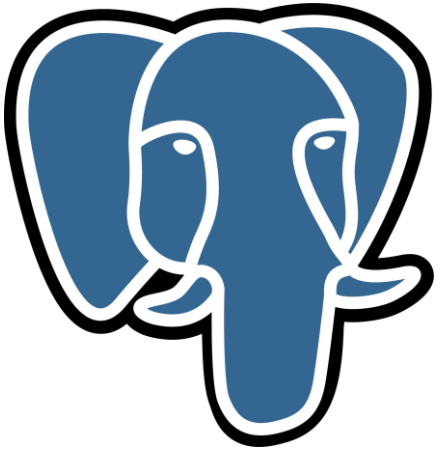


pgday Seoul 2024

PostgreSQL TDE & PCI DSS

DB장이
geartec82@gmail.com

Contents

WW Data breach incidents

KR
Credit card, bank,
E-commerce ETC

JP
Portal, Telco

CN
E-commerce, Portal

US
Bank, Telco, Potal

Why encrypt?

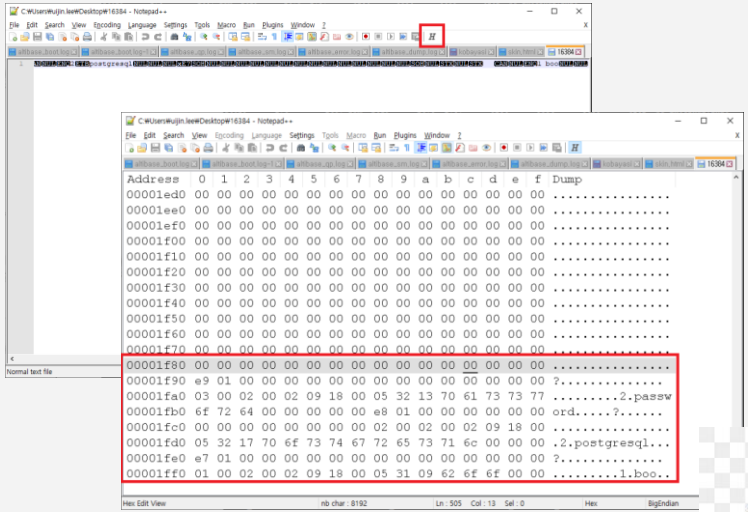
Why encrypt?

위험요소

- 데이터 파일 도난
- 백업 파일 도난
- 데이터 위/변조 (데이터 손상)
- 내부자 위협

Why is encryption important in PG?

```
#Data
postgres=# select * from t1;
c1 | c2
----+------------
1 | boo
2 | postgresql
2 | password
(3 rows)

#Datafile
/pgdata/base/14187/16384
```



https://blog.naver.com/geartec82/221989813505?trackingCode=blog_bloghome_searchlist

What is TDE?

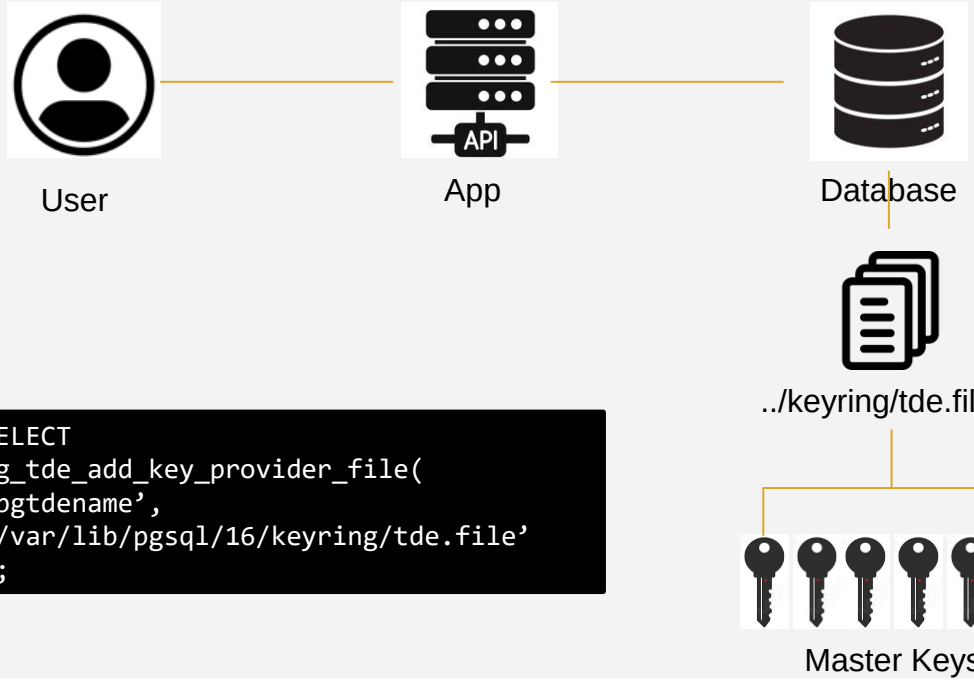
What is TDE?

