Web Application Security

Rajendra Kachhwaha
rajendra1983@gmail.com

July 27, 2015

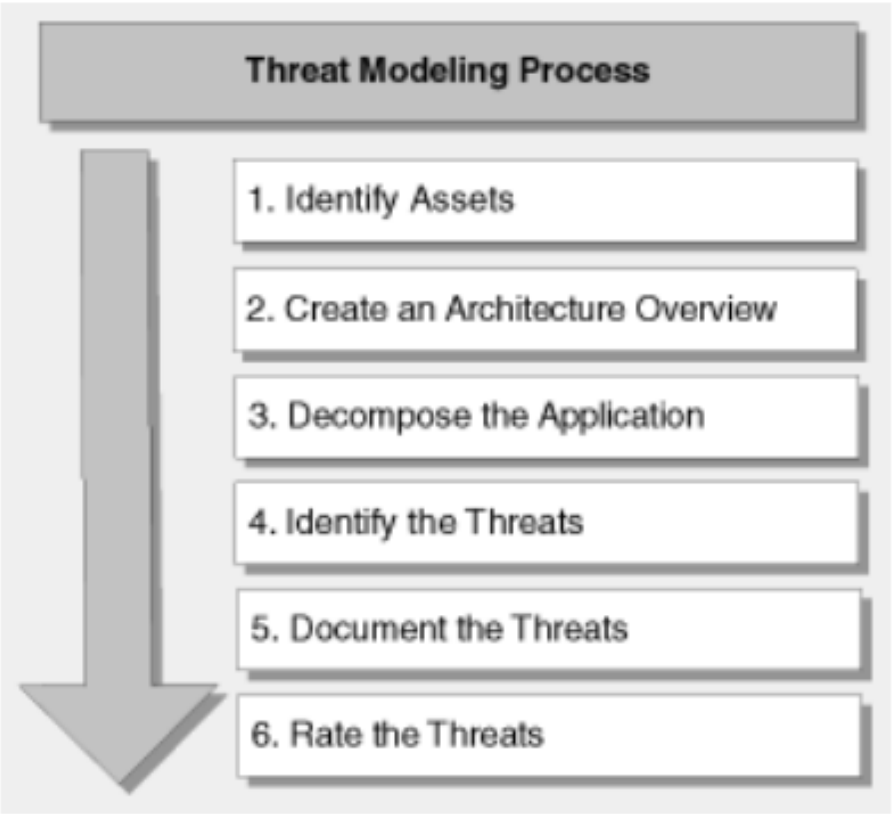
Outline

Threat Modeling

1. The purpose of threat modeling is to analyze your application's architecture and design and identify potentially vulnerable areas that may allow a user, perhaps mistakenly, or an attacker with malicious intent, to compromise your system's security.
2. Threat modeling should not be a one time only process.
3. It should be an iterative process that starts during the early phases of the design of your application and continues throughout the application life cycle. There are two reasons for this. First, it is impossible to identify all of the possible threats in a single pass. Second, because applications are rarely static and need to be enhanced and adapted to suit changing business requirements, the threat modeling process should be repeated as your application evolves.

Threat Modeling

The threat modeling process that you can perform using the following six-stage process:



