

#HITB2023AMS

<https://conference.hitb.org/>

HITB
2023
AMS

API Security in the Age of Microservices

Ali Abdollahi | Security engineer | Picnic Technologies B.V.




Agenda

1. Overview of microservices architecture components
 2. API/microservice-related incidents
 3. API vulnerabilities overview
 4. Microservices security challenges
 5. Security best practices
 6. Recap
-



#Whoami



- **Security enthusiast with over 11 years of experience**
- **Doing security stuff at Picnic Technologies B.V.** 
- **A regular speaker at industry conferences e.g.** DefCon3x, Security Bsides6x, Confidence, LeHack, Hacktivity, OWASP global AppSec, IEEE AI/ML, NoNameCon, COSAC, c0c0n, ISACA Euro CACS/CSX and ...

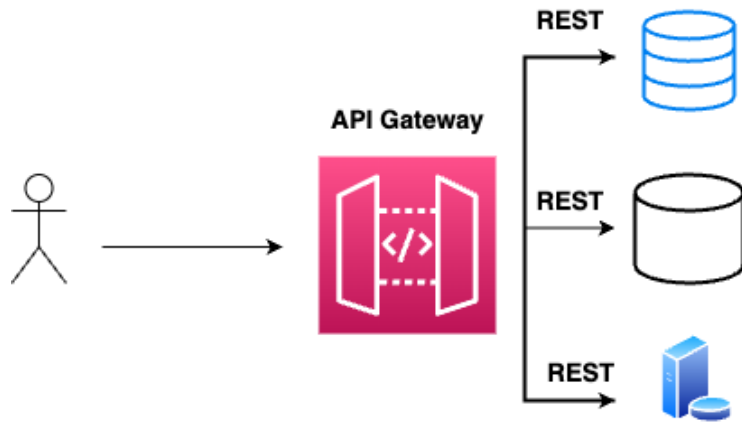


Overview of microservices architecture components

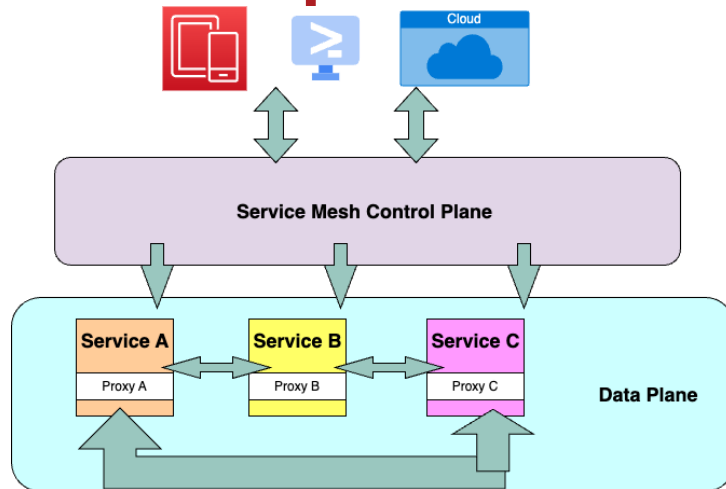
Overview of microservices architecture components

API gateways

- Centralized API entry point
- Manages routing and load balancing
- Enforces security policies
- Monitors API activity
- Simplifies API management



Overview of microservices architecture components



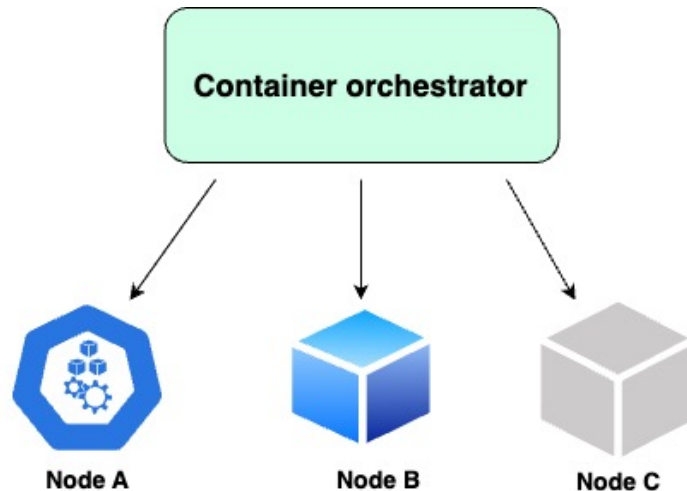
Service Mesh

- Network infrastructure layer
- Facilitates service-to-service communication
- Implements traffic management and resiliency
- Provides observability and monitoring
- Handles service-to-service authentication and encryption