TDE(Transparent Data Encryption)의 기본 원리는 데이터베이스에 저장된 데이터를 디스크에 쓰기 전에 자동으로 암호화하고, 데이터를 읽을 때는 자동으로 복호화하는 방식입니다.
응용 프로그램이나 사용자가 별도로 암호화 또는 복호화를 처리할 필요 없습니다.

- 저장된 데이터 파일/백업 파일을 암호화
- 암호화 키 관리 외부 / 내부

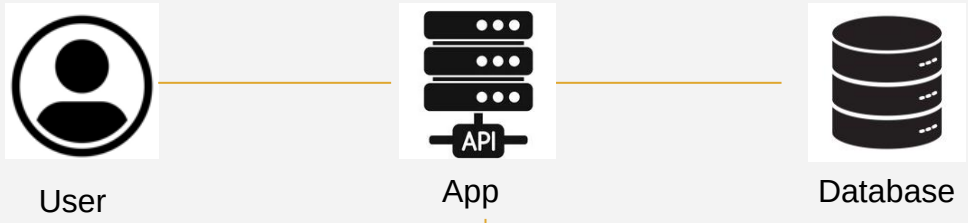
  * 데이터 암호화 키 (DEK)
  * 마스터 키 (MK)

TDE Master key Mgt (1)



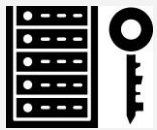

```
SELECT
pg_tde_add_key_provider_file(
'pgtdename’,
'/var/lib/pgsql/16/keyring/tde.file’
);
```

TDE Master key Mgt (2)



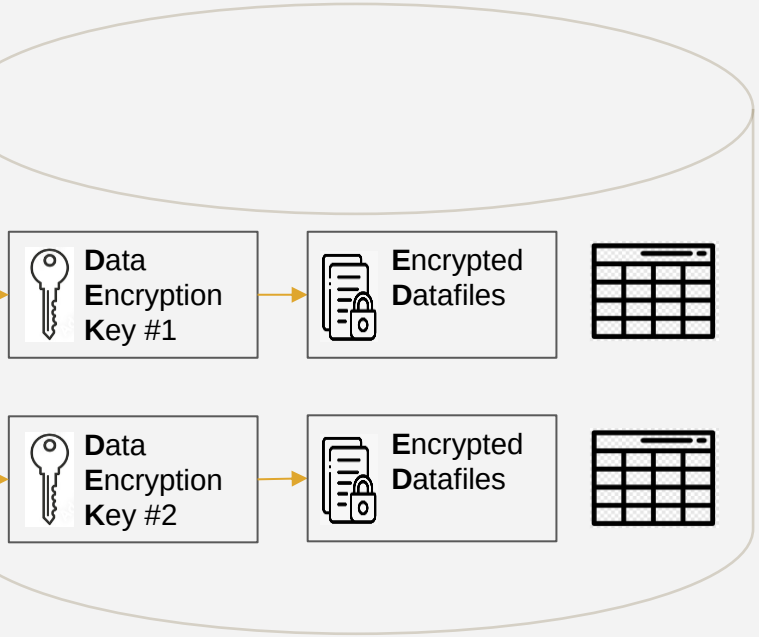

```
SELECT pg_tde_add_key_provider_vault_v2('
'pgtdename',
:'secret_token',
'url',
'mount'
'ca_path');
```

