

セキュアな外部パスワード・ストア

Oracleホワイト・ペーパー

2008年11月

セキュアな外部パスワード・ストア

はじめに	3
TNSNAMES.ORAの変更	3
ウォレットの場所の設定	4
ウォレットの作成	4
資格証明の保存	5
ウォレットの資格証明のテスト	5
コマンドライン・プロキシ	6
SSLでのパスワード・ストアの使用	7

セキュアな外部パスワード・ストア

はじめに

このホワイト・ペーパーでは、Oracle Database 10g Release 2で初めて導入された *Oracle Database* のセキュアな外部パスワード・ストア機能の使用方法について段階的に説明します。この機能は、Oracle Advanced Security オプションがなくても使用できます。ユーザー・インタラクションのないバッチ・プロセスやほかのタスクを実行するため、セキュアな外部パスワード・ストアでは、Oracleウォレットを使用して、1つ以上のユーザー名/パスワードの組合せを保存します。ウォレットは、3DESアルゴリズムを使用して暗号化されます。セキュアな外部パスワード・ストアによって、データベース接続用のパスワード資格証明に依存する大規模な配備が簡素化されます。パスワード・ストアを検討する最適な方法は、3つの列 (TNSALIAS、USERNAME、PASSWORD) を使用した表を仮定することです。基本的に、TNSALIASは、単一のユーザー名/パスワードの組合せにマップされる主キーです。そのため、ほとんどの配備シナリオにおいて、保管する資格証明ごとに新しいTNSALIASエントリを作成することになります。

ここでは、'rep_tool'ユーザーとしてデータベースに接続し、夜間レポートを作成するレポート・ツールの例を検討します。このアカウントのパスワードをレポート・ツールの複数の開発者で共有しないようにするために、DSA (データベース・セキュリティ管理者) は資格証明をセキュアな外部パスワード・ストアに保存することにします。この結果、明確なテキストによるパスワードがアプリケーションのソース・コード内にハードコードされなくなるため、リスクがさらに軽減されます。また、ユーザー名やパスワードが変更されたときにアプリケーション・コードを常に変更する必要がなく、パスワード管理ポリシーの適用も容易になります。

TNSNAMES.ORAの変更

例として、\$ORACLE_HOME/network/adminに配置されたtnsnames.ora ファイルを使用します。tnsnames.oraファイル内のORCLというエントリを探します。ORCLエントリを切り取り/貼付けし、コピー先の名前をREP_TOOLに変更するだけです。REP_TOOLは、ウォレット内に保管された資格証明を使用してデータベースに接続する際に使用するエイリアスです。tnsnames.oraファイル内にORCLエントリが見つからない場合は、以下のようにREP_TOOLのエントリを入力します。

```

ORCL =
  (DESCRIPTION = (ADDRESS =
    (PROTOCOL = TCP) (HOST = <hostname>) (PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = orcl))
  )

REP_TOOL =
  (DESCRIPTION = (ADDRESS =
    (PROTOCOL = TCP) (HOST = <hostname>) (PORT = 1521))

    (CONNECT_DATA = (SERVICE_NAME = orcl))
  )

```

以下のコマンドを使用して、この新しいエイリアスをテストします。

```

$ tnsping REP_TOOL

TNS Ping Utility for Linux: Version 11.1.0.6.0 - Production on
21-MAR-2008 10:01:51
Used parameter files:
/opt/oracle/product/11.1/db_1/network/admin/sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL =
TCP) (HOST = <hostname>) (PORT = 1521)) (CONNECT_DATA =
(SERVICE_NAME = orcl)))
OK (80 msec)

```

ウォレットの場所の設定

Oracle Net接続のためにウォレットを使用して資格証明情報をデータベースに渡すには、Oracle Netクライアントでウォレットの検索場所を把握しておく必要があります。この場所は、sqlnet.oraファイル内でWALLET_LOCATION/パラメータとして指定します。次章で作成するウォレットのディレクトリの場所を指定する必要があります。ここでは例として、\$ORACLE_HOME/network/admin/ディレクトリ内にウォレットを作成するため、sqlnet.oraファイルに以下のエントリを追加します。

```

WALLET_LOCATION =
  (SOURCE = (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /opt/oracle/product/11.1/db_1/network/admin)))

SQLNET.WALLET_OVERRIDE = TRUE
SSL_CLIENT_AUTHENTICATION = FALSE

```

この設定によって、sqlplus /@<db_connect_string>という文のすべてで、指定した場所にあるウォレット内の情報を使用してデータベースへの認証が行われるようになります。

ウォレットの作成

以下の構文を使用してウォレットを作成します。

```
mkstore -wrl <wallet_location> -create
```

ウォレットが格納されるディレクトリに移動し、ディレクトリとファイルの権限が適切に設定されていることを確認します（構文内で現在の作業ディレクトリを指定するには、"."を使用します）。

```
$ mkstore -wrl .-create
```

```
Enter password:
```

```
Enter password again:
```

ウォレットが以下のように作成されました。

```
-rw----- 1 oracle oinstall 7340 Mar 21 10:15 cwallet.sso
```

```
-rw----- 1 oracle oinstall 7312 Mar 21 10:15 ewallet.p12
```

