A Holistic Approach to Security Attack Modeling and Analysis

Tong Li, Elda Paja, and John Mylopoulos
University of Trento

Jennifer Horkoff
City University London

Kristian Beckers
Technische Universität München

8th International i* Workshop (iStar’15), Ottawa, Canada
August 24th, 2015
Outline

- Background
  - Motivation
    - Holistic security requirements analysis
  - Research outline
  - Challenges
    - Security attack analysis

- Proposal
  - A holistic security attack analysis framework

- Future Work
- Summaries
Motivation

- Socio-Technical Systems (STSs) consist of human, software, and physical infrastructure
- Holistic security analysis
Research outline

- A Holistic security requirements analysis framework

A three-layer security requirements framework [Li2014CAiSE]
Select and apply security patterns [Li2014PoEM, Li2014istar]
Enforce impact of security mechanisms on requirements specification [Li2015REFSQ]

Holistically analyze attacks to STSs
Challenges for attack analysis of STSs

- **Heterogeneous components**
  - A broad scope of attacks

- **Multistage attacks**
  - Difficult to detect

- **Lack of attack knowledge**
Challenges | Solutions
--- | ---
Heterogeneous components | Based on a three-layer requirements framework [Li2014CAiSE]:
• Business processes
• software applications
• physical infrastructure
## Challenges

<table>
<thead>
<tr>
<th>Challenges</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multistage attacks</td>
<td><strong>Takes an attacker’s perspective</strong>: Systematically capture and refine the attacker’s anti-goals</td>
</tr>
</tbody>
</table>

### Diagram

- **AG1**: Tamper energy consumption data
- **AG2**: Tamper energy consumption during data recording
- **AG3**: Defeat auditing protection
- **AG4**: Break physical locks
- **AG5**: Physically hacking the smart meter
- **AG6**: Overwrite smart meter firmware
- **AG7**: Attack the audit function
- **AG8**: Delete audit logs
## Challenges

<table>
<thead>
<tr>
<th>Lack of attack knowledge</th>
<th>Leverage attack patterns:</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPEC (Common Attack Pattern Enumeration and Classification)</td>
<td></td>
</tr>
<tr>
<td>• 463 patterns</td>
<td></td>
</tr>
<tr>
<td>• Broad coverage</td>
<td></td>
</tr>
<tr>
<td>• Detailed specification</td>
<td></td>
</tr>
</tbody>
</table>

## Solutions

### CAPEC-507: Physical Theft

**Summary:** An attacker targets a system that uses JavaScript Object Notation (JSON) as a transport mechanism between the client and the server to steal possibly confidential information transmitted from the server back to the client inside the JSON object by taking advantage of the loophole in the browser's Same Origin Policy that does not prohibit JavaScript from one website to be included and executed in the context of another website.

**Attack Motivation:** Read application data

**Attack Execution Flow:** Understand How to Request JSON Responses from the Target System...

**Attack Prerequisites:** JSON is used as a transport mechanism between the client and the server …

**Typical Severity:** High

**Solutions and Mitigations:** Ensure that server side code can differentiate between legitimate requests and forged requests...

### CAPEC-403: Social Engineering

**Summary:** An attacker targets a system that uses JavaScript Object Notation (JSON) as a transport mechanism between the client and the server to steal possibly confidential information transmitted from the server back to the client inside the JSON object by taking advantage of the loophole in the browser's Same Origin Policy that does not prohibit JavaScript from one website to be included and executed in the context of another website.

**Attack Motivation:** Read application data

**Attack Execution Flow:** Understand How to Request JSON Responses from the Target System...

**Attack Prerequisites:** JSON is used as a transport mechanism between the client and the server …

**Typical Severity:** High

**Solutions and Mitigations:** Ensure that server side code can differentiate between legitimate requests and forged requests...

### CAPEC-111: JSON Hijacking

...
## Proposal

• A holistic security attack analysis framework [Li2015REPoster]

<table>
<thead>
<tr>
<th>Challenges</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterogeneous components</td>
<td><strong>Based on a three-layer requirements framework [Li2014CAiSE]:</strong> Business processes, software applications, physical infrastructure</td>
</tr>
<tr>
<td>Multistage attacks</td>
<td><strong>Takes an attacker’s perspective:</strong> Systematically capture and refine the attacker’s anti-goals</td>
</tr>
<tr>
<td>Lack of attack knowledge</td>
<td><strong>Leverage attack patterns:</strong> CAPEC (Common Attack Pattern Enumeration and Classification)</td>
</tr>
</tbody>
</table>
Recent progress

- A refined attack modeling and analysis process

**Legend**
- User Activity
- Automatic Activity
- Data Object
- Data Store
- Sequence Flow
- Data Flow

**Attack Knowledge Pre-processing**
- CAPEC attack patterns
  - Model attack patterns

**Step 1**
- Identify root anti-goals
  - Three-layer requirements goal models

**Step 2**
- Anti-Goal Refinements
  - Interval-based refinements
  - Asset-based refinements
  - Target-based refinements
  - Protection-based refinements

