Practical DevSecOps

Agile Threat Modeling in 5 Simple Steps

In this guide you will learn 5 most effective ways of Threat Modeling.

practical-devsecops.com

CONTENTS

01 Threat Modeling and Speed

02 KISS Threat Modeling in 5 Steps

03 The Key Ingredients for the 5 Steps

Step 1 Define

Step 2 Identify

Step 3 Rank

Step 4 Address

Step 5 Validate

04 Tips and Fine Tuning

CHAPTER 1

Threat Modeling and Speed

Agile Software Development frameworks and methodologies have led to the sunset of older ways of software development. DevOps has only fuelled another revolution in the way software is delivered to customers, and end users. And there are newer paradigms such as SRE, Platform Engineering, Software Factories which all have their roots in the Agile Manifesto. They are either an extension of the Agile values, or merely an expression of Agile values in a different form.

Over the years of software development, Threat Modeling has proven to be an effective way to identify design principles that help build security right into the software or a system. However just like traditional ways of software development, threat modeling can inherently be perceived as slow, and time consuming activity.

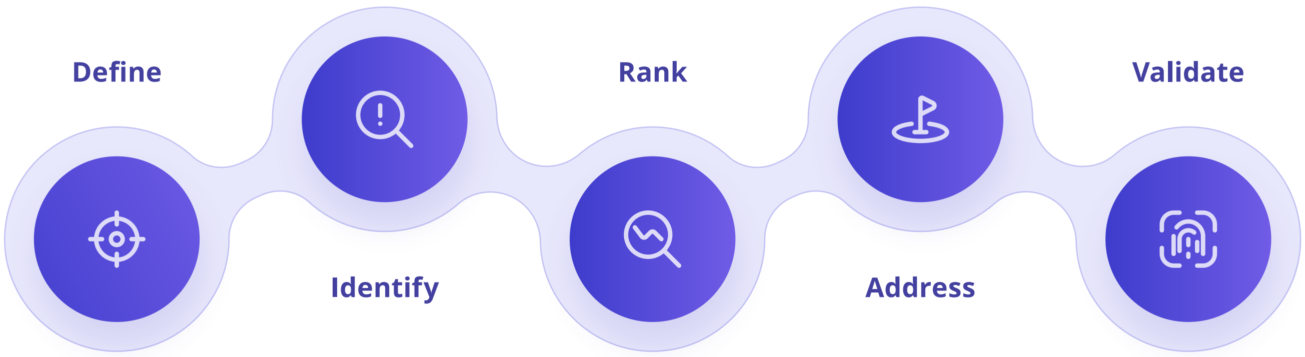
Speed and velocity are very crucial across Agile Software Development frameworks, so every activity that is a part of faster delivery cycles also needs to be faster.

Let’s explore some of the proven ways that would aid in faster ways of threat modeling leading to a secure design.

CHAPTER 2

KISS Threat Modeling in 5 Steps

To keep things simple and straightforward we are going to consider threat modeling as a five step process as listed below:



Step 1
Define the scope of threat modeling itself, the security requirements of a system.

Step 2
Identify threats that exist in a system.

Step 3
Rank the threats to prioritize which threat poses a higher risk to be addressed first.

Step 4
Address the threats and the risk it poses either by employing security controls, or through other ways.

Step 5
Validate if threats are addressed properly.

KISS Versus The 4 Question Framework

These 5 steps also fit into the four question framework of threat modeling.

Step 1 is **“What are we building?”**
Step 2 is **“What can go wrong?”**
Step 3, and step 4 are **“What can we do about the things that can go wrong?”**
Step 5 is about **“Did we do a good job?”**

CHAPTER 3

The Key Ingredients for the 5 Steps

In order to provide a bit of prescriptive advice we are going to identify the four key ingredients for every step. As an example, for the “Step 1 - Define”, we are going to look at:

1. What is needed to Define?
2. When is an appropriate time to Define in Agile Software Development?
3. How could the Define step be done in Agile Software Development?
4. When do we know we are done Defining?

01

What is needed?
Defines some prerequisites, and participants that make the step as swift as possible. They are not necessarily mandatory, however if some elements are absent, then it may significantly reduce the efficacy of the threat model itself.

02

When could it be done?
The Agile events and ceremonies where a step can be carried out. It does not necessarily mean that these are the only events where a step needs to be carried out. In the true spirit of empiricism, lessons learned based on experimentation is an indispensable aid in software development.

03

How could it be done?
Tried and tested ideas and tactics about how a step can be performed. There can be many other paths to achieve a certain goal, however the outlined steps could also help you achieve the same end goal.

