DB 보안 강화 성능 테스트

https://youtu.be/rgkSLqrlsSY

1. 개　　요

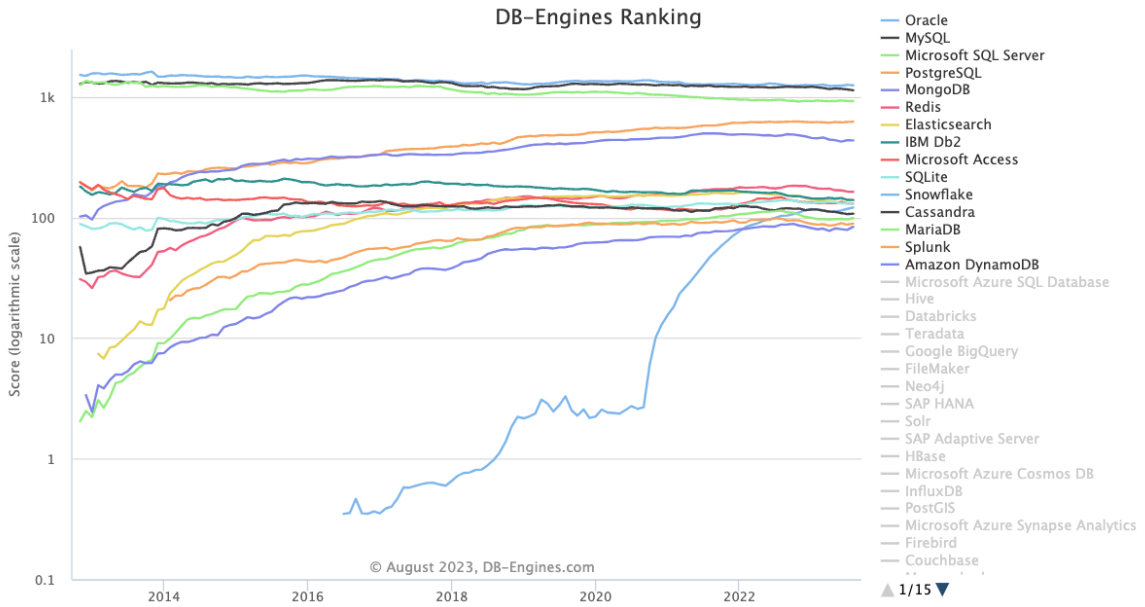
- 과제 목표
  - DB 데이터 암호화에 사용되는 알고리즘 성능 비교 분석

- 목표 알고리즘
  - AES-128 / AES-256

- 성능 측정 시 사용할 DBMS 선택 및 성능 측정 방법 조사
  - 다양한 DBMS 존재
  - 데이터 베이스의 용도가 다양

2. DBMS

- DBMS(Database Management System)
  - 데이터 베이스 관리 시스템, 데이터를 구조화하고 저장하며, 데이터에 접근하고 조작하는 기능을 제공하는 소프트웨어.
    - 데이터 베이스를 효율적으로 관리하고 조작할 수 있도록 다양한 기능과 도구를 제공



출처 : https://db-engines.com/en/ranking

2. DBMS

- Stackoverflow에서 해마다 개발자를 대상으로 하는 설문 결과
  - PostgreSQL과 MySQL는 전문 개발자들이 많이 사용하고, MySQL과 SQLite는 학습용으로 많이 사용하는 것으로 조사됨.

https://survey.stackoverflow.co/2023/

2. DBMS

- MySQL – 관계형 데이터베이스 관리 시스템
  - 오픈 소스라서 무료로 사용 가능
  - 표준 SQL 형식 사용

- PostgreSQL
  - 객체-관계형 데이터 베이스 시스템
  - macOS에서 기본 데이터 베이스로 사용

Oracle은 오픈 소스가 아니라서 무료로 이용할 수 없음

- SQLite
  - 독립형 파일 기반의 오픈 소스 관계형 데이터 베이스
  - 구글 안드로이드 운영 체제에 기본 탑재된 데이터베이스

2. DBMS

- 성능 측정 툴 조사
  - Sysbench
    - DB 서버에서 사용하는 벤치마크 툴
    - 다양한 DBMS를 지원(MySQL, PostgreSQL등, SQLite는 확인 필요)
  - 각 DBMS에서 제공하는 성능 툴
    - MySQL – mysqlslap
    - PostgreSQL – pgbench
- Sysbench를 활용한 성능 측정 후 비교 분석

