

Úvod do počítačovej bezpečnosti

Zadanie 9 – Zraniteľnosti web aplikácií

Cieľom zadanie bolo oboznámiť sa podrobnejšie s problematikou webových zraniteľností.

Na hľadanie webových zraniteľností bol použitý program OWASP ZAP (Zed Attack Proxy) <https://www.zaproxy.org/> a jeho funkcionality ako Passive Scannig, Spider, Active Scanning a Fuzzing

A1. Injection

SQL Injection je typ útoku, ktorý sa snaží vykonať SQL príkazy cez používateľské vstupy. Cieľom tohto útoku môže byť získanie, zmena alebo zmazanie dát v databáze. Je to jeden z najčastejších typov útoku na webové aplikácie.

SQL Injection	
URL:	http://192.168.56.101.nip.io/udpb/www-vulnerable/?name=cFyWHkyY%27+AND+%271%27%3D%271%27+--+&pass=&login=1
Riziko:	High
Confidence:	Medium
Parameter:	name
Útok:	cFyWHkyY' OR '1'='1' --
Evidence:	
CWE ID:	89
WASC ID:	19
Zdroj:	Aktívny (40018 - SQL Injection)

Obr. 1: SQL Injection

Zraniteľnosť bola odstránená použitím **Prepared Statements**.

```
function verify_login() {
    global $db;
    if($sql = $db->prepare("SELECT id,name,password FROM admins WHERE name=? AND password=? LIMIT 1")){

        $userName = $_POST['name'];
        $hashPassword = hash("sha512",$_POST['pass']);

        $sql->bind_param("ss", $userName, $hashPassword);
        $sql->execute();
        $sql->bind_result($userId, $name, $pass);
        $sql->fetch();
        $sql->close();
    }

    $db->close();

    if(!empty($userId)){
        $_SESSION['id'] = $userId;
        $_SESSION['name'] = $name;
        $_SESSION['session_id'] = session_id();
        return true;
    } else{
        return false;
    }
}
```

Obr. 2: Prihlasovanie do aplikácie (login.php)

```
if($sql = $db->prepare("SELECT * FROM articles WHERE title LIKE ? OR content LIKE ? ")){
    $search = "%".$_POST[search]."%";
    $sql->bind_param("ss",$search, $search);
    $sql->execute();
    $result = $sql->get_result();
    $article = $result->fetch_assoc();
}

?>
```

Obr. 3: Vyhľadávanie v aplikácii (search.php)

A2. Broken Authentication

Táto zraniteľnosť umožňuje útok na prihlasovacie časti aplikácie, ktoré sú často implementované nesprávne čo útočníkom umožňuje získať heslá, kľúče, tokeny relácii alebo využiť ďalšie chyby v implementácii na dočasné alebo trvalé získanie identity používateľa.

Information Disclosure - Sensitive Information in URL	
URL:	http://192.168.56.101.nip.io/udpb/www-vulnerable/?page=logout.php&session_id=5tk8tsccght7gvt9jgh6nj9336&go_page=index.php
Riziko:	Informational
Confidence:	Medium
Parameter:	session_id
Útok:	
Evidence:	session_id
CWE ID:	200
WASC ID:	13
Zdroj:	Pasívny (10024 - Information Disclosure - Sensitive Information in URL)
Description:	

Obr. 4: Broken Authentication and Session Management

V linke, ktorá je vyvolaná po stlačení odhlasovacieho tlačidla je zverejnené session_id, ktoré by malo byť uchované v tajnosti.

```
<?php
if(isLogin()){
    echo '<li><a href="./?page=logout.php&session_id='.session_id().'&go_page=index.php">Odhlásiť sa</a></li>';
}else{
    echo '<li><a href="./?page=login.php">Login</a></li>';
}
?>
```

Obr. 5: Nesprávna implementácia s verejným sessionId (index.php)

Zraniteľnosť bola odstránená vymazaním sessionId z URL (sessionId sa prenáša cez cookies, takže nie je potrebné aby bolo aj v URL).

```
<?php
if(isLogin()){
    echo '<li><a href="./?page=logout.php&go_page=index.php">Odhlásiť sa</a></li>';
}else{
    echo '<li><a href="./?page=login.php">Login</a></li>';
}
?>
```

Obr. 6: Správna implementácia bez verejného sessionId (index.php)

A3. XSS (Cross Site Scripting)

Cross-Site Scripting je typ útoku, ktorý sa snaží vykonať JS kód pomocou neošetrených používateľských vstupov. Tento častokrát škodlivý kód je vložený ako hodnota do vstupného poľa a pokiaľ nie je ošetrované vkladanie aj následne uložený do premennej prislúchajúcej danému vstupu. Zavolaním premennej, v ktorej sa škodlivý skript nachádza dôjde k jeho vykonaniu. Cieľom tohto útoku môže byť získanie citlivých údajov o návštevníkoch stránky alebo realizácia phishingu.

