暗号化オプション関連

特定の列に関する暗号化

●ポスグレは透過的に暗号化する機能がないため、contrib モジュールの pgcrypto を使用する。しかし、本モジュールも透過的暗号化ではなく、暗号化する列をユーザが決めて、SELECT、INSERT、UPDATE 等の SQL 文に対して、データを暗号化・復号する関数を明示的に呼ぶ方法をなる。

●本モジュールは、DB の CREATE 権限があれば、スーパーユーザではなくてもインストール可能

●セキュリティ上の問題は以下のとおり

①本モジュールは、対象列を DB サーバ内に暗号化して保存することが目的であり、本モジュールの関数はサーバ内で実行されるため、クライアント・サーバ間では平文である。よって、ローカル又は SSL による接続が必要。

②また、クライアントの暗号化キー等がサーバ内に存在する時間もあるため、DB 管理者による搾取もあり得る。根本的な解決法は、本モジュールを使用せず、クライアント側で単純に暗号化及び複合するしかない。

●使用する主な関数

①汎用ハッシュ関数：標準アルゴリズムは md5 から sha512 まであり、digest()関数と hmac()関数がある。

②パスワードハッシュ化関数：gen_salt()関数で PW に salt を加え、crypt()関数でハッシュ化する。

③PGP 暗号化関数：OpenPGP の対象鍵及び公開鍵暗号化をサポート。

PW の暗号化（ハッシュ化）

●CREATE ROLE（デフォルトで NOLOGIN）又は CREATE USER（デフォルトで LOGIN）コマンドの実行は、スーパーユーザ又は CREATEROLE 権限が必要であるが、一般ユーザは自信の PW を ALTER ROLE 等で変更できる。

●PW を設定又は変更する際、平文で入力された PW は password_encryption パラメータに基づきハッシュ化されて、global フォルダ内の pg_authid カタログ内に保存される。

●上記パラメータのデフォルトは、scram-sha-256（scram：salted challenge response authentication mechanism）であり、ソルトとハッシュ反復回数等によって、従前の md5（128bit）より強化されている。双方ともチャレンジ/レスポンス方式である。

●pg_hba.conf で設定（変更の反映は再起動又はリロード）できるクライアントから接続する時の認証方式であるパスワード認証は、md5、scram-sha-256 及び password（平文）の 3 通りある。ここで、password_encryption が scram-sha-256 ならば、pg_authid にも scram-sha-256 形式で PW が保存されているため、pg_hba.conf に scram-sha-256 又は password を設定できる。同様に password_encryption が md5 ならば、md5 又は password を設定できる。

●上記において、password 以外は PW をクライアントでハッシュ化（どの段階でハッシュ化するのかは不明）するため平文は流れない。

●ここで、password_encryption が scram-sha-256 の場合、pg_hba.conf に md5 を設定しても、安全性の高い scram-sha-256 に自動的に変更される。

●認証方式をパスワード認証にしているにもかかわらず、PW を設定していない場合には pg_authid に格納されている PW のハッシュ値は NULL になるため、認証は必ず失敗する。

ネットワーク越しの暗号化

●先のパスワード認証は、pg_hba.conf における認証方式に関するものであったが、接続方式として、local、host、hostssl、hostnossl、hostgssenc、hostnogssenc 等がある。

※postgresql.conf の listen_address と関係が深いので注意（listen_address が空白なら local 等）

●接続方式が host の場合、TCP/IP 接続であれば、ssl や gassenc の有無は問わないが、他の 4 つ（hostssl、hostnossl、hostgssenc、hostnogssenc）は有無を問う。

●hostssl を使用する場合には、クライアント・サーバで SSL 体制の構築と postgresql.conf の ssl パラメータ（デフォは off）を on にする必要がある。

●hostgssenc を使用する場合には、クライアント・サーバで GSSAPI 体制の構築が必要である。postgresql.conf の設定は不要である代わりに pg_ident.conf の設定が必要（ident 認証も同様）。また、GSSAPI の詳細は省くが、ケルベロスやシングルサインオンが絡んでいる。

データパーティションに関する暗号化

●ストレージの暗号化については、Linux の機能を用いて、ファイルシステム（FS）又はブロックレベルで行うことができる。これによって、ディスクから平文が読み取られることはない。

●ただし、FS がマウントされている、つまり、FS をディレクトリ（マウントポイント）に接続し、システムで使用可能にしている状態でシステムを含めて盗難にあった場合、複合化キーが発見された時は複合化される。