**Zadanie č.5**

Úvod do počítačovej bezpečnosti

Zraniteľnosť web aplikácií

2016/2017

**Autor: Michal Drahovský, Ján Dzvoník**

Obsah

[1 Postup riešnia 4](#_Toc469592989)

[1.1 SQL injection 4](#_Toc469592990)

[1.1.1 Analýza zraniteľnosti 4](#_Toc469592991)

[1.1.2 Odstránenie zraniteľnosti 4](#_Toc469592992)

[1.2 Broken Authentication and Session Management 5](#_Toc469592993)

[1.2.1 Analýza zraniteľnosti 5](#_Toc469592994)

[1.2.2 Odstránenie zraniteľnosti 5](#_Toc469592995)

[1.3 Cross-site scripting 6](#_Toc469592996)

[1.3.1 Analýza zraniteľnosti 6](#_Toc469592997)

[1.3.2 Odstránenie zraniteľnosti 6](#_Toc469592998)

[1.4 Insecure Direct Object References 6](#_Toc469592999)

[1.4.1 Analýza zraniteľnosti 6](#_Toc469593000)

[1.4.2 Odstránenie zraniteľnosti 6](#_Toc469593001)

[1.5 Security Misconfiguration 7](#_Toc469593002)

[1.5.1 Analýza zraniteľnosti 7](#_Toc469593003)

[1.5.2 Odstránenie zraniteľnosti 7](#_Toc469593004)

[1.6 Sensitive Data Exposure 7](#_Toc469593005)

[1.6.1 Analýza zraniteľnosti 7](#_Toc469593006)

[1.6.2 Odstránenie zraniteľnosti 7](#_Toc469593007)

[1.7 A7 - Missing Function Level Access Control 9](#_Toc469593008)

[1.7.1 Analýza zraniteľnosti 9](#_Toc469593009)

[1.7.2 Odstránenie zraniteľnosti 9](#_Toc469593010)

[1.8 A8 - Cross-Site Request Forgery (CSRF): 10](#_Toc469593011)

[1.8.1 Analýza zraniteľnosti 10](#_Toc469593012)

[1.8.2 Odstránenie zraniteľnosti 10](#_Toc469593013)

[1.9 A9 - Using Components with Known Vulnerabilities 11](#_Toc469593014)

[1.9.1 Analýza zraniteľnosti 11](#_Toc469593015)

[1.9.2 Odstránenie zraniteľnosti 11](#_Toc469593016)

[1.10 A10 - Unvalidated Redirects and Forwards 12](#_Toc469593017)

[1.10.1 Analýza zraniteľnosti 12](#_Toc469593018)

[1.10.2 Odstránenie zraniteľnosti 12](#_Toc469593019)

[2 Záver 12](#_Toc469593020)

# Postup riešnia

## SQL injection

### Analýza zraniteľnosti

SQL injection je bezpečnostná chyba založená na možnosti manipulovať s dátami v databáze bez nutnosti vlastníctva legitímnych prístupových údajov. Zneužitie môže viesť k získaniu citlivých údajov, ako napr. prihlasovacie údaje, osobné údaje (čísla bankových účtov, rodné čísla...) a v niektorých prípadoch môže viesť aj k vykonaniu systémového príkazu, prípadne k ovládnutiu celého serveru. Princípom je vkladanie nových SQL príkazov do už existujúcich príkazov.

V Search forme a tak isto aj v prihlasovacom forme je prítomna zraniteľnosť typu SQLi. Ak zadáte nasledujúci payload do Search formulára dostanete sa k hashu jedného z používateľov - %" UNION SELECT ALL 1,concat(password,char(58)),3,4,5,6 from admins where name='student'-- .

### Odstránenie zraniteľnosti

Na odstránenie SQLi bolo treba zmeniť zdrojový kód súboru "search.php" a takisto "login.php". Kód funkcie verify\_login a kód súboru "search.php", ktorý pristupuje k databáze sme nahradili pomocou prepared statement a funkcie real\_escape\_string. Takto ošetrené vstupy sú zabezpečené proti SQLi.

Ukážka zdrojového kódu funkcie verify\_login súboru "login.php"



Ukážka zdrojového kódu súboru "search.php"



## Broken Authentication and Session Management

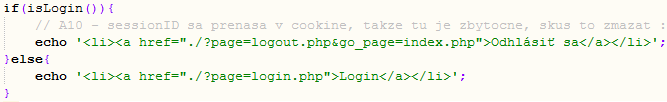
### Analýza zraniteľnosti

Táto zraniteľnosť umožňuje útok na prihlasovacie časti aplikácie. Je nutné zamerať sa na predávanie autentifikačných údajov a bezpečné úložisko identifikátora relácie. V linke, ktorá je vyvolaná po stlačení odhlasovacieho tlačidla je zverejnená sessionid, ktorá by mala byť uchovaná v tajnosti, jej prítomnosť v GET parametri otvára útočníkom možnosť nájsť platné sessionID na proxy serveroch cez, ktoré putuje požiadavka alebo aj v lokálnych logoch.

### Odstránenie zraniteľnosti

Na odstránenie vypísanej sessionID bolo treba vymazať príkaz ktorý ju URL vypisoval.

