|  |
| --- |
|  |

|  |
| --- |
| Stručni kurs Razvoj bezbednog softvera |
| Izveštaj |
| Pronađene ranjivosti u projektu “RealBookStore” |
| Bojan Veličković 373/2020 |

# Istorija izmena

|  |  |  |  |
| --- | --- | --- | --- |
| Verzija | Datum | Izmenio/la | Komentar |
| 1.0 | 28.2.2024. | Uroš Dragojević | Kreiran izveštaj |
| 1.1 | 23.4.2024. | Bojan Veličković | Opisani propusti |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[Istorija izmena 1](#_Toc160050508)

[Uvod 3](#_Toc160050509)

[O veb aplikaciji 3](#_Toc160050510)

[Kratak pregled rezultata testiranja 3](#_Toc160050511)

[SQL injection 4](#_Toc160050512)

[Napad: Ubacivanje novog usera u tabelu “persons” (SQL injection) 4](#_Toc160050513)

[Metod napada: 4](#_Toc160050514)

[Predlog odbrane: 4](#_Toc160050515)

[Cross-site scripting 5](#_Toc160050516)

[Napad: Ubacivanje novog usera u tabelu “persons” 5](#_Toc160050517)

[Metod napada: 5](#_Toc160050518)

[Predlog odbrane: 5](#_Toc160050519)

[Zaključak 6](#_Toc160050520)

# Uvod

Ovaj izveštaj se bavi ranjivostima pronađenim u dole opisanoj veb aplikaciji.

## O veb aplikaciji

RealBookStore je veb aplikacija koja pruža mogućnosti pretrage, ocenjivanja i komentarisanja knjiga.

Aplikacija RealBookStore omogućava sledeće:

* Pregled i pretragu knjiga.
* Dodavanje nove knjige.
* Detaljan pregleda knjige kao i komentarisanje i ocenjivanje knjige.
* Pregled korisnika aplikacije.
* Detaljan pregled podataka korisnika.

## Kratak pregled rezultata testiranja

*Ovde idu kratko opisani rezultati testiranja: pronađene ranjivosti i nivo opasnosti.*

|  |  |
| --- | --- |
| *Nivo opasnosti* | *Broj ranjivosti* |
| *Low* | *3* |
| *Medium* | *2* |
| *High* | *1* |

# SQL injection

## Napad: Ubacivanje novog usera u tabelu “persons” (SQL injection)

### Metod napada:

Na stranici za detalje o knjigama, preko polja za ostavljanje komentara moguce je izvriti SQLI napad koji dodaje novog korisnika u bazu ostavljanjem komentara:

neki komentar’); insert into persons (id, firstName, lastName, email) values ( 10, ‘NekoIme’, ‘NekoPrezime’, ‘NekiMejl@mejl’) --

## Predlog odbrane:

U CommentRepository-ju, u metodi create za komentare, zameniti Statement klasu klasom PreparedStatement.

# Cross-site scripting

## Napad: Ubacivanje novog usera u tabelu “persons”

### Metod napada:

Prethodni primer mozemo nadograditi da ukljucuje i XSS napad unutar SQL Injectiona. Kao neki od atributa novog korisnika mozemo proslediti skriptu koja ce se pokrenuti pri pretrazi datog korisnika, recimo:

neki komentar’); insert into persons (id, firstName, lastName, email) values ( 10, ‘NekoIme’, ‘NekoPrezime’, ‘<img src="x" onerror="alert(document.cookie)"/>) --

## Predlog odbrane:

Onemoguciti da se ovakve stvari desavaju tako sto ce nasa aplikacija sve ove atribute posmatrati kao obican tekst i nista vise. (Koristicemo textContent umesto innerHTML)

# Cross-site request forgery

## Napad: Promena podataka korisnika

### Metod napada:

Napraviti laznu stranicu koja ce privuci korisnika da pritisne neko dugme, koje ce da pokrene java script kod koji menja podatke korisnika. Npr. dugme koje poziva funkciju:

function exploit() {  
 const formData = new FormData();  
 formData.append('id', 1);  
 formData.append('firstName', 'Batman');  
 formData.append('lastName', 'Dark Knight');  
 fetch('http://localhost:8080/update-person',  
 {method: 'POST', body: formData, credentials: 'include'});  
}

## Predlog odbrane:

Imati csrf token sesije koji se koristi pri slanju bilo kakvih zahteva.

# Zaključak

Popravljeni su naglaseni propusti koji su omogucavali SQLI, XSS, CSRF napade na odredjenim mestima unutar nase aplikacije.