We will cover this topic with a hands-on exercise later throughout Ghanimah.com portal

Reference <https://owasp.org/www-community/Threat_Modeling>

**Threat Modeling**

**Author:** Victoria Drake

**Overview**

Threat modeling works to identify, communicate, and understand threats and mitigations within the context of protecting something of value.

A threat model is a structured representation of all the information that affects the security of an application. In essence, it is a view of the application and its environment through the lens of security.

Threat modeling can be applied to a wide range of things, including software, applications, systems, networks, distributed systems, Internet of Things (IoT) devices, and business processes.

A threat model typically includes:

* Description of the subject to be modeled
* Assumptions that can be checked or challenged in the future as the threat landscape changes
* Potential threats to the system
* Actions that can be taken to mitigate each threat
* A way of validating the model and threats, and verification of success of actions taken

Threat modeling is a process for capturing, organizing, and analyzing all of this information. Applied to software, it enables informed decision-making about application security risks. In addition to producing a model, typical threat modeling efforts also produce a prioritized list of security improvements to the concept, requirements, design, or implementation of an application.

In 2020 a group of threat modeling practitioners, researchers and authors got together to write the [Threat Modeling Manifesto](https://www.threatmodelingmanifesto.org/) in order to “…share a distilled version of our collective threat modeling knowledge in a way that should inform, educate, and inspire other practitioners to adopt threat modeling as well as improve security and privacy during development”. The Manifesto contains values and principles connected to the practice and adoption of Threat Modeling, as well as identified patterns and anti-patterns to facilitate it.

**Objectives of Threat Modeling**

Threat modeling is a family of activities for improving security by identifying threats, and then defining countermeasures to prevent, or mitigate the effects of, threats to the system. A threat is a potential or actual undesirable event that may be malicious (such as DoS attack) or incidental (failure of a Storage Device). Threat modeling is a planned activity for identifying and assessing application threats and vulnerabilities.

**Threat Modeling Across the Lifecycle**

Threat modeling is best applied continuously throughout a software development project. The process is essentially the same at different levels of abstraction, although the information gets more and more granular throughout the lifecycle. Ideally, a high-level threat model should be defined early on in the concept or planning phase, and then refined throughout the lifecycle. As more details are added to the system, new attack vectors are created and exposed. The ongoing threat modeling process should examine, diagnose, and address these threats.

It is a natural part of refining a system for new threats to be exposed. For example, when you select a particular technology – such as Java for example – you take on the responsibility of identifying the new threats that are created by that choice. Even implementation choices such as using regular expressions for validation can introduce potential new threats to deal with.

Updating threat models is advisable after events such as:

* A new feature is released
* Security incident occurs
* Architectural or infrastructure changes

**Threat Modeling: Four Question Framework**

A possible threat exists when the combined likelihood of the threat occurring and impact it would have on the organization create a significant risk. The following four question framework can help to organize threat modeling:

* What are we working on?
* What can go wrong?
* What are we going to do about it?
* Did we do a good job?

There are many methods or techniques that can be used to answer each of these questions. There is no “right” way to evaluate the search space of possible threats, but structured models exist in order to help make the process more efficient.

Attempting to evaluate all the possible combinations of threat agent, attack, vulnerability, and impact is often a waste of time and effort. It is helpful to refine the search space in order to determine which possible threats to focus on.

* **Assess Scope** - What are we working on? This might be as small as a sprint, or as large as a whole system.
* **Identify what can go wrong** - This can be as simple as a brainstorm, or as structured as using STRIDE, Kill Chains, or Attack Trees.
* **Identify countermeasures or manage risk** - Decide what you’re going to do about each threat. That might be to implement a mitigation, or to apply the accept/transfer/eliminate approaches of risk management.
* **Assess your work** - Did you do a good enough job for the system at hand?

**Structured Threat Modeling Process**

A structured, formal process for threat modeling of an application is described in [Threat Modeling Process](https://owasp.org/www-community/Threat_Modeling_Process).

**Benefits**

Done right, threat modeling provides a clear “line of sight” across a project that justifies security efforts. The threat model allows security decisions to be made rationally, with all the information on the table.

The threat modeling process naturally produces an assurance argument that can be used to explain and defend the security of an application. An assurance argument starts with a few high level claims, and justifies them with either subclaims or evidence.

**Further Reading**

* [Threat Modeling Process](https://owasp.org/www-community/Threat_Modeling_Process)
* [Microsoft’s Security Development Process](https://www.microsoft.com/en-us/securityengineering/sdl)
* [Microsoft Threat Modeling](https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling)