BaroPAM 가이드(Oracle)

목차

1. External Procedure

1.1 External Procedure란?

복잡한 수식계산을 Oracle에서 제공하는 기능으로만 충분하지 않을 경우가 있는데, 이럴 경우 C나 JAVA 같은 언어로 복잡한 기능을 작성한 후 Oracle 에서는 파라미터를 넘겨서 해당 결과를 받으면 수행속도의 개선을 가져올 수 있는데, 간단히 말하자면 C언어나 VB, JAVA 등의 언어를 사용하여 SQL에서 구현하기 어렵거나 복잡한 것을 구현한 뒤 SQL에서 호출해서 사용하는 것을 말한다.

1.2 장점 및 단점

External Procedure을 사용하여 얻는 장점으로는 Java나 C의 재 활용성이 우수하다. 반면 External Procedure을 사용으로 발생하는 단점은 Session이 종료되지 않으면 extProc는 Oracle에서 메모리를 관리하는 영역이 아니라 O/S영역이기 때문에 한번 호출될 때마다 해당 Session이 종료되지 않으면 끝까지 살아남게 되어 지속적인 메모리에 남아 있게 되어 O/S의 메모리 부하가 생길 수밖에 없다.

그래서, O/S의 메모리를 최대한 줄일 수 있는 방법은 다음과 같다.

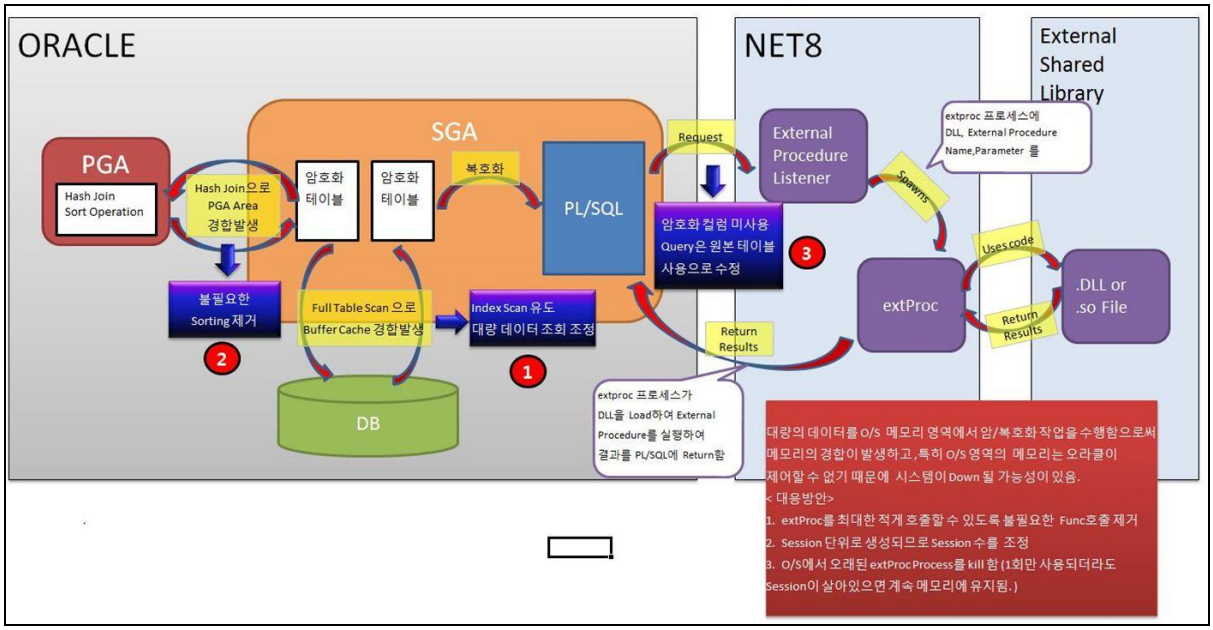
① Session의 수를 제한하여 O/S의 메모리 한계치를 벗어나지 않도록 조정한다.

② O/S에서 extProc로 생성된 것 중 오래된 Process를 Kill한다. Process를 Kill하더라도 없으면 재생성 되므로 큰 문제는 발생되지 않는다.

③ 애플리케이션에서 불필요하게 External Procedure를 호출하는 Function의 사용을 제거하고, 사용 완료 후 Session를 종료하여 메모리의 부하를 최소화하게 한다.

1.3 C 모듈(libbarokey.so)

C 모듈은 Java 모듈과 달리 External Procedure를 사용하는 것이 번거롭지만 External Procedure의 동작 순서는 다음과 같다.



① 사용자가 SQL에서 External Procedure에서 작성한 Function을 DB에 요청을 한다.

② Shared SQL Area에서 해당 문장을 Parsing하면서 External Procedure를 알고 NET8 Listener에서 사용자 SQL이 External Procedure를 호출했으니 해석해 달라고 요청한다.

③ Listener는 다시 extProc 프로세스를 생성하면서 O/S에 있는 External Procedure가 있는 DLL, Procedure Name, Parameter를 넘겨준다.

④ extProc를 O/S에 있는 DLL 파일을 찾아서 O/S의 메모리에 Load하여 요청받은 Function의 결과를 처리한다.

⑤ 처리된 결과를 SQL에 return해 준다.

⑥ Session이 종료되면 extproc도 자동으로 종료된다.

1) C 모듈 library 생성

C 모듈을 Oracle 내에 2차 인증(추가 인증)용 BaroPAM Library(libbarokey_verifykeyp)를 다음과 같이 각각 생성한다.

```
> sqlplus baropam/baropam

SQL*Plus: Release 11.2.0.1.0 Production on 화 9월 12 14:13:40 2017

Copyright (c) 1982, 2009, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE OR REPLACE LIBRARY libbarokey_verifykeyp AS '/usr/baropam/key/libbarokey.so' ;
  2  /
```

```
Library created.

SQL> commit;

Commit complete.
```

참고) Library 생성 중 "ORA-01031: insufficient privileges" 오류가 발생하는 경우 library를 생성하는 계정(baropam)에 library 생성 권한이 없어서 발생한다. 이런 경우 Oracle sysdba로 접속한 다음 library를 생성하는 계정에 library 생성권한을 부여해야 한다.

```
SQL> create library to baropam ;
```

2) 2차 인증 관련 Stored Function 생성 및 확인

Oracle에 sqlplus로 접속한 뒤 TO_VERIFYKEYP(2차 인증 함수)를 다음과 같이 생성 및 확인한다.

```
project:icam /usr/baropam> sqlplus baropam/baropam

SQL*Plus: Release 11.2.0.1.0 Production on 화 9월 12 14:16:06 2017

Copyright (c) 1982, 2009, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE OR REPLACE FUNCTION TO_VERIFYKEYP (secure_key VARCHAR2, cycle_time
VARCHAR2, corr_time VARCHAR2, key_method VARCHAR2) return VARCHAR2
        as external
        language C
        library libbarokey_verifykeyp
        name "generateKEYP"
        parameters  (secure_key STRING, cycle_time STRING, key_method STRING);
  2     3     4     5     6     7   /

Function created.

SQL> commit;

Commit complete.
```

Tibero에 tsql로 접속한 뒤 TO_VERIFYKEYP(2차 인증 함수)를 다음과 같이 생성 및 확인한다.

```
SQL> CREATE OR REPLACE FUNCTION TO_VERIFYKEYP (secure_key VARCHAR2, cycle_time
VARCHAR2, corr_time VARCHAR2, key_method VARCHAR2) return VARCHAR2
        as language C
        library libbarokey_verifykeyp
        name "generateKEYP"
        parameters  (secure_key STRING, cycle_time STRING, key_method STRING);
  2     3     4     5     6     7   /

Function created.
```

```
SQL> commit;

Commit complete.
```

3) listener 구성

C 모듈을 사용하기 전에 원하는 Shared Library에 대한 경로를 포함하도록 listener를 구성해야 한다. EXTPROC_DLLS 환경 변수를 설정하여 수행하는데, Oracle 계정에 로그인 후 다음과 같이 listener.ora 파일에 설정한다.

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /oracle/app/oracle/product/11.2.0)
      (PROGRAM = extproc)
      (ENVS="EXTPROC_DLLS=ANY")
    )
    (SID_DESC =
      (SID_NAME = ODB)
      (ORACLE_HOME = /oracle/app/oracle/product/11.2.0)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = project)(PORT = 1521))
    )
  )

ADR_BASE_LISTENER = /oracle/app/oracle
```

4) tnsnames.ora 구성

C 모듈을 사용하기 전에 listener.ors 파일의 설정한 Key, SID와 동일한 값으로 Oracle 계정에 로그인 후 EXTPROC_CONNECTION_DATA을 다음과 같이 tnsnames.ora 파일에 설정한다.

```
ODB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = project)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ODB)
    )
  )

EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
        (ADDRESS = (PROTOCOL = IPC)(HOST = project)(KEY = EXTPROC1521))
        (CONNECT_DATA = (SID = PLSExtProc))
```

```
)
```

tnsnames.ora 파일을 설정한 후 반드시 Oracle 계정에 로그인 후 listener를 재기동 해야 한다.

```
> lsnrctl start | stop | status
```

5) 2차 인증 함수(TO_VERIFYKEYP) 테스트

```
> sqlplus baropam/baropam

SQL*Plus: Release 11.2.0.1.0 Production on 화 9월 12 14:16:06 2017

Copyright (c) 1982, 2009, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SELECT TO_VERIFYKEYP('jlqlcHbVqdpj7b4PzBpM2DileBvmHFV/', '30', 'app512') FROM DUAL;

TO_ENCRYPTS('jlqlcHbVqdpj7b4PzBpM2DileBvmHFV/', '30', 'app512')
--------------------------------------------------------------------------------
454015
```

2. BaroPAM 설치

2.1 BaroPAM 개요

Oracle용 BaroPAM은 시간 동기화 방식의 일회용 비밀번호(Time-based One-time Password)로 사용자가 Oracle Database 내부에서 무엇이든 실행할 수 있을 때 2차 인증(추가 인증)을 추가로 보안을 강화하는 솔루션이다.

사용 예:

-데이터베이스 사용자를 위한 추가 보안 계층을 추가하여 사용자 암호의 보안을 강화.
-누군가가 스키마 비밀번호를 발견하면 DB 오브젝트에 대한 액세스가 제한되거나 액세스되지 않음.
-10gR2에서 최신 12c까지 모든 Oracle Database 버전 (SE 및 EE)에서 테스트됨.

PCI DSS 요구 사항 8.3

직원, 관리자, 제3자에 의한 네트워크로의 원격 접속에는 Multi-Factor 인증을 적용한다.

참고) PCI DSS(Payment Card Industry Data Security Standard)는 신용, 직불, 현금카드 거래의 보안 최적화, 개인정보 보호를 위해 지불 및 카드 결제 전과정에 필요한 보안 요구 사항을 규정한 국제 데이터 보안 표준으로 Visa, Master Card, Discover, American Express, JCB 등 주요 글로벌 신용 카드 회사에서 공동으로 만든 이 기준은 세계적으로 그 안전성을 인정받아 활용되고 있는 보안 표준임.

1. 작동 원리

사용자가 DBMS에 연결하면 그의 역할이 비활성화되고, 이를 활성화하는 유일한 방법은 BaroPAM 앱을 사용하여 생성된 올바른 일회용 인증키를 입력해야 한다. DBA는 누군가가 2차 인증(추가 인증)을 받기 전에 어떤 역할이 필요한지 쉽게 정의할 수 있다.

Oracle용 BaroPAM은 누구나 쉽게 사용할 수 있도록 설정 및 사용이 매우 쉽다.

일회용 인증키를 입력할 수 있는 응용 프로그램의 인터페이스가 없는 경우인 응용 프로그램, 일괄 작업 등으로 로그인에 사용되는 역할에 대해 2차 인증(추가 인증)을 사용하지 않아야 한다. 또한 내부 Oracle 프로세스에 영향을 줄 수 있으므로 기본 역할(DBA, RESOURCE 등)에서 활성화하지 않는 것이 가장 좋다. 이러한 역할을 다른 이름으로 복제하고 활성화하는 것이 가장 적합하다.

2. 특징

-항상 동일한 응용 프로그램과 컴퓨터에서 연결하는 경우 7일 동안 Oracle용 BaroPAM을 신뢰하도록 Oracle용 BaroPAM에 요청할 수 있다. 따라서 데이터베이스를 정확히 동일한 위치에서 연결하는 경우 일회용 인증키를 다시 입력할 필요가 없다.

-BaroPAM 앱을 정보를 변경하면 2차 인증(추가 인증)의 구성 요소를 재구성할 수 있다.

-일회용 인증키의 무차별 대입 공격으로부터 보호한다.

-사용자는 암호화 시드에 암호를 제공하여 다른 사람(나 자신도)이 되돌릴 수 없도록 할 수 있다.

3. 사용 절차

1) 일회용 인증키 생성기인 BaroPAM 앱(안드로이드용 또는 아이폰용)을 설치한다.

2) 데이터베이스에 Oracle용 BaroPAM을 설치한다.

3) 2차 인증(추가 인증)이 완료된 후에만 사용하도록 기존 ROLE을 작성하거나 대체한다.

4. 생성된 객체

Oracle용 BaroPAM은 시간 동기화 방식의 일회용 비밀번호(Time-based One-time Password) 구성을 갖는 모든 패키지 및 테이블을 보유하기 위해 하나의 새 스키마(설치 중에 정의할 수 있는 이름)를 작성한다. 이 사용자는 잠기고 만료되며 누구도 사용할 수 없게 된다.

생성되는 객체는 다음과 같다.

-1개의 새로운 스키마:
 3개의 패키지와 그 바디
 1개의 절차
 1개의 트리거
 3개의 테이블과 그 제약과 색인

-2 개의 공용 동의어(스키마 이름 입력을 피하기 위해)

-1개의 컨텍스트(2차 인증 제어)

또한, 스키마 테이블을 보호하기 위해서 1개의 기능과 4개의 VPD(Virtual Private Database) 정책을 생성한다.

참고) Oracle에 사용자가 접속하게 되면 하나의 세션이 클라이언트와 Oracle 서버 사이에 생성된다. VPD(Virtual Private Database)는 생성된 세션의 정보를 이용하여 RLS(Row Level Security: 사용자별로 같은 테이블을 조회하더라도 조회할 수 있는 권한을 테이블의 Row별로 보여하는 것)를 가능하게 한다.

Database Vault Option를 사용하면 데이터와 애플리케이션에 액세스할 수 있는 사람, 시간 및 장소를 제어하여 가장 일반적인 보안 위협인 악의적인 내부 사용자로부터 비즈니스를 보호할 수 있다. 관리자를 비롯한 내부 사용자 간에 역할 구분을 적용함으로써 Database Vault는 오늘날의 엄격한 규정 준수 및 개인 정보 보호 요구 사항을 충족하는 데 도움이 되는 강력한 예방 통제 수단으로 사용된다.

즉, Database Vault Option 이 있으면 모든 스키마 객체를 보호하는 영역도 생성된다.

Oracle용 BaroPAM을 사용할 때 알아야 할 객체는 다음과 같다.

1) TWOFACTOR 패키지
   2차 인증과 관련된 모든 작업을 수행하는데 사용되며 PUBLIC에 부여됨.
2) TWOFACTOR_ADMIN 패키지
   2차 인증과 관련된 모든 작업 관리 작업을 수행하는데 사용. DBA에만 부여해야 함.
3) ENABLE_ROLE 프로시저
   2차 인증의 보호 역할을 사용하는데 사용되며 PUBLIC에 부여됨.

2.2 BaroPAM 설치할 디렉토리 생성

BaroPAM 인증 모듈을 다운로드 및 설치하기 위해서 root 계정으로 접속한 후 모듈을 다운로드 및 설치하기 위한 디렉토리(/usr/baropam)를 다음과 같이 생성한다.

