

Stručni kurs Razvoj bezbednog softvera

Izveštaj

Pronađene ranjivosti u projektu “RealBookStore”

Dragana Katic

9-14-2025

Istorija izmena

Verzija	Datum	Izmenio/la	Komentar
1.0	14.09.2025.	Dragana Katic	SQL injection
2.0.	14.09.2025.	Dragana Katic	Cross-site scripting
3.0.	14.09.2025.	Dragana Katic	Autorizacija
4.0	14.09.2025.	Dragana Katic	Izvestaj

Sadržaj

Istorija izmena	1
Uvod	3
O veb aplikaciji	3
Kratak pregled rezultata testiranja	3
2. SQL injection i Cross-site scripting.....	4
2.1. SQL injection.....	4
Napad: Ubacivanje novog usera u tabelu “persons” (SQL injection)	4
Metod napada:.....	4
Predlog odbrane:	5
2.2. Cross-site scripting.....	7
Napad: Ubacivanje novog usera u tabelu “persons”	7
Metod napada:	7
Predlog odbrane:	8
3. Cross-Site Request Forgery(CSRF)	10
Predlog odbrane:	11
4. Autorizacija	12

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.

<i>Nivo opasnosti</i>	<i>Broj ranjivosti</i>
Low	3
Medium	2
High	1

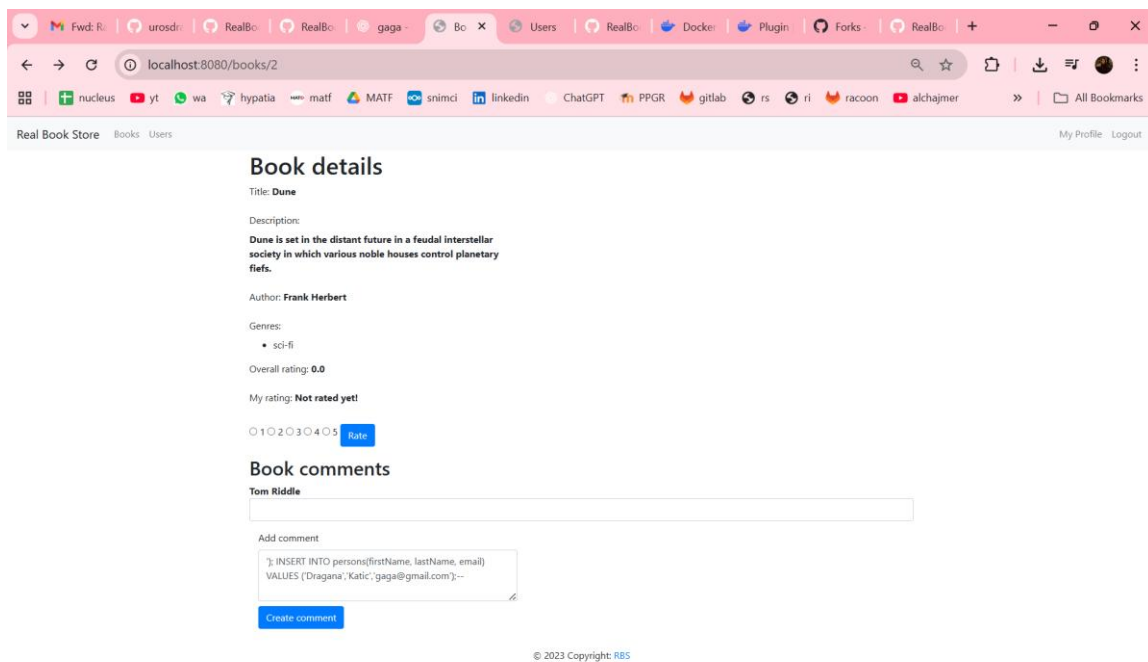
2. SQL injection i Cross-site scripting

2.1. SQL injection

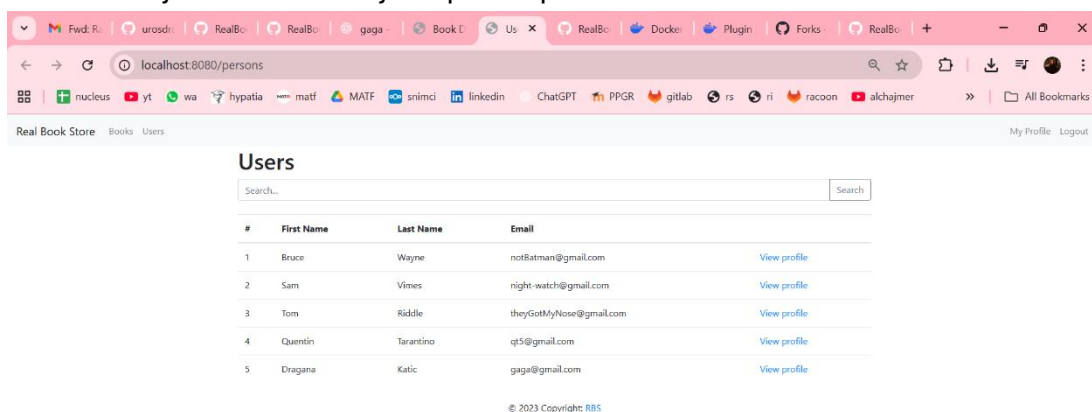
Napad: Ubacivanje novog usera u tabelu “persons” (SQL injection)

