|  |
| --- |
| Ana Mihajlović, 95/2020 |

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

# Istorija izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Verzija** | **Datum** | **Izmenio/la** | **Komentar** |
| **1.0** | 28.2.2024. | Uroš Dragojević | 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* |

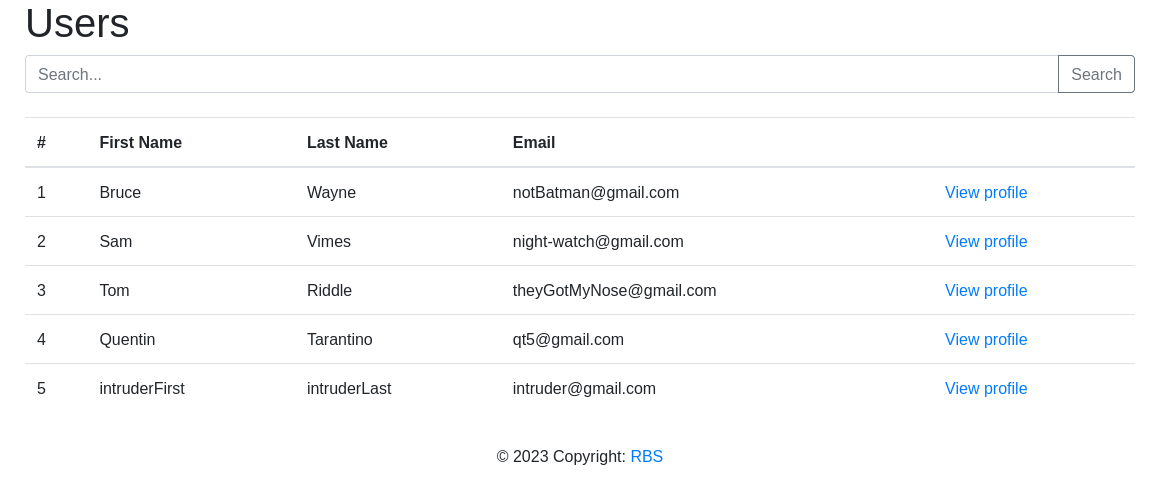
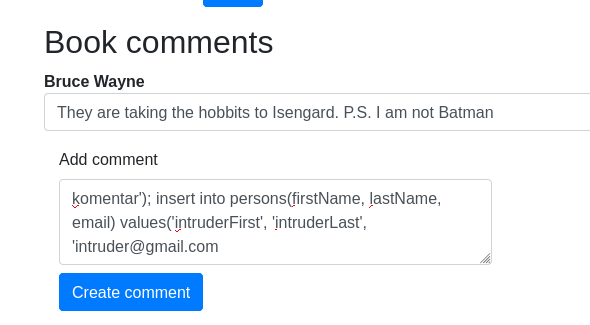
# SQL injection

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

### Metod napada:

Na stranici Persons aplikacije, uneti sledeći kod u input polje “First Name”:

komentar'); insert into persons(firstName, lastName, email) values('intruderFirst', 'intruderLast', ['intruder@gmail.com](mailto:'intruder@gmail.com)



## Odbrana:

Koristimo klasu PreparedStatement umesto klase Statement.

# Cross-site scripting

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

### Metod napada:

Injektiramo novog korisnika u tabelu persons koji ce kao polje da ima malicioznu skriptu.

komentar’); insert into persons(firstName, lastName, email) values ('Maliciozni','Uljez','<img src="x" onerror="alert(document.cookie)"/>

## Predlog odbrane:

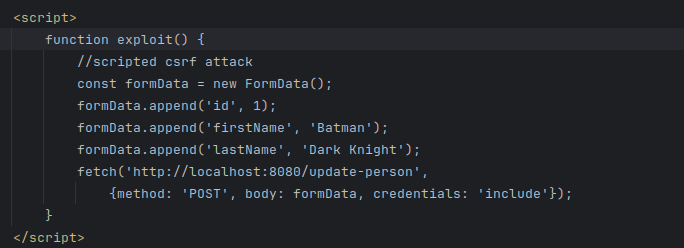
U persons.html menjam innerHTML u textContent kako bi tretirali polja koja se unose kao obican tekst.

