Sardar Patel College of Engineering, Bakrol

Cyber Security

Application Inspection Tools

Prepared by:-Keyur Patel

Contents

- Zed Attack proxy
- SQL map/ SQL injection
- DVWA
- Webgoat

The OWASP Zed Attack Proxy

About OWASP

- Full form: **Open Web Application Security Project**
- An open-source application security project
- Works to create freely-available
  - Articles
  - Methodologies
  - Documentation
  - Tools, and Technologies

Features of OWASP

- It provides free and open source
  - Application security tools and standards
  - Complete books on application security testing, secure code development, and security code review
  - Standard security controls and libraries
  - Local chapters worldwide
  - Cutting edge research
  - Extensive conferences worldwide
  - Mailing lists

History of OWASP

- OWASP was started on September 9, 2001
- It was started by Mark Curphey and Dennis Groves.
- Since late 2003, Jeff Williams served as the volunteer Chair of OWASP until September 2011.
- The current chair is Michael Coates, and vice chair is Eoin Keary.
- The OWASP Foundation was established in 2004 and supports the OWASP infrastructure and projects

OWASP-Zed Attack Proxy

- The Zed Attack Proxy (ZAP) is penetration testing tool for finding vulnerabilities in web applications.
- Designed to be used by people with a wide range of security experience
- Ideal for new developers and functional testers who are new to penetration testing
- Useful addition to an experienced pen testers toolbox
- Released September 2010
- Current Version -: 2.0.0

ZAP Principles

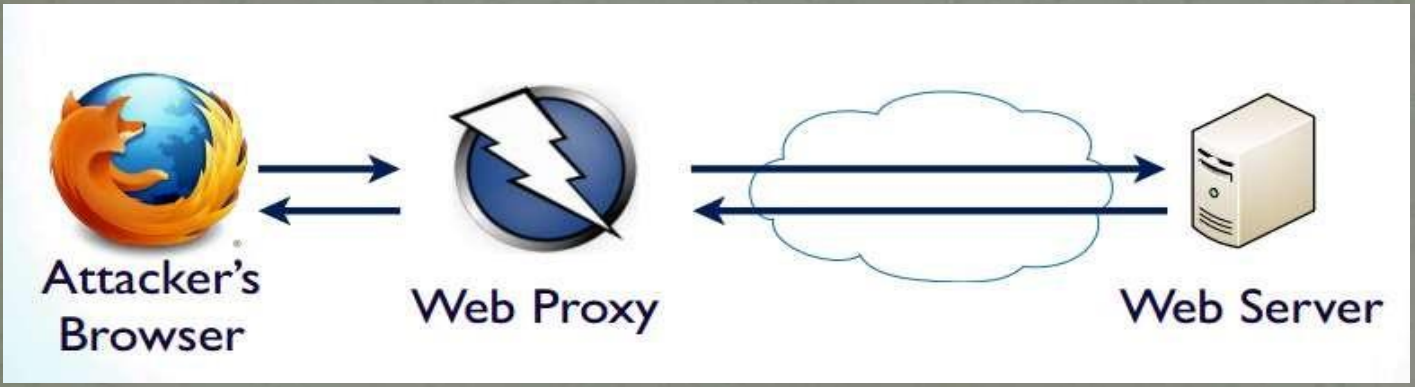
- Free, Open source
- Cross platform
- Easy to use
- Easy to install
- Internationalized
- Fully documented
- Involvement actively encouraged
- Reuse well regarded components

Features of ZAP

- Intercepting proxy
- Automated scanner
- Passive scanner
- Brute force scanner
- Spider
- Fuzzer
- Port scanner
- Dynamic SSL Certificates

Installation and Configuration of ZAP

- Download Link:
  - http://code.google.com/p/zaproxy/downloads/list
- Zap runs on proxy. To set up the proxy in ZAP
  - go to TOOLS > OPTIONS > LOCAL PROXY in ZAP
- Same configuration in the browser too



Functioning of ZAP

- Intercepting the traffic
- Traditional and AJAX spiders
- Automated scanners
- Analysing the scan results
- Reporting

Intercepting the traffic

- Configure the browser to use ZAP proxy server on local host
- Can intercept all traffic to a user specified website/server
- Can click on any link on the site to observe the captured request
- Can modify this request before forwarding it to the server
- The response can also be intercepted before forwarding it to the browser

Spidering

- ZAP spider is needed to crawl links that are not directly visible
- It automatically discovers and explores the hidden links for a site
- Newly discovered URLs are shown
- URLs whose domain is different from target are also listed

