# Threat modeling

## Summary

In this class exercise, we will learn about threat modeling, discuss its purpose, and practice performing the activity. Part 1, this should be performed without the aid of any electronic device.

## Prerequisites

* Windows OS machine with Microsoft Threat Modeling tool installed

## Details

### Part 1

Threat modeling is a technique meant to improve the effectiveness of a security assessment. By thinking from the perspective of the attacker, security analysts can identify components in a system that would have the highest risk of attack. These components should have priority in security assessments.

Discuss the threat modeling process (Meier et al., 2005) steps below

1. Identify assets
   1. Identify the valuable assets that the system must protect
2. Create an architecture overview
   1. Use tables and threat modeling diagrams to document the architecture of the application, including subsystems, trust boundaries, and data flow
3. Decompose the application
   1. Decompose the architecture of the application, including the underlying network and host infrastructure design, to create a security profile for the application
4. Identify the threats
   1. Keeping the goals of an attacker in mind, and with knowledge of the architecture and potential vulnerabilities of the application, identify the threats that could affect the application
5. Document the threats
   1. Document each threat using a common threat template that defines a core set of attributes to capture for each threat
6. Rate the threats
   1. Rate the threats to prioritize and address the most significant threats first



Figure 1-Overlapping types of assets (Shostack text)

* Analyze the figure above. What are some examples for assets in a software-based system?
* What are some examples for threats to a software-based system?
* What are some examples for countermeasures, or security features, which could protect assets against threats?

Consider the simple web system in the figure below



Figure 2-Simple web system (Shostack)

* Provide two notional examples for assets related to the system

Several activities support identification of threats and these will be used in the homework.

* STRIDE
* Attack trees
* Categorized threat lists and attack libraries
* Attack patterns

Before we practice these activities in the homework, what are some notional examples for threats?

### Part 2

Work through the following example exercise from the Shostack text and create the corresponding threat diagram using the Microsoft Threat modeling tool.

#### Background

A tool, named iNTegrity, is a simple file-integrity checking tool that reads resources, such as files in the filesystem, determining whether any files or registry keys have been changed since the last check. This is performed by looking at the following:

* File or key names
* File size or registry data
* Last updated time and date
* Data checksum (MD5 and/or SHA1 hash)

Architecturally, the tool is split into two parts: a host component and an administrative console. As shown below, one client can communicate with multiple servers, rather than running the tool locally on each computer.



Figure 3-The networked host/admin console nature of the iNTegrity tool

In another operational environment, it might be known that a machine has been compromised and can no longer be trusted, and the server and client software can be run off, say, a bootable CD or USB drive. In this case, the integrity checking code is running under a trusted, read-only Windows environment, and the host and admin components both read data from the compromised machine, but not using the potentially compromised OS. The host process does not run as a Windows service in this mode, but as a standalone console application.



Figure 4-Context diagram

#### The Host Component

This small host component is written in C++ and runs as a service on a Windows server. Its role is to take requests from the admin console and respond to those requests. Valid requests include getting information about host component version, and recursive and non-recursive file properties. Note that the host software performs no analysis; it sends raw integrity data (filenames, sizes, hashes, ACLs, and so on) to the admin console, which performs the core analysis.

#### The Admin Console

The admin console code stores and analyzes resource (file, registry) version information that comes from one or more host processes. A user can instruct the admin console to connect to a host running the iNTegrity host software, get resource information, and then compare that data with a local, trusted data store of past resource information to see if anything has changed.

#### Tasks

Draw the threat model that contains elements for:

* Registry
* Filesystem
* Integrity host software
* Integrity admin console
* Config data
* Integrity files
* Admin user

Include the following data flows and label them with the data types:

* Registry data
* Raw FS data
* Commands
* Resource integrity data
* Read configuration settings
* Read integrity files
* Update integrity files
* User instructions
* Integrity change information

Include trust boundaries for the administrator and domain administrator

Identify three or more threats: one for a data flow, one for a data store, and one for a process.

Rate these threats as low-medium-high.

What are some mitigations for these threats?

### Definitions

* Threat modeling-a process that identifies assets, identifies threats to those assets (from the perspective of the attacker), and assesses the security of the software architecture protecting those assets
* Security profile-categorized reference that aims to uncover vulnerabilities in the design, implementation, or deployment configuration of your application; categories includes threat types, asset types, technology, etc.
* Attack (or exploit)-An action taken by someone or something that harms an asset. This could be someone following through on a threat or exploiting a vulnerability.
* Entry points-show where an outsider can pass input into a system
* Attack surface-surface within a system, including entry points and all paths outside input takes through the system, that is vulnerable to external threats
* Vulnerability-A weakness in some aspect or feature of a system that makes a threat possible. Vulnerabilities might exist at the network, host, or application levels.