日日是Oracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2021年8月16日月曜日

データベース・セキュリティの活用(10) - 透過的機密データ保護

Data Redactionおよび仮想プライベート・データベースによる列の保護のために、透過的機密データ保護(Transparent Sensitive Data Protection)を構成することができます。

Data Redactionの記事にて行った伏字処理を、TSDPにて構成してみます。**データベース・アクション**にユーザー**ADMIN**にて接続し、構成します。

保護するデータの種類を定義します。伏字処理の対象は表HR.EMPの列SALですが、この列の**機密タイプ**を**salary_type**と定義し、伏字処理はこのsalary_typeに対して構成します。

プロシージャDBMS_TSDP_MANAGE.ADD_SENSITIVE_TYPEを呼び出します。

```
BEGIN
    DBMS_TSDP_MANAGE.ADD_SENSITIVE_TYPE(
        sensitive_type  => 'salary_type'
      , user_comment    => 'Type for salary columns using a number data type');
END;
/
```

seminar210825-tdsp_salary_type.sql hosted with ❤ by GitHub view raw

作成された機密タイプはビューDBA_SENSITIVE_COLUMN_TYPESより確認できます。

作成した機密タイプに表HR.EMPの列SALを登録します。プロシージャDBMS_TSDP_MANAGE.ADD_SENSITIVE_COLUMNを呼び出します。

```
BEGIN
    DBMS_TSDP_MANAGE.ADD_SENSITIVE_COLUMN(
        schema_name     => 'HR'
      , table_name      => 'EMP'
      , column_name     => 'SAL'
      , sensitive_type => 'salary_type'
      , user_comment    => 'Sensitive column addition of salary_type'
    );
END;
/
```

seminar210825-add_sensitive_column.sql hosted with ❤ by GitHub view raw

TSDPの**ポリシー**として**redact_sal**を作成します。プロシージャDBMS_TSDP_PROTECT.ADD_POLICYを呼び出します。

Data Redactionの設定と同様に、**function_type**として**DBMS_REDACT.PARTIAL**、**function_parameters**として**9,1,3**、**expression**に**1=1**を指定しています。

```
DECLARE
    redact_feature_options DBMS_TSDP_PROTECT.FEATURE_OPTIONS;
    policy_conditions DBMS_TSDP_PROTECT.POLICY_CONDITIONS;
BEGIN
    redact_feature_options ('expression') := '1=1';
    redact_feature_options ('function_type') := 'DBMS_REDACT.PARTIAL';
    redact_feature_options ('function_parameters') := '9,1,3';
    policy_conditions(DBMS_TSDP_PROTECT.DATATYPE) := 'NUMBER';
    policy_conditions(DBMS_TSDP_PROTECT.LENGTH) := '16';
    DBMS_TSDP_PROTECT.ADD_POLICY (
        'redact_sal'
      , DBMS_TSDP_PROTECT.REDACT
      , redact_feature_options
      , policy_conditions
    );
END;
/
```

seminar210825-tdsp_protect.sql hosted with ❤️ by GitHub view raw

作成したポリシーについては、ビューDBA_TSDP_POLICY_CONDITION、DBA_TSDP_POLICY_FEATURE、DBA_TSDP_POLICY_PARAMETERより確認できます。

作成した機密タイプ**salary_type**とポリシー**redact_sal**を関連付けます。プロシージャDBMS_TSDP_PROTECT.ASSOCIATE_POLICYを呼び出します。

```
BEGIN
    DBMS_TSDP_PROTECT.ASSOCIATE_POLICY(
        policy_name     => 'redact_sal'
      , sensitive_type  => 'salary_type'
      , associate       => true
    );
END;
/
```

seminar200825-associate_policy.sql hosted with ❤️ by GitHub view raw

機密タイプsalary_typeに登録されている全ての列について、ポリシーに従った(この場合Data Redactionによる伏字処理)保護を有効にします。プロシージャDBMS_TSDP_PROTECT.ENABLE_PROTECTION_TYPEを呼び出します。

```
BEGIN
```

```
    DBMS_TSDP_PROTECT.ENABLE_PROTECTION_TYPE(
        sensitive_type          => 'salary_type'
    );
END;
/
```

seminar200825-enable_protection_type.sql hosted with ❤ by GitHub view raw

以上で透過的機密データ保護によるData Redactionの設定が完了しました。

設定内容はビューDBA_TSDP_POLICY_PROTECTIONより確認できます。

テスト用のAPEXアプリケーションを実行します。

従業員名に以下を入力し、TSDPによる伏字処理を確認します。

SCOTT' or '1' = '1

列Salに伏字処理が適用されていることが確認できます。

TSDPの無効化を行います。プロシージャDBMS_TSDP_PROTECT.DISABLE_PROTECTION_TYPEを呼び出します。

```
BEGIN
    DBMS_TSDP_PROTECT.DISABLE_PROTECTION_TYPE(
        sensitive_type          => 'salary_type'
    );
END;
/
```

seminar210825-disable_protection.sql hosted with ❤ by GitHub view raw

TSDPのポリシーを削除します。プロシージャDBMS_TSDP_PROTECT.DROP_POLICYを呼び出します。

機密タイプから列の登録を削除には、プロシージャ
DBMS_TSDP_MANAGE.DROP_SENSITIVE_COLUMN、機密タイプの削除には
DBMS_TSDP_MANAGE.DROP_SENSITIVE_TYPEを呼び出します。

```
BEGIN
    DBMS_TSDP_MANAGE.DROP_SENSITIVE_COLUMN(
        schema_name         => 'HR'
      , table_name          => 'EMP'
      , column_name         => 'SAL'
    );
    DBMS_TSDP_MANAGE.DROP_SENSITIVE_TYPE(
        sensitive_type => 'salary_type'
    );
    DBMS_TSDP_PROTECT.DROP_POLICY(
        policy_name => 'redact_sal'
    );
END;
/
```

seminar200825-drop_tsdp_policy.sql hosted with ❤️ by GitHub view raw

Data Redactionの他に、仮想プライベート・データベース、統合監査、ファイングレイン監査のポリシーについても透過的機密データ保護(TSDP)ポリシーを使用することができます。

透過的機密データ保護はアプリケーション・データ・モデルを作成して機密列をリストアップする機能など、オラクルの管理ツールであるOracle Enterprise Managerの利用を前提としている部分があります。Enterprise Managerを使用しない場合は、列の一括操作などができないため、機能ごとにポリシーを設定しても大きな違いはないと感じます。

続く

Yuji N. 時刻: 17:51

共有

‹ ホーム ›

ウェブ バージョンを表示

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。
こちらの記事につきましては、免責事項の参照をお願いいたします。

詳細プロフィールを表示

Powered by Blogger.