Web Application Security

David Epler
Sr. Software Developer
depler@aboutweb.com

About Me

- Application Developer
- Web Application Security Enthusiast
- OWASP Individual Member
- Created Unofficial Updater 2 to patch Adobe ColdFusion 8.0.1 & 9.0.1

About the Demos

- Virtual Machines, not real servers
  - OWASP Broken Web Apps
  - Samurai Web Testing Framework

- **DO NOT** perform any activities shown on any network/system or on a network connected device without proper permission!

OWASP Top Ten (2010)

A1: Injection

A2: Cross-Site Scripting (XSS)

A3: Broken Authentication and Session Management

A4: Insecure Direct Object References

A5: Cross Site Request Forgery (CSRF)

A6: Security Misconfiguration

A7: Insecure Cryptographic Storage

A8: Failure to Restrict URL Access

A9: Insufficient Transport Layer Protection

A10: Unvalidated Redirects and Forwards

OWASP Top Ten (2010)

A1: Injection

A2: Cross-Site Scripting (XSS)

A3: Broken Authentication and Session Management

A4: Insecure Direct Object References

A5: Cross Site Request Forgery (CSRF)

A6: Security Misconfiguration

A7: Insecure Cryptographic Storage

A8: Failure to Restrict URL Access

A9: Insufficient Transport Layer Protection

A10: Unvalidated Redirects and Forwards

In the News

- March 27 - FTC fines RockYou $250,000 for storing user data in plain text
  - http://www.zdnet.com/blog/security/ftc-fines-rockyou-250000-for-storing-user-data-in-plain-text/11274
- June 6 - 6.46 million LinkedIn passwords leaked online
  - http://www.zdnet.com/blog/btl/6-46-million-linkedin-passwords-leaked-online/79290
- June 7 - eHarmony admits to leaking 1.5 million passwords
  - http://www.h-online.com/security/news/item/eHarmony-admits-to-leaking-1-5-million-passwords-1612654.html
- July 12 - Hackers expose 453,000 credentials allegedly taken from Yahoo service
  - http://arstechnica.com/security/2012/07/yahoo-service-hacked/
- July 23 - Eight Million Email Addresses And Passwords Spilled From Gaming Site Gamigo Months After Hacker Breach
  - http://www.forbes.com/sites/andygreenberg/2012/07/23/eight-million-passwords-spilled-from-gaming-site-gamigo-months-after-breach/
- Along with Last.fm, Phandroid, Billabong, Formspring, and Nvidia

A1 - Injection

- Tricking an application into including unintended commands in the data sent to an interpreter
  - SQL, LDAP, XPath, OS Shell
  - Impact
    - Usually severe, entire database can usually be read or modified
    - May also allow full database schema, or account access, or even OS level access

A1 - Injection (SQLi)

- Stacked Queries
  - http://www.victim.com/products.asp?id=1;exec +master..xp_cmdshell+'dir'
- Tautology
  - http://www.victim.com/logon.aspx? username=admin' or 1=1;--
- UNION Statements
  - http://www.victim.com/products.asp? id=12+UNION+SELECT +userid,first_name,second_name,password +FROM+customers

Applications Places System

Firefox Web Browser
Browse the World Wide Web

Sat Aug 4, 9:03 PM

A1 - Injection

- Avoid the interpreter completely (if possible)
- Use an interface that supports bind variables
  - Prepared Statements, Stored Procedures
- Encode all user input before passing it to the interpreter
  - Always perform “white list” input validation on all user supplied input
- Minimize database privileges to reduce the impact of a flaw

- https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet
- https://www.owasp.org/index.php/Query_Parameterization_Cheat_Sheet

A7 - Insecure Cryptographic Storage

- Not encrypting data that deserves encryption
- Use of weak or unsalted hashes to protect passwords
- Unsafe key generation and storage, not rotating keys, and weak algorithm usage
- Impact
  - Failure frequently compromises all data that should have been encrypted. Typically this information includes sensitive data

SQLmap

File Edit View Terminal Help

```
    Title: MySQL UNION query (NULL) - 5 columns
    Payload: username=' LIMIT 1,1 UNION ALL SELECT NULL, NULL, CONCAT(0x3a7176723a,0x537766465259656a5961,0x3a766e743a
), NULL, NULL#&password=&user-info-php-submit-button=View Account Details
---
[21:06:14] [INFO] the back-end DBMS is MySQL
[21:06:14] [INFO] fetching banner
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14
back-end DBMS operating system: Linux Ubuntu
back-end DBMS: MySQL 5.0
banner:    '5.1.41-3ubuntu12.6-log'
[21:06:14] [INFO] fetching current user
current user:    'mutillidae@%'
[21:06:15] [INFO] fetching current database
current database:    'mutillidae'
[21:06:16] [INFO] testing if current user is DBA
[21:06:16] [INFO] fetching current user
current user is DBA:    False
[21:06:16] [INFO] fetched data logged to text files under '/usr/bin/samurai/sqlmap/output/owaspbwa'

[*] shutting down at 21:06:16

samurai@samurai-wtf:/usr/bin/samurai/sqlmap$
samurai@samurai-wtf:/usr/bin/samurai/sqlmap$
```

