OWASP TOP 10 izveštaj – Tim 23

A1 – Injection

Injection napad predstavlja familiju napada koje se ispoljavaju kada se interpreteru proslede nevalidirani podaci pomoću kojih se izvršavaju neautorizovane instrukcije i uz pomoć kojih se pristupa neautorizovanim podacima. Injection napadi obuhvataju SQL, OS, XXE, XML entity i dr. napade.

Najčešći napad, Sql Injection, se eksploatiše direktnim prosleđivanjem ulaznih parametara u SQL upite, pri čemu ukoliko se koristi konkatenacija stringova za upit napadač može izazvati štetu poput uzimanja kredenijala admina, brisanja baze itd.

Ranjivosti detektovane u sistemu(●) i njihova zaštita(○):

- SQL Injection
    - Upotreba Hibernate-ovog validatora i parametrizovanih upita
- XXE Injection
    - Konfigurisanje JAX-B parsera da onemogući eksterne entitete(default-na konfiguracija, IS_SUPPORTING_EXTERNAL_ENTITIES=false)
- HQL Injection
    - Upotreba parametrizovanih upita
- XML Entity Injection
    - Validacija putem šeme

A2 – Broken Authentification and Session Management

Aplikacije koje ispoljavaju ovu ranjivost su podložne kompromitovanju lozinki i sesije.

Rešenje problema:

- Ovaj problem je u nekoj meri rešen samim time što koristimo REST arhitekturu, koja je stateless.
- Lozinke se čuvaju na bezbedan Hash and Salt način, moraju biti duge minimum 8 karaktera gde mora postojati barem jedno veliko i malo slovo, cifra i specijalan karakter(@#$%^&+=!)
- Autentifikacija, a i cela aplikacija se koristi preko TLS odnosno HTTPS protokola
- Server generiše tajni token koji služi za praćenje sesije s toga nije moguće da se neko drugi predstavi kao korisnik koji nije. Tokeni imaju vek trajanja te se posle određenog vremena(1h) sami invalidiraju

- Sesija se prekidaju *Logout* metodom čime se sesija invalidira na backend-u. Identifikator sesije nije dostupan putem URL-a.

A3 – Sensitive Data Exposure

Komunikacija između klijenta i servera se obavlja preko TLS/SSL, tj. HTTPS konekcije.

Potrebno je izbegavati čuvanje osetljivih informacija kao otvoreni tekst. Lozinke korisnika su enkriptovane korišćenjem Bcrypt Hash and Salt. Takođe, da se koriste informacije kao što su npr broj kreditne kartice, i one bi bile enkriptovane.

Osetljivi podaci u application.properties fajlu nisu običan tekst, već su učitani kao environment variable te nisu vidljive korisniku. Takođe potrebno je podesiti ACL kako bi svako imao pravo čitanja/pisanja fajlova koji su mu dozvoljeni.

Takođe, bilo bi dobro u finalnoj verziji aplikacije imati i repliciranu bazu podataka.

A4 – XML External Entities(XXE)

Za razmenu poruka preko SOAP-a koristi se verzija 1.2, koja je sigurnija po pitnaju XXE napada i manja je verovatnoća napada nego u verziji 1.1.

Svi dolazni podaci se validiraju putem XML šeme (odbrana od XML Entity Injection).

Korišćenjem JAXB parsera po default-u su onemogućeni eksterni entiteti te je XXE Injection malo verovatno da će se desiti.

A5 – Broken Access Control

Na backend-u se koristi prethodno pomenuti token, te korsnik ne može da pristupi podacima koji nisu vezani za njega. Token se invalidira posle određenog vremena ili prilikom logout-a.

Korišćene su permisije(RBAC model) kako bi se obezbedilo da određeni korisnik sistema može pristupiti samo akcijama koje su mu dozvoljene. Dodate su privilegije za svaki tip korisnika.

Takođe, putem ACL je korisnicima zabranjena izmena, i limitirano čitanje foldera i fajlova aplikacije.

