Web Attacks and How to Stop Them

John Rogers
Senior Application Security Engineer
Lincoln Financial Group
Objectives

Overview
- Threats
- Vulnerabilities
- Remediation
- Education & Awareness

Invite Discussion
- Please ask questions
  - Short ones during
  - Long ones after
Always Get Written Permission!

Warning Information, programs, and techniques contained in this presentation should not be used on corporate equipment or systems unless you have explicit written permission of the system owner, your management team and your Information Security department.

There are cases of well intentioned individuals having their employment terminated because they utilized penetration tools or techniques on company hardware or systems without written permission.
Introduction

John Rogers (JR)
- Senior Application Security Engineer
- Certified Information Systems Security Professional (CISSP)
- InfraGard Nebraska Board Member
- Application and User Interface developer
- Email: John.Rogers@lfg.com
Web Application Threats

- The Top Cyber Security Risks
  - SANS Report – September 2009

- >60% of all Internet attacks

- >80% of the vulnerabilities being discovered
  - Cross-Site Scripting
  - SQL Injection
Web Application Threats

- Immature Security Awareness
- In-House Development
- Deceptive Simplicity
- Rapidly Evolving Threat Profile
- Resource and Time Constraints
- Overextended Technologies
Web Application Threats

- Users can submit Arbitrary Input!
  - Never, ever, trust the client, no control.

- Users can interfere with any piece of data transmitted.

- Client side validation is considered a vulnerability.

- Don’t Trust Client Demo.
Web Application Threats

- **Don’t Trust Client Demo**
  - Firefox

- Burp Suite
  - Integrated platform for analyzing web applications.
  - Intercepting Proxy, Spider, Decoder, etc.
  - Combine manual and automated techniques to enumerate, analyze, attack and exploit web applications.
Web Application Threats

Don’t Trust the Client Demo

- User
- Browser
- Network
- SSL
- Web Server
- Web Application Server
- Database Server

<table>
<thead>
<tr>
<th>Browser</th>
<th>Burp Interception Proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE/Firefox</td>
<td>Intercepting Proxy</td>
</tr>
</tbody>
</table>

- HTTP Request
- HTTP Response
- SSL
Web Application Threats

Don’t Trust the Client Demo

- URL: http://demo.testfire.net

- Sign In
  - Username: jsmith
  - Password: Demo1234

- Modify Account Details
  - Add Select element
Web Application Threats

▷ Don’t Trust the Client Demo

Hello John Smith

Welcome to Altoro Mutual Online.

View Account Details:

1001160140 Checking
1001160140 Checking
1001160141 Savings
1001160141 Savings JR Hacked This

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.
Outline

1. Web Application Threats
2. Web Application Vulnerabilities
3. Remediation Techniques
4. OWASP
Web Application Vulnerabilities

- Firewall
- Cross-Site Scripting (XSS)
- SQL Injection
- Malicious File Execution
- Top Web Application Attacks 2008/2009
Web Application Vulnerabilities

- **Firewall**
  - False sense of security
  - Firewall = No Application Security
  - Port 80/443 are wide open
  - Firewall only = “Crunchy Outside, Soft Chewy Inside”
Cross-Site Scripting (XSS)

OWASP Rank/Description

#1 on OWASP Top Ten

- Attack against the client (initially)

- Application does not first validate data sent from client.

- Allows attackers to execute script in the victim's browser.
Web Application Vulnerabilities

Cross-Site Scripting (XSS)

1. An attacker finds an XSS hole in a web application.
2. The attacker creates an attack URL for stealing sensitive information and disguises it so that it appears legitimate.
3. The attacker distributes the malicious XSS link via social engineering to unsuspecting users.
4. When the victim logs in, Javascript embedded with the malicious XSS link executes and transmits the victim’s login information to the attacker.
Web Application Vulnerabilities

Cross-Site Scripting (XSS)

Types
- Reflective
- Stored
- DOM Injection

Demo
- AltoroMutual Search – Reflective XSS
- WebGoat XSS Lesson – Stored XSS
Web Application Vulnerabilities

Cross-Site Scripting (XSS)

- Demo – AltoroMutual Search – Reflective XSS
  - http://demo.testfire.net

- Unauthenticated vulnerability

- Test <script>alert('hi mom');</script>
- <iframe src=http://ha.ckers.org/scriptlet.html <

- RSnake XSS Cheatsheet
  - http://ha.ckers.org/xss.html
Cross-Site Scripting (XSS)

Demo – AltoroMutual Search – Reflective XSS
Cross-Site Scripting (XSS)

- Demo – WebGoat XSS Lesson – Stored XSS
- Start WebGoat – WebGoat 5.3
  - http://127.0.0.1:8080/webgoat/attack
  - user = guest, password = guest
  - Restart XSS Lesson
Cross-Site Scripting (XSS)

Demo – WebGoat XSS Lesson – Stored XSS

Solution Videos

Stage 2: Block Stored XSS using Input Validation.