Metod napada:

U polje za unos komentara knjige ubačeno je sledece:



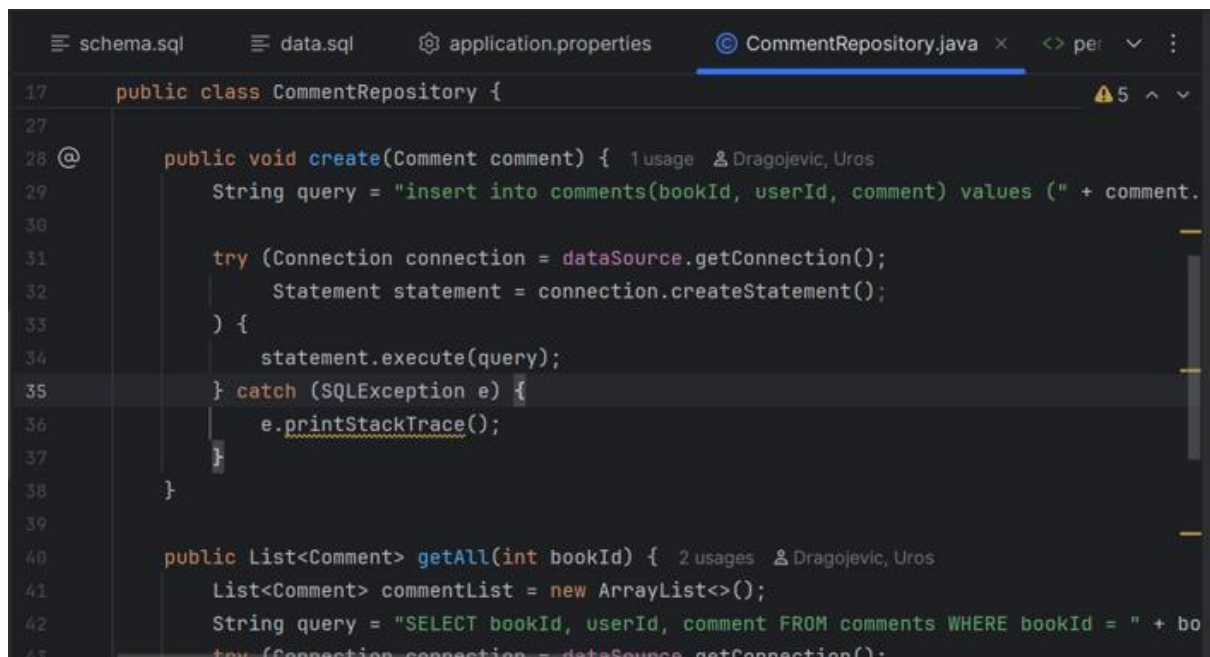
Na sledecoj slici vidimo da je napad uspeo i da se u listu usera dodao novi user:



Predlog odbrane:

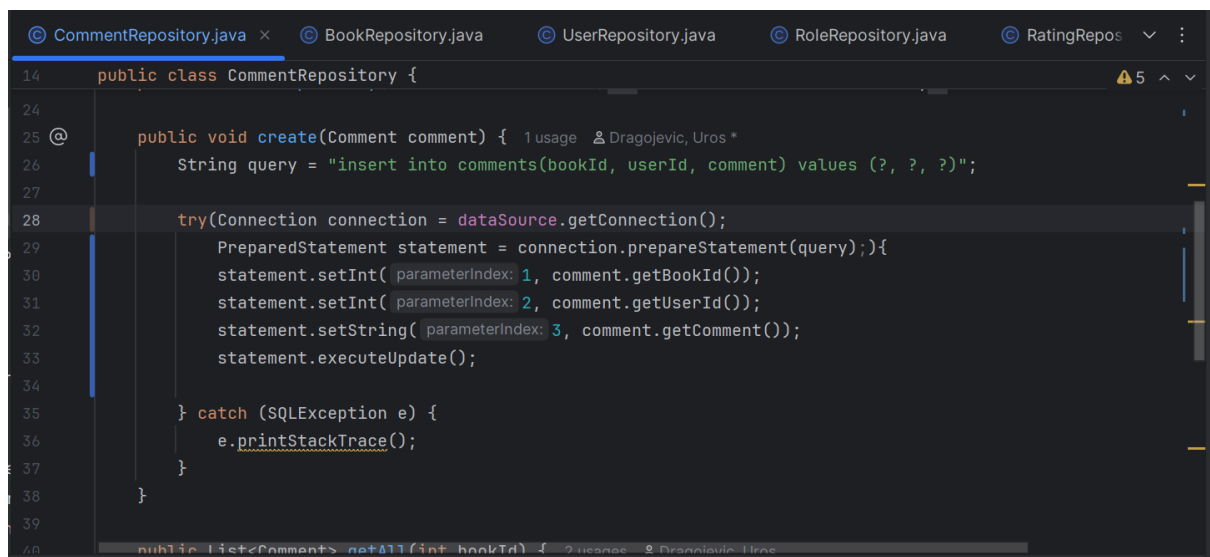
Koristicemo PreparedStatement umesto Statement.

