

Розгортання мікросервісів у Zero-Trust середовищі з використанням SPIRE

Модель загроз та оцінка операційних витрат

Нанівська Надія

Проблема: "Як довіряти?"

У мікросервісній архітектурі ключове питання:

1. Як `orders-service` може бути впевнений, що він говорить зі *справжнім* `payments-service`?
2. Як `payments-service` може бути впевнений, що запит надходить саме від `orders-service`, а не від зловмисника?

Чому тема актуальна?

- Мікросервіси та Kubernetes стали стандартом
- Більшість атак сьогодні — через витік секретів
- Компанії масово переходят на **Zero Trust**
- Потреба у автоматизованому, захищенному управлінні ідентичностями

Традиційні Методи Аутентифікації

Метод	Як працює	Основна проблема
JWT	Клієнт передає цифровий токен.	Викрадений токен не можна відкликати . Refresh-токени — ще гірше.
Static TLS	Ноди/сервіси мають статичні сертифікати.	Життєвий цикл ключів дуже довгий. Їх важко обслуговувати та ротувати.

Спільний недолік:

Довгоживучі секрети → високий ризик компрометації.

Чому цього недостатньо?

- Токени часто зберігаються у файлах (K8s Secrets).
- Статичні TLS ключі можуть існувати **місяцями або роками**.
- Будь-який витік → негайний доступ зловмисника.
- Ручна ротація = дорого, повільно, неавтоматизовано.

Zero-Trust Архітектура

Ключові ідеї:

- **Ніколи не довіряй, завжди перевіряй.**
- Довіра не ґрунтуюється на IP, мережі або розташуванні.
- Кожен запит має бути автентифікований.
- Ідентичність прив'язується до *workload*, а не до машини чи порту.

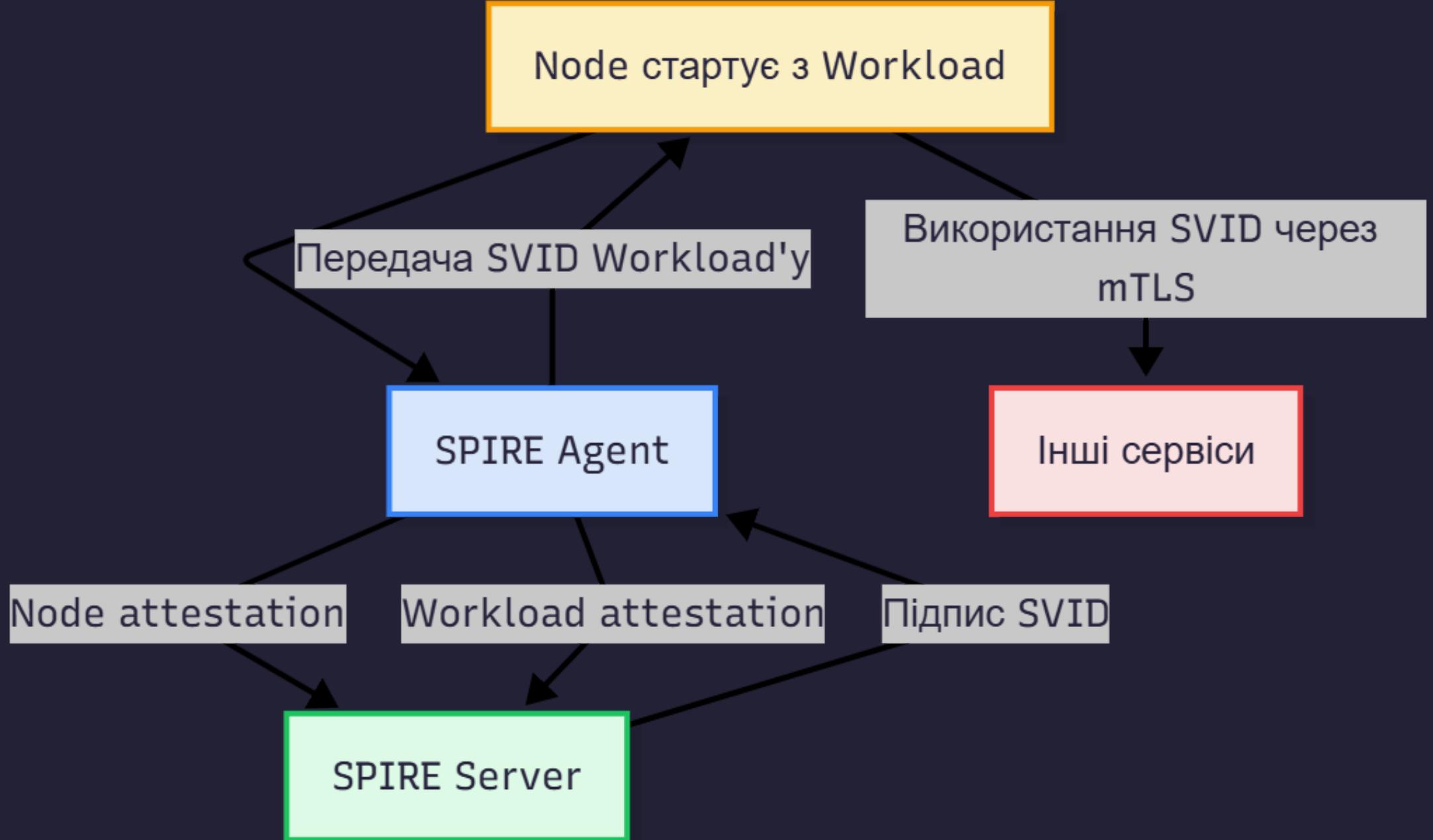
Що таке SPIFFE і SPIRE

SPIFFE — стандарт для надання криптографічної ідентичності сервісам у мікросервісних середовищах.

Ідентичності видаються у вигляді короткоживучих сертифікатів або токенів (SVID), які підтверджують, що сервіс справжній.

SPIRE — реалізація SPIFFE, що автоматизує видачу цих ідентичностей:

- **SPIRE Server** — центральний авторитет, який керує видачею SVID.
- **SPIRE Agent** — встановлюється на вузлах, перевіряє сервіси та передає їм SVID.



Мета дипломної

Порівняти три підходи до сервісної аутентифікації:

- JWT
- Static TLS
- SPIRE (динамічні SVID)

Підхід

- Побудова PoC: 2 Spring Boot сервіси
- Вимірювання операційних витрат (latency, throughput, CPU/mem)
- Аналіз безпеки за моделлю **STRIDE**

Стек технологій

- Kubernetes
- SPIRE Server + Agents
- Spring Boot microservices
- Prometheus + Grafana

Метрики:

- p95/p99 latency
- CPU/memory
- Overhead SPIRE Agent
- Частота ротації SVID

Дякую за увагу!