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

|  |
| --- |
| Miloš Obrenović  29.4.2024 |

# Istorija izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Verzija** | **Datum** | **Izmenio/la** | **Komentar** |
| **1.0** | 29.4.2024. | Miloš Obrenović | Kreiran izveštaj |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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* |

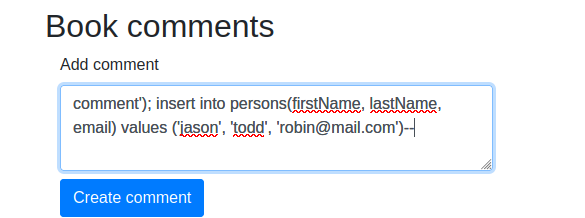
# SQL injection

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

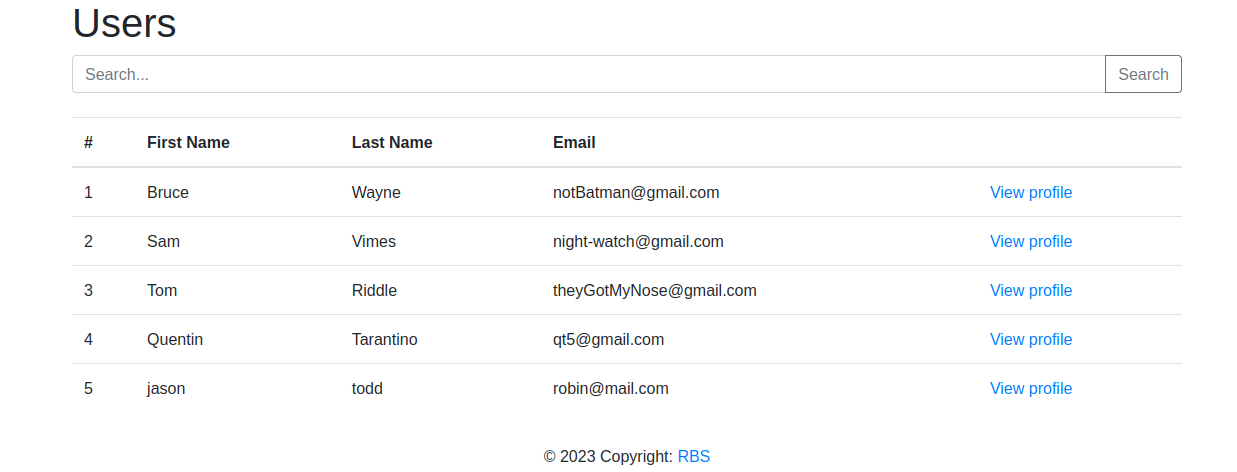
### Metod napada:

Na stranici za pregledanje pojedinačne knjige , uneti sledeći kod u input polje “Comment”:

// opasan kod



Nakon čega se vidi da je novi korisnik dodat u bazu:



## Predlog odbrane:

Potrebno je da u našem kodu koristimo parametrizovane upite za dobavljanje komentara.

# Cross-site scripting

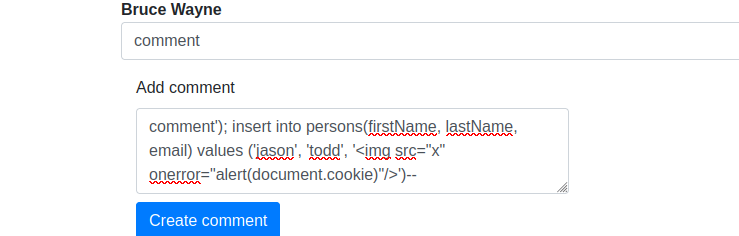
## Napad: Ubacivanje novog usera u tabelu “persons”

