

# Stručni kurs Razvoj bezbednog softvera

# Sadržaj

Uvod.....	1
SQL injection.....	2
Cross-site scripting.....	3
Cross-site request forgery.....	4
Authorization.....	5
Zaključak.....	6

# Uvod

RealBookStore je veb aplikacija koja nam služi na ovom kursu za potrebe seminarskog rada, gde ćemo demonstrirati primere napada i odbrane koristeći postojeće ranjivosti aplikacije.

U okviru RealBookStore aplikacije moguće su sledeće funkcionalnosti: pretraga, pregled, komentarisanje, dodavanje, ocenjivanje knjiga, kao i pregled korisnika aplikacije.

# SQL Injection

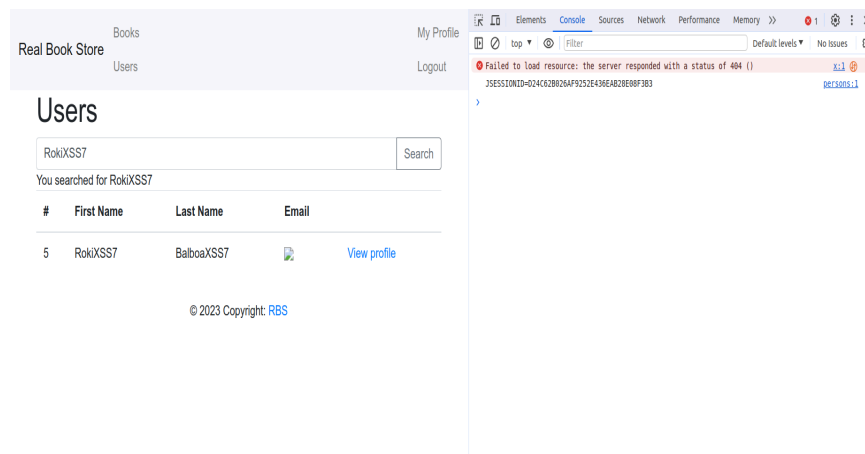
Napad: Ubacivanje novog korisnika u tabelu „persons“

Način na koji ćemo primeniti ovaj napad je korišćenje ranjivosti RealBookStore aplikacije na SQL Injection, prilikom komentarisanja knjige.

Konkretno na neku od ponuđenih knjiga ostavićemo komentar:

*comment'); insert into persons(firstName, lastName, email) values ('RokiXSS7', 'BalboaXSS7', '')--*

Nakon što smo uneli komentar, primećujemo da je nastao novi korisnik aplikacije, korisnik kojeg smo uspesno ubacili u sistem našim napadom!



Na ovoj slici ujedno vidimo i posledice još jednog napada kojeg ćemo kasnije objasniti!

Odbrana:

Parametrizovani upiti za dohvaćanje komentara su spas od ovakvog napada!

# Cross-site scripting

Napad: Ubacivanje novog korisnika u tabelu „persons“

U prošlom poglavlju smo opisali način na koji ćemo ujedno izvesti i ovaj napad. Konkretno, kada smo ubacili novog korisnika u bazu korisnika, ujedno smo i ubacili zlonamernu skriptu, koja će nam otkriti vrednost kolačića trenutne korisničke sesije.

*comment'); insert into persons(firstName, lastName, email) values ('RokiXSS7', 'BalboaXSS7', '')--*

The screenshot shows a web application interface for 'Real Book Store'. The 'Users' section is active, displaying a search result for 'RokiXSS7'. The search results table shows a user with ID 5, first name 'RokiXSS7', last name 'BalboaXSS7', and an email field with a broken image icon and a 'View profile' link. The browser's developer console on the right shows a 404 error: 'Failed to load resource: the server responded with a status of 404 ()'. The error details include 'JSESSIONID=024C62B026AF9252E430EAB28E08F3B3' and the resource path 'persons:1'.

#	First Name	Last Name	Email
5	RokiXSS7	BalboaXSS7	<a href="#">View profile</a>

© 2023 Copyright: [RBS](#)

Odbrana:

Korišćenje innerHTML-a nosi sa sobom razne rizike, a to se ovde najbolje pokazalo. Umesto toga puno je bezbednije koristiti `textContent`.

# Cross-site request forgery

Napad: Menjanje podataka korisnika.