Cross Site Scripting (Reflected)	
URL:	http://192.168.56.101.nip.io/udpb/www-vulnerable/index.php?page=search.php
Riziko:	High
Confidence:	Medium
Parameter:	search
Útok:	</h1><script>alert(1);</script><h1>
Evidence:	</h1><script>alert(1);</script><h1>
CWE ID:	79
WASC ID:	8
Zdroj:	Aktívny (40012 - Cross Site Scripting (Reflected))

Obr. 7: Cross Site Scripting

Ak zadáme priamo do Search formulára JS kód, po odoslaní formuláru dôjde k jeho vykonaniu.

```
<?php
$search = $db->query('SELECT * FROM articles WHERE title LIKE "%'.$_POST[search].'" OR content LIKE "%'.$_POST[search].'"');
?>

<!--Co tak dat vysledky vyhľadavania a data[title] do htmlspecialchars? -->
<h1> Výsledky vyhľadavania: <?=$_POST['search']?></h1>

<div>
<?php
try {
    while($data = $search->fetch_array(MYSQL_ASSOC)){
        echo 'Article: <a href=/index.php?id='.$data["id"].'>'.$data["title"].'</a><br />';
    }
} catch (Exception $e) {
    header("LOCATION: error_page.php");
}
?>
</div>
```

Obr. 8: XSS - zraniteľný kód (search.php)

Zraniteľnosť bola odstránená použitím funkcie htmlspecialchars(), ktorá zaručuje, že špeciálne HTML znaky budú prepísané na ascii kód. Čo znamená, že ak používateľ zadá `<script>alter(1);</script>` do vyhľadávania, takýto string sa v HTML kóde zobrazí ako `<script>alter(1);</script>` (teda nedôjde k vykonaniu js funkcie) a v prehliadači sa opäť zobrazí ako `<script>alter(1);</script>`

```
<?php
if($sql = $db->prepare("SELECT * FROM articles WHERE title LIKE ? OR content LIKE ? ")){
    $search = "%".$_POST[search]."%";
    $sql->bind_param("ss",$search, $search);
    $sql->execute();
    $result = $sql->get_result();
    $article = $result->fetch_assoc();
}
?>

<h1> Výsledky vyhľadavania: <?=htmlspecialchars($_POST['search'])?></h1>

<div>
<?php
try {
    while($data = $result->fetch_assoc()){
        echo 'Article: <a href=index.php?id='.$data["id"].'.htmlspecialchars($data["id"]).'>'.htmlspecialchars($data["title"]).'</a><br />';
    }
} catch (Exception $e) {
    header("LOCATION: error_page.php");
}
?>
</div>
```

Obr. 9: XSS - Bezpečná implementácia spracovania vstupov od používateľa (search.php)

A4. Insecure Directory Object References

Tento typ útoku umožňuje útočníkovi prístup k súborom, adresárom a príkazom, ktoré sa potenciálne nachádzajú mimo koreňového adresára web aplikácie. Útočník môže manipulovať s adresou URL takým spôsobom, že webová stránka vykoná alebo odhalí obsah ľubovoľných súborov kdekoľvek na webovom serveri.

Path Traversal	
URL:	http://192.168.56.101.nip.io/udpb/www-vulnerable/index.php?page=../../../../../../../../etc/passwd
Riziko:	High
Confidence:	Medium
Parameter:	page
Útok:	../../../../../../../../etc/passwd
Evidence:	root:x:0:0
CWE ID:	22
WASC ID:	33
Zdroj:	Aktívny (6 - Path Traversal)

Obr. 10: Path traversal

```
<?php
@$pages=$_GET["page"];
if(!isLogin()) { $pages='login.php'; }
if (!isset($pages) || empty($pages)){
    require("content/home.php");
}elseif (file_exists("content/$pages")) {
    require("content/$pages");
}else{require ("content/error_page.php");}
?>
```