Scanning the website

- **Active Scanning**
  - Can select a site to be attacked under the ‘Attack’ section
  - Tool actually attacks the application in all possible ways to find out all possible vulnerabilities
  - Some of the issues active scan looks for are :
- Cross Site Scripting
- SQL Injection
- External Redirect
- Parameter tampering
- Directory browsing
- All findings shown under ‘Alerts’ tab

Scanning the website

- **Passive scanning**
  - Unlike active scanning, passive scanning does not change any responses coming from server
  - Only looks at responses to identify vulnerabilities
  - Safe to use
  - Some of the issues passive scanning looks for :
- Incomplete or no cache-control and pragma HTTP Header set
- Cross-domain JavaScript source file inclusion
- Cross Site Request Forgery
- Password Autocomplete in browser
- Weak authentication

Analysis and Reporting

- No tool's report is free from false positives
- Security analyst can determine which vulnerabilities are false positives
- It also shows the level of threat associated with the vulnerability mHigh, Medium, Low
- Analysed results are used to generate the report
- Can generate a detailed report of all vulnerabilities; can be exported to HTML file and viewed in a browser

Other ZAP features

- Port Scan
  - This feature scans open ports on the target site and lists them accordingly
- Encode/Decode Hash
  - This feature is used to encode/ decode the text entered
- Fuzzing
  - Fuzzing is the process of sending invalid and unexpected input to the application to observe the behaviour
- Extensions for ZAP
  - ZAP has plugins like LDAP Injection, session fixation etc. and many others that can be found on
  - http://code.google.com/p/zap-extensions/

ZAP- Firefox of Web Security

- ZAP is a free, open-source community developed tool aimed at making the online world more secure
- Some of the ideals that have driven ZAP are listed below
    - Help users develop and apply application security skills
    - Build a competitive, open source, and community oriented platform
    - Provide an extensible platform for testing
    - Designed to be easy to use
    - Raise the bar for other security tools

OWASP Top 10 Application Security Risks

Top 10 Application Security Risks

- A1 – Injection
- A2 – Cross-Site Scripting (XSS)
- A3 – Broken Authentication and Session Management
- A4 – Insecure Direct Object References
- A5 – Cross-Site Request Forgery (CSRF)
- A6 – Security Misconfiguration
- A7 – Insecure Cryptographic Storage
- A8 – Failure to Restrict URL Access
- A9 – Insufficient Transport Layer Protection
- A10 – Invalidated Redirects and Forwards

SQl Injection

- SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to affect the execution of predefined SQL statements.
- It is a common threat in web applications that lack of proper sanitization on user-supplied input used in SQL queries.

SQL Injection

- Problem: Incorrectly validated or non- validated string literals are concatenated into a
- dynamic SQL statement, and interpreted as code by the SQL engine.

