

Stručni kurs Razvoj bezbednog softvera

# Izveštaj

Pronađene ranjivosti u projektu “RealBookStore”

Jana Radojevic

09/29/25

## Istorija izmena

[illegible]

# Sadržaj

Istorija izmena.....	1
Uvod.....	3
O veb aplikaciji.....	3
Kratak pregled rezultata testiranja.....	3
SQL injection.....	4
Napad: Ubacivanje novog usera u tabelu “persons” (SQL injection).....	4
Metod napada:.....	4
Predlog odbrane:.....	4
Cross-site scripting.....	5
Napad: Ubacivanje novog usera u tabelu “persons”.....	5
Metod napada:.....	5
Predlog odbrane:.....	5
Zaključak.....	6

# 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.*

<b>Nivo opasnosti</b>	<b>Broj ranjivosti</b>
<b>Low</b>	3
<b>Medium</b>	2
<b>High</b>	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”:

## Book comments

**Bruce Wayne**

They are taking the hobbits to Isengard. P.S. I am not Batman

Add comment

```
insert into persons (firstName, lastName, email) values  
( 'Evil', 'Comment', 'evilemail@gmail.com' )
```

Create comment

insert into persons ( 'Evil', 'Comment', 'evilemail@gmail.com' )

Ovim je u nasu bazu podataka unet User pomocu upita, a to ne bi smelo da bude dozvoljeno.

**Predlog odbrane:**

Potrebno da je da u kodu koristimo parametrizovane upite.

## Cross-site scripting

Napad: Ubacivanje novog usera u tabelu “persons”

Metod napada:

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

## Book comments

**Bruce Wayne**

They are taking the hobbits to Isengard. P.S. I am not Batman

Add comment

```
insert into persons (firstName, lastName, email) values  
( 'Evil', 'Comment', 'evilemail@gmail.com' )
```

Create comment

Predlog odbrane:

Umesto `th:text` u HTML tagovima treba dakoristimo `th:utext(unescaped text)`, I umesto `textContent` trebalo bi da koristimo `innerHTML` radi osiguranja koda

# Cross-site request forgery

Napad: Promena user podataka

Metod napada:

Maliciozna stranica (/3000) sadrži element (npr. sliku pehara) koji, pri kliku, šalje POST zahtev na /update-person sa podacima za izmenu korisnika ID=1. Korisnik koji je ulogovan u aplikaciju nenamerno aktivira zahtev, zbog čega se njegovo ime i prezime promene.

Nakon napada se vidi sledece:

Users				
<input type="text" value="Search..."/>			<input type="button" value="Search"/>	
#	First Name	Last Name	Email	
1	Batman	Dark Knight	notBatman@gmail.com	<a href="#">View profile</a>
2	Sam	Vimes	night-watch@gmail.com	<a href="#">View profile</a>
3	Tom	Riddle	theyGotMyNose@gmail.com	<a href="#">View profile</a>
4	Quentin	Tarantino	qt5@gmail.com	<a href="#">View profile</a>

© 2023 Copyright: [RBS](#)

Predlog odbrane:

Implementirati CSRF zaštitu pomoću tokena. Svaka forma za izmenu sadrži sakriveni CSRF token koji se prilikom slanja zahteva proverava na serveru; promena podataka se prihvata samo ako je token validan.

# Autorizacija

Implementiramo autorizacioni model u bazi podataka:

```
insert into user_to_roles(userId, roleId)
values (1, 3),
       (2, 3),
       (3, 1),
       (4, 2);

insert into permissions(id, name)
values (1, 'ADD_COMMENT'),
       (2, 'VIEW_BOOKS_LIST'),
       (3, 'CREATE_BOOK'),
       (4, 'VIEW_PERSONS_LIST'),
       (5, 'VIEW_PERSON'),
       (6, 'UPDATE_PERSON'),
       (7, 'VIEW_MY_PROFILE'),
       (8, 'RATE_BOOK')
       ;

insert into role_to_permissions(roleId, permissionId)
values (1,1),
       (1,2),
       (1,3),
       (1,4),
       (1,5),
       (1,6),
       (1,7),
       (1,8),
       (2,1),
       (2,2),
       (2,3),
       (2,4),
       (2,6),
       (2,7),
       (2,8),
       (3,1),
       (3,2),
```

Da bi korisnik pristupio podacima mora imati odgovarajući nivo autorizacije.



## DevOps

U okviru DevOps prakse u projektu implementirani su **auditing** i **logging** mehanizmi radi praćenja promena i detektovanja grešaka u radu aplikacije.

- **Logging** je korišćen za beleženje upozorenja i grešaka, na primer u slučajevima kada upit nije uspešno izvršen ili kada pretraga nije dala očekivane rezultate.
- **Auditing** je korišćen za evidentiranje svih promena nad podacima (npr. knjige ili korisnici), pri čemu su se beležile i stare i nove vrednosti, kako bi se omogućilo potpuno praćenje istorije izmena.

## Zaključak

Implementirali smo popravke za SQL Injection, XSS i CSRF, kao i autorizaciju korisnika i logovanje grešaka. Tokom implementacije, usled nepažnje, došlo je do stvaranja propusta koji potencijalno mogu predstavljati ozbiljnu pretnju po bezbednost aplikacije. Iako su u pitanju sitni propusti, oni ukazuju na aspekte na koje u budućnosti treba obratiti pažnju. Posebno je važno voditi računa o metodama koje na prvi pogled deluju slično, ali se u svojoj suštini razlikuju i proizvode drugačije efekte.