Cross-site request forgery (CSRF) napad

Opis:

Aplikacija nema zaštitu protiv CSRF napada. Neophodno je demonstrirati napad na sledeći način: koristeći aplikaciju koja simulira napad (folder csrf-exploit unutar projekta) treba napraviti skriptu tj. poziv ka endpointu /update-person unutar PersonsController.java klase. Napad treba da promeni lične podatke korisnika sa ID = 1, tako da je fisrtname = “Dobby” i lastName = “Free Elf”.

Neophodno je implementirati zaštitu od CSRF napada korišćenjem tokena. Odbranu je potrebno primeniti samo na gorenavedeni endpoint. Popravka ranjivosti ne sme da naruši ili promeni funkcionalnost aplikacije.

Rešenje:

Naša aplikacija koristi koncept sesije i kreira JSESSIONID kolačić. Možemo videti da naša aplikacija dozvoljava pomoću CORS filtera pristup serveru domenu <http://localhost:3000> na kom se nalazi naša aplikacija koja simulira CSRF napad. Naš napad je vrlo jednostavan, potrebno je kreirati POST zahtev ka endpointu <http://localhost:8080/update-person> koji u sebi sadrži parametre koji su od nas zahtevani uz par dodatnih atributa a to su credentials: “include” koji zahteva Access-Control-Allow-Credentials header koji naš server ima kako bi se prosledili kredencijali korisnika, kao i mode: “no-cors” koji dozvoljava samo bezbedne CORS zahteve kako se naš pretraživač ne bi bunio.

A screen shot of a computer program

Description automatically generated

Referenca za fetch API u JavaScript: <https://javascript.info/fetch-api>

Zaštitu od CSRF napada implementiramo koristeći tokene, odnosno Synchronizer Token Pattern. Token je jedinstven za sesiju korisnika i generiše se kao nasumična vrednost koja se šalje kao skriveno polje forme pri svakom zahtevu i onda se na serveru proverava da li je taj token jednak tokenu koji smo sačuvali za datu sesiju datog korisnika.

U endpointima /persons/{id} i /myprofile čitamo token iz sesije i upisujemo ga u model koji ćemo renderovati na frontendu.

A screen shot of a computer program

Description automatically generated

U endpointu /update-person dodajemo proveru da li je token koji nam je poslat u formi jednak tokenu koji smo sačuvali za datu sesiju datog korisnika i ako nije vraćamo HTTP 403 Forbidden:

A screen shot of a computer program

Description automatically generated

Još nam ostaje samo da na front-endu u okviru person.html stranice dodamo prosleđivanje tokena iz modela pri pozivu /update-person-endpointa:

A screenshot of a computer program

Description automatically generated

Aleksandar Ivanović, 2023/3468