OWASP Top Ten (2010) Tools and Tactics

Russ McRee

Welcome

- Director, Security Analytics for Microsoft Online Services Security & Compliance
  - Writer (toolsmith), researcher (holisticinfosec), presenter (Defcon, Black Hat, RSA), SANS GSE and ISC handler
- We see MANY web application security flaws
  - 50,000 unique MS-related domains
- These tools and tactics help us assess and defend too

OWASP Top Ten Tools and Tactics

- The Open Web Application Security Project
  - All attack and defense content is drawn directly from the OWASP Top 10 Wiki*
- A tool for each of the Top Ten to aid in discovering and remediating each of the OWASP Top 10
- We'll try to demo as many of these tools as time allows

Lessons from WhiteHat Security report

- **Lesson 1:**
  - Software will always have bugs and by extension, security vulnerabilities. Therefore, a practical goal for a secure software development lifecycle (SDLC) should be to reduce, not necessarily eliminate, the number of vulnerabilities introduced and the severity of those that remain.*

* from the 11th edition of the WhiteHat Security Website Security Statistic Report

Lessons from WhiteHat Security report

- **Lesson 2:**
  - Exploitation of just one website vulnerability is enough to significantly disrupt online business, cause data loss, shake customer confidence, and more. Therefore, the earlier vulnerabilities are identified and the faster they are remediated the shorter the window of opportunity for an attacker to maliciously exploit them.*

* from the 11th edition of the WhiteHat Security Website Security Statistic Report

OWASP Top Ten Web Application Security Risks

- **A1: Injection:**
  - Injection flaws, such as SQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query.
- **A2: Cross-Site Scripting (XSS)**
  - XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation and escaping.
- **A3: Broken Authentication and Session Management**
  - Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, session tokens, or exploit other implementation flaws to assume other users' identities.
- **A4: Insecure Direct Object References**
  - A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key.
- **A5: Cross-Site Request Forgery (CSRF)**
  - A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application.

OWASP Top Ten Web Application Security Risks

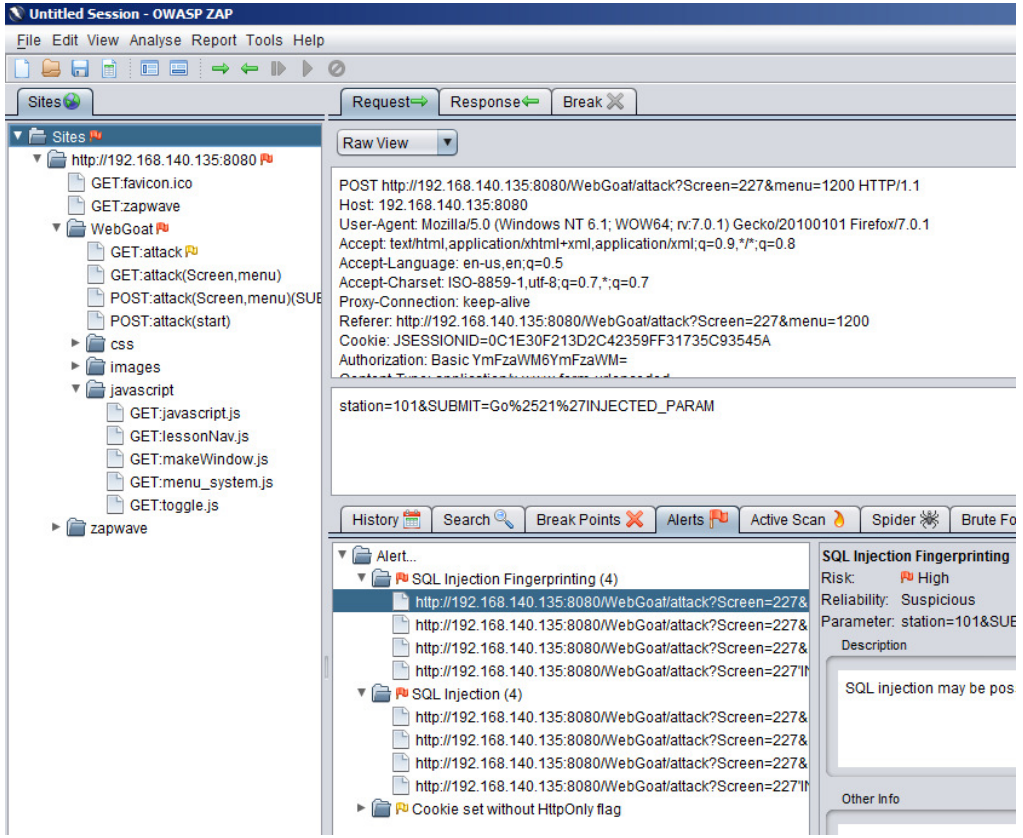
- **A6: Security Misconfiguration**
  - Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform.
- **A7: Insecure Cryptographic Storage**
  - Many web applications do not properly protect sensitive data, such as credit cards, SSNs, and authentication credentials, with appropriate encryption or hashing.
- **A8: Failure to Restrict URL Access**
  - Many web applications check URL access rights before rendering protected links and buttons.
- **A9: Insufficient Transport Layer Protection**
  - Applications frequently fail to authenticate, encrypt, and protect the confidentiality and integrity of sensitive network traffic.
- **A10: Unvalidated Redirects and Forwards**
  - Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages.

