Top 10 Web Application Security Risks

BSEP – Tim 8

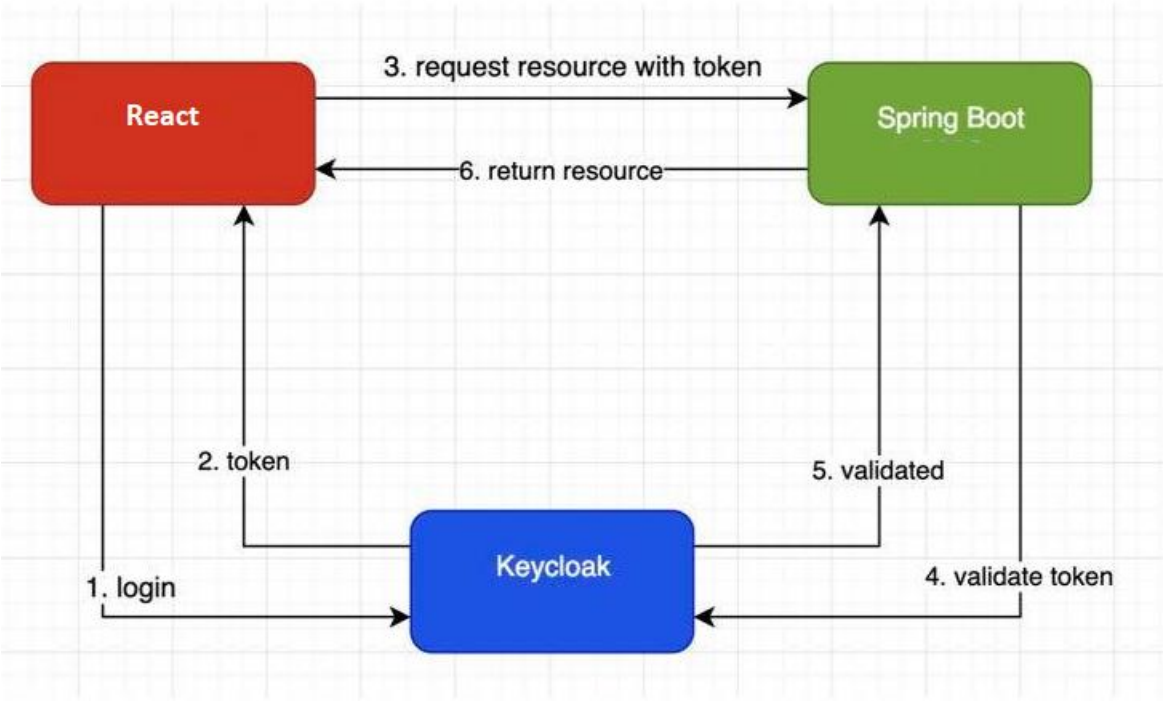
Milena Laketić SW15/2017, Mitar Perović SW71/2017, Vera Kovačević SW19/2017

1. Injection

Injection napad se dešava kada korisnik (napadač) pošalje podatke koji mogu da ugroze sigurnost aplikacije preko parametara upita u bazi. U našoj aplikaciji korištena je *Postgres* relaciona baza i *MongoDB* NoSQL baza. Pored toga što baze koriste parametrizovane upite koji omogućuju da se korisnički unosi tretiraju isključivo kao podaci, odrađena je i „*whitelist*“ validacija podataka na serveru pomoću *Spring Boot* anotacija, čime je zabranjen unos specijalnih karaktera (npr. „;“) na mjestima gdje ne bi trebalo da budu (npr. email, id pacijenta itd.)

2. Broken Authentication

Za rad sa korisnicima korišten je *Keycloak* server, koji nudi podršku za autentifikaciju korišćenjem JWT tokena. Na slici je prikazan način na koji *Keycloak* šalje i validira token.



Pri dodavanju novog korisnika omogućena je promjena lozinke, pri čemu je obavezno da lozinka ispunjava određene uslove kako bi bila šo bezbjednija: minimalna dužina lozinke je 8 karaktera, lozinka mora da sadrži bar jedno veliko slovo i bar jednu cifru. Takođe, koristi se i „*blacklist*“ sa 10000 najčešće korištenih lozinki, koje su zabranjene.

