SECURITY RISK MANAGEMENT BY QUALITATIVE VULNERABILITY ANALYSIS

Golnaz Elahi, Eric Yu, Nicola Zannone

University of Toronto, Canada
Eindhoven University of Technology, Netherlands

September 21st 2011
What is the probability of cracking the media player? 10%? 90%? 67%?

How much is the damage of revealing these credentials? $92k? $609k?
INTRODUCTION AND MOTIVATION

1. Obfuscation
2. Tempter Proofing
3. Physical Token

Risks
Stakeholders’ Goals
Design Objectives
Project Goals
Function and Non-Functional System Requirements

Activation Code
License Check

Illustrative Scenario
CHALLENGES

- It is hard to quantify or measure:
  - The damage and probability of vulnerability exploitations
  - The impacts of security solutions on
    - Requirements achievement
    - Risk

- There is a gap between security requirements engineering and vulnerability analysis:
  - Vulnerabilities are code and architecture-level flaws
  - Evaluating the risk of vulnerabilities is challenging and subjective
  - Tracing the impact of vulnerabilities and solutions on requirements is difficult
RESEARCH OBJECTIVES

- A qualitative and vulnerability-centric risk assessment framework:
  1. Identify requirements in the $i^*$ model
  2. Identify and add vulnerabilities
  3. Add attacks and relate them to vulnerabilities by exploit relations
**RESEARCH OBJECTIVES**

- A **qualitative** and **vulnerability-centric** risk assessment framework:
  1. Identify requirements in the *i* model
  2. Identify and add vulnerabilities
  3. Add attacks and relate them to vulnerabilities by *exploit* relations
  4. Evaluate exploitation probability and damage by vulnerability metrics
  5. Identify critical risks and suggest security solutions
  6. Add countermeasures and analyze their consequences

---

**CVSS Metrics**

<table>
<thead>
<tr>
<th>Sub-Metrics</th>
<th>Possible Values</th>
<th>Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Metrics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access Vector (AV)</td>
<td>High, Medium, High</td>
<td>Base = AV + AC + Au</td>
</tr>
<tr>
<td>Access Complexity (AC)</td>
<td>Low, Medium, High</td>
<td></td>
</tr>
<tr>
<td>Authentication (Au)</td>
<td>Low, Medium, Hard</td>
<td></td>
</tr>
<tr>
<td>Temporal Metrics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exploitability (E)</td>
<td>Low, Medium, Hard</td>
<td>Temporal = E + RL + RD</td>
</tr>
<tr>
<td>Remediation Level (RL)</td>
<td>High, Medium, Low</td>
<td></td>
</tr>
<tr>
<td>Report Confidence (RC)</td>
<td>Low, Medium, High</td>
<td></td>
</tr>
<tr>
<td>Environment Metrics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collateral Damage Potential (CDP)</td>
<td>Low, Medium, High</td>
<td>Environment + CDP + TD</td>
</tr>
<tr>
<td>Target Distribution (TD)</td>
<td>Low, Medium, High</td>
<td></td>
</tr>
<tr>
<td>Damage on Goals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impact of exploitation on goal</td>
<td>Impact(E,P)</td>
<td>Impact = Impact(E,P1) + Impact(E,P2) + ...</td>
</tr>
</tbody>
</table>

---

**Diagram:**

- **Modeling Risks**
  - 1. Identify requirements in the *i* model
  - 2. Identify and add vulnerabilities
  - 3. Add attacks and relate them to vulnerabilities by *exploit* relations
- **Assessing Risks**
  - 1. Identify requirements in the *i* model
  - 2. Identify and add vulnerabilities
  - 3. Add attacks and relate them to vulnerabilities by *exploit* relations
  - 4. Evaluate exploitation probability and damage by vulnerability metrics
  - 5. Identify critical risks and suggest security solutions
  - 6. Add countermeasures and analyze their consequences
**RESEARCH OBJECTIVES**

- A **qualitative** and **vulnerability-centric** risk assessment framework:

  1. Identify requirements in the $i^*\text{ model}$
  2. Identify and add vulnerabilities
  3. Add attacks and relate them to vulnerabilities by exploit relations
  4. Evaluate exploitation probability and damage by vulnerability metrics
  5. Identify critical risks and suggest security solutions
  6. Add countermeasures and analyze their consequences
  7. Develop a consequence table for alternative security solutions
  8. Pick a pair of alternatives
  9. Apply the Even Swaps method
  10. Remove the dominated alternative