OWASP Top Ten Tools

• A1: Injection –	**ZAP**
• A2: Cross-Site Scripting (XSS) -	**BeEF**
• A3: Broken Authentication and Session Management -	**HackBar**
• A4: Insecure Direct Object References -	**Burp Suite**
• A5: Cross-Site Request Forgery (CSRF) –	**Tamper Data**
• A6: Security Misconfiguration –	**Watobo**
• A7: Insecure Cryptographic Storage	N/A
• A8: Failure to Restrict URL Access -	**Nikto/Wikto**
• A9: Insufficient Transport Layer Protection -	**Calomel**
• A10: Unvalidated Redirects and Forwards –	**Watcher**

A1: Injection – ZAP (Zed Attack Proxy)

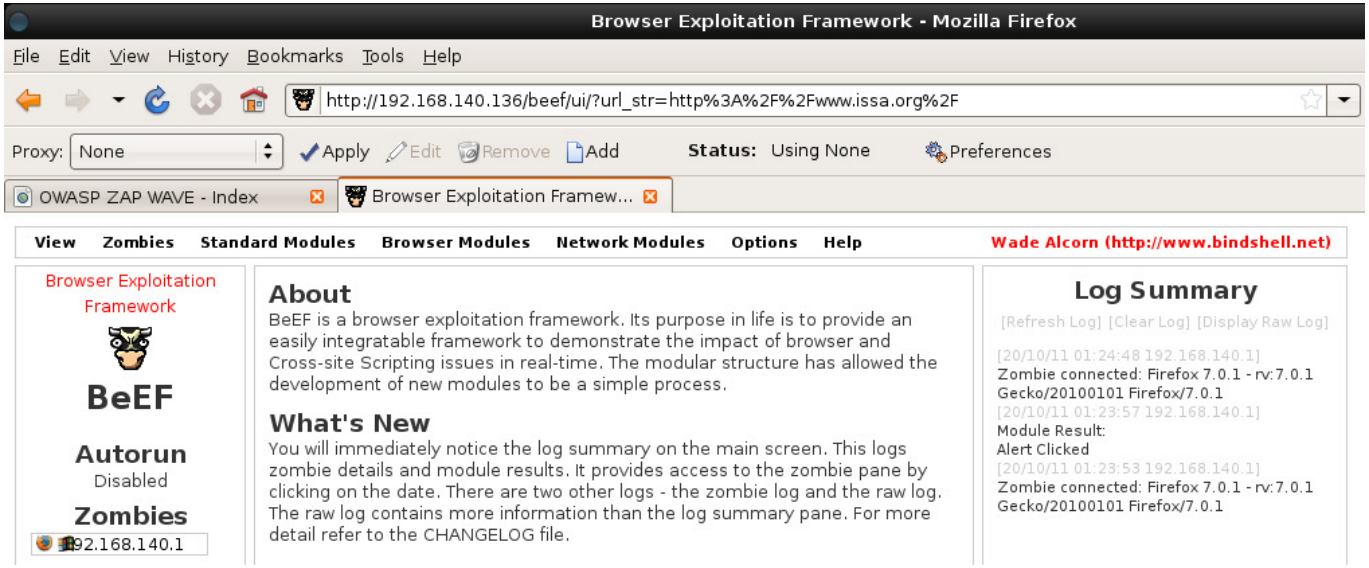
- Attack:
  - The attacker's hostile data can trick the interpreter into executing unintended commands or accessing unauthorized data
- Defense:
  - Prepared Statements (Parameterized Queries)
  - Use of Stored Procedures
  - Escaping all User Supplied Input



