TOREON

Your coach in digital security

# Threat Modeling Playbook

Level up your threat modeling game

*Version 1.0, 10 September 2020*

*And donated to the OWASP threat modeling community.*

*Also available on GitHub: https://github.com/Toreon/threat-model-playbook*

Toreon BV
Grotehondstraat 44 1/1, B-2018 Antwerpen, Belgium | T +32 33 69 33 96
VAT BE0542.795.568 info@toreon.com | www.toreon.com

# Table of Contents

# 1 Introduction

We aim to improve product and software security with our new threat modeling playbook. We consider threat modeling as a foundational activity to improve your software assurance. We are convinced that a good threat modeling practice will measurably decrease security issues of delivered products.

As strong believers in open source, active OWASP collaborators and to increase our impact beyond our Toreon customers we donate this threat modeling playbook to the community.

We hope you will use this playbook to improve your threat modeling practice. We also encourage you to provide feedback to our OWASP threat modeling community in order to make this playbook even better in our next release.

I thank our collaborators (in alphabetic order): Jonas Muylaert, Joris Van den Broeck, Sebastien Deleersnyder, Steven Wierckx and Thomas Heyman to help us create this first release. I also thank Toreon for its decision to donate this work to the threat modeling community.

Sebastien Deleersnyder

CEO Toreon

OWASP volunteer

10 September 2020

## 1.1 Our vision

The goal of a threat model is to provide an objective, risk-based evaluation of security problems within your system's architecture. The information gathered, is provided to the stakeholders so they can make a well-informed decision on how to mitigate threats and lower risks to an acceptable level.

Although this is the main focus of threat modeling [1], there are plenty of other reasons to utilize threat modeling:

- To get stakeholders to agree on a shared vision of your systems security
- To increase awareness and knowledge about your system with a stakeholder
- To document your system design, document due diligence for certain legislation requirements such as privacy by design
- To serve as input for other processes such as testing the implementation, validating requirements, etc. For example: A penetration test will be much more efficient if testers can fall back on a threat model which explains what needs to be protected and lists your system's requirements
- To spread knowledge on secure architecture in your organization. Lessons learned from a threat model can be useful in future designs of other systems

Implementing and maturing the threat model process within an organization is hard. This playbook[2] will guide you through the process and will help you reach the level of maturity in threat modeling that you want to achieve within your organization.

## 1.2 Strategy

This threat modeling playbook assumes you are creating or integrating software, hardware, or combining both using an agile methodology. We will measure the maturity level of the threat modeling practice based on three topics: people, processes, and tools or technology.

If your organization is not using agile development methodologies or this process is new and not yet optimized, this guide can still be used. Chapters 2.2, 4.4 and 4.6 will need some evaluation concerning tools and processes when development methodologies are not in place. Nevertheless, Threat modeling is being done in many organizations that do not use agile development methodologies.

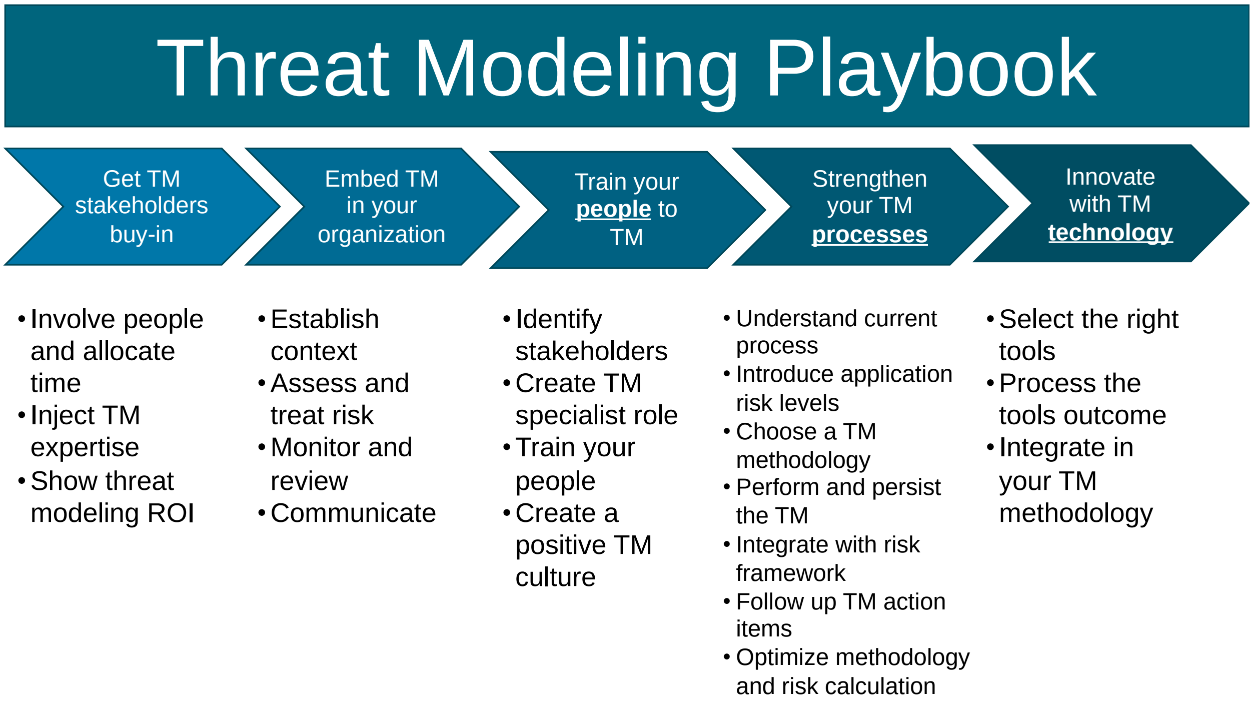
To measure your maturity concerning the threat model process, there are a couple of methodologies and frameworks available. We are heavily influenced by the OWASP SAMM[3] project and will measure maturity at three levels.

[1] See Glossary of Terms

[2] See Glossary of Terms

[3] See Glossary of Terms

## 1.3 How to use this playbook



Threat Modeling Playbook

Get TM stakeholders buy-in

- Involve people and allocate time
- Inject TM expertise
- Show threat modeling ROI

Embed TM in your organization

- Establish context
- Assess and treat risk
- Monitor and review
- Communicate

Train your **people** to TM

- Identify stakeholders
- Create TM specialist role
- Train your people
- Create a positive TM culture

Strengthen your TM **processes**

- Understand current process
- Introduce application risk levels
- Choose a TM methodology
- Perform and persist the TM
- Integrate with risk framework
- Follow up TM action items
- Optimize methodology and risk calculation

Innovate with TM **technology**

- Select the right tools
- Process the tools outcome
- Integrate in your TM methodology

There are several strategies for using this playbook. The choice between one of these strategies depends on the number of systems and the level of “thoroughness” needed for those threat models.

