Mini Zanzibar

Git repozitorijum: https://github.com/popovbojana/RBS-Tim22

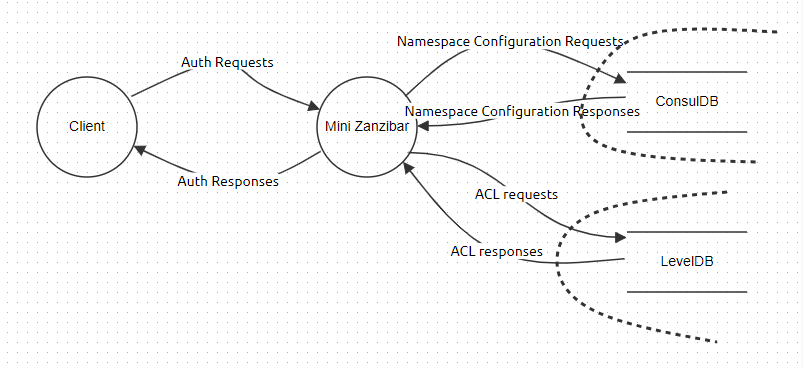
Studenti:

* Bojana Popov SV70/2020
* Andrej Mitrović SV81/2021

Mini-Zanzibar je aplikacija za globalnu kontrolu pristupa, inspirisana Google-ovim sistemom Zanzibar, koja omogućava definisanje i upravljanje politikama autorizacije kroz fleksibilan konfiguracioni jezik. Koristi Google-ov LEVEL DB za skladištenje ACL-ova kao relacionih torki, što omogućava precizno vezivanje korisnika za objekte sa specifičnim pravima pristupa. Pored toga, koristi se i ConsulDB za konfiguraciju namespace-ova, pružajući mogućnost definisanja različitih tipova pristupa i koncentričnih relacija. Mini-Zanzibar obezbeđuje konzistentne i skalabilne odluke o autorizaciji uz minimalno kašnjenje i visoku dostupnost.

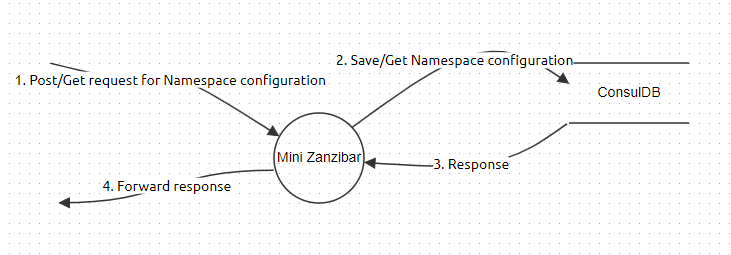
Modeli pretnji

Konteksni model pretnji

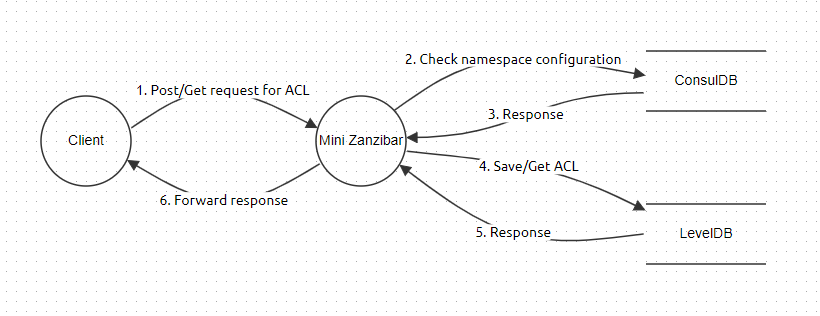


Procesni modeli pretnji

1. Dodavanje/Dobavljanje namespace konfiguracija



1. Dodavanje/Provera ACL-a



OWASP ASVS standardi

OWASP ASVS (Application Security Verification Standard) je set smernica i kontrolnih tačaka koji se koriste za verifikaciju i procenu bezbednosti aplikacija. Neke od standarda koji su primenjeni pri implementaciji Mini zanzibar aplikacije su:

V1.1 Secure Software Development Lifecycle

* 1.1.2 – Modelovanje pretnji

V1.4 Access Control Architecture

* 1.4.1 – Primenjene kontrole pristupa

V1.5 Input and Output Architecture

* 1.5.1 – Zahtevi za ulaz i izlaz jasno definišu kako se rukuju i obrađuju podaci
* 1.5.3 – Validacija ulaznih podataka sprovodi se na poverljivom servisu

V4.1 General Access Control Design

* 4.1.1 – Aplikacija primenjuje pravila kontrole pristupa na pouzdanom servisnom sloju, posebno ako postoji kontrola pristupa na strani klijenta koja bi mogla biti zaobiđena
* 4.1.5 – Kontrole pristupa sigurno otkazuju, uključujući kada dođe do izuzetka

V5.1 Input Validation

* 5.1.3 – Sav unos je validiran koristeći pozitivnu validaciju

V5.2 Sanitization and Sandboxing

* 5.2.2 – Nestrukturirani podaci se sanitizuju kako bi se primenile sigurnosne mere poput dozvoljenih karaktera i dužine

V5.3 Output Encoding and Injection Prevention

* 5.3.3 – Kodiranje izlaza čuva izabrani skup karaktera i lokal korisnika, tako da je svaka Unicode tačka karaktera validna i sigurno obrađena
* 5.3.4 – Selekcija podataka ili upiti baze podataka koriste parametrizovane upite, ORM-ove, entity frameworks ili su na drugi način zaštićeni od napada injekcijom baze podataka

V7.4 Error Handling

* 7.4.1 – Prikazuje se generička poruka kada dođe do neočekivane ili bezbednosno osetljive greške
* 7.4.2 – Obrada izuzetaka (ili funkcionalni ekvivalent) se koristi širom koda kako bi se obrađivali očekivani i neočekivani uslovi grešaka

V10.1 Code Integrity

* 10.1.1 – Koristi se alat za analizu koda koji može otkriti potencijalno zlonamerni kod

V11.1 Business Logic Security

* 11.1.1 – Aplikacija obrađuje tokove poslovnih logika za istog korisnika redosledom koraka bez preskakanja koraka
* 11.1.2 – Aplikacija obrađuje tokove poslovnih logika tako da se svi koraci obrade u realnom vremenu, odnosno da transakcije nisu podnete prebrzo

V13.1 Generic Web Service Security

* 13.1.2 – API URL-ovi ne otkrivaju osetljive informacije, kao što su API ključevi, sesijski tokeni itd.

V13.2 RESTful Web Service

* 13.2.2 – Validacija JSON šeme implementirana i proverena pre prihvatanja unosa
* 13.2.5 – REST servisi eksplicitno proveravaju da li dolazni Content-Type odgovara očekivanom, kao što su application/xml ili application/json

V14.2 Dependency

* 14.2.4 – Third-party komponente dolaze iz unapred definisanih, pouzdanih i kontinuirano održavanih repozitorijuma