- 2011 Toolsmith Tool of the Year

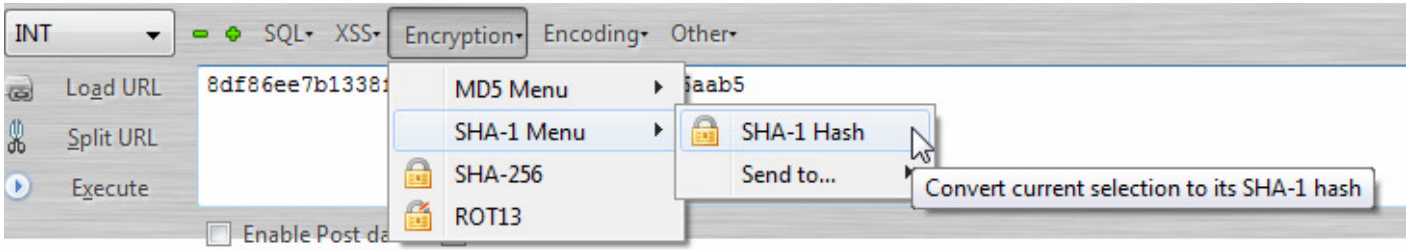
A2: Cross-Site Scripting (XSS) - BeEF

- Attack:
  - Allows attackers to execute scripts in victim's browser, can hijack user sessions, deface web sites, or redirect user to malicious sites
- Defense:
  - Escape all untrusted data based on the HTML context
  - Positive or "whitelist" input validation
  - Current browser, NoScript



A3: Broken Authentication & Session Management - Hackbar

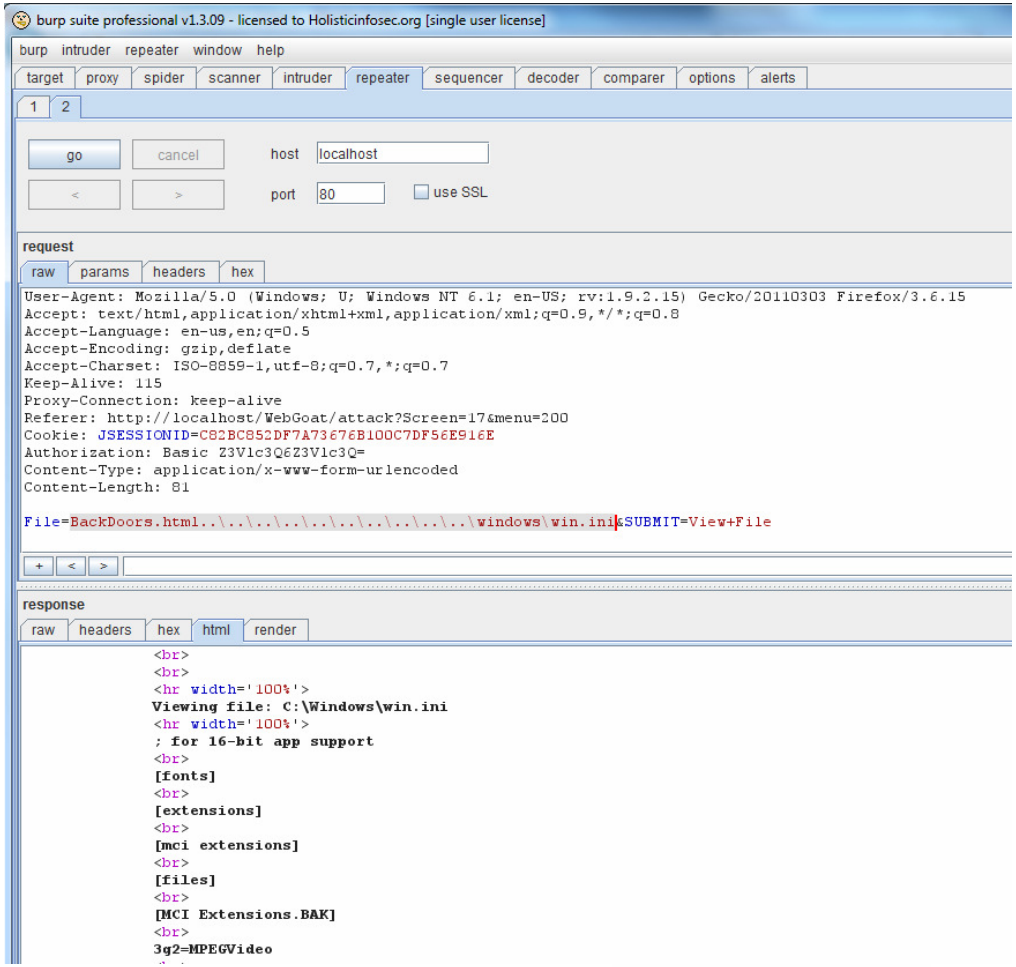
- Attack:
  - Allows attackers to compromise passwords, keys, session tokens, or exploit other implementation flaws to assume other users' identities
- Defense:
  - A single set of strong authentication and session management controls
    - ESAPI



