Stručni kurs Razvoj bezbednog softvera  
  
Izveštaj

**Pronađene ranjivosti u projektu “RealBookStore”**

Marko Lazarević

27.2.2025.

# Istorija izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Verzija** | **Datum** | **Izmenio/la** | **Komentar** |
| **1.0** | 27.2.2025. | Marko Lazarević | Kreiran izveštaj |
| **1.1.** | 27.2.2025. | Marko Lazarević | SQL Injection |
| **1.2.** | 27.2.2025. | Marko Lazarević | XSS |
| **1.3.** | 28.2.2025. | Marko Lazarević | CSRF |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

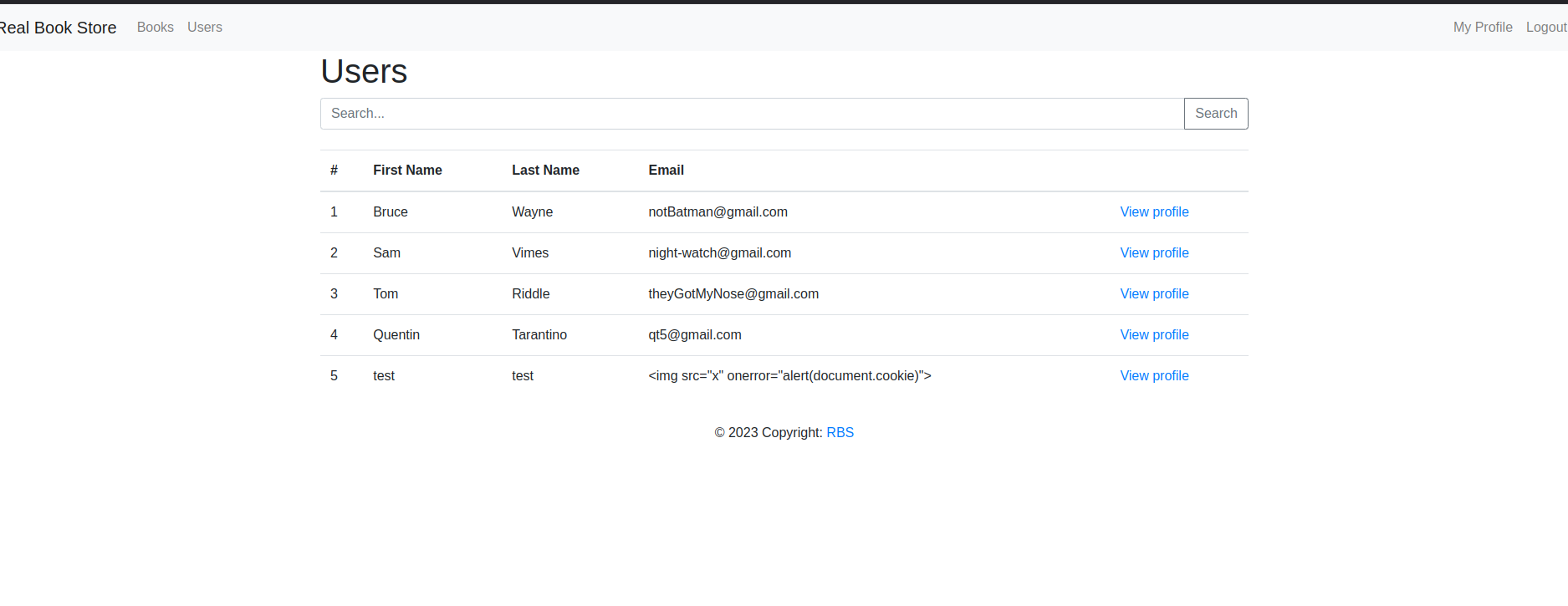
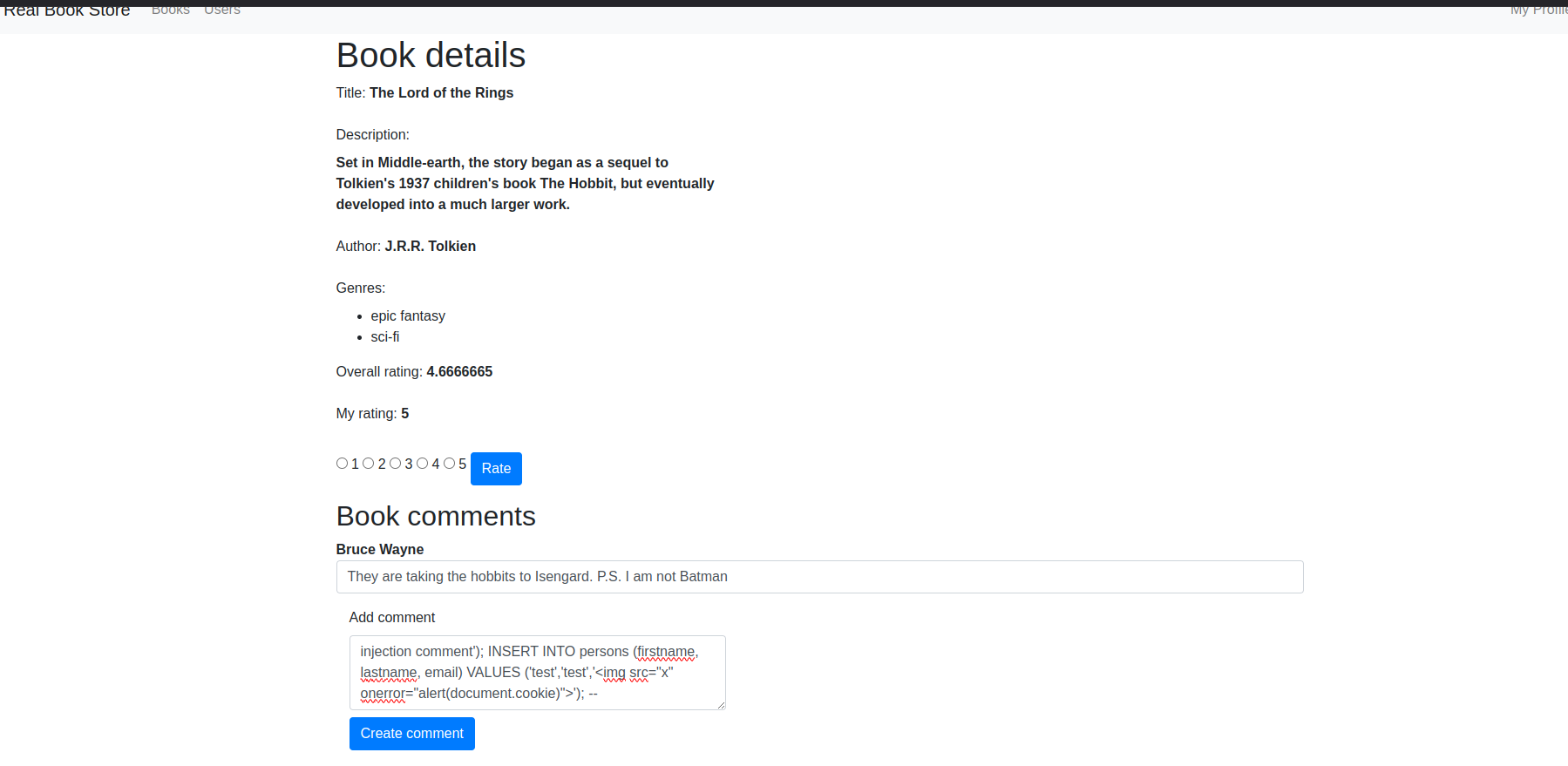
*Pronadjene su ranjivosti za SQL injection i za XSS napade gde korisnici kroz unos komentara mogu direktno da pišu SQL upite i menjaju stanje baze podataka.*

