Web Application Security: Exercise Development Approaches

James Walden

waldenj@nku.edu

Approaches

1. Write your own web application

   Students evaluate and fix your code.

2. Students write a web application

   Students evaluate and fix their own code.

- Construct exercises with 3rd party tools

   i. Use a web security teaching tool (WebGoat)

   ii. Use a web application designed for learning about security (BadStore)

   iii. Analyze an open source web application with known vulnerabilities.

Tools for Exercises

Browser Plugins

- Modify HTTP headers + form parameters.
- Examples: Tamper Data for Firefox

Proxy Suites

- Modify parameters +
  - Spidering
  - Fuzz testing.
  - Session key analysis.
  - Decoding.
- Examples: Burp Suite, Paros Proxy, WebScarab

Static + Dynamic Analysis

Write your own web application

Most flexible approach.

Also the most time-consuming.

Can be used for

- Individual vulnerability education
- Penetration testing exercise
- Pen test + code maintenance exercise
- Framework for students to build upon.

My web applications

BlogEngine: PHP-based blog application with many types of vulnerabilities including access ctl, dir traversal, SQL injection, XSS.

SQL Injection Demos: Perl-based SQL injection demonstrations, with 2 vulnerable perl CGI scripts, 3 fixed CGI scripts with different approaches to fixing.

Distribution Issues

1. Compatibility

   Can the application run on students' PCs?

2. Permissions

   Do students have rights to install + run?

3. Security

   If students can hack app, so can others.

   Need to isolate insecure app from Internet.

Distribution Solutions

Virtual Machines

- VM environment identical for all students.
- VM can be isolated to host-only network.
- VMWare Player free for Linux + Windows
- Used for SQL injection demos.

XAMPP

- Apache + MySQL + PHP + Perl
- Easy to install distribution
- Linux, Windows, Mac OS X, Solaris
- Used for BlogEngine.

Students write a web application

Advantages

- Students see what bugs *they* write.
- Compare different implementations of app.
- Good technique for integrating into SwEng.

Disadvantages

- Cannot predict vulnerabilities in advance.
- Limited by time students have to develop.

Exercises

Abuse Cases

- Use attack patterns to create abuse cases.

Architectural Risk Analysis

- Draw + review DFDs for application.
- Risk analysis based on DFDs + abuse cases.
- Most useful after first iteration.

Code Review + Static Analysis

- Use Fortify SCA to analyze source code.
- Code review: moderator, author

Penetration Testing

- Find bugs in their own or another group's project.

Exercises with 3rd party tools

1. Use a web security teaching tool
   - Exercises for specific vulnerabilities.
   - May include hints, completion tracking.
2. Use a web application designed for learning about security
   - Application designed with vulnerabilities.
   - Vary based on web platform, vuln types.

- Analyze an open source web application with known vulnerabilities.

Web Security Teaching Tools

WebGoat

- GPL J2EE teaching application

Hack This Site

- Online security exercises, incl web security.

NTO Hackme Site

- Only two live lessons (XSS and SQL inject)

Using Web Security Teaching Tools

Focus on a single vulnerability

- Learn about single vulnerability in isolation.
- No need to understand entire application.

Useful for

- In-class demonstrations of vulnerabilities.
- Single vulnerability assignments.
- Multi-vulnerability assignments for classes that have only a single unit on web security.

Web Security Demo Apps

BadStore

- GPL shopping app available as ISO image

Hacme Bank, Books, and Travel

- J2EE, MS, and C++ apps for pen testing

WebMaven (aka Buggy Bank)

- GPL bank app, MS install instructions only

International Capture the Flag

- Annual competition focusing on web apps.

Using Web Security Demo Apps

Focus on penetration testing

- Broad range of web vulnerabilities.
- Requires > effort & skill than teaching tools

Advantages

- Whole application security perspective.
- Provide a more authentic experience.

Useful for

- Penetration testing assignments (find 10 vulnerabilities in the next week.)

Using Open Source Web Apps

Focus on testing and fixing vulnerabilities

- Not as many known vulnerabilities.
- May take effort to find insecure versions.
- Provides a more authentic experience.

Useful for

- Penetration testing assignments.
- Code maintenance assignments.
- Static and dynamic analysis assignments.

Key Points

Write your own web application

- Flexible but time-consuming approach.

Student-written applications

- Assignments throughout the SDLC.
- Cannot predict vulnerabilities in advance.

Third party applications

- Use WebGoat to teach about vulnerabilities.
- Use BadStore to teach about vulnerabilities in semi-authentic context, penetration testing.
- Open source to teach about authentic vulnerabilities.