|  |
| --- |
| Stručni kurs Razvoj bezbednog softvera |
| Izveštaj |
| Pronađene ranjivosti u projektu “RealBookStore” |

|  |
| --- |
| Tamara Šaponjić  8-22-2025 |

# Istorija izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Verzija** | **Datum** | **Izmenio/la** | **Komentar** |
| **1.0** | 22.08.2025 | Tamara Šaponjić | Kreiran izveštaj |
| **1.1** | 22.08.2025 | Tamara Šaponjić | SQL injection/XSS |
| **1.2** | 22.08.2025 | Tamara Šaponjić | Cross-site scripting |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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.*

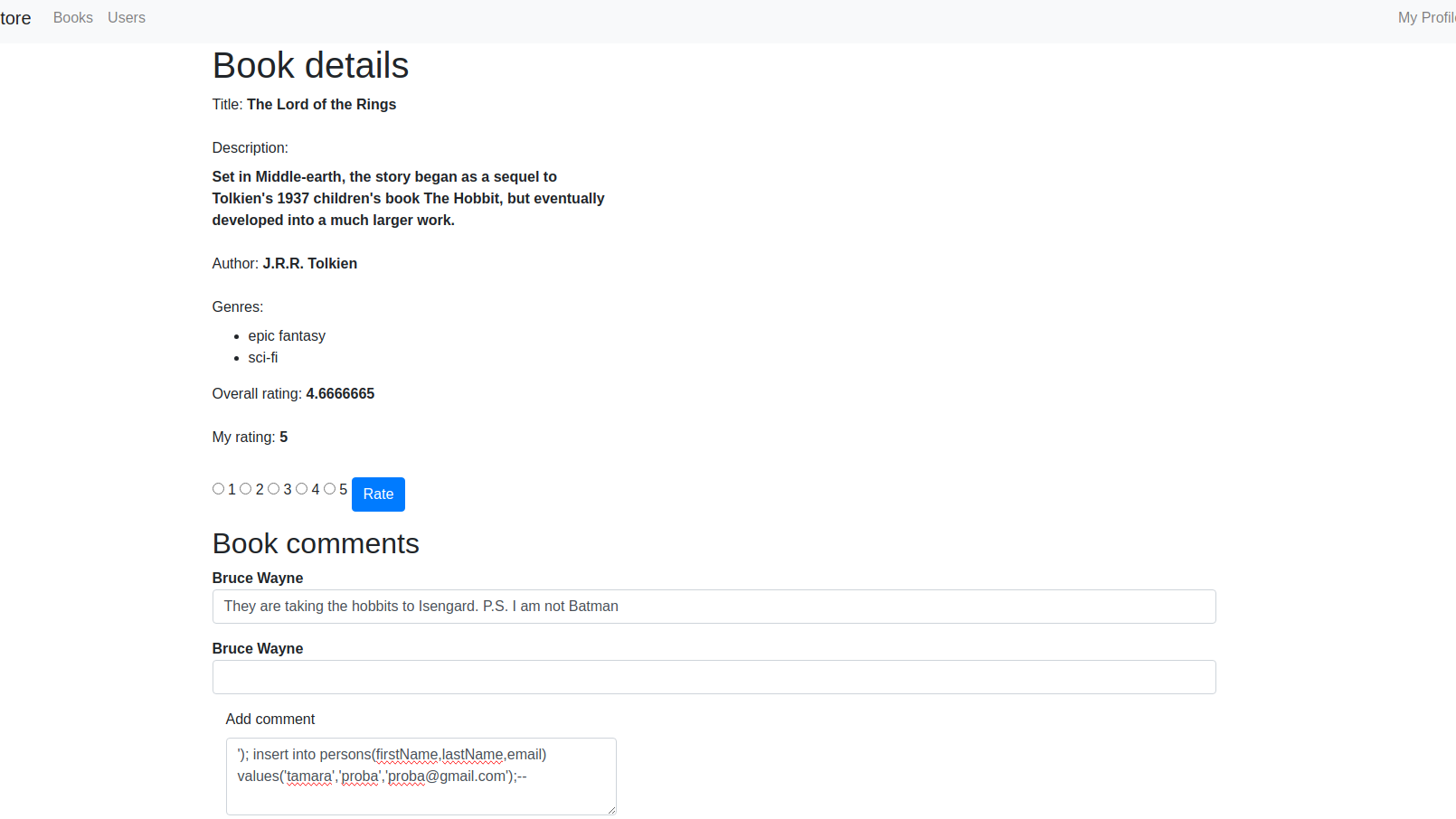
|  |  |
| --- | --- |
| ***Nivo opasnosti*** | ***Broj ranjivosti*** |
| ***Low*** | *3* |
| ***Medium*** | *2* |
| ***High*** | *1* |

