A Software Security Risk Classification System

Eric Dalci
Sr. Security Consultant
edalci@cigital.com

Robert Hines
Managing Consultant
rhines@cigital.com

Metricon 2.0 - August 2007
Agenda

- Goals and purpose of RCS
- Context
- Risk Evaluation
  - Security Metrics/Factors
- Risk Classification
- Preliminary Results
- Conclusion
First, an analogy…

The Indian village needs to fortify its huts. Where to start first?

Image Source: http://pelotes.jea.com/NativeAmerican/LeMoyne/FORTIFIED%20TOWNS.GIF
Goals and purpose of RCS

- Estimation practice of application’s potential risk (system’s insecurity) with respect to other systems in the portfolio, quickly and with nominal level of effort.

- Determination of what SLDC actions to require for systems with a given risk profile
Outcome of RCS

- Prioritization of application portfolio
  - Segregate different risk profile (High, Medium, Low)

- Portfolio Risk Evaluation
  - Identify weaknesses across portfolio

- Applicable Risk Mitigation
  - Depending on the risk profile and Lifecycle stage apply set of mitigation practices.
Cigital Risk Management Framework

1. Understand the Business Context

2. Identify the Business Risks
   - Artifact Analysis
   - Business Context
   - Identify the Technical Risks
   - Artifact Analysis

3. Validation Loop

4. Synthesize & Prioritize the Risks

5. Define the Risk Mitigation Strategy

6. Fix the Artifacts

7. Validate the Artifacts

RCS

Measure & Report

Initiate Process Improvement
<table>
<thead>
<tr>
<th>Categories of Risk</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business Risk</strong> – Risks Inflicted upon the System by External Parties</td>
<td></td>
</tr>
<tr>
<td>Market/User</td>
<td>Issues with the desires, requirements and satisfaction of the end users of the system</td>
</tr>
<tr>
<td>Resource (availability &amp; capability)</td>
<td>Issues with Staff, Capabilities, Budget, etc.</td>
</tr>
<tr>
<td><strong>Technical Risk</strong> – Risks Experienced as a Result of Direct System Activities</td>
<td></td>
</tr>
<tr>
<td>Architecture &amp; Design</td>
<td>Issues with the system architecture and design</td>
</tr>
<tr>
<td>Implementation</td>
<td>Issues with the technology stack used to implement the system</td>
</tr>
<tr>
<td>Quality</td>
<td>Issues with the accuracy, reliability and predictability of the system</td>
</tr>
<tr>
<td>Security</td>
<td>Issues with the confidentiality, integrity and availability of the system and its data</td>
</tr>
<tr>
<td>Operations &amp; Maintenance</td>
<td>Issues with the operation and maintenance of the deployed system</td>
</tr>
</tbody>
</table>
Factors

- **Business Risk**
  - Corollary impacts
  - **Data Sensitivity**
  - Sunk Level of Effort
  - Production Failure
  - User Count
  - User Domain

- **Technical Risk**
  - Third party COTS/OSS
  - Code Size
  - Defect Density
  - Web Vulnerability Results
  - Static Analysis Tools Results
  - Competency in Technology

---

### Data Sensitivity

<table>
<thead>
<tr>
<th>Data Sensitivity</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
<td>1</td>
</tr>
<tr>
<td>Internal Use Only</td>
<td>2</td>
</tr>
<tr>
<td>Confidential</td>
<td>3</td>
</tr>
<tr>
<td>Confidential restricted</td>
<td>4</td>
</tr>
</tbody>
</table>

### Number of Users

<table>
<thead>
<tr>
<th>Number of Users</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>n/a</td>
<td>0</td>
</tr>
<tr>
<td>&lt; 50 – Department</td>
<td>1</td>
</tr>
<tr>
<td>&lt; 500 – Business unit</td>
<td>2</td>
</tr>
<tr>
<td>&lt; 10,000 – Company wide</td>
<td>3</td>
</tr>
<tr>
<td>&gt; 10,000 – General public</td>
<td>4</td>
</tr>
</tbody>
</table>
Measuring the COTS/OSS Factor