Obr. 11: Zraniteľný kód umožňujúci prístup k súborom mimo aplikácie (index.php)

Zraniteľnosť bola odstránená pridaním **Whitelistu** stránok, ktoré je povolené includovať.

```
<?php
@$pages=$_GET["page"];
if(!isLogin()) {
    $pages='login.php';
}

$allowedPages = array('search.php','logout.php', 'login.php', 'kontakt.php', 'home.php');

if (!isset($pages) || empty($pages)){
    require("content/home.php");
}elseif (in_array($pages, $allowedPages) && file_exists("content/$pages")) {
    require("content/$pages");
} else{
    require ("content/error_page.php");
}
?>
```

Obr. 12: Bezpečná implementácia umožňujúci vkladanie len špecifikovaných súborov (index.php)

A5. Security Misconfiguration




Najčastejšie sa vyskytujúci problém spočíva v zlom nastavení konfigurácie zabezpečenia. Je to obvykle výsledkom nezabezpečených predvolených konfigurácií, neúplných konfigurácií, otvoreného cloudového úložiska, nesprávne nakonfigurovaných hlavičiek a podrobných chybových správ obsahujúcich citlivé informácie. Je potrebné a nutné aby všetky operačné systémy, frameworky, knižnice a aplikácie boli bezpečne nakonfigurované a včas opravené alebo aktualizované.

Directory Browsing	
URL:	http://192.168.56.101.nip.io/udpb/
Riziko:	Medium
Confidence:	Medium
Parameter:	
Útok:	Parent Directory
Evidence:	
CWE ID:	548
WASC ID:	48
Zdroj:	Aktívny (0 - Directory Browsing)
Description:	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files, backup source files, etc. which can be accessed to read sensitive information.

Obr. 13: Directory browsing

Na servery je zapnutý dir listing čo umožňuje prehľadávanie adresárov aj mimo stránky.

Zraniteľnosť bola odstránená pridaním súbor .htaccess s obsahom Options -Indexes (zakázanie indexovania) do root adresára našej webstránky. A preto aby .htaccess súbory neboli ignorované je tiež potrebné nastaviť AllowOverride z None na All

/var/www/		
Názov	Veľkosť	Zmenené
 .		30.11.2015 20:20:36
 udpb		5.12.2020 16:05:06
 .htaccess	1 KB	3.12.2020 23:30:56

Obr. 14: Štruktúra adresáru

```
<Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    allow from all
</Directory>
```


Obr. 15: Nastavenie akceptovania .htaccess súborov

Forbidden

You don't have permission to access /udpb/ on this server.

Obr. 16: Ukážka obrazovky pri pokuse zobrazíť neprístupný adresár

Aplikácia v prípade chyby vypisuje na obrazovku informácie, ktoré by mohli byť použité na prípadný útok.

Application Error Disclosure
URL: http://192.168.56.101.nip.io/udpb/
Riziko:  Medium
Confidence: Medium
Parameter:
Útok:
Evidence: Parent Directory
CWE ID: 200
WASC ID: 13
Zdroj: Pasívny (90022 - Application Error Disclosure)
Description:
This page contains an error/warning message that may disclose sensitive information I

Obr. 17: Error disclosure

Zraniteľnosť bola odstránená vypnutím vypisovania chybových hlášok na obrazovku.
ini_set('display_errors', 0); (pôvodne 1)

```
//Zobrazovanie chyb okrem notice
error_reporting(E_ALL & ~E_NOTICE);
ini_set('display_errors', 0);
```

Obr. 18: Vypnutie výpisu error hlášok (db.php)

A6. Sensitive Data Exposure

Mnoho webových aplikácií a rozhraní API nechráni správne citlivé údaje, ako sú finančné údaje a informácie umožňujúce identifikáciu osôb. Útočníci môžu takto slabo chránené údaje ukradnúť alebo upraviť s cieľom spáchať podvod. Citlivé údaje môžu byť ohrozené bez osobitnej ochrany, napríklad šifrovania.

Information Disclosure - Sensitive Information in URL	
URL:	http://192.168.56.101.nip.io/udpb/www-vulnerable/?name=student&pass=student&logiN=1
Riziko:	Informational
Confidence:	Medium
Parameter:	pass
Útok:	
Evidence:	pass
CWE ID:	200
WASC ID:	13
Zdroj:	Pasívny (10024 - Information Disclosure - Sensitive Information in URL)

Obr. 19: Odosielanie prihlasovacích údajov nešifrovane cez GET request

Prihlasovacie údaje sa prenášajú cez nezabezpečený HTTP protokol cez GET request priamo v URL.