Stari kod:



```
17 public class CommentRepository {
28 @
29     public void create(Comment comment) { 1 usage & Dragojevic, Uros
30         String query = "insert into comments(bookId, userId, comment) values (" + comment.
31
32         try (Connection connection = dataSource.getConnection();
33             Statement statement = connection.createStatement();
34         ) {
35             statement.execute(query);
36         } catch (SQLException e) {
37             e.printStackTrace();
38         }
39
40     public List<Comment> getAll(int bookId) { 2 usages & Dragojevic, Uros
41         List<Comment> commentList = new ArrayList<>();
42         String query = "SELECT bookId, userId, comment FROM comments WHERE bookId = " + bo
43         try (Connection connection = dataSource.getConnection();
```

Novi kod:



```
14 public class CommentRepository {
24
25 @
26     public void create(Comment comment) { 1 usage & Dragojevic, Uros *
27         String query = "insert into comments(bookId, userId, comment) values (?, ?, ?)";
28
29         try (Connection connection = dataSource.getConnection();
30             PreparedStatement statement = connection.prepareStatement(query);) {
31             statement.setInt(1, comment.getBookId());
32             statement.setInt(2, comment.getUserId());
33             statement.setString(3, comment.getComment());
34             statement.executeUpdate();
35         } catch (SQLException e) {
36             e.printStackTrace();
37         }
38     }
39
40     public List<Comment> getAll(int bookId) { 2 usages & Dragojevic, Uros
```

Sad napad nece uspeti I dodace se komentar kako smo I zeleti:

Book Details

localhost:8080/books/1

Set in Middle-earth, the story began as a sequel to Tolkien's 1937 children's book The Hobbit, but eventually developed into a much larger work.

Author: J.R.R. Tolkien

Genres:

- epic fantasy
- sci-fi

Overall rating: 4.6666665

My rating: 5

1 2 3 4 5 Rate

Book comments

Bruce Wayne

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

Tom Riddle

"); insert into persons(firstName, lastName, email) values ('dragana', 'katic', 'gaga@gmail.com')!!--

Add comment

Comment...