04

When do we know we are done?
Every step needs to be continuous, meaning that with the cadence of Agile Software Development if your iteration is for 2 weeks or 3 weeks, then all the steps need to be continuously performed to adapt to the evolving business needs and the software itself. However, are there some ways to bring a step to a satisfactory conclusion for an iteration?

STEP 1

Define

Define security requirements and scope of the threat model

What is needed?

1. Participants: Development Team, Architects, Business Owners, Scrum Masters, Product Owners, Secur ty (or a representative from each of the roles)
2. A list of requirements in the form of themes, or epics, or user stories
3. Assets, data either in the form of a High Level Design or in the form of a list

When could it be done?

1. Product Backlog Refinement sessions where work items for the future iterations are refined to add more clarity
2. Sprint Planning sessions where work items for the current iteration are planned
3. Product Increment planning sessions where work items for the future iterations are identified and refined
4. Or any other event as the Agile Teams see fit

How could it be done?

Defining the scope of the threat model and identifying security requirements can be done by answering one or more of the following questions:

1. Are there compliance requirements when not satisfied pose a threat to the system?
2. Are there compliance requirements that guide in the process of identifying security control gaps?
3. What is the focus of the threat model? An entire system, or a theme, or an epic, or one or more user stories?
4. Does the security team have a list of hardening guidelines or security guidelines to secure the system?
5. Are there any previous security assessment reports where threats and security controls can be borrowed from?
6. Could the requirements be borrowed from OWASP ASVS Levels or CIS Benchmark Levels?

When do we know we are done?

1. When we know the scope of the threat model which can be an entire system that is yet to be developed, or a theme that is being planned, or an epic or a collection of user stories that would be developed and delivered in the future.
2. Identified security requirements should help the security team, architects, and developers be able to design and deliver the system securely.
3. Identified security requirements should help the security team, architects, and developers be able to uncover threats in the form of security design or control gaps
4. When security goals are identified (Examples: The uptime and availability of a system, the critical data that should be revered as the crown jewel, the asset that should the most secure of all)

STEP 2

Identify

Identify threats in the system

What is needed?

1. Participants: Development Team, Architects, Security (or a representative from each of the roles)
2. A diagram of an architecture, or a mental model of a proposed architecture that can be discussed to help in threat identification
3. Criticality of data and assets
4. Security properties of a system or security goals

When could it be done?

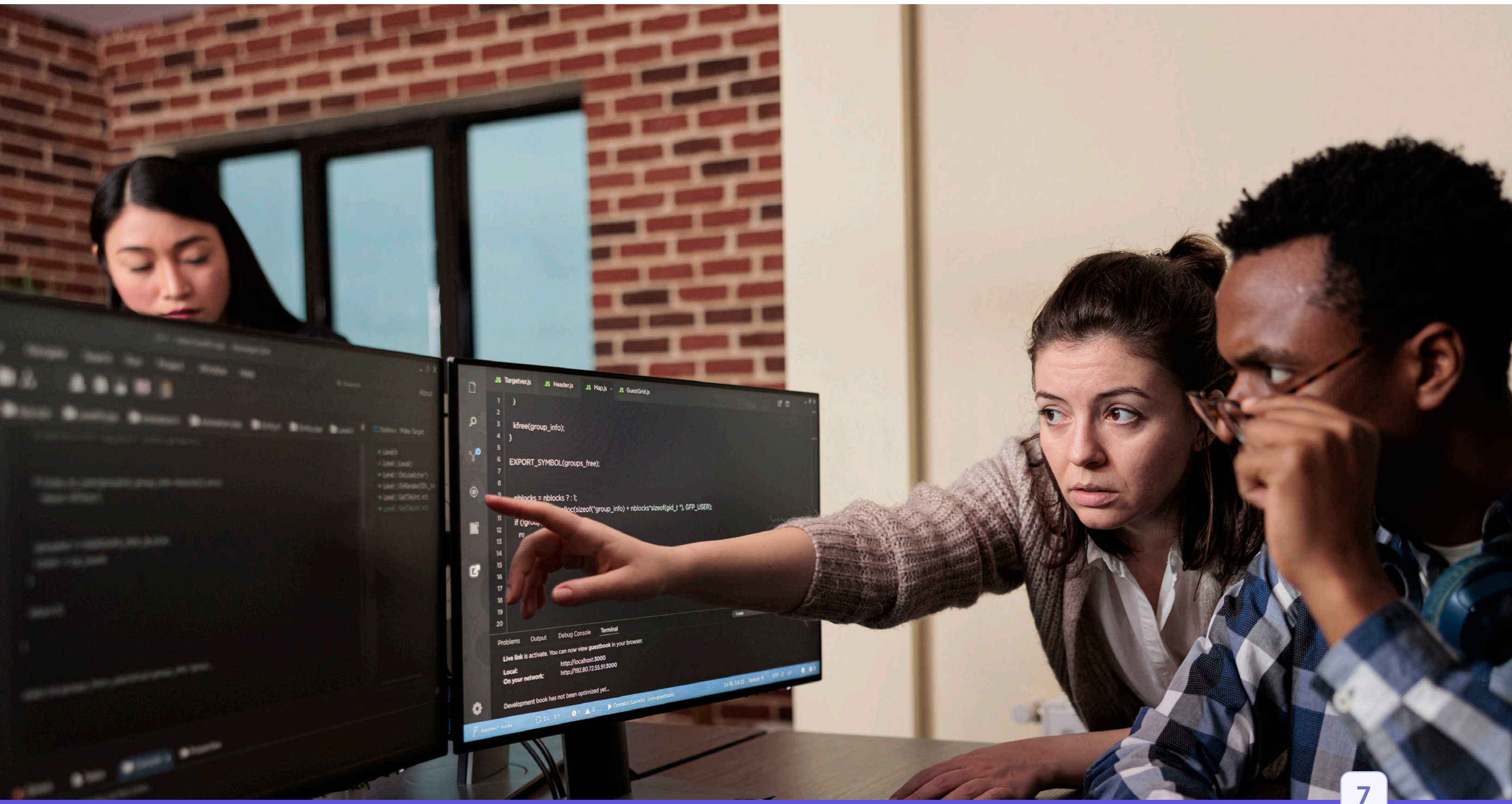
1. Product Backlog Refinement sessions where work items for the future iterations are refined to add more clarity
2. Sprint Planning sessions where work items for the current iteration are planned
3. Product Increment planning sessions where work items for the future iterations are identified and refined
4. Or any other event as the Agile Teams see fit

