OWASP TOP 10 IZVEŠTAJ

1. Injection

   SQL injection

   Odnosi se na radnjivost pri kojoj napadac moze da inject-uje SQL izraz i time sebi obezbedi pristup vrednim podacima koje moze eksploatisati.
   Ova vrsta napada se moze spreciti kroz pravilnu upotrebu parametrizovanih upita za pristup bazi. Za sprecavanje ove vrste napada koristili smo JPA za rad sa bazom.
   Iako JPA obezbeđuje prilično dobru zaštitu od SQL napada, mi aplikaciju branimo i proverom upisane vrednosti u polje sa definisanim regex šablonom (poput znakova ` i =) na frontu i na back-u.

2. Broken authentication

   Odnosi se na rizik da maliciozni korisnik dobije pristup nekom resursu, odnosno da preuzme identitet i tako dođe do vrednih podataka.
   Napravili smo blacklist-u sa čestim lozinkama radi blokiranja korisnika da prilikom registracije unese neku šifru sa te liste i time bude podložniji Password spraying napadu.
   Za svako polje se vrši validiranje unosa preko regex izraza, kao i za lozinku koja mora ispunjavati određene kriterijume da bi bila što jača.
   Lozinke se štite hash & salt mehanizom za heširanje lozinki preko BCrypt algoritma.
   Sesija korisnika se uništava logout metodom, a ukoliko korisnik zaboravi da se izloguje – nameštено je da jwt token traje neko vreme nakon čeka se sesija prekida.
   Rad aplikacije se odvija preko HTTPS protokola.

3. Sensitive Data Exposure

   Komuniakcija između browser-a i servera se odvija preko TLS/SSL. Prenos podataka u plain text-u zaštićen HTTPS protokolom.
   Za kriptovanje podataka je korišćen RSA asimetrični algoritam sa dužinom ključa 2048 bita.
   Lozinka se štiti, kako je i gore navedeno, preko hash & salt mehanizma koji obezbeđuje da se na unetu lozinku korisnika doda random string, a zatim se takav string hash-uje – pomoću BCrypt algoritma.
   Ključevi se čuvaju u posebnim keystore-ovima (rootKeyStore, intermediateKeyStore i endEntityKeyStore)
   Ukoliko dođe do greške, korisnik se obaveštava sa samo potrebim podacima da ne bi napadač mogao otkriti tačne nedostatke aplikacije.

4. XML External Entities

   XXE napad se može rešiti pravljenjem šeme radi validacije podataka. U ovom projektu se ne koristi XML kao format za slanje podataka već JSON.

5. Broken access control

   Implementiran je RBAC model tako da svakom ulogovanom korisniku je dodeljena rola koja ima svoje permisije i na osnovu tih permisija kontrolišemo pristup.

6. Security Misconfiguration

   Loše sigurnosne postavke mogu dovesti do ugrožavanje celog sistema i zbog toga je na front-u urađena provera svih ranjivosti pomoću komande *npm audit*. Na back-u je korišćen Dependency-check plugin za otkrivanje ranjivosti.

7. Cross-site scripting

   U ovoj vrsti napada se prisiljava web stranica da izvrši unete malicionze skripte. Takav napad na veb čitač je rešen pomoću mustache tag-ova koje omogućava Angular tako što escape-uje specijalne karaktere pre obrade od strane browsera.

8. Insecure Deserialzation

   Prilikom čitanja pristiglih podataka potrebno je izvršiti deserijalizaciju odnosno pretvoriti u prvobitan oblik kako bi se izbegla ranjivost.

9. Using components with known vulnerabilities

   Za analizu ranjivosti korišćen je Dependency-check plugin.

10. Insufficient Logging & Monitoring

    Pomoću logova se beleže događaji u sistemu što svakako može pomoći u ranijem otkrivanju ranjivosti i izbegavanjem maliciozog napada.