Encrypt Key Management



Master
Key

SW / HW
Internal / **External**

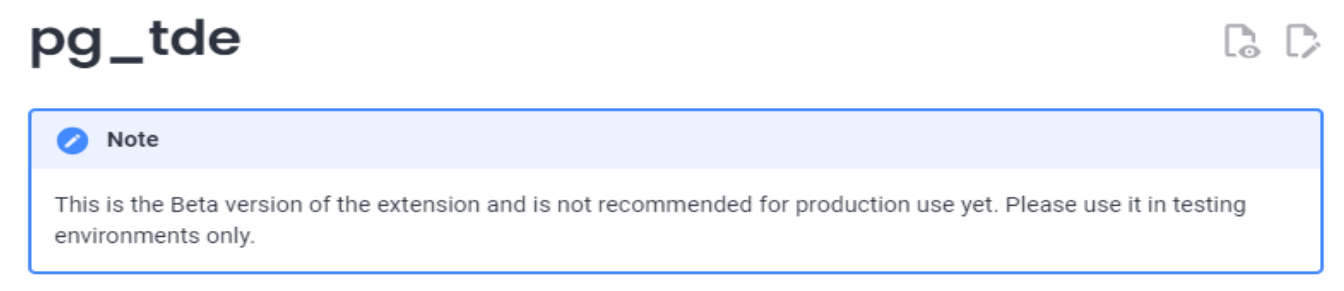


암호키는 서버 또는 하드웨어 토큰에 저장되어질 수 있다. 암호키를 저장하는 서버는 웹 서버 또는 DB 서버와 같은 서버일 수 있으나, 물리적으로 분리되어 있는 서버를 사용하는 것을 권고한다. 하드웨어 토큰은 저장된 정보가 위·변조 또는 외부로 노출되기 어려운 장치로, 스마트카드, USB 토큰 등의 보안토큰(HSM, Hardware Security Module)을 의미한다.

KISA 암호화 기술구현 안내서

```
SELECT pg_tde_set_principal_key(
'my-principal-key',
'file');
```

Ext.pg_tde



PERCONA

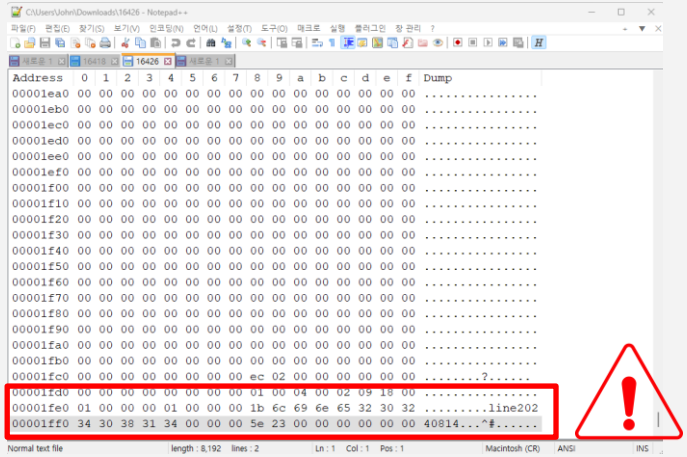
MIT License

1. 해당 소프트웨어를 누구라도 무상으로 제한없이 취급해도 좋다. 단, 저작권 표시 및 이 허가 표시를 소프트웨어의 모든 복제물 또는 중요한 부분에 기재해야 한다.
2. 저자 또는 저작권자는 소프트웨어에 관해 아무런 책임을 지지 않는다