There are several strategies you can use:

1. Start with one system and get to a very mature state of your threat modelling practice with a very mature threat model. Afterwards, proceed to the next system.
2. Start with many systems in parallel. While you are maturing your threat model practice, come back to each threat model and improve it.
3. A combination of the above. Start with the top 10% of your (most important) applications in parallel if your organization has a large number of systems.

For each existing system you also need to decide a strategy:

1. Do a complete threat model of the system as it exists today. The rationale: this system is important, and we want to reduce the risks associated as much as possible.
2. Perform threat modelling on the changes being done on the system. The rationale: this system has been used for a long time, if any big security problems existed, they would have been exploited by now so we can limit ourselves to new functionalities.
3. Perform a threat model on the most important use case(s) in the system. The rationale: the time needed for a full threat model is not economically viable, but we still want to cover the most important risks. These risks will generally be situated in the most important use cases since that is where the system makes money.

This threat model playbook contains several elements from which you can design your ‘playbook’ as you would in American football. Each subchapter can be considered a ‘play’ that you can use in a ‘playbook’ that is customized to your organization/situation. You should play for “maximum forward progress” towards your end goal.

# 2 Get stakeholder buy-in

A threat model can only achieve its goal, i.e., to reduce risk and increase security, if there is sufficient buy-in of all the stakeholders involved. Specifically, in order to implement the playbook, you will need the following resources:

1. Time of the people involved in creating the threat model.
2. Threat modeling expertise (especially if you are just starting out).
3. Time, resources, and authority to address the resulting threats.

As these are important to get management buy-in and commitment to manage your risks with threat modeling. Let us look at these in turn.

## 2.1 Involve people and allocate time

It is easy to underestimate the number of people that are directly or indirectly involved in creating a threat model, but you will need to address their concerns in order to get them to collaborate. In order to do that, you first need to understand what it will cost them, and what they can potentially see as obstacles to investing time in the threat model. Note that we do not mention the threat modeling champion in this listing, as their motivation to do threat modeling is assumed.

| Business stakeholder | Cost, obstacles for the stakeholder |
|---|---|
| **Management** | • Need to allocate time of valuable team members for the threat modeling exercise, which might delay other activities.<br>• Wants to see return on investment but might not see the added value of a threat model (especially if threat modeling is new for your organization).<br>• Might be hesitant to have all potential threats made explicit, i.e., "if I know about it, I will have to do something about it". |
| **Application owner** | • Might already have indications of existing threats and might not want for those threats to become explicitly documented.<br>• Faced with a strict roadmap, hesitant to have someone add a potentially long list of mitigations to that roadmap.<br>• Will need to invest some time to assist in the threat modeling workshops. |
| **Architect** | • Might feel that the threat model is like an exam: An external team is reviewing the architecture and will assign a grade,<br>• Will need to invest some time to assist in the threat modeling workshops. |
| **Developer** | • Might feel that the threat model is like a code review and that he or she will be graded on their secure coding skills.<br>• Might be hesitant that the threat model will highlight that the developer does not have some specific security skills.<br>• Will need to invest some time to assist in the threat modeling workshops. |
| **Security and/or DevOps engineer** | • Might be hesitant that the threat model is like a review and will highlight gaps in the current security.<br>• Will need to invest some time to assist in the threat modeling workshops. |
| **Project manager** | • Already faced with several (probably very strict) deadlines, hesitant to add more work to the project roadmap. |

| | |
|---|---|
| | • Hesitant that threat modeling exercise and results will derail the project roadmap completely. |

In order to defuse some of these arguments and convince the stakeholders that threat modeling is also in their best interest, it is necessary to first listen to their concerns and acknowledge them.

| Business stakeholder | Potential gains |
|---|---|
| **Management** | • Demonstrate that they are taking a proactive stance on security.<br>• Useful as an argument for e.g. GDPR compliance and privacy and security by design.<br>• Useful part of an information security management system (ISMS), e.g., for ISO 27001<br>• Having an explicit list of risks enables risk-based security management: Management can show that they are investing their security budget to address the highest risks first. |
| **Application owner** | • Having an explicit list of risks and suggested mitigations enables risk-based security management: The application owner can assign priorities based on evidence.<br>• The threat model can serve as a tool to request additional security budget. |
| **Architect** | • A threat model is not a review, but should lead to constructive advice on improving the architecture.<br>• Might lead to reusable security patterns that can be instantiated in other parts of the architecture, or for other application domains. |
| **Developer** | • A threat model is not a review, but should serve as constructive security advice.<br>• Might be used as a driver to request additional security budget (for, e.g., a security specific training). |
| **Security and/or DevOps engineer** | • A threat model is not a review, but should serve as constructive security advice.<br>• Might be used as a driver to request additional security budget (for, e.g., a security specific training). |
| **Project manager** | • A threat model is an ideal exercise to get all stakeholders on the same page, and ensure that there is a coherent view on security.<br>• A threat model can serve as a tool to request additional security budget. |

## 2.2 Inject threat modeling expertise

A second ingredient you need to acquire in order to make threat modeling a success, is the relevant expertise. In Chapter 2, we mentioned the importance of finding a threat modeling specialist and to train your people. In order to obtain the relevant expertise, there are three approaches you can take. Let’s look at these in turn.

### The do-it-yourself approach

In organizations that are just starting with dipping their toes into the proverbial threat modeling pond, one option is to start acquiring threat modeling expertise by reading books and accessing some freely available online resources. This is especially the case where there are no extensive security budgets, or if the current need is low.

The advantages of this approach are that it can start right away and does not take a lot of preparation or budgeting. The downside, however, is that threat modeling can be tricky when you are just starting out, especially for people without prior security expertise. In that case, a failed first ad-hoc threat modeling attempt might undermine the goodwill of the rest of the stakeholders to invest further in setting up a threat modeling approach. This approach also does not scale for larger organizations.

Therefore, the recommendation is to only start with and ad-hoc approach if the people involved have some prior security exposure, and there is a willingness to experiment (and possibly fail). For other cases, it might be better to start off by hiring an external expert.

**Hiring an expert**

A good, fairly lightweight way to start adopting a threat modeling approach, is to have a threat model done by an expert, in close collaboration with your team. In that way, the team gets to see hands-on how threat modeling is performed, and there is a larger guarantee that the first threat model will be a success.

The advantages of this approach are that it is significantly lower risk than the ad-hoc approach and can create a lot of goodwill and willingness to adopt a broader threat modeling approach in a fairly short time. Furthermore, it scales reasonably well, as the same expert can be hired to perform follow-up threat models for other teams. The downsides of this approach are that it does require a larger budget than the ad-hoc approach, and it does not automatically scale to large organizations with dozens or hundreds of applications in their portfolio.