```
[root]# mkdir /usr/baropam
```

BaroPAM 모듈을 다운로드 및 설치하기 위한 디렉토리의 권한(읽기, 쓰기, 실행)을 다음과 같이 부여한다.

```
[root]# chmod -R 777 /usr/baropam
```

2.3 BaroPAM 설치할 모듈 다운로드

Oracle용 BaroPAM 인증 모듈은 root 계정으로 접속한 후 모듈을 다운로드 및 설치하기 위한 디렉토리 (/usr/baropam)로 이동하여 모듈을 다운로드 하는 방법은 다음과 같다.

```
[root] /usr/baropam > wget http://nuriapp.com/download/libpam_baro_ora-x.x.tar
```

Oracle용 BaroPAM 인증 모듈의 다운로드가 완료되면 tar 파일의 압축을 해제하는 방법은 다음과 같다.

```
[root] /usr/baropam > tar -xvf libpam_baro_ora-x.x.tar
```

Oracle용 BaroPAM 인증 모듈의 압축을 해제하면 baropam 디렉토리에 다음과 같은 Oracle용 BaroPAM 관련 모듈이 생성된다.

```
[root] /usr/baropam > ls -al
total 384
drwxr-xr-x  5 root root     78 Jan  6 10:18 .
drwxrwxrwx. 8 root root   4096 Jan  6 10:17 ..
drwxr-xr-x  2 root root     25 Mar 18  2020 jilee
drwxr-xr-x  2 root root    286 Oct 23 15:05 key
-rw-r--r--  1 root root 389120 Jan  6 10:18 libpam_baro_ora-8.1.1-x64.tar
drwxr-xr-x  3 root root     36 Jan  6 10:15 ora
```

2.4 BaroPAM 모듈 설치

Oracle용 BaroPAM을 설치하기 위해서는 oracle 계정으로 접속한 후 Oracle용 BaroPAM 인증 모듈이 존재하는 디렉토리(/usr/baropam/ora)로 이동한 후 다음과 같은 명령어를 실행하여 설치 작업을 진행하도록 한다.

```
[oracle] /usr/baropam/ora > sqlplus /nolog

SQL*Plus: Release 11.2.0.1.0 Production on Sat Jan 2 10:21:23 2021

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

SQL> @INSTALL
Schema Name for 2-Factor [TOTP]: TOTP
String to connect as SYS [/ as sysdba]: / as sysdba
Connected.
DB Vault Users script skipped - Database Vault not enabled.
Connected.
User created.
Connected.
```

```
User privs granted.
Connected.
User objs create.
BaroPAM create.
twofactor internal create.
twofactor admin create.
twofactor create.
enable role.
grant synonym.
Objects created.
Policies created.
Connected.
DB Vault Realms script skipped - Database Vault not enabled.
=> SCRIPT EXECUTED SUCCESSFULLY! <=
Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

