Agile yes, but secure?

OWASP Meeting Stuttgart

3.8.2015
About me

Andreas Falk
NovaTec Consulting GmbH
andreas.falk@novatec-gmbh.de

@NT_AQE
@Agile_Security

8/3/2015
Chapter Meeting Stuttgart
Who’s next to be hacked?

Hacking Team -- purveyor of exploits and spyware to a variety of government agencies all over the world -- has been hacked. Late Sunday night, its Twitter account name was changed to "Hacked Team" and its bio to read:
Who’s next to be hacked?

Large caches of data stolen from online cheating site AshleyMadison posted online by an individual or group that company’s user databases, financial records unfolding leak could be quite damaging to whose slogan is “Life is short. Have an affair.”

Adult FriendFinder
@adultfriendfind

We recently became aware of a potential data security issue. Protecting our members' info remains our top priority bit.ly/1PB3yox

8:53 PM - 22 May 2015

OWASP
Open Web Application Security Project
Who's next to be hacked?

OutFrontCNN @OutFrontCNN · 7 Std.
Researcher who hacked Jeep Cherokee: "We're the good guys"
 cnnmon.ie/1Vx5BL4

CNN

Hacking America

FBI: Computer expert briefly made plane fly sideways

Elizabeth Weise
Sunday, 17 May 2015 | 11:45 AM ET

USA Today

Researchers who hacked Jeep Cherokee: "We're the good guys"
See more at cnn.com

8/3/2015  Chapter Meeting Stuttgart
Are firewalls really sufficient?

http://owasp.org

8/3/2015

Chapter Meeting Stuttgart
### Web Application Security: OWASP Top 10

<table>
<thead>
<tr>
<th></th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Injection</td>
</tr>
<tr>
<td>A2</td>
<td>Broken Authentication and Session Management</td>
</tr>
<tr>
<td>A3</td>
<td>Cross-Site Scripting (XSS)</td>
</tr>
<tr>
<td>A4</td>
<td>Insecure Direct Object References</td>
</tr>
<tr>
<td>A5</td>
<td>Security Misconfiguration</td>
</tr>
<tr>
<td>A6</td>
<td>Sensitive Data Exposure</td>
</tr>
<tr>
<td>A7</td>
<td>Missing Function Level Access Control</td>
</tr>
<tr>
<td>A8</td>
<td>Cross-Site Request Forgery (CSRF)</td>
</tr>
<tr>
<td>A9</td>
<td>Using Known Vulnerable Components</td>
</tr>
<tr>
<td>A10</td>
<td>Unvalidated Redirects and Forwards</td>
</tr>
</tbody>
</table>

Merged with 2010-A7 into new 2013-A6
New Security Challenges

Cloud Computing & Big Data

Microservices

Internet of Things (IoT)
<table>
<thead>
<tr>
<th>Sprint 1</th>
<th>Sprint 2</th>
<th>Sprint ...n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Story B</td>
<td>Story C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Story D</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Story E</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Security Features</td>
</tr>
<tr>
<td></td>
<td>Story F</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Story G</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Story H</td>
<td>Penetration Test</td>
</tr>
</tbody>
</table>

8/3/2015
Chapter Meeting Stuttgart
Potentially Releasable?

Scrum Guide:

“ The Development Team consists of professionals who do the work of delivering a potentially releasable increment of “Done” product at the end of each Sprint ”

Release potentially unsecure?

http://www.scrumguides.org
Attacker Schedule: 24h x 7d
Microsoft Security Development Lifecycle (SDL)

https://www.microsoft.com/en-us/sdl/
Automotive SDL (V-Model)
Next Stop: **Secure** Agile Development
Manifesto for Agile Software Development

**Individuals and Interactions** over processes and tools

**Working software** over comprehensive documentation

**Customer collaboration** over contract negotiation

**Responding to change** over following a plan

8/3/2015  Chapter Meeting Stuttgart
Manifesto for Secure Agile Software Development

Individuals and Interactions over processes and tools

Secure working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

complex security policies

vague security requirements

static threat model

8/3/2015 Chapter Meeting Stuttgart
Agile Development
The Rugged Manifesto

I recognize that my code will be used in ways I cannot anticipate, in ways it was not designed, and for longer than it was ever intended.

I recognize that my code will be attacked by talented and persistent adversaries who threaten our physical, economic, and national security.

I am rugged because I refuse to be a source of vulnerability or weakness.