Therefore, hiring an expert should be considered by organizations that are just starting out with threat modeling and want to get some experience, or small organizations with only a few applications in their portfolio. For larger organizations that want to systematically adopt threat modeling throughout various teams, external threat modeling training programs are more suitable.

**Threat modeling training**

With a threat modeling training program, a trainer is hired to train an initial core team of people to threat model. The trainees should be highly motivated people that can subsequently take up the role of threat modeling specialists (or even evangelists) within that organization.

The advantages of this approach are that it scales extremely well and has the highest chance for success that your organization is able to fully adopt and internalize threat modeling. The downsides, however, is that it takes a while before results are produced, as the training should still be followed by actually creating initial threat models. It also requires a larger up-front investment, which might be a hurdle for organizations in which threat modeling has not proven its value yet.

Therefore, finding threat modeling training should be considered by organizations that are already mostly convinced that threat modeling is the way forward, but are looking to scale up their threat modeling approach and internalize it fully.

## 2.3 Threat modeling return on investment

Lastly, an extremely important aspect of setting up a sustainable threat modeling initiative within an organization, is to demonstrate return on investment (ROI) and get management commitment on this. A threat modeling approach is only able to show ROI if it results in a demonstrable improvement of the security of that organization, or its products.

Some things that need to be considered, are:

- To lead to security improvements, a threat modeling approach should result in recommendations that are clearly documented to the relevant stakeholders. A document on a file share which no one knows exists, will never lead to ROI.
- To lead to change, the recommendations of a threat model should be realistic and implementable. If the recommendations exceed the timing, budget, expertise, or other constraints of a project, they will not effect change and not lead to ROI. It is crucial that the threat modeler has a good understanding of the project constraints before making recommendations!
- To show that the change is effective with threat modeling, mitigations and recommendations made in a threat model should be clearly linked to development artefacts such as new user stories, bug fixes, JIRA tickets, and so on. If you can measure the number of projects, user stories, or tickets that are impacted by threat modeling, you can more easily track the level of adoption of threat modeling throughout your organization, and its effect. That, in turn, makes it easier to demonstrate ROI.

Finally, to show that the change is positive, the changes effected by threat modeling should be linked to the number of security bugs or incidents that are reported after go-live of the system that was threat modeled. An important consideration is to distinguish between the number of security issues that are reported as a direct result of threat modeling (this number should be high!) and the number of security issues that are reported after the fact (this number should go down). By making it clear that threat modeling is responsible for detecting and fixing more security issues proactively, you can clearly show ROI.

# 3 Embed in your organization

Threat modeling is a methodology to identify risks and hence should be integrated in your organizations' risk management process. As a best practice we look at the risk management process described in ISO27005:2018 and map our threat modeling activities on this process.

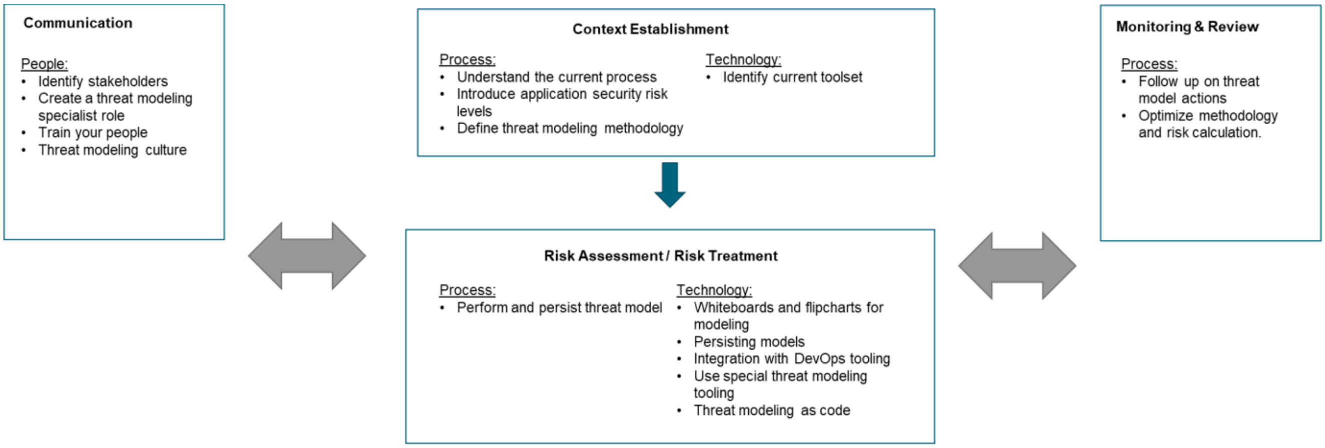
We visualized a simplified overview of the main stages, who are part of the risk management process in Figure 1. We can summarize the threat modeling activities in three categories:

- People: who to involve how
- Process: which processes need to exist or need to be adapted
- Technology: which tools and technologies can help and facilitate threat modeling

The risk management stages we consider for threat modeling are:

- Context establishment
- Communication
- Risk assessment and treatment
- Monitoring and review

In each of these stages, we map related threat modeling activities. These threat modeling activities are grouped by people, process, or technology categories.



**Communication**

People:
- Identify stakeholders
- Create a threat modeling specialist role
- Train your people
- Threat modeling culture

**Context Establishment**

Process:
- Understand the current process
- Introduce application security risk levels
- Define threat modeling methodology

Technology:
- Identify current toolset

**Risk Assessment / Risk Treatment**

Process:
- Perform and persist threat model

Technology:
- Whiteboards and flipcharts for modeling
- Persisting models
- Integration with DevOps tooling
- Use special threat modeling tooling
- Threat modeling as code

**Monitoring & Review**

Process:
- Follow up on threat model actions
- Optimize methodology and risk calculation.

*Figure 1 Threat Modeling in your organization – integration with the risk management process*

## 3.1 Context establishment

First, you need to understand how your organization handles and manages risk. The same risk can have a totally different impact in different organizations. For threat modeling the following activities are important concerning context establishment:

**Process:**

- 5.1 Understand the current process**:** it is crucial to understand existing processes in your organization and how to integrate threat modeling in them.

- 5.2 Introduce application security risk levels: by using application security risk levels and deciding when to apply threat modeling you can focus on the most important applications first.

- 5.3 Define threat modeling methodology: there are many ways to define a threat model. You should select the methodology that fits your organization best.

**Technology:**

- 6.1 Identify current toolset: identify tools and technologies used in your organization. This will help to assess how to integrate threat modeling in the existing toolset.

## 3.2 Risk assessment and treatment

Secondly, you execute the threat modeling activity as part of the risk assessment stage. Here you follow the selected threat modeling methodology.

**Process**:

- 5.4 Perform and persist threat model: you create and store your threat model.