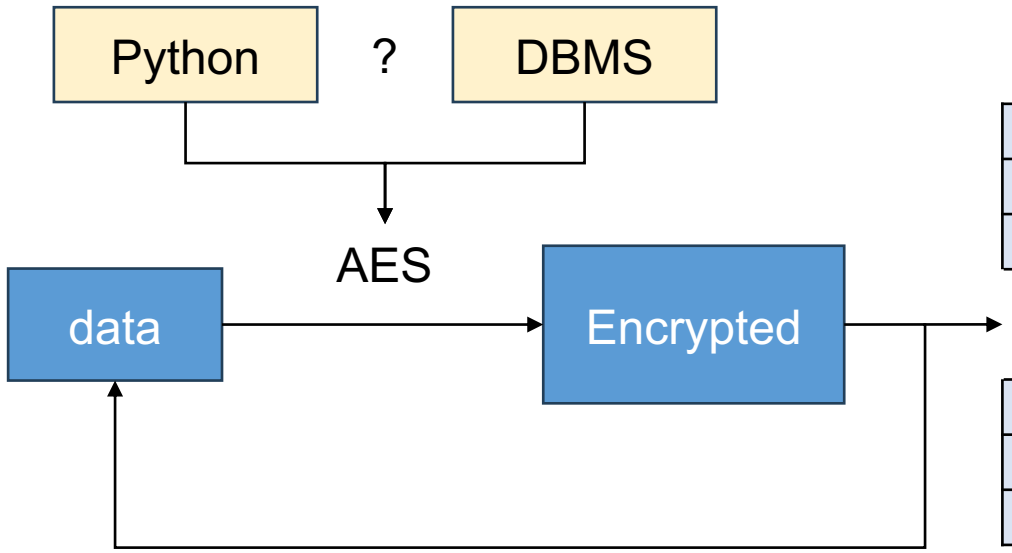
3. 성능 측정

- PostgreSQL를 설치해서 pgbench를 통해 간단하게 성능 테스트를 진행
  - 성능 테스트를 자동으로 해줘서 편리하지만, AES 암호화가 동작하는지 알 수 없고 동작하더라도 AES-256으로 바꾸는게 가능한지 조사해봐야함

```
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 100
query mode: simple
number of clients: 4
number of threads: 4
duration: 30 s
number of transactions actually processed: 165275
latency average = 0.726 ms
initial connection time = 5.156 ms
tps = 5509.708638 (without initial connection time)
```

4. 대체 방안

- Python으로 테스트 툴 구현
  - Python으로 DB 쿼리를 구현하여 반복 실행 후 시간 비교
  - 각 DBMS를 사용하며, 여러 시나리오로 구현하여 성능 분석



0	…	…	…	…	…	…	…
1	…	…	…	…	…	…	…
2	…	…	…	…	…	…	…

. . . .

n-2	…	…	…	…	…	…	…
n-1	…	…	…	…	…	…	…
n	…	…	…	…	…	…	…

5. Python AES 성능 비교

```python
from Crypto.Cipher import AES
import timeit
import secrets

# 16바이트 크기의 랜덤한 값을 생성
# random_bytes = secrets.token_bytes(16)

2개의 사용 위치
def aes_ecb_encrypt(data, key, mode):
    cipher = AES.new(key, mode)
    return cipher.encrypt(data)

data = b'TechTutorialsX!!TechTutorialsX!!'
key128 = secrets.token_bytes(16)  # 16 bytes for AES-128
key256 = secrets.token_bytes(32)  # 32 bytes for AES-256

num_iterations = 1000000

# AES-128 성능 측정
time_taken_aes128 = timeit.timeit(lambda: aes_ecb_encrypt(data, key128, AES.MODE_ECB), number=num_iterations)
print(f"AES-128 Encryption Time for {num_iterations} iterations: {time_taken_aes128} seconds")

# AES-256 성능 측정
time_taken_aes256 = timeit.timeit(lambda: aes_ecb_encrypt(data, key256, AES.MODE_ECB), number=num_iterations)
print(f"AES-256 Encryption Time for {num_iterations} iterations: {time_taken_aes256} seconds")
```

```
AES-128 Encryption Time for 1000000 iterations: 4.812508250004612 seconds
AES-256 Encryption Time for 1000000 iterations: 4.985031458985759 seconds
```

- 32바이트를 암호화할 때, 1,000,000번을 하면 0.1초 정도의 차이
- 생각보다 차이가 많지 않음
  - DB에서 실제로 입력되는 데이터의 정보(종류, 크기 등)을 조사해야함

감 사 합 니 다

https://youtu.be/qV3k-bQgmK0