AppScan® & Secure Coding Standards

Ed Murphy
Assistant Director, Enterprise Applications, UITS
The University of Arizona
Today's topics:

- Introduction to web application security
- Introduction to IBM's Rational AppScan
- Use of IBM's Rational AppScan at UA
Intro to Web App Security
What is a web app?

- Web site vs web application
  - No precise definition
  - Subtle distinction

- Web site
  - Provides access to static documents
  - No user input that affects business functionality

- Web app
  - Takes user input which affects backend business logic
  - Web server usually interacts with other backend servers
The myth: “Our site is safe”

- “We use network vulnerability assessments”
- Neglect the security of the software on the web server.
- “We have firewalls in place”
- Port 80 & 443 are open
- “We encrypt our data with SSL”
- Only protect data between site and the user not the web application itself
Web App Security: What Can Happen?

- Sensitive data leakage
- Identity Theft
- Defacement - Content Modification
- ApplicationShutdown (Site Unavailable)
- Remote Execution
WASC

Web Application Security Consortium

To develop, adopt, and advocate for standards for web app security

http://www.webappsec.org

Web Security Threat Classification Project

Clarify and organize the threats to the security of a web site

Develop and promote industry standard terminology for these issues
Open Web Application Security Project

http://owasp.org

Purpose of OWASP Top Ten project:

A broad consensus about what the most critical web app security flaws are

Raise awareness of web app security issues
# OWASP Top 10

<table>
<thead>
<tr>
<th>Threat</th>
<th>Impact</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross Site Scripting</td>
<td>Indentity theft, sensitive info leakage</td>
<td>Hackers impersonate legitimate users and control their accounts</td>
</tr>
<tr>
<td>Injection Flaws</td>
<td>Manipulate queries to DB/LDAP/other</td>
<td>Hackers access backend database info, alter or steal it</td>
</tr>
<tr>
<td>Malicious File Execution</td>
<td>Execute shell commands on server</td>
<td>Site modified to transfer all interactions to hacker</td>
</tr>
<tr>
<td>Insecure Direct Object Reference</td>
<td>Access to sensitive files and resources</td>
<td>Web app returns contents of sensitive file</td>
</tr>
<tr>
<td>Cross Site Request Forgery</td>
<td>Attacker invokes blind actions on web apps, impersonating trusted user</td>
<td>Blind requests to bank account transfer $ to hacker</td>
</tr>
<tr>
<td>Info Leakage &amp; Improper Error Handling</td>
<td>Attacker gains detailed system info</td>
<td>Malicious system recon helps develop further attacks</td>
</tr>
<tr>
<td>Broken Authentication &amp; Session Management</td>
<td>Session tokens not guarded or invalidated properly</td>
<td>Hacker forces session token on victim; tokens stolen on logout</td>
</tr>
<tr>
<td>Insecure Crytopgraphic Storage</td>
<td>Weak encryption techniques may lead to broken encryption</td>
<td>Confidential info (SSN, Credit Cards) can be decrypted</td>
</tr>
<tr>
<td>Insecure Communications</td>
<td>Sensitive info sent unencrypted over insecure channel</td>
<td>Unencrypted credentials sniffed and used by hacker</td>
</tr>
<tr>
<td>Failure To Restrict URL Access</td>
<td>Hacker can access unauthorized resources</td>
<td>Hacker can forcefully access a page past login page</td>
</tr>
</tbody>
</table>
Intro to AppScan

- AppScan Overview
- Security issues AppScan tests for
- How AppScan works
- Demo of AppScan
AppScan Overview

What is it? - A web app security assessment tool

Why do I need it? - To simplify finding and fixing web app security problems

What does it do? - Tests web apps, finds security issues and reports in actionable fashion

Who uses it? - Security Auditors, QA, Developers
What does AppScan test?

- Web Applications
- 3rd Party Components
- Web Server Configuration
- Web Server
- Database
- Applications
- Operating System
- Network
How does AppScan work?

- Traverse a web application
- Approaches the app as a “black box”
- Sends modified HTTP requests
- 1,000’s of tests for identifying vulnerabilities
Scan Stages

Explore
- Simulates a user clicking on links and filling in form fields.
- Builds site model based on config options
- Applies analysis engine to create list of potential vulnerabilities
- Creates tests based on potential vulnerabilities

Test
- Performs tests and evaluates vulnerabilities
Web Apps & Web Services

Web apps
- Supply AppScan with starting URL and login credentials

Web services
- Supply AppScan with WS URL (WSDL files)
- Builds GUI allowing user to input parameters and send requests to WS
- Process is recorded by AppScan and used to create test for the WS
Blind SQL Injection Testing

AppScan sends different HTTP requests, which test whether the entered data is used in a SQL context.

- **false:** `<parameter value>`' and `barfoo`='foobar

- **true:** `<parameter value>`' and `barfoo`='barfoo' and `<parameter value>`' or `barfoo`='foobar

If responses of “true” are the same as the original and the responses of “false” are different from original HTTP request than blind SQL injection is possible.
Reviewing Scan Results

- Issues are mapped to URL, Parameter and Cookie (last 2 are optional)
- Paths discovered in tests appear in tree
  - Successfully forcefully browsed to /admin
Issue Information

The following is displayed for each issue:

- **Advisory**: What is the security problem?
- **Recommendation**: general and specific (ASP.NET, J2EE and PHP)
- **Request/Response**: test variants, how test was executed, shows original HTTP traffic
AppScan Reports

- Security
- Industry Standard (OWASP Top 10, PCI)
- Regulatory Compliance (SOX, HIPPA)
- Delta Analysis
- Template Based
Ready to use AppScan?

- Access is through the Information Security Office
- web: http://security.arizona.edu/appscan
- email: iso@u.arizona.edu
- phone: (520) 621-UISO (8476)
Thank You!
Any Questions?