Threat Modeling

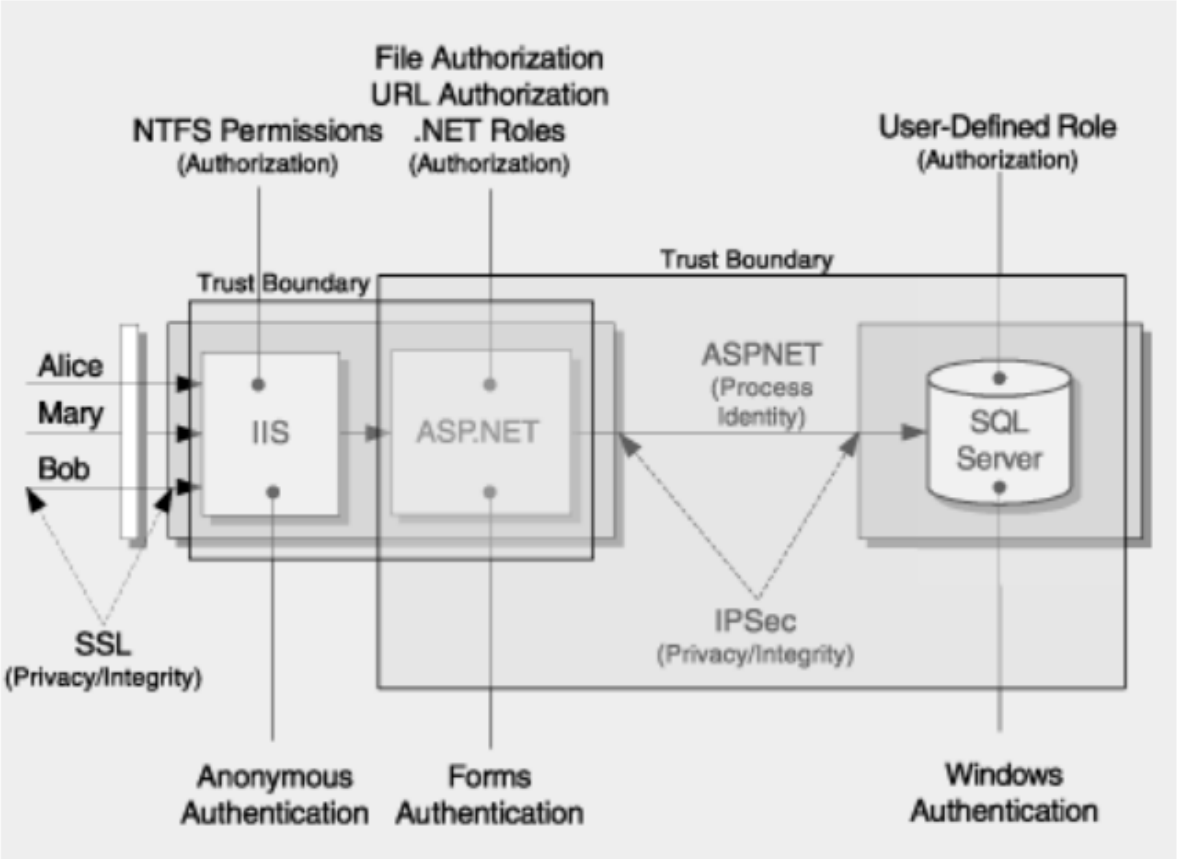
Step 1. Identify Assets: Identify the assets that you need to protect. This could range from confidential data, such as your customer or orders database, to your Web pages or Web site availability.

Step 2. Create an Architecture Overview: At this stage, the goal is to document the function of your application, its architecture and physical deployment configuration, and the technologies that form part of your solution. During this step, you perform the following tasks:

- Identify what the application does.
- Create an architecture diagram.
- Identify the technologies.

Threat Modeling

Create an Architecture Diagram:



Threat Modeling

Identify the Technologies:

Technology/Platform	Implementation Details
Microsoft SQL Server on Microsoft Windows Advanced Server 2000	Includes logins, database users, user defined database roles, tables, stored procedures, views, constraints, and triggers.
Microsoft .NET Framework	Used for Forms authentication.
Secure Sockets Layer (SSL)	Used to encrypt HTTP traffic.

