CIS 700/002 : Special Topics : OWASP ZED (ZAP)

Hitali Sheth

CIS 700/002: Security of EMBS/CPS/IoT

Department of Computer and Information Science

School of Engineering and Applied Science

University of Pennsylvania

OWASP ZAP – zed attack proxy

- Security vulnerabilities in web applications while developing and testing applications
- Open source tool, GUI
- Helps in manual and automated testing
- Should be used with only own web applications or the applications you have permission to test

- Comparison with Burp : similar tool
  - BURP is a hard core tool, should have very good knowledge in security matters
  - ZAP has got some neat features, covers most of the bases and it is easier to use

Basic Functionalities

- Intercepting proxy server
- Web Crawling
- Active Attacks
- Fuzzer

Penetration testing

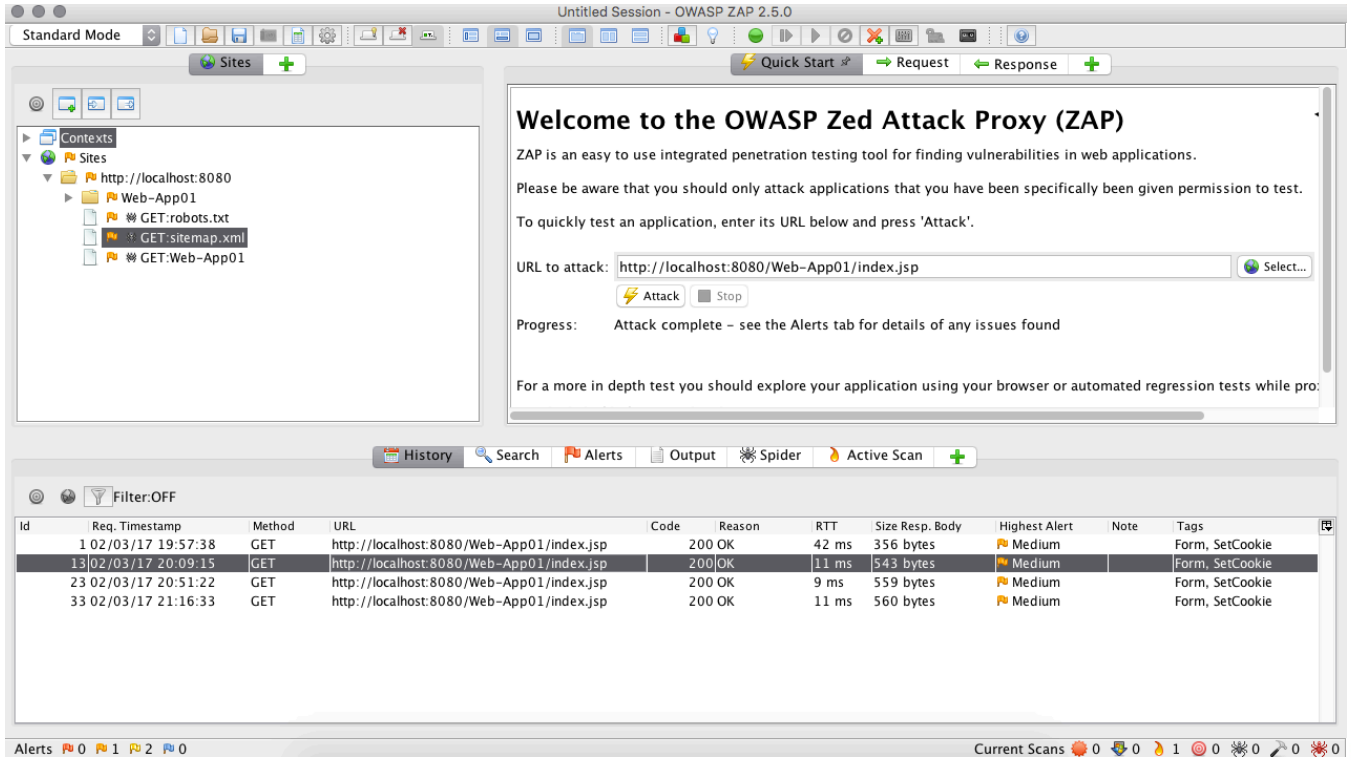
- When developing web applications, it is important that it is secure in every phase

- Attacks performed by embedding malicious strings :
  - Query strings
  - Form Fields
  - Cookies
  - HTTP Headers
    - command execution
    - cross-site scripting (XSS)
    - SQL injection
    - buffer overflow attacks.

