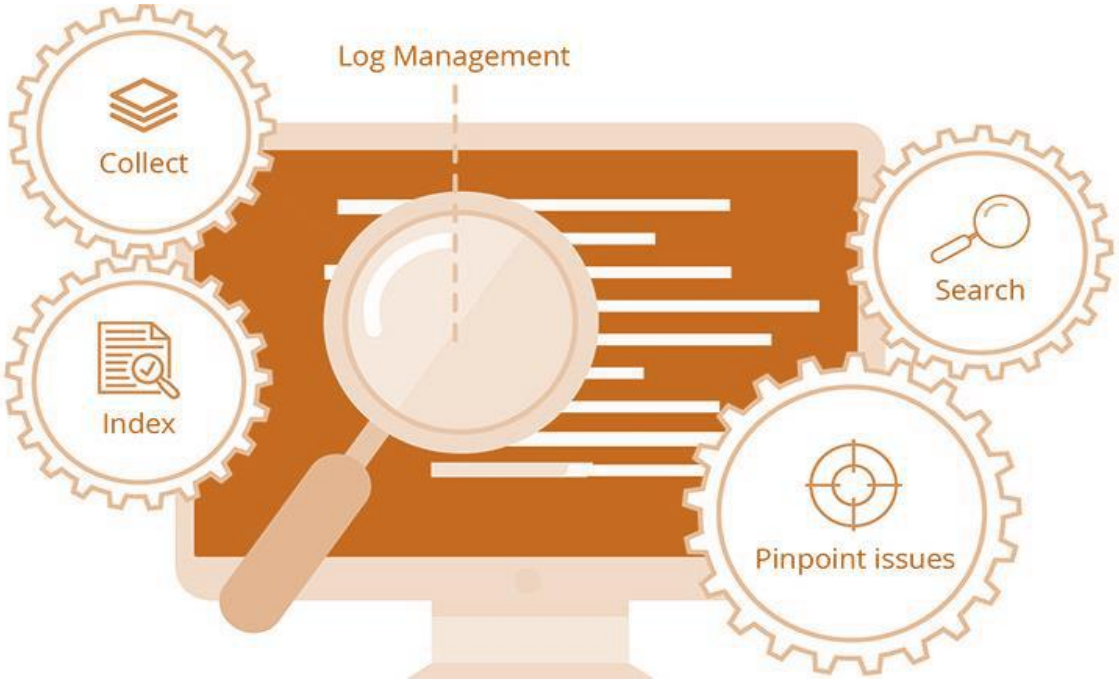

Logging i ranjive komponente

Kontekst

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 razvijači 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.

Specifikacija

Potrebno je implementirati logging mehanizam koji ispunjava sledeće zahteve:

1. **Kompletnost** – log zapis mora da sadrži dovoljno informacija da dokaže neporecivost i svaki događaj za koji je neporecivost 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 napraviti mehanizam za rotaciju logova), kao i integritet log datoteka.

3. **Upotrebljivost** – podržati efikasnu ekstrakciju događaja za neporecivost, kao i security-related događaja.

4. **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.

Takođe, neophodno je izvršiti bezbednosnu analizu svih *third-party* komponenti na koje se vaše rešenje oslanja (od operativnog sistema do *front-end* biblioteka i sve između). Neophodno je:

- Isproveravati komponente i sakupiti listu ranjivosti.
- Analizirati ozbiljnost ranjivosti i mogućnost eksploatacije.
- Definisati i izvršiti strategiju za razrešenje mogućih rizika.

Napomena: Neophodno je kreirati izveštaj neke vrste, koji će istaći temeljnost analize i krajnje rezultate. Format i tačan sadržaj je proizvoljan. Možete koristiti OWASP Dependency Check (ili neki njegov pandan) za detektovanje poznatih ranjivosti u okviru dependency-ja.