PostgreSQLのセキュリティを極める

PGConf.ASIA 2018 Day 2
2018年12月12日

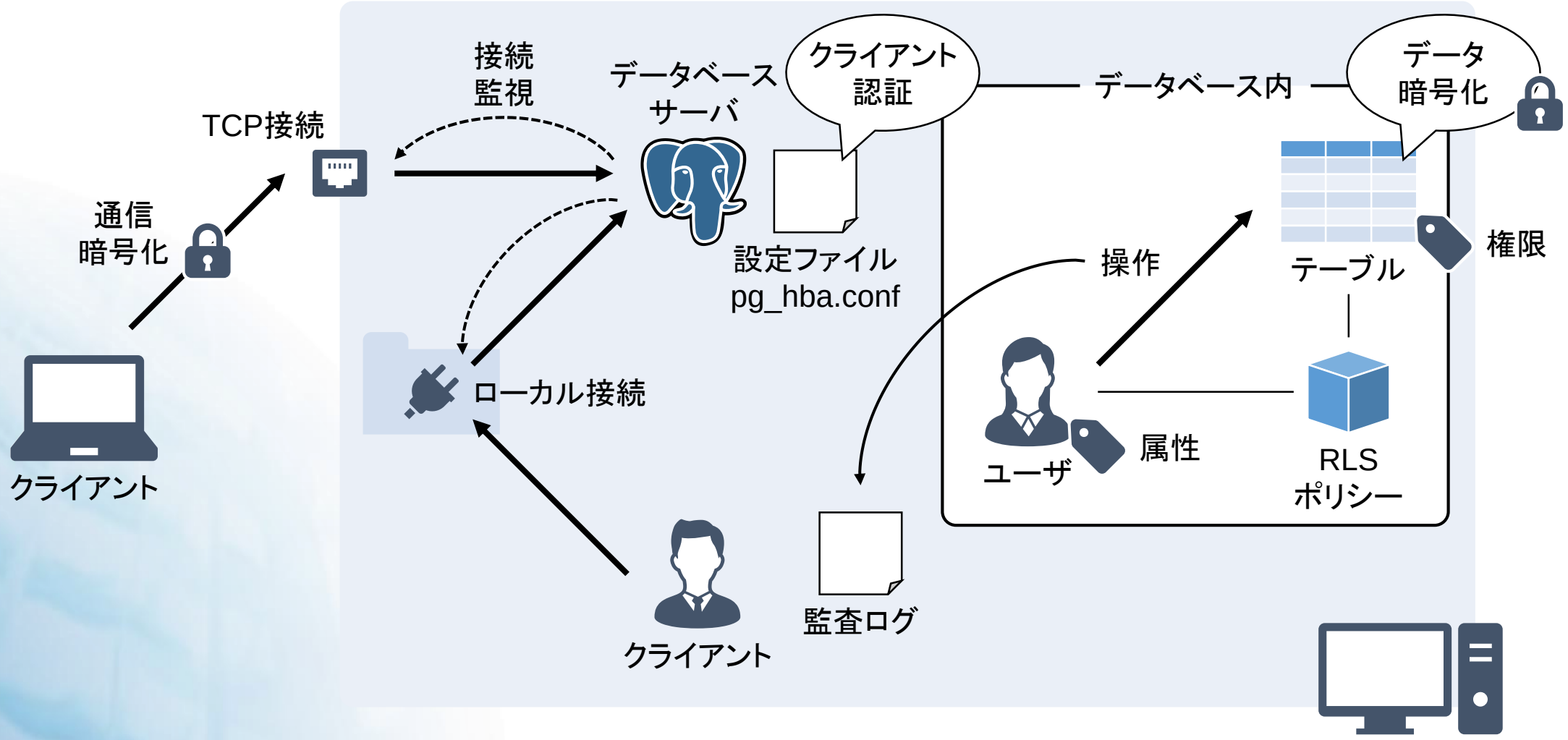
SRA OSS, Inc. 日本支社
佐藤 友章
sato@sraoss.co.jp