# SQL injection

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

### Metod napada:

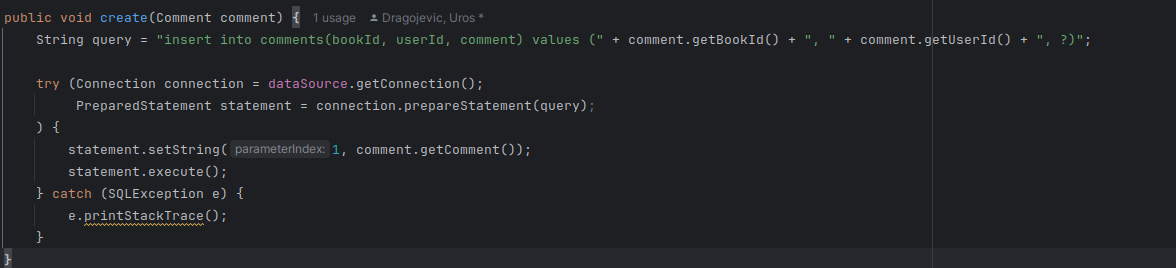
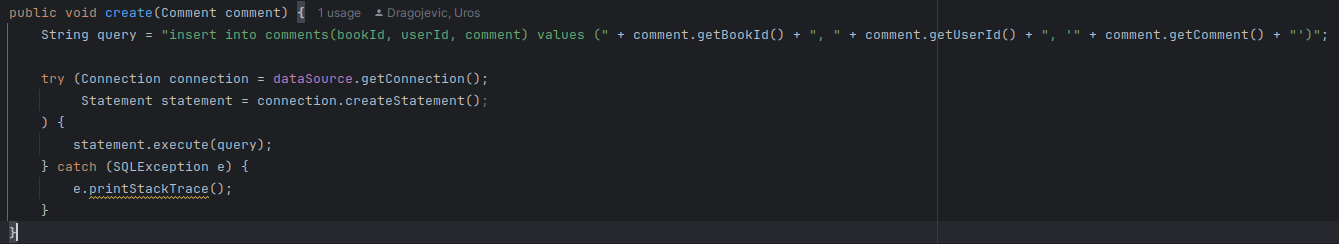
Na stranici /books/1 (ili bilo koji drugi /books/{id}) aplikacije, uneti sledeći kod u input polje “Add comment”:

*injection comment'); INSERT INTO persons (firstname, lastname, email) VALUES ('test','test','<img src="x" onerror="alert(document.cookie)">'); --*

## 

## Predlog odbrane:

Implementirati čuvanje komentara koristeći PreparedStatement umesto Statement   
  
Umesto koda:

  
Koristiti kod:

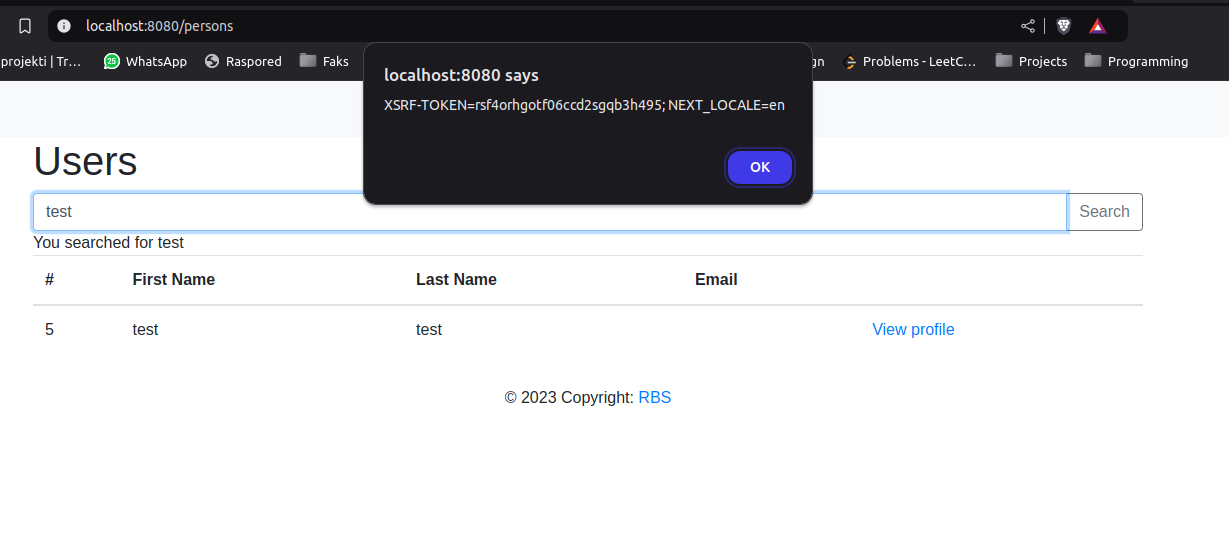
Nakon izmene nije moguće izvršiti SQL Injection.

