Toward Building and Validating a Secure Software Development Self-Efficacy Scale

Desiree Abrokwa, Daniel Votipka, and Michelle L. Mazurek

**Goal**
- Build and validate a Secure Software Development Self-Efficacy survey to measure a developer’s confidence in completing secure development tasks

**Survey**
- Each of the 58 tasks created fit into one of these categories

**Survey Results**

<table>
<thead>
<tr>
<th>Task</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>I can perform a threat risk analysis (e.g., identified vulnerabilities and impact of exploitation)</td>
<td>3.71</td>
<td>1.17</td>
</tr>
<tr>
<td>I can determine security threats for the system</td>
<td>3.81</td>
<td>0.89</td>
</tr>
<tr>
<td>I can identify potential security threats for the system</td>
<td>3.82</td>
<td>0.87</td>
</tr>
<tr>
<td>I can correctly implement authentication protocols</td>
<td>3.82</td>
<td>0.87</td>
</tr>
<tr>
<td>I can correctly implement authorization protocols</td>
<td>3.82</td>
<td>0.87</td>
</tr>
<tr>
<td>I can communicate system details with other developers to ensure a thorough security code review</td>
<td>3.82</td>
<td>0.87</td>
</tr>
<tr>
<td>I can discuss lessons learned from internal and external security incidents to ensure all development team members are aware of potential threats</td>
<td>3.73</td>
<td>1.12</td>
</tr>
<tr>
<td>I can effectively communicate to company leadership identified security issues with new hardware and software technologies</td>
<td>3.73</td>
<td>1.12</td>
</tr>
<tr>
<td>I can determine the appropriate point of contact/response team for a vulnerability</td>
<td>3.73</td>
<td>1.12</td>
</tr>
</tbody>
</table>

**Motivation**
- Prior security education measures can be time consuming/noisy and are difficult to measure their effectiveness

**Item Generation**

**Frameworks Used**
- NIST’s NICE Framework
- Building Security in Maturity Model
- Open Web Application Security Project
- Software Assurance Forum for Excellence in Code
- Fundamental Practices for Secure Software Development

**Testing**
- Enumerate boundary conditions and mimic potential threats
- Assess that security requirements are met (e.g., through security design and code reviews)

**Communicating Security**
- Communicate system details with other developers to ensure a thorough security review of the code
- Maintain awareness of security issues with new hardware and software technologies and their potential implications

**Expert Review**
- Surveyed 22 security professionals with an average of 20.5 years of secure development experience on the applicability of our scale and tasks

**Future Work**
- Recruit 120 more subjects to complete the survey and validate the underlying scale factor structure
- Compare to other psychometric measures
- Use as a before and after metric for future studies of intervention effectiveness

**University of Maryland, Maryland Cybersecurity Center  sec-professionals.cs.umd.edu**