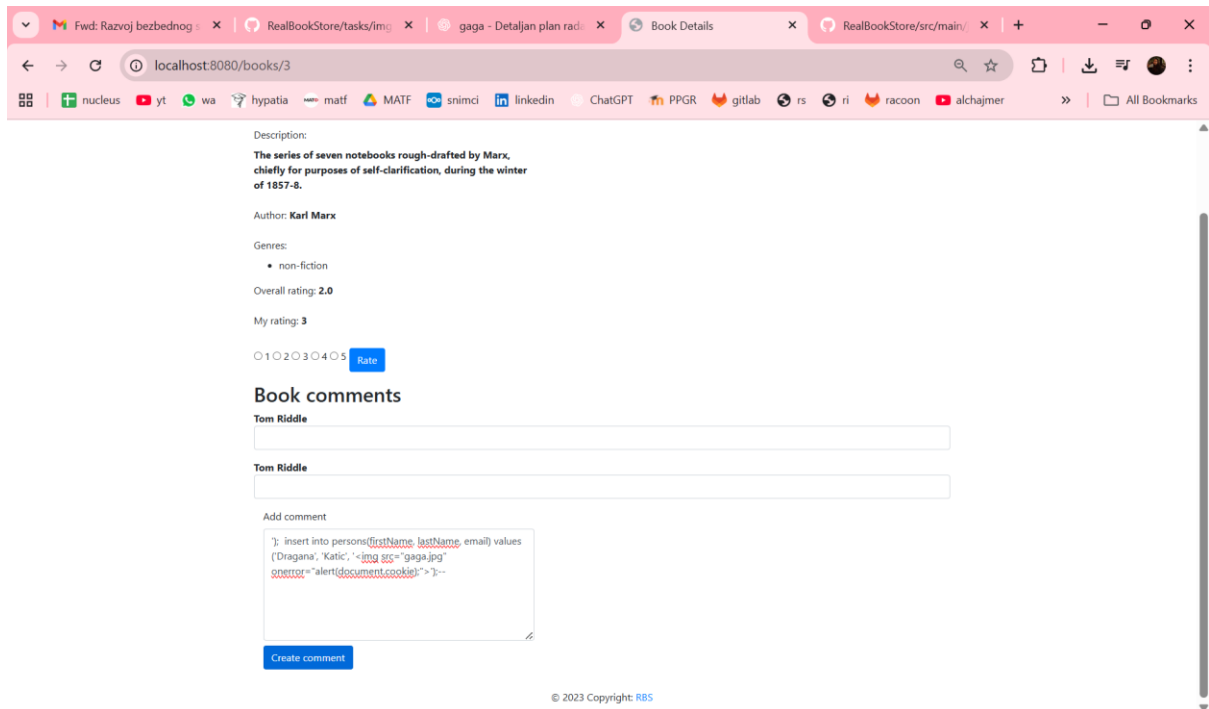
Create comment

© 2023 Copyright: RBS

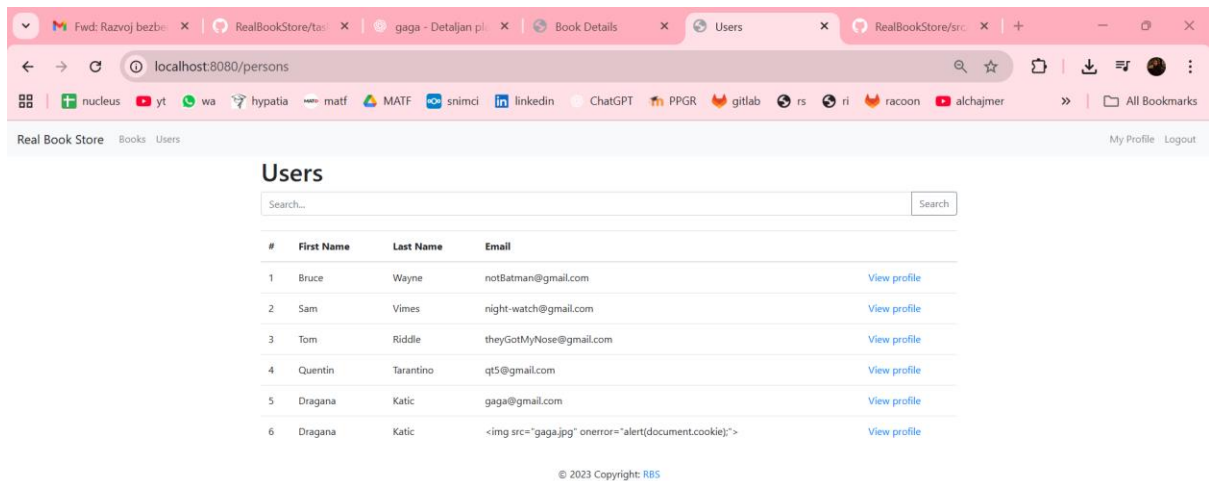
2.2. Cross-site scripting

Napad: Ubacivanje novog usera u tabelu “persons”

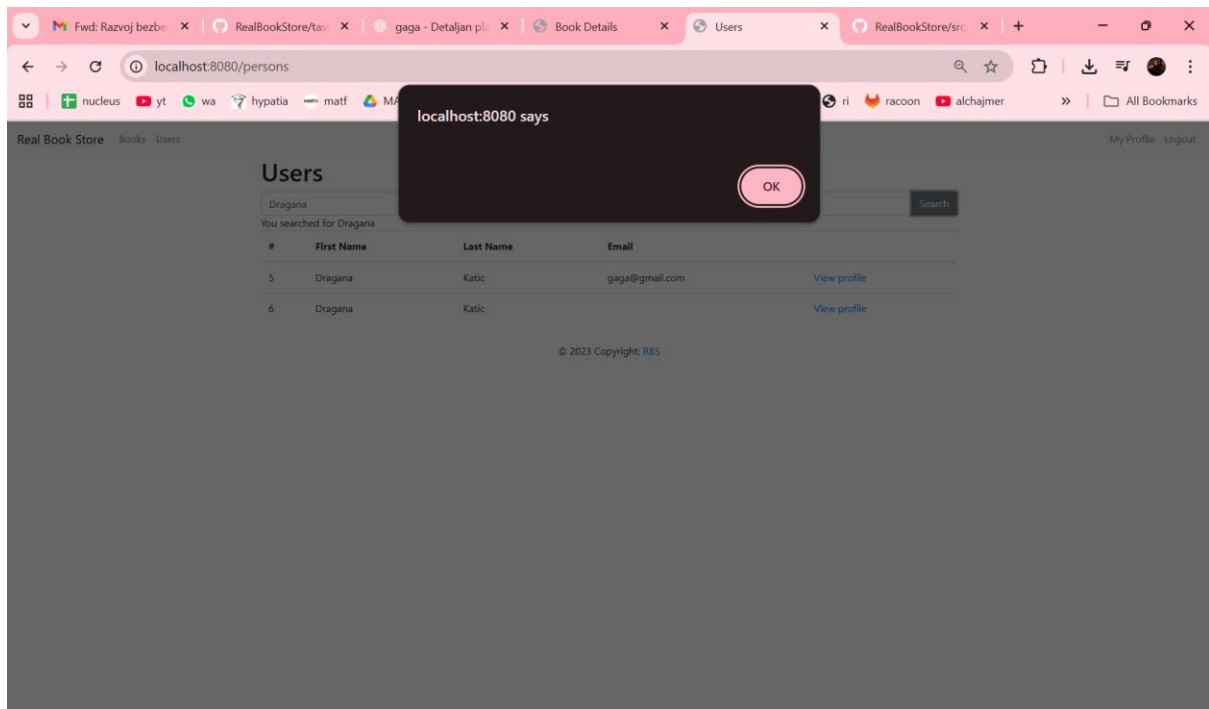
Metod napada:



Po sledecoj slici vidimo da je napad I uspeo:



I kada pokusamo da pretrazimo korisnika sa ovim firstNameom izlazi nam:



Predlog odbrane:

