OWASP Changing the Game

A study of heroic behavior

Jason Kent
Director, Web Application Security
KzEuNjE0LjQ0Ni4wODcw

Qualys
whoami
Dean’s inventions
Dean’s inventions
Solutions

• When we are faced with problems, we try to find solutions
• The problems facing today’s Application Security professionals are many, but the solutions to them need to be simple
Why Web App Security Matters

Visible Targets

“The inherent need for many web applications to be Internet visible makes them a logical target”

Associated with Data Loss

“Web Applications....were associated with over a third of total data loss”

Popular and Successful Attack Vector

“Web applications abound in many larger companies, and remain a popular (54% of breaches) and successful (39% of records) attack vector. “
Why Web App Security Matters

Hacking vectors by percent of breaches within Hacking

- Remote access/desktop services: 20%
- Backdoor or control channel: 34%
- Web application: 54%
- Unknown: 17%

Larger Orgs
# Why Web App Security Matters

<table>
<thead>
<tr>
<th>Type</th>
<th>Category</th>
<th>All Orgs</th>
<th>Larger Orgs</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS server (store controller)</td>
<td>Servers</td>
<td>50%</td>
<td>1%</td>
</tr>
<tr>
<td>POS terminal</td>
<td>User devices</td>
<td>35%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>Desktop/Workstation</td>
<td>User devices</td>
<td>18%</td>
<td>34%</td>
</tr>
<tr>
<td>Automated Teller Machine (ATM)</td>
<td>User devices</td>
<td>8%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>Web/application server</td>
<td>Servers</td>
<td>6%</td>
<td>80%</td>
</tr>
<tr>
<td>Database server</td>
<td>Servers</td>
<td>6%</td>
<td>96%</td>
</tr>
<tr>
<td>Regular employee/end-user</td>
<td>People</td>
<td>3%</td>
<td>1%</td>
</tr>
<tr>
<td>Mail server</td>
<td>Servers</td>
<td>3%</td>
<td>2%</td>
</tr>
<tr>
<td>Payment card (credit, debit, etc.)</td>
<td>Offline data</td>
<td>3%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>Cashier/Teller/Waiter</td>
<td>People</td>
<td>2%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>Pay at the Pump terminal</td>
<td>User devices</td>
<td>2%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>File server</td>
<td>Servers</td>
<td>1%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>Laptop/Netbook</td>
<td>User devices</td>
<td>1%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>Remote access server</td>
<td>Servers</td>
<td>1%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>Call Center Staff</td>
<td>People</td>
<td>1%</td>
<td>&lt;1%</td>
</tr>
</tbody>
</table>

*Assets involved in less than 1% of breaches are not shown*
Best Practice - Security Early in Lifecycle

The costs for fixing security flaws are dramatically lower the earlier in the development lifecycle they are fixed.

Conventional Approach
Bottlenecked at IT Security

Developer

App Owner

IT Security

Scanner

IT Security Bottleneck

Report of Scan Results

Vulnerabilities not identified or fixed early in lifecycle

Developer

Fix Code

Dev App1

QA App1

UAT-App1

Prod-App1

Scan App

Promote

Promote

Promote
Today

- Web Applications are often not secure
- We spend time chasing the application owners to fix code
- They don’t have a project for it and one has to be created
- We spend more time creating paperwork than doing work
App Sec Today
Dean’s philosophy

• Never build a south pointing chariot
• Attack the problem in a manner that suits everyone
• Try to find a solution that uses normal behaviors as a guide
Trends

• Some organizations are mandating scanning in the SDLC
  – Most are failing at it

• DEV Teams are begging for a way to get the App Sec Team off of their back
Why failing?
**Acronyms used in this document**

- **ATL** - App Team Lead
  - The manager on the Apps team that is leading their half of the process
- **TT** – Tech Team
  - The IS Security resource performing the Scanning and Reporting
- **PM** – Project Manager
  - The IS Security resource scheduling the meetings and coordinating schedules between teams

---

**Identify App**
- Identify App Team Lead
- Send Discovery Doc to ATL
- Set Discovery Meeting

**Discovery**
- Discovery Meeting
  - ATL +1, TT, PM

**Pre-Scan**
- Spider App
- Set up W/L and B/L
- Verify Complete URL list with ATL
  - Yes
  - No

**Scan**
- Run Scan during approved window
- Clean Scan
  - Yes
  - No

**Report**

**Discovery Doc from ATL**
- Base URL
- Username/Password
- Scan Window
- Data overview (PII, sensitivity)
- Architectural Diagram

**Last Scan Response Doc**
- Troubleshoot Scanner Issues
  - No
  - Yes
- Open ticket with scanning vendor
App Sec Tomorrow

- Security bugs are function bugs
- The same QA processes apply
- The QA team and DEV are familiar with App Sec Tools - Scanners, Proxies (ZAP anyone?) are used as a QA step
- Tools all feed standard DEV reporting tools (Bugzilla)
How close are we?

• ZAP is gaining popularity with QA
• Some tools on the market can be setup for QA to use
• Open Source is ruling processes, we need to harness that
  – Selenium
  – Thread Fix
The ultimate workflow

• DEV checks code into their DEV/QA system
  – QA performs function tests and app scans at the same time
  – They return bugs to DEV
  – DEV realizes they are using a bad validation routine or regularly forgetting tokens etc...
  – DEV fixes their libraries to match best security practice
Living in a vacuum

DEV
Creating Problems

QA Function and Security

Tested app deployed

Audit

Prod Scan

Pen Test
A reality

Automation is an efficiency force multiplier

– Jason Kent
Eliminating IT Security Bottleneck

All Stakeholders Participate

- Developer
- QA
- App Owner
- IT Security
- Cloud Portal

Fix Code

Cloud Portal

Scan Dev

Scan QA

Reports

Oversight / Production Scanning

Vulnerabilities are identified and fixed early in lifecycle
Thank You

jkent@qualys.com

@jkentakula