**Technology:**

- 6.1 Whiteboards and flipcharts for modeling: most threat modeling methodologies are easy to start on a whiteboard or flipchart.

- 6.2 Persisting models: tools and technology to store threat models.

- 6.3 Integration with DevOps tooling: when working in a development environment, integrating with the development tooling is highly recommended.

- 6.3 Use special threat modeling tooling: threat modeling tools exists that can support you to threat model.

- 6.3 Threat modeling as code: following infrastructure as code – threat modeling as code also exists and can have several benefits.

The identified risks should be handled according to the risk management policy/process in use in your organization. The first step is to consider different risk treatment options such as: risk reduction, risk retention, risk avoidance or risk transfer. Based on a cost / benefit calculation, you select your best options.

## 3.3 Monitoring & review

Thirdly, risks are not static and will change over time. Exposure of the vulnerability leading to the risk may change, sensitivity of the information in the application may change, a risk may not be remediated in time, and so on. Hence it is important that your risks and their factors are regularly monitored and reviewed. For threat modeling this consists of the following activities:

**Process:**

- 5.6 Follow up on threat model actions: action should be taken on findings that come out of a threat model.

- 5.7 Optimize methodology and risk calculation: to facilitate continuous improvement, you should monitor and optimize your threat modeling methodology.

## 3.4 Communication

Finally, communication is key when creating a threat model. It is not possible to create a proper threat model without collaboration.

**People:**

- 4.1 Identify stakeholders: different stakeholders you involve in creating a threat model.

- 4.2 Create a threat modeling specialist role: a threat model specialist role will facilitate threat modeling in your organization.

- 4.3 Train your people: security awareness is critically important. Threat modeling training is a must when you start with threat modeling.

- 4.4 Threat modeling culture: it is important to create a supporting culture for threat modeling.

# 4 Train your people to threat model

Before starting with the definition of an actual threat modeling process, care must be taken that the relevant people are identified, trained, and adopt the right mindset. In Chapter 4.1, we talk about identifying the relevant stakeholders for threat modeling. In Chapter 4.2, we focus on the creation of a threat model specialist role, who will serve as the focal point for threat model activities. In Chapter 4.3, we highlight the importance of appropriate training to support threat modeling. In Chapter 4.4, we end by talking on how to create and nurture a positive threat modeling culture in which threat modeling can flourish.

## 4.1 Identify stakeholders

One of the strengths of threat modeling is that it brings together various stakeholders involved in the security of an IT system or project and ensures that they are aligned. Threat modeling helps this group of people to share a common understanding of the business value of the system or project. At the same time, it also helps those people share a view on the main threats and what mitigations can be put in place to address them.

But who are those stakeholders? Involve too few, and the threat modeling exercise loses its main benefit as it does not create a shared understanding of business value and threats. Involve too many, and the exercise runs the risk of devolving into a costly set of meetings. You know the situation: most of the participants are too busy checking their e-mail and, in the end, nothing gets decided. In our experience, threat modeling is best performed within a core team of limited size. Ideally, at least the following roles are represented:

| Role | Motivation |
|---|---|
| **Business stakeholder** | Ensure that business value and potential business impact is clear. |
| **Architect** | Provide a high-level overview of the application ecosystem and the underlying rationale. |
| **Developer** | Provide details on used libraries, frameworks, and coding guidelines. |
| **Security and/or DevOps engineer** | Provide details on existing security and/or infrastructure configuration. |
| **Project manager** | Validate proposed mitigations in terms of timing and budget. |
| **Threat model specialist** | Ensure proper execution of the threat model process. |

*Reference roles in threat modeling.*

This team composition can be very powerful, as it contains stakeholders with complementary views. First, it contains people with in-depth technical know-how, such as developers, security, and DevOps engineers. Second, it contains people that have a broader view, either technical such as architects, or non-technical such as business stakeholders and project managers. This is a proven recipe for productive discussions, as one might imagine. If managed well, this protects the threat modeling exercise from threats such as not representing reality, underestimating certain technical risks, or missing the point by failing to appropriately frame the uncovered risks with respect to their business impact.

Of course, there is a drawback too. These people are usually extremely busy, and their time is a valuable resource. It is therefore not ideal to simply invite everyone for all threat modeling meetings. You must prepare the threat modeling workshops for them to be efficient and optimize the time spend by the participants, which we will revisit in Chapter 5 when we talk about processes. An important role who will achieve this is the threat model specialist. The specialist role will be explained next.

## 4.2 Create a threat modeling specialist role

The primary purpose of a threat model specialist is to help incorporate threat model practices and a strong security culture into all aspects of an organization's development processes. Threat model specialists are typically permanent staff that act more like floating specialists supporting the squads[4] as needed. They provide threat modeling advice, support squads, and occasionally drop in for a sprint or two.

An example "job description" for a threat modeling specialist could look like this:

**Responsibilities:**

- Act as a threat model point of contact for the squads and their security champions.
- Responsible for leading threat model-related activities within the squad.
- Act as a liaison between the stakeholders and squad members.

**Tasks:**

- Raise the overall security awareness and threat modeling knowledge within the squads.
- Organizes and facilitates threat modeling workshops for the squads.
- Assures that lessons learned of threat modeling is communicated towards the squads.
- Develops and improves your organization threat modeling methodology.
- Selects, introduces, and maintains threat modeling tooling to support and automate your organization threat modeling practice.
- Lead efforts in identification and remediation of weaknesses and vulnerabilities in the product design and development processes of the squads.
- Develop security-focused user stories for squads using agile development strategies and designing unit and integration tests together with the squad's test engineer.
- Organize threat modeling education and training, advocate for security-focused culture changes, and recruit, mentor, and train additional threat model specialists and squad champions.

**Required skills and experience:**

- At least 2 years of experience in threat modeling.
- Expert knowledge of threat model techniques and tools.

[4] See References

- Excellent communication and meeting moderation skills.
- Proven to be a team player.
- Have an interest in security and willingness to learn and grow to meet the security needs of the squads.
- Knowledge of security concepts, tools, and practices in development (automated security testing, dependency checking) are a plus.

Once you have this role figured out for your organization, you second step is to find or hire candidate threat modeling specialists within your organization to help you start and improve your threat modeling practices.

## 4.3 Train your people

As mentioned earlier, it is important to train the involved people on how to do practical threat modeling. Before anything else, the people in your threat modeling sessions need to understand the why and how of threat modeling.

Start to gradually involve people from your different DevOps teams in lunch & learn sessions that explain and demonstrate what threat modeling is all about.