目次

- おもなセキュリティの観点
- データベース接続時のセキュリティ
- データベース内のセキュリティ
- そのほかのセキュリティ

おもなセキュリティの観点

- 考慮すべきセキュリティの観点は多岐に渡る



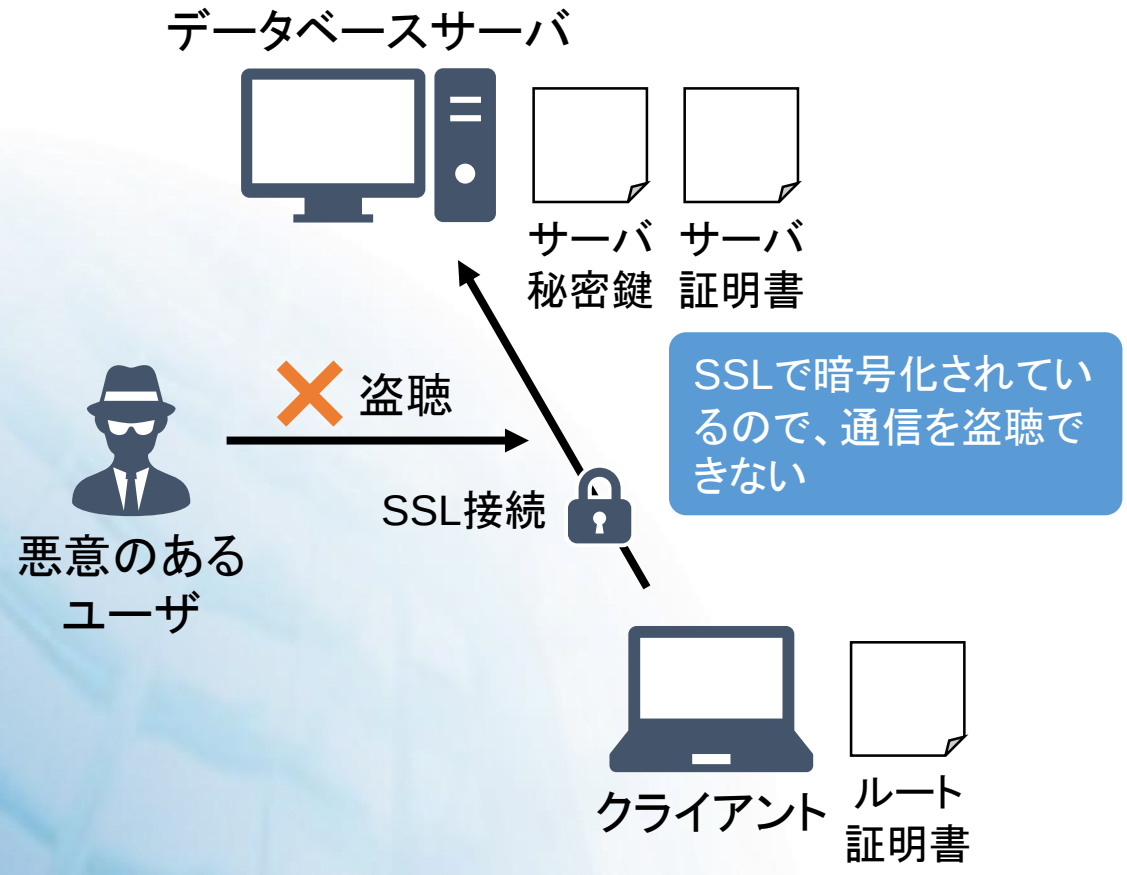
データベース接続時のセキュリティ

通信暗号化

- SSLでクライアントとデータベースサーバ間の通信を暗号化

設定例

- ssl = on　　# SSLによる暗号化を有効に

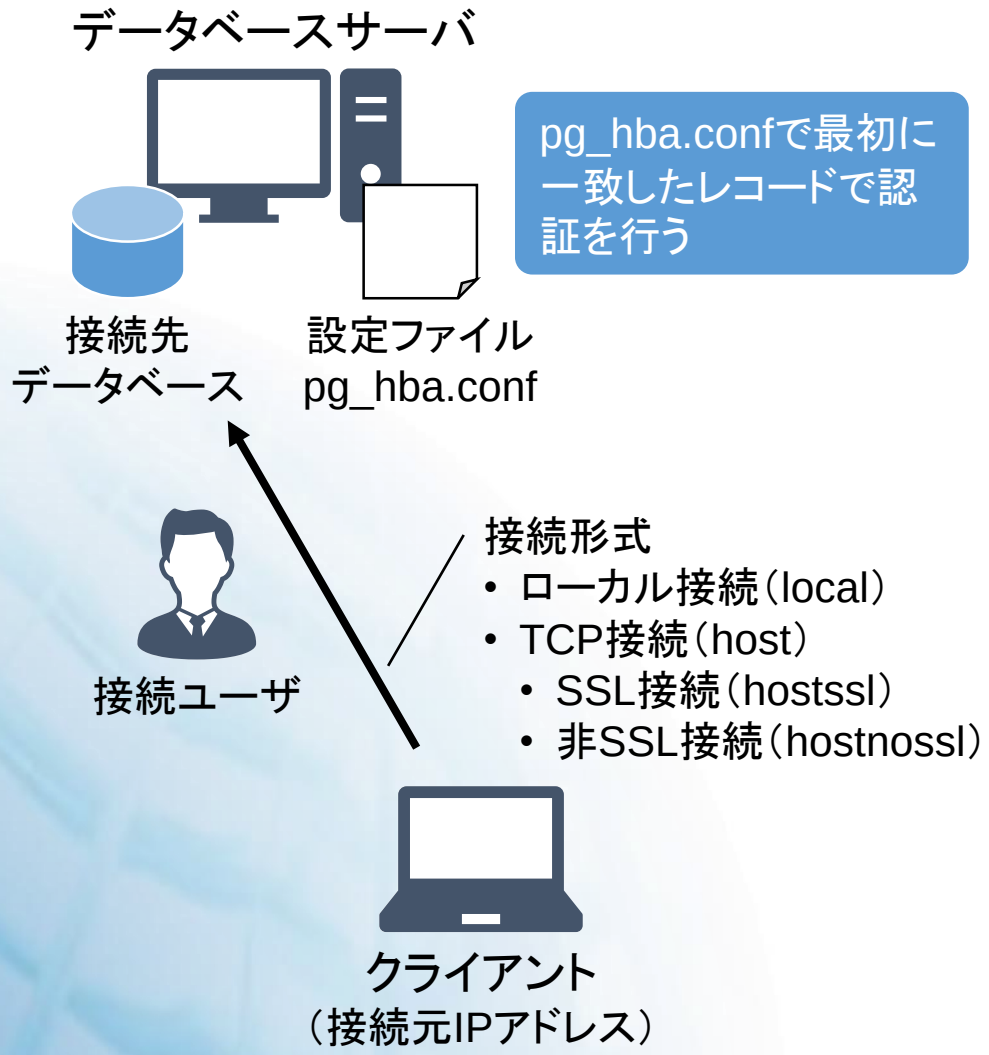


- SSL接続の強制はクライアント認証で設定