THIS LESSON ONLY WORKS WITH THE DEVELOPER VERSION OF WEBGOAT

Implement a fix to block the stored XSS before it can be written to the database. Store a name such as 'Eric' with 'David' as the manager. Verify that 'David' is not written to the database.

* You have completed Stored XSS.
* Welcome to Block Stored XSS using Input Validation.
Web Application Vulnerabilities

Cross-Site Scripting (XSS)
- Attack Vectors
  - Steal Cookies
  - Key Logging
  - Port Scanning internal network
  - Steal Clipboard contents
  - Browser Exploits
  - Redirect to other malicious content
- Age of JavaScript attacks
  - JINX – JavaScript Malware Framework
  - Browser Exploitation Framework (BeEF)
Web Application Vulnerabilities

➤ **Cross-Site Scripting (XSS)**
  ➤ Defenses - Input Validation
    ➤ Use Standard input validation mechanism.

➤ Validate all input data for length, type, syntax, and business rules before trusting data.

➤ Worry about non-user entered input data.
Cross-Site Scripting (XSS)

- Defenses – Input Validation.
- OWASP Input Validation Strategies
  - Accept Known Good
  - Reject Known Bad
  - Sanitize
  - No Action – Best way to make the news.
  - Maybe a combination of first three.
Cross-Site Scripting (XSS)

Defenses – Output Encoding

Ensure that all user-supplied data is appropriately entity encoded (either HTML or XML depending on the output mechanism) before rendering to client.

“<“ and “>” = “&lt” and “&gt”

Specify the output encoding (such as ISO 8859-1 or UTF 8). Do not allow the attacker to choose this for your users.
Web Application Vulnerabilities

Cross-Site Scripting (XSS)

Defenses – Issues

- Use Input Validation and Output Encoding together.
  - Don’t allow hostile content to live in DB.

- Perform Input Validation Close to “Front Door”
  - Need to handle error processing.

- Blend Input Validation Strategies
  - Accept Known Good for everything but free form text fields. Use Reject Known Bad these fields.
Web Application Vulnerabilities

- **SQL Injection**
  - OWASP Rank/Description
    - #2 OWASP Top Ten
  - Attack against server
  - Common vulnerability
  - Occurs when user-supplied data is sent to an command/query interpreter.
  - Hostile data tricks the interpreter into executing unintended commands.
Web Application Vulnerabilities

- SQL Injection
  - Demo – AltoroMutual Authentication Bypass
    - Bypass Authentication
    - Single Quote Test
      - Talk directly to Interpreter
      - Lots of Information Leakage
      - Yum!
An Error Has Occurred

Summary:

Web Application Vulnerabilities

SQL Injection
  Demo – AltoroMutual Authentication Bypass
  Bypass Authentication
  Exploit Steps
    Username: admin
    Password: abc’ or 1=1;--
    Result: Hello Admin User
Web Application Vulnerabilities

- SQL Injection
  - Little Bobby Tables

Courtesy of XKCD – http://xkcd.com/327/
Web Application Vulnerabilities

- **SQL Injection**
  - Attack Vectors
    - Unauthorized Data Access
    - Unauthorized Data Modification
    - Database Destruction
    - OS Command Execution
      - Yee Haw!
Web Application Vulnerabilities

- **SQL Injection**
  - Defenses
  - Best
    - SQL Prepared Statements
      - Placeholder Substitution Markers
      - User Input is Strongly Typed.
Web Application Vulnerabilities

- **SQL Injection**
  - Defenses
  - Next Best
    - Escape Data. If possible, escape data before it is sent to the database query engine.
      - Escaping is bad for the Irish.
      - O’Malley -> OMalley
  - Input Validation
    - Same as Cross-Site Scripting.
Web Application Vulnerabilities

SQL Injection
- Defenses
  - Enforce least privilege when connecting to databases and other backend systems.
  - Avoid detailed error messages that are useful to an attacker.
  - Stored procedures can also be injected.
Vulnerabilities - Malicious File Execution