- "I implore everybody to migrate to a stronger password scrambler without undue delay."
  - MD5 developer Poul-Henning Kamp

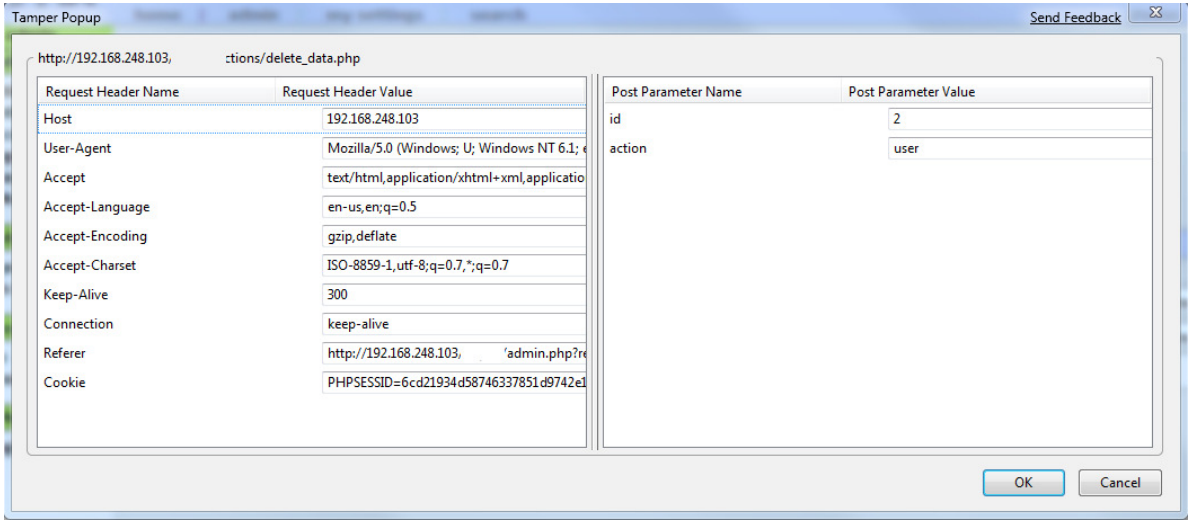
A4: Insecure Direct Object References - Burp

- Attack:
  - Without an access control check or other protection, attackers can manipulate references to access unauthorized data
- Defense:
  - Use per user or session indirect object references
  - Check access



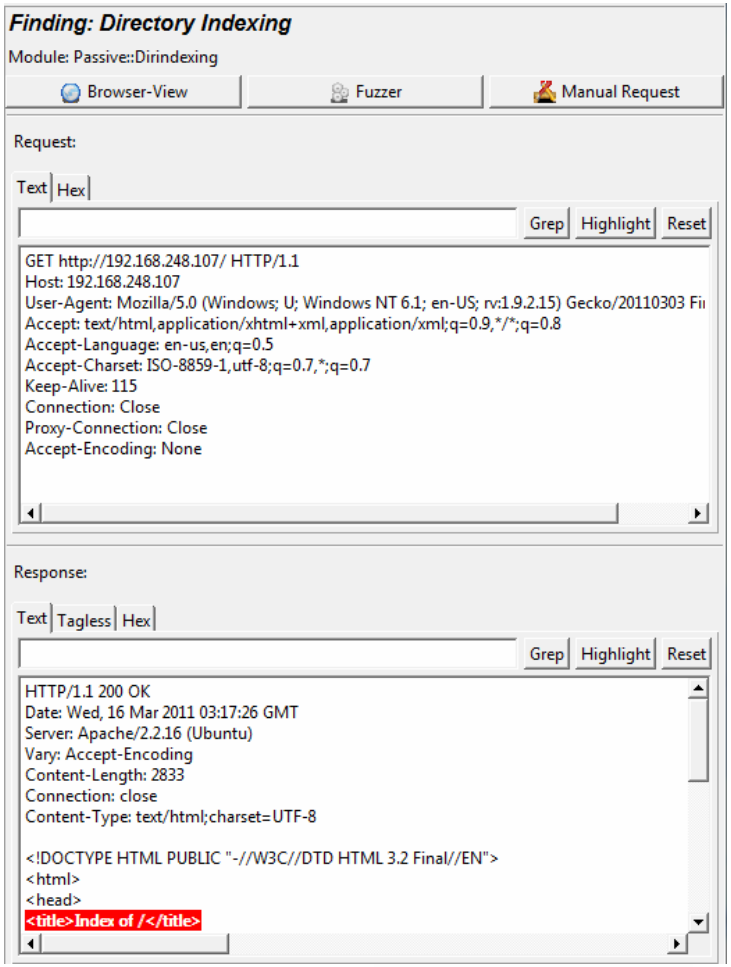
A5: Cross-Site Request Forgery (CSRF) – Tamper Data