How could it be done?

1. Using techniques like Rapid Threat Model Prototyping
2. Using a short list of threats such as OWASP Top 10, or CWE Top 25 and questioning if the system is vulnerable to threats in the list
3. Quickly drawing cards from the Elevation of Privilege card game, or OWASP Cornucopia, or Security Cards
4. Answering a list of questions with respect to Authentication, Authorization, Input validation, Logging, Encryption, Hashing (Eg: What are the authentication requirements? What are the input validation requirements?)
5. Using Mind Maps or Attack Trees to trace an attacker’s goal
6. Identifying security gaps with the current implementation and the desired security requirements such as OWASP ASVS, or PCI DSS, or CIS Benchmarks

When do we know we are done?

1. When an agreeable number of threats are identified (the agreeable number can be 3, or 5, or 10, or 20, or 50 depending on the scope of threat model, and when the threat model activity is being done)
2. When all the cards in card game are drawn and the relevance of the threats in the cards are considered
3. When all the list of threats are enumerated to identify their relevance for the system
4. When an agreed timebox expires (the timebox can be 10 minutes to a couple of hours depending on when the threat modeling activity is performed)



STEP 3

Rank

Rank risks based on likelihood and impact ratings

What is needed?

1. Participants: Development Team, Architects, Security, Business Owners, Product Owners (or a representative from each of the roles)
2. A list of threats (can be vulnerabilities, design improvements, security requirements, security control gaps)

When could it be done?

1. Product Backlog Refinement sessions where work items for the future iterations are refined to add more clarity
2. Sprint Planning sessions where work items for the current iteration are planned
3. Product Increment planning sessions where work items for the future iterations are identified and refined
4. Or any other event as the Agile Teams see fit

How could it be done?

1. Threats need to be rated first to assign themselves a risk score using some of the following ways:
   - Risks can be scored with Rapid Risk Assessment technique from Mozille or using a Bug Bar which is technique used at Microsoft
   - Risks can also be scored with OWASP Risk Rating Methodology with online calculators
   - Risks can also be scored with a simple likelihood versus impact rating based on the impact of Confidentiality, Integrity, and Availability
   - Risks can also be scored based on the "shortest path to a high privileged asset" as prescribed in the Rapid Threat Model Prototyping technique
   - Risks can also be quickly scored on based on a class of weaknesses (Example: Many a times Injection flaws can be more catastrophic than not having an audit trail of events)
2. Risks need to be ordered according to their score. Risks are ordered based on their score, the higher the score the best it is to be addressed first

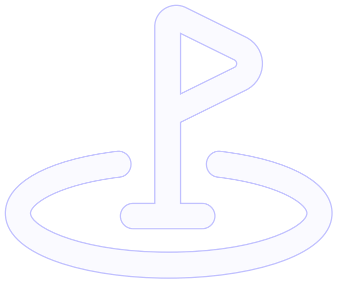
When do we know we are done?

