Tips for Building a Successful Application Security Program

Reducing Software Risk

Denver Chapter
January 16, 2013
• Dave Ferguson
  – Veracode Solutions Architect
  – Certifications: CISSP, CSSLP

• In the Past
  – Principal Consultant with FishNet Security
  – Software Applications Developer

• Industry Involvement
  – OWASP
    • Member, Contributor, Former KC chapter leader
    • Forgot Password Cheat Sheet
  – Blog
    • http://appsecnotes.blogspot.com
State of Software Security?
OWASP
The Open Web Application Security Project

applications

end points

network

data
applications

end points

network

data

$35 billion spent in 2011
Applications Are Complex
Why Is Software a Target?

» $300 billion in software produced or sold each year

» One of the world's largest manufacturing industries

» No uniform standards or insight into security risk or liability of the final product

75% percent of attacks take place at the application layer.

62% of companies experienced security breaches in critical applications within the last year.

Upon first test, 8 in 10 applications contained XSS and/or SQL injection flaws.
"Developers and defenders need to be right every time. An attacker only has to be right once."
Big Companies Are Hacked via Web Apps

What kind of application vulnerabilities are out there?
Web Apps

Source: Veracode State of Software Security Report, Volume 4
Non-Web Apps

Top Vulnerability Categories
(Percent of Applications Affected for Non-Web Applications)

- Cryptographic Issues: 46%
- Directory Traversal: 34%
- Error Handling: 24%
- Information Leakage: 23%
- Potential Backdoor: 23%
- Time and State: 19%
- Buffer Management Errors: 17%
- OS Command Injection: 15%
- Credentials Management: 15%
- Buffer Overflow: 14%
- CRLF Injection: 13%
- Numeric Errors: 12%
- SQL Injection: 11%
- Untrusted Search Path: 11%
- Dangerous Functions: 10%

Source: Veracode State of Software Security Report, Volume 4
Source: Veracode State of Software Security Report, Volume 4
Targeting the Software Supply Chain

Lockheed says Cyber Attacks Up Sharply, Suppliers Targeted
By Andrea Shalal-Esa | Reuters – Mon, Nov 12, 2012

The Information Security Forum Announces Top Five Security Threats in 2013
Posted November 29, 2012
2. Supply Chain Security
More organizations will fall victim to information security incidents at their suppliers.

Gartner Study Questions Integrity Of IT Supply Chain
By Ken Presti
October 18, 2012 8:04 PM ET
The Risks of Third-Party Software

"SOUP" : Software of Unknown Pedigree

80% of 3rd party software fail basic OWASP test for security compliance.
— PWC 2012 Security Report

60% of companies do not do any security risk mitigation when outsourcing development.
— Gartner "Global Sourcing Risks and Success Factors"

less than 1 in 5 enterprises are conducting security assessments from 3rd parties
Challenges

- Managing application sprawl
- Understanding criticality / business risk
- Choosing assessment technique
  - Static analysis?
  - Dynamic analysis?
  - Manual pen testing?
- Cost – tools & labor
- Tracking/measuring progress
And then there's...

- Lack of executive support
- Expertise required to run tools
- Turnover
- 3rd party / vendor apps
- Mobile apps
- Deciding which flaws to fix
- Friction between dev teams & security
- Communication inefficiencies
- Lack of secure coding skills
- Siloed development teams
- Remediation effort / re-testing
The Solution?
Application Security Program

- Centralized
- Policy-Driven
- Comprehensive
The Four Pillars

Policies, Management

Scans, Pen Testing, Remediation

Developer Training

Metrics

AppSec Program
Phased Implementation

- **Level 1**: Ad-Hoc Testing
  - Consultants
  - Tools
  - One-off audits

- **Level 2**: Security Program
  - Identify critical applications
  - Assign security policy
  - Achieve compliance

- **Level 3**: Process Integration
  - Standardized process
  - SDLC systems integration
  - Procurement process integration

- **Level 4**: Optimized Decision-Making
  - IT decisions based on application intelligence
  - Improved risk profile

- **Level 5**: Mature
  - Proactive risk management
  - Seamless on-boarding of developers, suppliers and acquisitions
Tips for a Successful Program

- Executive sponsorship
- Center of Excellence approach
- Application inventory and classification
- Use of security policies
- Have defined roles
- Automated scans (static & dynamic)
- Manual penetration testing
- Metrics & reporting
- Automation / integration into SDLC
- Remediation
- Validate 3rd-party & outsourced software
- Developer training
- Collaboration with devs (avoid scan & scold)
Executive Support

- Essential for a successful program
- Approves corporate resources
- Sets software security as a priority
- Validates the business risks
- Assures that acceptable security is achieved within release timelines

Center of Excellence

- Mission: To establish a standard for ensuring software security across the organization
- Formalize elements of the program
- Consistent approach across risk domains
  - Internally-developed
  - Outsourced
  - Purchased
  - Open source
  - Mobile apps
Establish Security Policies
- Assign different policies for different business risk
- Remediation requirements naturally fall out
- Don't require perfect code!
  - Consider standards such as OWASP, SANS Top 25, or PCI
  - Consider severity rating or specific CWEs
- Specify required testing techniques and test frequency

Enforce Policies
- Instill accountability for policy compliance with application owners
- Define escalation paths
Define Roles

- Need clear distinction between management and operational roles
- Who establishes appropriate security policies?
- Who determines the business criticality of the apps?
- Who decides the proper policy to assign to an app?
- Who is responsible for testing the app?
- How are conflicts resolved?
- Who monitors the program and reports to executive management?