Overview of microservices architecture components

Container orchestrators

- Manages container deployment and scaling
- Automates container lifecycle management
- Ensures high availability and fault tolerance
- Handles load balancing and networking
- Provides monitoring and logging capabilities



Security incidents 'Real-world examples'





Real-world examples

Uber data breach (2016) 

Vector:

Github repo **Shopify (2020)**



AWS API keys



S3 bucket

T-Mobile data breach (2018) 

Vector:

T-Mobile's customer support
staff API

Typeform (2018) **Facebook Data Leak (2019)**

Panera Bread Data Leak (2018)



API vulnerabilities/attacks

#	OWASP API Top 10	Example of API Attack/Vulnerability
✓	API1 Broken Object Level Authorization	Unauthorized access to user records, modifying object properties
	API2 Broken User Authentication	Credential stuffing, session hijacking
✓	API3 Excessive Data Exposure	Exposing sensitive user data, leaking API keys
✓	API4 Lack of Resources & Rate Limiting	Brute force attacks, denial of service
	API5 Broken Function Level Authorization	Accessing restricted resources or performing unauthorized actions
	API6 Mass Assignment	Modifying unintended object properties
	API7 Security Misconfiguration	Default configurations, improper error handling
	API8 Injection	SQL injection, NoSQL injection
✓	API9 Improper Assets Management	Exposing sensitive endpoints, outdated documentation
	API10 Insufficient Logging & Monitoring	Delayed detection or response to security incidents



Microservices Security Challenges

Increased attack surface

Microservices Security Challenges

Multiple APIs and Services

1. More endpoints, multiple APIs, and services = Larger attack surface and an increasing number of vulnerabilities and risks.
2. Complexity in management and addressing security gaps.
3. Multiple APIs and services = Misconfigured security settings, weak authentication mechanisms, and insufficient access controls.
4. Insecure service-to-service communications = Data leakage, man-in-the-middle attacks, and unauthorized access.





Microservices Security Challenges

Unique security vulnerabilities



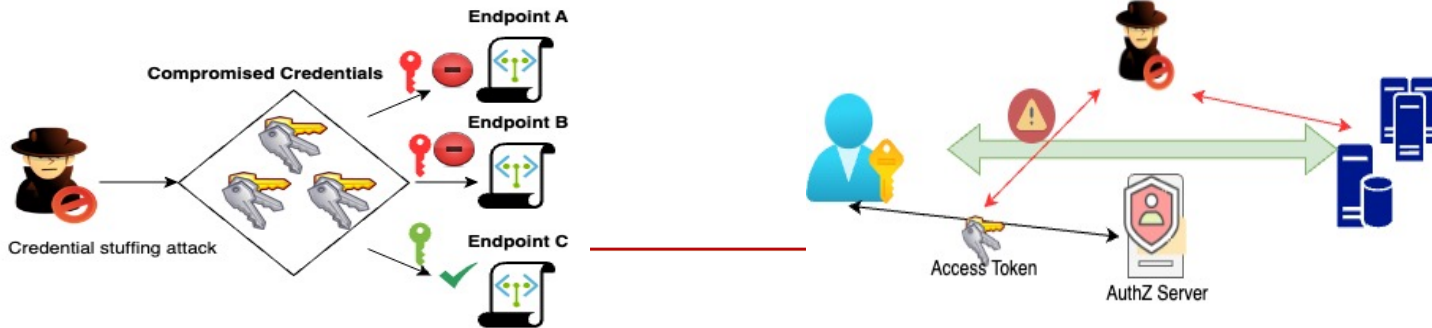
Microservices Security Challenges

API gateway misconfigurations

- Weak authentication and authorization policies
- Improper rate limiting and IP filtering
- Insufficient CORS management