# Cross-site scripting

## Napad: Prikazivanje kolačića korisnika

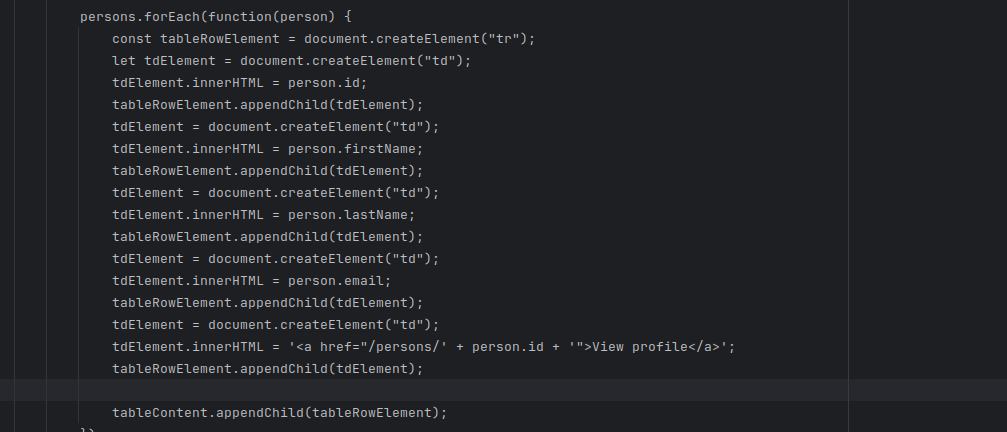
### Metod napada:

Na stranici Persons aplikacije, unet je sledeći kod u input polje “Email” kroz SQL injection napad:  
  
