Risk and Argument: A Risk-based Argumentation Method for Practical Security

Virginia N. L. Franqueira, Thein Than Tun, Yijun Yu, Roel Wieringa and Bashar Nuseibeh

Trento - 02 September 2011
Agenda

- Problem
- Solution
- Walk-through of RISA method using PED example
Engineering of secure systems is bound by practical limitations.

- **Limited resources**
- **Incomplete information**
- **Uncertainties**
- **Unnoticeable, hardly measurable evidence**
Consequence & Solution Direction

**In practice:**
- No absolute security
- No 100% security requirements satisfaction

**Way forward:**
- Good enough security satisfaction
- As low as possible level of security risk

**Our solution:**

RISA – Risk assessment in Security Argumentation
From Haley et al. framework to RISA method

Key steps from Haley et al. framework

1. Identify Functional Requirements
2. Identify Security Goals
3. Identify Security Requirements
4. Construct Outer Argument
5. Identify Risks
6. Classify Risks
7. Mitigate Risks
8. Prioritize Risks

Steps for risk assessment in RISA

Structured Argumentation

Public Security Catalogues

Formal logic

good enough security
PIN Entry Devices (PED) example

RISA step 1: Identify Functional Requirements

- **functional goal**
  Provide convenient payment option at Points-Of-Sale to consumers

- **functional requirement**
  Allow consumers to pay at Points-Of-Sale with PIN
RISA step 2: Identify Security Goals

- valuable assets
  - PIN
  - card details
  - transaction value
  - design characteristics
  - smartcard itself
  - cryptographic keys

- security goal

Protect the PIN
RISA step 3: Identify Security Requirements

- **security requirements**
  - confidentiality of PIN
  - integrity of PIN

- **security functions**
  - enclosure of PED components provides *tamper detection & response* mechanisms to resist physical attacks
  - *encryption/decryption of PIN* ensures that the PIN is encrypted within the PED immediately after PIN entry

- **system context diagram**
RISA step 4: Construct Outer Argument

Formal proof that behavior of PIN related to confidentiality is satisfiable

Outer argument for confidentiality of PIN:

(Behavioral premises) P1, P2, P3, P4, P5, P6
- confirmation-transaction
Risk assessment steps of RISA are supported by the CAPEC & CWE public catalogues

http://capec.mitre.org/
http://cwe.mitre.org/
RISA step 5: Identify Risks

Structured argumentation used to challenge behavioral premises in practice via risk assessment

<table>
<thead>
<tr>
<th>Challenged</th>
<th>Risk</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Premise P2</td>
<td>R1.6: PIN is revealed if sent unencrypted within the PED and the PED enclosure can be tampered</td>
<td>CWE-311 &amp; CAPEC-436</td>
</tr>
</tbody>
</table>
CWE-311: Missing Encryption of Sensitive Data

Description Summary
The software does not encrypt sensitive or critical information before storage or transmission.

Extended Description
The lack of proper data encryption passes up the guarantees of confidentiality, integrity, and accountability that properly implemented encryption conveys.

Time of Introduction
- Architecture and Design
- Operation

Applicable Platforms

Languages
Language-independent

Common Consequences

Scope  Effect
Confidentiality  Technical Impact: Read application data

If the application does not use a secure channel, such as SSL, to exchange sensitive information, it is possible for...
RISA step 6: Classify Risks

Risks are classified in terms of:

- **risks transferred to system context**
- **risks to be mitigated by the system**

Possibilities:

- risk to be completely mitigated by the PED
- risk to be completely mitigated by the PED context
- risk to be partially mitigated by both
RISA step 7: Mitigate Risks

Mitigations restore the satisfaction of security requirements by rebutting risks.

<table>
<thead>
<tr>
<th>Risk</th>
<th>Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1.6 &amp; R1.7 &amp;</td>
<td>M2.4: Any transmission of PIN should use well-vetted encryption algorithms &amp;</td>
</tr>
<tr>
<td>R1.8</td>
<td>recommended key sizes</td>
</tr>
</tbody>
</table>
CWE-311: Missing Encryption of Sensitive Data (2.0) - Mozilla Firefox

Potential Mitigations

Phase: Requirements
Clearly specify which data or resources are valuable enough that they should be protected by encryption. Require that any transmission or storage of this data/resource should use well-vetted encryption algorithms.