- Attack:
  - Allows attacker to force victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim
- Defense:
  - Unique token in hidden field



A6: Security Misconfiguration – Watobo

- Attack:
  - Settings that are not defined, well implemented and maintained, or shipped with weak defaults are subject to exploit
- Defense:
  - Repeatable hardening process
  - Awareness and deployment of new updates and patches
  - Strong application architecture
  - Scan and audit



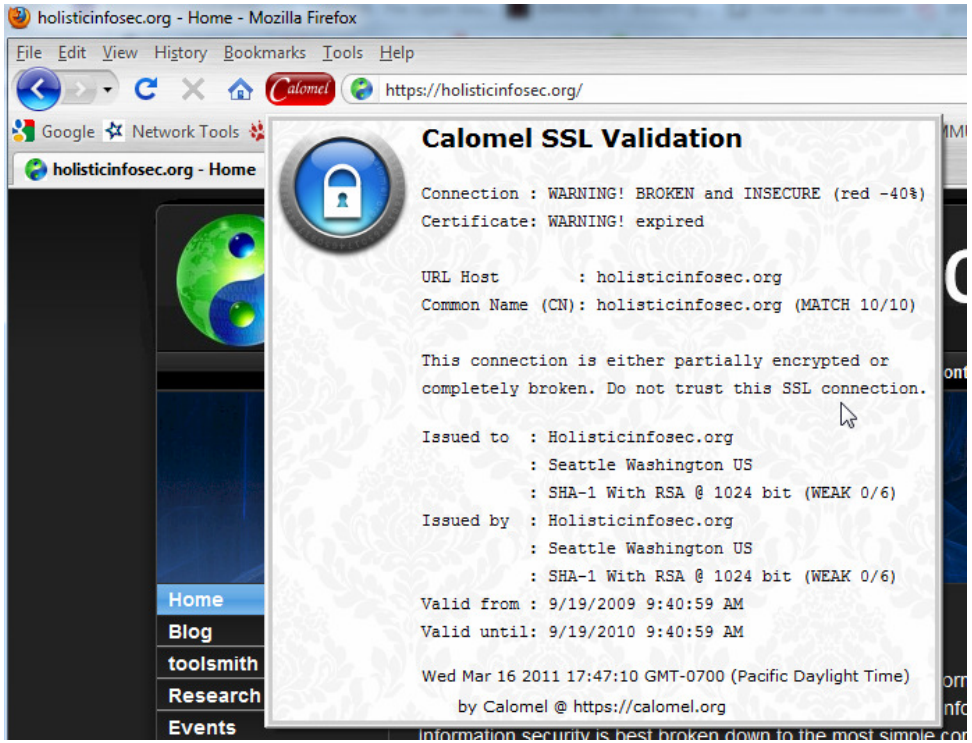
A8: Failure to Restrict URL Access - Nikto/Wikto