- サーバ証明書も検証するには、ルート証明書をクライアントに配置して、接続時にSSLモードを指定
- SSLモード

SSLモード	説明
disable	非SSL接続
allow	非SSL接続を試みて、できなければSSL接続
prefer	SSL接続を試みて、できなければ非SSL接続（デフォルト）
require	SSL接続
verify-ca	SSL接続/サーバ証明書を検証
verify-all	SSL接続/サーバ証明書とホスト名の一致を検証

クライアント認証

- クライアントに対してデータベースサーバ接続時に行う認証



- pg_hba.confの書式

```
local      データベース名 ユーザ名                        認証方式
host       データベース名 ユーザ名 IPアドレス範囲 認証方式
hostssl    データベース名 ユーザ名 IPアドレス範囲 認証方式
hostnossl データベース名 ユーザ名 IPアドレス範囲 認証方式
```

- おもな認証方式

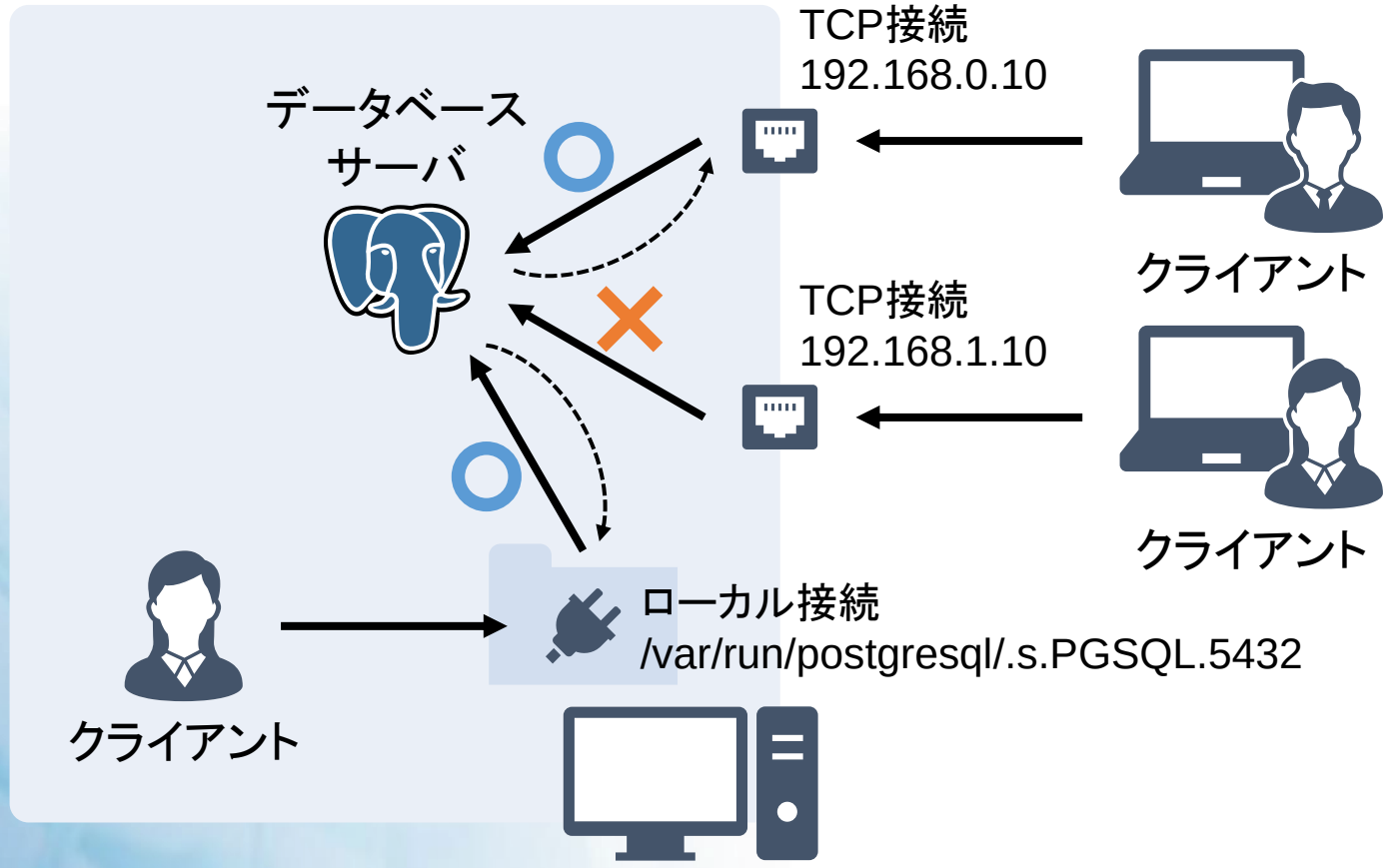
認証方式	説明
trust/reject	無条件で許可/拒否
scram-sha-256	SCRAM暗号化でパスワード認証(10以降)
md5	MD5暗号化でパスワード認証
password	平文でパスワード認証(非推奨)
ident/peer	OSとデータベースユーザ名との一致で認証
cert	SSLクライアント証明書で認証