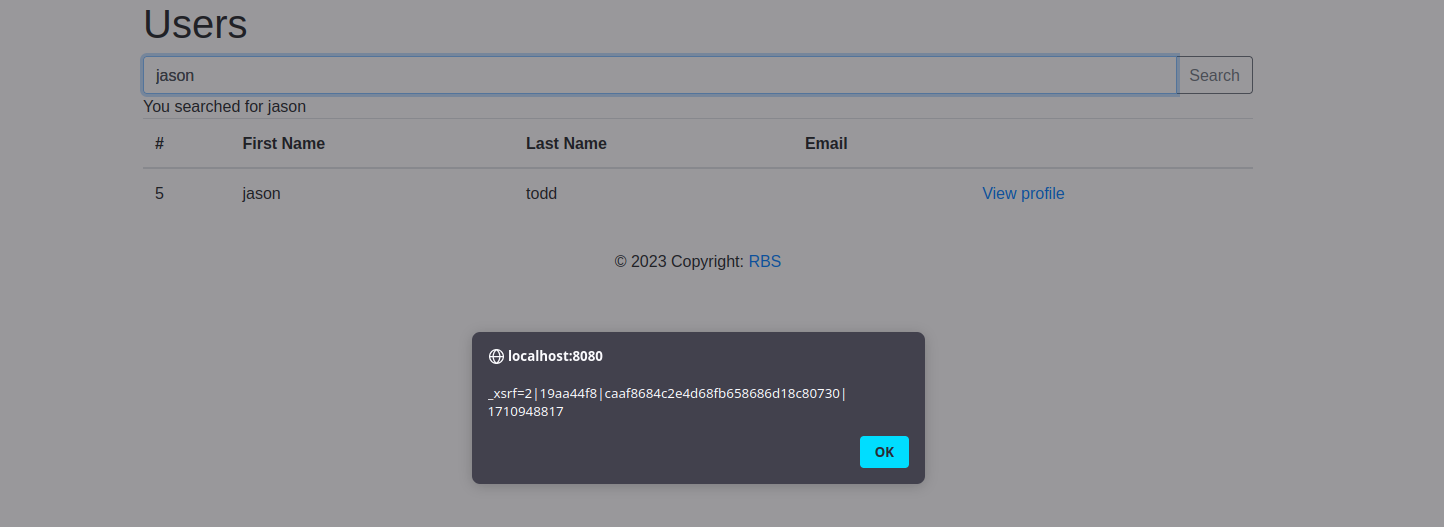
### Metod napada:

Na stranici za pregledanje pojedinačne knjige , uneti sledeći kod u input polje “Comment”:

// opasan kod

Kombinovanjem SQL Injection i XSS napada možemo da ubacimo u bazu korisnika koji će kao neki od svojih atributa imati zlonamernu skriptu.





## Predlog odbrane:

Potrebno je da u okviru HTML DOM objekta koristimo textContent umesto innerHTML, I da na HTML tagu umesto th:text koristimo th:utext(unescaped text)

# Cross-site request forgery

## Napad: Menjanje podataka usera.

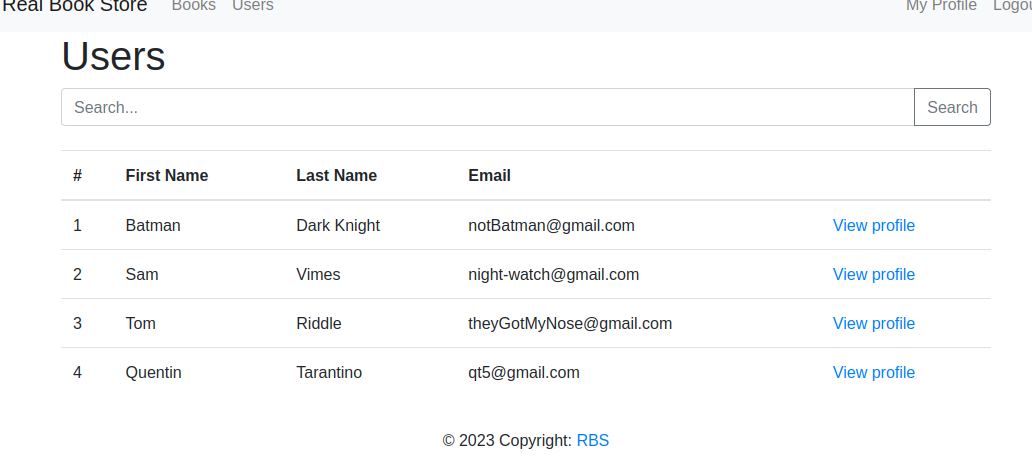
## Metod napada:

Klikom na maliciozni link, pokreće se skripta koja šalje zahtev serveru i menja podatke korisnika.