[oracle] /usr/baropam/ora >
```

"SCRIPT EXECUTED SUCCESSFULLY!" 메시지가 표시되면 정상적으로 설치 작업이 진행된 상태이다. 설치 작업 중 오류가 발생하면 설치 작업이 중단된다.

1. TWOFACTOR 패키지

생성된 **TWOFACTOR 패키지**는 모든 DB 사용자에게 부여되어야 하며, 사용자가 **2차 인증**을 설정하고 인증할 수 있도록 한다. 그 이후에는 **2차 인증**으로 보호되는 역할을 활성화할 수 있다.

1) SETUP(SECURE_KEY IN VARCHAR2, CYCLE_TIME IN VARCHAR2, KEY_METHOD IN VARCHAR2);

현재 로그인한 사용자를 설정하고 **일회용 인증키**를 생성하기 위한 항목을 생성하는 함수. 이것은 누구나 해야 하는 첫 번째 단계다. 실행하기 전에 SERVEROUTPUT을 활성화해야 한다.

매개변수	설명	설정값	비고
SECURE_KEY	반드시 벤더에서 제공하는 Secure key(라이선스 키)	jlqlcHbVqdpj7b4PzBpM2DileBvmHFV/	
CYCLE_TIME	**일회용 인증키**의 인증주기(초, 3~60초)	30	
KEY_METHOD	**일회용 인증키**의 인증방식(app1, app256, app384, app512: 앱)	app512	

사용 예)

```
SQL> set serveroutput on
SQL> exec twofactor.setup('jlqlcHbVqdpj7b4PzBpM2DileBvmHFV/','30','app512');

PL/SQL procedure successfully completed.

SQL>
```

2) VALIDATE(PCODE IN VARCHAR2);

사용자가 SETUP이면 **BaroPAM** 앱에서 생성한 **일회용 인증키**를 제공하여 유효성을 검사하는 함수. 유효성 검사는 **BaroPAM** 앱을 올바르게 설정했으며 생성 중인 **일회용 인증키**가 유효 함을 증명한다. 사용자가 **2차 인증**으로 보호되는 역할을 인증하고 활성화할 수 있는지 확인한 후에만 가능하다.

매개변수	설명	설정값	비고
PCODE	BaroPAM 앱에서 생성한 일회용 인증키		

사용 예) - 검증 성공

```
SQL> set serveroutput on
SQL> exec twofactor.validate(988208);

PL/SQL procedure successfully completed.

SQL>
```

사용 예) - 검증 실패

```
SQL> set serveroutput on
SQL> exec twofactor.validate(123456);
BEGIN twofactor.validate(123456); END;

*
ERROR at line 1:
ORA-20000: Code not valid.
ORA-06512: at "TOTP.TWOFACTOR_ADMIN", line 89
ORA-06512: at "TOTP.TWOFACTOR", line 35
ORA-06512: at line 1

SQL>
```

3) AUTHENTICATE(PCODE IN VARCHAR2);

새로 로그인 할 때마다 권한이 있는 역할을 활성화하고, 활성화 할 수 있도록 인증하는 함수.

매개변수	설명	설정값	비고
PCODE	BaroPAM 앱에서 생성한 일회용 인증키		

사용 예) - 인증 성공

```
SQL> set serveroutput on
SQL> exec twofactor.authenticate(691199)

PL/SQL procedure successfully completed.

SQL>
```

사용 예) - 인증 실패

```
SQL> set serveroutput on
SQL> exec twofactor.authenticate(499775);
BEGIN twofactor.authenticate(499775); END;

*
ERROR at line 1:
ORA-20000: Code not valid.
```

```
ORA-06512: at "TOTP.TWOFACTOR_ADMIN", line 108
ORA-06512: at "TOTP.TWOFACTOR", line 40
ORA-06512: at line 1

SQL>
```

4) DECONFIG (PCODE IN VARCHAR2 DEFAULT NULL);

설정된 2차 인증의 구성을 취소하고 정리하는 함수.

매개변수	설명	설정값	비고
PCODE	BaroPAM 앱에서 생성한 일회용 인증키		

사용 예)

```
SQL> set serveroutput on
SQL> exec twofactor.deconfig(308068);

PL/SQL procedure successfully completed.

SQL>
```

5) REMEMBER(PCODE IN VARCHAR2);

7 일 동안 컴퓨터, 응용 프로그램 및 2차 인증을 기억하는 함수. 이 시간 동안 새 연결을 설정하면 자동으로 인증된다. 역할을 활성화하기만 하면 된다.

매개변수	설명	설정값	비고
PCODE	BaroPAM 앱에서 생성한 일회용 인증키		

사용 예) - 인증 성공

```
SQL> set serveroutput on
SQL> exec twofactor.remember(241575);

PL/SQL procedure successfully completed.