Phase: Architecture and Design

Strategy: Threat Modeling
Using threat modeling or other techniques, assume that your data can be compromised through a separate vulnerability or weakness, and determine where encryption will be most effective. Ensure that data you believe should be private is not being inadvertently exposed using weaknesses such as insecure permissions (CWE-732).

Phase: Architecture and Design

Ensure that encryption is properly integrated into the system design, including but not necessarily limited to:

- Encryption that is needed to store or transmit private data of the users of the system
- Encryption that is needed to protect the system itself from unauthorized disclosure or tampering

Identify the separate needs and contexts for encryption:

- One-way (i.e., only the user or recipient needs to have the key). This can be achieved using public key cryptography, or other techniques in which the encrypting party (i.e., the software) does not need to have access to a private key.
- Two-way (i.e., the encryption can be automatically performed on behalf of a user, but the key must be available so that the plaintext can be automatically recoverable by that user). This requires storage of the private key in a format that is recoverable only by the user (or perhaps by the operating system) in a way that cannot be recovered by others.

Phase: Architecture and Design

Strategy: Libraries or Frameworks
Select a well-vetted algorithm that is currently considered to be strong by experts in the field, and select well-tested implementations. As with all cryptographic mechanisms, the source code should be available for analysis.
RISA step 8: Prioritize Risks

Empirical data about typical severity of risks or likelihood of exploit can also be found in the catalogues.

<table>
<thead>
<tr>
<th>Risk</th>
<th>Mitigation</th>
<th>Typical risk severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1.6 &amp; R1.7 &amp; R1.8</td>
<td>M2.4: Any transmission of PIN should use well-vetted encryption algorithms &amp; recommended key sizes</td>
<td>Low to very high</td>
</tr>
</tbody>
</table>

Interpretation of risk severity depends on many factors.
CWE-311: Missing Encryption of Sensitive Data (2.0)

Languages
Language-independent

Common Consequences

Scope
Confidentiality

Technical Impact: Read application data
If the application does not use a secure channel, such as SSL, to exchange sensitive information, it is possible for an attacker with access to the network traffic to sniff packets from the connection and uncover the data. This attack is not technically difficult, but does require physical access to some portion of the network over which the sensitive data travels. This access is usually somewhere near where the user is connected to the network (such as a colleague on the company network) but can be anywhere along the path from the user to the end server.

Confidentiality

Technical Impact: Modify application data
Omitting the use of encryption in any program which transfers data over a network of any kind should be considered on par with delivering the data sent to each user on the local networks of both the sender and receiver. Worse, this omission allows for the injection of data into a stream of communication between two parties -- with no means for the victims to separate valid data from invalid. In this day of widespread network attacks and password collection sniffers, it is an unnecessary risk to omit encryption from the design of any system which might benefit from it.

Likelihood of Exploit
High to Very High

Detection Methods

Manual Analysis
The characterization of sensitive data often requires domain-specific understanding, so manual methods are useful. However, manual efforts might not achieve desired code coverage within limited time constraints. Black box methods may produce artifacts (e.g. stored data or unencrypted network transfer) that require manual evaluation.

Effectiveness: High

Automated Analysis
Automated measurement of the entropy of an input/output source may indicate the use or lack of encryption, but human
RISA recursion

- Other rounds of argumentation may follow
- recursion stops when the system security is considered good-enough and/or resources for analysis of security have been used
Opportunities for future work

- Addressing residual risks
- Improvement to estimation/prioritization of risks
- Tool support for the method: OpenArgue tool to be adapted
- Validation in the field with industrial case studies
- Addressing impact of transferred risks in terms of system mitigations
- Support to search catalogues for risk identification
Conclusion: Mutual benefits

- **Satisfaction analysis (SA) benefits from risk assessment (RA)**
  - RA provides systematic input for security argumentation in SA
  - RA allows prioritization of arguments and security requirements from prioritization of risks
  - RA scales the process of argumentation with breadth-first approach

- **Risk assessment (RA) benefits from satisfaction analysis (SA)**
  - SA provides systematic description of system context: source of risks
  - SA provides top structure for RA
  - SA argumentation organizes several rounds of RA & facilitates traceability
Questions?

franqueirav@ewi.utwente.nl
http://wwwwhome.ewi.utwente.nl/~franqueirav/

virginia.franqueira@gmail.com