Security, a part of QA
My claim

In custom software, if you haven’t properly tested it, it probably doesn’t work. This goes for both functional and nonfunctional requirements.

Worse yet if you don’t even know what ‘it’ is supposed to be.
Who is this then?

Boy Baukema
Security Specialist @ Ibildings.nl
Security what?

Senior Engineer
+ interest in WebAppSec
+ 4 hours a week R&D
+ internal training & consultancy
+ internal & external auditing
Okay, and you do this where?

Ibuildings.nl
web & mobile, 20+ devs, mostly PHP
You

developer, manager, executive

pentester, security consultant, ?
The plan

1. The journey
2. The holy grail
3. Riding off into the sunset
A assignment

Make security something I can sell,
give managers a knob to turn
OWASP ASVS

Open Web Application Security Project

Application Security Verification Standard
<table>
<thead>
<tr>
<th>Chapter 1</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement 1.1</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Requirement 1.2</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Requirement 1.3</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 2</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement 2.1</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ASVS Levels (2013)

Level 0 - Bullshit compliance level (0)
Level 1 - Opportunistic (47)
Level 2 - Standard (136)
Level 3 - Advanced (164)
|---------------------|---------------------|------------------------|-------------------|---------------------|--------------------------|-----------------------------|------------------------|--------------------------|-------------------|--------------------------|------------------|--------------------------|-------------|
An example

V1.4. Verify that credentials and all other identity information handled by the application does not traverse unencrypted or weakly encrypted links.
(level 1, 2 & 3)
So how does this tie into QA?
First attempt

V2.7 Verify that the strength of any authentication credentials are sufficient to withstand attacks that are typical of the threats in the deployed environment.

(OWASP ASVS 2009 Level 2)
AASVS, Scanners &
A Report Generator

V1 - Security Architecture Documentation Requirements

V1.1 - Verify that the Security Architecture is present in the context of the application.

V1.2 - Verify that all components that are not part of the application to operate are identified.

Why can this not be verified?

Example:

<table>
<thead>
<tr>
<th>Threat agent factors</th>
<th>Vulnerability factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skill level</td>
<td>Ease of discovery</td>
</tr>
<tr>
<td></td>
<td>Ease of exploit</td>
</tr>
<tr>
<td>Motive</td>
<td>Awareness</td>
</tr>
<tr>
<td>Opportunity</td>
<td>Intrusion detection</td>
</tr>
<tr>
<td>Size</td>
<td></td>
</tr>
</tbody>
</table>

0 - Not Applicable    0 - Not Applicable
Enter ASVS 2013 (Beta)

Release any day now!
+ is for effort

... scope of the verification may go beyond the application’s custom-built code and include external components. Achieving a verification level under such scrutiny can be represented by annotating a “+” symbol to the verification level.
3. Table of Contents

- 1. Introduction
  - 1.1. Target of Verification (TOV)
  - 1.2. Scope
  - 1.3. Confidentiality
- 2. Document history
- 3. Table of Contents
- 4. Conclusions
  - 4.1. Vulnerabilities
- 5. V1: Authentication
  - 5.1. V1.1: Principle of complete mediation
  - 5.2. V1.2: Password fields
  - 5.3. V1.3: Fails securely
  - 5.4. V1.4: Strongly encrypted transport
  - 5.5. V1.5: No clear text passwords
  - 5.6. V1.6: No username enumeration
  - 5.7. V1.7: No default passwords
- 6. V2: Session Management
  - 6.1. V2.1: Uses default session management
  - 6.2. V2.2: Sessions are invalidated on user log out
  - 6.3. V2.3: Session times out after inactivity
  - 6.4. V2.4: Shows logout link
  - 6.5. V2.5: Does not disclose session id
  - 6.6. V2.6: Change or clear session id on logout
  - 6.7. V2.7: Authenticated session tokens are protected with HttpOnly
  - 6.8. V2.8: Authenticated session tokens are protected with Secure and HSTS
A plan for the future

Software Assurance Maturity Model
A guide to building security into software development
VERSION - 1.0
<table>
<thead>
<tr>
<th>Objective</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establish process to perform basic security</td>
<td>A. Derive test cases from known security</td>
</tr>
<tr>
<td>tests based on implementation and software</td>
<td>requirements</td>
</tr>
<tr>
<td>requirements</td>
<td>B. Conduct penetration testing on software</td>
</tr>
<tr>
<td></td>
<td>releases</td>
</tr>
<tr>
<td>Make security testing during development more</td>
<td>A. Utilize automated security testing tools</td>
</tr>
<tr>
<td>complete and efficient through automation</td>
<td>B. Integrate security testing into development</td>
</tr>
<tr>
<td></td>
<td>process</td>
</tr>
<tr>
<td>Require application-specific security testing</td>
<td>A. Employ application-specific security testing</td>
</tr>
<tr>
<td>to ensure baseline security before deployment</td>
<td>automation</td>
</tr>
<tr>
<td></td>
<td>B. Establish release gates for security testing</td>
</tr>
</tbody>
</table>
The End

Questions?

boy.baukema@owasp.org
boy@ibuildings.nl
https://twitter.com/relaxnow