# SQL injection

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

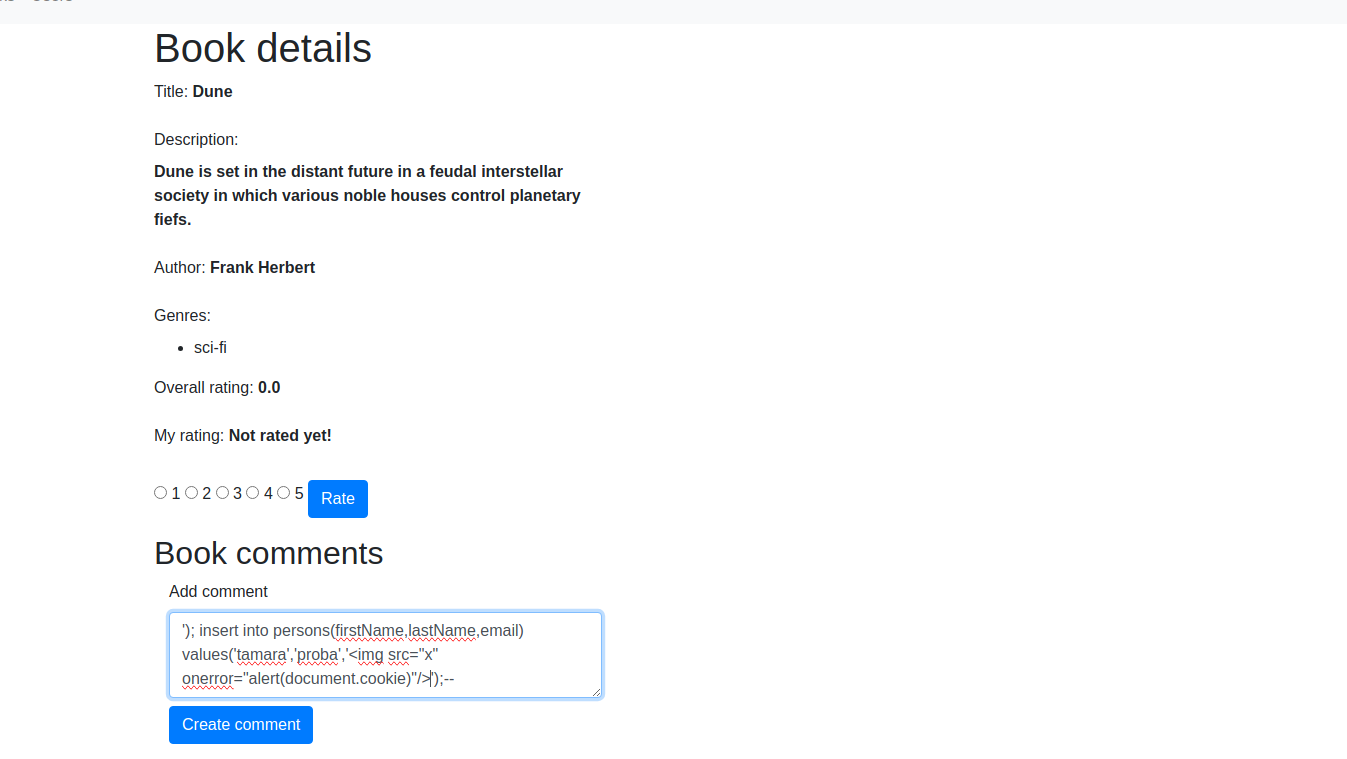
### Metod napada:

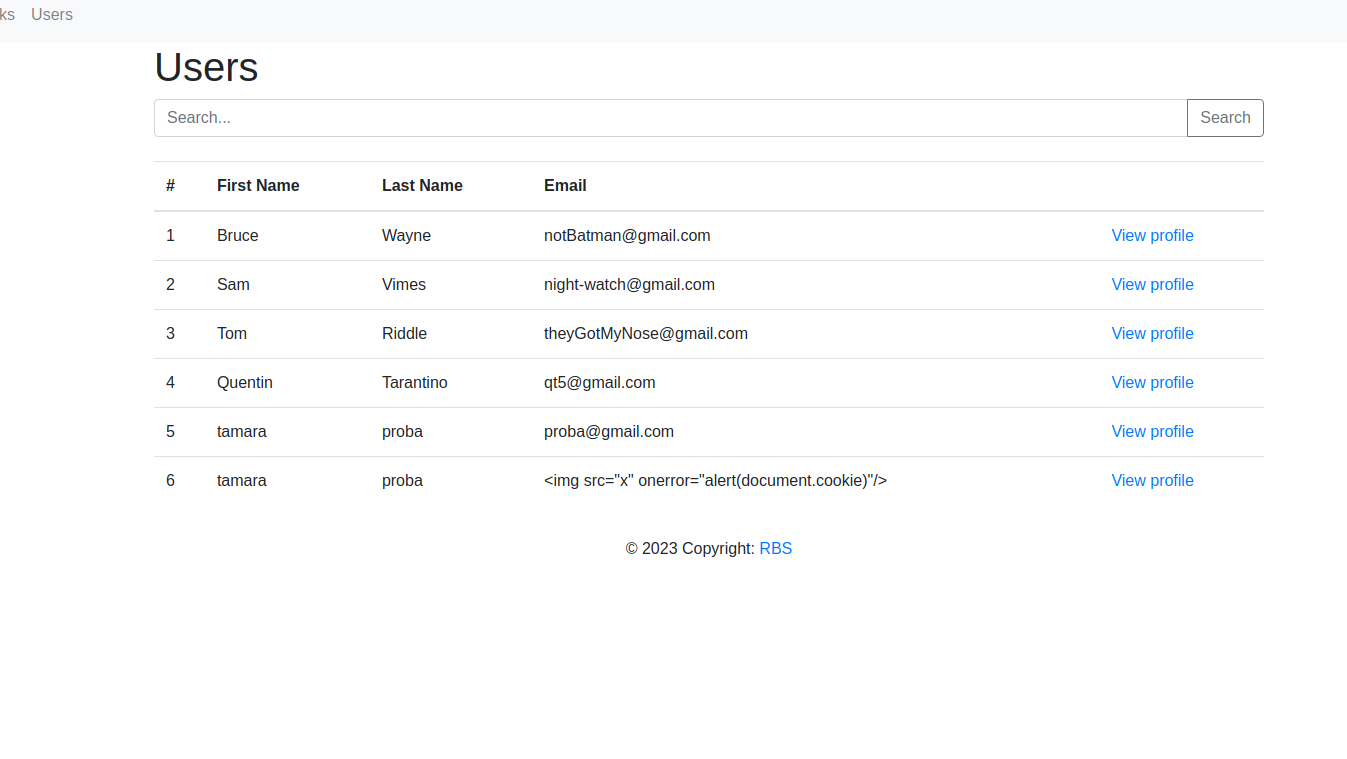
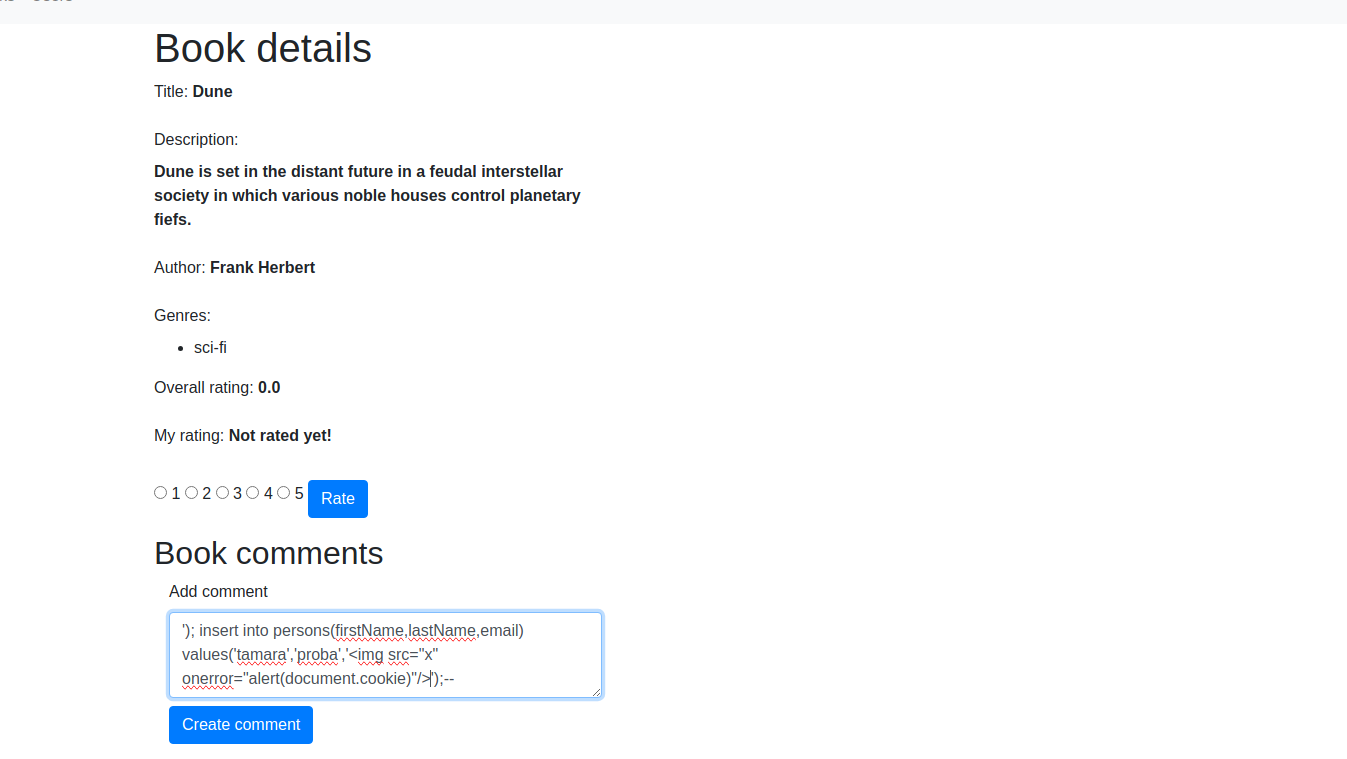
**1. SQL injection: Napad izvršavamo tako što unosemo upit u polje **„Book comments“ i** pokušavamo da unesemo novog korisnika u našu listu **„Persons“**.**