Pored PreparedStatement potrebno je da izmenimo i sledeci kod:

```
1 <html lang="en" xmlns:th="http://www.thymeleaf.org" xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout" >
11 <body>
12 <div layout:fragment="content">
44 <script>
53     tableContent.innerHTML = '';
54
55     persons.forEach(function(person) {
56         const tableRowElement = document.createElement("tr");
57         let tdElement = document.createElement("td");
58         tdElement.innerHTML = person.id;
59         tableRowElement.appendChild(tdElement);
60         tdElement = document.createElement("td");
61         tdElement.innerHTML = person.firstName;
62         tableRowElement.appendChild(tdElement);
63         tdElement = document.createElement("td");
64         tdElement.innerHTML = person.lastName;
65         tableRowElement.appendChild(tdElement);
66         tdElement = document.createElement("td");
67         tdElement.innerHTML = person.email;
68         tableRowElement.appendChild(tdElement);
69         tdElement = document.createElement("td");
70         tdElement.innerHTML = '<a href="/persons/' + person.id + '>View profile</a>';
71         tableRowElement.appendChild(tdElement);
72
73         tableContent.appendChild(tableRowElement);
74     });
75
76     document.getElementById('searchContainer').className = '';
77     document.getElementById('searchTerm').innerHTML = searchTerm;
```

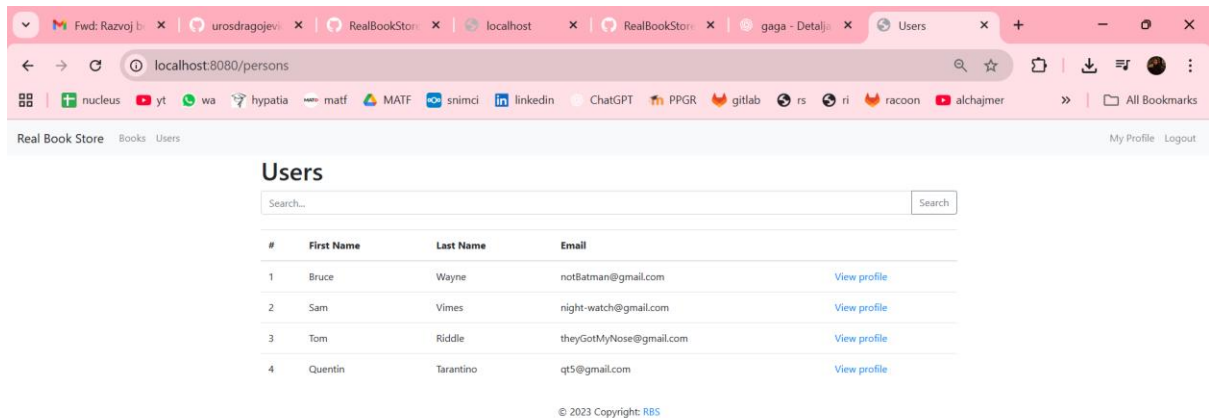
a sada da bude:

```
1 <html lang="en" xmlns:th="http://www.thymeleaf.org" xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout" >
11 <body>
12 <div layout:fragment="content">
43 <script>
44     window.addEventListener('load', function() {
45         let search = function () {
46             const searchTerm = document.getElementById("searchInput").value;
47
48             fetch('/persons/search?searchTerm=' + searchTerm)
49                 .then(function (result) {return result.json()})
50                 .then(function (persons) {
51                     const tableContent = document.getElementById("tableContent");
52                     tableContent.textContent = '';
53                     persons.forEach(function (person) {
54                         const tableRowElement = document.createElement("tr");
55                         let tdElement = document.createElement("td");
56                         tdElement.textContent = person.id;
57                         tableRowElement.appendChild(tdElement);
58                         tdElement = document.createElement("td");
59                         tdElement.textContent = person.firstName;
60                         tableRowElement.appendChild(tdElement);
61                         tdElement = document.createElement("td");
62                         tdElement.textContent = person.lastName;
63                         tableRowElement.appendChild(tdElement);
64                         tdElement = document.createElement("td");
65                         tdElement.textContent = person.email;
66                         tableRowElement.appendChild(tdElement);
67                         tdElement = document.createElement("td");
68                         tdElement.textContent = '<a href="/persons/' + person.id + '>View profile</a>';
69                         tableRowElement.appendChild(tdElement);
70
71                         tableContent.appendChild(tableRowElement);
72                     });
73
74                     document.getElementById('searchContainer').className = '';
75                     document.getElementById('searchTerm').textContent = searchTerm;
76                 });
77         };
78     });
```

Ako sada pokusamo pretragu, sve ce biti uredi tj nece izlaziti alert.