Once people have a basic understanding, start with a role-based training program. People in the roles of architects, security champions, testing engineers and security specialists should follow thorough threat modeling training that covers at least the following topics:

- Threat modeling as part of a secure development lifecycle
- The threat modeling stages and process
- Threat modeling methodologies (covering at least STRIDE[5])
- Diagramming
- Threat identification
- Threat mitigation
- Risk management concepts
- Hands-on exercises, preferably based on your organization systems

Ideally you include organization specific playbooks and templates, examples, and lessons learned. Also make sure to adapt the training to your technology stack and project governance. Assure that the learning objectives are clear and met, that you focus on outcome, techniques and that you perform hands-on exercises in group to mimic real live threat modeling workshops.

Other people will be involved in threat modeling tangentially or are stakeholders or recipients of your threat modeling actions. Examples are developers, DevOps engineers, business analysts or product owners. For these roles you provide a high-level introduction training on threat modeling.

[5] See Glossary of Terms

## 4.4 Create a positive threat modeling culture

Create a positive culture around threat modeling to get the most out of threat modeling. Threat modeling is not an audit, but an activity to align the involved team on a shared vision on the security of the products that they are working on.

When you start threat modeling as a practice, you should make it clear that a no-blame culture is the right starting point to do threat modeling. People make mistakes. During the workshops you will discover issues and design flaws. Take these as learning points to do it better next time, do not point fingers or blame the involved people for making errors. Look for root causes and learn from past mistakes to improve your product and DevOps processes.

Likewise, when you do threat modeling, leave your ego at the door. Make sure you focus on the bigger picture of the threat model and not just your own vision and understanding.  When doing threat modeling workshops, apply active listening and be open for the position of everyone involved. This means that you actively involve all the participants in your threat modeling session. By doing so, participants will get the full picture of your threat model which will result in support from your team. It also requires mutual respect, inclusivity, and the establishment of a space that is safe enough for the participants to speak their minds.

As threat modeling is an activity where people with different backgrounds and skill levels will participate, you spend time to gain a good common understanding of the system in scope, including business and technical aspects. Make sure that everyone understands the terminology and concepts, if needed provide some reading materials before the workshops.

Creating a threat model is fun. But do not forget the objective of your activity: aligning your stakeholders on the security vision you created, together with the people that were involved in the workshops. Make sure not to simply “toss your threat model over the cubicle wall” but take your time to go and explain it to the affected people or groups, ideally in-person. Translate your threat model outcome to the target audience. E.g. for senior management be prepared to map your technical findings to business risks and the involved budget impacts. Another example is to provide input for security testing towards your testing engineers in a way that they understand the technical implications of the discovered flaws in your threat model.

Threat modeling is all about breaking your “tunnel vision” on the security aspects of your system. Tunnel vision originates from looking too far in the future, only looking at what is familiar in terms of security, and not being aware of external threats to your system. By involving other people and creating an open culture around threat modeling you will be able to break free from this tunnel vision trap and discover more and other security issues than you ever imagined.

# 5 Strengthen your threat model processes

Setting up the right processes is vital for increasing the maturity level of your threat model. This chapter of the playbook will guide you in setting up or updating these processes. When determining your process of threat modelling, you must first understand your current processes in place, as mentioned in chapter 5.1. You will need to determine the application risk level to see what threat model activities need to be performed and why, chapter 5.2 will guide you in doing this. Chapter 5.3 will assist you in choosing the correct methodology that must be used when developing a threat model. In chapter 5.4 we will discuss ways to persist your threat model according to their deliverables. The integration of your threat model with your risk management framework will be discussed in chapter 5.5. Chapter 5.6 will give you an insight on how to follow up on mitigations that were produced by your threat model. The last chapter 5.7, elaborates on different processes that already exist and how your threat model process can interact with these processes.

## 5.1 Understand your current process

OWASP SAMM[6] describes threat modeling in the SAMM threat assessment practice. Basic threat modeling should include best-effort, risk-based threat modeling using brainstorming and existing diagrams with simple threat checklists.

To determine where to introduce threat modeling in your organization, it is helpful to understand and document your current threat modeling processes. Make sure you have a basic understanding of what your teams are doing (if they do threat modeling at all). When do you carry out threat modeling? What kind of inputs and outputs do you have? What steps are taken?

We recommend drawing a basic overview of your current threat modeling steps. Use this drawing as a map together with this playbook to see where you can improve your existing threat modeling based on the activities described in this playbook.

## 5.2 Introduce application risk levels

As threat modeling can be a time-consuming activity you need to decide for what applications you are going to do this. Typically, applications will not all have the same risk level within your organization. We recommend ordering your applications in different risk “buckets” or risk profiles. Usually, you use three risk profiles: high, medium, and low risk. When you have application risk levels you can leverage other OWASP resources that rely on these levels, such as OWASP SAMM or OWASP ASVS[7].

But first you will need to define the application risk profile levels and a classification method that work for your organization. The level of risk for a system will determine the level of threat model activities that need to be performed to protect them. For more information on the risk related processes you can consult chapter 3.

Use a simple method to evaluate the application risk per application, estimating the potential business impact that it poses for your organization in case of an attack. This is also explained in the Application

[6] See Glossary of Terms

[7] See Glossary of Terms

Risk Profile activity of OWASP SAMM. An example and simple risk classification into three levels can use the following scheme:

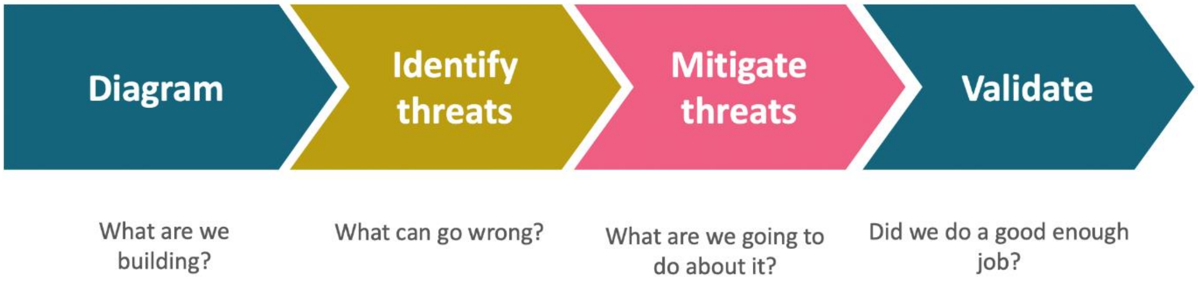
- Level 1 (low risk) is for systems that contain no critical data and complete loss, data theft etc. will have no impact on your organization. Examples: application to reserve a parking spot, application to order lunch … These applications do not need a full-blown threat model.
- Level 2 (medium risk) contains all systems not in level 1 or level 3. In practice these are systems that are not critical but do contain sensitive data such as GDPR impacted data etc. These systems need a threat model that handles the use cases of this system.
- Level 3 (high risk) systems are those systems that contain very sensitive data, large amounts of sensitive data or that control processes that, if compromised, threaten the existence of your organization. Examples: applications that process intellectual property, systems that result in actions in the physical world and could harm people (safety aspect). These systems need a threat model that goes into great detail and might go into very detailed systems aspects, up to inter-process communication on a CPU if that is part of your system scope.