U aplikaciji se koristi loger koji zapisuje sve čudne/neželjene akcije korisnika, kako bi admini imali uvid u neželjene akcije i probleme.

Prilikom podizanja aplikacije na internet trebalo bi isključiti web server direktorijum listing i pobrinuti se da se metadata fajlovi(npr .git) i backup fajlovi ne nalaze u web roots-u.

A6 – Security Misconfiguration

Neophodno je koristiti ažurirane verzije softvera koji uključuju oeprativni sistem, sistem za upravljanje bazom podataka, komponenatama i spoljašnjim šablonima.

Potrebno je konfigurisati predefinisane(admin) naloge tako da im korisničko ime i lozinka ne budu generički i predvidivi(admin/admin).

U finalnoj verziji aplikacije, tj. prilikom podizanja aplikacije na internet, potrebno je podesiti angular tako da ne bude u dev mode-u, već production mode-u. Production mode uklanja sve logove koji su korišćeni za debug-ovanje i zatvara debug-port aplikacije, koji bi mogli otkriti previše informacija korisnicima prilikom rukovanja greškama.

A7 – Cross-Site Scripting(XSS)

Dva primera XSS napada su Stored XSS I Reflected XSS. Stored XSS se realizuje u slučaju da se maliciozni izvršivi podaci sačuvaju na server. Reflected XSS napad podrazumeva direktno prikazivanje korisničkog unosa u browser-u, pre nego što se na server unos validira.

Ovo je u značajnoj meri sprečeno korišćenjem Angular framework-a. On vrši zaštitu aplikacije od XSS napada escape-ovanjem specijalnih karaktera pre obrade od strane browser-a.

Takođe izvršeno je i validiranje korisničkog unosa, tj. manuelno escape-ovanje specijalnih karaktera kako na frontend-u tako i na backend-u.

A8 – Insecure Deserialization

Ključna stvar je ne prihvatati serijalizovane objekte iz nepoverljivih izvora i korišćenje sanitizatora koji dozvoljavaju samo primitivne tipove.

Potrebno je logovati deserijalizacione izuzetke i otkaze, gde dolazni tip podatka nije očekivani tip, ili deserijalizacija baci izuzetak.

A9 – Using Components with Known Vulnerabilities

U cilju analiziranja ranjivosti spoljašnjih zavisnosti, korišćen je OWASP-ov Dependency-Ckeck plugin. Rezultati se nalaze generisani u fajlu html formata. High severity problemi su bili vezani za tomcat i spring cloud commons. Rešenje za tomcat je korišćenje verzija 9.0.20 ili novije, i 8.5.40 i novije. Takođe Spring framework verzija 5.0 pre 5.0.5 i verzije 4.3 pre 4.3.15 i starije nepodržane verzije su podložne client-side mutlipart zahtevima, te je potrebno ažuriranje verzije.

A10 – Insufficient Logging & Monitoring

Veoma je važno beležiti čudna i neželjena ponašanja aplikacije. Za ovo je korišćen logger iz Logger klase slf4j biblioteke, gde je takođe omogućeno da aplikacija nastavi sa radom i prilikom otkaza logera. Nakon određenog vremena ili dostignute postavljene veličine log fajla, on se arhivira i pravi novi u koji se upisuju podaci.

Prilikom vršenja komandi koje su dosta osetljive(poput logovanja i sl.) vrši se validacija i beleži se IP adresa i podaci o neuspešnoj akciji.

Takođe potrebno je potpuno ukloniti ili svesti na minimum prikazivanje nepotrebnih informacija korisniku i napadaču koje dolaze iz debug mode-a, console log-a i sl. i to se postiže podizanjem aplikacije u production mode-u.

Postoje dva tipa logova, info logovi (gde se sve beleži) i severe logovi gde se beleže samo kritične akcije.

Logove nije moguće ručno menjati, i to je nameštено preko ACL.