I am rugged, not because it is easy, but because it is necessary... and I am up for the challenge.

https://www.ruggedsoftware.org
# Microsoft SDL for Agile Development

<table>
<thead>
<tr>
<th>Agile Process</th>
<th>SDLC Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>One Time</strong></td>
<td>• Baseline thread model</td>
</tr>
<tr>
<td></td>
<td>• Establish security response plan</td>
</tr>
<tr>
<td></td>
<td>• ...</td>
</tr>
<tr>
<td><strong>Regular Basis</strong></td>
<td>• Privacy review</td>
</tr>
<tr>
<td></td>
<td>• Manual &amp; automatic security code review</td>
</tr>
<tr>
<td></td>
<td>• ...</td>
</tr>
<tr>
<td><strong>Every Sprint</strong></td>
<td>• Security training</td>
</tr>
<tr>
<td></td>
<td>• Threat modeling</td>
</tr>
<tr>
<td></td>
<td>• Secure coding</td>
</tr>
<tr>
<td></td>
<td>• Code reviews</td>
</tr>
<tr>
<td></td>
<td>• ...</td>
</tr>
</tbody>
</table>
OWASP Open Software Assurance Maturity Model (OpenSAMM)

http://www.opensamm.org/
OpenSAMM - Construction

- Are project teams provided with a list of recommended third-party components?
- Are most project teams aware of secure design principles and applying them?

- Do you advertise shared security services with guidance for project teams?
- Are project teams provided with prescriptive design patterns based on their application architecture?

- Are project teams building software from centrally controlled platforms and frameworks?
- Are project teams being audited for usage of secure architecture components?
OpenSAMM - Verification

- Are projects specifying some security tests based on requirements?
- Do most projects perform penetration tests prior to release?
- Are most stakeholders aware of the security test status prior to release?

- Are projects using automation to evaluate security test cases?
- Do most projects follow a consistent process to evaluate and report on security tests to stakeholders?

- Are security test cases comprehensively generated for application-specific logic?
- Do routine project audits demand minimum standard results from security testing?
Agile Development with Scrum

Product Owner

Scrum Master

Developer Team

Developer

Test & QA

OWASP
Open Web Application Security Project

8/3/2015
Chapter Meeting Stuttgart
Security Trainings

- Security Risk Identification & Management
- Data Privacy Requirements
- Threat Modeling
- Security Features
- Security User Stories („Abuse Stories“)

Product Owner

- Threat Modeling
- Secure Coding
- Security Code Reviews
- Security Testing (Manual & Automatic)

Development Team

8/3/2015
Chapter Meeting Stuttgart
Threat Modeling is „Agile“...

1. Define Software-Architecture
   User Stories, UML Diagrams

2. Adapt Threat Model
   Discussion Basis

3. Identify and Mitigate Threats
   „Elevation of privilege“ game

4. Create Security Testcases and Abuse User Stories

5. Test Driven Development (TDD)
   Write Security Tests First

6. Create Production Code
   Make Tests Pass

8/3/2015
Chapter Meeting Stuttgart
Playing games...

Scrum Planning Poker

Threat Modeling Game
Secure Agile Development with Scrum

- Update threat model (on-going)
- Define abuse user stories
- Plan security features early
- Security acceptance criteria
- Extend „Definition of Ready“ with security
Abuse (Evil) User Stories

As a customer I want to select products and add them to my shopping cart in order to buy these.

As an evil user I want to manipulate requests to change prices when adding products to my shopping cart.
Secure Agile Development with Scrum

- Update threat model (on-going)
- Plan abuse user story tasks
- Plan security features tasks
- Refine security acceptance criteria
- Define and plan security test strategies
Secure Agile Development with Scrum

**Daily Scrum**
- Discuss security risks
- (Re-)plan security tasks

**Sprint**
- Secure coding
- Pair programming/test with security expert
- „Security Officer“ developer
- „Security-Aware“ Definition of Done
- Security regression testing (CI)
- Security code reviews
Secure Coding – „Clean Code“

- Boundaries (3rd party code)
- Error Handling
- Good Comments
- Unit Tests
- Constant Refactoring
- Keep it simple & DRY
- ...

8/3/2015

Chapter Meeting Stuttgart
Secure Coding – Security Patterns

• Input validation
• Output escaping
• Use prepared SQL statements
• No sensible data in logs
• No stack traces on the UI
• Session management
• Access Controls
• ...