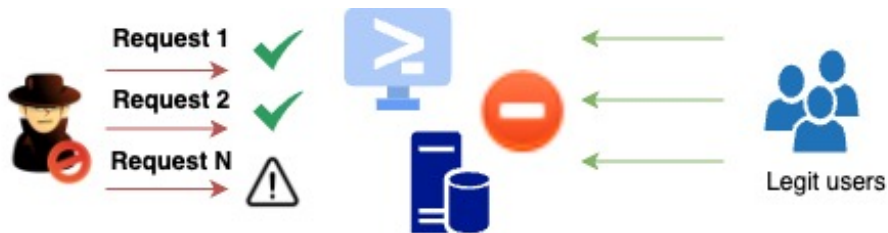
Weak authentication and authorization policies

- Poorly implemented authentication and authorization mechanisms, which may allow unauthorized users to access or manipulate API resources.
- Scenario:
An attacker exploits weak authentication by brute-forcing credentials or exploiting a known vulnerability in the authentication mechanism, gaining unauthorized access to sensitive data or administrative privileges.



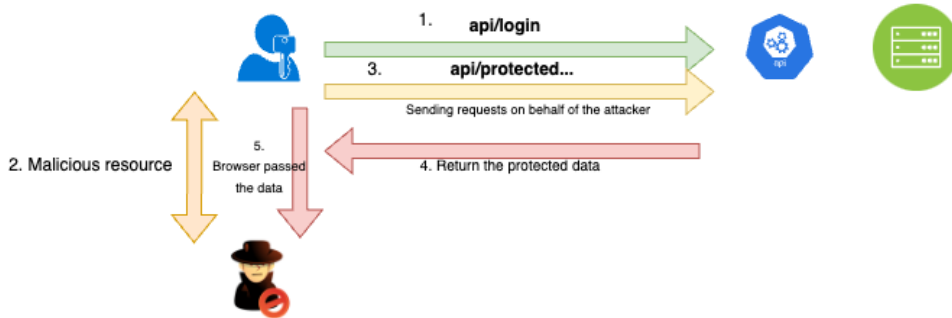
Improper rate limiting and IP filtering

- Insufficient or missing rate limiting and IP filtering measures, which can leave the API susceptible to abuse and denial-of-service (DoS) attacks.
- Scenario:
An attacker initiates a distributed denial of service attack by sending a large number of queries to the API, crushing the server and causing the service to be degraded or unavailable to legitimate users.



Insufficient CORS management

- Incorrect configuration of Cross-Origin Resource Sharing (CORS) policies, potentially allowing unauthorized domains to access or interact with the API.
- Scenario:
An attacker crafts a malicious website that sends requests to the API servers from an unauthorized domain. Due to misconfigured CORS, the attacker can access sensitive data from the API server or perform unauthorized actions on behalf of users who visit the malicious website.





Microservices Security Challenges

Service mesh vulnerabilities

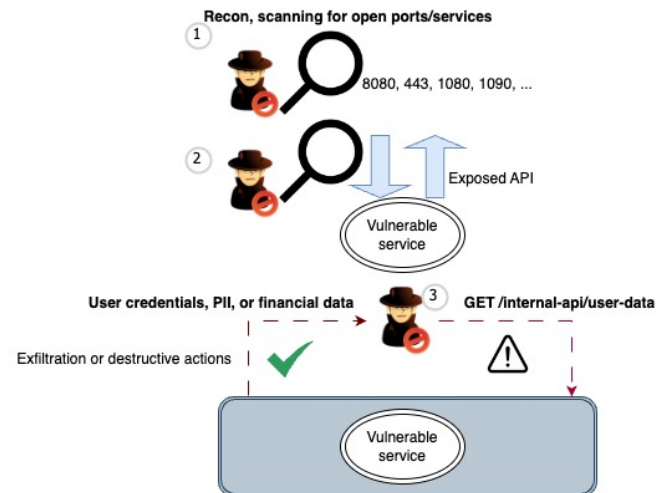
- Misconfigured security policies
- Insecure service-to-service authentication
- Weak data encryption in transit

Misconfigured security policies

- Incorrectly configured or missing security policies in the service mesh, such as exposing internal APIs to the public internet as a result of insecure traffic routing rules or incorrectly configured ingress rules, make services vulnerable to attacks.