listen_addressesパラメータ

- クライアントからの接続を監視するサーバのアドレスを指定するパラメータ

設定例

- listen_addresses = 'localhost,129.168.0.10'



- クライアントではなく、サーバのアドレスを指定
- 未指定のアドレスでは接続要求を受けつけない
- デフォルトは「local」でローカルホストのみ、「*」ですべてのアドレスを監視

ネットワークインタフェースが複数ある場合 ＋ リモートホストからの接続を受けつける場合

不要な接続要求を防止するため、必要なアドレスのみを指定すべき

データベース内のセキュリティ

ロールの属性

- ロールに対して設定する権限/パスワード/パラメータ
  - CREATE ROLE/ALTER ROLEで設定
  - パスワードの設定は、平文での送信を回避するため、psqlの¥passwordで行う


スーパーユーザ権限
（SUPERUSER）


データベース作成権限
（CREATEDB）


ロール作成権限
（CREATEROLE）


ログイン権限
（LOGIN）


レプリケーション権限
（REPLICATION）

RLS無視権限（9.5以降）
（BYPASSRLS）


パスワード
（PASSWORD パスワード）

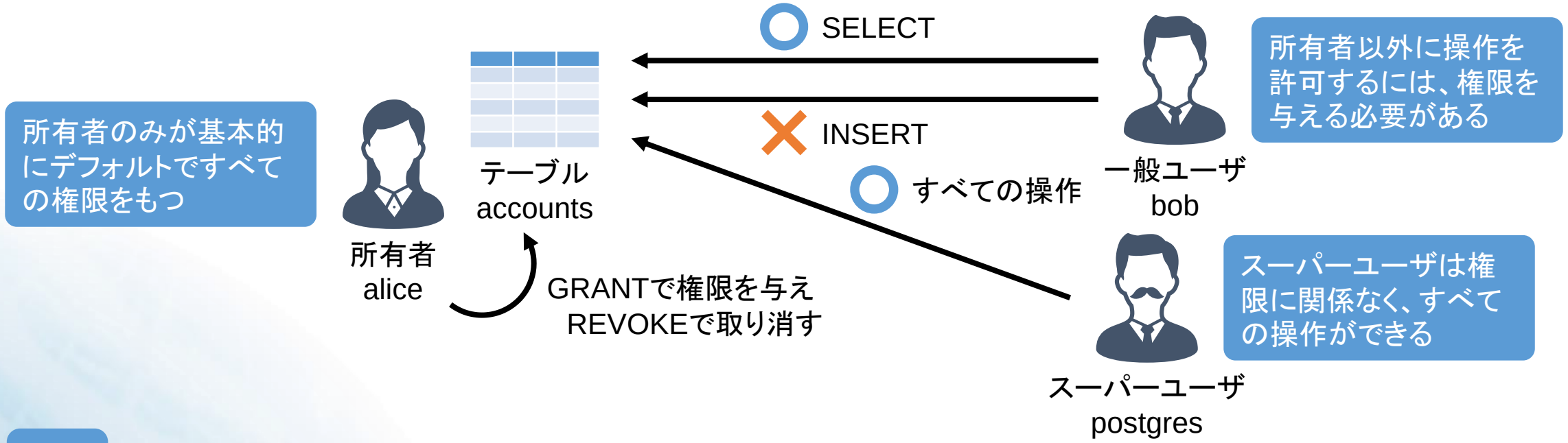

パラメータ
（SET パラメータ名 TO 値）

実行例

```
=# CREATE ROLE alice LOGIN;                    -- ログイン権限をもつロールaliceを作成
=# ALTER ROLE alice CREATEDB;                  -- ロールaliceにデータベース作成権限を与える
=# ¥password alice                             -- ロールaliceのパスワードを設定
=# ALTER ROLE alice SET work_mem TO '8MB';     -- ロールaliceが接続時のwork_memパラメータを8MBに設定
```

データベースオブジェクトの権限

- データベースオブジェクトに対してロールができる操作の権限



実行例

```
=> GRANT SELECT ON TABLE accounts TO bob;   -- テーブルaccountsにロールbobのSELECT権限を与える
=> ¥dp accounts
 Schema |   Name   | Type  |  Access privileges   | Column privileges | Policies
--------+----------+-------+----------------------+-------------------+----------
 public | accounts | table | alice=arwdDxt/alice+|                   |
        |          |       | bob=r/alice          |                   |
```

- aliceはすべての権限(arwdDxt)をもつ
- bobはSELECT権限(r)をもつ
- 権限はaliceによって与えられた

おもなデータベースオブジェクトの権限一覧

データベースオブジェクト	権限	操作
テーブル/列/ビュー/外部テーブル（TABLE）	SELECT（r）	SELECT、COPY TO
	INSERT（a）	INSERT、COPY FROM
	UPDATE（w）	UPDATE、SELECT ... FOR UPDATE/SHARE
	DELETE（d）	DELETE
	TRUNCATE（D）	TRUNCATE
	REFERENCES（x）	外部キー制約作成
	TRIGGER（t）	トリガ作成
シーケンス（SEQUENCE）	USAGE（U）	currval、nextval関数実行
	SELECT（r）	currval関数実行
	UPDATE（w）	nextval、setval関数実行

データベースオブジェクト	権限	操作
データベース（DATABASE）	CREATE（C）	データベース内にスキーマ作成
	CONNECT（c）	データベース接続
	TEMPORARY（T）	データベース内に一時テーブル作成
関数/プロシージャ（FUNCTION）	EXECUTE（X）	関数、プロシージャ、演算子実行
スキーマ（SCHEMA）	CREATE（C）	スキーマ内にオブジェクト作成
	USAGE（U）	スキーマ内のオブジェクトアクセス
テーブル空間（TABLESPACE）	CREATE（C）	テーブル空間内にテーブル、インデックス、一時ファイル作成

データベースのCONNECT、TEMPORARY権限、関数/プロシージャのEXECUTE権限、
手続き言語、データ型/ドメインのUSAGE権限はデフォルトですべてのユーザに与えられる