SQL>
```

6) GETSECUREKEY();

설정된 일회용 인증키를 생성하기 위한 SECURE_KEY, CYCLE_TIME, CORR_TIME, KEY_METHOD 정보를 얻는 함수.

사용 예)

```
SQL> set serveroutput on
SQL> exec twofactor.getsecurekey();
Secure key: jlqlcHbVqdpj7b4PzBpM2DileBvmHFV/-30-0-app512

PL/SQL procedure successfully completed.

SQL>
```

7) FORGET();

사용자에게 연결된 모든 기억된 연결 소스를 정리하는 함수.

사용 예)

```
SQL> set serveroutput on
SQL> exec twofactor.forget();

PL/SQL procedure successfully completed.

SQL>
```

2. ENABLE_ROLE 프로시저

ENABLE_ROLE 프로시저는 2차 인증으로 보호된 역할을 세션에서 활성화하는데 사용되며 PUBLIC에 부여됨.

ENABLE_ROLE (ROLE_NAME IN VARCHAR2);

"SET ROLE"을 실행하여 매개 변수로 전달된 보호된 역할을 사용 가능한 역할 목록에 추가. 실행하려면 2차 인증을 받아야 한다.

매개변수	설명	설정값	비고
ROLE_NAME	세션에서 활성화하려는 보호된 역할 이름		

사용 예) - Role 부여 성공(2차 인증을 선행한 경우)

```
SQL> set serveroutput on
SQL> exec enable_role('APPOBJACCESS');

PL/SQL procedure successfully completed.

SQL>
```

사용 예) - Role 부여 실패(2차 인증을 선행하지 않은 경우)

```
SQL> set serveroutput on
SQL> exec enable_role('APPOBJACCESS');
BEGIN enable_role('APPOBJACCESS'); END;

*
ERROR at line 1:
ORA-20000: User not authenticated in 2Factor.
ORA-06512: at "TOTP.ENABLE_ROLE", line 19
ORA-06512: at line 1

SQL>
```

2.5 ROLE 부여

정상적으로 모듈이 설치되었으면, 첫 번째 단계로 2차 인증으로 보호할 ROLE을 다음과 같이 작성해야 한다.

```
[oracle] /home/oracle/OraTOTP > sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Sat Jan 2 10:22:53 2021

Copyright (c) 1982, 2009, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE ROLE APPOBJACCESS IDENTIFIED USING TOTP.ENABLE_ROLE;

Role created.

SQL>
```

2차로 사용자를 인증한 후에만 기존 역할을 활성화하도록 선택적으로 변경할 수 있다.

2.6 사용자 생성 및 권한 부여

2차 인증을 사용하기 위한 사용자 생성 및 권한을 부여해야 한다. 사용자가 이미 존재한 경우는 사용자를 생성 것은 생략해도 된다.

```
SQL> CREATE USER USER1 IDENTIFIED BY "User1";

User created.

SQL> GRANT CREATE SESSION TO USER1;

Grant succeeded.

SQL> GRANT APPOBJACCESS TO USER1;

Grant succeeded.

SQL>
```

역할이 생성되어 USER1에게 부여되었다. 역할을 연결하고 활성화해 보자.

```
SQL> conn User1/User1
Connected.
SQL> select * from session_roles;

no rows selected
```

```
SQL> set role APPOBJACCESS;
set role APPOBJACCESS
*
ERROR at line 1:
ORA-28201: Not enough privileges to enable application role 'APPOBJACCESS'


SQL> exec enable_role('APPOBJACCESS');
BEGIN enable_role('APPOBJACCESS'); END;

*
ERROR at line 1:
ORA-20000: User not authenticated in 2Factor.
ORA-06512: at "TOTP.ENABLE_ROLE", line 19
ORA-06512: at line 1

SQL>
```

2.7 2차 인증 사용

보시다시피 set 명령을 통해 역할을 활성화할 수 없다. 올바른 방법은 enable_role 프로시저를 사용하는 것이지만 사용자를 먼저 인증해야 한다. 이 사용자는 아직 2차 인증이 구성되지 않았으므로 설정해 보겠다.

```
SQL> set serveroutput on
SQL> exec twofactor.setup('jlqlcHbVqdpj7b4PzBpM2DileBvmHFV/','30','app512');

PL/SQL procedure successfully completed.

SQL>
```

SETUP 절차를 실행한 뒤 2차 인증이 정상적으로 작동되는지 유효성을 다음과 같이 검사해야 한다.

```
SQL> exec twofactor.validate(682286);

PL/SQL procedure successfully completed.

SQL>
```

2차 인증의 검증을 통해 이제 새 세션을 만든 후 수행해야 할 모든 작업이 인증되고 역할을 활성화한다.

```
SQL> select * from session_roles;

no rows selected

SQL> exec twofactor.authenticate(390564);

PL/SQL procedure successfully completed.