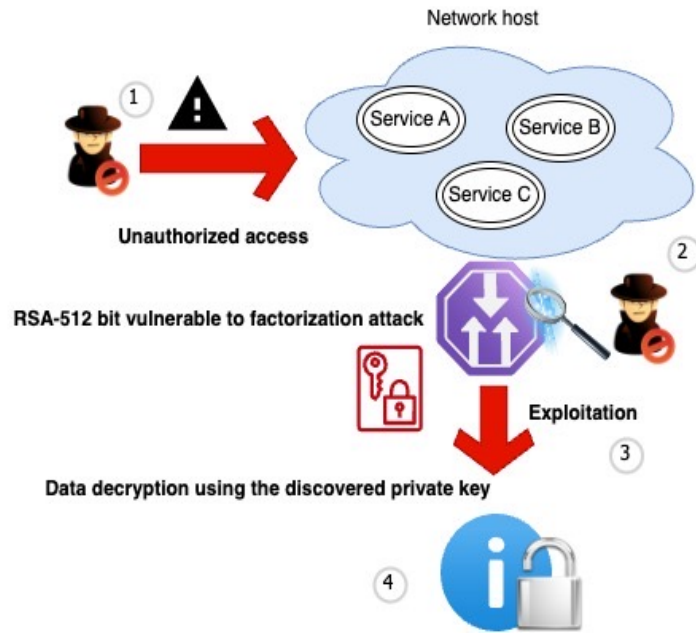
- **Scenario:**

An attacker exploits the misconfigured ingress rule in a service mesh to send malicious requests or access sensitive data from an exposed service.



Weak data encryption in transit

- Insufficient or missing encryption of data transmitted between services in the service mesh, such as using outdated encryption algorithms or failing to implement mTLS, can lead to data leakage or interception.
- Scenario:
An attacker infiltrates the service mesh network and intercepts unencrypted communication between services, exploiting an outdated, vulnerable encryption algorithm (e.g., small key size RSA). This allows data access or manipulation, risking breaches or service disruptions.





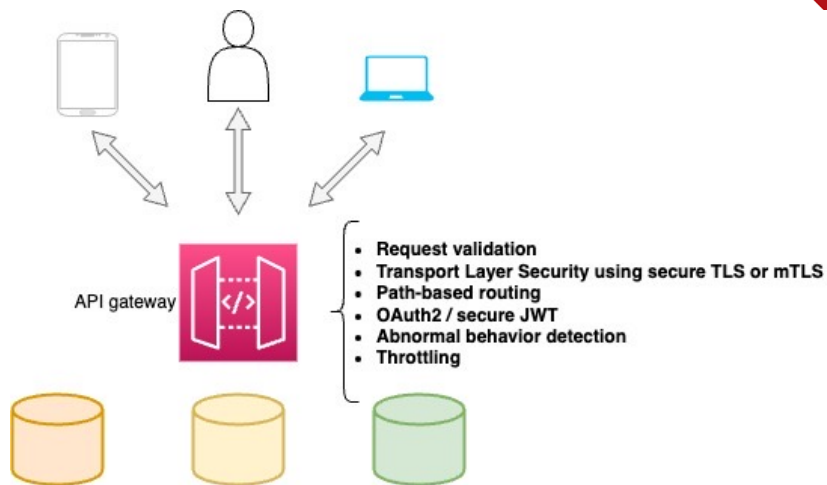
Securing APIs in Microservices (Best Practices)

Implement strong authentication and authorization:

Implement strong authentication and authorization

Secure API gateway layer

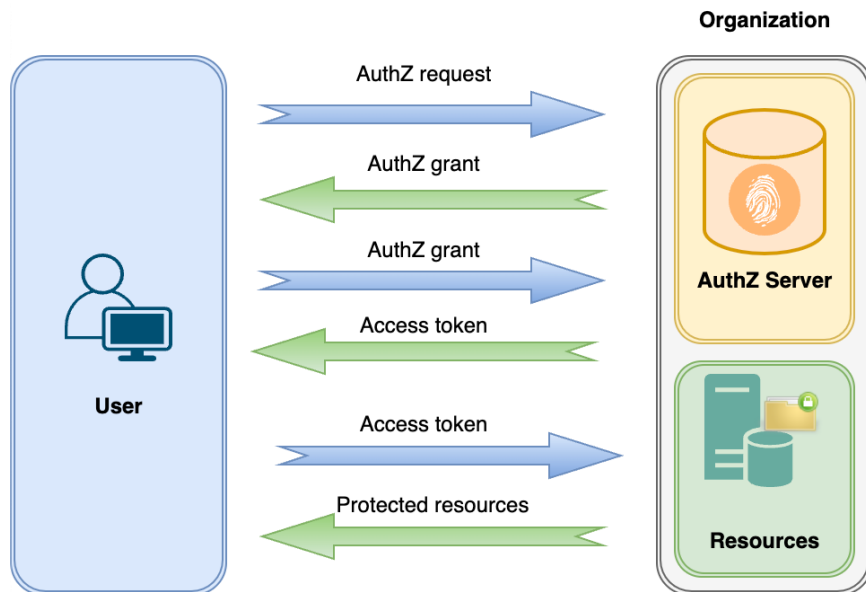
- A centralized entry point for managing API requests
- Handles authentication, authorization, and rate limiting
- Facilitates communication between external clients and microservices
- Provides monitoring, logging, and security features



Implement strong authentication and authorization

OAuth 2.0

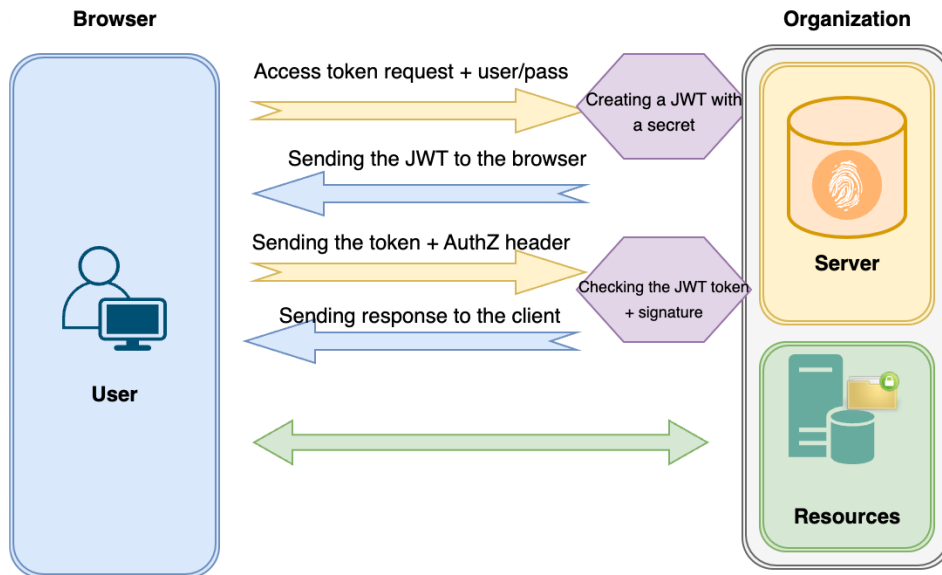
- Centralized authorization framework
- Uses access tokens for API access
- Supports multiple grant types