Exploit funkcija:

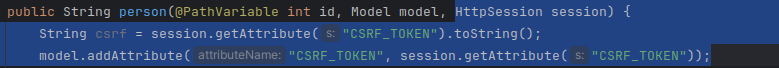


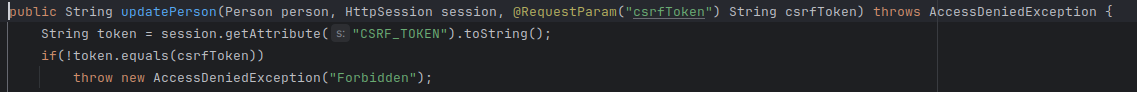
Klikom na link vidimo da se stvarno promenio korisnik:

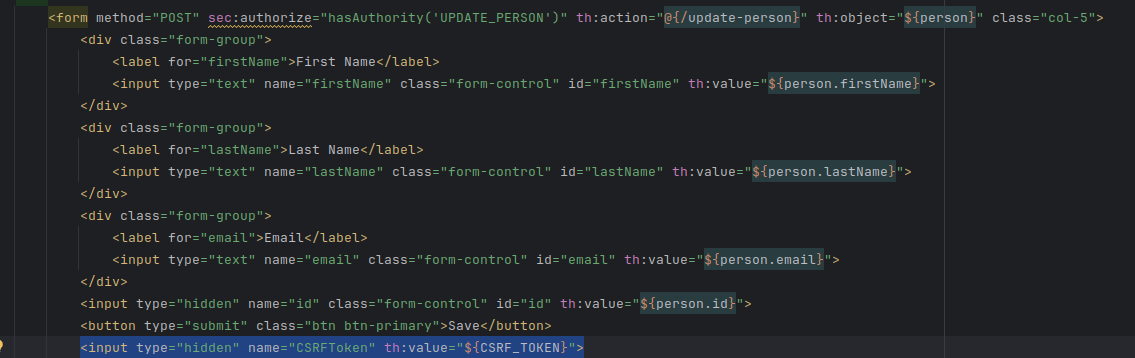


## Predlog odbrane:

Da bismo se odbranili od ovog tipa napada, potrebno je da kreiramo token pomoću CSPRNG na početku sesije korisnika i da se zatim token uskladišti u podatke sesije korisnika.



