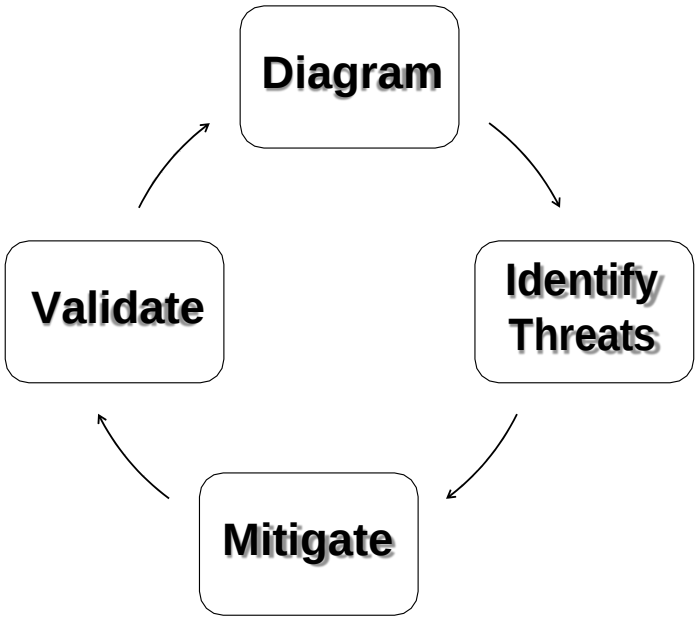
Threat Modeling

Introduction

The goal of this lab is to practice and think about the threat modelling process.

Almost all software systems today face a variety of threats, and the number of threats grows as technology changes. Malware that exploits software vulnerabilities grew 151 percent in the second quarter of 2018, and cyber-crime damage costs are estimated to reach $6 trillion annually by 2021. Threats can come from outside or within organizations, and they can have devastating consequences. Attacks can disable systems entirely or lead to the leaking of sensitive information, which would diminish consumer trust in the system provider. To prevent threats from taking advantage of system flaws, administrators can use threat-modeling methods to inform defensive measures.

In this lab, we will follow a process of threat modelling and have a basic view about how to design a secure system. The following diagram shows a nutshell of the process:



1 Diagram

The Data Flow Diagram (DFD) is a structured analysis and design method. It is traditional visual representation of the information flows within a system. Data Flow Diagram (DFD) is widely used for software analysis and design. A neat and clear DFD can depict a good amount of the system requirements graphically.

1.1 DFD Diagram Notations:

- **External Entity**
  An external entity can represent a human, system, or subsystem. It is where certain data comes from or goes to. It is external to the system we study, in terms of the business process. For this reason, people used to draw external entities on the edge of a diagram.

- **Process**
  A process is a business activity or function where the manipulation and transformation of data takes place. A process can be decomposed to finer level of details, for representing how data is being processed within the process.

- **Data Store**
  A data store (also known as data repository) holds information for processing. It represents a situation when the system must retain data because one or more processes need to use the stored data in a later time.

- **Data Flow**
  A data flow represents the flow of information, with its direction represented by an arrowhead that shows at the end(s) of flow connector.

- **Trust Boundaries**
  Trust boundaries represent the border between trusted and untrusted elements. They can be points/-surfaces where an attacker can interject. (e.g., Machine boundaries, privilege boundaries, and integrity boundaries are examples of trust boundaries) Threads in a native process are often inside a trust boundary, because they share the same privileges, rights, identifiers, and access. Processes talking across a network always have a trust boundary.

Question 1:
Assume we are going to design a frequent flyer update operation of an airline:

Frequent flyer points of valid customers are updated by a separate program. As soon as the plane lands at the destination, it triggers a program (say copy program) which copy (transmits) the frequent flyer information of the passengers in that flight into another area. An update program takes this data and updates the frequent points of the passengers. There was a breach detected where an individual who had access to the system which updates the frequent flyer info was able to rack up millions of miles without even flying!

Draw a DFD for this sub-system and how it interacts with other parts of the system.

2 Identify Threat

In this step, we will use the STRIDE threat-modeling method to step through the diagram elements and identify threats. Invented in 1999 and adopted by Microsoft in 2002, STRIDE is currently the most mature threat-modeling method. It can help you identify potential threats in your product during a security analysis.