https://github.com/percona/pg_tde

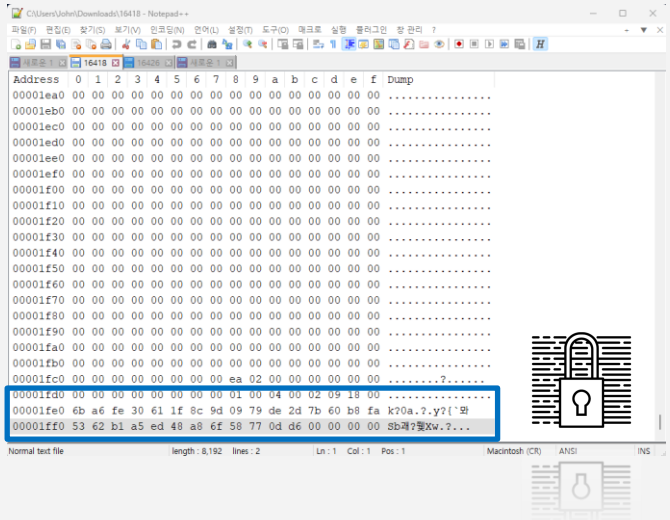
None TDE Encrytion

```
CREATE TABLE albums
(
album_id INTEGER GENERATED ALWAYS AS
IDENTITY PRIMARY KEY,
artist_id INTEGER,
title TEXT NOT NULL,
released date NOT NULL
);

insert into
albums(artist_id,title,released)
values(1,'line20240814',now());
```



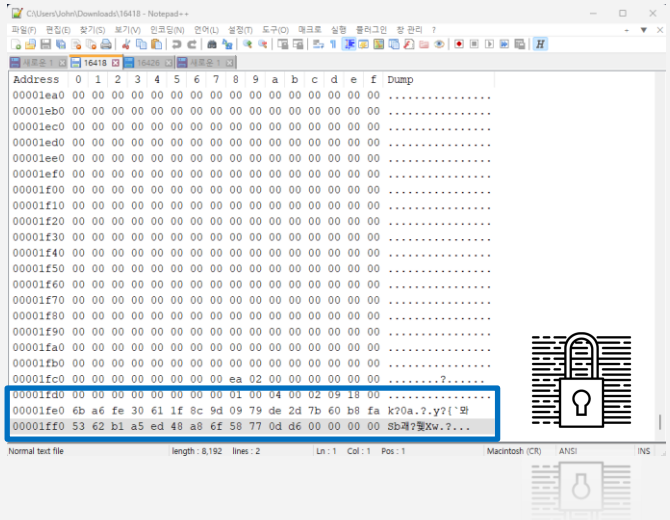
TDE Encrytion

```
CREATE TABLE albums_tde
(
album_id INTEGER GENERATED ALWAYS AS
IDENTITY PRIMARY KEY,
artist_id INTEGER,
title TEXT NOT NULL,
released date NOT NULL
) using pg_tde;

insert into albums_tde(
artist_id,title,released)
values(1,'line20240814',now()
);
```



Backup file

```
#Remote Backup
pg_basebackup -R -D ${PGDATA} -h
source_db_ip -p port -U postgres

#If you don't have a master key file

postgres=# select * from albums_tde;
ERROR:  failed to retrieve master key
"tde-20240814-1_1" from keyring.
```



Other else ?

(PCI DSS)

PCI DSS

PCI DSS(Payment Card Industry Data Security Standard)는 결제 카드 업계에서 카드 소유자의 민감한 데이터를 보호하기 위해 만든 글로벌 보안 표준

PCI DSS는 카드 결제 처리 과정에서 발생할 수 있는 데이터 유출과 사기로부터 카드 소유자의 데이터를 보호하고, 이를 통해 카드 결제 시스템의 보안을 강화하는 것이 목표

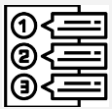
목표)

- 안전한 네트워크 유지
- 카드 소유자 데이터 보호
- 취약점 관리 프로그램 유지
- 접근 제어 강화를 통한 데이터 보호
- 모니터링 및 테스트

