DENIM GROUP

build | integrate | secure

Grow with Denim Group

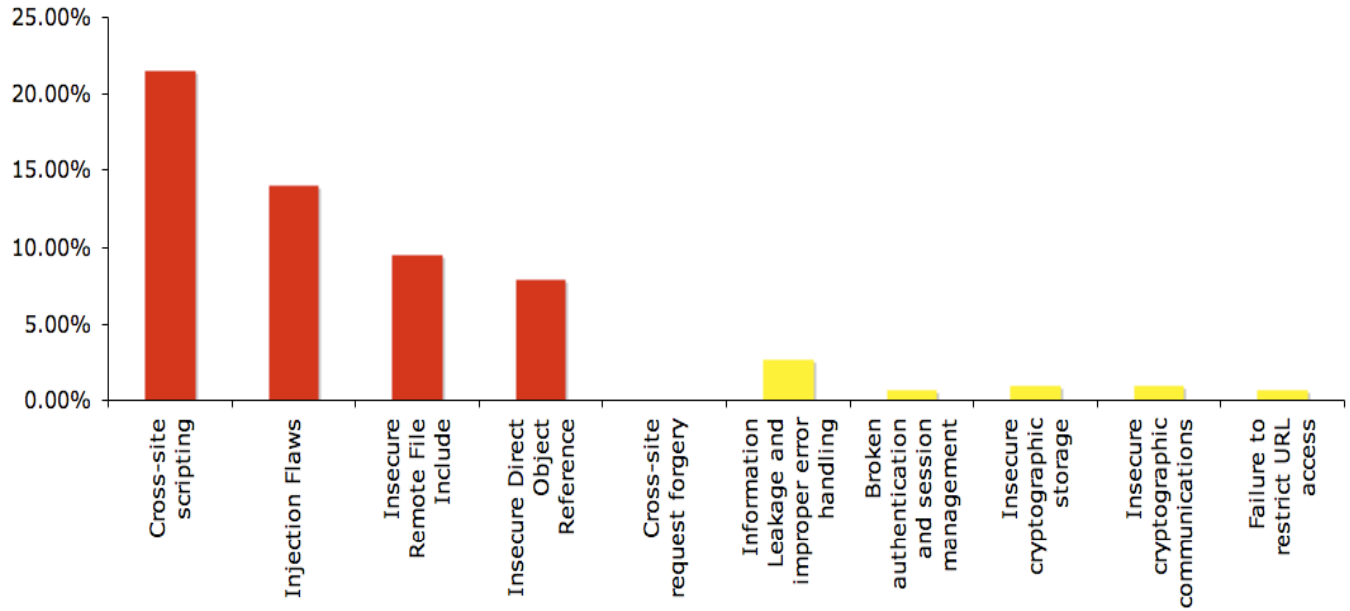
OWASP Top Ten:
Evolution from 2004 to 2007

What Is In Store For the OWASP Top Ten?

- Now based on MITRE vulnerability statistics
- Focus is on Vulnerabilities (not Attacks)
- Items being removed:
  - *Unvalidated Input*
  - *Buffer Overflows*
  - *Denial of Service*
  - *Insecure Configuration Management*
- New items being added
  - *Insecure Remote File Include*
  - *Cross Site Request Forgery (CSRF)*
  - *Insecure Communications*
- Broken Access Control split into two items:
  - *Insecure Direct Object Reference*
  - *Failure to Restrict URL Access*

Based on MITRE Statistics

- MITRE Vulnerability Trends 2006
  - cwe.mitre.org/documents/vuln-trends.html



Focus on Vulnerabilities Not Attacks

- Previous Top 10 was a mixture of vulnerabilities, attacks and countermeasures
- Now the focus is specifically on vulnerabilities
- These vulnerabilities may *support* attacks
- Examples of Attacks
  - *Phishing*
  - *Privacy violations*
  - *Identity theft*
  - *System compromise*
  - *Financial loss through unauthorized transactions*
  - *Reputation loss*