8/3/2015 Chapter Meeting Stuttgart
Secure Coding – Standard Frameworks

- Standard cryptography
- Validating/escaping UI frameworks e.g.
  - JavaServer Faces
  - Vaadin (GWT)
- Proven security frameworks
  - Spring Security
  - Apache Shiro
  - OWASP ESAPI
Secure Coding – Secure app in 5 minutes
Secure Coding – Security issues in the IDE

http://www.contrastsecurity.com/eclipse

8/3/2015

Chapter Meeting Stuttgart
Secure Agile Testing

Functional Tests
Examples
Story Tests
Prototypes
Simulations

Exploratory Testing
Scenarios
Usability Testing
UAT (User Acpt. Testing)
Alpha / Beta

Unit Tests
Component Tests

Performance & Load Testing
Security Testing
"ility" Testing

Automated & Manual
Supporting the Team
Automated

Manual
Critique Product
Technology Facing

Crispin, Lisa; Gregory, Janet (2008). Agile Testing:
Secure Agile Testing

- Manual, Exploratory Testing
- Automated Security UI-Tests
- Service Layer Tests (API-Layer)
  - Subcutaneous Tests (Martin Fowler)
- Unit & Component Tests

Secure Agile Testing

8/3/2015 Chapter Meeting Stuttgart
Stage 1: Static Security Testing

Continuous Integration (CI)

1. **Pull-Request**
2. **Trigger Build**
3. **Check-Out**
4. **JUnit Build & Tests & Static Code Analysis & Dependency Check**
5. **Report Build Result**
7. **Push to Stable**

- **Developer 1**
- **Developer 2**

8/3/2015

Chapter Meeting Stuttgart

8/3/2015
Static Code Analysis: FindBugs

- Detects 63 different vulnerability patterns
- OWASP TOP 10 and CWE coverage
- IDE- and CI-integrations

http://findbugs.sourceforge.net
http://h3xstream.github.io/find-sec-bugs
Static Code Analysis: SonarQube

+ Java Plugin

• Detects lots of security problems
  – SANS TOP 25
  – OWASP Top 10
  – CWE references (http://cwe.mitre.org)

http://www.sonarqube.org
http://docs.sonarqube.org/display/PLUG/Java+Plugin

8/3/2015  Chapter Meeting Stuttgart
Dependency Check

- Detects vulnerabilities in 3rd party libraries
  - Java, .NET, Python applications
  - Checks library information against imported CVE data
  - Sometimes needs suppressing false positives
  - Supports Command line, Ant, Maven, Gradle, Jenkins

Project: Bodgeit Store Vulnerable Application

Scan Information (show all):
- dependency-check version: 1.2.11
- Dependencies Scanned: 3
- Vulnerable Dependencies: 0
- Vulnerabilities Found: 0
- Vulnerabilities Suppressed: 0
- ...

Display: Showing All Dependencies (click to show less)

<table>
<thead>
<tr>
<th>Dependency</th>
<th>CPE</th>
<th>GAV</th>
<th>Highest Severity</th>
<th>CVE Count</th>
<th>CPE Confidence</th>
<th>Evidence Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.servlet-api:3.1.0.jar</td>
<td>javax.servlet javax.servlet-api:3.1.0</td>
<td>0</td>
<td>0</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>commons-lang:3.3.2.jar</td>
<td>org.apache.commons:commons-lang:3.3.2</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hsqldb:2.3.2.jar</td>
<td>org.hsqldb:hsqldb:2.3.2</td>
<td>0</td>
<td>0</td>
<td>21</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

http://jeremylong.github.io/DependencyCheck

8/3/2015 Chapter Meeting Stuttgart
Stage 2: Dynamic Security Testing

Acceptance Testing

1. Deploy
   - Apache Tomcat
   - Nexus
   - Jenkins

2. UI-Testing
   - Selenium

3. Active Scanning
   - OWASP ZAP

4. Reporting
   - Thread Fix
   - Atlassian JIRA Agile

Proxy
Secure Agile Development with Scrum

- Verify threat model coverage
- Review abuse user stories and security acceptance criteria
- Make security topics visible for customer
- Inspect and adapt security activities
Security == Agile!

<table>
<thead>
<tr>
<th>Sprint 1</th>
<th>Sprint 2</th>
<th>Sprint ...n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story A</td>
<td>Story C</td>
<td>Story F</td>
</tr>
<tr>
<td>Story B</td>
<td>Story D</td>
<td>Story G</td>
</tr>
<tr>
<td>Abuse Story</td>
<td>Story E</td>
<td>Story H</td>
</tr>
<tr>
<td>Security Features</td>
<td>Abuse Story</td>
<td>Pen-Test</td>
</tr>
</tbody>
</table>

8/3/2015 Chapter Meeting Stuttgart
Don’t make it that EASY to break software!