Ukážka kódu po odstránení vypisovanie sessionID



## Cross-site scripting

### Analýza zraniteľnosti

Cross-site scripting (XSS) je metóda narušenia WWW stránok využitím bezpečnostných chýb v skriptoch (predovšetkým neošetrené vstupy). Útočník vďaka chybám podstrčí do stránok vlastný kód, čo vyvoláva poškodenie vzhľadu stránok, ich znefunkčnenie, získavanie citlových údajov návstevníkov stránok, obídenie bezpečnostných prvkov aplikácie a phishing. V prípade, že sa do Search formulára zadáte nasledujúci payload, vyvolá sa útok typu Reflected XSS. Pomocou tohto útoku (s prispením sociálneho inžinierstva ) <script>alert(document.cookie)</script>

### Odstránenie zraniteľnosti

Zraniteľnosť na XSS sme odstránili pomocou funkcie real\_escape\_string, ktorá odstráni všetko špeciálne znaky.

Ukážka zdrojového kódu, ktorý odstráni zraniteľnsoť na XSS v súbore "search.php".



## Insecure Direct Object References

### Analýza zraniteľnosti

Nezabezpečené priame odkazy objektov je zraniteľnosť, ktorá umožňuje neoprávnením užívateľom získať prístup k súborom, ku ktorým nemajú mať prístup.

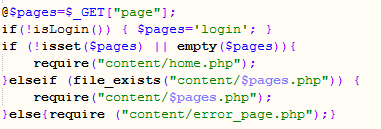
V prípade, že sa pokúsite načítať nasledujúcu linku podarí sa Vám získať prístup k súboru, kam by ste sa za normálnych okolností nemali nikdy dostať. Táto zraniteľnosť typu LFI (local file inclusion) spočíva v nesprávnom includovaní (vyžadovaní) súborov. http://192.168.56.102/udpb/www-vulnerable/?page=../../../../../etc/passwd

Zabrániť tomuto útoku je možné vhodným ošetrením vstupov, ktoré idú do funkcie include() alebo require().

### Odstránenie zraniteľnosti

Zraniteľnosť voči priamym odkazom sme ošetrili doplnením prípony ".php" za každý požadovaný súbor. Teraz sa útočníkovi nepodarí zobraziť hocijaký súbor.

Ukážka upraveného zdrojového súboru "index.php", ktorý načítava zadané súbory.

****

## Security Misconfiguration

### Analýza zraniteľnosti

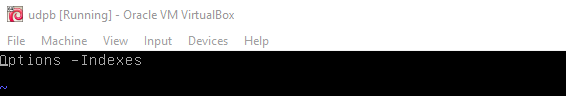
Zraniteľnosť chybnej konfigurácie je zraniteľnosť, ktorá môže uľahčiť prácu útočníkovi pri útoku na našu aplikáciu. Pre dobré zabezpečenie aplikácie treba mať bezpečne nakonfigurovanú platformu. Tak isto je potrebné aj mať aktuálnu verziu softwaru.

Aplikácia zlou konfiguráciou môže byť ľahko napadnuteľná. V aplikácií je možné prechádzať cez súbory na servery. Proti tomuto typu útoku treba vytvoriť .htaccess, ktorý bráni dir listingu.

### Odstránenie zraniteľnosti

Zraniteľnosť dir listingu sme odtrávnili vytvorením súboru .htaccess v zdrojovej zložke /var/www, ktorý bráni tomu typu útoku.

Obsah súboru .htaccess



## Sensitive Data Exposure

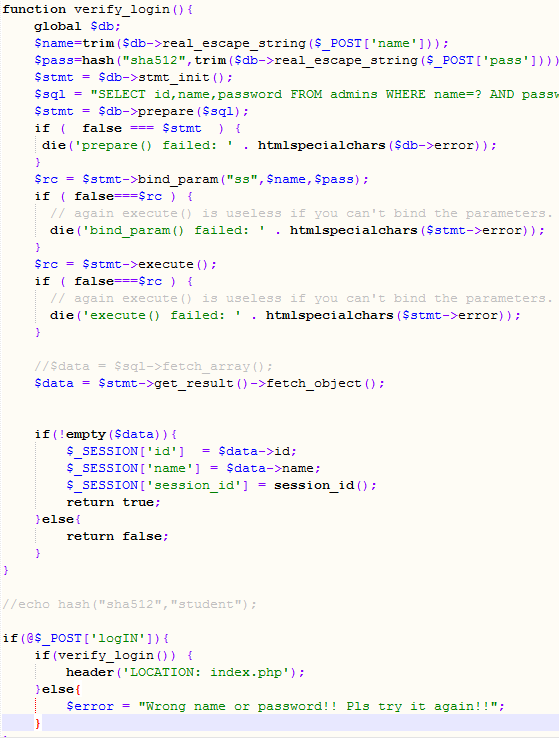
### Analýza zraniteľnosti

Niektoré aplikácie nesprávne chránia citlivé dáta, preto môžu mať k nim útočníci ľahký prístup. Tieto dáta si vyžadujú osobitnú ochranu. Prihlasovacie údaje sa prenášajú cez nezabezpečený HTTP protokol cez GET a prenos nie je šifrovaný.

