MAKALAH

KEAMANAN DALAM MICROSERVICES

Mata Kuliah Microservice Teori

Disusun Oleh:

Nama : Hafizh Fadhlurrohman
NIM : 2301081006
Kelas : 2-A
Dosen : Ervan Asri, S.Kom., M.Kom

PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI PADANG
2024

DAFTAR ISI

DAFTAR ISI ii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan 2
BAB II PEMBAHASAN 3
2.1 Tantangan Keamanan Dalam Microservices 3
2.1.1 Komunikasi Antar Layanan 3
2.1.2 Manajemen Identitas dan Akses 3
2.1.3 Keamanan API 3
2.1.4 Manajemen Konfigurasi dan Rahasia 3
2.2 Strategi Keamanan Untuk Microservice 3
2.2.1 Zero Trust Security Model 3
2.2.2 Implementasi Gateway API 4
2.2.3 Penggunaan Token JWT (JSON Web Token) 4
2.2.4 Service Mesh 4
2.2.5 Pengelolaan Rahasia dengan Vault 4
BAB III PENUTUP 5
3.1 Kesimpulan 5

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang pesat telah mendorong perubahan paradigma dalam pengembangan perangkat lunak. Salah satu pendekatan arsitektur yang kini banyak diadopsi adalah microservices architecture, yang menawarkan fleksibilitas, skalabilitas, dan kemudahan dalam proses pengembangan dan deployment aplikasi. Arsitektur ini memecah aplikasi menjadi layanan-layanan kecil yang berjalan secara independen dan saling berkomunikasi melalui antarmuka terstandarisasi, seperti API atau protokol komunikasi lainnya.

Namun, di balik keunggulan yang ditawarkan, arsitektur microservices juga membawa tantangan tersendiri, terutama dalam hal keamanan. Tidak seperti arsitektur monolitik yang memiliki satu titik keamanan terpusat, microservices memiliki banyak titik serangan (attack surface) karena banyaknya layanan independen yang saling terhubung. Setiap layanan bisa saja dikembangkan dengan teknologi, bahasa pemrograman, atau framework yang berbeda, yang pada akhirnya membuat pengelolaan keamanan menjadi lebih kompleks.

Masalah keamanan dalam arsitektur microservices mencakup berbagai aspek seperti otentikasi dan otorisasi antar layanan, enkripsi data dalam komunikasi, keamanan API, serta pengelolaan identitas dan akses pengguna. Selain itu, aspek monitoring, logging, serta deteksi dan respons terhadap insiden keamanan juga menjadi hal krusial yang harus dirancang dengan baik.

Di tengah meningkatnya ancaman siber dan kompleksitas sistem TI modern, keamanan bukan lagi hal yang bisa dianggap sebagai pelengkap, melainkan sebagai bagian integral dari perancangan sistem. Oleh karena itu, pemahaman yang mendalam mengenai tantangan dan strategi pengamanan dalam arsitektur microservices sangat penting, terutama bagi pengembang perangkat lunak, arsitek sistem, dan tim keamanan siber.

1.2 Rumusan Masalah

1. Apa saja tantangan keamanan yang muncul dalam penerapan arsitektur microservices?
2. Strategi atau pendekatan apa yang dapat digunakan untuk mengatasi tantangan tersebut?
3. Bagaimana implementasi praktik keamanan terbaik dalam microservices?

1.3 Tujuan

1. Memberikan penjelasan tentang konsep arsitektur microservices dan tantangan keamanannya.
2. Mengidentifikasi risiko keamanan utama dalam sistem microservices.
3. Menjelaskan pendekatan dan strategi keamanan yang dapat diterapkan untuk menjaga integritas, kerahasiaan, dan ketersediaan sistem berbasis microservices.

BAB II
PEMBAHASAN

2.1 Tantangan Keamanan Dalam Microservices

2.1.1 Komunikasi Antar Layanan

Microservices saling berkomunikasi melalui jaringan, yang membuat komunikasi antar layanan menjadi rentan terhadap penyadapan atau serangan man-in-the-middle. Oleh karena itu, dibutuhkan mekanisme enkripsi seperti TLS.

2.1.2 Manajemen Identitas dan Akses

Setiap layanan harus dapat mengenali siapa yang mengaksesnya dan menentukan apakah akses tersebut sah. Penerapan otentikasi dan otorisasi terdistribusi membutuhkan sistem manajemen identitas yang kuat, seperti OAuth2 atau OpenID Connect.

2.1.3 Keamanan API

API adalah pintu masuk utama layanan microservices. API yang tidak diamankan dengan baik dapat menjadi titik masuk serangan. Tantangan meliputi pembatasan laju akses (rate limiting), validasi input, dan perlindungan terhadap serangan umum seperti injection dan XSS.

2.1.4 Manajemen Konfigurasi dan Rahasia

Penggunaan file konfigurasi yang berisi kredensial secara langsung dapat menimbulkan risiko. Penyimpanan rahasia seperti API key, token, dan password harus dilakukan melalui secret management tools.

2.2 Strategi Keamanan Untuk Microservice

2.2.1 Zero Trust Security Model

Prinsip Zero Trust menyatakan bahwa tidak ada entitas yang dapat dipercaya secara default, baik dari dalam maupun luar jaringan. Setiap permintaan akses harus diverifikasi secara ketat.

2.2.2 Implementasi Gateway API

API Gateway dapat membantu mengelola akses ke layanan microservices dengan menerapkan autentikasi, otorisasi, rate limiting, dan logging pada satu titik pusat.

2.2.3 Penggunaan Token JWT (JSON Web Token)

JWT digunakan untuk mengelola otorisasi secara aman antar layanan. Token ini membawa informasi klaim pengguna dan dapat diverifikasi tanpa harus mengakses database terpusat.

2.2.4 Service Mesh

Service mesh seperti Istio atau Linkerd menyediakan infrastruktur untuk mengelola komunikasi antar layanan secara aman dan dapat dipantau. Fitur seperti enkripsi, autentikasi mTLS, dan pemantauan lalu lintas bisa diaktifkan tanpa mengubah kode layanan.

2.2.5 Pengelolaan Rahasia dengan Vault

Tools seperti HashiCorp Vault memungkinkan penyimpanan rahasia secara aman, mengelola rotasi kredensial, dan membatasi akses berdasarkan kebijakan.

BAB III
PENUTUP

3.1 Kesimpulan

Arsitektur microservices menawarkan banyak keuntungan dalam pengembangan perangkat lunak modern, namun juga membawa tantangan besar dalam hal keamanan. Karena sifatnya yang terdistribusi, setiap komponen microservices perlu diamankan dengan pendekatan holistik yang mencakup otentikasi, otorisasi, enkripsi, pengelolaan rahasia, hingga pemantauan sistem secara menyeluruh.

Dengan penerapan strategi seperti Zero Trust, API Gateway, JWT, Service Mesh, dan secret management, organisasi dapat meningkatkan postur keamanan mereka secara signifikan. Oleh karena itu, keamanan harus menjadi bagian integral sejak awal perancangan sistem microservices agar dapat memastikan kerahasiaan, integritas, dan ketersediaan layanan tetap terjaga.