Izveštaj

OWASP TOP 10

1. **Injection**
   SQL Injection je napad koji je preventovan pomoću PreparedStatement metode. Ovim je onemogućen bilo kako spoljni uticaj na bazu. Šta god da se prosledi kroz input polja ili putem softvera kao što je Postman to će biti upisano kao polje u koloni u odgovarajućem redu u bazi, nikad se neće izvršiti nad bazom. Ova metoda može biti ranjiva samo u slučaju ako se update izvši samo na osnovu stringa prepeared statement-a. U ovom sistemu, vrednosti su se dodeljivale redovima putem metoda setString, setLong,..koje su bezbedne za SQL Injection napad.

2. **Broken Authentication**
   Korisnicima su dodeljene uloge i permisije pomoću koji je određeno kojim delovima sistema mogu da pristupe i do kog nivoa. Permisije korisnika mogu biti omogućene i uklonjene samo za njega i one su proveravane svaki put kada korisnik pokuša bilo šta da izvrši u aplikaciji.

   Implementirana je provera slabe šifre i korisnik ne može da unese šifru kraću od 10 karaktera, kao i onu koja nema bar jedno malo i veliki slovo, kao i broj. Takođe šifra je heširana kako niko ne bi mogao sem korisnika da zna njegovu šifru.

   Sesija je liminitana na 8 sati.

3. **Sensitive Data Exposure**
   Osetljivi podaci se ne čuvaju u bazi. Sve šifre su heširane i koristi se https.

4. **XML External Entities (XXE)**
   Korišćen je JSON format. Samo zahtevi koji su autorizovani dobijaju response nazad.

5. **Broken Access Control**
   ACL, RBAC - svaki korisnik ima permisije koje se proveravaju svaki put kada on izvrši neku akciju u aplikaciji. Akcije ne mogu biti izvršene tako što se promeni URL baš zato što će se svaki put proveriti permisija.

6. **Security Misconfiguration**
   ACL
   Uklonjene nepotrebne stranice, portovi, servisi.
   Poseduje aktivaciju akaunta putem mejla i ne postoje defaultne šifre.

7. **Cross-Site Scripting XSS**
   Ova vrsta napada preventovana je kako na frontendu tako i na bekendu upitom da li se u nekom od input polja nalaze tagovi < i > iz razloga što se ovaj napad vrši putem script taga koji u sebi može da nosi virus ili nešto ranjivo po sistem.

8. **Insecure Deserialization**

9. **Using Components with Known Vulnerabilities**
   Uklonjeni su svi dependency-ji koji se ne koristi, kao i fajlovi. Verzije svih dependency-ja su najnovije. Takođe, komponente koje su korišćene su one sa oficijalnih sajtova.

10. **Insufficient Logging & Monitoring**
    Logovi su raspoređeni u fajlove za info, error i warn logove radi bolje preglednosti. Svaki log pored teksta ko je i šta uradio sadrži vreme i datum kako bi onaj ko treba da pogleda u logove znao šta se u sistemu desilo samo iz njihovog čitanja. Veličina log fajlova ne može biti iznad 30Kb i ako se pređe, pravi se novi log fajl.

DEPENDENCY VULNERABILITIES

Backend

Spring Boot Security, Spring Boot Starter Data REST, Netflix Eureka Client
Cross-site scripting (XSS) ranjivost rešena je pomoću validacija da li bilo koji unos u sistem sadrži html tagove. Update-ovani dependency-ji na najnovije verzije.

Frontend

Pre svega, pri svakom instiranju komandna linija bi ispisivala da li nešto od prethodno instaliranih paketa ima ranjivosti i zatim bi ponudila da se te ranjivosti uklone izvršavanjem komande npm audit fix.

Axios

DoS odnosno Denial of Service je ranjivost koji je axios paket sadržao do verzije 0.18.1. Trenutna verzija je vulnerability free, a u našoj aplikaciji je verzija 0.19.2. Nastajao je napad jer i dalje posle, dostignutog maxContentLenght, sadržaj je mogao da se prihvati. Proof of concept: U svaki axios poziv, bio trebao da salje i maxContentLenght.

Bootstrap

Takođe je imao u verzijama do 4.3.1 XSS ranjivost. Verzija prisutna u našem projektu je 4.5.0 za koju trenutno nema prijavljenih ranjivosti.