버젼별 특징)

- v1.0 : 초기 표준 제정, 기본 보안 요구 사항 (04’)
- v2.0 : 클라우드 서비스 연동 제공 (10’)
- v3.0 : 위험 기반 접근법, 운영 절차 관리 강화. (13’)
- v3.1 : SSL/TLS의 보안 문제 대응 (15’)
- v3.2 : MFA 필수화, 취약점 관리 (16’)
- v4.0 : 주기적인 보안 검토, 더 엄격해진 패스워드 정책 (22’)

PCI DSS on PG



Requirements

r1. 데이터 암호화
r2. 접근제어
r3. 인증
r4. 민감정보 Masking (PAN)
r5. 감사 기능/ 정기 보안 점검
r6. 데이터 라이프 사이클 관리

Solutions

s1. TDE, pg_crypto
s2. pg_bouncer, pg_hba.conf
s3. pg_bouncer(sha-256, open ssh v1.1)
s4. DataMasking
s5. pgAudit
s6. External Partitioning / pg_partman

Ext.Anonymizer

- **Masking function**
- **RULE**
- **Static/Dynamic**
- **Data Generalization**
- **randomization, faking, partial scrambling, shuffling, noise**

https://postgresql-anonymizer.readthedocs.io/en/stable/

Ex. Data Masking

```
sudo dnf install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-8-x86_64/pgdg-redhat-repo-latest.noarch.rpm
$sudo yum install postgresql_anonymizer_16

postgres=# ALTER DATABASE postgres SET session_preload_libraries = 'anon';
postgres=# CREATE EXTENSION anon CASCADE;
postgres=# SELECT anon.init();
postgres=# SELECT anon.start_dynamic_masking();

postgres=# postgres=# create table people (
 id varchar(20),
firstname varchar(20),
lastname varchar(20),
cellphone varchar(11)
);
postgres=# insert into people values ('ktds','sangkee','kim','01000000000');
```

Fake Data / Data Masking

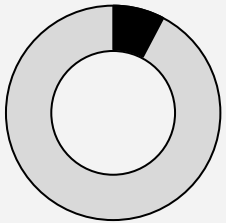
```
postgres=# CREATE ROLE outsider LOGIN;
CREATE ROLE
postgres=# SECURITY LABEL FOR anon ON ROLE
outsider IS 'MASKED';

SECURITY LABEL FOR anon ON COLUMN
people.lastname
IS 'MASKED WITH FUNCTION anon.fake_last_name()';

SECURITY LABEL FOR anon ON COLUMN
people.cellphone
IS 'MASKED WITH FUNCTION
anon.partial(cellphone,2,$$******$$,2)';
```

```
postgres=# select * from people;
  id  | firstname | lastname |  cellphone
------+-----------+----------+-------------
 ktds | sangkee   | kim      | 01000000000
(1 row)

postgres=#
```

```
postgres=# \c postgres outsider
You are now connected to database "postgres" as user "outsider".
postgres=> select * from people ;
  id  | firstname | lastname | cellphone
------+-----------+----------+------------
 ktds | sangkee   | Kerluke  | 01******00
(1 row)

postgres=> select * from people ;
  id  | firstname | lastname | cellphone
------+-----------+----------+------------
 ktds | sangkee   | Goodwin  | 01******00
(1 row)
```