3. Cross-Site Request Forgery(CSRF)

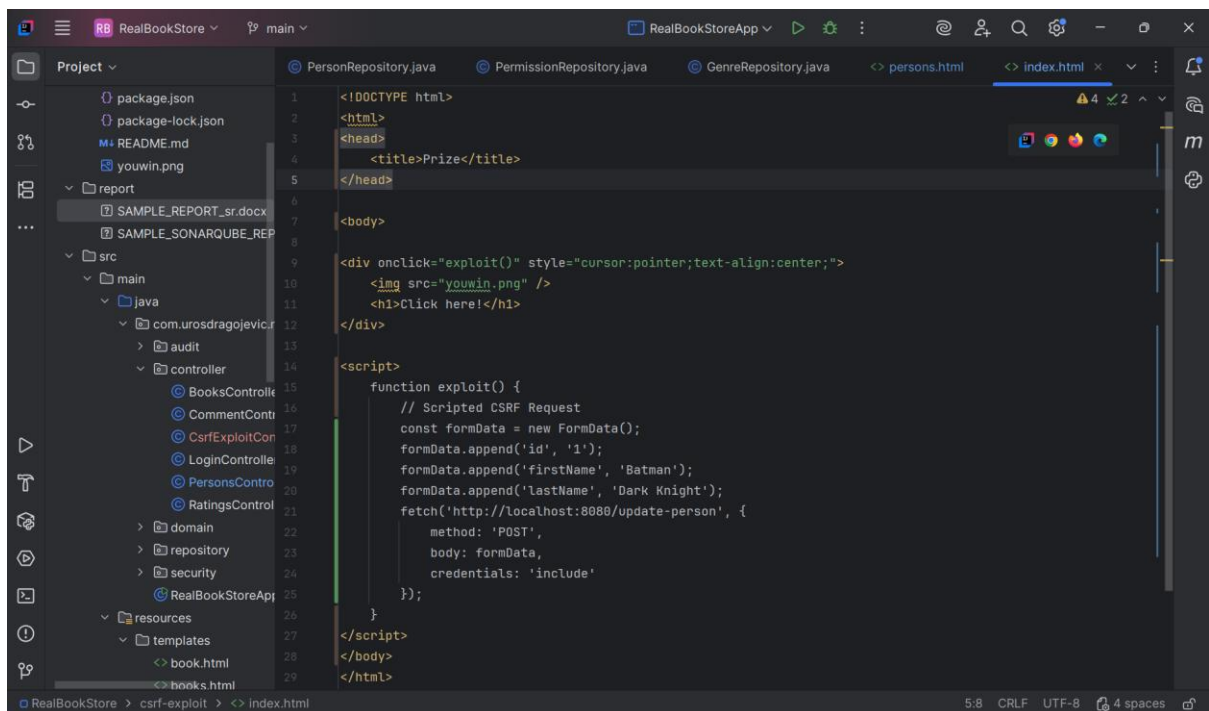
Pre napada tabela sa userima izgleda ovako:



#	First Name	Last Name	Email	
1	Bruce	Wayne	notBatman@gmail.com	View profile
2	Sam	Vimes	night-watch@gmail.com	View profile
3	Tom	Riddle	theyGotMyNose@gmail.com	View profile
4	Quentin	Tarantino	qt5@gmail.com	View profile

© 2023 Copyright: RB5

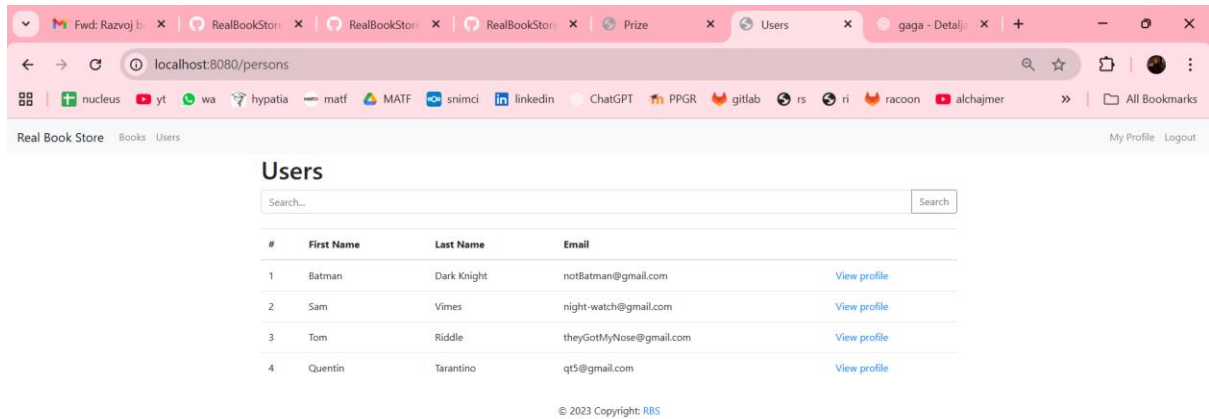
Nakon izmene koda u csrf-exploit/index.html:



```
<!DOCTYPE html>
<html>
<head>
<title>Prize</title>
</head>
<body>
<div onclick="exploit()" style="cursor:pointer;text-align:center;">

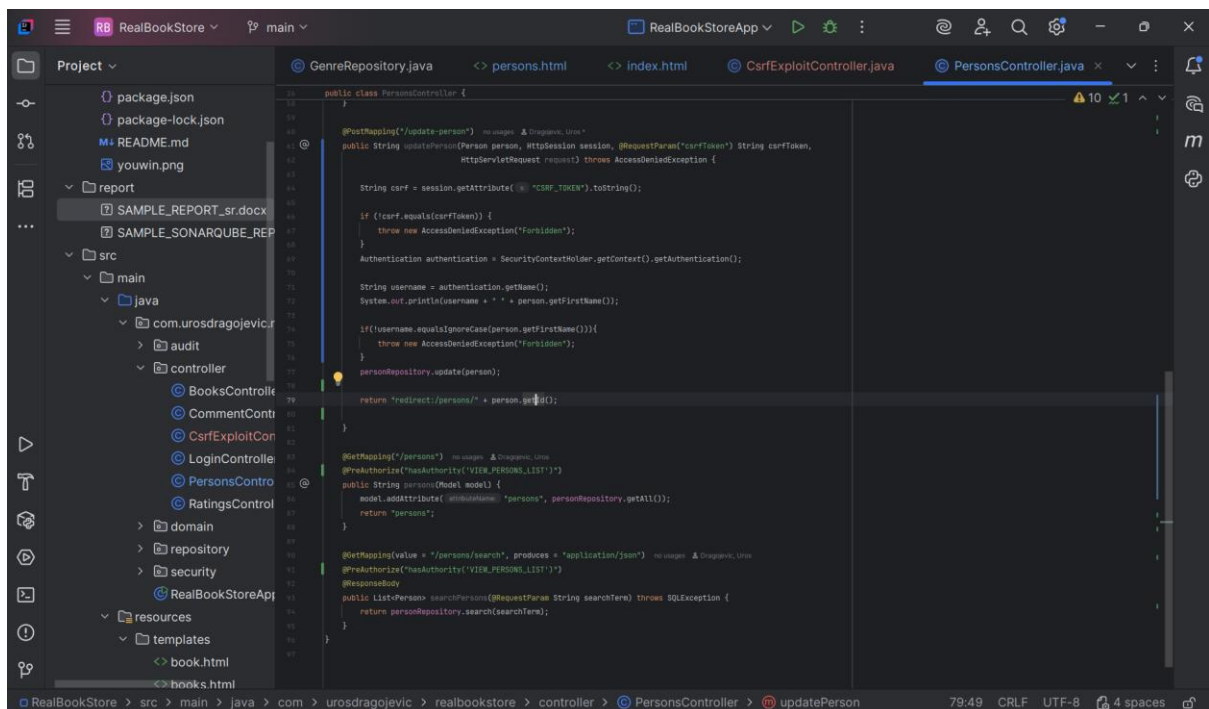
<h1>Click here!</h1>
</div>
<script>
function exploit() {
// Scripted CSRF Request
const formData = new FormData();
formData.append('id', '1');
formData.append('firstName', 'Batman');
formData.append('lastName', 'Dark Knight');
fetch('http://localhost:8080/update-person', {
method: 'POST',
body: formData,
credentials: 'include'
});
}
</script>
</body>
</html>
```

