Threat Modeling: The Art of Identifying, Assessing, and Mitigating security threats

Mohamed Ali Saleh Abomhara
University of Agder
mohamed.abomhara@uia.no

Winter School in Information Security,
Finse – May 3 – 8, 2015
Agenda

- Introduction
  - Causes of Compromised Security
  - What is Threat Modeling?
  - Why Threat Modeling?

- Threat Modeling Process (Microsoft Security Development Lifecycle (SDL))

- Demo (SDL Threat Modeling Tool)

- Conclusion
Introduction

- Causes of Compromised Security
  - Technology weaknesses
  - Configuration weaknesses
  - Policy weaknesses
  - Human error and malice
Introduction

□ What is Threat Modeling?
  ➢ Threat modeling is a structured way to identify, understand, and mitigate threats
  ➢ A road map for developer to write secure code
  ➢ Threat modeling is all about finding problems
Introduction

- Why Threat Modeling?
  - The most reliable way to
    - Find security issues in system architecture and business processes
    - Identify threats and vulnerabilities relevant to your system
    - Identifies where more resources are required to reduce risk
  
  - Helps you to
    - Understand your organization/user weaknesses
    - Shape your system design to meet your business objectives
    - Increase awareness of threats
    - Improve the security of your system by implementing effective countermeasures
Threat Model Process

- Threat modeling Terminology
  - **Role** – The set of business process capabilities of human who interacts with the system
  - **Asset** – It is something of value (in threat modeling is called a threat target).
  - **Action** – Something a role can do to asset: Create, Read, Modify, Delete
  - **Threat** – Something that takes advantage of security weaknesses in a system and has a negative impact on it.
  - **Attacks** – Actions taken to harm a system or disrupt normal operations by exploiting vulnerabilities using various techniques and tools.
  - **Vulnerability** – Is a weakness in system design, implementation, or operation.
  - **Risk** – Is the probability that something bad could happen.
Threat Modeling Process

- Threat modeling Terminology
  - **Actor** – Threat agent
  - **Data Flow Diagram (DFD)** – A diagram which models the flow of data through the system.
  - **Trust Boundary** – A DFD annotation that indicates a connection crosses between trust levels
  - **Trust** – The level of trust placed on individuals in a specific role
  - **Security Control** – Product and/or processes employed to mitigate a specific threat (or a group of threats) to an acceptable level.
Threat Modeling Process (Microsoft Security Development Lifecycle (SDL))

Vision → Diagram → Identify Threats → Mitigate → Validate → Vision
Build a list of assets and system objectives that require protection including:

- Things attackers want
  - System components (hardware and software)
  - Information such as ID number and credit card numbers
  - Anything else that, if compromised, would prevent correct operation of your system

- Scenarios
  - Use cases/Use Stories
  - Add security to scenarios, use cases
Describe System Architecture

- Create a system architecture
  - System components
  - Understand data and data classification

- Diagram the system
  - Show subsystems
  - Show data flow

- Focus on confidentiality, integrity, and availability
  - What can we prevent?
  - What do we care about most?
  - What is the worst thing that can happen?
Decompose the system

- Break down the system
  - Show the events that drive the system
  - Show the processes that are driven
  - Identify entry points
  - Identify technologies
  - Diagram trust boundaries

- Begin to think like an attacker
  - Where are my vulnerabilities?
  - What am I going to do about them?
A Real Diagram
**Identify Threats**

- **Spoofing**
  Can an attacker gain access using a false identity?

- **Tampering**
  Can an attacker modify data as it flows through the application?

- **Repudiation**
  If an attacker denies doing something, can we prove he did it?

- **Information disclosure**
  Can an attacker gain access to private or potentially injurious data?

- **Denial of service**
  Can an attacker crash or reduce the availability of the system?

- **Elevation of privilege**
  Can an attacker assume the identity of a privileged user?
Mitigate

- Option 1: Accepting the risk
- Option 2: Transferring the risk
- Option 3: Address the risk
  - Four ways to address threats:
    - Redesign to eliminate
    - Apply standard mitigations
    - Invent new mitigations (Riskier)
Validate the whole TM

- Does diagram match final code?
- Are threats enumerated?
- Minimum: STRIDE per element that touches a trust boundary
- Has Test reviewed the model?
  - Created appropriate test plans
  - Tester approach often finds issues with TM, or details
- Is each threat mitigated?
  - Are mitigations done right
Demo

- Microsoft Threat Modeling Tool
Conclusion

- The security development process requires thorough understanding of a systems assets, followed by identifying different vulnerabilities and threats that can exist.
- Use threat modeling to develop security testing strategy.
- Know your enemy and know yourself.
  - What techniques and technologies will hackers use?
  - What techniques and technologies can testers use?

Without threat modelling, protecting yourself is like “shooting in the dark”
References

• The Microsoft Security Development Lifecycle (SDL)

• The Microsoft SDL Threat Modeling Tool

• SDL blog
  http://blogs.msdn.com/sdl/

Book

THANK YOU!