Zraniteľnosť bola odstránená v prvom rade vytvorením bezpečnej šifrovanej komunikácie cez HTTPS protokol podľa nasledujúceho návodu:

<https://www.digitalocean.com/community/tutorials/how-to-create-a-ssl-certificate-on-apache-for-debian-7> + automatické presmerovanie na HTTPS

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    Redirect permanent / https://192.168.56.101.nip.io/
```

Obr. 20: Presmerovanie na HTTPS

Ako ďalšie bolo potrebné zmeniť pôvodnú GET metódu formuláru na POST aby dáta neboli prenášané cez URL.

```
<form method="post" name="login">
  <label>Meno</label>
  <input name="name" value="" type="text" placeholder="LamaCoder" autofocus />
  <label>Heslo</label>
  <input name="pass" value="" type="password" placeholder="*****" />
  <br />
  <button class="button" name="logIN" value="1">Prihlasiť</button>
</form>
```

Obr. 21: POST metóda (login.php)

A7. Missing Function Level Access Control

Útoky sa sústreďia na zneužitie prístupových práv do systému (napr. prihlasovací token neprivilegovaného používateľa) na prístup do chránenej časti, ktorá by mala byť prístupná iba prihlásenému užívateľovi. Backend aplikácie by si mal pri každej požiadavke overiť, či má k danej funkcionalite používateľ prístup a zabezpečiť prihlasovacie tokeny proti ich zneužitiu.

Directory Structure	Response Code	Response Size
www-vulnerable	200	905
www-vulnerable	200	2803
www-vulnerable	???	???
udpb	???	???
js	200	1778
content	200	2180
error_page.php	200	203
home.php	200	7782
kontakt.php	200	634
login.php	500	228
logout.php	302	205
search.php	500	228
css	200	3010
index.php	200	2747
js	200	1778

Current speed: 3575 requests/sec (Select and right click for more options)

Obr. 22: Štruktúra webstránky

Aplikácia umožňuje neautentifikovaný prístup k stránkam, ku ktorým by mal byť povolený prístup iba po autentifikácii.

Zraniteľnosť bola odstránená pridaním súbor .htaccess s obsahom `Deny from all` (zabráňuje aby bolo možné pristupovať k obsahu adresáru priamo cez URL, backend kód ma k obsahu aj naďalej prístup) do content adresára našej webstránky.

Forbidden

You don't have permission to access /udpb/www-vulnerable/content/home.php on this server.

Obr. 23: Ukážka obrazovky pri pokuse zobrazit' neprístupný súbor

A8. Cross-Site Request Forgery (CSRF)

Technika umožňujúca útočníkovi podvrhnúť formulár na inej stránke alebo pomocou HTTP metódy presmerovať prehliadač obete na script spracujúci legitímny formulár dátovej aplikácie, ktorá poškodzuje obeť.

Absence of Anti-CSRF Tokens

URL: <http://192.168.56.101.nip.io/udpb/www-vulnerable/?page=kontakt.php>

Riziko: 🔒 Low

Confidence: Medium

Parameter:

Útok:

Evidence: `<form action="" method="POST" enctype="multipart/form-data">`

CWE ID: 352

WASC ID: 9

Zdroj: Pasívny (10202 - Absence of Anti-CSRF Tokens)

Description:

No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim.

Obr. 24: Cross-Site request Forgery

Na kontaktnom a prihlasovacom formulári nie sú implementované CSRF tokeny.

Zraniteľnosť bola odstránená pridaním CSRF tokenom na login a kontakt formulár podľa nasledujúceho návodu: [https://www.wikihow.com/Prevent-Cross-Site-Request-Forgery-\(CSRF\)-Attacks-in-PHP](https://www.wikihow.com/Prevent-Cross-Site-Request-Forgery-(CSRF)-Attacks-in-PHP)

```
$csrf = new csrf();

// vygenerovanie token_id a token_value
$token_id = $csrf->get_token_id();
$token_value = $csrf->get_token($token_id);