TDE Benchmark

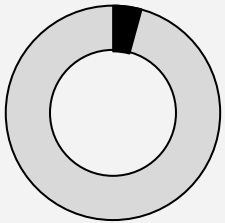
Benchmark Result



+5%

CPU %

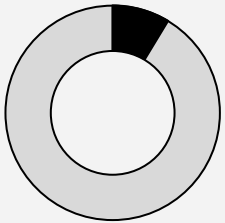
Sys% 는 동일, Usr% 만 4~5% 의 증가가 확인 됨



+2%

Memory

메모리 사용량은 1~2% 차이



7.3%

Overhead

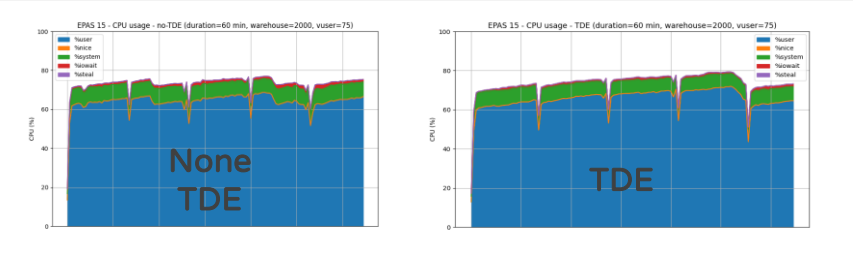
TDE를 사용함에 따라 thoughtput 7.3% 감소

Tools : HammerDB
Senario : TPROC-C
Size : 2000 warehouse (200GB)
vUser : 75 Client
Running Time : 1 Hour

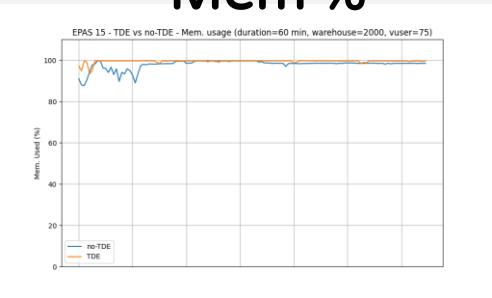
https://www.enterprisedb.com/blog/TDE-Postgres-Advanced-Server-15-Launch

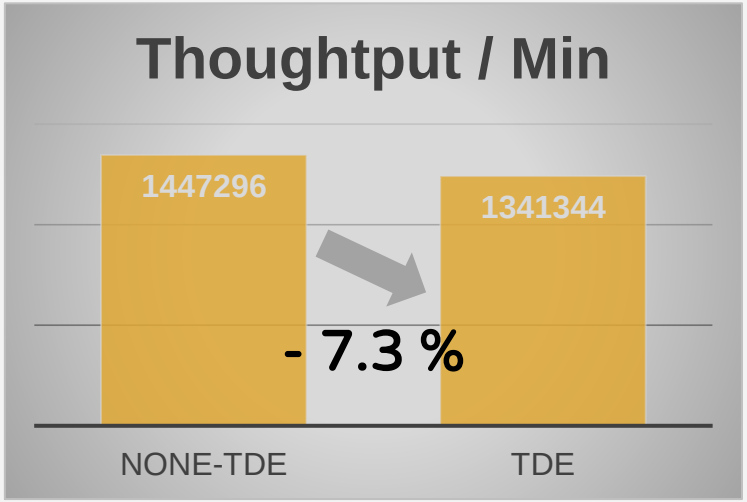
Benchmark Test (HammerDB)

CPU %



Mem %





https://www.enterprisedb.com/blog/TDE-Postgres-Advanced-Server-15-Launch

References

Data Encrypt

- KISA

  한국인터넷진흥원 암호화 기술구현 안내서
  https://www.kisa.or.kr/2060305/form?postSeq=7&lang_type=KO&page=#fnPostAttachDownload

- HID

  HID 글로벌, '2024 보안 현황 보고서
  https://www.hidglobal.com/sites/default/files/documentlibrary/2024-security-trends-eb.pdf

Images / Icons

- Microsoft Designer

TDE

- pg_tde
  https://github.com/percona/pg_tde/
- Benchmark
  https://www.enterprisedb.com/blog/TDE-Postgres-Advanced-Server-15-Launch

PCI DSS

- PCI.org
  https://listings.pcisecuritystandards.org/documents/PCI_DSS-QRG-v3_2_1.pdf

Data Masking

- Anonymizer
  https://postgresql-anonymizer.readthedocs.io/en/stable/

Thanks!

Do you have any questions?

geartec82@gmail.com
+82 10 2508 0800
https://www.linkedin.com/in/geartec82/