



Introduction to Software Security and Threat Modeling

Adam Pridgen
The Cover of Night, LLC
February 13, 2009

Agenda

- Information Security Primer
- Threat Modeling
- Basic Example
- Threat Modeling's Role in the SDL
- Elements of a Threat Model
- Types of Threat Modeling
- Concluding Remarks
- Questions

Background Information

- Founded The Cover of Night
 - Focus on Security and Research
 - Perform a number of information security services
- Previously Employed by McAfee
- Worked in Several Research and Technical Roles
- Graduated in '05 and '07
- Member of UT IEEE ComSoc

Information Security Primer

- Information Security Makes Risk Affordable
 - System security does not have to be complete, just effective
 - Security helps reduce risk
 - Make cost to attacker really expensive
 - Reduce the probability that one system will lead to failure
- Threats attack or exploit vulnerabilities
- Vulnerabilities are Holes (Technical or Human)
- Mitigations are Actions Taken Against Vulnerabilities
- Risk is a factor of Vulnerabilities, Impact, and Occurrence

Why is Security Important?

- Confidentiality
- Availability and Reliability
- System Integrity
- Public Opinion and Company Image
- People Pay Good Money for Software
- ...

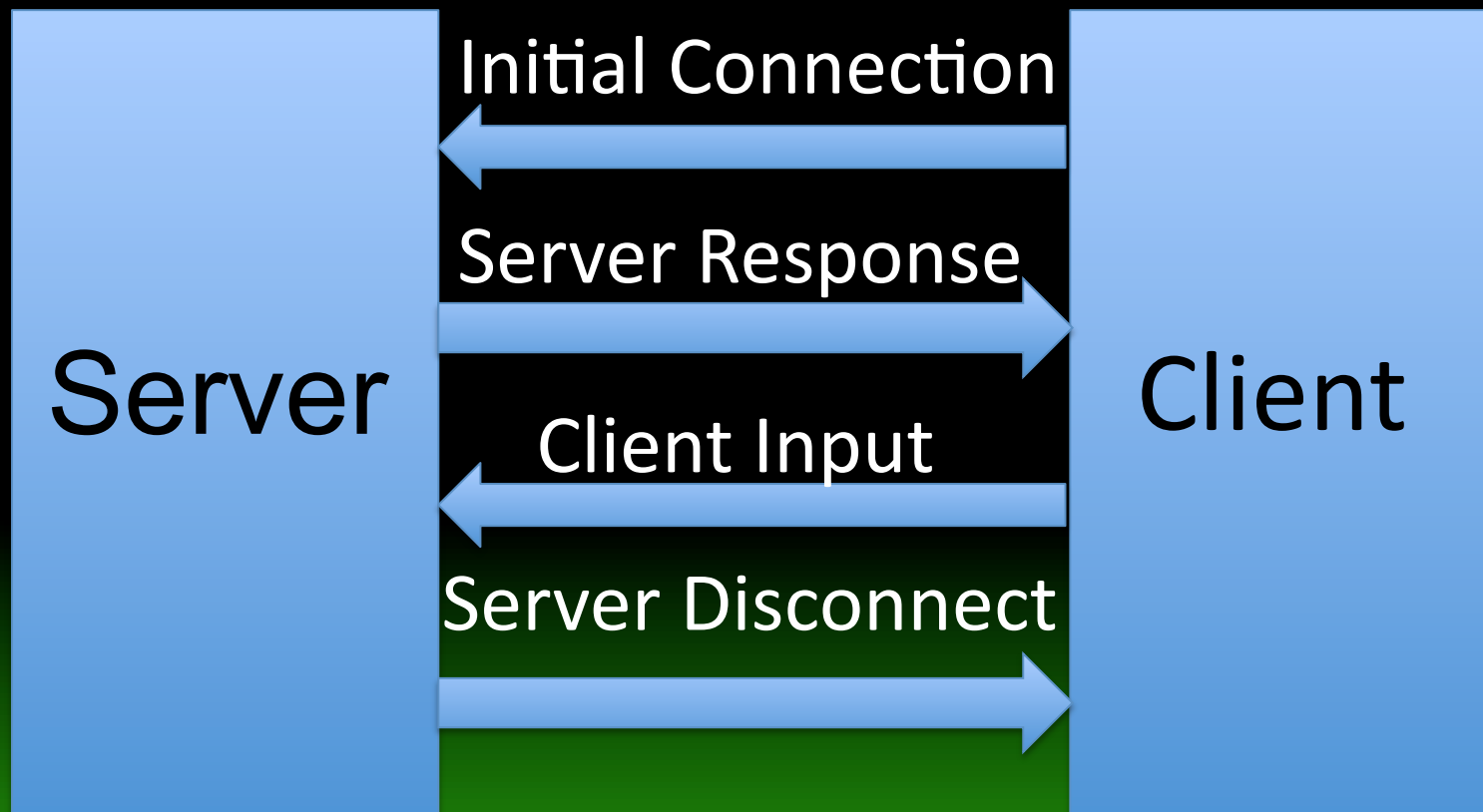
What is Threat Modeling?