SAMURAI WEB TESTING FRAMEWORK

HTTP://SAMURAI.INGUARDIANS.COM

A7 - Insecure Cryptographic Storage

- Considering the threats you plan to protect this data from, make sure you encrypt all such data at rest in a manner that defends against these threats
- Ensure all keys and passwords are protected from unauthorized access
- Ensure offsite backups are encrypted, but the keys are managed and backed up separately
- Ensure appropriate strong standard algorithms and strong keys are used, and key management is in place
- Ensure passwords are hashed with a strong standard algorithm and an appropriate salt is used

- https://www.owasp.org/index.php/Cryptographic_Storage_Cheat_Sheet
- https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet

OWASP Top Ten (2010)

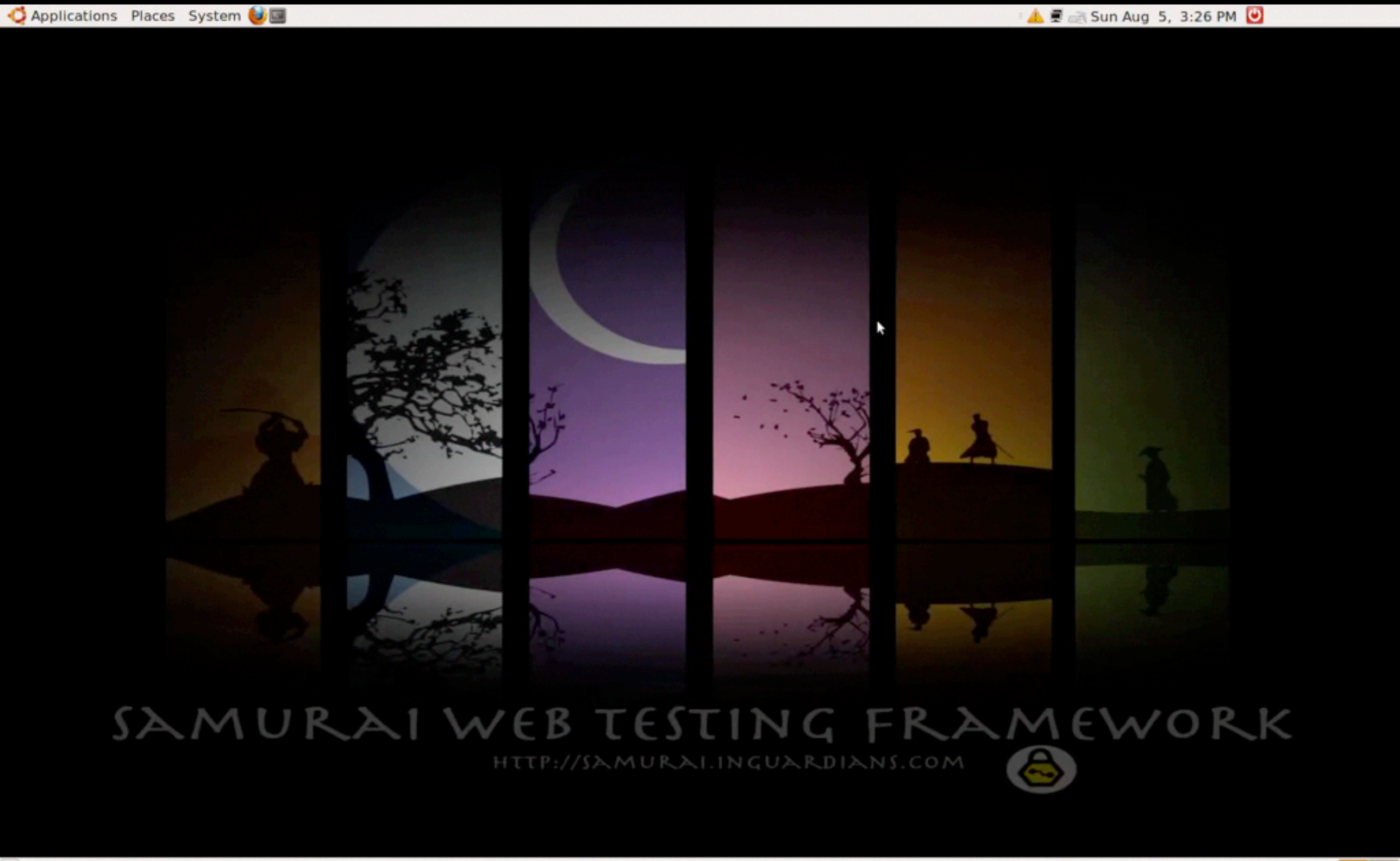
A1: Injection	**A2: Cross-Site Scripting (XSS)**	A3: Broken Authentication and Session Management	A4: Insecure Direct Object References
A5: Cross Site Request Forgery (CSRF)	A6: Security Misconfiguration	A7: Insecure Cryptographic Storage	A8: Failure to Restrict URL Access
	A9: Insufficient Transport Layer Protection	A10: Unvalidated Redirects and Forwards	

