Heroes vs Villains:
Building an Application Security Program that Scales

Kevin Delaney, B.IT Hons. NetSec
Director of Solutions Architecture
Security Compass
Over 160 Million Credit Cards lifted over 7 years
Villains are PROACTIVE
Heroes are REACTIVE
5 Step Process
Why does this happen?

- Inexperienced developers
- Apathy towards secure development
- Overwhelming requirements documents
- Too much reliance on static and dynamic analysis tools
Pin-pointing vulnerabilities before cyber criminals do

Customer requirements and ever changing compliance standards

The Struggle is Real.
Good help is hard to find

Your company is not the only one that struggles to find the experienced IT professionals and security architects necessary to perform risk assessments

- **70%** of respondents believe their organization does not have enough IT Security Staff
- **36%** of security positions were unfilled.
- **58%** of senior security positions were unfilled.
Shallow Talent Pool

Figure 4. Average percentage of IT security positions not filled

Understaffed and at Risk:
Today’s IT Security Department - Ponemon Institute
The Numbers

• Demand for InfoSec jobs growing 3.5x faster than other IT jobs, 12x faster than all jobs.
• 12,000 InfoSec professionals surveyed believe that talent shortage weakened their defenses [ISC2]
• 70% of companies surveyed in the US believe their IT Security department is understaffed.
• 50,000 CISSP postings in the US alone, but only 60,000 CISSP’s worldwide.
An Expensive Endeavor

Average Security Architect salary in the United Kingdom is £75,000
Employers want certified domain experts with multiple years of experience in:

- Network security governance
- Policies
- Procedures
- Application Security

General Security Knowledge is not Enough
Do more with less

• Stop relying on just your security team for security

• Identify security champions in your development team and empower them.

• Incentivize with training and certifications - transferrable skills.

• Teach your heroes to think like VILLAINS!
• How to develop an application security program

• How to reduce production costs, application vulnerabilities, and delivery delays

• How to ensure that secure software is accepted and delivered effectively.
What makes a GREAT AppSec Program?
Adaptable

Security Requirements
- Scaled Security Information
- Tailored Security Information
- Security Baseline
A great appsec program is focused on the strengths of the people participating.

Ideally, security tasks should be generated on-the-fly based on the profile of the application and its associated risks and delivered directly into your developers’ ALM tools like JIRA or TFS.

Ensures nothing is missed and reduces time spent searching for what’s applicable to a project by multitudes.
<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>P</th>
<th>Status</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEP-22</td>
<td>T38: Bind variables in SQL statements</td>
<td>Unassigned</td>
<td>Kevin Delaney</td>
<td></td>
<td>TO DO</td>
<td>Unresolved</td>
</tr>
<tr>
<td>KEP-9</td>
<td>T21: Ensure confidential data is sent over an encrypted channel</td>
<td>Unassigned</td>
<td>Kevin Delaney</td>
<td></td>
<td>TO DO</td>
<td>Unresolved</td>
</tr>
<tr>
<td>KEP-35</td>
<td>T375: Release resources when no longer needed</td>
<td>Unassigned</td>
<td>Kevin Delaney</td>
<td></td>
<td>TO DO</td>
<td>Unresolved</td>
</tr>
<tr>
<td>KEP-34</td>
<td>T70: Implement account lockout or authentication throttling for system accounts</td>
<td>Unassigned</td>
<td>Kevin Delaney</td>
<td></td>
<td>TO DO</td>
<td>Unresolved</td>
</tr>
<tr>
<td>KEP-33</td>
<td>T32: Always perform input validation on the server</td>
<td>Unassigned</td>
<td>Kevin Delaney</td>
<td></td>
<td>TO DO</td>
<td>Unresolved</td>
</tr>
<tr>
<td>KEP-32</td>
<td>T50: Use indirect object reference maps if accessing files</td>
<td>Unassigned</td>
<td>Kevin Delaney</td>
<td></td>
<td>TO DO</td>
<td>Unresolved</td>
</tr>
<tr>
<td>KEP-31</td>
<td>T37: Avoid DOM-based cross site scripting (XSS)</td>
<td>Unassigned</td>
<td>Kevin Delaney</td>
<td></td>
<td>TO DO</td>
<td>Unresolved</td>
</tr>
<tr>
<td>KEP-30</td>
<td>T31: Perform input validation on all forms of input</td>
<td>Unassigned</td>
<td>Kevin Delaney</td>
<td></td>
<td>TO DO</td>
<td>Unresolved</td>
</tr>
<tr>
<td>KEP-29</td>
<td>T42: Avoid relying on untrusted data for server side page, view, or template selection</td>
<td>Unassigned</td>
<td>Kevin Delaney</td>
<td></td>
<td>TO DO</td>
<td>Unresolved</td>
</tr>
</tbody>
</table>
T37: Avoid DOM-based cross site scripting (XSS)

You need to be aware that some of the DOM properties can be controlled by the user and might include unsafe values. An example is `<b>document.location</b>`. If this property is used in your dynamic JavaScript code to create HTML content in an unsafe manner (e.g. assigned to `element.innerHTML`), it can allow attackers to inject script by creating malicious links and essentially leading to a cross-site scripting vulnerability.

To prevent a DOM-based XSS, you need to:

- Treat the user-controlled DOM values as unsafe.

Some of the DOM properties that may be manipulated for XSS include (note that these properties might have sub-attributes such as `location.hash` that are considered unsafe as well):

```javascript
document.URL
document.URLUnencoded
document.location (and many of its properties)
document.referrer
window.location (and many of its properties)
location (note that window.location can be access as just location)
```
Collaborative

• No more “us vs. them” mentality between developers and security.

• Developers must take responsibility for security tasks.

• You cannot create a security culture – it is created from within the development org.
Recap

- Proper management of security requirements early in the SLDC prevents problems before they happen and turns down the noise from static/dynamic analysis tools.

- Delivering these requirements directly to developers in the tools they use every day is critical for acceptance.

- Leverage and empower your existing resources, because finding new ones is no easy task.

- Make sure your AppSec program is adaptable, focused, and collaborative.
Thank You, Villains!

Kevin Delaney
Director, Solutions Architecture
kdelaney@securitycompass.com

http://securitycompass.com/