User Interface

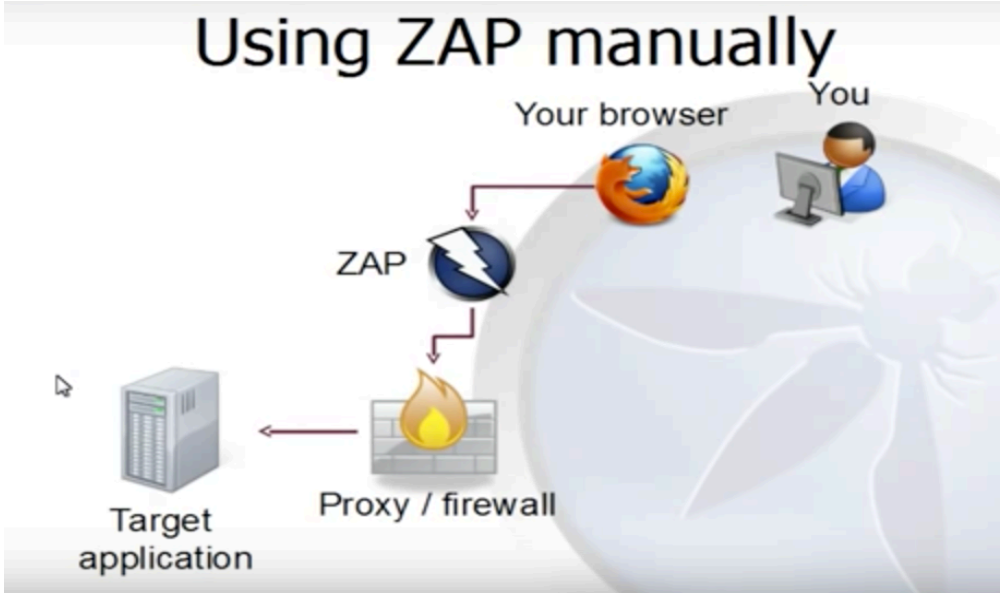
– Active attack, Request and Response headers

– Tools : Active scan, Local proxy, Spider



Intercepting Proxy Server

- Configure browser to allow Proxy (select hostname and port from application)



Active Attacks

- Web application running on Apache Tomcat : form accepts inputs and displays it.
- Attacks the application : All possible pages it gets directed to
- Displays alerts

Alert Information

– Common Weakness Enumeration (CWE)

– WASC ID – Threat Classification ID

Item Name	WASC ID
Insufficient Authentication	WASC-01
Insufficient Authorization	WASC-02
Integer Overflows	WASC-03
Insufficient Transport Layer Protection	WASC-04
Remote File Inclusion	WASC-05
Format String	WASC-06
Buffer Overflow	WASC-07
Cross-site Scripting	WASC-08
Cross-site Request Forgery	WASC-09
Denial of Service	WASC-10
Brute Force	WASC-11
Content Spoofing	WASC-12
Information Leakage	WASC-13
Server Misconfiguration	WASC-14
Application Misconfiguration	WASC-15
Directory Indexing	WASC-16
Improper Filesystem Permissions	WASC-17
Credential/Session Prediction	WASC-18
SQL Injection	WASC-19
Improper Input Handling	WASC-20

