Threat Modeling Best Practices
**What Are the Benefits of Doing Threat Modeling?**

Application threat modeling is one of the most critical activities for building secure applications. It helps you understand and define security requirements for a system. It also helps you identify the underlying threats that an application could be exposed to and define the risk tolerance for an application. The risk tolerance of applications varies. For example, risk tolerance for an autopilot component of a flight management system cannot be the same as that of an online game.

Threat modeling also helps uncover security holes earlier in the software development life cycle. In today’s world, we see security breaches happen every day. A breach not only impacts the reputation of an organization, it also leads to financial losses and legal implications. For example, the 2011 PlayStation breach, which affected more than 77 million online accounts, cost Sony $170 million and led to a shutdown of the company’s online gaming service for 23 days. Sony was also fined $395,000. Threat modeling helps reduce the huge cost due to late defects and enables the delivery of a product with a better security posture.

**Threat Modeling Best Practices**

Let’s look at some best practices for threat modeling.

**Start early**

More often than not, threat modeling activity is started late in the development cycle. You should avoid starting threat modeling post-development. Remember, the cost of fixing security issues toward the end of the development cycle is high. Threat modeling should be performed as soon as the product’s architecture and design are sufficiently defined. This will enable you to identify threats and suggest mitigations at the design stage.
Know the application internals
The more you know the application internals, the better your threat modeling results. Before starting with threat modeling, you should carefully review all requirement and design documents. Make note of critical flows and data elements. It is highly recommended that you draw your own detailed data flow diagram—this will help you get a better understanding of the application internals.

Understand the underlying application domain and technology stack
If you are not familiar with the domain for which the application is being built, you might not be able to understand the security risks involved. For instance, if an application is being designed for the healthcare domain, you need to be familiar with data privacy requirements and other compliance regulations for the healthcare domain while doing threat modeling. Having domain knowledge is essential, as it helps you identify the risks and mitigate them by building the right defenses into the design.

Similarly, every technology has its own dos and don'ts. A good understanding of those enables you to design your application with the right defenses.

Have a complete data flow diagram
The lack of a data flow diagram is often cited as a common reason for not having a threat model. If you are responsible for creating the threat model and you realize that there is no data flow diagram available, you can take this approach:

- Review all the available details related to the product requirements and design.
- Draw a high-level data flow diagram based on your understanding.
- Have it reviewed with the architect and team members, take notes, and make flow changes based on their inputs.
- Once the high-level data flow is defined, shift your focus at each component/module by including the detailed flow for them.
- Get component flow-level interaction reviewed with individual component owners.

Drawing your own data flow diagram helps you gain a better understanding of the application internals because you get to interact with the architects and component owners.

Create the threat modeling diagram
Once you have the complete data flow diagram defined, the next step is to derive the threat modeling diagram. Here are a few tips:

- Use color coding.
- Draw trust boundaries.
- Mark the entry point.
Get Better at Threat Identification

Threat modeling is an art: it gets better with experience. Below are some ideas that will help you improve your ability to identify threats:

Ask questions

Focus on the potential attack surface (the collection of entry points) and brainstorm to understand what the possible threats could be. Ask yourself these kinds of questions:

- What needs to be protected? (Examples: user data, user system, encryption keys, communication channel, third-party library calls, and more)
- What are the consequences of an attack? (Examples: privilege elevation, leakage of sensitive information, and impact on brand image)
- What mitigations are already present?

Here is another set of relevant questions for an application flow:

- What is the goal of a specific flow in the application?
- From where does it take input?
- Can the source of this input be trusted?
- What input validations measures are in place?
- How does it produce output?
- What is the output?
- Does it use any configuration or system files to generate the output?

Get it reviewed

It’s a good idea to have the threat model reviewed by:

- Experienced security professionals and mentors
- Product development team and stakeholders

Such review discussions are likely to bring in new threat vectors that were not thought of earlier.

Read and research

You should read and research known security vulnerabilities that similar applications had faced and plan defenses to mitigate those in your application. You should also research best practices relevant to your application domain.

Keep it up to date

Many times, teams consider threat modeling as merely a checklist item in their DoDs, and they just create a very basic version of the threat model that is never updated. Outcomes of such threat modeling efforts are not helpful, as they fail to dig deeper into design layers. Remember, threat modeling is a live document, and it must be kept up to date release over release until the application is at end of life (EOL).

An up-to-date and complete threat model must be referenced while running penetration tests, as it could help you identify new attack vectors. It must also be referred to while making any design changes to see if there is an impact on existing defenses. This could also result in additional requirements for any new mitigations to be built in due to the new change that is introduced.
Brainstorming session

Once the threat model is sufficiently defined, organize a brainstorming session with the entire project team, including developers, quality assurance testers, architects, managers, and DevOps. As part of this session, walk the team through the threat model for the first 30 minutes, followed by a round-robin brainstorming session to identify new threat vectors.

Having a disciplined brainstorming session with the entire project team will help you uncover hidden threats. Each member of the team has their own area of expertise with regard to the application. During these brainstorming sessions, they bring their own perspective, which aids in identifying threats that otherwise would not be thought of. Here are a few recommendations for running brainstorming sessions:

- Set a clear agenda (for example, brainstorming on specific application modules and flows).
- Seat all members in a round table arrangement to facilitate the discussion.
- Make the session time bound.
- Go around the table to generate threat ideas and note each of them without getting into longer debates.
- Post-brainstorming, review each of the threat ideas and analyze them further to understand the potential risks.

Enumerate threats, regardless of mitigation status

It’s worth noting that threat modeling is not just about creating threat modeling diagram. It’s also about analyzing the diagram to identify potential threats and reviewing/suggesting mitigations. When identifying threats, always focus on enumerating threats and not vulnerabilities. Look for what could go wrong with the application in the threat model, and focus on enumerating all threats, regardless of whether mitigations exist or not. Once threats are listed, also document whether each of the threats has any mitigation in place. If mitigation exists, note whether it mitigates the threat completely or partially, and provide a detailed description.

Review and Revisit

Security is an ever-evolving domain, and so is our understanding of it. Each day we acquire new information and knowledge, which can be applied to improving our existing threat models. Furthermore, threat modeling is more like an art, as there is always room for improvement. It's a good idea to revisit existing threat models at least once every quarter, with the goal of applying the new perspective and knowledge you have gained.

Threat modeling is a very powerful tool in the security development lifecycle, but it is effective only if it is done correctly—a half-hearted effort will not yield the desired results. Unlike automated tools (such as static code analyzers, vulnerability scanners, and others), which have a generic, predefined set of checks, threat modeling focuses entirely on your system’s architecture and design.

For beginners, threat modeling can be quite challenging and requires a lot of patience. Remember, it’s an art that takes a lot of effort and disciplined practice to master.
About McAfee
McAfee is the device-to-cloud cybersecurity company. Inspired by the power of working together, McAfee creates business and consumer solutions that make our world a safer place. By building solutions that work with other companies’ products, McAfee helps businesses orchestrate cyber environments that are truly integrated, where protection, detection, and correction of threats happen simultaneously and collaboratively. By protecting consumers across all their devices, McAfee secures their digital lifestyle at home and away. By working with other security players, McAfee is leading the effort to unite against cybercriminals for the benefit of all.