A2 - Cross-Site Scripting (XSS)

- Raw data from attacker is sent to an innocent user's browser
- Three known types: Stored, Reflected, DOM Based
- Impact
  - Steal user's session, steal sensitive data, rewrite web page, redirect user to phishing or malware site
  - Most Severe: Install XSS proxy which allows attacker to observe and direct all user's behavior on vulnerable site and force user to other sites

A2 - Cross-Site Scripting (XSS)

- Stored
  - Attacker's script is stored on the server (e.g. blog comments, forums) and later displayed in HTML pages, without proper filtering
- Reflected
  - HTML page reflects user input data back to the browser, without sanitizing the response
- DOM Based


Applications
Places
System
Sun Aug 5, 3:26 PM
SAMURAI WEB TESTING FRAMEWORK
HTTP://SAMURAI.INGUARDIANS.COM

A2 - Cross-Site Scripting (XSS)

- Eliminate
  - Don’t include user supplied input in the output page
- Defend
  - Output encode all user supplied input with proper encoder
  - Always perform “white list” input validation on all user supplied input included in the page
  - Use OWASP’s AntiSamy to sanitize user supplied HTML to make it safe

- https://www.owasp.org/index.php/XSS_Prevention_Cheat_Sheet
- https://www.owasp.org/index.php/DOM_based_XSS_Prevention_Cheat_Sheet

A5 - Cross-Site Request Forgery (CSRF)

- Attacker creates forged HTTP requests and tricks a victim into submitting them via image tags, XSS, or numerous other techniques
- Impact
  - Can cause victims to change any data the victim is allowed to change or perform any function the victim is authorized to use

A5 - Cross-Site Request Forgery (CSRF)

- Real World Example: Netflix 2006
  -

A5 - Cross-Site Request Forgery (CSRF)

- Real Wo 2006
  - <img src="h e?
    movieid=70

A5 - Cross-Site Request Forgery (CSRF)

- Add a secret, not automatically submitted, token to ALL sensitive requests
  - This makes it impossible for the attacker to spoof the request
    - unless there's an XSS hole in your application
    - Tokens should be cryptographically strong or random
- Store a single token in the session and add it to all forms and links
  - Hidden Field, Single use URL, Form Token
- Can have a unique token for each function
  - Use a hash of function name, session id, and a secret
- Can require secondary authentication for sensitive functions

- https://www.owasp.org/index.php/CSRF_Prevention_Cheat_Sheet

OWASP Top Ten (2010)

A1: Injection	A2: Cross-Site Scripting (XSS)	A3: Broken Authentication and Session Management	A4: Insecure Direct Object References
A5: Cross Site Request Forgery (CSRF)	A6: Security Misconfiguration	A7: Insecure Cryptographic Storage	A8: Failure to Restrict URL Access
	A9: Insufficient Transport Layer Protection	A10: Unvalidated Redirects and Forwards	

OWASP Top Ten (2010)

A1: Injection

A2: Cross-Site Scripting (XSS)

A3: Broken Authentication and Session Management

A4: Insecure Direct Object References

A5: Cross Site Request Forgery (CSRF)

A6: Security Misconfiguration

A7: Insecure Cryptographic Storage

A8: Failure to Restrict URL Access

A9: Insufficient Transport Layer Protection

A10: Unvalidated Redirects and Forwards

So should you just turn everything off and unplug it?