Test Third-Party Applications

- Hold vendors, contractors, and outsourcers to the same standards as internal development teams
- Employ independent verification
- Pre-define security policy so all parties understand acceptance criteria
Case Study: Metrics show rapid improvement

Flaw Density (Flaw/MB)

App A | App B | App C

| Flaws per MB on initial scan | Flaws per MB on rescan |

"App A" flaws found over time

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Flaw Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012-01-23</td>
<td>04:19:24</td>
<td>63</td>
</tr>
<tr>
<td>2012-01-26</td>
<td>06:25:00</td>
<td>18</td>
</tr>
<tr>
<td>2012-01-27</td>
<td>09:53:58</td>
<td>2</td>
</tr>
<tr>
<td>2012-01-27</td>
<td>15:01:22</td>
<td>1</td>
</tr>
</tbody>
</table>
Remediation is Achievable

⇒ Over 80% of applications that were remediated to a satisfactory level did so in 1 week or less

**Time to Acceptable Quality by Industry Type**

<table>
<thead>
<tr>
<th>Industry Type</th>
<th>0-1 Week</th>
<th>2-3 Weeks</th>
<th>3-4 Weeks</th>
<th>4+ Weeks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software</td>
<td>71%</td>
<td>5%</td>
<td>9%</td>
<td>15%</td>
</tr>
<tr>
<td>Government</td>
<td>80%</td>
<td>2%</td>
<td></td>
<td>18%</td>
</tr>
<tr>
<td>Financial</td>
<td>76%</td>
<td>4%</td>
<td>5%</td>
<td>15%</td>
</tr>
<tr>
<td>Overall</td>
<td>82%</td>
<td>3%</td>
<td>4%</td>
<td>11%</td>
</tr>
<tr>
<td>Other</td>
<td>90%</td>
<td>2%</td>
<td>1%</td>
<td>7%</td>
</tr>
</tbody>
</table>
Injecting Security into the SDLC

**Software Lifecycle**

1. **Requirements**
   - Training
     - To help developers learn about the secure coding practices. Also provides CPE credit for certifications like CISSP.

2. **Design**
   - Static Analysis
     - Submit scan through build server

3. **Construction**
   - Static & Dynamic Analysis
     - Scanning engine analyzes the application code and returns results.

4. **Testing**
   - Static & Dynamic Analysis
     - Set up dynamic scanning (if web app) during the testing phase. Continue static scanning on reworked modules of the application.

5. **Release**

6. **Operations**

**VERIFICATION**
Developer Training

• Instructor Led
  ➢ Allows hands-on hacking
  ➢ Allows for interaction
  ➢ Costly
  ➢ Limited bandwidth

• CBT / eLearning
  ➢ More flexible / higher bandwidth
  ➢ Customized curricula
  ➢ Less costly per student
  ➢ Retention of content may be less

• Both:
  ➢ Secure code examples
  ➢ Assessments & quizzes
Case Study:
Secure code training measurably reduces SQL injection

Four (4) flaws determined to be non-exploitable (e.g. mitigated). Developers proposed mitigation; security team reviewed & approved.
Continuous Integration / Build Systems

- Ant
- Maven
- Hudson
- Jenkins
- Microsoft TFS
- IBM Rational Build Forge
- Collabnet TeamForge
- CruiseControl
- QuickBuild
- AntHillPro
Automating Static Analysis

1) Grab Code
2) Compile
3) Launch Static Scan
4) Download Results
5) Import to Bug Tracking
6) Load into IDEs

Source Code Repository
Continuous Integration Server
Code Scanning Engine
Bug Tracking Systems and IDEs
Scans Automated via Build System
Advantages of Early Flaw Detection

“The National Institute of Standards and Technology (NIST) estimates that code fixes performed after release can result in 30 times the cost of fixes performed during the coding/development phase.”

Source:
Scan Early and Often!
Summary: Building a Successful App Sec Program

- **Critical to Success**: Executive Support/Sponsorship
- Center of Excellence Approach
- Application inventory and classification
- Defined security policies
- Technology for automated testing
- Manual pen testing
- Integration of static code analysis with internal SDLC
- Assessment of 3rd party/purchased software
- Remediation guidance
- Developer Training
- Metrics
Thank You

Contact info:
dferguson[at]veracode.com