Agile and Secure
Can we do both?

Jerry Hoff
Antisamy .NET project lead
Aspect Security
jerry.hoff@aspectsecurity.com

OWASP
Oct 30, 2009
Quick Security Overview

http://example.com/search?q=Alessandra+Ambrosio

<html>
<body>
You searched for Alessandra Ambrosio
  <li>... </li>
</body>
</html>
Quick Security Overview: XSS

http://example.com/search?q=\<script\>/*evil*/\</script\>

<html>
<body>
You searched for \<script\>/*evil*/\</script\>
  \<li\>...\</li\>
</body>
</html>
Quick Security Overview

<img src="http://mail.example.com/logo.gif"/>
Quick Security Overview: CSRF

<img src="http://mail.example.com/logo.gif">

<img src="http://mail.example.com/deleteAllMsgs?confirm=true">
Quick Security Overview

http://example.com/viewStatement?custid=123153
Quick Security Overview: Access Control

http://example.com/viewStatement?custid=123154

SELECT * FROM statements WHERE CustomerID=123154
Quick Security Overview: SQL Injection

http://example.com/viewStatement?custid=1;
DROP TABLE statements;

SELECT * FROM statements WHERE CustomerID=1;
DROP TABLE statements;
Agenda

- About Us
- Waterfall Process Background
- Agile Process Background
- Leveraging Agile Characteristics
- Accounting for Agile Traits
- Putting It All Together
Traditional Waterfall Process

- Requirements
- Design
- Implementation
- Verification
- Maintenance
## Security in the Waterfall Process

<table>
<thead>
<tr>
<th>Stage</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>• Security Requirements</td>
</tr>
<tr>
<td>Design</td>
<td>• Security Architecture Review</td>
</tr>
<tr>
<td>Implementation</td>
<td>• Secure Code Review</td>
</tr>
<tr>
<td>Verification</td>
<td>• Application Vulnerability Testing</td>
</tr>
<tr>
<td>Maintenance</td>
<td>• External Application Security Testing</td>
</tr>
</tbody>
</table>

### Advantages:
- Well understood process
- Leverages subject matter experts to identify security concerns

### Disadvantages:
- Findings from early security reviews are often ignored as “theoretical”
- Costly to go backwards in the development timeline
Agile Process

1. Begin Iteration # N
2. Choose User Stories
3. Implement Stories
4. Unit Testing
5. Deploy

 OWASP Brazil 2009
Traditional Security + Agile Process?

1. **Begin Iteration # N**
2. **Gather Security Requirements**
3. **Choose User Stories**
   - **Security Code Review**
   - **Implement Stories**
     - **Security Architecture Review**
     - **Perform Unit Testing**
     - **Application Vulnerability Testing**
   - **Deploy**
     - **External Application Security Testing**
Traditional Security + Agile Process?
Leverage User Stories

- User stories useful for access control, encryption, logging, and several other security areas

- Some technical risks need extra consideration to be represented by user stories
  - XSS
  - CSRF
Creating User Security Stories

Threat Agent: Attack

Vulnerability: Vulnerability, Vulnerability, Vulnerability

Control: Control, Control

Technical Impact: Asset, Asset, Function

Business Impact: Business Impact, Business Impact, Business Impact
Require Security Training

Attacks continuously evolve
  - Developers must understand the attacks and controls to properly mitigate the threats

Agile developers write their own tests
  - Must test security adequately

Ultimately, everyone on the team responsible for security
  - Therefore, all developers should have a background in web application security
Leverage Unit Testing

Continuous testing done by all team members
- Unit tests should include security mechanisms
- Integrate peer code reviews

Check for common security flaws
- Test input validation by verifying behavior in edge cases
- Test access control by verifying behavior from multiple roles
Use Standard Security Controls

OWASP Enterprise Security API (ESAPI)
http://www.owasp.org/index.php/ESAPI

Custom Enterprise Web Application

Enterprise Security API

Existing Enterprise Security Services/Libraries
Leverage Sprints

Sprint # N

- User Profile Story
- Password Security Story
- User Login Story
Putting It All Together

- **Create Threat Model**
  - Capture key threats to the application

- **Define Security Stories**
  - Encapsulate threat model in user stories

- **Create Unit Security Tests**
  - Test edge cases for inputs
  - Verify use of security controls

- **Consolidate Sprints**
  - Combine related security stories
## Putting It All Together

| Use Standard Security Controls | • Developers should use standard controls  
  • See the OWASP ESAPI Project |
| Secure Coding Standards        | • Avoid patterns that lead to security flaws  
  • How to use security controls correctly |
| Provide Security Training      | • Developers need application security awareness  
  • Train developers to use your controls |
| Leverage Security Experts      | • Even with training and standard, security is hard |
References

▶ Integrating Application Security into Agile Methodologies
  ▪ Aspect Security

▶ Beyond Functional Requirements On Agile Projects
  ▪ Scott W. Ambler - September 16, 2008
    http://www.ddj.com/security/210601918

▶ Agile Security Requirements Engineering
  ▪ Johan Peters
Questions?

Aspect Security

http://www.aspectsecurity.com

Jerry Hoff

jerry.hoff@aspectsecurity.com