Preparation Notice

This tutorial is a summary of the STRIDE threat modeling framework used in Microsoft Security Development Lifecycle (SDL) [1]. The tutorial covers the STRIDE steps up to ‘obtaining threats’ step (does not consider threats mitigation or risk management). The summarized steps are based on [2] and [3]. The social network example, Figure 1, and Table 2 are adopted from [3]. Table 1 is adopted from [4]. The threat tree patterns are replicated from [1].

This notice is not given to the participants during the user study to avoid disclosing the name of the threat modeling framework.

References

[1] Howard, Michael, and Steve Lipner. The security development lifecycle. Vol. 8. Redmond: Microsoft Press, 2006.

[2] Torr, P.: Demystifying the threat modeling process. IEEE Security & Privacy 3(5), 66–70 (2005)

[3] Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering 16(1), 3–32 (2011)

[4] Microsoft Threat Modeling Tool 2016 User Guide, https://www.microsoft.com/en-us/download/details.aspx?id=49168

Threat Modeling Framework A - Tutorial

Consists of three steps:

1. System Model Characterization.
2. Threat Identification.
3. Threat Scenario Enumeration.

1. System Model Characterization.

Before we begin, we need to understand the system.

- First, define the **use scenarios** of the system that cover its key functionality, any **external dependencies** on other systems or technologies, and any **assumptions** the system makes.
- Second, break the system into **components** based on its functionality and for each component identify its entry points, the participants, their trust level, the protected assets such as computing resources, storage, user's personal information, etc.
- Third, model the components as **data flow diagrams (DFDs)**. A DFD contains the following elements: data flows (i.e. communication data), data stores (i.e. databases, files, etc.), processes (i.e. programs and functionalities), external entities (i.e. end users, external services, etc.).

Example Scenario: *Social Network 2.0. Alice is a registered user of a social network. Each time she connects with a new friend she first connects to the social network's web portal. The portal communicates with the social network's server and the friendship information of Alice and her friend in the database is updated.*

Running example application:

List of components: *update friendship information.*

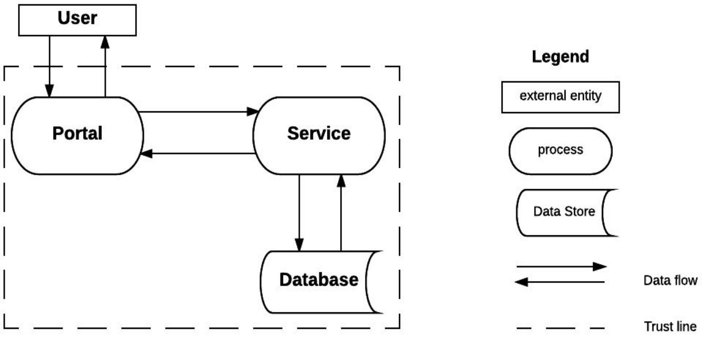
List of participants: *server and users.*

List assets of value: database of user's personal information.

List assumptions and external dependencies:

- *the server that manages the users' information database and provides the social network website (or portal) is trusted.*
- *End users are not trusted.*

DFD: *next figure.*



2. Threat Identification.

In this step we identify the threats that may take place in the system based on the assets of value and the activities performed by the participants. It might be helpful to use the threat categories in Table 1 and their mapping to the DFD elements found in Table 2.

Table 1: Threat categories.

Threat	Threat Definition	Examples
Spoofing	Spoofing threats involve an adversary creating and exploiting confusion about who is talking to whom. Spoofing threats apply to the entity being fooled, not the entity being impersonated. Thus, external elements are subject to a spoofing threat when they are confused about what or whom they are talking to.	• Accounts.contoso.com is spoofed when it thinks that a user is giving it authorized credentials. • An adversary may have poisoned the DNS cache so accounts.contoso.com now points at a malicious system that looks exactly like the real accounts.contoso.com.
Tampering	Tampering threats involve an adversary modifying data, usually as it flows across a network, resides in memory, on disk, or in databases.	• An adversary tampers with network packets, and changes commands after the user has logged in. • An adversary tampers with a registry key, making us run any program they choose.
Repudiation	Repudiation threats involve an adversary denying that something happened.	• Joe denies that he clicked on that link, for example to deny that he has benefited from a financial transaction. • Amy receives an email from Joe in which he agrees to a contract between the two. Later, Joe denies ever having sent that email.
Information disclosure	Exposing information to someone not authorized to see it.	• Examples include passwords for known or unknown users, copies of emails, and names and social security numbers in a database.
Denial of service	Deny or degrade service to users.	• An adversary prevents customers from connecting to a website. • An adversary prevents the client from getting a DNS response.
Elevation of privilege	Gain capabilities without proper authorization.	• An adversary who starts as an anonymous internet user can send commands to an application that execute as the web server. • An adversary with a web server can make code run as the local user.

Table 2: Mapping of DFD elements to threat categories found in Table 1.

Threat	**Data Flow**	**Data Store**	**Process**	**External Entity**
Spoofing			✓	✓
Tampering	✓	✓	✓	
Repudiation		✓	✓	✓
Information disclosure	✓	✓	✓	
Denial of service	✓	✓	✓	
Elevation of privilege			✓	

Running example application:
Threats: Tampering with the data flow between the user and the portal, spoofing the server, denial of service against the social network server, etc.

3. Threat Scenario Enumeration

The next step is to elicit the threats in the system as follows:

- Examining each threat and enumerate all its possible scenarios.
- Write down a brief description for each threat scenario that must be addressed by the system design.
- You may find the **threat tree patterns** (attached to this tutorial) useful in this process.

Due to time constraints, you will ***work on only one threat*** *in this study*. **STOP AT THIS POINT AND ASK THE MONITOR TO HAND YOU THE REST OF THE STUDY THAT DETERMINES WHICH THREAT TO WORK ON.**

Example:
Threat: Spoofing the server.

1) The portal does not implement any authentication service, an attacker can pretend to be Alice and manipulate her information through the server.
2) The portal implements a weak authentication service, an attacker is able to obtain the credentials or downgrade the authentication.
3) Alice stores her credential in an insecure location, an attacker is able to access this location and steal her credentials and spoofed her.
4) Alice is using weak credentials, an attacker is able to guess them and access the portal.