IB 2023/2024 zahtevi

1 BEZBEDNOST SISTEMA I ZAŠTITA PODATAKA

Razvoj bezbednog softvera podrazumeva ugrađivanje bezbednosnih kontrola u softver tokom njegovog razvoja, prateći koncept poznat kao *built-in security*. Trošak ugrađivanja bezbednosti na ovaj način je najmanji i, po pravilu, bezbednost je implementirana najkvalitetnije, jer se koncipira bezbedan dizajn koji će kod poštovati, umesto da se bezbednost *ad hoc* prilagođava već napisanom kodu.

Potrebno je implementirati sledeće bezbednosne mehanizme:

1.1 Validaciju podataka

- Sprečiti relevantne *Injection* napade;
- Sprečiti XSS napade;
- Izvršiti validaciju podatka, koristeći kriterijume validacije definisane po najboljim praksama za pisanje bezbednog koda.

1.2 HTTPS komunikaciju

- Potrebno je demonstrirati bezbednu komunikaciju između API gateway-a i klijentske aplikacije

1.2a Demonstracija bezbedne komunikacije između servisa

1.3 Autentifikaciju i kontrolu pristupa

- Omogućiti mehanizme za potvrdu naloga, oporavak lozinke i promenu lozinke;
- Kontrolisanje pristupa *endpoint*-ima po RBAC modelu;
- Kontrola pristupa frontend dela (detalji implementacije prepušteni studentima);
- Testirati i demonstrirati da sve kontrole pristupa rade (pozitivan i negativan ishod);

1.4 Zaštita podataka

Osetljivi podaci sa kojim aplikacija radi treba da budu obezbeđeni u skladištu, u transportu i tokom upotrebe. Identifikovati osetljive podatke, definisati i implementirati prikladne bezbednosne kontrole. Podaci čije skladištenje se ne može izbeći treba da budu šifrovani ili heširani ukoliko je to prikladno. Poruke u internoj komunikaciji treba da imaju očuvanu poverljivost, integritet i neporecivost, kao i da budu zaštićene od *replay* napada.

2 LOGGING I RANJIVE KOMPONENTE

Log zapisi koje generišu aplikacije i operativni sistemi nekog postrojenja su veoma korisni, kako sa aspekta debagovanja problema, tako i za potrebe bezbednosti. Log zapisi predstavljaju osnovni mehanizam za postizanje neporecivosti. Dodatno, kolekcije log zapisa se mogu slati alatima za *monitoring*, poput SIEM alata, čiji zadatak je da prati događaje u sistemu i da okine alarm svaki put kada se sumnjivo ponašanje detektuje. U sklopu odvojene priče, današnji softverski sistemi značajno zavise od komponenti koje nisu dizajnirali i programirali inženjeri originalnog sistema. Od infrastrukture (operativni sistem, baza podataka) do alata (radni okvir, biblioteke), značajan deo koda nije pod našom kontrolom. Međutim, to ne smanjuje našu odgovornost kada nam softver bude eksploatisan zbog ranjivosti u nekoj *third-party* komponenti jer iako nismo pravili tu komponentu, svesno smo je integrisali u naše rešenje.

Potrebno je implementirati logging mehanizam koji ispunjava sledeće zahteve:

1. **Kompletnost** – log zapis mora da sadrži dovoljno informacija da dokaže neporečivost i svaki događaj za koji je neporečivost potrebna treba da bude zabeležen. Dodatno, svaki *security-related* događaj, interesantan za potrebe *monitoring*-a, treba da bude zabeležen.

2. **Pouzdanost** – logging mehanizam treba da bude pouzdan, što podrazumeva dostupnost samog mehanizma (gde je neophodno voditi računa o memorijskom zauzeću log datoteka i napraviti mehanizam za rotaciju logova), kao i integritet log datoteka. Dodatno, dizajnirati kod tako da aplikacija nastavi sa radom u slučaju da logging mehanizam otkaže.

3. **Konciznost** – logging mehanizam treba da proizvodi najmanju količinu zapisa koji su potrebni da ispuni svoju svrhu. Dodatno, optimizovati svaki zapis da sadrži sve informacije, a zauzima najmanju količinu memorije.