Implement strong authentication and authorization

JWT

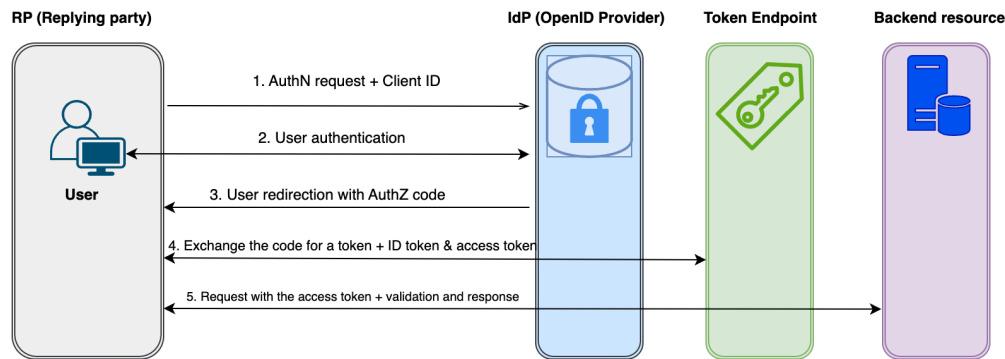
- Lightweight, web-friendly token structure
- Encodes claims as JSON object
- Signed using a digital signature or HMAC



Implement strong authentication and authorization

OIDC (OpenID Connect)

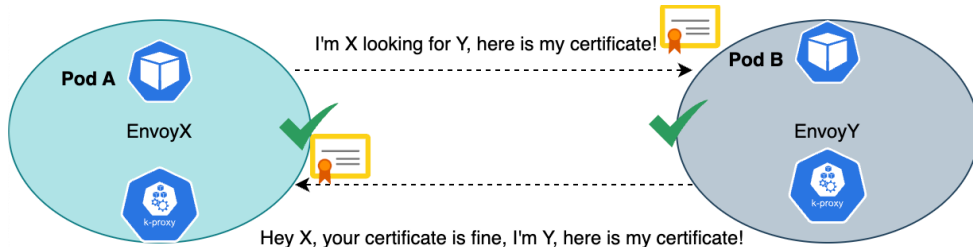
- Authentication layer built on OAuth 2.0
- Provides user identity information
- Utilizes ID tokens (JWT format)



Service-to-service authentication and encryption

Mutual TLS (mTLS):

- Two-way authentication between client and server
- Verifies client and server certificates
- Strengthens security for inter-service communication
- Protects data in transit from eavesdropping and tampering



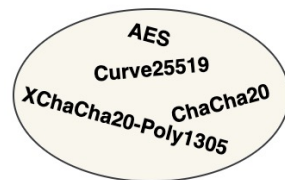


Service-to-service authentication and encryption

More examples:

- Utilize Istio for secure service communication.
- Data encryption with modern algorithms
- Use VPN/IPSec for secure service connections.

e.g. Istio, Linkerd





Anomaly detection

- Spike detection
- Monitoring failed login attempts
- Geolocation analysis
- Monitoring unusual IP addresses
- Request payload analysis



Network Segmentation

- Cloud Security Groups: Restrict access and manage traffic between microservices in cloud platforms (AWS, GCP, Azure).
- Kubernetes Policies: Limit access between pods within a Kubernetes namespace.
- Network VLANs: Establish isolated virtual networks to separate different microservices.
- Firewalls & NAT: Govern traffic across security zones and regulate access to particular resources.



Recap

Secure Design

Secure Comm.

IAM

Least privilege

Defense -in-depth

Data in transit

Data at rest

AuthN & AuthZ

Secret & key mgmt

RBAC
OPA

API gateway
FW/IDPS

mTLS
JWT (RS/ES256)
TLS1.3






DB encryption
RSA/ChaCha20
KMS

OAuth2.0
OIDC
JWT

AWS Secret Manager
Vault



Do not forget...

- ✓ Security monitoring 
- ✓ SAST, DAST, and IAST 
- ✓ Periodic audit (Specifically on your cloud assets e.g. containers) 
- ✓ Hardening (Network and deployments) 
- ✓ Security frameworks e.g. OWASP API Security Top 10, NIST SP 800-204, etc. 

#HITB2023AMS

<https://conference.hitb.org/>



Thank you!