// vygenerovanie nahodnych nazvov pre polia formularu
$form_names = $csrf->form_names(array('user', 'password'), false);

if(isset($_POST[$form_names['user']], $_POST[$form_names['password']])) {
    // kontrola ci su token_id a token_value platne
    if($csrf->check_valid('post')) {
        // ziskanie premennych formularu
        $user = $_POST[$form_names['user']];
        $password = $_POST[$form_names['password']];

        if(@$_POST['logIN']) {
            if(verify_login($user, $password)) {
                header('LOCATION: index.php');
            } else {
                $error = "Wrong name or password!! Pls try it again!!";
            }
        }
    }
    // vygenerovanie novych nahodnych hodnot pre polia formularu
    $form_names = $csrf->form_names(array('user', 'password'), true);
}
```

Obr. 25: Implementácia aplikácie CSRF tokenov na prihlasovacom formulári (login.php)

```
<?if(!isLogin()){?>
<div style="width:20%;">
    <?=@$error?>
    <form method="post" name="login">
        <input type="hidden" name="<?=$token_id; ?>" value="<?=$token_value; ?>" />
        <label>Meno</label>
        <input name="<?=$form_names['user'];?>" value="" type="text" placeholder="LamaCoder" autofocus />
        <label>Heslo</label>
        <input name="<?=$form_names['password'];?>" value="" type="password" placeholder="*****" />
        <br />
        <button class="button" name="logIN" value="1">Prihlasiť</button>
    </form>
</div>
<?}else{?>
    <div style="width:20%;">
        <?=@$error?>
        <a href="./?page=logout.php"><button class="button">Odhlásiť sa</button></a>
    </div>
<?}?>
```

Obr. 26: Prihlasovací formulár s CSRF tokenmi (login.php)

A9. Using Component with Known Vulnerabilities

Komponenty ako sú knižnice, frameworky a ďalšie softvérové moduly, majú rovnaké oprávnenia ako samotná aplikácia. Ak dôjde k zneužitiu zraniteľnej súčasti systému, takýto útok môže spôsobiť vážnu stratu údajov alebo prevzatie servera. Aplikácie a API využívajúce komponenty so známymi slabými miestami môžu narušiť obranu aplikácií a umožniť rôzne útoky.

Zo scanu servera vieme, že verzia serveru nie je aktuálna. **Apache/2.2.22 appears to be outdated**

Na stránke httpd.apache.org sa môžeme dočítať, že táto verzia je už zastaralá a od Decembra 2017 by sa nemala viac používať.

Viac informácií a prehľad zraniteľností tejto verzie Apache si môžete prečítať na tejto stránke: [httpd 2.2 vulnerabilities - The Apache HTTP Server Project](http://httpd.apache.org/2.2/vulnerabilities.html)

A10.Unvalidated Redirect and Forwards

Webové aplikácie často presmerujú užívateľov na iné stránky, útočník môže použiť nedôveryhodné údaje na určenie cieľovej stránky. Bez správneho overenia môže útočník presmerovať obeť na phishing alebo malware stránky.

External Redirect	
URL:	http://192.168.56.101.nip.io/udpb/www-vulnerable/?page=logout.php&session_id=6t8t0r1o3koq7rj53lf8m4p862&go_page=http%3A%2F%2F7290115502241894197.owasp.org
Riziko:	High
Confidence:	Medium
Parameter:	go_page
Útok:	http://7290115502241894197.owasp.org
Evidence:	http://7290115502241894197.owasp.org
CWE ID:	601
WASC ID:	38
Zdroj:	Aktívny (20019 - External Redirect)

Obr. 27: External redirect

Web aplikácia dynamicky načítava obsah kam by napríklad v prípade odhlásenia mal byť užívateľ presmerovaný. Tento obsah je načítaný na základe vstupu z URL. Vstup je nedostatočne ošetrovaný a útočník naň môže vložiť modifikovanú URL, ktorá presmeruje užívateľa na potenciálne škodlivú stránku.

```
<?php
//zrusime session
$_SESSION = array();
session_destroy();
header("LOCATION: ".$_GET['go_page']);
?>
```

Obr. 28: Nebezpečný spôsob presmerovania (logout.php)

Zraniteľnosť bola odstránená nastavením statickej adresy kam má byť používateľ presmerovaný po odhlásení.

```
<?php
//zrusime session
$_SESSION = array();
session_destroy();
header("LOCATION: index.php");
?>
```

Obr. 29: Statické presmerovanie (logout.php)