OWASP - Top 10 Web Application Security Risks

1. **Injection**
   Injection napad se dešava kada napadač pošalje podatke takve da sadrže upit bazi koji bi ugrozili integritet i sigurnost baze podataka, pa samim tim i aplikacije. U našoj aplikaciji korišćene su PostgreSQL i MongoDB baza, s tim da se korisnici čuvaju u Keycloak-u. Da bi se zaštitili od Injection napada u serverskoj aplikaciji smo svaku DTO klasu validirali koristeći Java constraints anotacije, dok su na klijentskoj korišćeni Validatori, s tim da je najčešće korišćena pattern validacija. Time smo zabranjivali unos određenih karaktera koji bi mogli narušiti bezbednost.

2. **Broken Authentication**
   Za rad sa korisnicima koristili smo Keycloak, koji nudi podršku za autentikaciju korišćenjem JWT tokena koji smo čuvali u session storage-u. Pri kreiranju korisnika Keycloak zahteva promenu lozinke uz određene uslove za novu lozinku. Osim toga, nudi dobru podršku za brute force napade, te ako se desi da korisnik više od 10 puta zaredom ne uspe da se uloguje s ciljem da izazove napad, njegov nalog će biti blokiran.

3. **Sensitive Data Exposure**
   Podaci vezani za korisnika čuvani su u Keycloak-u, koji sam po sebi nudi podršku za heširanje lozinki, koje nisu vidljive administratoru. Ostali podaci koji su čuvani u PostgreSQL šifrovani su korišćenjem AES algoritma za šifrovanje, a pri njihovom dobavljanju bi se istim algoritmom dešifrovali. Isto tako, pri slanju verifikacionog maila email korisnika bi se šifrovao uz korišćenje salt-a.

4. **XML External Entities(XXE)**
   U aplikaciji ne postoji slanje XML dokumenata. Samim tim ova zaštita nije odrađena jer smo smatrali da nije potrebna.

5. **Broken Access Control**

   Kontrola pristupa je omogućena i na klijentskoj i na serverskoj strani aplikacije. Na serverskoj strani realizovana je pomoću rola korisnika i permisija na Keycloak-u, dok je na klijentskoj strani realizovana putem guard-ova.

6. **Security Misconfiguration**

   Što se tiče podešavanja konfiguracije bezbednosti, konfigurisan je https protokol za bezbednu komunikaciju. To uključuje komunikaciju između dva servera, komunikaciju server-klijent i komunikaciju sa Keycloak-om. Osim toga, konfigurisana su i prava pristupa određenim resursima u zavisnosti od uloge korisnika u sistemu.

7. **Cross-Site Scripting XSS**

   XSS napadi se dešavaju kada napadač pošalje podatke koji se mogu izvršiti kao skripte u browseru. U aplikaciji smo koristili Angular radni okvir, koji ima svoju ugrađenu zaštitu od XSS-a(proverava za svaki input da li sadrži neki tag kao script koji bi se mogao izvršiti, i takve prepoznaje kao loše). Osim toga, rađena je i ručna provera koristeći validatore i patterne.

8. **Insecure Deserialization**

   Svaki objekat koji dolazi na server pre svega mora proći kroz već spomenutu validaciju, što je prvi sloj odbrane od “loših” podataka. Ako se validacija prođe, obezbeđena je još jedna provera pri upisivanju u bazu podataka(try catch blok ili provera null vrednosti nakon pokušaja čuvanja), kojom se garantuje da loši podaci nikada neće biti upisani u bazu.

9. **Using Components with Known Vulnerabilities**

   U aplikaciji nismo koristili komponente koje bi mogle ugroziti bezbednost.

10. **Insufficient Logging and Monitoring**

    Tokom celog rada aplikacije vrši se logovanje. Svi logovi(što ukljucuje logove aplikacije, keycloak-a i simulatora logova) se čuvaju u log fajlu i u MongoDB bazi podataka. Logovi prolaze i kroz drools pravila, te će okinuti odgovarajući alarm ako je to potrebno. Praćenje stanja logova omogućeno je adminu bolnice.