



Social engineering


Malware


Web-based exploits


Threat modeling


 **Video:** A proactive approach to security  
3 min


 **Ungraded Plugin:** Identify: Steps of the threat model process  
10 min


 **Video:** Chantelle: The value of diversity in cybersecurity  
2 min


 **Video:** PASTA: The Process for Attack Simulation and Threat Analysis  
4 min

 **Reading:** Traits of an effective threat model  
20 min

 **Ungraded Plugin:** Explore: Apply PASTA to an application's threat model  
10 min

 **Practice Quiz:** Activity: Apply the PASTA threat model framework  
1 question

 **Reading:** Activity Exemplar: Apply the PASTA threat model framework  
10 min

 **Practice Quiz:** Test your knowledge: Threat modeling  
5 questions

**Review: Threats in cybersecurity**  
**Congratulations on completing course 5**

# Activity Exemplar: Apply the PASTA threat model framework

Here is a completed exemplar along with an explanation of how the exemplar fulfills the expectations for the activity.

## Completed Exemplar

To review the exemplar for this course item, click the link and select *Use Template*.

Link to exemplar: [PASTA worksheet exemplar](#) 

OR

If you don't have a Google account, you can download the exemplar directly from the following attachment.

 **PASTA worksheet exemplar**  
DOCX File

## Assessment of Exemplar

Compare the exemplar to your completed activity. Review your work using each of the criteria in the exemplar. What did you do well? Where can you improve? Use your answers to these questions to guide you as you continue to progress through the course.

**Note:** *The exemplar represents one possible way to complete the activity. Yours will likely differ in certain ways. What's important is that your activity includes information at each stage of the process. Threat modeling is an advanced practice in cybersecurity. It normally requires experience in the field, deep knowledge of computer technology, and many different people to participate.*

Let's review each stage of this PASTA threat modeling exercise:

### Stage I: Define business and security objectives

**Summary:** These objectives are defined early by asking broad questions about the purpose of the application. For example, how does the app make the business money? Understanding the answer to these questions helps guide the detailed work that will follow.

**Recommendations:** A shopping application like this will need to process payments. Based on this description, we know certain technologies are required to keep information private and secure and that everything will need to be compliant with PCI-DSS.

### Stage II: Define the technical scope

**Summary:** The objective here is to understand the attack surface by identifying the technologies being used by the application and understanding their dependencies.

**Recommendations:** APIs facilitate the exchange of data between customers, partners, and employees, so they should be prioritized. They handle a lot of sensitive data while they connect various users and systems together. However, details such as which APIs are being used should be considered before prioritizing one technology over another. So, they can be more prone to security vulnerabilities because there's a larger attack surface.

### Stage III: Decompose the application

**Summary:** Stage three builds upon the previous stage by investigating how the application's components communicate together. The objective here is to review how the application works and how security controls are currently implemented.

**Recommendations:** The sample data flow diagram shows how a typical search request passes through multiple layers. One thing you might review here would be to ensure the MySQL database is using prepared statements when queries are input.

### Stage IV: Threat analysis

**Summary:** The main objective of stage four is to consider the types of threats that might affect your application. This relates to the technologies you've already scoped. Another thing to consider is the types of data your application will be processing.

**Recommendations:** Injection attacks are common for SQL databases. Session hijacking is possible because the app communicates cookies between multiple layers. It's important to consider your technological attack surface and any relevant threats to your product to effectively implement your information security responsibilities.

### Stage V: Vulnerability analysis

**Summary:** Stage five is about associating asset vulnerabilities with potential threats. The objective here is to identify what is wrong with the design of the app or its codebase based on your security testing.

**Recommendations:** A lack of prepared statements can make our SQL database vulnerable to injection attacks. And session hijacking is possible if cookies are mishandled between input and output sources.

### Stage VI: Attack modeling

**Summary:** In this stage, the objective is to link the threats and vulnerabilities identified in the previous steps using attack trees. The purpose of using attack trees here is to show that the potential threats that you've identified are actually viable. Resources like MITRE ATT&CK and the CVE® list are useful references to find evidence that validates the information that you've modeled in your attack tree.

**Recommendations:** This sample attack tree models how user data is vulnerable to the attacks that were identified earlier. Like the sample data flow diagram, an actual attack tree for a mobile application would be much more complex than this.

### Stage VII: Risk analysis and impact

**Summary:** The objective of the final stage of PASTA is to identify ways to mitigate the risks that were identified from stages IV - VI and plan for any remaining risks that can't be remediated.

**Recommendations:** SHA-256, incident response procedures, password policy, and principle of least privilege are a few examples of technical, operational, and managerial controls that can be implemented before launch to reduce risk.



Mark as completed

 Like  Dislike  Report an issue