1. When Top 10 risks are identified, when top 25 risks are identified
2. When all threats are ranked based on their risk levels
3. When the timebox expires (to help not to get lost in likelihood discussions, which many a times may be highly subjective)

STEP 4

Address

Address risks to secure the system



What is needed?

1. Participants: Development Team, Architects, Security, Business Owners, Product Owners (or a representative from each of the roles)
2. A list of ordered Risks

How could it be done?

1. Risks can be addressed by one of the risk addressing techniques:
   - Threats can be addressed by implementing security controls to reduce risk
   - Threats can be avoided by scrapping a feature or by changing the design of a system
   - Threats can be accepted if the risk is low
   - Threats can be transferred to a vendor or a third party if needed
2. The additional effort that requires design or implementation changes needs to be justified to the business stakeholders

When could it be done?

1. Product Backlog Refinement sessions where work items for the future iterations are refined to add more clarity
2. Sprint Planning sessions where work items for the current iteration are planned
3. Product Increment planning sessions where work items for the future iterations are identified and refined
4. Or any other event as the Agile Teams see fit

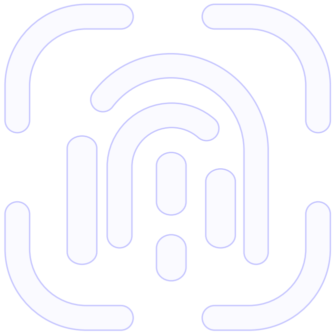
When do we know we are done?

1. When the Top 10 risks are addressed
2. When all the identified risks are addressed
3. When the timebox expires

STEP 5

Validate

Validate risks are addressed appropriately



01 What is needed?

1. Participants: Development Team, Architects, Security (or a representative from each of the roles)
2. The threat model itself which can be a combination of a model and threats in the form of risks or user or abuser stories
3. If present, tests, test cases or other forms of validation

02 When could it be done?

1. Sprint Review sessions where software increments are reviewed
2. System Demo events, Solution Demo events, or any suitable event in the Agile Release Train
3. Or any other event as the Agile Teams see fit

03 How could it be done?

1. By answering some of the below questions:
   - Are the top 10 risks addressed and if the security mitigations or controls are working as expected
   - Are the risks accepted or transferred if addressed through risk acceptance or risk transfer
2. By verifying if automated compliance checks succeed
3. By verifying if security works as expected through manual or automated verification techniques
4. By checking if the threat model is still relevant to the changes in the software increment of if the threat model needs to be adjusted

04 When do we know we are done?

1. When the top 10 risks are found to be addressed or not addressed
2. If the verification of security controls passed or failed
3. If we know whether the threat model is still relevant to the system or not

CHAPTER 4

Tips and Fine Tuning

All of the 5 steps can be completed quickly in less than 10 or 30 minutes by a highly agile team of professionals that practice the techniques every iteration.

For new teams that adopt threat modeling in their Agile Software development some of the above steps may be condensed or merged. Initially iterations planning may take more time to complete a certain step, however as the teams mature, threat modeling in 5 simple steps is going to become much more efficient.

If some of the participants are not available as recommended, then the activity of threat modeling can be performed with the available list of participants. The most important participants in an Agile set up are eventually the three underpinning Scrum roles, that is the Development Team, Scrum Master, and the Product Owner.

Many times threats and risks, requirements and mitigations or user stories can be used interchangeably if the collaboration between Development, Operations, and Security strives to achieve the security end goal.

Timeboxing is another essential tool to have focussed discussions, timebox also keeps the involved participants on their toes, to be nimble, and comprehensive at the same time.

Some of the tactics may look like shortcuts, however the end goal is a secure system. Threat Modeling does not necessarily need to be comprehensive or complete, as long as it is continuously refined through iterations as the system evolves.

"

Some of the tactics may look like shortcuts, however the end goal is a secure system. Threat Modeling does not necessarily need to be comprehensive or complete, as long as it is continuously refined through iterations as the system evolves.

Certified Threat Modeling Professional

Practical DevSecOps

Become a Certified
Threat Modeling Professional

Get Started