Integrating Web Application Security into the IT Curriculum

James Walden
Northern Kentucky University
Topics

1. Why should we teach web application security?
2. What material do we need to cover?
3. How should we cover that material?
4. Where do we go from here?
Is Web Hacking Really That Easy?

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.

OH, DEAR - DID HE BREAK SOMETHING? IN A WAY-

DID YOU REALLY NAME YOUR SON (Robert'); DROP TABLE Students; -- ?

OH, YES. LITTLE BOBBY TABLES, WE CALL HIM.

WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.

AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

"Exploits of a Mom"
Vulnerability Growth

CERT Vulnerabilities

Year
Web Vulnerabilities Dominate

Web 61%
Application 30%
Operating system 9%

Cross-site scripting 38%
File/code includes 23%
Information disclosure 8%
SQL injection 23%
Authorization 8%
Reasons for Attacking Web Apps

<table>
<thead>
<tr>
<th>Attack Goal</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stealing Sensitive Information</td>
<td>42%</td>
</tr>
<tr>
<td>Defacement</td>
<td>23%</td>
</tr>
<tr>
<td>Planting Malware</td>
<td>15%</td>
</tr>
<tr>
<td>Unknown</td>
<td>8%</td>
</tr>
<tr>
<td>Deceit</td>
<td>3%</td>
</tr>
<tr>
<td>Blackmail</td>
<td>3%</td>
</tr>
<tr>
<td>Link Spam</td>
<td>3%</td>
</tr>
<tr>
<td>Worm</td>
<td>1%</td>
</tr>
<tr>
<td>Phishing</td>
<td>1%</td>
</tr>
<tr>
<td>Information Warfare</td>
<td>1%</td>
</tr>
</tbody>
</table>
Firewalls Don’t Protect Web Apps

- telnet
- ftp
- HTTP Traffic
- Port 80
- Web Client
- Web Server
- Database Server
- Application

SIGITE 2008: 16-18 Oct
Browser Malware Bypasses Firewall

Port Scanning with JavaScript
SPI Dynamics.com - Security Brief

This is a proof of concept page for port scanning arbitrary IP addresses from JavaScript. Given a range of IP addresses, the scanner will detect if there is a host running at that IP. It will then look for a web server running on port 80 and try to fingerprint what kind of web server it is. Only fingerprinting of Microsoft IIS and Apache are currently supported. If the scanner cannot fingerprint the server it will report it as "Unknown webserver. This page will not automatically scan your network, will not attack any hosts it discovers, and will not report any information about your network back to SPI Dynamics.

Known issues with the scanner.

<table>
<thead>
<tr>
<th>IP</th>
<th>Host Exists?</th>
<th>Webserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.100</td>
<td>false</td>
<td>NA</td>
</tr>
<tr>
<td>192.168.1.101</td>
<td>false</td>
<td>NA</td>
</tr>
<tr>
<td>192.168.1.102</td>
<td>false</td>
<td>NA</td>
</tr>
<tr>
<td>192.168.1.103</td>
<td>true</td>
<td>none</td>
</tr>
<tr>
<td>192.168.1.104</td>
<td>false</td>
<td>NA</td>
</tr>
<tr>
<td>192.168.1.105</td>
<td>false</td>
<td>NA</td>
</tr>
<tr>
<td>192.168.1.106</td>
<td>true</td>
<td>none</td>
</tr>
<tr>
<td>192.168.1.107</td>
<td>false</td>
<td>NA</td>
</tr>
<tr>
<td>192.168.1.108</td>
<td>false</td>
<td>NA</td>
</tr>
<tr>
<td>192.168.1.109</td>
<td>false</td>
<td>NA</td>
</tr>
<tr>
<td>192.168.1.110</td>
<td>true</td>
<td>Unknown Webserver</td>
</tr>
</tbody>
</table>
Goals

1. Identify and explain common vulnerabilities.
2. Explain security implications of client-side technologies like Javascript and ActiveX.
3. Detect security vulnerabilities in web applications using appropriate tools.
4. Design and implement web applications that do not contain common vulnerabilities.
5. Deploy and configure a web application in a secure manner.
Topic Outline

1. Web Application Input
2. Client-side Technologies
3. Input-based Attacks
4. Injection Attacks
5. Cross-site Attacks
6. Authentication
7. Secure Programming
8. Operational Security
Web App Security in IT2005

Web Application Security

IPT5
Software Security

WS5
Web Security

IAS6
Security Domains

IAS11
Vulnerabilities

SIGITE 2008: 16-18 Oct
Labs

1. WebGoat exercises on specific vulnerabilities.
2. Using a testing proxy to solve more advanced WebGoat exercises.
3. Assessing an application using a web vulnerability scanner.
4. Assessing a web application using a testing proxy.
5. Reviewing the code of an application using a static analysis tool.
6. Deploying a web application firewall.
7. Participating in the international CTF competition.
WebGoat

Http Basics

Hint
- Show Params
- Show Cookies
- Show HTML
- Show Java

Enter your name in the input field below and press "go" to submit. The server will accept the request, reverse it, and display it back to the user, illustrating the basics of handling an http request.

The user should become familiar with the features of the WebGoat by manipulating the above buttons to show source hint, Java source code, http request parameters, and http request cookies.