Data is considered sensitive if compromising the data would negatively impact your organization.

## 5.3 Choose a threat modeling methodology

There is only one rule when it comes to choosing a threat model methodology. If it works for you, keep it. If not, change it. This is a very pragmatic approach but does not allow us to compare and evaluate different threat model methodologies.

A threat model methodology should at least answer the following questions/steps:



These steps are:

**Diagram the application.**

A detailed understanding of the mechanics of your application makes it easier for you to uncover more relevant and more detailed threats. This includes the identification of security objectives. Clear objectives help you to focus on the threat modeling activity and determine how much effort to spend on subsequent steps. Itemizing your system's important characteristics and actors helps you to identify relevant threats during the next step.

**Identify threats.**

Use details from the previous step together with techniques to identify threats relevant to your system scenario and context. Examples of such techniques are attack trees[8], STRIDE[9], LINDDUN[10], etc.

**Mitigate Identify vulnerabilities.**

Review the layers of your system to identify weaknesses related to your threats. Use vulnerability categories to help you focus on those areas where mistakes are most often made.

**Validate.**

Validate the whole threat model. Is each threat mitigated? If not: is the residual risk clearly explained and tied into business risk. Scoping of security tests is typically done during this step.

Evaluating a threat model methodology can be split in 2 distinct steps:

1. Evaluate the soundness of the methodology
2. Evaluate if the methodology works for you

Academic research (Yskout, et al., 2020) shows us the elements, that should be present in an effective threat model methodology:

- Model based
- Traceable
- Systematic
- Business integration
- Context aware
- Scalable

A recent overview of different threat model methodologies can be found in this master's thesis from Selin Juuso (2019). Once you eliminated all threat model methodologies that do not have these elements you will have a list of potential candidates for your organization. You should also already have an idea of the requirements for the methodology. Are you looking for a pragmatic approach or rather a formal approach? Who will be the main stakeholders and users of the methodology? Do you have generic security and compliance requirements and can the methodology handle those? Do you have an established risk calculation methodology, and can the methodology apply this?

Evaluating if a threat model methodology is going to work in your organization often requires you to have at least one try-out of the methodology. It is strongly advised to hire help of a specialist in the methodology when doing this.

In an ideal world the threat model methodology gives you sufficient tools and techniques to answer the 4 questions no matter what type of system/application of your organization is being assessed. In reality there might be such a diverse ecosystem of assets to threat model, that techniques of other

[8] See Glossary of Terms
[9] See Glossary of Terms
[10] See Glossary of Terms

methodologies might need to be applied. This should be a minority of the cases and we should keep the first and only rule in mind: if it works for you, keep it. If not, change it.

## 5.4 Perform and persist the threat model

Once a threat model is created, you should persist or store your threat model for later reference, or if you want to update it. There are two types of deliverables that are created after each threat model - and for each of them different options are available to persist them.

- Threat modeling supporting files: all artifacts created to perform the threat model e.g. data flow diagrams, architectural drawings, questionnaires, documentation, meeting minutes or STRIDE analysis.
    - These can be stored in the platform your squad is using, e.g. SharePoint[11], MS Teams[12], G Suite[13], Azure DevOps[14], JIRA[15], …

- Risks identified in the threat model.
    - Stored in the risk register, stored in a bug/user-story system, ...
    - For each identified risk you should include a risk level and the agreed upon follow-up action.

  Whatever tool you use, we recommend storing your threat modeling artifacts as close as possible to your team documentation or source code repository. We also recommend using tools that the involved people can easily use.

## 5.5 Integrate with your risk management framework

To ensure risks identified in the threat model are properly handled, a defined and management supported risk management framework is recommended. Without an agreed manner on how to handle risks – anything identified in the threat model will float around in empty spaces without actions or owners. You can have your own risk management framework, or you can derive one from the aforementioned ISO 27005 standard.

The following components are essential to ensure action is taken upon risk identified during threat modeling:

- Risk level: It should be possible to clearly identify the risk level for each risk to your organization. The risk level can depend on your business, type of application, data involved, or etc. The description of the risk level should include all the necessary components. When included, you can determine the specific risk level for a certain risk, that is representative for the risks posed to your organization.
- Implications of the risk level: an agreement on which actions to take based upon the risk level. Based on the risk threshold of your organization, an agreement is necessary on what to do with each level

[11] See Glossary of Terms
[12] See Glossary of Terms
[13] See Glossary of Terms
[14] See Glossary of Terms
[15] See Glossary of Terms

of risk. E.g. will a critical risk be resolved in one week? What happens when the risk is not resolved within one week?

- Risk escalation and acceptance: when a risk is not taken up and the risk level is sufficiently high, a risk escalation procedure must be followed. In addition, risk acceptance must be possible.
- Risk review process: it is necessary to regularly review identified risk and whether appropriate action is taken. This can be done by reviewing a risk register or by reviewing user-stories or bugs in tools like JIRA or Azure DevOps.

Make sure you identify these components in the risk management process in use at your organization and integrate your threat modeling methodology into these. If these are not present, your first step is to bootstrap a basic risk management process and agree on the essential components listed above with your key stakeholders.

## 5.6 Follow-up threat modeling action items

One of the most important outcomes of a threat model is the list of mitigations and the order in which they need to be implemented. Many organizations struggle with this process. A perfect threat model is almost useless if the mitigations are not followed up upon.

Start with creating a follow-up process. This process should ensure that every agreed upon action is implemented or executed by the assigned due date or handle any deviation from that plan. This process will certainly need to capture the following elements:

- Who is accountable for the progress and due date?
- What is the current status of the mitigation?
- What is the risk of the mitigation?
- Who is responsible for the execution / implementation? What are the actions that are needed?
- What is the current state of each of the actions needed to finish this mitigation?

Keep in mind that the risk level of threat modeling findings will change over time and might require new due dates and re-ordering of mitigations.

## 5.7 Optimize methodology and risk calculation

Threat modeling might be new to your organization, but elements of the threat model process are probably already being produced. Examples are architecture diagrams, user stories, risk handling methodologies and risk calculation methodologies, security baselines like the CIS guide etc.

Where possible you will want to re-use these since the process to gather them already exists and the effort to create the information is already done. It might be needed to adapt or enhance certain pieces of information to make them more applicable or more efficient for use in threat modeling.

Several other activities can provide input for threat modeling. For example, the penetration test reports might hint to a structural problem in the architecture, risk assessment done in the beginning of the project (e.g. the business case) might provide valuable information on the expected return for the system and could point to associated threats. There might be the need for compliance to certain legislation coming from the DPO office or your legal team. They might also already have defined the requirements for the audit trails the system needs to provide. Your service delivery team might provide you with expectation on the quality of service that they will want to put in SLA's together with the marketing/sales team etc.