Impact: Arbitrary SQL Execution, Data Corruption, Data Theft

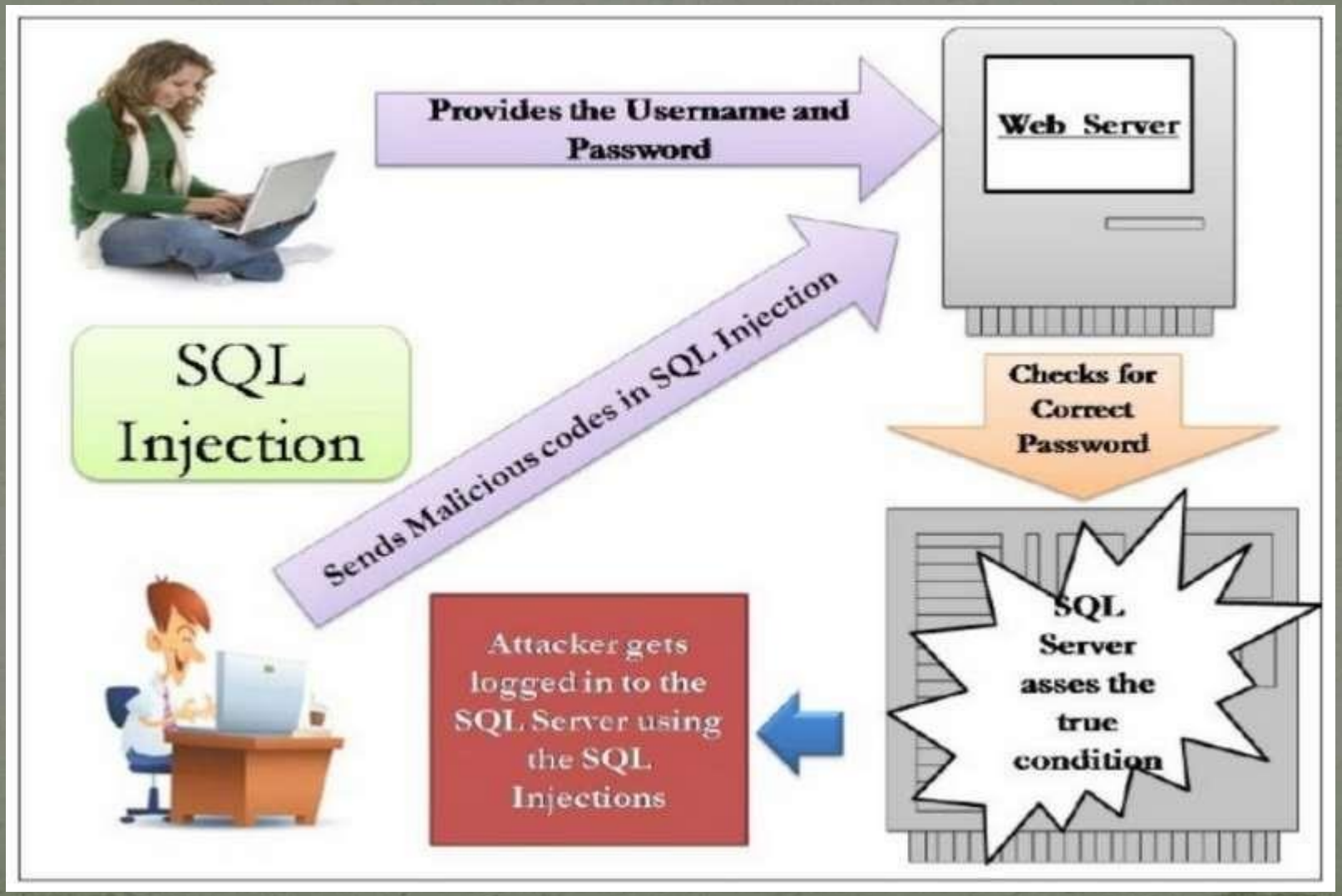
Basic SQL Injection Tests:

OR 1=1--' OR '1'='1'--

Example Vulnerable Query:

```
sqlQ = "Select user from UserTable where
name= '+username+ ' and pass='+password+ '
"
```

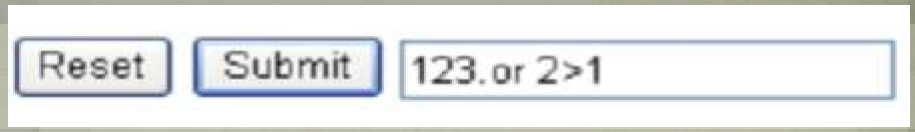
How to hack a website using Sql injection ?



The Vulnerable is execution of inputs without scan it.

Inputs like *username* maybe a sql statement!

Which executed at Database of server by Hackers.



1)Normal password : **karcobia**

$sql = "select * from users where pass=$password";

2) Attacker's password : **abc. or 1=1**

$sql = "select * from users where pass=$password".or 1=1;

```
$c =new con;

$c->con();

$password=$HTTP_POST_VARS['password'];

$sql="SELECT  * FROM cus where pass=$password";

$q=mysql_query($sql);
```

As we can see here we got all users and passwords in the Database!

Result

Hacker can execute any sql statement like Admin privileges!

Types of SQL Injections

- First Order Attack

  The attacker can simply enter a malicious string and cause the modified code to be executed immediately.

- Second Order Attack

  The attacker injects into persistent storage (such as a table row) which is deemed as a trusted source. An attack is subsequently executed by another activity.

- Lateral Injection.

  The attacker can manipulate the implicit functionTo_Char() by changing the values of the environment variables

Prevention of SQL Injection

Reduce the attack surface.

- Ensure that all excess database privileges are revoked
- Avoid dynamic SQL with concatenated input
- Use bind arguments.
- Filter and sanitize input.
  - The Oracle-supplied DBMS_ASSERT package contains a number of functions that can be used to sanitize user input

DVWA

- Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

Webgoat

- OWASP WebGoat Project provides:

Web application security is difficult to learn and practice. Not many people have full blown web applications like online book stores or online banks that can be used to scan for vulnerabilities. In addition, security professionals frequently need to test tools against a platform known to be vulnerable to ensure that they perform as advertised. All of this needs to happen in a safe and legal environment. Even if your intentions are good, we believe you should never attempt to find vulnerabilities without permission. The primary goal of the WebGoat project is simple: create a de-facto interactive teaching environment for web application security.

Conclusions

- Keep server and third-party applications and library up-to-date.
- Do not trust user input.
- Review code & design and identify possible weaknesses.
- Monitor run-time activity to detect ongoing attacks/probes.

Thank you !!!