Old Release:
- Patch not applied or not applicable
- Old release has known vulnerabilities

Previous Release (older than “accepted release”):
- Critical/Sensitive Patches are applied infrequently (as needed) and/or with a considerable time delay
- Latest patch may not be applicable without upgrade

Recent and Mature release (“accepted release”):
- Critical/Sensitive Patches applied systematically if Security risk involved.
- Patches managed by Patch management system

Premium Support:
- Proactive Vendor,
- SLA,
- 24 hours/support
- On Site support
- CSS with very large User Community

Regular Support:
- Business hours,
- Mature OSS with large User Community

No Support:
- OSS with small or none Existing User Community
- OSS with very infrequent update release
- Vendor does not exist anymore

Low Risk functionality implemented:
- Logging
- Reporting
- Scheduler

Medium Risk functionality implemented:
- Storage
- Configuration Management

High Risk functionality implemented:
- Authentication/Authorization
- Access Control
- Session Management
- Data Validation
- Encryption
- User Interface

© 2007 Cigital Inc. All Rights Reserved
Factors that we dropped

- Cyclomatic complexity
  - Code basis heterogeneous (.NET, Java, C, etc.)

- Process related metrics
  - Organization is not using consistent security processes across projects.

- Other Factors which would return subjective answers or expensive to collect.

- Poor results with “Competency in Technology”
Portfolio ranking

Analysis
• Portfolio Risk Distribution
• Standard Deviation
• Correlation Matrix

System Inputs
• Questionnaire,
• Tools,
• Defect tracking system,
• etc.

Calibration
• Weights
• Scale
• Pairwise Comparison

Feedback loop
Portfolio segregation

- Which Systems had **high score**?
  - Web facing Systems
  - Large code size applications
  - Complex applications
  - New applications (No DR, new Technology, etc.)

- Which Systems had **low score**?
  - Low user count and/or Internal applications
  - Low corollary impacts (downstream impacts)
  - Small code size applications
## Calibration (Weight Systems)

<table>
<thead>
<tr>
<th>Measure</th>
<th>Weight</th>
<th>Correlation with aggregated score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corollary Impacts</td>
<td>1.5</td>
<td>0.39</td>
</tr>
<tr>
<td>Data Sensitivity</td>
<td>2</td>
<td>0.07</td>
</tr>
<tr>
<td>Sunk Level of Effort</td>
<td>0.25</td>
<td>0.35</td>
</tr>
<tr>
<td>Production Failure</td>
<td>0.5</td>
<td>0.11</td>
</tr>
<tr>
<td>User Domain</td>
<td>1</td>
<td>0.36</td>
</tr>
<tr>
<td>User Count</td>
<td>1</td>
<td>0.49</td>
</tr>
<tr>
<td><strong>Total Business Risk</strong></td>
<td>6.25</td>
<td>0.58</td>
</tr>
<tr>
<td>Competency in technology</td>
<td>1.5</td>
<td>0.19</td>
</tr>
<tr>
<td>Third party COTS/OSS</td>
<td>1</td>
<td>0.29</td>
</tr>
<tr>
<td>Code Size</td>
<td>0.75</td>
<td>0.60</td>
</tr>
<tr>
<td>Defect Density</td>
<td>1</td>
<td>0.27</td>
</tr>
<tr>
<td>Web Vulnerability Results</td>
<td>1.25</td>
<td>0.28</td>
</tr>
<tr>
<td>Static Analysis Tool Results</td>
<td>1</td>
<td>0.60</td>
</tr>
<tr>
<td>Contingency plan</td>
<td>1.5</td>
<td>0.41</td>
</tr>
<tr>
<td><strong>Total Technical Risk</strong></td>
<td>8</td>
<td>0.73</td>
</tr>
</tbody>
</table>
Conclusion

- Heuristic approach
- Preliminary results reflect expert’s opinion
- Calibration specific to your organization
Questions?