デフォルトロール

- スーパーユーザのみができる操作の権限を部分的に与えるためのロール（9.6以降）

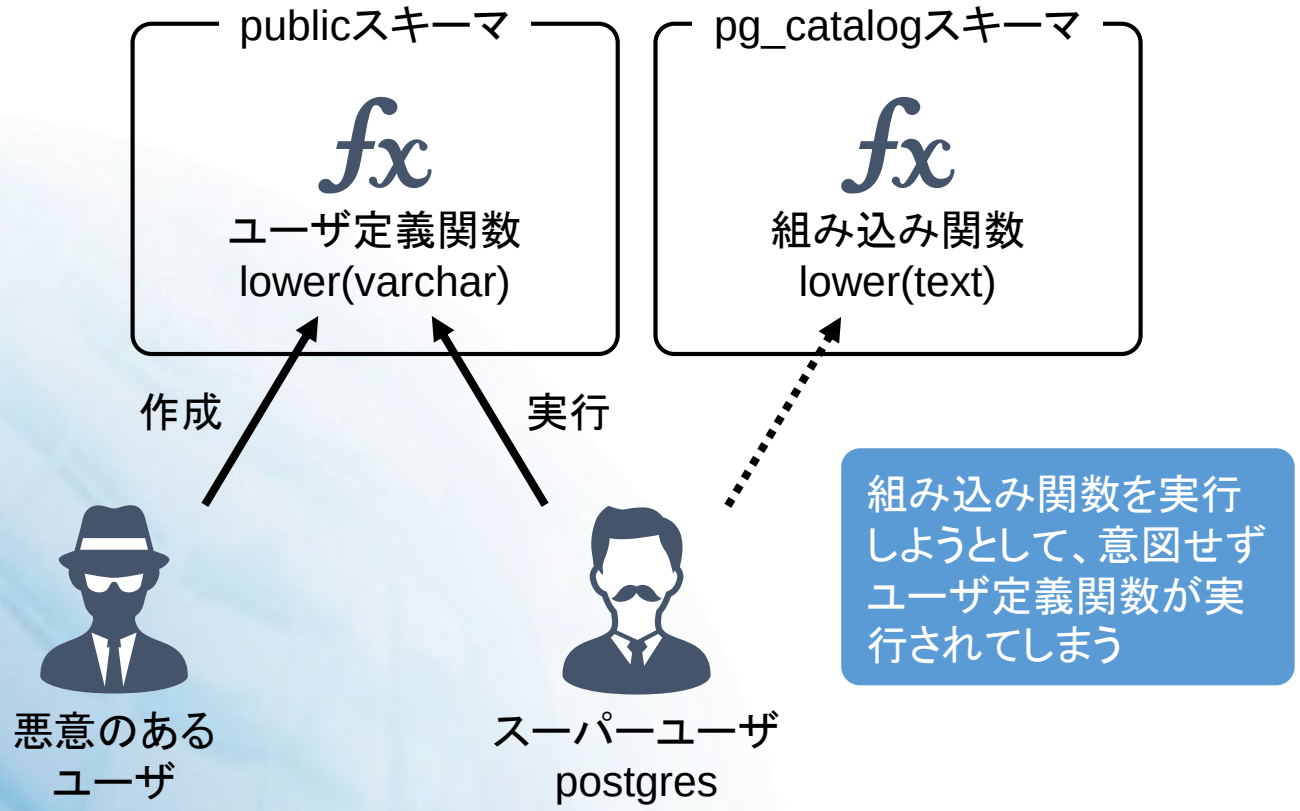
ロール	操作
pg_read_all_settings	すべてのパラメータを参照できる（10以降）
pg_read_all_stats	すべての統計情報を参照できる（10以降）
pg_stat_scan_tables	ACCESS SHAREロックを長時間取得する統計情報関数を実行できる（10以降）
pg_signal_backend	サーバプロセスにシグナルを送信できる
pg_read_server_files	データベースサーバ上のファイルを読み取りできる（11以降）
pg_write_server_files	データベースサーバ上のファイルに書き込みできる（11以降）
pg_execute_server_program	データベースサーバ上のプログラムを実行できる（11以降）
pg_monitor	pg_read_all_settings + pg_read_all_stats + pg_stat_scan_tablesと同じ（10以降）

実行例

```
=# ALTER ROLE pg_monitor TO bob;  -- ロールbobにデフォルトロールpg_monitorの権限を与える
=# ¥du bob
 Role name | Attributes |  Member of
-----------+------------+-------------
 bob       |            | {pg_monitor}
```

publicスキーマ

- デフォルトで存在するスキーマ
  - すべてのユーザがデータベースオブジェクトを作成できる
  - デフォルトで検索パスに含まれる



組み込み関数を実行しようとして、意図せずユーザ定義関数が実行されてしまう

攻撃を回避するには

- publicスキーマに対するすべてのユーザのCREATE権限を取り消す

```
=# REVOKE CREATE
-#     ON SCHEMA public FROM PUBLIC;
```