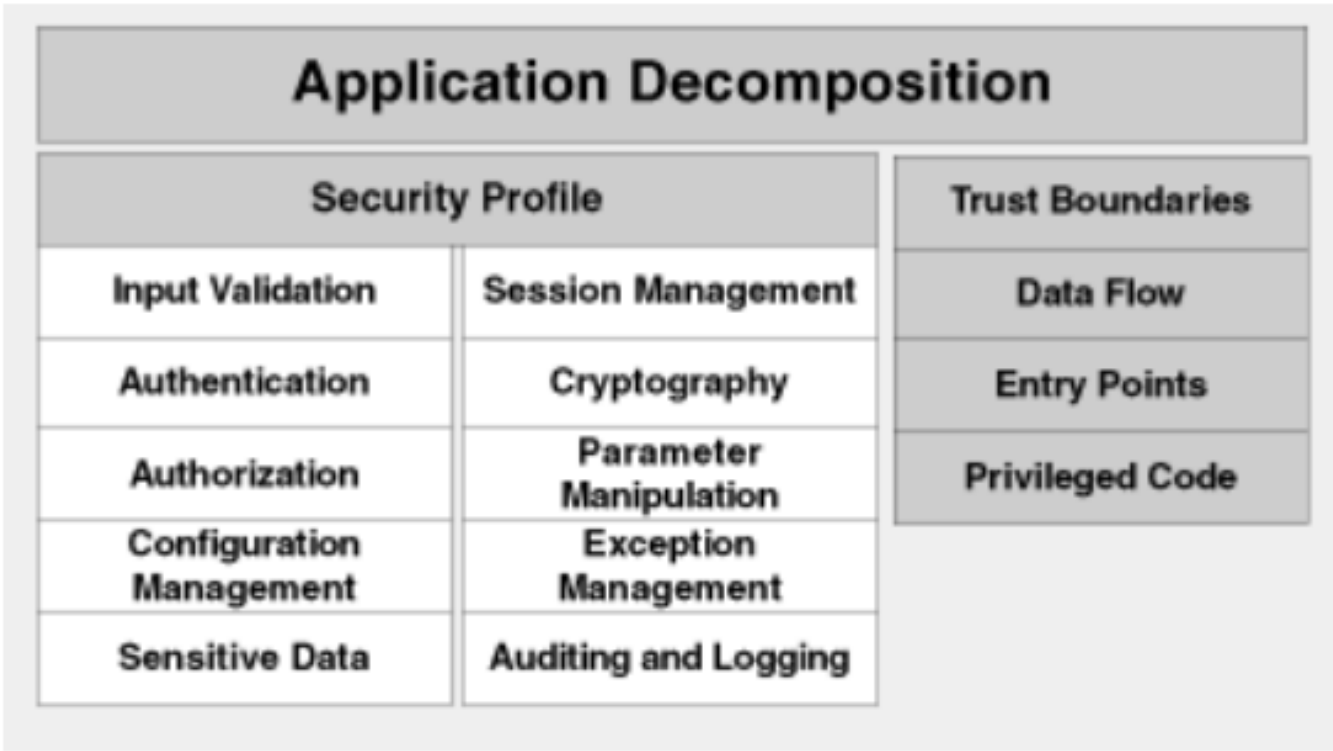
Threat Modeling

Step 3. Decompose the Application:
In this step, you break down your application to create a security profile for the application based on traditional areas of vulnerability.During this step, you perform the following tasks:

- Identify trust boundaries.
- Identify data flow.
- Identify entry points.
- Identify privileged code.
- Document the security profile.

Threat Modeling

Step 3. Decompose the Application:



Threat Modeling

Step 4. Identify the Threats:
In this step, you identify threats that might affect your system and compromise your assets. To conduct this identification process, bring members of the development and test teams together to conduct an informed brainstorming session in front of a whiteboard. During this step, you perform the following tasks:

- Identify network threats.
- Identity host threats.
- Identify application threats.

Threat Modeling

Step 5. Document the Threats:
To document the threats of your application, use a template that shows several threat attributes similar to the shown on next page. The threat description and threat target are essential attributes. Leave the risk rating blank at this stage. This is used in the final stage of the threat modeling process when you prioritize the identified threat list. Other attributes you may want to include are the attack techniques, which can also highlight the vulnerabilities exploited, and the countermeasures that are required to address the threat.

Threat Modeling

Step 5. Document the Threats:

Threat 1

Threat Description	Attacker obtains authentication credentials by monitoring the network
Threat target	Web application user authentication process
Risk	
Attack techniques	Use of network monitoring software
Countermeasures	Use SSL to provide encrypted channel

Threat 2

Threat Description	Injection of SQL commands
Threat target	Data access component
Risk	
Attack techniques	Attacker appends SQL commands to user name, which is used to form a SQL query
Countermeasures	Use a regular expression to validate the user name, and use a stored procedure that uses parameters to access the database.

Threat Modeling

Step 6. Rate the Threats:
At this stage in the process, you have a list of threats that apply to your particular application scenario. In the final step of the process, you rate threats based on the risks they pose. This allows you to address the threats that present the most risk first, and then resolve the other threats.

Risk = Probability * Damage Potential

This formula indicates that the risk posed by a particular threat is equal to the probability of the threat occurring multiplied by the damage potential, which indicates the consequences to your system if an attack were to occur. You can use a simple High, Medium, or Low scale to prioritize threats.

Threat Modeling

DREAD Model:

The problem with a simplistic rating system is that team members usually will not agree on ratings. At Microsoft, the DREAD model is used to help calculate risk. By using the DREAD model, you arrive at the risk rating for a given threat by asking the following questions:

- **D**amage potential: How great is the damage if the vulnerability is exploited?
- **R**eproducibility: How easy is it to reproduce the attack?
- **E**xploitability: How easy is it to launch an attack?
- **A**ffected users: As a rough percentage, how many users are affected?
- **D**iscoverability: How easy is it to find the vulnerability?

Threat Modeling

DREAD Model:Thread Rating Table

	Rating	High (3)	Medium (2)	Low (1)
D	Damage potential	The attacker can subvert the security system; get full trust authorization; run as administrator; upload content.	Leaking sensitive information	Leaking trivial information
R	Reproducibility	The attack can be reproduced every time and does not require a timing window.	The attack can be reproduced, but only with a timing window and a particular race situation.	The attack is very difficult to reproduce, even with knowledge of the security hole.
E	Exploitability	A novice programmer could make the attack in a short time.	A skilled programmer could make the attack, then repeat the steps.	The attack requires an extremely skilled person and in-depth knowledge every time to exploit.
A	Affected users	All users, default configuration, key customers	Some users, non-default configuration	Very small percentage of users, obscure feature; affects anonymous users
D	Discoverability	Published information explains the attack. The vulnerability is found in the most commonly used feature and is very noticeable.	The vulnerability is in a seldom-used part of the product, and only a few users should come across it. It would take some thinking to see malicious use.	The bug is obscure, and it is unlikely that users will work out damage potential.

Threat Modeling

For example, consider the two threats described earlier:

- Attacker obtains authentication credentials by monitoring the network.
- SQL commands injected into application.

Threat	D	R	E	A	D	Total	Rating
Attacker obtains authentication credentials by monitoring the network.	3	3	2	2	2	12	High
SQL commands injected into application.	3	3	3	3	2	14	High