GIFAR - Code Smuggling

- Uploaded User Generated Content
- Billy Rios/Nathan McFeters at BlackHat
- GIF Image file combined with Java Archive file (JAR) = GIFAR
- GIF stores ident/control at front of file
- JAR (ZIP) stores ident/control at end of file
Top Web Application Attacks

- Top Web Application Attacks 2008/2009
  - 2008
  - 2009
Checkpoint

Web Application Threats

Web Application Vulnerabilities

Remediation Techniques

OWASP
Remediation Techniques

- No Silver Bullet
- Security & Software Development Lifecycle (SDLC)
- Static Source Scanning
- Dynamic Scanning
- Vulnerability Tracking
- Developer Training & Awareness
- Infrastructure
Remediation Techniques

» No Silver Bullet
  » Plan

» Resources
  » Hardware
  » Software
  » Human

» Hard Work
  » You will see results
Security & Software Development Lifecycle (SDLC)

- Security Tasks in the SDLC
- Developer Training for Security Tasks in the SDLC
  - Less than 10% (empirical evidence)
Remediation Techniques

- **SDLC - Security Tasks**
  - Planning
    - Security Task Durations
    - Plan to address outstanding security issues
  - Requirements
    - Reference Security Policy
    - Authentication
    - Cryptography
    - Protect Sensitive Data in transit
    - Protect Sensitive Data at rest
Remediation Techniques

- SDLC – Security Tasks
  - Design
    - Security Reviews Design Documents
  - Development
    - Static Scanning – Developer, Build, Ad-Hoc
  - QA/UAT
    - Dynamic Scanning – Manual and Automated
  - Implementation/Deployment/Operational
    - System Object Security
Remediation Techniques

- **Static Source Scanning**
  - Overview
  - Developer vs Build vs Ad Hoc Scans
  - Issue Management Strategy
  - Hard part is Enterprise Implementation
Remediation Techniques

Static Source Scanning

- Issue Management Strategy
  - Turn on all checkers

- Initial Scan = 50K+ Issues

- Fix top Critical and High

- Defer all others until next version

- No new issues during development
## Remediation Techniques

- **Static Source Scanning**
- **Fortify 360 Demo**

### Results Outline

**Vulnerability Examples by Category**

**Category:** Poor Logging Practice: Use of a System Output Stream (6 Issues)

<table>
<thead>
<tr>
<th>Number of Issues</th>
<th>&lt;Unaudited&gt;</th>
<th>Not an Issue</th>
<th>Reliability Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### HelloWorldJava1.java, line 21 (Poor Logging Practice: Use of a System Output Stream)

<table>
<thead>
<tr>
<th>Fortify Priority: Low</th>
<th>Encapsulation</th>
<th>Folder</th>
<th>Low</th>
</tr>
</thead>
</table>

**Abstract:** Using `System.out` or `System.err` rather than a dedicated logging facility makes it difficult to monitor the behavior of the program.

**Sink:**

```java
HelloWorldJava1.java:21 FunctionCall: println()

19
20
21
int intVar;
System.out.println("Hello World Java 1");
22
System.out.println();
23
System.out.println("Argument 1 = " + args[0]);
```
Remediation Techniques

- **Dynamic Scanning**
  - Manual
    - Intercepting Proxy - Burp
  - Automated
    - IBM Rational AppScan
    - Skipfish – New, Google
      - 404 Generator

- "Give a man an audit and he will be secure for a day. Teach a man to audit and he will be secure for the rest of his life." David Rhoades (SANS)
Remediation Techniques

- Dynamic Scanning
  - Automated - IBM Rational AppScan
Remediation Techniques

- Vulnerability Tracking
  - Open Source Vulnerability Database
    - http://osvdb.org/
  - NIST National Vulnerability Data
    - http://nvd.nist.gov/
  - Secunia
    - http://secunia.com/advisories/search/
  - IBM Internet Security Systems (ISS) X-Force
    - http://xforce.iss.net/
  - Symantec Deep Sight Threat Management
## Vulnerability Tracking

**NIST National Vulnerability Database (NVD)**

### Vulnerability Summary for CVE-2009-2902

- **Original release date:** 01/28/2010
- **Last revised:** 04/01/2010
- **Source:** US-CERT/NIST

### Overview

Directory traversal vulnerability in Apache Tomcat 5.5.0 through 5.5.28 and 6.0.0 through 6.0.20 allows remote attackers to delete work-directory files via directory traversal sequences in a WAR filename, as demonstrated by the ...war filename.