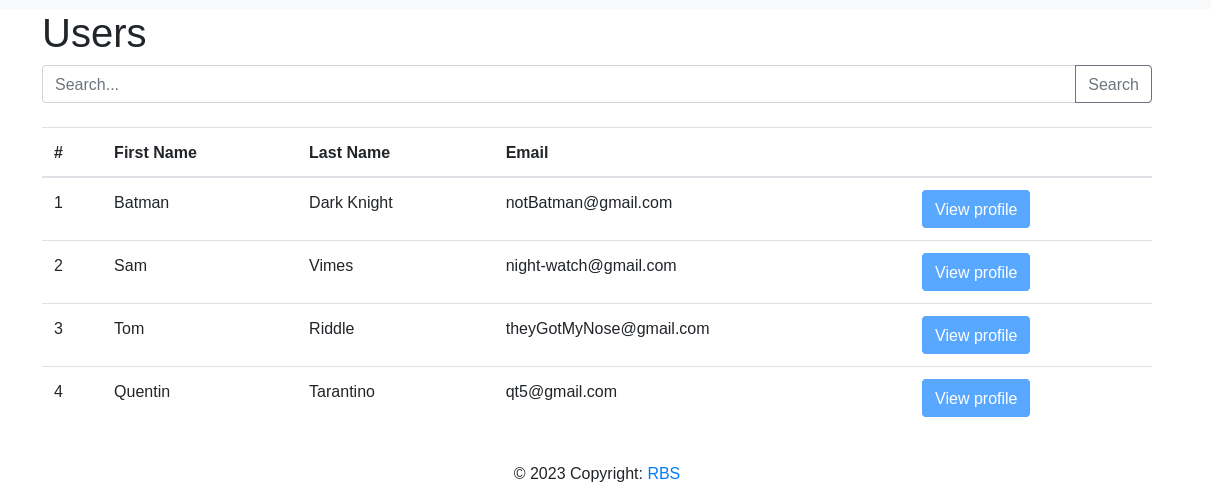
# Cross-site request forgery

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

### Metod napada:

Navodimo korisnika da klikne dugme na stranici što će u pozadini poslati zahtev serveru za promenu podatka u bazi.

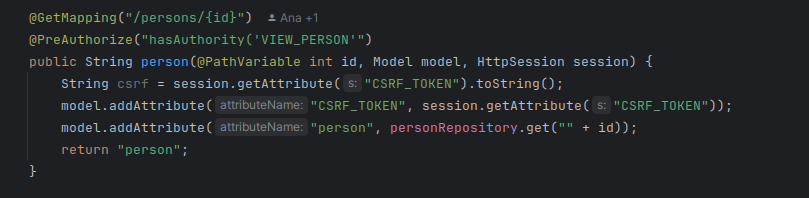


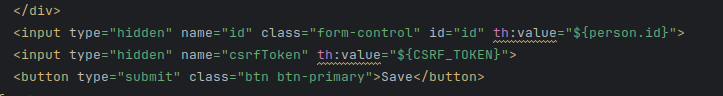


## Predlog odbrane:

Dodajemo token, server će pri svakom zahtevu da proverava da li primljeni token odgovara onom uskladištenom u podacima sesije korisnika.

Metod person u klasi PersonController je zadužen za prikazivanje stranice za promenu detalja korisnika. Dodajemo kod koji čita token iz sesije I upisuje ga u model kako bi omogućili da token bude upisan deo HTML stranice za promenu detalja korisnika.

person.htlm – stranica za promenu detalja korisnika

u formu dodajemo input koji će da sadrži vrednost csrf tokena (čitamo iz modela). Token će biti poslat svaki put kada se izvrši submit forme.

U PersonsController dodajemo proveru tokena nakon svakog submita korisnika.

# Zaključak

Implementirane su popravke za SQL Injection, XSS I CSRF.