**CVSS Metrics**

<table>
<thead>
<tr>
<th>Sub-Metrics</th>
<th>Possible Values</th>
<th>Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Metrics</td>
<td>Access Vector</td>
<td>High Medium Low</td>
</tr>
<tr>
<td></td>
<td>Attack Vector</td>
<td>Low Medium High</td>
</tr>
<tr>
<td></td>
<td>Environment</td>
<td>Low Medium High</td>
</tr>
<tr>
<td></td>
<td>Temporal</td>
<td>Low Medium High</td>
</tr>
<tr>
<td></td>
<td>Exploitability</td>
<td>Low Medium High</td>
</tr>
<tr>
<td></td>
<td>Collateral Damage</td>
<td>Low Medium High</td>
</tr>
<tr>
<td>Damage on Goals</td>
<td>Impact(ETP2)</td>
<td>Impact(ETP1)</td>
</tr>
</tbody>
</table>

**Matrix**

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>Risk 1</th>
<th>Risk 2</th>
<th>Goal 1</th>
<th>Goal 2</th>
<th>Goal m</th>
</tr>
</thead>
<tbody>
<tr>
<td>No security solution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative One</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative Two</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Diagram**

- An alternative is dominated
- The last alternative left is the best solution
- One alternative remains
- Until one alternative is dominated
**AGENDA**

- Identifying and Modeling Vulnerabilities
- Adapting CVSS for Risk Assessment
  - CVSS metrics
  - Metrics aggregation
- Risk Treatment
  - Risk ranking
  - Applying security solution
- Deciding over Alternative Security Solutions
- Discussion, Conclusions and Limitations
IDENTIFYING AND MODELING VULNERABILITIES

- Media provider
- User
- User key
- Listen to audio
- User activation code
- Encrypted media
- Decrypt digital content
- Play DRM content
- Protect DRM content
- Protect player license
- DRM player
- Check the activation code
- Valid activation code
- Hard coded credentials
**Hard coded credentials**

**CWE-798: Use of Hard-coded Credentials**

**Description Summary**

The software contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data.

**Extended Description**

Hard-coded credentials typically create a significant hole that allows an attacker to bypass the authentication that has been configured by the software administrator. This hole might be difficult for the system administrator to detect. Even if detected, it can be difficult to fix, so the administrator may be forced into disabling the product entirely. There are two main variations:

Inbound: the software contains an authentication mechanism that checks the input credentials against a hard-coded set of credentials.

Outbound: the software connects to another system or component, and it contains hard-coded credentials for connecting to that component.

In the Inbound variant, a default administration account is created, and a simple password is hard-coded into the product and associated with that account. This hard-coded password is the same for each installation of the product, and it usually cannot be changed or disabled by system administrators without manually modifying the program, or otherwise patching the software. If the password is ever discovered or published (a common occurrence on the Internet), then anybody with knowledge of this password can access the product. Finally, since all installations of the software will have the same password, even across different organizations, this enables massive attacks such as worms to take place.

The Outbound variant applies to front-end systems that authenticate with a back-end service. The back-end service may require a fixed password which can be easily discovered. The programmer may simply hard-code those back-end credentials into the front-end software. Any user of that program may be able to extract the password. Client-side systems with hard-coded passwords pose even more of a threat, since the extraction of a password from a binary is usually very simple.
IDENTIFYING AND MODELING VULNERABILITIES

- Hard-coded credentials
- V1
- V2
- Cracker
- Decrypt other encrypted digital media
- Extract the player key by static analysis of binary code
- Use DRM player without buying an activation code
- Extract the valid code by static analysis of binary code
- Tamper the binary code to bypass the license check
- Cracker
- Decrypt other encrypted digital media
- Extract the player key by static analysis of binary code
- Use DRM player without buying an activation code
- Extract the valid code by static analysis of binary code
- Tamper the binary code to bypass the license check

Diagram shows relationships and processes involved in DRM content protection and vulnerabilities.
AGENDA

