Microsoft SDL For Agile Released

Agenda

- Presenter Background
- Microsoft SDL (High-level)
- Agile Development (High Level)
- Traditional SDL Activities and Pain points for Agile development
- SDL-Agile Key Concepts
- SDL-Agile in Detail
- Tips for Making SDL-Agile Manageable
Bio

- AT&T Consulting:
  - Application Security
    - Penetration testing
    - Code review
    - Architecture and design reviews
  - Application security program development
  - Secure development methodology improvement

Projects and Publications

- ISSA Journal: Web Application Security Portfolios
- SAMM Interview Template
- Turn Application Assessment Reports into Training Classes
- Arshan Dabirsiaghi's Struts 2 Gap Analysis Whitepaper
- Observed Secure Software Development Stages
- Vulnerability Tracking, Workflow, and Metrics with Redmine
- Using Microsoft's AntiXSS Library 3.1
- Internal AppSec Portals
- Struts 2 Security Addons Code Repository
- JITSecure Code - light_weight_code_review_as_you_program
Microsoft Security Development Lifecycle (SDL)

Components:
- Best Practices
- Processes
- Standards
- Security Activities
- Tools

Goal:
“minimize security-related vulnerabilities in the design, code, and documentation and to detect and eliminate vulnerabilities as early as possible in the development life cycle.”
Which Software?

SDL applies to software that:

- Is used in Business environments
- Stores or transmits PII
- Communicates over the Internet or other networks

Source: Microsoft’s Product Website
SDL Principles and Process

**SD3+C**
- Secure by Design
- Secure by Default
- Secure in Deployment
- Communications

**PD3+C**
- Privacy by Design
- Privacy by Default
- Privacy in Deployment
- Communications
Agile Development

- Cross-functional, self-organizing teams
- Short, time-boxed development iterations
- Delivery of small functional stories
- No extensive up front design or documentation

Source: http://www.scrumalliance.org/pages/what_is_scrum
User Stories and Documentation

http://www.flickr.com/photos/fmcamargo
Traditional SDL Pain Points for Agile

- Can’t complete all SDL activities for each sprint
- Requirements, architecture, and design evolves over time
- Threat model becomes dated quickly
- Data sensitivity and connection to third parties may not be immediately known
Microsoft SDL For Agile Development

- SDL Requirement Categories:
  - Every-Sprint
  - Bucket
    - Verification Tasks
    - Design Review Tasks
    - Response Planning Tasks
  - One-Time

Source: Microsoft SDL v4.1a
## Appendix Q: SDL-Agile Bucket Requirements

### Bucket A: Security Verification

<table>
<thead>
<tr>
<th>Title</th>
<th>Requirement/Recommendation</th>
<th>Applies to Online Services</th>
<th>Applies to Managed Code</th>
<th>Applies to Native Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug the application with the Application Verifier enabled</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Disable tracing and debugging in ASP.NET applications</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Investigate and service any reported bugs that crash in GS</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Perform ActiveX control fuzzing</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Perform attack surface analysis</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Perform binary analysis (BinScope)</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Perform COM object testing</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Perform cross-domain scripting testing</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Perform file fuzz testing</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Perform RPC fuzz testing</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Conduct in-depth manual and automated code review for high-risk code</td>
<td>Recommendation</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
# Appendix R: SDL-Agile One-Time Requirements

<table>
<thead>
<tr>
<th>Title</th>
<th>Requirement/Recommendation</th>
<th>Completion Deadline (months)</th>
<th>Applies to Online Services</th>
<th>Applies to Managed Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avoid writable PE segments</td>
<td>Requirement</td>
<td>6</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Configure bug tracking to track the cause and effect of security vulnerabilities</td>
<td>Requirement</td>
<td>3</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create a baseline threat model</td>
<td>Requirement</td>
<td>3</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Determine security response standards</td>
<td>Requirement</td>
<td>6</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Establish a security response plan</td>
<td>Requirement</td>
<td>6</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Identify primary</td>
<td>Requirement</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Every-Sprint SDL Requirements

“...so essential to security that no software should ever be released without these requirements being met.”