When you start the threat model process you need to have a good overview on all other processes in your organization in order to make the most efficient use of the limited resources available. In the same sense the threat model can be used as input for other activities such as test automation, penetration testing, training, awareness etc.

And last but not least you might want to align or standardize on an organization wide risk calculation framework. This can be based on CVSS[16], the OWASP risk rating methodology[17] or any other methodology that makes sense for you teams. It does not really matter which calculation methodology you use, as long as you agree on a consistent way of implementing it within your organization. This way you can compare and manage the outcomes of your threat models across different teams.

[16] See Glossary of Terms

[17] See Glossary of Terms

# 6 Innovate with threat model technology

At its core, threat modeling is a process that brings the right people together to think about (and hopefully improve) the security of an application or IT system. Technology should always be judged in its ability to support that process, and never as a goal in itself.

In this section, we will talk about a number of guidelines that provide guidance on selecting the right technology and integrating it in your way of working so that the technology maximally supports the threat modeling process, and not the other way around.

## 6.1 Select the right tools

Every threat modeler will make use of some technology. This can be very simple such as a piece of paper or more advanced such as specialist threat model tools. When deciding on a tool there is one core rule that should always be followed: a tool should support your process, never change your process to accommodate a tool. As a result of this rule you should first gather the requirements for your tool(s).

Start with the tools already used within your organization for other purposes. Perhaps they can either partially or completely fulfill the requirements already.

For example, the architects might already use a diagramming tool to create their diagrams. Each of the questions of the threat model process might have one or more tools being used. Typically, the models will be created in a drawing program such as MS Visio or Diagram.net (formerly Draw.io). The threat model itself is documented in a word processor whereas a spreadsheet is often used to calculate the risk scoring.

Identifying the toolset currently in use will make it easier to incorporate the threat model practice in the existing toolset. This will create less friction and should help the adoption of threat model practices in your organization. Every organization is different, and each one implements the threat model practice and processes in a different way to fit the needs of that organization. This means there is no definitive list of tools that are 'fixed' for a certain level of maturity of the threat model process.

Since the tools are supposed to fulfill the requirements, it is important that you document the use of tools when you use them during the threat model process.

The tooling should be tailored to your organization. It should fit the size of the number of threat models needed. It should support the complexity of those threat models as well as the maturity of your threat model process. Of course, the tools should fit within the budget you have available.

As an example, we will discuss one of the cheapest and readily available tools, the flipchart. There are several good reasons to use flipcharts, whiteboards or magic paper in an organization that is new to threat modeling. Creating models is cooperative work of different stakeholders. Standing around a whiteboard will enable all your stakeholders to take a marker and make changes to the model(s). It will force active participation of your stakeholders and will facilitate a consensus on the system scope. The same reasoning is valid for the other times during a threat model that the stakeholders work together. A threat model is a consensus between these stakeholders and as such there is a need for conversation to discuss the doomsday scenarios, scope, models, threats, mitigations, risks, etc. Flipcharts are a cheap tool that can be used to create initial threat models in any organization at the start of its threat model journey. It will stay a very relevant tool as your organization grows in maturity, but there will probably be a need for a tool that starts from a digital version of the model for updating the threat model. A flipchart is not the most ideal tool for that so it will be replaced in organizations that have created an initial threat model and are adding significant functionality that requires an update of the existing model.

Creating a threat model using remote participation is certainly possible but this requires specific tools and some experience within the team to pull off in an efficient way.

## 6.2 Processing the outcome of tools

The primary goal of a threat model is allowing the decision makers to use an objective, risk-based approach to mitigate threats against a system.

However, there are many other reasons to do threat modeling. These reasons could be:

- To create awareness with your stakeholders
- To document due diligence
- To serve as documentation on the system
- To feed input into other practices, such as security reviews and penetration testing
- Feedback lessons learned into other systems and threat models
- To share threat modeling knowledge
- …

To cover all these requirements your threat model needs to be accessible by different groups of people, some of them needing only access to certain parts of the threat model.

Once the first threat model creation has started you will need to persist the results of these sessions. Any tool(s) you select should therefore fulfill two requirements:

1. Allow / facilitate the conversations between your stakeholders that happen during threat modelling
2. Persist the threat model in such a way that it can be easily updated and is available to all stakeholders. The threat model contains the modelling artefacts as well as supporting evidence.

When you evaluate tools to persist threat models, the operational security must also be guaranteed. A threat model will contain sensitive information in the form of unsolved security problems and the access should be limited as much as possible. This creates a field of friction between the operational security and the other requirements for threat modeling. A vision on how to handle both sets of requirements should be decided on and serve as a guideline during tool selection.

Combining the two requirements from above with the operational security might lead you to have several places / tools where (partial) information can be shared with your different shareholders.

## 6.3 Integration in your threat modeling methodology

Let`s first remind ourselves of the rule we stipulated in chapter 4.1: a tool should support your process, never change your process to accommodate a tool. As a result of this rule you should first gather the requirements for your tool(s).

There can be several tools already in use in your organization. Where possible you should reuse these tools since this will limit the friction to implement the process, limit the costs (both in potential licensing and learning curve).

While looking at the Dev(Sec)Ops[18] process and reviewing typical tools being used here, we can distinguish several possible things to harmonize:

- Several architecture diagrams might already exist, they can potentially be re-used. These diagrams tend to need extensions for threat modeling. Most diagramming tools can handle this well.
- A wiki like tool where developers put information on the sprints, stories etc. might already exist. This might be the most current version of the documentation of the system, a threat model could be considered part of that documentation.
- To follow up the actions coming from a threat model (i.e. implementing approved mitigations) you should use the same ticket tracking tool used for other development activities.
- The risk calculation methodology used to score security problems (for example from a penetration test) should be re-used during threat modeling so that different risks can easily be compared to each other.

Make sure that the threat model tools fits within your Dev(Sec)Ops pipeline, either by re-using tools or by acquiring tools that support that process. The output of the threat model process can be viewed as development artefacts. The best example of this is to treat threat models as code. Threat modeling as code follows the trend to have everything ‘as code’. The different elements of a threat model are described, often in a text format that is both human and machine readable. From this, some of the frameworks can generate documentation, automate tests, automate the risk calculation etc. This style of threat modeling lends itself very well to continuous development practices. The threat model process will also be a continuous process with a heavy focus on tool support and integration in the development team.

Treat modeling as code is not a mature field of expertise at the time of writing this document (2020). Additionally, not that many organizations are ready to do infrastructure as code, threat modeling as code etc. therefore we consider this to be an optional step in the playbook reserved for organizations with the highest level of maturity in both threat modeling practices as agile processes and tools.