Ukoliko otvorimo ovaj html fajl i kliknemo na Click here!, napad se uspesno desio i promenio se prvi user, tj sada je postao Batman Dark Knight:

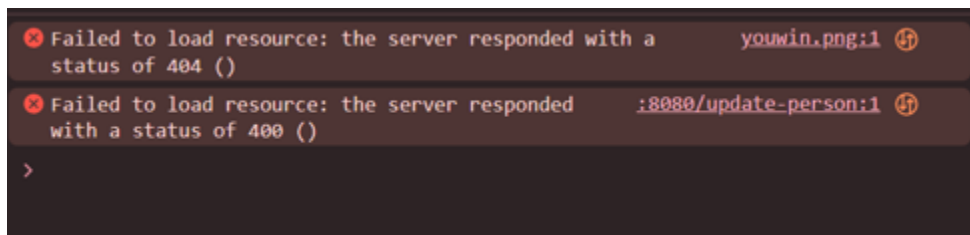


Predlog odbrane:

Odbranu postizemo koriscenjem CSRF tokena, tako sto u PersonController izmenimo:

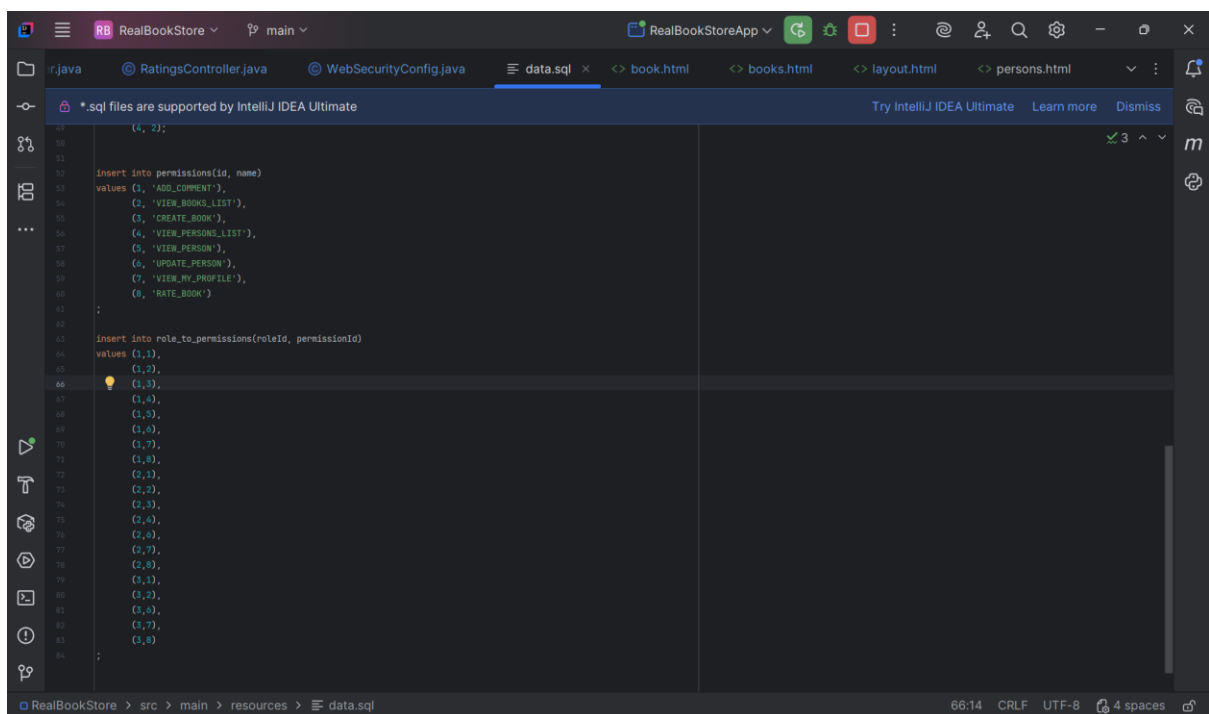


I sada ako ponovimo napad, videmo da ne može:



4. Autorizacija

Prvo implementiramo matricu permisija. U njoj svaka kolona je jedna rola a svaki red jedno ovlascenje.



Autorizacija se uradila pomocu anotacija `@PreAuthorize`, takodje u klasi `WrbSecurityConfig` je dodato `@EnableMethodSecurity`

```
RealBookStoreApp
main
WebSecurityConfig.java
data.sql
book.html
books.html
layout.html
persons.html
WebSecurityConfig.java
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configurers.AbstractHttpConfigurer;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.web.SecurityFilterChain;
import org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;

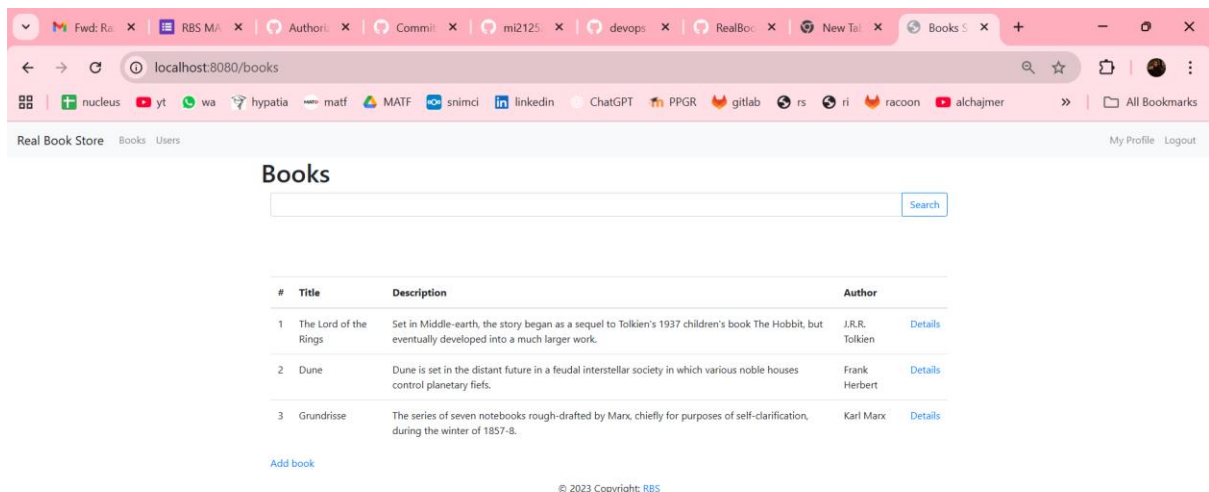
@Configuration
@EnableWebSecurity
@EnableMethodSecurity

public class WebSecurityConfig {

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
        http
            .csrf(AbstractHttpConfigurer::disable)
            .authorizeHttpRequests((AuthorizationManagerRequestMatcher requests) -> requests.anyRequest().authenticated())
            .formLogin((FormLoginConfigurer<HttpSecurity> form) -> form
                .loginPage("/login")
                .defaultSuccessUrl("/books", alwaysUse: true)
                .permitAll()
            )
            .logout((LogoutConfigurer<HttpSecurity> logout) -> logout
                .logoutSuccessUrl("/login")
                .invalidateHttpSession(true)
                .deleteCookies("JSESSIONID")
            )
        );
    }
}
```

Nakon svega ovog odradjenog primecujemo da se situacija menja u zavisnosti od toga sa kojim usernamemom se ulogujemo.

Kada se ulogujemo sa tom/guessmeifyoucan, koji ima premisije, primecujemo da korisnik ima opciju da doda nova knjigu:



Medjutim ukoliko se ulogujemo preko bruce/wayne primecujemo da ne mozemo dodati

novu knjigu jer taj korisnik nema tu premisiju:

localhost:8080/books

nucleus yt wa hypatia matf MATF snimci linkedin ChatGPT PPGR gitlab rs ri racoon alchajmer

Real Book StoreUsersMy ProfileLogout

Books

Search

#	Title	Description	Author
1	The Lord of the Rings	Set in Middle-earth, the story began as a sequel to Tolkien's 1937 children's book The Hobbit, but eventually developed into a much larger work.	J.R.R. TolkienDetails
2	Dune	Dune is set in the distant future in a feudal interstellar society in which various noble houses control planetary fiefs.	Frank HerbertDetails
3	Grundrisse	The series of seven notebooks rough-drafted by Marx, chiefly for purposes of self-clarification, during the winter of 1857-8.	Karl MarxDetails

© 2023 Copyright: RBS