Examples:

- Update the threat model
- Communicate privacy-impacting design changes to the team’s privacy advisor
- Fix all issues identified by code analysis tools for unmanaged code
- Follow input validation and output encoding guidelines to defend against cross-site scripting attacks
Bucket SDL Requirements

Examples:

- Teams prioritize the pool of tasks over many sprints
- Each sprint, one task from each bucket completed
- Each tasks must be completed at least every 6 months

- Security Verification Tasks
  - Run fuzzing tools
  - Manual and automated code review

- Design Review Tasks
  - Conduct privacy review
  - In-depth threat model

- Response Planning Tasks
  - Define security/privacy bug bar
  - Create support documents
One-Time Requirements

Why?

- Repetition not necessary
- Must occur at the beginning of the project
- Not possible at the beginning of the project

Examples:

- Configure bug tracking system (3 months)
- Identify security/privacy experts (1 month)
- Baseline threat model (3 months)
- Establish a security response plan (6 months)
Final Security Review

- Occurs at the end of every sprint
- Checklist:
  - ✓ All every-sprint requirements have been completed
  - ✓ No one-time requirements have exceeded deadline
  - ✓ At least one requirement from each bucket category has been completed
  - ✓ No bucket requirements exceed the six month deadline
  - ✓ No security or privacy bugs are open that exceed the severity threshold
## SDL-Agile Process Demonstration

### Backlog
- User Story
- User Story

### One-Time
- One-Time

### Verification
- Verif.

### Design
- Design

### Resp. Plan
- Resp. Plan

### Sprint 1
**In Progress**
- QA
- Done

- **Every Sprint**
  - Every Sprint
  - ry
  - nt
  - ry
  - nt
  - ry
  - nt

### Sprint 2
**In Progress**
- QA
- Done

- **Every Sprint**
  - Every Sprint
  - ry
  - nt
  - ry
  - nt
  - ry
  - nt

### Sprint 3
**In Progress**
- QA
- Done

- **Every Sprint**
  - Every Sprint
  - ry
  - nt
  - ry
  - nt
  - ry
  - nt

### User Story
- One-Time
- One-Time

### One-Time Verification
- Verif.
### SDL-Agile Process Demonstration

#### Backlog

<table>
<thead>
<tr>
<th>User Story</th>
<th>One-Time</th>
<th>Verification</th>
<th>Design</th>
<th>Resp. Plan</th>
</tr>
</thead>
</table>

#### Sprint 1

<table>
<thead>
<tr>
<th>In Progress</th>
<th>QA</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Story</td>
<td>Verif.</td>
<td>Design</td>
</tr>
<tr>
<td>One-Time</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Final Security Review

- ✓ ✓ ✓ ✓ ✓

#### Sprint 2

<table>
<thead>
<tr>
<th>In Progress</th>
<th>QA</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Story</td>
<td>Verif.</td>
<td>Design</td>
</tr>
<tr>
<td>One-Time</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Final Security Review

- ✓ ✓ ✓ ✓ ✓

#### Sprint 3

<table>
<thead>
<tr>
<th>In Progress</th>
<th>QA</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Story</td>
<td>Verif.</td>
<td>Design</td>
</tr>
<tr>
<td>One-Time</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Final Security Review

- ✓ ✓ ✓ ✓ ✓
Making SDL-Agile Manageable

- Automation
  - CI server, unit testing that include security
- Tooling
  - Automated code analysis and pen testing tools
- Continuous updates to the threat model
- Documented standards
- Security training
- Light on security artifacts
TeamCity SDL-Agile Demonstration

- TeamCity Continuous Integration Server
  - Continuous Build
    - FxCop
    - CAT.NET
    - BinScope
    - Commercial Penetration Testing or Code Review Tools
    - Security Unit Testing
Microsoft releases SDL-Agile Guidance in Nov. 2009

• Treats SDL Activities like team-prioritized User Stories
  • 3 Categories: One-time, Every-time, and Bucket

• Increased success with the implementation of training, automation, tools, and standards

More Information:
http://www.microsoft.com/sdl

Nick Coblentz, CISSP
Senior Consultant, AT&T Consulting
Nick.Coblentz@gmail.com
http://nickcoblentz.blogspot.com
http://www.twitter.com/sekhmetn