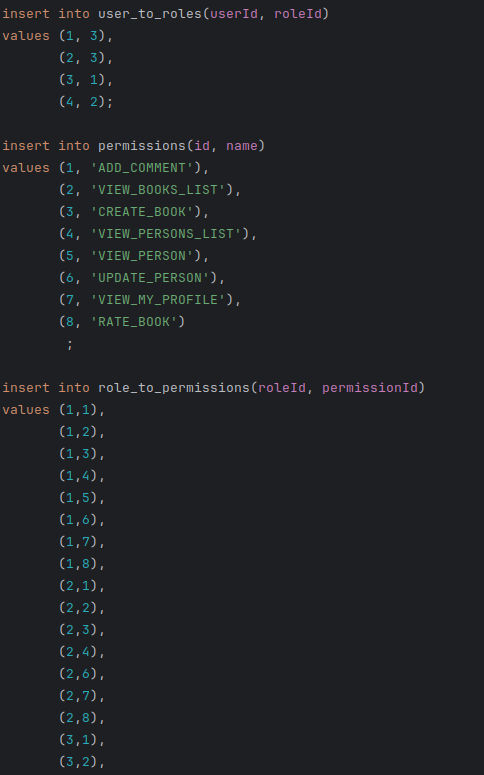




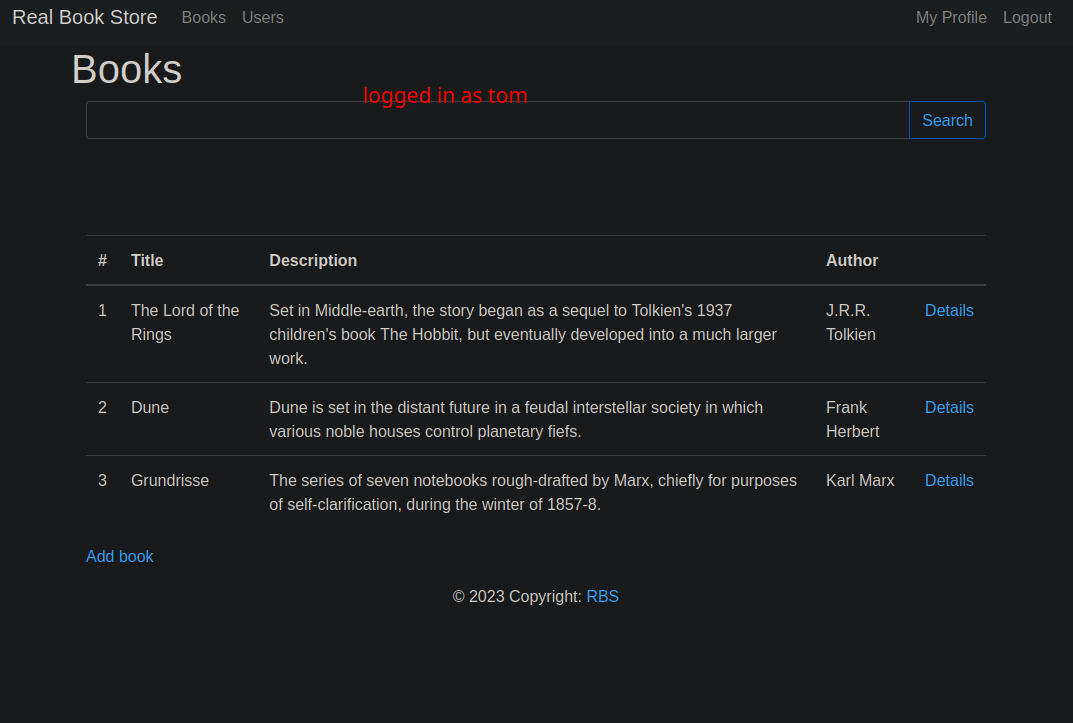
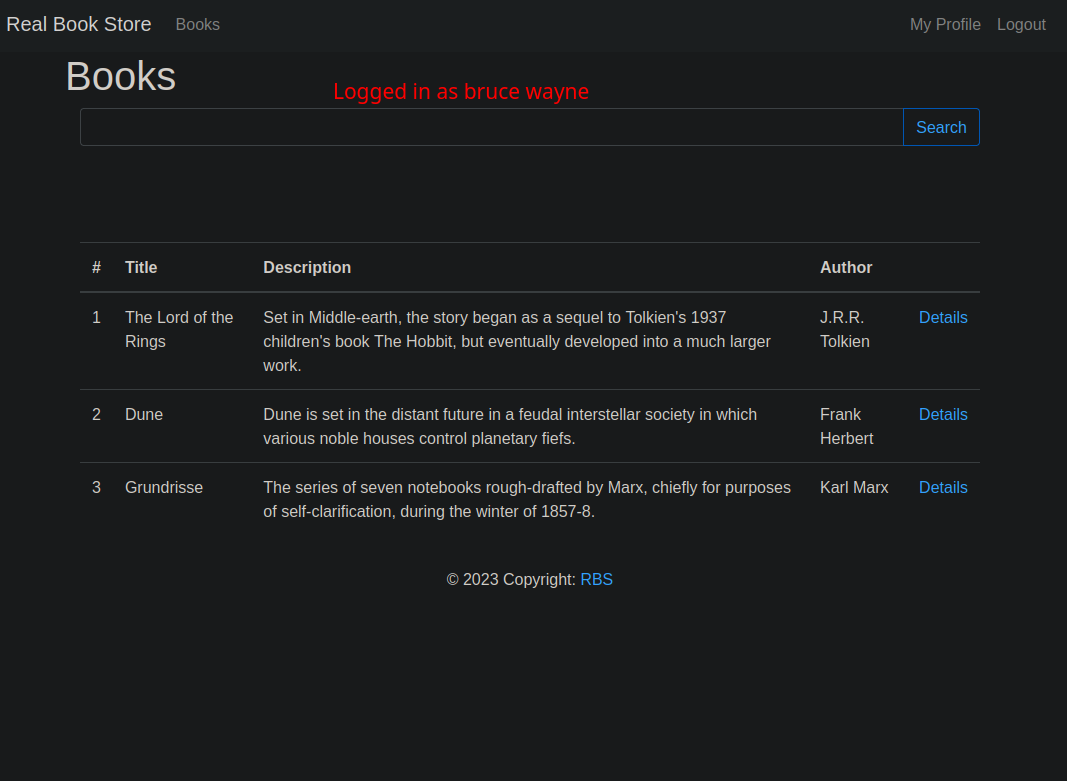
I time smo popravili ovu ranjivost.

# Implementacija autorizacije

Što se tiče autorizacije, prvo moramo implementirati autorizacioni model u bazi podataka.



Nakon toga sledi učitavanje permisija i provera permisija pri pristupanju podacima.



# DevOps

Kod DevOps-a smo koristili auditing i logging radi pratnje promena i grešaka u našem programu. Logging je korišćen za warning, ukoliko neki upit ne prođe uspešno ili ako npr. search nije dobro izvršen. Auditing je korišćen za bilo koju promenu podataka (knjige ili usera), zajedno sa informacijama o staroj/novoj vrednosti.

# Zaključak

Uopšteni zaključci testiranja i predlozi za sledeće korake kako bi najbolje obezbedili aplikaciju.