**Step 3**
- Anti-Goal Operationalization
  - Find relevant attack patterns
  - Context checking
  - Generate alternative attacks
  - Top X critical attacks

**Step 4**
- Risk Assessment
  - Generate security controls

**Step 5**
- Generate security controls
  - A list of security controls

24/08/15
Step 1: Identify root anti-goals

- Structured anti-goals:

  *Threat*: Tampering,  
  *Asset*: Energy demand,  
  *Target*: Energy Supplier,  
  *Interval*: interval(G1)

Part of the 3-layer Goal Model

- SG1 (S)  
  - High Data Integrity  
  - [energy demand, G1]

- G1  
  - Real-time pricing is applied

- G2  
  - Real-time price is obtained

- G3  
  - Customer is notified about the price
Step 2: Anti-goal refinement

- Define four refinement patterns
  - Asset-based refinement
  - Target-based refinement
  - Interval-based refinement
  - Protection-based refinement

Asset-based refinement

Reference Model

Energy consumption data

Water consumption data

Electricity consumption data

Asset: Water consumption data,
Threat: Tampering,
Target: Energy Supplier,
Interval: interval(G1)]

Asset: Electricity consumption data,
Threat: Tampering,
Target: Energy Supplier,
Interval: interval(G1)]

Asset: Energy demand,
Threat: Tampering,
Target: Energy Supplier,
Interval: interval(G1)]

Asset: Electricity consumption data,
Threat: Tampering,
Target: Energy Supplier,
Interval: interval(G1)]

Asset: Water consumption data,
Threat: Tampering,
Target: Energy Supplier,
Interval: interval(G1)]

AG1

AG2

AG3
Step 3: Anti-goal operationalization

- Using CAPEC attack pattern repository
  - Includes 463 attack patterns
  - Example:

<table>
<thead>
<tr>
<th>CAPEC-111: JSON Hijacking</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary</strong>: An attacker targets a system that uses JavaScript Object Notation (JSON) as a transport mechanism between the client and the server to steal possibly confidential information transmitted from the server back to the client inside the JSON object by taking advantage of the loophole in the browser's Same Origin Policy that does not prohibit JavaScript from one website to be included and executed in the context of another website.</td>
</tr>
<tr>
<td><strong>Attack Motivation</strong>: Read application data</td>
</tr>
<tr>
<td><strong>Attack Execution Flow</strong>: Understand How to Request JSON Responses from the Target System...</td>
</tr>
<tr>
<td><strong>Attack Prerequisites</strong>: JSON is used as a transport mechanism between the client and the server ...</td>
</tr>
<tr>
<td><strong>Typical Severity</strong>: High</td>
</tr>
<tr>
<td><strong>Solutions and Mitigations</strong>: Ensure that server side code can differentiate between legitimate requests and forged requests...</td>
</tr>
</tbody>
</table>
Step 3: Anti-goal operationalization

- Model and analyze CAPEC attack patterns
  - Selection step 1: Identify relevant patterns
  - Selection step 2: Identify applicable patterns

CAPEC-111
JSON Hijacking
(Detailed, Complete)

The JSON object returned from the server can be accessed by the attackers’ malicious code via a script tag

Understand how to request JSON response from the target system

Craft a malicious website

Launch JSON hijack

Launch the malicious scripts to request JSON object from the target system

Intercept incoming JSON objects

The target server cannot differentiate real requests from forged requests

CAPEC-111: JSON Hijacking

Summary
Attack Motivation
Attack Prerequisites
Attack Execution Flow

……

CA1: architecture(target_software, Client-Server) & program_language(target_software, AJAX) & use(target_software, JSON)

Lure the victim to visit the malicious website to activate the malicious script

The target server cannot differentiate real requests from forged requests

The JSON object returned from the server can be accessed by the attackers’ malicious code via a script tag

Lure the victim to visit the malicious website to activate the malicious script

Threat: Information Disclosure
Target: software

CA1: architecture(target_software, Client-Server) & program_language(target_software, AJAX) & use(target_software, JSON)
Attack assessments and treatments

- **Step 4: Risk assessments**
  - Analyze severity and likelihood of each attack (CAPEC)

- **Step 5: Attack treatments**
  - Prioritize attacks
  - Design security controls (CAPEC)
Future work

1) Method

(Step 1) Identify root anti-goals

(Step 2) Anti-Goal Refinements

- Interval-based refinements
- Asset-based refinements
- Target-based refinements
- Protection-based refinements

(Step 3) Attack Knowledge Pre-processing

- CAPEC attack patterns

(Step 4) Three-layer requirements goal models

(Step 5) Risk Assessment

Top X critical attacks

2) Modeling

3) Tool

Legend

Data Store

Data Object

Data Flow

Sequence Flow

User Activity

Automatic Activity

Future work

24/08/15
Summaries

- Holistically analyze security of STSs
  - Ongoing work: Identify potential attacks to test system security

- Propose a holistic security attack analysis framework

- Present and illustrate a refined process and discuss subsequent research objectives
Thank You!

Contact: tong.li@disi.unitn.it