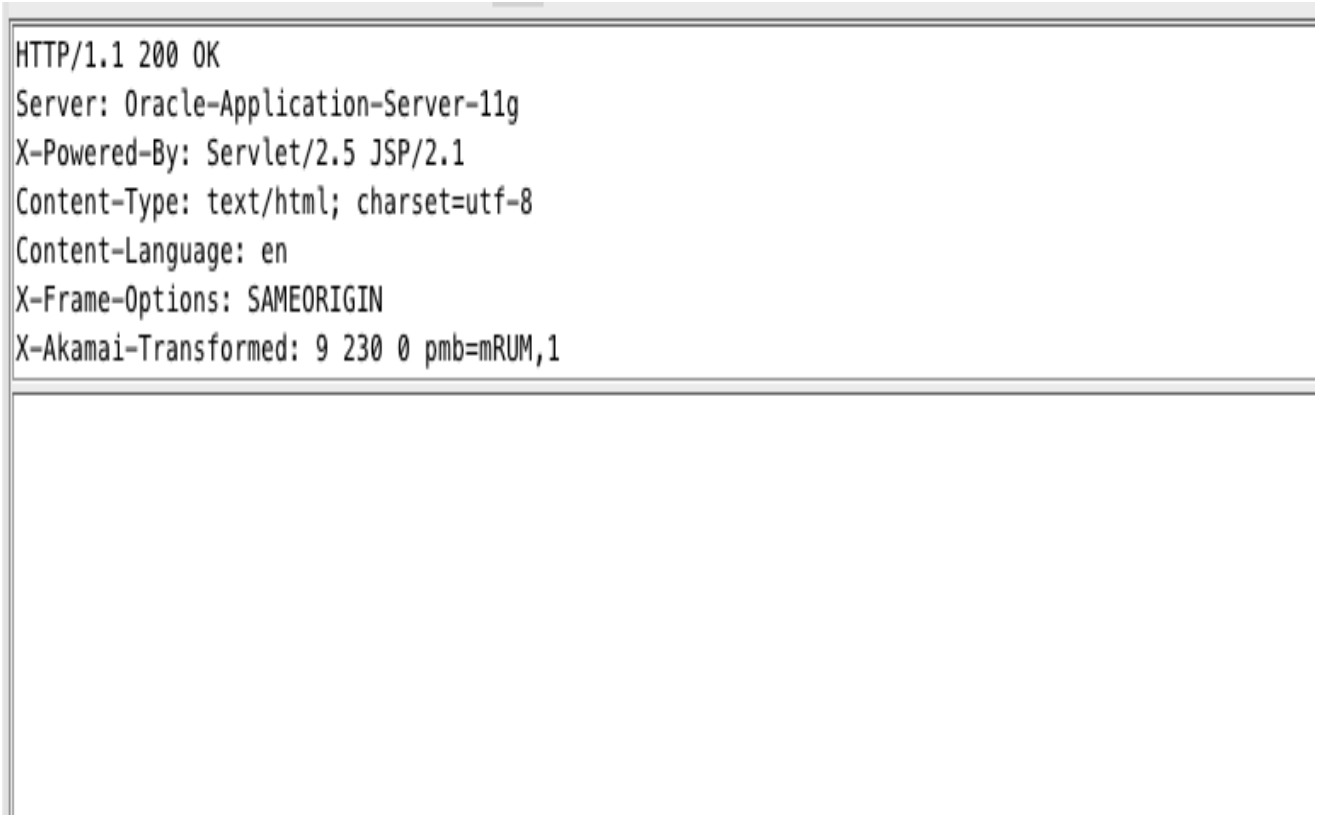
Some Alerts

- X-Frame Options Header Not Set

- Possible attack :
  - Clickjacking, also known as a "UI redress attack", is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the the top level page.

- Solution:
  - X-Frame-Options: SAMEORIGIN
  - X-Frame-Options: ALLOW-FROM https://example.com/

Some Alerts



```
HTTP/1.1 200 OK
Server: Oracle-Application-Server-11g
X-Powered-By: Servlet/2.5 JSP/2.1
Content-Type: text/html; charset=utf-8
Content-Language: en
X-Frame-Options: SAMEORIGIN
X-Akamai-Transformed: 9 230 0 pmb=mRUM,1
```

X-Frame-Options set here : SAME ORIGIN (in a response from Oracle application)

Some Alerts

- Web Browser XSS Protection is not enabled

- Possible attack :
  - Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted web sites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.

- Solution:
  - Set X-XSS-Protection HTTP response header to '1'.

Some Alerts

- X-Content-Type-Options Header Missing: Header set X-Content-Type-Options "nosniff”
- Possible Attack:
  - *MIME Sniffing* is a technique allowing the browser to dynamically guess the content type of downloaded files. If there is a mismatch between the content type of the server and the one defined by the magic bytes, then it uses its own content type guess.
- Solution:
  - X-Content-Type-Options header to 'nosniff' for all web pages.

Some Common active Attacks :

- **SQL injection** is a code injection technique, used to attack data-driven applications, in which nefarious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker)

- **Web Parameter Tampering** attack is based on the manipulation of parameters exchanged between client and server in order to modify application data, such as user credentials and permissions, price and quantity of products, etc.

- **Path Traversal** attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory.
  - "http://192.168.1.133/mutillidae/?page=%2Fetc%2Fpasswd" basically fetches you "/etc/passwd" on linux , the contents of the Linux password file

CRAWLING

- Given the seed URL, the application crawls to different pages in the application

- Can specify depth to crawl

FUZZ

Fuzz testing or *Fuzzing* is a Black Box software testing technique, which basically consists in finding implementation bugs using malformed/semi-malformed data injection in an automated fashion.

- Thank you!
- Questions ?