## 

**2. XSS: Napad XSS izvodimo tako što u polje „Book comments“ unosimo JavaScript kod umesto email adrese . Za razliku od prvog primera gde je unos bio bezopasan, sada pokušavamo da izvršimo maliciozni skript kroz komentar.**





## 

## Predlog odbrane:

**1. Zaštita od SQL injction-a** se postiže korišćenjem klase PreparedStatement umesto Statement. Na taj način svi uneti podaci tretiraju se kao običan tekst, a ne kao deo SQL komande, pa se onemogućava izvršavanje malicioznih upita. Rezultat je da se komentar i dalje uspešno dodaje, ali zloupotrebe poput kreiranja novog korisnika nisu moguće.

**2. Zaštita od XSS napada** podrazumeva nekoliko izmena. Pored upotrebe PreparedStatement objekata, u fajlu persons.html potrebno je umesto atributa innerHTML koristiti textContent, kako bi se sprečilo izvršavanje ubačenog JavaScript koda.

# Cross-site scripting

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

### Metod napada:

U okviru projekta csrf-exploit, u fajlu index.html treba ubaciti funkciju exploit. Ova funkcija treba da napravi zahtev ka serveru i izmeni podatke korisnika sa identifikatorom id=1, tako da njegovo ime bude postavljeno na „Batman“, a prezime na „Dark Knight“. Na taj način funkcija simulira CSRF napad, jer korisnik nesvesno izvršava akciju kojom se menjaju njegovi lični podaci.

## 

## 

## 

## Predlog odbrane:

U klasi PersonController, u metodi person uzimamo CSRF token iz korisničke sesije i prosleđujemo ga modelu. Zatim, u metodi updatePerson proveravamo da li se token koji je stigao iz forme poklapa sa onim koji je sačuvan u sesiji. Pored toga, u samu formu za izmenu podataka dodaje se skriveno polje (input) koje nosi vrednost tog CSRF tokena, kako bi se validacija mogla obaviti prilikom obrade zahteva.

# Zaključak

Za bezbednost web aplikacije primenjujemo više mehanizama zaštite. PreparedStatement sprečava SQL injekcije, prilagođen način prikazivanja sadržaja onemogućava XSS napade, dok upotreba CSRF tokena obezbeđuje da izmene podataka mogu vršiti samo ovlašćeni korisnici. Kombinacijom ovih tehnika aplikacija postaje otpornija i zaštićena od najčešćih bezbednosnih pretnji.