### Impact

**CVSS Severity (version 2.0):**

**CVSS v2 Base Score:** 4.3 (MEDIUM)  
(AV:N/AC:M/Au:N/C:N/I:P/A:N)  
(legend)

**Impact Subscore:** 2.9

**Exploitability Subscore:** 8.6

**CVSS Version 2 Metrics:**

- **Access Vector:** Network exploitable; Victim must voluntarily interact with attack mechanism
- **Access Complexity:** Medium
- **Authentication:** Not required to exploit
- **Impact Type:** Allows unauthorized modification
Remediation Techniques

Training & Awareness

OWASP

SANS/Mitre top 25 Security Errors report
http://www.sans.org/top25-programming-errors/

CERT Secure Coding
https://www.securecoding.cert.org/confluence

Google Browser Security Handbook - Part 1
http://code.google.com/p/browsersec/wiki/Part1

Google Browser Security Handbook - Part 2
http://code.google.com/p/browsersec/wiki/Part2
Remediation Techniques

Training & Awareness
- Create Training & Awareness Courses
- Present to Developers and Architects
- In person
- Content
  - What Security Engineering/Accurance does
  - Regulatory
  - Web Application
  - Defensive Coding
  - Etc.
Remediation Techniques

- **Infrastructure**
  - System Object Security
    - File Access
    - J2EE Container Security

- **Web Application Firewalls**
  - Quick and Easy = 1 week purchase to deployment
  - Need to tune for each Web Application
  - Nice defense in depth posture
  - PCI Approved
OWASP

Open Web Application Security Project (OWASP) - http://www.owasp.org

- Description

- Projects
  - Guides – Development/Testing
  - Tools – WebScarab
  - Education – WebGoat
  - Awareness - Top Ten
Description

The Open Web Application Security Project (OWASP) is a worldwide free and open community focused on improving the security of application software.

- Make application security "visible"
- All materials and software are open source.
Projects

Guides - Development

Architects, developers, consultants and auditors

Comprehensive manual for designing, developing and deploying secure web applications.
OWASP

Projects

Guides - Testing

“Best Practices" penetration testing framework

URL:
Projects

Tools - WebScarab

- Framework for analyzing applications that communicate using HTTP and HTTPS protocols.

- Includes intercepting proxy, spider, fuzzer, etc. Written in Java, portable to many platforms.

URL:
Projects

- Education - WebGoat
  - Deliberately insecure J2EE web application maintained by OWASP

- Web Application security lessons

Projects

- Awareness - Top Ten (2007 & 2010)
  - Powerful awareness document

- Broad consensus about most critical web application security flaws

- Project members include a variety of security experts from around the world

Presentation Message

- Don’t trust the client
- Scary and Dangerous Out There
- No Silver Bullet, Hard Work
Questions?
References

- **OWASP**
  - **Main**
    - [http://www.owasp.org](http://www.owasp.org)
  - **Top Ten**
  - **Cross-Site Scripting**
References

- **OWASP**
  - Output Encoding
  - SQL Injection
  - WebGoat
    - URL:

- **Phoenix Tools**
References

- OWASP
  - Development Guide
  - Testing Guide
    - URL:

- WebScarab
  - URL:
References

» SANS
  » Main
    » http://www.sans.org
  » SANS Report – September 2009
    » http://www.sans.org/top-cyber-security-risks/

» AltoroMutual Test Site
  » http://demo.testfire.net

» RSnake XSS Cheatsheet
  » http://ha.ckers.org/xss.html
References

- **XKCD**

- **GIFAR**
  - Billy Rios
  - How to create a GIFAR
    - [http://riosec.com/how-to-create-a-gifar](http://riosec.com/how-to-create-a-gifar)
  - More on GIFARs and Other Java Smuggling
    - [http://riosec.com/more-on-gifars-and-other-java-smuggling](http://riosec.com/more-on-gifars-and-other-java-smuggling)
References

> SQL Attacks By Example
  > http://www.unixwiz.net/techtips/sql-injection.html

> Top Web Application Attacks 2008/2009
  > 2008
  > http://jeremiahgrossman.blogspot.com
  > 2009
  > http://jeremiahgrossman.blogspot.com
References

- Vulnerability Tracking
  - Open Source Vulnerability Database
    - http://osvdb.org/
  - NIST National Vulnerability Data
    - http://nvd.nist.gov/
- Secunia
  - http://secunia.com/advisories/search/
- IBM Internet Security Systems (ISS) X-Force
  - http://xforce.iss.net/
- Symantec Deep Sight Threat Management
Thank You!