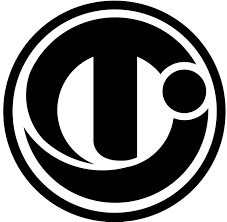
**MAKALAH**

**MICROSERVICE**



**SEMESTER 2**

**Disusun Oleh:**

|  |  |  |
| --- | --- | --- |
| **Nama** | **:** | **Andini Dwiputri Basril** |
| **Nim** | **:** | **2301083011** |
| **Kelas** | **:** | **2A Teknik Komputer** |

**JURUSAN TEKNOLOGI INFORMASI**

**PROGRAM STUDI TEKNIK KOMPUTER**

**POLITEKNIK NEGERI PADANG**

**2025**

**KATA PENGANTAR**

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan makalah yang berjudul "**Keamanan dalam Arsitektur Microservice**" ini dengan baik dan tepat waktu.Makalah ini disusun sebagai salah satu tugas mata kuliah yang bertujuan untuk menambah wawasan dan pemahaman mahasiswa mengenai aspek keamanan dalam pengembangan sistem berbasis microservice yang saat ini banyak digunakan di dunia industri.

Penulis menyadari bahwa penyusunan makalah ini tidak lepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada dosen pengampu mata kuliah, teman-teman, serta semua pihak yang telah memberikan dukungan. Penulis menyadari bahwa makalah ini masih jauh dari sempurna. Oleh karena itu, segala kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa yang akan datang.

Akhir kata, semoga makalah ini dapat memberikan manfaat bagi pembaca dan menambah wawasan di bidang teknologi informasi, khususnya mengenai keamanan dalam arsitektur microservice.

Padang , 08 April 2025

Penulis

**DAFTAR ISI**

**KATA PENGANTAR 2**

**DAFTAR ISI 3**

**BAB 1 PENDAHULUAN 4**

1.1 Latar Belakang 4

1.2 Rumusan Masalah 4

1.3 Tujuan 5

**BAB 2 PEMBAHASAN 6**

2.1 Konsep arsitektur microservices 6

2.2 Dasar Keamanan dalam Sistem Informasi dalam konteks microservices 6

2.3 Tantangan Keamanan pada Microservice 6

2.4 Strategi dan Solusi Keamanan dalam Arsitektur Microservices 7

**BAB 3 PENUTUP 9**

3.1 Kesimpulan 9

3.2 Saran 9

**DAFTAR PUSTAKA 10**

**BAB 1**

**PENDAHULUAN**

**1. 1 Latar Belakang**

Meskipun menawarkan berbagai keuntungan, arsitektur microservice juga menghadirkan tantangan tersendiri, terutama dalam aspek keamanan. Dengan banyaknya layanan yang berjalan secara terpisah dan saling berkomunikasi melalui jaringan, permukaan serangan menjadi lebih luas dibandingkan arsitektur monolitik. Setiap layanan menjadi titik potensial yang dapat diserang jika tidak dilindungi dengan baik.

Oleh karena itu, penting bagi pengembang dan tim keamanan untuk memahami serta menerapkan praktik keamanan yang efektif dalam setiap lapisan arsitektur microservice. Tanpa pendekatan keamanan yang tepat, sistem dapat menjadi rentan terhadap berbagai ancaman seperti pencurian data, penyusupan layanan, dan manipulasi komunikasi antar layanan.

Seiring dengan pesatnya perkembangan teknologi, banyak perusahaan dan pengembang perangkat lunak mulai beralih dari arsitektur monolitik ke arsitektur microservice. Pendekatan ini memecah aplikasi menjadi layanan-layanan kecil yang saling terpisah namun saling terhubung melalui Application Programming Interface (API). Setiap layanan dapat dikembangkan, diuji, dan di-deploy secara independen, sehingga memberikan fleksibilitas dan skalabilitas yang tinggi dalam pengembangan perangkat lunak modern.