[18] See Glossary of Terms

# 7 Get certified as a Threat Modeling Practitioner

If you are serious about security, and software is core to your organization, we would like to emphasize the need for in-deep Threat Modeling expertise. Reading this whitepaper is a great start, but if you want to hone your skills you need to apply them.

Training your team in Threat Modeling ensures they have **the skills to identify threats and continuously iterate** on existing threat models. This way, security becomes part of the development lifecycle of your software or hardware, lowering total costs and increasing security.

Toreon's 'Threat Modeling Practitioner' course is a 20 hour course that is based on our own experiences with Threat Modeling. It combines theory with a lot of hands-on application. You'll learn to build your own Threat Model and review this with our trainers.

Learn more about our upcoming training:

**'Threat Modeling Practitioner'**

Past participant feedback:

*"This training takes a very practical approach. You can feel that it is rooted in Seba's extensive field experience with Threat Modeling"*

*"The hands-on real-life exercises truly helped."*

# 8 Glossary of Terms

This glossary covers a list of words that are frequently used while threat modeling and their meaning.

- **ASVS:** The OWASP Application Security Verification Standard (ASVS) Project provides a basis for testing web application technical security controls and also provides developers with a list of requirements for secure development. (OWASP Application Security Verification Standard, n.d.)
- **Attack tree:** Attack trees provide a formal, methodical way of describing the security of systems, based on varying attacks. Basically, you represent attacks against a system in a tree structure, with the goal as the root node and different ways of achieving that goal as leaf nodes (Schneier, 1999)
- **Azure DevOps:** Azure DevOps provides developer services to support teams to plan work, collaborate on code development, and build and deploy applications. Developers can work in the cloud using Azure DevOps Services or on-premises using Azure DevOps Server. Azure DevOps Server was formerly named Visual Studio Team Foundation Server (TFS). (What is Azure DevOps, n.d.)
- **CVSS:** The Common Vulnerability Scoring System (CVSS) provides a way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity. The numerical score can then be translated into a qualitative representation (such as low, medium, high, and critical) to help organizations properly assess and prioritize their vulnerability management processes (Common Vulnerability Scoring System SIG, n.d.)
- **DevOps**: DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality
- **DevSecOps**: DevSecOps means thinking about application and infrastructure security from the start. It also means automating some security gates to keep the DevOps workflow from slowing down. (What is DevSecOps, n.d.)
- **G-suite:** G Suite—formerly known as Google Apps for Work—is a Software as a Service (SaaS) product that groups all the cloud-based productivity and collaboration tools developed by Google for businesses, institutes, and non-profits (Gavin, 2019)
- **JIRA**: Jira Software is part of a family of products designed to help teams of all types manage work. Originally, Jira was designed as a bug and issue tracker. But today, Jira has evolved into a powerful work management tool for all kinds of use cases, from requirements and test case management to agile software development (What is Jira used for, n.d.)
- **LINDDUN:** A privacy threat modeling methodology that supports analyst systematically eliciting and mitigating privacy threats in software architectures. LINDDUNN is a mnemonic for the privacy threat categories it supports: Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance (LINDDUN privacy engineering, n.d.)
- **MS teams:** Microsoft Teams is a communication and collaboration platform that combines workplace chat, video meetings, file storage, and application integration. The service integrates with the Office 365 subscription office productivity suite and features extensions that can integrate with non-Microsoft products (Microsoft Teams, 2020)
- **OWASP risk rating methodology:** Risk rating methodology introduced by OWASP which approaches risk rating in six steps: Identifying a risk, factors for estimating likelihood, factors for

estimating impact, determining severity of the risk, deciding what to fix, and customizing your risk rating model.

- **Playbook**: A book containing a team's strategies and plays. A set of rules or suggestions that are suitable for a particular activity, industry, or job.
- **SAMM**: OWASP SAMM stands for Software Assurance Maturity Model. This is an open source project from OWASP, more details are available on https://owaspsamm.org/about/
- **Security champion**: Security Champions are active members of a team that may help to make decisions about when to engage the Security Team. They act as a core element of security assurance process within the product or service and hold the role of the Single Point of Contact (SPOC) within the team. More details are available at: https://github.com/c0rdis/security-champions-playbook
- **Sharepoint:** Organizations use Microsoft SharePoint to create websites. You can use it as a secure place to store, organize, share, and access information from different devices
- **STRIDE**: a model of threats developed by Microsoft for identifying computer security threats. It provides a mnemonic for security threats in six categories: Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege
- **Squad**: Similar to a scrum team, Squads are cross-functional, autonomous teams (typically 6-12 individuals) that focus on one feature area. Each Squad has a unique mission that guides the work they do, an agile coach for support, and a product owner for guidance.
- **Threat Modeling**: Threat modeling is the activity of identifying and managing application risks. Also known as architectural risk analysis.

# 9 References

*ABOUT US*. (n.d.). Retrieved from OWASPP SAMM: https://owaspsamm.org/

c0rdis. (2020, January 21). *security champions playbook.* Retrieved from https://github.com/c0rdis/security-champions-playbook

*Common Vulnerability Scoring System SIG*. (n.d.). Retrieved from First: https://www.first.org/cvss/

Gavin, B. (2019, November 12). *What is G Suite, Anyway?* Retrieved from howtogeek: https://www.howtogeek.com/411808/what-is-g-suite-anyway/

(2018). *ISO/IEC 27005.* ISO. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/ISO/IEC_27005

Juuso, S. (2019, May). Evaluation of Threat Modeling. JAMK University of Applied Sciences.

*LINDDUN privacy engineering*. (n.d.). Retrieved from LINDDUN: https://www.linddun.org/

*Microsoft Teams*. (2020, 9 8). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Microsoft_Teams

*OWASP Application Security Verification Standard*. (n.d.). Retrieved from OWASP: https://owasp.org/www-project-application-security-verification-standard/

Schneier, B. (1999). Attack Trees. *Dr. Dobb's Journal*.

*What is Azure DevOps*. (n.d.). Retrieved from microsoft docs: https://docs.microsoft.com/nl-nl/azure/devops/user-guide/what-is-azure-devops?view=azure-devops

*What is DevSecOps*. (n.d.). Retrieved from redhat: https://www.redhat.com/en/topics/devops/what-is-devsecops

*What is Jira used for*. (n.d.). Retrieved from atlassian: https://www.atlassian.com/software/jira/guides/use-cases/what-is-jira-used-for

Yskout, K., Sion, L., Heyman, T., Wuyts, K., Van Landuyt, D., & Joosen, W. (2020, May 23). Threat modeling: from infancy to maturity. *International Conference on Software Engineering.* Seoul, South Korea: Institute of Electrical and Electronics Engineers. Retrieved from https://lirias.kuleuven.be/2947593