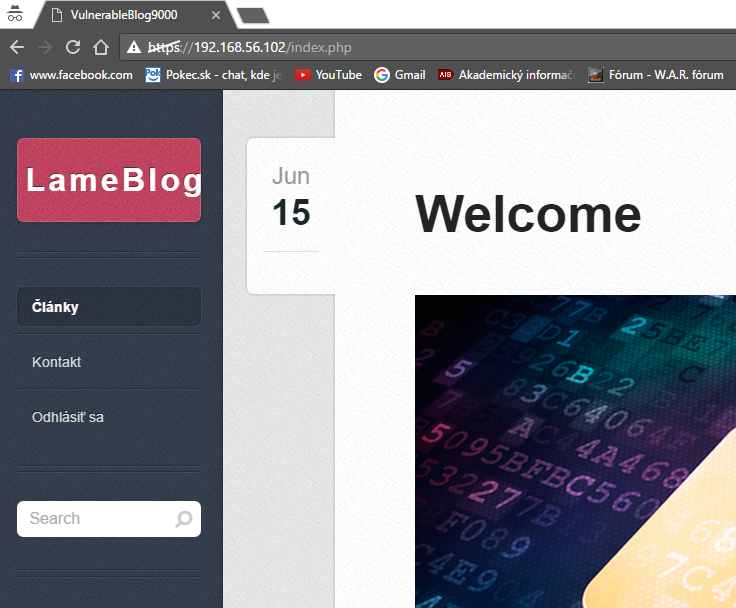
### Odstránenie zraniteľnosti

Zraniteľnosť odhalenia dát sme odstránili zmenou implementácie prihlasovacieho formuláru z metódy GET na POST. Tak isto sme doinštalovali SSL spojenie pre pristupovanie na stránky pod protokolom HTTPS. Pripojenie na stránku je šifrované a dáta sa odosielanou pomocou metódy POST.

Ukážka kódu prihlasovania súboru "index.php".



Ukážka pripojenia na stránku cez protokol HTTPS



## A7 - Missing Function Level Access Control

### Analýza zraniteľnosti

Aplikácia umožňuje neautentifikovaný prístup k stránkam, ku ktorým by mal byť povolený prístup iba po autetintifikácii, existuje zranitelnosť, keď odkazovaná zobrazí informácie, ktoré majú byť prístupné iba autentifikovaným užívateľom. Zraniteľnosť tohto webu spočíva v tom, že útočník je schopný načítať obsah stránok aj bez toho aby bol prihlásený.

### Odstránenie zraniteľnosti

Pristúpili sme na metódu, pri ktorej sme do priečinka constent s našimi podstránkami vložili súbor .htaccess a v ňom je jedna direktíva Deny from all. Tá zakáže priamy prístup k súborom v priečinku zo všetkých hostov, takže už nie je možné získať obsah súboru inak ako cez stránku index.php

## A8 - Cross-Site Request Forgery (CSRF):

### Analýza zraniteľnosti

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. Medzi vhodné nástroje na odstránenie tohto útočného vektora patrí implementácia CSRF tokenov Na kontaktnom a prihlasovacom formuláre nie sú implementované CSRF tokeny.

### Odstránenie zraniteľnosti

Na odstránenie zraniteľnosti sme implementovali použitie CSRF tokenov.

Ukážka kódu s implementáciou CSRF tokenov pri prihlásení.



Ukážka kódu s implementáciou CSRF tokenov pri posielaní mailu.



Následne dôkaz o dosiahnutí tejto zraniteľnosti už je nefunkčný. Po stlačení tlačidla submit je užívateľ presmerovaný na kontakt.php



## A9 - Using Components with Known Vulnerabilities

### Analýza zraniteľnosti

### Odstránenie zraniteľnosti

## A10 - Unvalidated Redirects and Forwards

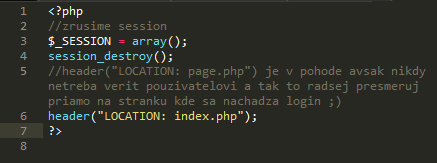
### Analýza zraniteľnosti

Webové aplikácie často presmerujú užívateľa na iné stránky a použijú nedôveryhodné údaje na určenie cieľovej stránky. Bez správneho overenia môže útočník presmerovať obeť na phishing alebo malware stránky.

### Odstránenie zraniteľnosti

Zraniteľnosť sme odstránili presmerovaním používateľa po odhlásení na pevnú stránku index.php. Na nej je implementovaná kontrola existencie session-u na serveri a pretože pri odhlásení sme v skripte logout.php zničili aktuálnu session, nie je možné, aby užívateľ prešiel kontrolou.

Súbor logout.php po úprave:



# Záver

Je potrebné dodržiavať určité bezpečnostné pravidlá ak chceme ochrániť našu aplikáciu pred únikom informácií. Preto pri úprave aplikácie sme sa snažili dodržiavať zadané bezpečnostné pravidlá, aby sme aplikáciu ochránili pred útokmi hackerov.