**1.2 Rumusan Masalah**

1. Apa yang dimaksud dengan arsitektur microservice dan bagaimana prinsip kerjanya?
2. Apa saja tantangan keamanan yang umum terjadi dalam penerapan microservice?
3. Strategi dan solusi apa saja yang dapat diterapkan untuk meningkatkan keamanan dalam arsitektur microservice?

**1.3 Tujuan**

1. Menjelaskan dan memahami konsep dan prinsip kerja arsitektur microservice.
2. Mengidentifikasi tantangan keamanan yang dihadapi dalam arsitektur microservice.
3. Menyusun strategi dan solusi untuk meningkatkan keamanan microservices

**BAB 2**

**PEMBAHASAN**

**2.1 Konsep arsitektur microservices**

Arsitektur microservice adalah pendekatan pengembangan perangkat lunak yang memisahkan aplikasi menjadi layanan-layanan kecil yang independen dan dapat dikembangkan, diuji, serta di-deploy secara terpisah.

**2.2 Dasar Keamanan dalam Sistem Informasi dalam konteks microservices**

Keamanan sistem informasi adalah perlindungan terhadap data dan layanan dari ancaman akses ilegal, perusakan, atau gangguan layanan yang dapat merugikan pengguna atau organisasi. bagaimana jika ini saya masukkan ke pembaasn utk konsep microservices apa judul yang baik dibuat dlm makakahm tidak mungkin saya bikinnya tinjaun pustaja

**2.3 Tantangan Keamanan pada Microservice**

* **Permukaan Serangan yang Lebih Luas**

Microservice memiliki banyak layanan independen yang masing-masing mengekspos API. Hal ini secara signifikan memperluas permukaan serangan, karena setiap API merupakan potensi titik masuk bagi serangan seperti SQL injection, cross-site scripting (XSS), atau man-in-the-middle (MiTM) jika tidak diamankan dengan baik.

* **Komunikasi Antar-Layanan yang Tidak Aman**

Setiap microservice berkomunikasi melalui jaringan, baik secara internal maupun eksternal. Jika komunikasi ini tidak diamankan dengan enkripsi (misalnya TLS/SSL), maka data sensitif yang dikirimkan dapat disadap atau dimodifikasi oleh pihak ketiga. Serangan seperti sniffing dan MiTM menjadi sangat mungkin terjadi dalam skenario ini.

* **Manajemen Identitas dan Akses (IAM)**

Setiap layanan perlu memiliki mekanisme otentikasi dan otorisasi yang aman. Kesalahan dalam konfigurasi IAM dapat menyebabkan layanan mengakses data atau fungsi yang tidak semestinya. Tanpa pengelolaan hak akses yang granular, pelanggaran data sangat mungkin terjadi akibat akses berlebihan atau akses tidak sah antar layanan.

* **Konfigurasi dan Secrets Management**

Microservices umumnya membutuhkan banyak konfigurasi, termasuk kredensial seperti API key, token, password, atau konfigurasi database. Jika informasi rahasia ini tidak disimpan secara aman atau tersimpan dalam kode sumber, maka dapat dimanfaatkan oleh penyerang untuk mengeksploitasi sistem.

* **Logging dan Pemantauan Terdistribusi**

Dengan arsitektur yang terdistribusi, log dari setiap layanan tersebar di berbagai tempat. Hal ini menyulitkan proses pemantauan dan deteksi dini ancaman secara real-time. Tanpa sistem logging dan monitoring yang terpusat, aktivitas mencurigakan bisa luput dari perhatian.

* **Ketahanan terhadap Kegagalan (Fault Tolerance)**

Dalam arsitektur microservice, kegagalan satu layanan dapat menyebabkan efek berantai pada layanan lain jika tidak ada mekanisme isolasi dan fallback yang tepat. Ini tidak hanya berdampak pada ketersediaan layanan, tetapi juga pada aspek keamanan, karena layanan yang gagal bisa menjadi pintu masuk bagi eksploitasi.