- スキーマの検索パスからpublicスキーマを取り除く

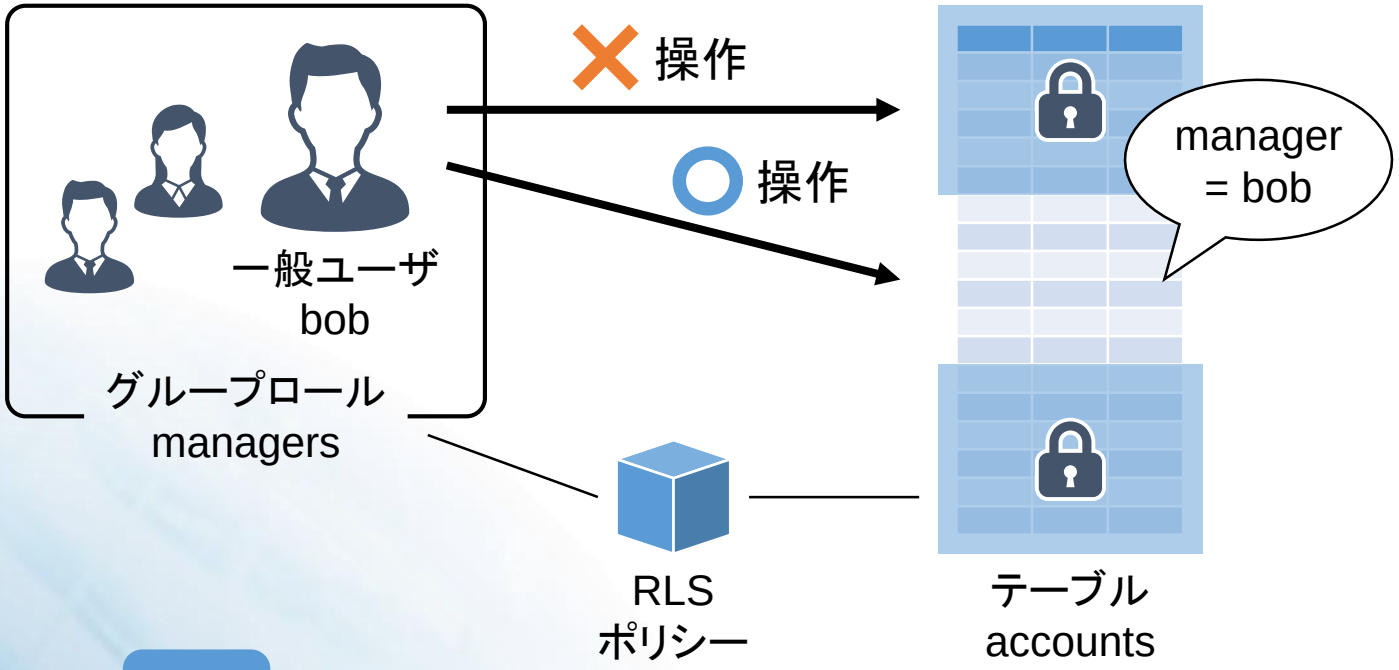
```
=# SET search_path TO '$user';
```

- データベースオブジェクトのスキーマ名を明示的に指定する

```
=# SELECT pg_catalog.lower(email)
-#     FROM accounts;
```

行単位セキュリティ(RLS)

- テーブルに対してロールが操作できる行を制限する仕組み(9.5以降)



- テーブルの権限に追加して設定
- 操作できない行を参照しようとしても、可視できないだけで、エラーにならない
- TRUNCATEなど、テーブル全体への操作は対象外

実行例

```
=# ALTER ROLE managers TO bob;                -- グループロールmanagersにロールbobを追加
=# ALTER TABLE accounts                       -- テーブルaccountsの行単位セキュリティを有効にする
-#     ENABLE ROW LEVEL SECURITY;
=# CREATE POLICY account_managers             -- テーブルaccountsにグループロールmanagersのメンバが
-#     ON accounts TO managers                 -- 自分がマネージャになっている行のみを操作できるように
-#     USING (manager = current_user);         -- RLSポリシーを作成
```