OWASP Enterprise Security API

- ESAPI is NOT a framework
  - Collection of security building blocks
  - Designed to help retrofit existing applications
- Available for multiple languages
  - Java, .NET, Perl, PHP, ColdFusion
- Security Frameworks already exist
  - Spring Security
  - ASP.NET Web Application Security

ESAPI baked into CFML

- Adobe ColdFusion 10 uses ESAPI 2.0.1
  - Canonicalize, DecodeForHTML, DecodeFromURL, EncodeForCSS, EncodeForHTML, EncodeForHTMLAttribute, EncodeForJavaScript, EncodeForURL, EncodeForXML
- With APSB11-04 and higher ESAPI installed into ColdFusion
  - ESAPI 1.4.4 for CF 8.0.x
  - ESAPI 2.0_rc10 for CF 9.0.x
- Railo 4.0 Beta
  - ESAPI* functions

- CFESAPI project by Damon Miller
  - https://github.com/damonmiller/cfesapi
- CFBackPort project by David Boyer
  - https://github.com/misterdai/cfbackport

Web Application Firewall

- Web application firewall (WAF) are used to protect web applications without the need to modify them
  - Can be an appliance, server plugin, or filter
- Commercial
  - Trustwave - WebDefend Web Application Firewall
  - Cisco - ACE Web Application Firewall
  - Citrix - NetScaler App Firewall
  - F5 - BIG-IP Application Security Manager
  - Privacyware - ThreatSentry IIS Web Application Firewall
- Free
  - Trustwave SpiderLabs - ModSecurity
  - Microsoft - URLScan 3.1

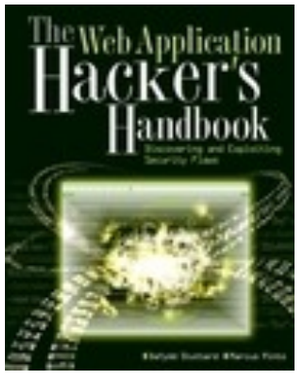
ColdFusion Application Firewall/Filters

- Fuseguard by Foundeo
  - See Mike Henke's presentation for more details
    - http://henke.ws/post.cfm/web-application-firewall-and-fuseguard-presentation-tomorrow
- Portcullis by John Mason
  - http://portcullis.riaforge.org/

ModSecurity

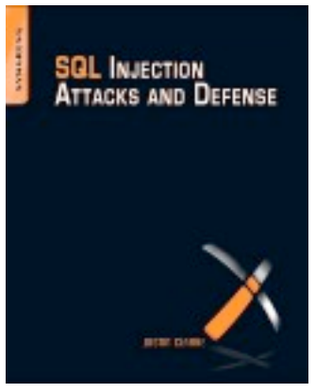
- Open source, free web application firewall (WAF) Apache module
- Security Models
  - Negative Security Model
  - Positive Security Model
  - Virtual Patching
  - Extrusion Detection Model
- Run as Apache module or reverse proxy
- OWASP ModSecurity Core Rule Set Project

Books



The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Second Edition
by Dafydd Stuttard and Marcus Pinto
John Wiley & Sons © 2012 (912 pages)
ISBN: 9781118026472

Web Application Obfuscation: '-/WAFs..dEvasion..dFilters//alert (/Obfuscation/)-'
by Mario Heiderich, Eduardo Alberto Vela Nava, Gareth Heyes and David Lindsay
Syngress Publishing © 2011 (290 pages)
ISBN: 9781597496049



SQL Injection Attacks and Defense
by Justin Clarke
Syngress Publishing © 2009 (494 pages)
ISBN: 9781597494243

XSS Attacks: Cross Site Scripting Exploits and Defense
by Jeremiah Grossman, Robert "RSnake" Hansen, Petko "pdp" D. Petkov and Anton Rager
Syngress Publishing © 2007 (479 pages)
ISBN: 9781597491549

Penetration Tester's Open Source Toolkit, Third Edition
by Jeremy Faircloth
Syngress Publishing © 2011 (465 pages)
ISBN: 9781597496278

Seven Deadliest Web Application Attacks
by Mike Shema
Syngress Publishing © 2010 (187 pages)
ISBN: 9781597495431

References

- http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202010.pdf
- http://owasptop10.googlecode.com/files/OWASP_Top_10_-_2010%20Presentation.pptx
- http://owasp-esapi-java.googlecode.com/files/OWASP%20ESAPI.ppt
- https://www.owasp.org/index.php/Category:OWASP_AntiSamy_Project
- https://www.owasp.org/index.php/Cheat_Sheets
- http://news.cnet.com/Netflix-fixes-Web-2.0-bugs/2100-1002_3-6126438.html?part=rss&tag=6126438&subj=news
- http://sourceforge.net/apps/mediawiki/mod-security/index.php?title=FAQ
- http://www.owasp.org/images/2/21/OWASPAppSec2007Milan_ModSecurityCoreRuleSet.ppt
- https://www.owasp.org/index.php/Category:OWASP_Securing_WebGoat_using_ModSecurity_Project
- http://www.akamai.com/html/about/press/releases/2009/press_121409.html
- http://sourceforge.net/projects/samurai/
- https://www.owasp.org/index.php/OWASP_Broken_Web_Applications_Project
- http://blog.taddong.com/2011/03/browser-exploitation-for-fun-profit.html