- Identifying and Modeling Vulnerabilities

- Adapting CVSS for Risk Assessment
  - CVSS metrics
  - Metrics aggregation

- Risk Treatment
  - Risk ranking
  - Applying security solution

- Deciding over Alternative Security Solutions

- Discussion, Conclusions and Limitations
### Adapting CVSS for Risk Assessment

<table>
<thead>
<tr>
<th>CVSS Metrics</th>
<th>Sub-Metrics</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base Metrics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access Vector (AV)</td>
<td>Locally (High)</td>
<td>Adjacent Network (Medium)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>General network (Low)</td>
</tr>
<tr>
<td>Access Complexity</td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td>Authentication (Au)</td>
<td>No authentication (Low)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Single authentication (Medium)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multiple authentications (High)</td>
<td></td>
</tr>
<tr>
<td><strong>Temporal Metrics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exploitability (E)</td>
<td>Unproven method (High)</td>
<td>Proof-of-Concept (Medium)</td>
</tr>
<tr>
<td>Remediation Level</td>
<td>Unavailable fix (Low)</td>
<td>Temporary Fix or Workaround approach (Medium)</td>
</tr>
<tr>
<td>(RL)</td>
<td></td>
<td>Official Fix (High)</td>
</tr>
<tr>
<td>Report Confidence</td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td>(RC)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Environment Metrics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collateral Damage</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Potential (CDP)</td>
<td></td>
<td>High</td>
</tr>
<tr>
<td>Target Distribution (TD)</td>
<td>Low percentage</td>
<td>Medium percentage (Medium)</td>
</tr>
<tr>
<td>Impact of exploitation on goal g_x (EXPg_x)</td>
<td>Low</td>
<td>Medium</td>
</tr>
</tbody>
</table>
CVSS Metrics for Untrusted Environment

- In an untrusted environment
  - Attackers own the application and/or hardware
  - Attackers have full access to the application binary
  - Attackers can try to reverse engineer, modify, and redistribute the application binary (cracking)
  - Attackers can try to locate and extract secrets within the application binary (e.g., cryptography keys)

- Some CVSS metrics do not apply to untrusted environment
  - Authentication
  - Access vector
  - Exploitability
## CVSS Metrics for Untrusted Environment

<table>
<thead>
<tr>
<th>CVSS Metrics for Untrusted Environment</th>
<th>Sub-Metrics</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base Metrics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Size</td>
<td>Large (High)</td>
<td>Medium (Medium)</td>
</tr>
<tr>
<td>Binary Code Complexity</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Temporal Metrics</strong></td>
<td></td>
<td>Same as other types of attacks (Table I)</td>
</tr>
<tr>
<td><strong>Environment Metrics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collateral Damage Potential</td>
<td>Same as other types of attacks</td>
<td></td>
</tr>
<tr>
<td>Number of copies distributed</td>
<td>Nearly all revenue space (High)</td>
<td>Considerable revenue space (Medium)</td>
</tr>
<tr>
<td><strong>Damage on Goals</strong></td>
<td>Same as other types of attacks</td>
<td></td>
</tr>
</tbody>
</table>
Adapting CVSS for Risk Assessment: Example

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Value for EXPv₁</th>
<th>Aggregated Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Size</td>
<td>Small (Low)</td>
<td></td>
</tr>
<tr>
<td>Binary Code Complexity</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Exploitability (E)</td>
<td>Functional method (Low)</td>
<td>ℓ = # Low = 2  \ m = # Medium = 3  \ h = # High = 0  \ m &gt; ℓ + h  thus a = Medium  \ Probability = Medium</td>
</tr>
<tr>
<td>Remediation Level (RL)</td>
<td>Workaround (Medium)</td>
<td></td>
</tr>
<tr>
<td>Report Confidence (RC)</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Collateral Damage Potential (CDP)</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Number of copies distributed</td>
<td>Considerable revenue space (Medium)</td>
<td>Max(Medium, Medium, High) = High  \ Damage = High</td>
</tr>
<tr>
<td>Impact of exploitation on goal G₁: Impact(EXPv₁,G₁)</td>
<td>High</td>
<td></td>
</tr>
</tbody>
</table>

- **Rule 1**
  - If \( l \geq h + m \) then \( a = \text{Low} \)
- **Rule 2**
  - If \( h \geq l + m \) then \( a = \text{Med} \)
- **Rule 3**
  - If \( m \geq l + h \) then \( a = \text{High} \)