Koristimo sledeću exploit funkciju:

```
6 csrf-exploit/index.html
@@ -14,6 +14,12 @@ <h1>Click here!</h1>

14 14      <script>
15 15          function exploit() {
16 16              // Scripted CSRF Request
17 17              const formData = new FormData();
18 18              formData.append('id', 1);
19 19              formData.append('firstName', 'Batman');
20 20              formData.append('lastName', 'Dark Knight');
21 21              fetch('http://localhost:8080/update-person', {method : 'POST', body : formData, credentials : 'include'});
22 22          }
17 23      }
18 24      </script>
19 25  </body>
```

A nakon što korisnik klikne na ovaj maliciozni link, šalje se zahtev serveru, nakon kojeg će se podaci korisnika sa vrednošću id-a 1 promeniti na način kao na slici iznad.



Primer malicioznog linka.

Ima li nekog da ne voli da pobjedi i da ne bi kliknuo!?

Click here!

Kao što vidimo, napad je uspešno prošao

Users				
Search...			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>

Odbrana:

Pomoću CSPRNG ćemo kreirati token na početku korisničke sesije korisnika, koji ćemo uskladištiti u podatke sesije korisnika.

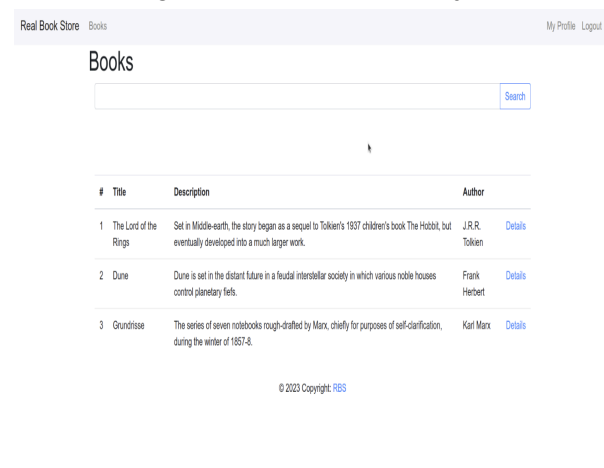
# Autorizacija

```
51 + insert into permissions(id, name)
52 + values (1, 'ADD_COMMENT'),
53 +       (2, 'VIEW_BOOKS_LIST'),
54 +       (3, 'CREATE_BOOK'),
55 +       (4, 'VIEW_PERSONS_LIST'),
56 +       (5, 'VIEW_PERSON'),
57 +       (6, 'UPDATE_PERSON'),
58 +       (7, 'VIEW_MY_PROFILE'),
59 +       (8, 'RATE_BOOK')
60 +       ;
61 +
62 + insert into role_to_permissions(roleId, permissionId)
63 + values (1,1),
64 +       (1,2),
65 +       (1,3),
66 +       (1,4),
67 +       (1,5),
68 +       (1,6),
69 +       (1,7),
70 +       (1,8),
71 +       (2,1),
72 +       (2,2),
73 +       (2,3),
74 +       (2,4),
75 +       (2,6),
76 +       (2,7),
77 +       (2,8),
78 +       (3,1),
79 +       (3,2),
80 +       (3,6),
81 +       (3,7),
82 +       (3,8)
83 +       ;
```

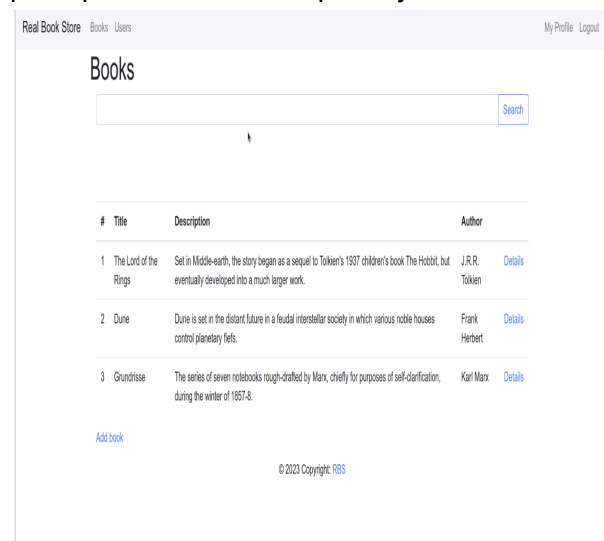
Implementiramo autorizacioni model u bazi podataka na sledeći način. Neophodno je svakoj roli odrediti koje funkcionalnosti su joj odobrene.

Nakon toga možemo proveriti ispravnost implementacije.

Pogled iz ugla Bruce Wayne-a, koji ne može da pristupi pregledu korisnika aplikacije.



Pogled iz ugla Quentina Tarantina, koji može pristupiti listi korisnika aplikacije!



# Zaključak

U ovom radu, prikazane su neke od poznatih ranjivosti, kao i načini njihove eksploatacije, i na koji način se od napada odbraniti!