SQL> exec enable_role('APPOBJACCESS');
```

```
PL/SQL procedure successfully completed.

SQL> select * from session_roles;

ROLE
------------------------------
APPOBJACCESS

SQL>
```

선택적으로, 2차 인증 시스템에 다음 7 일 동안 사용자의 위치를 신뢰하도록 요청할 수 있으므로 동일한 시스템, 터미널, IP, 프로그램 및 OS 사용자로부터 새로 로그인 할 때마다 다시 인증할 필요가 없다.

```
SQL> conn User1/User1
Connected.
SQL> exec enable_role('APPOBJACCESS');
BEGIN enable_role('APPOBJACCESS'); END;
*
ERROR at line 1:
ORA-20000: User not authenticated in 2Factor.
ORA-06512: at "TOTP.ENABLE_ROLE", line 14
ORA-06512: at line 1

SQL> exec twofactor.authenticate(388648);

PL/SQL procedure successfully completed.

SQL> exec enable_role('APPOBJACCESS');

PL/SQL procedure successfully completed.

SQL> exec twofactor.remember(471508);

PL/SQL procedure successfully completed.

SQL> conn User1/User1
Connected.
SQL> exec enable_role('APPOBJACCESS');

PL/SQL procedure successfully completed.

SQL>
```

2.8 NTP(Network Time Protocol) 설정

최근에는 Windows/서버/데이터베이스/네트워크 장비/저장장치에 대한 시간 동기화(타임서버 시간 동기화) 하는 방법으로 NTP(Network Time Protocol)을 이용하여 root 계정에서 시스템의 시각을 현재 시각으로 설정할 수 있다.

NTP를 사용하기 위해서는 기본적으로 NTP 패키지가 반드시 설치되어 있어야 한다. 설치 확인은 다음의 명

령어를 실행하여 확인한다. 만약, 설치되어 있지 않으면 Redhat, CentOS 8 이하 버젼은 "**yum install ntp**" 그외는 "**sudo apt-get install ntp**" 명령어로 설치하면 된다.

```
[root]# rpm -qa | grep ntp
ntp-4.2.2p1-18.el5.centos
chkfontpath-1.10.1-1.1
```

ntpd 서비스를 서버 부팅 시 시작프로그램에 등록 및 ntp 활성화 여부 확인은 다음과 같은 명령어로 확인할 수 있다.

```
[root]# chkconfig ntpd on
[root]# chkconfig —list | grep ntp
ntpd            0:해제  1:해제  2:활성  3:활성  4:활성  5:활성  6:해제
```

chkconfig 이용하여 서버 부팅시 ntpd 데몬 활성화 여부 확인 3, 5 level에 off(해제) 가 되어 있으면 자동 활성화되지 않는다. 자동 활성화하기 위해서는 3, 5에 on(활성)으로 다음과 같은 명령어로 변경해야 한다.

```
[root]# chkconfig —level 3 ntpd on
[root]# chkconfig —level 5 ntpd on
```

우리나라에서 운영되고 있는 NTP 서버는 다음과 같다.

```
server kr.pool.ntp.org
server time.bora.net
server time.kornet.net
```

우리나라에서 운영되고 있는 NTP 서버를 ntpd 데몬 설정을 위한 설정 파일인 "**/etc/ntp.conf**"에 다음과 같이 설정한다.

```
[root]# vi /etc/ntp.conf
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.centos.pool.ntp.org
#server 1.centos.pool.ntp.org
#server 2.centos.pool.ntp.org
#server 3.centos.pool.ntp.org
server kr.pool.ntp.org iburst
server time.bora.net   iburst
server time.kornet.net iburst
```

iburst 옵션은 일종의 옵션 설정으로써 동기화 하는데 걸리는 시간을 짧게 줄여주는 옵션임.

ntpd 데몬 설정을 위한 설정이 끝나면 반드시 NTP 설정이 제대로 추가되었는지 확인한 후 NTP 데몬의 Restart 작업이 반드시 필요하다.

```
[root]# /etc/init.d/ntpd restart
ntpd를 종료 중: [  OK  ]
ntpd (을)를 시작 중: [  OK  ]
```

ntpd 시간 확인은 다음과 같은 명령어로 확인할 수 있다.