*'<img src="x" onerror="alert(document.cookie)">'*

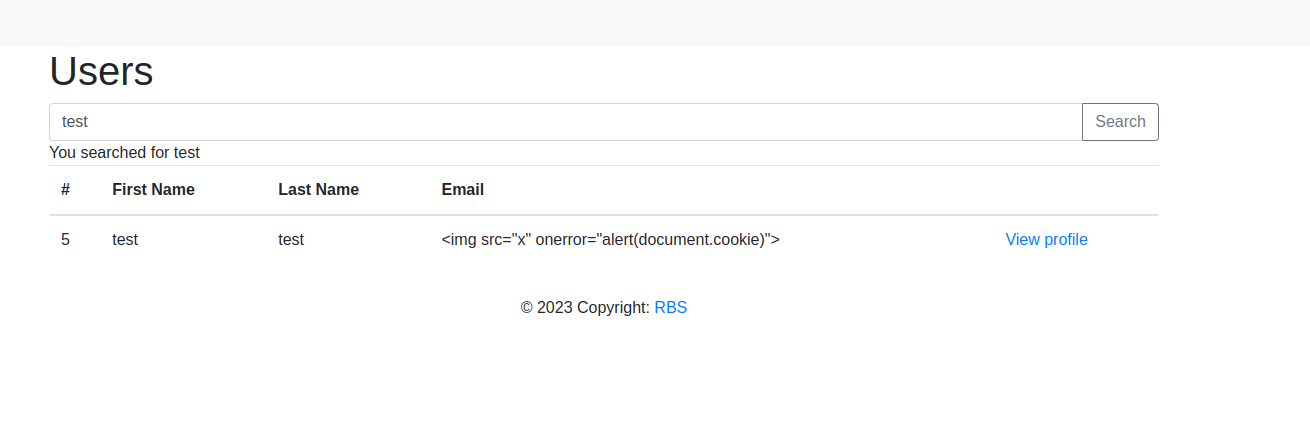
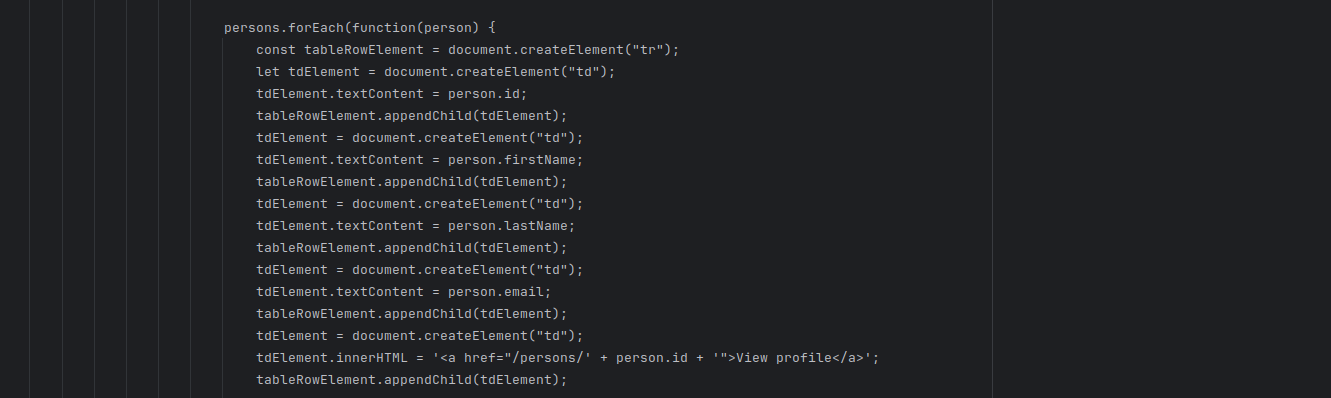
Nakon pretrage “test”:

## Predlog odbrane:

Kod pretrage umesto innerHtml koristiti textContent za popunjavanje vrednosti u poljima  
  
Umesto koda:



Koristiti kod:

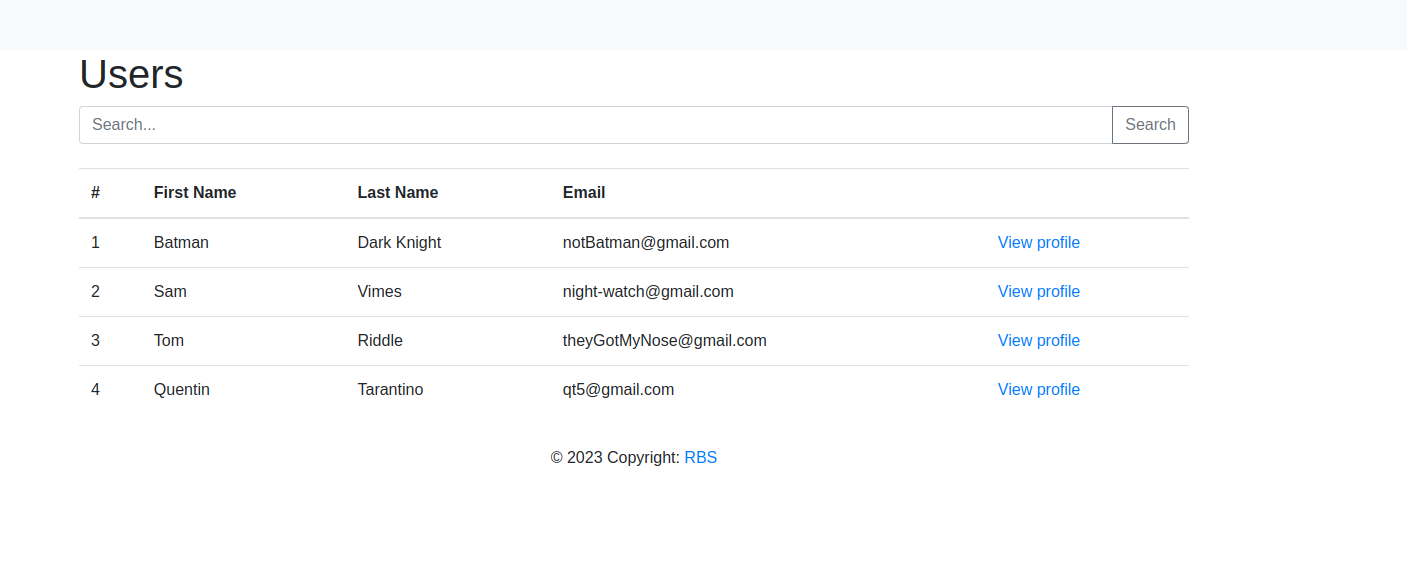
  
Rezultat pretrage nakon izmene:

# Cross-site request forgery

## Napad: Promena podataka korisnika

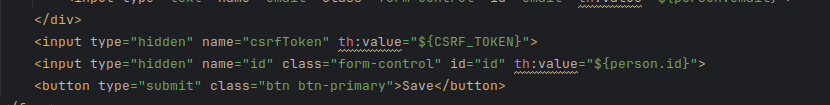
### Metod napada:

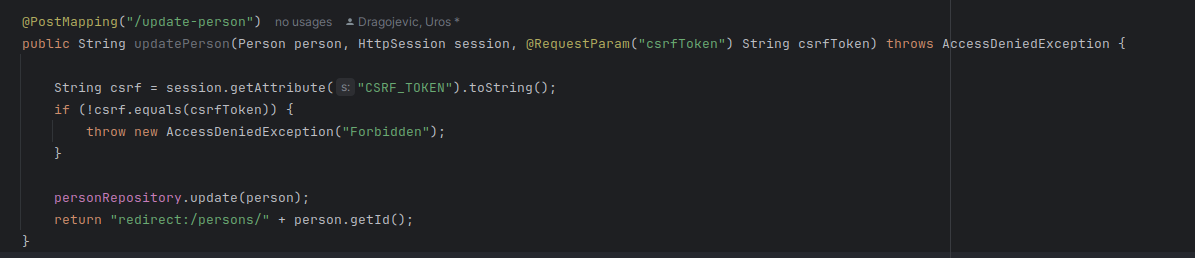
U malicioznoj aplikaciji klikom na pehar poslati su podaci na /update-person i izmenjeni su ime i prezime korisnika sa id 1.  
  
*function exploit() {  
 const formData = new FormData();  
  
 formData.append('id', 1);  
 formData.append('firstName', 'Batman');  
 formData.append('lastName', 'Dark Knight');  
 fetch('http://localhost:8080/update-person',  
 {method: 'POST', body: formData, credentials: 'include'});  
  
}*

Rezultat napada:

### Predlog odbrane :

### Implementiranje CSRF zaštite korišćenjem tokena. Kada se kreira forma dodaje se skriveno polje csrfToken

Kada se šalje zahtev za izmenom proverava se token i ukoliko je validan tada se menjaju podaci:



# Zaključak

Napadi su mogli biti izvršeni zbog propusta u kodu koji se mogu lako izbeći.   
Bilo bi poželjno prekontrolisati ostatak projekta i proveriti da li ima sličnih propusta.