Enter your name: [Input Field] Go!
Tools

Web Proxies

- MileSCAN Web Security Auditor
- PortSwigger.net

Web Application Firewalls

- modsecurity
  - www.modsecurity.org

Vulnerability Scanners

- WebInspect
- acunetix

Static Analysis

- FORTIFY SOFTWARE
- Klocwork
- coverity
Web Proxies

![Tamper Data - Ongoing requests](image)

<table>
<thead>
<tr>
<th>Time</th>
<th>Method</th>
<th>Status</th>
<th>Content Type</th>
<th>URL</th>
<th>Load Flags</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Request Header Name</th>
<th>Request Header Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td><a href="http://www.google.com">www.google.com</a></td>
</tr>
<tr>
<td>User-Agent</td>
<td>Mozilla/5.0 (Windows; U; Windows NT 5.1; en...</td>
</tr>
<tr>
<td>Accept</td>
<td>text/xml,application/xml,application/xhtml+xml+...</td>
</tr>
<tr>
<td>Accept-Language</td>
<td>en-us,en;q=0.5</td>
</tr>
<tr>
<td>Accept-Encoding</td>
<td>gzip, deflate</td>
</tr>
<tr>
<td>Accept-Charset</td>
<td>ISO-8859-1,utf-8;q=0.7,*;q=0.7</td>
</tr>
<tr>
<td>Keep-Alive</td>
<td>300</td>
</tr>
<tr>
<td>Connection</td>
<td>keep-alive</td>
</tr>
<tr>
<td>Referrer</td>
<td><a href="http://www.google.com/">http://www.google.com/</a></td>
</tr>
<tr>
<td>Cookie</td>
<td>SID=DQAAAGcAAADHzma8UwXvT_5vhSNQ8...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response Header Name</th>
<th>Response Header Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Found - 302</td>
</tr>
<tr>
<td>Location</td>
<td><a href="http://www.google.com/webhp?hl=en&amp;btnG">http://www.google.com/webhp?hl=en&amp;btnG</a>...</td>
</tr>
<tr>
<td>Cache-Control</td>
<td>private</td>
</tr>
<tr>
<td>Content-Type</td>
<td>text/html</td>
</tr>
<tr>
<td>Server</td>
<td>GWS/2.1</td>
</tr>
<tr>
<td>Transfer-Encoding</td>
<td>chunked</td>
</tr>
<tr>
<td>Content-Encoding</td>
<td>gzip</td>
</tr>
<tr>
<td>Date</td>
<td>Tue, 06 Mar 2007 17:46:32 GMT</td>
</tr>
</tbody>
</table>
Altering Form Parameters

![Tamper Popup window displaying form parameters](image)
Fuzz testing consists of

- Sending unexpected input.
- Monitoring for exceptions.
Web Application Firewalls

What is a WAF?
- Web monitoring.
- Access control.
- Behind SSL endpoint.

A/K/A
- Deep packet inspection.
- Web IDS/IPS.
- Web App Proxy/Shield.

mod_security
- Open source.
- Embeds in Apache.
- Reverse proxy.
Vulnerability Scanners

1. Spiders site.
2. Identifies inputs.
3. Sends list of malicious inputs to each input.
Static Analysis

Automated assistance for code auditing
Speed: review code faster than humans can
Accuracy: hundreds of secure coding rules

Tools
- Coverity
- FindBugs
- Fortify
- Klocwork
- Ounce Labs
Labs

- WebGoat exercises on specific vulnerabilities.
- Using a **testing proxy** to solve more advanced WebGoat exercises.
- Assessing an application using a web **vulnerability scanner**.
- Assessing a web application using a **testing proxy**.
- Reviewing the code of an application using a **static analysis** tool.
- Deploying a web application **firewall**.
- Participating in the international CTF competition.
Approaches

1. **Students evaluate and fix their own code.**
   - Students learn about their own coding mistakes.
   - Scale of project limited to what students can write.

2. **Students evaluate and fix your code.**
   - Write a web application designed for teaching students.

3. **Students evaluate and fix someone else’s code.**
   1. Use a web application designed for teaching.
   2. Analyze an open source web application with known vulnerabilities reported in NVD or other bug db.
Teaching Applications
Asynchronous Javascript and XML

- Expanded server side API.
- Server API calls can be issued in any order by attacker; cannot assume calls issued in order by your client.
- Larger amount of client state.
- Client/server communication using data (XML/JSON) rather than presentation (HTML.)
Future Directions: Web Sec Class

1. Web Application Input
2. Client-side Technologies
3. Service Oriented Architectures
4. AJAX
5. Input-based Attacks
6. Injection Attacks
7. Race Conditions
8. Cross-site Attacks
9. Authentication
10. Secure Programming
11. Operational Security
Conclusions

1. Defense is shifting from network to application layer.
   →
   Firewalls, anti-virus, SSL input validation, WAF

2. Students need to learn to identify vulnerabilities.
   ■ Static analysis of source code.
   ■ Web proxies and scanners for testing.

3. Students need to learn to remediate vulnerabilities.
   1. Web application firewalls for immediate short-term fixes.
   2. Repairing source code for long term fixes.