Recall: S.T.R.I.D.E. stands for:

- **Spoofing:** Spoofing refers to the act of posing as someone else (i.e., spoofing a user) or claiming a false identity (i.e., spoofing a process).
- **Tampering:** Tampering refers to malicious modification of data or processes. Tampering may occur on data in transit, on data at rest, or on processes.
- **Repudiation:** Repudiation refers to the ability of denying that an action or an event has occurred.
- **Information disclosure:** Information Disclosure refers to data leaks or data breaches. This could occur on data in transit, data at rest, or even to a process.
- **Denial of Service:** Denial of Service refers to causing a service or a network resource to be unavailable to its intended users.
- **Elevation of privilege:** Elevation of Privileges refers to gaining access that one should not have.

Question 2:

Example Question

a) What property each threat type is targeting?

Threat Type	**Security Property**
Spoofing	
Tampering	
Repudiation	
Information Disclosure	
Denial of Service	
Elevation of Privilege	

b) Could you provide an example for each threat type?

Question 3:

The table in Appendix lists some of the common threats you will come across when developing software applications.

a) In the lab, for each threat, your task is to identify the type of threat according to the STRIDE classification.

b) Apply STRIDE to each element in your DFD and identify the threat(s).

3 Mitigate

After identifying the threats of a system, we need to find an appropriate mitigation approach for each one of them. Mitigation is an important part of Threat Modeling. since it constitutes the way to address or alleviate the threats and design a secure software solution.

There are four ways to address threats:

- Redesign to eliminate.
- Apply standard mitigations.
- Invent new mitigations.
- Accept vulnerability in design.

While redesigning and accepting the vulnerability do address the threats, we would focus more on the mitigation.

Inventing mitigations is hard. Commonly, mitigations are an area of expertise, such as networking, databases, or cryptography. Amateurs would make mistakes and the mitigation failures will appear to work until an expert looks at them. In this lab, we will focus more on applying standard mitigations.

Recall:

Some standard mitigation examples for each threat type:

Threat Type	Security Property
Spoofing	authentication
Tampering	validation of users' inputs and proper encoding of outputs
Repudiation	audit logging
Information Disclosure	encryption
Denial of Service	log rotation and monitoring/alerting when disk is nearing capacity
Elevation of Privilege	authorization mechanism

Question 4:

a) Identify the potential mitigation technique(s) for the threats in the Appendix table based on the threat type you found in the previous step.

b) Describe mitigation strategies for your DFD.

4 Validate

Now you can validate the whole threat model. If some threats that are not mitigated correctly are found, then this issue can be corrected in a next iteration of the threat model design process.

Asking yourself:

- Does diagram match the final code?
- Are all the threats enumerated?
- Is each threat mitigated?
- Are mitigations done in a right manner?

Question 5:

a) Identify the issues related to the threat and/or its mitigation for the threats in the table based on the threat type you find in the previous step.

b) Further improve your design by improving your DFD.

Additional Link about Threat Modeling:

- 'The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software'(Chapter 9) *https://download.microsoft.com/download/8/1/6/816C597A-5592-4867-A0A6-A0181703CD59/Microsoft_Press_eBook_TheSecurityDevelopmentLifecycle_PDF.pdf*
- 'Improving Web Application Security: Threats and Countermeasures'(Chapter 3 Threat Modeling) *https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648644(v%3dpandp.10)*
- 'Security Briefs: Reinvigorate your Threat Modeling Process' *http://msdn.microsoft.com/en-us/MAGAZINE/cc700352.aspx*

Some open-source tools that you can use for this tutorial:
Microsoft Threat Modeling Tool: *https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling*
Threat Dragon: *https://owasp.org/www-project-threat-dragon/*

Appendix Table

THREAT	THREAT TYPE	MITIGATION TECHNIQUE(S)	DESCRIPTION OF ISSUES
Access to or modification of confidential HTTP data			
Access to or modification of confidential RPC or DCOM data			
A device that contains confidential data might be lost			
Flood service with too many connections			
Attacker attempts to guess passwords			
Read confidential cookie data			
Tamper with cookie data			
Access private/secret data			
Attacker spoofs a server			
Attacker posts HTML or script to your site			
Attacker opens thousands of connections but does nothing with them			
Unauthenticated connection can consume memory			
Your data packets can replayed			
Attacker attaches a debugger to your process			
Attacker gains physical access to hardware			
Attacker shuts down your process			