Mapping: 2007 to 2005

OWASP Top 10 2007	OWASP Top 10 2004	MITRE 2006 Raw Ranking
A1. Cross Site Scripting (XSS)	A4. Cross Site Scripting (XSS)	1
A2. Injection Flaws	A6. Injection Flaws	2
A3. Insecure Remote File Include (NEW)		3
A4. Insecure Direct Object Reference	A2. Broken Access Control (split in 2007 T10)	5
A5. Cross Site Request Forgery (CSRF) (NEW)		36
A6. Information Leakage and Improper Error Handling	A7. Improper Error Handling	6
A7. Broken Authentication and Session Management	A3. Broken Authentication and Session Management	14
A8. Insecure Cryptographic Storage	A8. Insecure Storage	8
A9. Insecure Communications (NEW)	Discussed under A10. Insecure Configuration Management	8
A10. Failure to Restrict URL Access	A2. Broken Access Control (split in 2007 T10)	14
	A1. Unvalidated Input	7
	A5. Buffer Overflows	4, 8, and 10
	A9. Denial of Service	17
	A10. Insecure Configuration Management	29

Items Being Removed

- Unvalidated Input
- Buffer Overflows
- Denial of Service
- Insecure Configuration Management

Unvalidated Input

- This was actually a superclass of several other vulnerabilities
  - *Injection Flaws*
    - Command Injection
    - SQL Injection
  - *Cross Site Scripting*
  - *Buffer Overflows*

Buffer Overflows

- Hard to gather statistics for this in web environments
- Considered to be more of a network or infrastructure security issue
- Still important because of web connections to legacy code
  - *COM objects*
  - *JNI calls*

Denial of Service

- Serious issues with web applications
- However MITRE ranking did not merit inclusion at the time of the Top 10 revision

Insecure Configuration Management

- Hard to gather statistics on this
- Considered to be more of a process issue rather than a web application vulnerability

Items Being Added

- Insecure Remote File Include
- Cross Site Request Forgery (CSRF)
- Insecure Communications

Insecure Remote File Include

- Mostly based on issues with PHP
  - *Include $_REQUEST['filename'];*
- Can also affect other web environments if developers are not careful
  - *Document or file management systems that allow for file uploads*

Cross Site Request Forgery (CSRF)

- Similar to Cross Site Scripting (XSS)
- Inject HTML or JavaScript to make rogue requests
  - *XMLHttpRequest calls*
  - *<img src=http://site/target.aspx?param=value>*

Guarding Against CSRF

- Protect against Cross Site Scripting (XSS) attacks
- Use the ViewStateUserKey Page property

```
void Page_Init (object sender, EventArgs e) {
   ViewStateUserKey = Session.SessionID;
}
```

Insecure Communications

- HTTP traffic can be sniffed off a local network or otherwise when in transit
- Use SSL for all sensitive communications
  - *Obviously use HTTPS for access to sensitive web pages*
  - *Also think about using it for back-end communications*
    - Database access
    - Legacy integration
    - Web services

Changes to Broken Access Control

- Split into two items
  - *Insecure Direct Object Reference*
  - *Failure to Restrict URL Access*

Insecure Direct Object Reference

- Objects are referenced directly with no authorization checks being applied
- Examples:
  - *DocumentID*
  - *AccountID*
  - *StatementID*
  - *And so on*
- Check GET and POST parameters
- Could also occur due to use of cookies

Failure to Restrict URL Access

- This was typically what was discussed as broken access control
- Prevent using platform-specific Authentication/Authorization primitives
  - *ASP.NET*
  - *JEE*

For More Information

- Main OWASP site: www.owasp.org
- OWASP Top 10 site: www.owasp.org/index.php/OWASP_Top_Ten_Project

Questions

Dan Cornell
dan@denimgroup.com
(210) 572-4400

Web: www.denimgroup.com
Blog: denimgroup.typepad.com