CHAM ROCH THI

Backend Security & Authorization Engineer (Freelance / Remote)

Location: Ho Chi Minh City, Vietnam (Remote Available)
Email: rochthi59@gmail.com
GitHub: github.com/Rochthii
Phone: +84329812996
Portfolio: rochthi.id.vn

SUMMARY

Second-year Software Engineering student specializing in secure backend systems, authorization architectures, and multi-tenant SaaS security.

Built multiple security-focused projects involving:

- OAuth2 / OpenID Connect
- JWT / DPoP
- RBAC / ABAC
- Policy-based authorization
- PostgreSQL Row-Level Security
- Zero Trust Architecture
- Financial-grade API security

Seeking freelance and remote opportunities involving API authorization, backend security, and secure SaaS architectures.

CORE EXPERTISE

Identity & IAM:	OAuth 2.1, OpenID Connect (OIDC), JWT, DPoP, Token Binding
Authorization:	RBAC, ABAC, Policy-Based Access Control (PBAC), Fine-Grained Authorization (FGA)
SaaS & Data:	PostgreSQL Row-Level Security (RLS), Tenant Isolation, WORM Audit Logging
Frameworks:	Zero Trust Architecture (ZTA), FAPI Security Profiles, OWASP Top 10 API Security

TECHNICAL SKILLS

Languages:	Go (Golang), TypeScript, JavaScript, SQL, C++, Python
Backend:	RESTful APIs, Authorization Middleware, Policy Evaluation Engines, Context-Aware Security
Databases:	PostgreSQL (Deep RLS implementation), Tenant Schema Isolation, Secure Connection Pooling
Tools:	Git/GitHub, Docker, Linux Systems, Secure API Gateways

FEATURED PROJECTS

Standalone Policy Engine

github.com/Rochthii/standalone-policy-engine

Technologies: Go, Authz Middleware, Contextual ABAC Framework

Designed and implemented a standalone authorization engine supporting policy-based access control with inspirations from Open Policy Agent (OPA), AWS Cedar, and Google Zanzibar concepts.

- Built an extensible Policy Domain-Specific Language (DSL) execution layer to evaluate complex dynamic access rules.
- Developed stateless, low-latency authorization middleware capable of decoupling permission logic from core business services.
- Supported hybrid RBAC/ABAC models evaluating runtime context (IP, timestamp, resource attributes) to grant/deny access instantly.

Secure Multi-Tenant SaaS Platform

github.com/Rochthii/secure-multitenant-saas

Technologies: Go, PostgreSQL, TypeScript, RLS Engine

Engineered a multi-tenant backend architecture emphasizing strong tenant isolation guarantees and explicit access boundaries at the database level.

- Leveraged native PostgreSQL Row-Level Security (RLS) combined with application context to guarantee tenant data separation, reducing cross-tenant access risks.
- Implemented an immutable, append-only WORM (Write-Once-Read-Many) audit logging system for basic security verification and compliance tracking.
- Architected a unified RBAC system mapping enterprise tenant hierarchies to granular API resource routes.

Secure FAPI-ZTA Dark Services

github.com/Rochthii/secure-fapi-zta-darkservices

Technologies: OAuth 2.1, OIDC, JWT, DPoP Proofs, FAPI Profile

A conceptual architectural demonstration for financial-grade backend services integrating modern identity standards and service concealment concepts.

- Enforced Demonstrating Proof-of-Possession (DPoP) at the HTTP layer to bind access tokens to client cryptographic keys, significantly reducing token replay risks.
- Aligned API design rules with Financial-grade API (FAPI) profiles and Zero Trust Architecture principles.
- Designed a network-level concealment mechanism ("dark services") ensuring APIs remain hidden from unauthorized endpoint discovery scans.

EDUCATION

Posts and Telecommunications Institute of Technology

2024 – Present

Bachelor of Science in Software Engineering (Second-Year Student)

- **Focus Areas:** Software Engineering, Security Engineering, System Architecture, Authorization Systems.

FREELANCE SERVICES AVAILABLE

- **OAuth2/OIDC integration:** Setting up industry-standard federated identity and token validation flows.
- **JWT authentication:** Implementing stateless user sessions with strict verification and cryptographic token handling.
- **RBAC/ABAC systems:** Building flexible user role and dynamic context-attribute permission schemas.
- **Authorization middleware:** Intercepting API endpoints to decouple logic using dedicated access enforcement layers.
- **PostgreSQL RLS:** Writing strict, isolated Row-Level Security rules directly into relational database layers.

- **Multi-tenant SaaS design:** Planning multi-tenant schemas and context routing for safe client data partitioning.
- **API security review:** Auditing backend codebases and endpoint design against access control flaws and OWASP API security risks.