- Attack:
  - Applications must perform access control checks to protected pages each time accessed to prevent attackers from gaining unauthorized access
- Defense:
  - RBAC
  - Deny all by default
  - Check workflow

```
- Nikto v2.1.3
---------------------------------------------------------------------------
+ Target IP:          127.0.0.1
+ Target Hostname:    localhost
+ Target Port:        80
+ Start Time:         2011-03-16 23:13:30
---------------------------------------------------------------------------
+ Server: Apache/2.2.16 (Ubuntu)
+ OSVDB-3268: : Directory indexing found.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3268: /./: Directory indexing found.
+ OSVDB-3268: /?mod=node&nid=some_thing&op=view: Directory indexing found.
+ OSVDB-3268: /?mod=some_thing&op=browse: Directory indexing found.
+ /./: Appending '/./' to a directory allows indexing
+ OSVDB-3268: //: Directory indexing found.
+ OSVDB-3268: /?Open: Directory indexing found.
+ OSVDB-3268: /?OpenServer: Directory indexing found.
+ OSVDB-3268: /%2e/: Directory indexing found.
+ OSVDB-3268: /?mod=<script>alert(document.cookie)</script>&op=browse: Directory indexing found.
+ OSVDB-3268: /?sql_debug=1: Directory indexing found.
+ OSVDB-3268: ///: Directory indexing found.
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-3268: http://127.0.0.1:2301/ HTTP/1.0: Directory indexing found.
+ OSVDB-561: /server-status: This reveals Apache information. Comment out appropriate line in httpd.conf or restri
ct access to allowed hosts.
```

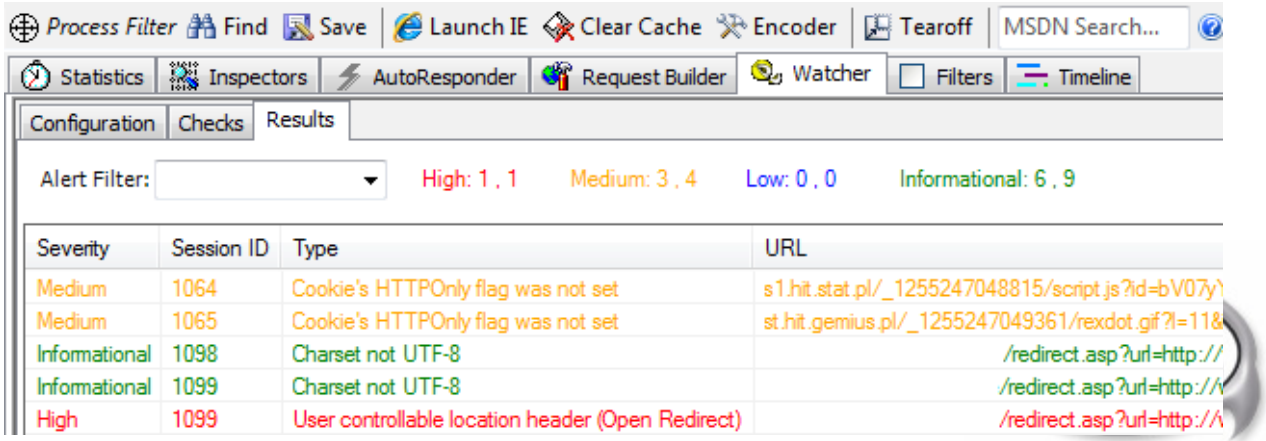
A9: Insufficient Transport Layer Protection - Calomel

- Attack:
  - Compromise confidentiality & integrity of sensitive network traffic due to lack of auth or encrypt, weak algorithms, expired or invalid certificates, or misuse
- Defense:
  - Require SSL (strong)
  - Secure flag on cookies
  - Valid certs



A10: Unvalidated Redirects and Forwards – Watcher

- Attack:
  - Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages
- Defense:
  - Avoid using redirects and forwards
  - Don't user parameters in calculating the destination
    - If using parameters, ensure valid values and user authorization



Summary

- Employ SDL/SDLC practices
  - Test regularly
- Conform to uniform standards
- Work with development teams to create delivery schedules that allow security-oriented checkpoints:
  - Static code analysis
  - Development/integration deployments for testing
- Review all available resources
- Training: GWAPT, GSSP Java & .NET, GWEB

Resources

- ZAP Web Application Vulnerability Example
  - http://code.google.com/p/zaproxy/downloads/detail?name=zap-wave-0.2.zip&can=2&q=
- Tamper Data (from Firefox Add-ons)
- Fiddler
  - http://www.fiddler2.com/fiddler2/version.asp
- Watcher
  - http://websecuritytool.codeplex.com/releases/view/22212
- Damn Vulnerable Web Application
  - http://www.dvwa.co.uk/
- NOWASP Mutillidae
  - http://sourceforge.net/projects/mutillidae/

Q & A

- russ at holisticinfosec dot org
- rmcree at microsoft dot com