Keycloak daje mogućnost detekcije *Brute Force* napada, pa je omogućeno i da se blokira

korisnik koji 30 puta zaredom bezuspješno pokuša da se uloguje. Sama aplikacija takođe prati *Brute Force* napade pomoću logova, i na osnovu toga obavještava admina ukoliko se desi napad.

3. Senstive Data Exposure

Osjetljivi podaci koji se koriste u aplikaciji su lozinke korisnika i lične informacije o pacijentima.

Keycloak server vodi računa o čuvanju lozinki korisnika, hešira ih i ne dozvoljava adminu da vidi lozinke ostalih korisnika.

Lični podaci pacijenta (JMBG, ime, prezime), pri upisu u bazu se enkriptuju pomoću AES algoritma sa tajnim ključem servera, a pri čitanju se istim algoritmom dekriptuju. Korištena je klasa *AttributeEncryptor* koja implementira *AttributeConverter*

Poruke koje uređaji šalju bolnici sadrže samo id pacijenta i numeričke vrijednosti koje opisuju njegovo zdravstveno stanje, te se ne smatraju osjetljivim podacima i nisu šifrovane.

4. XML External Entities (XEE)

U aplikaciji ne postoji slanje XML dokumenata, te nije odrađena zaštita od ove vrste napada.

5. Broken Access Control

Kontrola pristupa je omogućena pomoću uloga u *Spring Boot*-u i permisija na *Keycloak*-u, što znači da korisnik može da pristupi samo onim resursima i metodama koji su dozvoljeni za njegovu ulogu i za koje ima odgovarajuću permisiju. U aplikaciji takođe postoji i Single Sign On, koji omogućava superadminu da pristupi bolničkoj aplikaciji, pa je superadminu dozvoljeno i da pregelda pacijente, logove i da šalje zahtjev za kreiranje sertifikata sa bolničke aplikacije.

6. Security Misconfiguration

Što se tiče podešavanja konfiguracije aplikacije, ono što je odrađeno je da se sva komunikacija (komunikacija između dvije serverske *Spring Boot* aplikacije, između servera i klijenta, između servera i python skripte koja predstavlja uređaj, kao i komunikacija svih ostalih aplikacija sa *Keycloak* serverom) odvija preko HTTPS protokola uz korištenje inicijalnih ili generisanih sertifikata i privatnih i javnih ključeva.

7. Cross-Site Scripting XSS

XSS napadi se dešavaju kada napadač pošalje podatke koji će se kasnije izvršiti kao skripte u browser-u korisnika. Aplikacija koristi *React* framework koji je zaštićen od XSS napada, promjenljive se automatski escape-uju, osim ako se koristi funkcija *dangerouslySetInnerHTML*.

Takođe su implementirani i XSS filteri na bekendu, koji provjeravaju sadržaj HTTP zahtjeva i uklanjaju sve što bi moglo da se izvrši u browser-u. Dodat je i *Content Security Policy* (CSP) u web security konfiguraciji, koji ublažava XSS i *Injection* napade.

8. Insecure Deserialization

Podaci koji se šalju kao java objekti (CSR koji se šalje sa bolničke aplikacije na admin aplikaciju, nova konfiguracija koja se šalje sa admin aplikacije na bolničku aplikaciju, poruka uređaja koja se šalje na bolničku aplikaciju) su digitalno potpisane, kako bi se utvrdio njihov integritet. Klase čiji objekti se šalju su klase modela ili DTO klase anotirane odgovarajućim validacijama, što sprečava mogućnost slanja nebezbijednih podataka.

9. Using Components with Known Vulnerabilities

U aplikaciji nisu korištene komponente za koje je poznato da imaju nedostatke koji bi mogli ugroziti sigurnost aplikacije.

10. Insufficient Logging & Monitoring

Obezbijeđeno je logovanje važnih aktivnosti u aplikaciji, kao što su uspješno i neuspješno logovanje (preko *Keycloak*-a), pojava greške u aplikaciji, neuspješan HTTP zahtjev, itd. Minimalan „*severity*“ loga podešen je na TRACE, što znači da će se logovati svi važni događaji, a ne samo pojava greške. Svi logovi aplikacije i *Keycloak*-a se čuvaju u bazi i okidaju alarme koji se prikazuju administratoru aplikacije, što omogućava praćenje aktivnosti u sistemu.