- Threat Modeling is a Methodical Process
- Realize the Level of Risk at Various Views
- Develop Insight about Relationships in the System
- Identify the Areas with the Most Exposure
- Design Against Common Vulnerabilities
- Realize Common Threats Facing the Application
- Identify What to Test and How to Test Components
- Preparation for Unforeseen Threats

Threat Modeling

- Perspective may be Adversarial or Defensive
- Continuous and Evolves with the Life Cycle
- From the Perspective of Adversaries
 - Identify Holes and Vulnerabilities
 - Exploit them to Gain Access to the Objective
- From Defensive Perspective
 - Identify Probable Vulnerabilities
 - Remove as many of the vulnerabilities as possible
 - Employ countermeasures to reduce the attack risk

Basic Application Example



Threats and Attack Vectors

- Client Provides Malformed Input
- Client Provides Too Much Input
- Client Attacks Availability of the Service

Example Code Listing

```
080488E9 var_138      = dword ptr -138h
080488E9 var_134      = dword ptr -134h
080488E9 var_124      = dword ptr -124h
080488E9 var_120      = dword ptr -120h
080488E9 var_11C      = dword ptr -11Ch
080488E9 var_118      = dword ptr -118h
080488E9 var_114      = dword ptr -114h
080488E9 StaticCharBuff = byte ptr -108h
080488E9 arg_0          = dword ptr 8
080488E9
080488E9          push    ebp
080488EA          mov     ebp, esp
080488EC          sub     esp, 138h
080488F2          mov     [ebp+var_114], 0
080488FC          mov     [ebp+var_118], 0
08048906          mov     [ebp+var_11C], 0
08048910          mov     [esp+138h+var_134], 0
```

Stack Layout

....

ebp - 0x114

ebp - 0x108 (Start of Static Buffer)

old ebp (from caller)

old eip (return address for the caller)

arg_0

Example Code Listing 2

```
call    _close
mov     [esp+138h+var_138], offset aWelcomeToTheSi ; "Welcome to the simple time .
call    _printf
mov     [esp+138h+var_138], offset aWhatIsYourName ; "What is your name?\n"
call    _printf
mov     eax, ds:stdout@@GLIBC_2_0
mov     [esp+138h+var_138], eax
call    _fflush
lea     eax, [ebp+StaticCharBuff] ; Static Char Buffer of 260 Bytes
mov     [esp+138h+var_138], eax
call    _gets
lea     eax, [ebp+StaticCharBuff]
mov     [esp+138h+var_134], eax ; Saving the Buffer Address on the Stack
mov     [esp+138h+var_138], offset aPleasedToMeetY ; "Pleased to meet you %s!\n"
```

The Big Picture

- TMing is Part of the Software Development Lifecycle
- Deliver Secure and Reliable Software
- Lifecycle Covers All Aspects of the Process
- Process Includes All Stakeholders
 - Developers
 - Customers
 - ...
- Security is Only Recently Being Integrated Here

The Big Picture – SSDL Overview

- Developer Education and Awareness
- Definition of Best Practices and Attack
- Product Risk Assessment
- Threat Modeling & Risk Analysis
- Security Documentation and BP for Customers (TM)
- Security Coding Policies (TM)
- Testing Procedures and Practices (TM)
- Security Push (TM)
- Security Review (TM)

Threat Modeling - Action

- Small Portion of the Software Security Process
- ...But Makes the Strongest Impact
- Phases
 - Define Requirements
 - Propose Architecture
 - Identify Threats and Point out Mitigations
- Three Primary Means of Threat Modeling
 - Attack-Centric
 - Asset-Centric
 - Software-Centric

TM - Requirements

- Define Functional and Security Requirements
 - What are the business objectives?
 - Who will use the application?
 - How will the application be used?
 - What kind of data will the application handle?
 - What are some use and abuse cases for the app?
 - What are technical and human constraints?
 - Where will the application be located/ hosted?

TM - Architecture

- Define and Model the Application Architecture
 - What components will be used in the applications?
 - How will the systems be used?
 - What are best practices and procedures for their configuration?
 - Who will be responsible/have access to the systems?
 - What are the trust relationships between systems?
 - What are any external (3rd party) application components?

TM - Identification

- Identify the Threats (Brainstorm)
 - Identify probable threats
 - Determine probable motivations or objectives
 - Developers write bad code
 - Systems improperly configured
 - Poor Access Control
- Identify the Risk
 - Rank risk in a qualitative fashion (High, Med, Low)
 - Give the risk a probable value of occurrence
 - Give the risk a probable value for impact

TM - Mitigation

- Assess the Value
 - Determine the value of the asset or component
- Identify Mitigations that Address the Risk
 - Access Control List, limit user access
 - Input Validation, prevent code/SQL/etc. injection
 - Store Passwords Securely, ensure confidentiality
- Not All Risk is Mitigated
 - Accept Risk
 - Transfer Risk

Software Centric TM

- Also Known As
 - Design-Centric
 - Architecture-Centric
 - System-Centric
- Focuses on each System Component
 - Data-Flow and Control-Flow
 - Process Features
 - Functionality

Attack Centric TM

- Focuses on the Attackers Perspective
 - Goals
 - Motivations
 - Capabilities
- Looks for Attack Vectors in a System
 - Attacks may originate from assets (DATA)
 - Attacks may originate from entry points (INPUTS)
- Look for Vulnerabilities in Elements that Compose the System

Asset-Centric TM

- Focuses Primarily on Assets of a System
 - Stored Personal Information or Inventory
 - Credit Card/ Financial Information
 - User Derived Data or Content
 - ...
- Modeling an Asset
 - What is the asset
 - Who is handling the asset
 - What are trust relationships between handlers
 - How is the asset being used/modified

Conclusions

- Threat Modeling Plays a Significant Role in the Software Security Process
- Threat Modeling Helps Define Design Goals
- Threat Modeling Integrates Security into the Software Architecture, Design, and Configuration

Questions & Sources

- adam.pridgen@thecoverofnight.com
- website: thecoverofnight.com

[1] M. Howard and S. Lipner, *The Security Development Lifecycle*. Redmond: Microsoft Press, 2006.

[2] Threat Model (2009). *Wikipedia* [Online]. Available, http://en.wikipedia.org/wiki/Threat_model.