資格証明の保存

以下の構文を使用して、データベース接続用の資格証明を作成し、ウォレット内に保存します。

```
mkstore -wrl <wallet_location> -createCredential
```

```
<db_connect_string> <username> <password>
```

```
$ mkstore -wrl .-createCredential rep_tool rep_tool
```

```
rep_tool_password
```

```
Enter password:
```

```
Create credential oracle.security.client.connect_string1
```

ウォレットの資格証明のテスト

以下は、"/"を使用したデータベースへの接続例です。

```
$ sqlplus /@rep_tool
```

```
SQL*Plus:Release 11.1.0.6.0 - Production on Fri Mar 21 12:34:22  
2008
```

```
Copyright © 1982, 2008, Oracle.All rights reserved
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 -  
Production
```

```
With the Partitioning, Oracle Label Security, OLAP, Data Mining  
and Real Application Testing options
```

```
REP_TOOL>
```

SQL>プロンプトを現在のユーザー名に置き換えるには、

\$ORACLE_HOME/sqlplus/admin/glogin.sqlを編集し、そこに

```
set sqlprompt '_user>'を追加します。
```

sqlplus /@<db_connect_string>では、単純にウォレットを使用して、一致するTNSALIASに保管されているユーザー名とパスワードが検索され、これらの情報がデータベースに渡されます。

この方法は一見すると、あるセキュリティ・ホールにつながるように思えます。“ユーザー名/パスワードが保管された/@<db_connect_string>を使用すれば誰でも接続できるのではないだろうか”という疑問が湧いてきます。この疑問への回答は、ウォレットの保管場所と、その保管場所に対する権限の所有者にあります。顧客がウォレットを使用して、CRONなどのシステム・スケジューラによって処理を実行するような場合は、ウォレットおよびウォレットのディレクトリへの権限を最小限に抑えるように慎重に検討する必要があります。ウォレットの保存先を限定し、信頼できるクライアントからバッチタイプの“cron”駆動の処理のみを実行することで、完全自動処理のために必要となる資格証明がこの機能によって保護されます。

コマンドライン・プロキシ

セキュアな外部パスワード・ストアの別の例として、次のようなシナリオを取り上げます。バックエンド・サーバーで実行されるルーチン・バッチ・プログラムでは、夜間にHRアプリケーション・スキーマにアクセスする必要がありますが、新しいセキュリティ・ポリシーではHRアプリケーション・スキーマへの直接的なアクセスが制限されています。この場合、プログラムでアプリケーション所有者以外の資格証明を使用してデータベースへの認証を行い、かつ同じレベルのアクセス権を維持するにはどうすれば良いでしょうか。

解決方法は次のとおりです。セキュアな外部パスワード・ストアでコマンドライン・プロキシを使用するための、このプログラム専用のデータベース・アカウントを作成します。

```
SQL> grant create session to HRPROC identified by
HRPROC_PASSWORD;
```

ユーザーHRを、新しいアカウント経由でアクセスできるように変更します。

```
SQL> alter user HR grant connect through HRPROC;
```

ウォレットとtnsnames.oraファイルを構成します。

tnsnames.oraに以下のエントリを追加します。

```
HRPROC =
  (DESCRIPTION = (ADDRESS =
    (PROTOCOL = TCP)(HOST = <hostname>)(PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = orcl))
  )
```

ウォレットに以下のような別の資格証明のセットを追加します。

```
$ mkstore -wrl .-createCredential HRPROC HRPROC HRPROC_PASSWORD
```

```
Enter password:
```

```
Create credential oracle.security.client.connect_string2
```

これで、コマンドラインで資格証明を指定しなくても、バッチ・プログラムで HRPROC アカウントを使用してデータベースへの認証を行えるようになります。

```
sqlplus /@HRPROC
```

```
HRPROC>
```

HRPROC では HR ユーザー経由でのプロキシが許可されているため、バッチ・プログラムで以下のように使用することもできます。

```
sqlplus [HR]/@HRPROC
```

```
HR>
```

SSLでのパスワード・ストアの使用

アプリケーションですでに SSL を使用して暗号化を行っている場合は、`sqlnet.ora` パラメータの `SQLNET.AUTHENTICATION.AUTHENTICATION_SERVICES` で SSL が指定されて、SSL ウォレットが作成されます。このアプリケーションで (SSL 証明書の代わりに) ウォレットを使用して資格証明を保管し、データベースへの認証を行う場合は、これらの資格証明を SSL ウォレット内に保管する必要があります。SSL 認証の後、`SQLNET.WALLET_OVERRIDE = TRUE` の場合に、ウォレットのユーザー名とパスワードを使用してデータベースへの認証が行われます。`SQLNET.WALLET_OVERRIDE = FALSE` の場合は、SSL 証明書が使用されます。

ORACLE®

セキュアな外部パスワード・ストア
2008年11月

著者：Richard Wark

共著者：Peter Wahl、Paul Needham

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

海外からのお問い合わせ窓口：
電話：+1.650.506.7000
ファクシミリ：+1.650.506.7200
www.oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2008, Oracle. All rights reserved.

本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

Oracleは米国Oracle Corporationおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

Hardware and Software, Engineered to Work Together