2.1 Ranjivosti

Neophodno je:

- Definisati alate koji će se koristiti za proveru različitih skupova komponenti.
- Isproveravati svaku od komponenti i sakupiti listu ranjivosti.
- Analizirati ozbiljnost ranjivosti i mogućnost eksploatacije.
- Definisati i izvršiti strategiju za razrešenje mogućih rizika.
- Kreirati izveštaj neke vrste, koji će istaći temeljnost analize i krajnje rezultate. Format i tačan sadržaj je proizvoljan.

Zahteve navedene iznad je moguće formalno ispuniti bez istraživanja i mnogo truda, no to rešenje neće biti kvalitetno. Kvalitet je upravo ono što se ocenjuje, i da bi se date stavke ispunile neophodno je razmotriti savete i najbolje prakse koje možete pronaći online, poput onih navedenih u OWASP ASVS standardu.

Napomena:

- Obratiti pažnju na *best practice* konfiguraciju bezbednosnih funkcija koje se koristite.
- Potrebno je implementirati funkcionalnosti tek toliko da se podrži smislena demonstracija bezbednosnih kontrola.
- Pojedine tačke je moguće rešiti uz pomoć tehnologije za implementaciju softvera (jezika, radnog okvira) ili alata – ovo je dozvoljeno, no neophodno je razumeti kako tehnologija rešava problem i o čemu treba voditi računa da se pružena bezbednosna kontrola „ne pokvari“.
- Prilikom istraživanja i implementacije kontrola, neophodno je voditi računa o bezbednoj konfiguraciji kontrole – skup parametara koje kontrola ima i njihova *best practice* vrednost.

Ocenjivanje

Za ocenu 6 neophodno je implementirati sledeće zahteve:

- **1.1**
- **1.3**
- **1.4**

Za ocenu 7 neophodno je implementirati sve za ocenu 6 i sledeće zahteve:

- 1.2
- integrisati postojeći sistem sa nekim od alata za statičku analizu koda (SonarCloud, SonarQube, …). Analizirati i rešiti ranjivosti u skladu sa preporukama alata.

Za ocenu 8 neophodno je implementirati sve za ocenu 7 i sledeće zahteve:

- **2**

Za ocenu 9 neophodno je implementirati sve za ocenu 8 i sledeće zahteve:

- 2.1

Za ocenu 10 neophodno je implementirati sve za ocenu 9 i sledeće zahteve:

- 1.2a

Kontrolne tačke

U toku semestra rad na projektu biće proveravan 2 puta. Prva kontrolna tačka će se održati sredinom semestra (okvirno kraj novembra), a druga kontrolna tačka ujedno će predstavljati i finalnu odbranu projekta (okvirno januar). Od ukupno 35 bodova koliko nosi KT1, na finalnoj odbrani-KT2 može se nadoknaditi maksimalno 10 bodova.

KT1 (35 bodova)

- 1.1 za implementirani deo projekta na drugim predmetima
- 1.4 za implementirani deo projekta na drugim predmetima
- 1.3
- 1.2

KT2 (36 bodova)

- 1.1 za ceo projekat
- 1.4 za ceo projekat
- 1.2a
- 2
- 2.1

Usmeni ispit

- Usmeni ispit **je obavezan za ocene 9 i 10**.
- Usmeni ispit **nije obavezan za ocene 6,7,8**.
- Ko na predispitim obavezama ispuni zahteve za ocene 6 i 7 može izaći na usmeni maksimalno za ocenu 8.

Dodatni rokovi

Pored odbrane na KT2 u januaru, organizovaće se još dva termina odbrane predispitnih obaveza, u junu i septembru. Odbrane će biti održane maksimalno za ocenu 6, bez mogućnosti izlaska na usmeni ispit.

Projekat za studente koji polažu samo informacionu bezbednost

Potrebno je implementirati sve iz specifikacije SOA-NoSQL osim stavki 1.12 1.14, 1.15 i 1.16. Aplikaciju po izboru implementirati kao monolitnu ili mikroservisnu. Kao bazu podataka koristiti baze po izboru. Za ocenu 10 aplikacija mora poštovati mikroservisnu arhitekturu. Ovu verziju projekta izrađuju jednočlani timovi.