**Damage** = \( \max(d₁, d₂, d₃) \)
AGENDA

- Identifying and Modeling Vulnerabilities
- Adapting CVSS for Risk Assessment
  - CVSS metrics
  - Metrics aggregation
- Risk Treatment
  - Risk ranking
  - Applying security solution
- Deciding over Alternative Security Solutions
- Discussion, Conclusions and Limitations
SECURITY SOLUTIONS: EXAMPLE

<table>
<thead>
<tr>
<th>Risk of EXP’v2 with Presence of Solutions</th>
<th>Probability</th>
<th>Damage</th>
</tr>
</thead>
<tbody>
<tr>
<td>No security solution</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Code Obfuscation</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Tamper Proofing</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Distribution with Physical Token</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>
AGENDA

- Identifying and Modeling Vulnerabilities
- Adapting CVSS for Risk Assessment
  - CVSS metrics
  - Metrics aggregation
- Risk Treatment
  - Risk ranking
  - Applying security solution
- Deciding over Alternative Security Solutions
- Discussion, Conclusions and Limitations
# Selecting a Security Solution

<table>
<thead>
<tr>
<th>Alternative</th>
<th>Minimize Prob EXP’v2</th>
<th>Minimize Damg EXP’v2</th>
<th>Minimize Prob EXP v1</th>
<th>Minimize Damg EXP v1</th>
<th>Maximize G4 Portable</th>
<th>Minimize G3 Delay</th>
<th>Minimize C1 Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>No security solution</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
<td>✓</td>
<td>50 ms</td>
<td>$0</td>
</tr>
<tr>
<td>Code Obfuscation</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
<td>✓</td>
<td>200 ms</td>
<td>$20K</td>
</tr>
<tr>
<td>Tamper Proofing</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>✓</td>
<td>500 ms</td>
<td>$15K</td>
</tr>
<tr>
<td>Distribution with Physical Token</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>✗</td>
<td>100 ms</td>
<td>$40K</td>
</tr>
</tbody>
</table>
**EVEN SWAPS BASICS**

- Trading one goal for another
- Based on stakeholders’ preferences

*If I buy you a coffee how much longer you stay at this talk? …*

*Depends on your preferences about coffee, and on the context, and the quality of coffee…*
### Example

<table>
<thead>
<tr>
<th>Alternative</th>
<th>Minimize</th>
<th>Maximize</th>
<th>Minimize</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tamper Proofing</strong></td>
<td>Medium</td>
<td>G4</td>
<td>500ms</td>
</tr>
<tr>
<td><strong>Distribution with Physical Token</strong></td>
<td>Low</td>
<td>G3 Portable</td>
<td>$18K</td>
</tr>
</tbody>
</table>

- **Maximize**
  - G4 Portable
  - 500ms
  - $18K

- **Minimize**
  - G3 Delay
  - 100ms
  - $40K

### Tamper Proofing

<table>
<thead>
<tr>
<th>Alternative</th>
<th>Minimize</th>
<th>Maximize</th>
<th>Minimize</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tamper Proofing</strong></td>
<td>Medium</td>
<td>G4</td>
<td>500ms</td>
</tr>
<tr>
<td><strong>Distribution with Physical Token</strong></td>
<td>Low</td>
<td>G3 Portable</td>
<td>$18K</td>
</tr>
</tbody>
</table>

- **Maximize**
  - G4 Portable
  - 500ms
  - $18K

- **Minimize**
  - G3 Delay
  - 100ms
  - $40K
**Discussion, Conclusions and Limitations**

- Evaluating risk require subjective opinion of stakeholders and domain experts
- Using CVSS metrics, risk is broken down into fine-grained vulnerability metrics
  - Instead of directly asking for a high-level risk value for each vulnerability
- More accurate evaluation

**Limitations:**
- Inaccurate and unreliable metrics evaluation
- Reduction of accuracy by aggregating metrics
THANKS... QUESTIONS?

Golnaz Elahi’s contact:

http://www.cs.utoronto.ca/~gelahi/

gelahi@cs.toronto.edu

Eric Yu’s contact:

http://www.cs.utoronto.ca/~eric/

eric@cs.toronto.edu

Nicola Zannone

http://security1.win.tue.nl/~zannone/
n.zannone@tue.nl