**2.4 Strategi dan Solusi Keamanan dalam Arsitektur Microservices**

Keamanan dalam microservices memerlukan pendekatan menyeluruh yang mencakup pengamanan komunikasi, pengelolaan identitas, isolasi layanan, serta pemantauan sistem. Berikut adalah strategi dan solusi utama yang dapat diterapkan:

1. **Penggunaan HTTPS dan TLS**

Seluruh komunikasi antar-layanan harus dienkripsi menggunakan protokol HTTPS dengan TLS untuk mencegah penyadapan data oleh pihak ketiga (sniffing dan MiTM).

1. **Service Mesh**

Teknologi seperti Istio atau Linkerd menyediakan lapisan infrastruktur yang menangani komunikasi antar layanan dengan fitur keamanan seperti mutual TLS (mTLS), kontrol akses, serta monitoring tanpa perlu mengubah kode layanan.

1. **API Gateway**

API Gateway bertindak sebagai pintu gerbang bagi semua permintaan masuk ke sistem. Ia menyediakan fitur keamanan seperti rate limiting, throttling, filtering, dan autentikasi awal, sehingga meminimalkan risiko serangan dari luar.

1. **Autentikasi dan Otorisasi Berbasis Toke**

Standar seperti OAuth 2.0 dan JSON Web Token (JWT) digunakan untuk memverifikasi identitas pengguna dan layanan. Setiap permintaan harus membawa token yang valid agar layanan dapat mengautentikasi **dan mengotorisasi akses dengan aman.**

1. **Isolasi Layanan (Container Security)**

Setiap layanan dijalankan dalam container terpisah (misalnya menggunakan Docker), dan diatur oleh orchestrator seperti Kubernetes. Ini memastikan isolasi proses dan sumber daya untuk mencegah efek domino jika terjadi pelanggaran pada salah satu layanan.

1. **Prinsip Least Privilege**

Setiap layanan hanya diberikan akses terhadap sumber daya yang benar-benar dibutuhkan menggunakan mekanisme Role-Based Access Control (RBAC). Hal ini mengurangi potensi kerusakan jika terjadi pelanggaran akses.

1. **Manajemen Identitas Terpusat**

Mengelola identitas pengguna dan layanan dari satu sistem pusat membantu menjaga konsistensi dan memudahkan pengaturan hak akses. Sistem ini harus terintegrasi dengan sistem autentikasi dan otorisasi yang aman.

1. **Pemantauan dan Logging Terpusat**

Menggunakan alat seperti Prometheus, Grafana, ELK Stack (Elasticsearch, Logstash, Kibana), atau SIEM tools, organisasi dapat memantau sistem secara real-time dan mendeteksi aktivitas mencurigakan lebih cepat. Logging yang baik juga memudahkan investigasi saat terjadi insiden keamanan.

**BAB 3**

**PENUTUPAN**

**3.1 Kesimpulan**

Arsitektur microservice mempermudah pengembangan dan skalabilitas sistem, namun juga menghadirkan tantangan keamanan yang lebih kompleks. Risiko seperti serangan pada API, komunikasi tidak terenkripsi, dan kesalahan manajemen akses perlu diantisipasi. Oleh karena itu, dibutuhkan strategi keamanan seperti penggunaan HTTPS/TLS, service mesh, API Gateway, dan sistem autentikasi yang kuat untuk menjaga sistem tetap aman.

**3.2 Saran**

Pengembangan sistem berbasis microservices sebaiknya disertai perencanaan keamanan sejak awal. Penggunaan tools keamanan modern, pembatasan akses, dan pemantauan secara real-time harus menjadi bagian dari arsitektur agar sistem tetap aman dan andal.

**Daftar Pustaka**

[**https://www.softwareseni.co.id/blog/mengenal-microservices-pengertian-contoh-dan-kelebihannya**](https://www.softwareseni.co.id/blog/mengenal-microservices-pengertian-contoh-dan-kelebihannya)

Fowler, M. (2014). "Microservices: A Definition of This New Architectural Term." Diakses dari <https://martinfowler.com/articles/microservices.html>

Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.