Microsoft Consulting Services. Simone has 20 years of experience covering various technical roles in Microsoft Services, and has fully devoted himself to Security for more than 5 years. A renowned Threat Modeling and Microsoft Security Development Lifecycle (SDL) expert, Simone is also one of the leaders of the Worldwide Microsoft Community on Application Security and a SME for the Security Community.

Some of Simone's contributions are available through his Blog. He can also be reached via LinkedIn.

Jack Freund

Head of Cyber Risk Methodology, VisibleRisk

As Head of Cyber Risk Methodology for VisibleRisk (the Moody's/Team8 JV), Jack has overall responsibility for the systemic development and application of frameworks, algorithms, and quantitative and qualitative methods to measure cyber risk. Previously, Jack was Director, Risk Science at quantitative risk management startup RiskLens. Jack has 22 years of experience consulting, building, and leading technology and risk management programs for Fortune 100 organizations including TIAA, Nationwide, and Lucent Technologies.

Jack was awarded a Ph.D. in Information Systems after his research in disaster informatics and cyber resilience at Nova Southeastern University. He also holds a Masters in Telecommunications and Project Management and a BS in CIS. He holds the CISSP, CISA, CISM, CRISC, CGEIT, CDPSE, CIPP, and PMP designations. Jack has been named a Senior Member of the IEEE and ACM, a Fellow of the IAPP and FAIR Institute, and a Distinguished Fellow of the ISSA. He is the 2020 recipient of the (ISC)2 Global Achievement Award, 2018 recipient of ISACA's John W. Lainhart IV Common Body of Knowledge Award, the FAIR Institute's 2018 FAIR Champion Award, and presented Nova Southeastern University's Distinguished Alumni Award.

Arun Prabhakar

Security Consultant, Security Compass

Arun is a Senior Consultant in the DevSecOps practice at Security Compass. He has distinctive and resourceful experience in Secure System Development Life Cycle activities including secure design, threat modeling, vulnerability management and solutioning across different domains and platforms. Arun has a keen interest in transformative technologies like Machine Learning, Blockchain and IoT He is a Certified Information Systems Security Professional (CISSP).

Altaz Valani

Director of Insights Research, Security Compass

Altaz Valani is the Director of Insights Research at Security Compass. He conducts ongoing research in the Software Security domain. Prior to joining Security Compass, he was a Senior Research Director and Executive Advisor at Info-Tech Research Group providing trusted advice around application development, application rationalization, agile, cloud, mobile, and the SDLC. Other past positions include Senior Manager at KPMG, and other positions working alongside senior stakeholders to drive business value through software development. Altaz is currently Vice Chair of The Open Group Security Forum, a member of the SAFECode Technical Leadership and CIO Strategy Councils, and also sits on several IEEE Working Groups where DevSecOps and Privacy challenges are being tabled at the international standards level.

Hasan Yasar

Technical Director, Adjunct Faculty Member at SEI, Carnegie Mellon University

Hasan Yasar is the Technical Director of Continuous Deployment of Capability group in Software Engineering Institute, CMU. Hasan leads an engineering group to enable, accelerate and assure Transformation at the speed of relevance by leveraging DevSecOps, Agile, Lean AI/ML and other emerging technologies to create a Smart Software Platform/Pipeline. Hasan has more than 25 years' experience as senior security engineer, software engineer, software architect and manager in all phases of secure software development and information modeling processes. He is also the Adjunct Faculty member in CMU Heinz College and Institute of Software Research where he currently teaches "Software and Security" and "DevOps: Engineering for Deployment and Operations".

Other Contributors:

Anupriya Basu - Editor
Vernon Villanueva and Morgan Dunbar - Design
Janet Khoshaba and Raquel Rodrigues - Community Support and Promotion

Warranty Disclaimer

The content in this document is provided "AS IS" for general information purposes only and does not constitute professional advice or any sort of professional relationship between you and the authors. The authors confer no rights to you, and make no representation or warranty of any kind, express or implied, regarding the accuracy, adequacy, reliability, validity, availability, or completeness of the content in this document. The opinions in this document are those of the authors only, and does not represent the opinions, thoughts, intentions, plans or strategies of the authors' employers.