そのほかのセキュリティ

データ暗号化

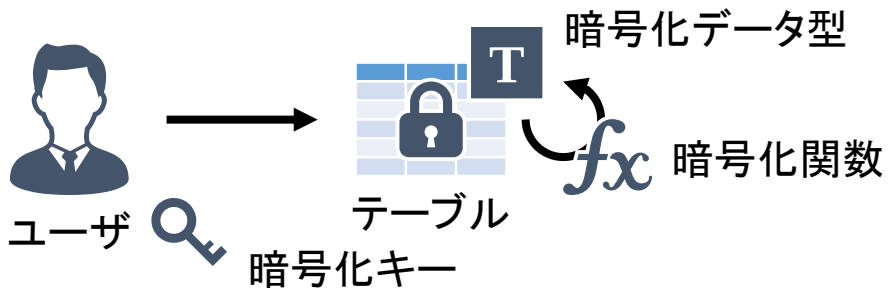
- データベースに格納するデータを暗号化
  - おもなデータ暗号化の実現方式

pgcryptoモジュール

暗号化関数を提供する付属モジュール

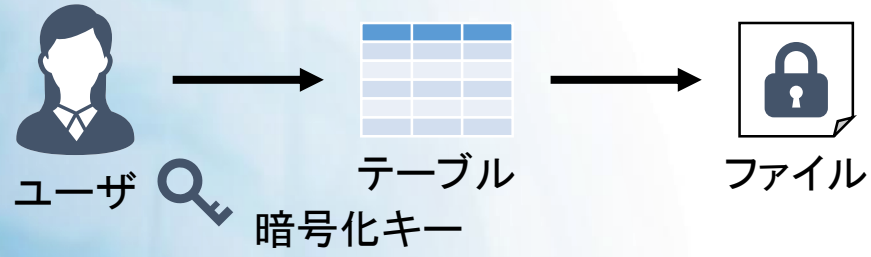
TDE for PG

pgcryptoをベースにTDE機能を提供するモジュール



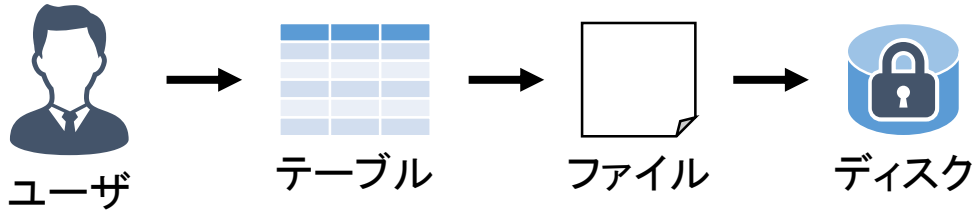
PowerGres Plus

TDE機能を追加したPostgreSQLベースの商用製品



eCryptfs/EncFS/dm-crypt + LUKS

様々な暗号化ファイルシステム

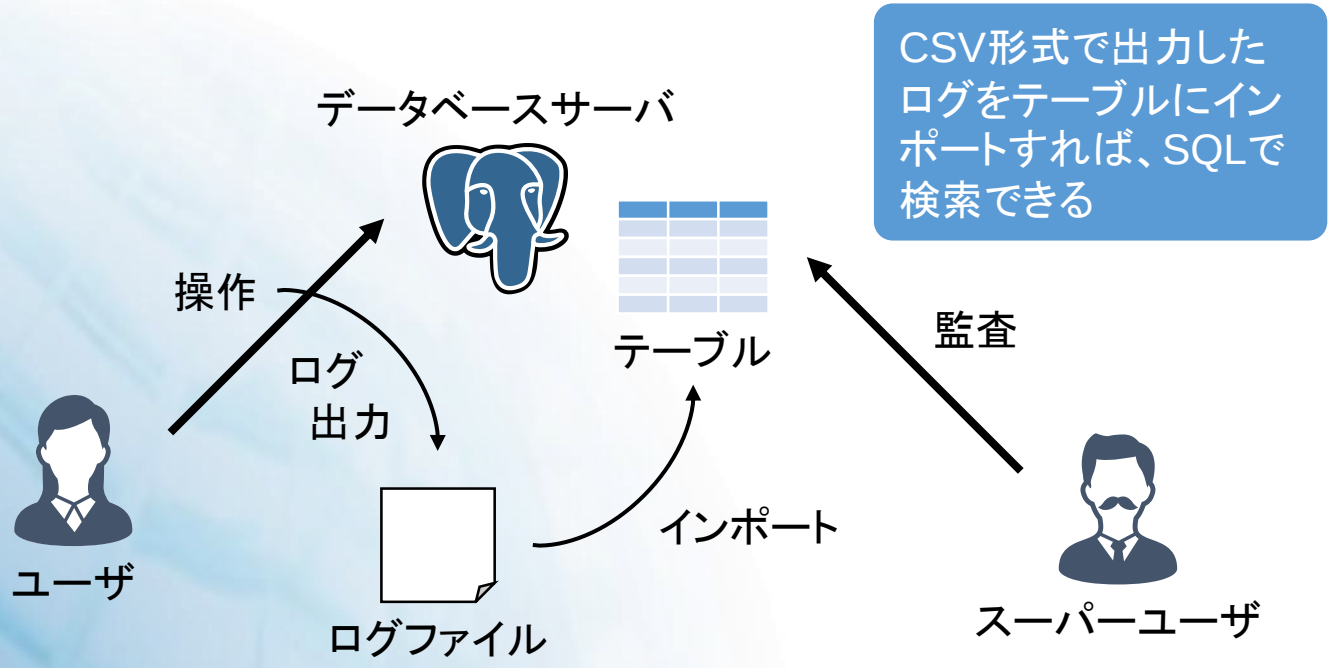


監査ログ

- 不正アクセスの検出のため、データベースの操作を記録したログ

設定例

- log_destination = 'csvlog'　　# CSV形式でログを取得
- log_connections = on　　# 接続のログを記録
- log_disconenctions = on　　# 切断のログを記録
- log_statement = all　　# すべてのSQLをログに記録



CSV形式で出力したログをテーブルにインポートすれば、SQLで検索できる

- OSの機能と組み合わせれば、ログの設定でもある程度実現できる
- 監査ログ専用の機能ではないため、要件によってはpgaduditモジュールの導入が必要

pgauditを使用すれば

- 取得対象のSQLの詳細な種類や、テーブル/列を指定できる
- 操作対象の完全なテーブル名や、SQLのパラメータを取得できる

オープンソースとともに

SRA OSS, INC.