```
[root]# ntpq -p
```

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 static.betaidc. 106.247.248.106  3 u    7   64    1    2.884  287.718   0.001
 time.bora.net   .INIT.          16 u    -   64    0    0.000    0.000   0.000
 183.110.225.61  .INIT.          16 u    -   64    0    0.000    0.000   0.000
 LOCAL(0)        .LOCL.          10 l    4   64    1    0.000    0.000   0.001
```

* 표시된 ip 가 현재 시간을 가져오고 있는 ntp 서버임

NTP를 사용하기 위해서는 기본적으로 NTP 패키지가 반드시 설치되어 있어야 한다. 설치 확인은 다음의 명령어를 실행하여 확인한다. 만약, 설치되어 있지 않으면 Redhat, CentOS 8 이상 버젼은 "**yum install chrony**" 명령어로 설치하면 된다.

```
[root@baropam ~]# rpm -qa | grep chrony
chrony-3.5-1.el8.x86_64
```

우리나라에서 운영되고 있는 NTP 서버는 다음과 같다.

```
server kr.pool.ntp.org
server time.bora.net
server time.kornet.net
```

우리나라에서 운영되고 있는 NTP 서버를 ntpd 데몬 설정을 위한 설정 파일인 "**/etc/chrony.conf**"에 다음과 같이 설정한다.

```
[root@baropam ~]# vi /etc/chrony.conf

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#pool 2.centos.pool.ntp.org iburst
server kr.pool.ntp.org iburst
server time.bora.net   iburst
server time.kornet.net iburst

# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift

# Allow the system clock to be stepped in the first three updates
# if its offset is larger than 1 second.
makestep 1.0 3

# Enable kernel synchronization of the real-time clock (RTC).
rtcsync

# Enable hardware timestamping on all interfaces that support it.
#hwtimestamp *

# Increase the minimum number of selectable sources required to adjust
# the system clock.
#minsources 2

# Allow NTP client access from local network.
allow 192.168.0.0/16
```

```
# Serve time even if not synchronized to a time source.
#local stratum 10

# Specify file containing keys for NTP authentication.
keyfile /etc/chrony.keys

# Get TAI-UTC offset and leap seconds from the system tz database.
leapsectz right/UTC

# Specify directory for log files.
logdir /var/log/chrony

# Select which information is logged.
#log measurements statistics tracking
```

ntpd 데몬 설정을 위한 설정이 끝나면 반드시 NTP 설정이 제대로 추가되었는지 확인한 후 NTP 데몬의 Restart 작업이 반드시 필요하다. (chrony 서비스 시작 및 부팅시 구동 등록)

```
[root@baropam ~]# sudo systemctl enable chronyd
[root@baropam ~]# sudo systemctl restart chronyd
```

ntpd 시간 확인은 다음과 같은 명령어로 확인할 수 있다.

시간을 받아오는 서버 리스트 / chrony.conf 파일에 등록한 server 리스트)

```
[root@baropam ~]# chronyc sources
210 Number of sources = 2
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* ec2-54-180-134-81.ap-nor>     2   6   377    43   -349us[-1059us] +/-   24ms
^- time.bora.net                 2   6   377    42  +1398us[+1398us] +/-   90ms
```

시간을 받아 오는 서버 정보)

```
[root@baropam ~]# chronyc tracking
Reference ID    : 36B48651 (ec2-54-180-134-81.ap-northeast-2.compute.amazonaw)
Stratum         : 3
Ref time (UTC)  : Sun Mar 22 07:07:43 2020
System time     : 0.000130027 seconds slow of NTP time
Last offset     : -0.000710122 seconds
RMS offset      : 0.000583203 seconds
Frequency       : 19.980 ppm fast
Residual freq   : +0.142 ppm
Skew            : 3.235 ppm
Root delay      : 0.013462566 seconds
Root dispersion : 0.017946836 seconds
Update interval : 65.0 seconds
Leap status     : Normal
```

시간 상태 및 동기화 등 정보 확인)

```
[root@baropam ~]# timedatectl status
               Local time: Sun 2020-03-22 16:08:45 KST
```

```
           Universal time: Sun 2020-03-22 07:08:45 UTC
                 RTC time: Sun 2020-03-22 07:08:44
                Time zone: Asia/Seoul (KST, +0900)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
```

3. About BaroPAM

Version 1.0 - Official Release - 2016.12.1

http://www.nurit.co.kr

제 조 사 : 주식회사 누리아이티
등록번호 : 258-87-00901
대표이사 : 이종일
이 메 일 : mc529@nurit.co.kr
주 소 : 서울시 강서구 마곡중앙2로 15, 913호(마곡동, 마곡테크노타워2)

조달총판 : 주식회사 루시드네트웍스
등록번호 : 848-86-00615
대표이사 : 박병호
대표전화 : 031-8018-4770(영업문의) / 031-8018-4771(기술지원)
이 메 일 : sales@lucidnet.co.kr
주 소 : 경기도 하남시 미사대로520, CA동 904,905호(덕풍동, 현대지식산업센터 한강미사2차)

공 급 사 : 주식회사 트루인테크놀로지스
등록번호 : 314-86-56237
대표이사 : 손원찬
대표전화 : 010-3404-1156(영업문의) / 080-488-8803(기술지원)
이 메 일 : wcson@truin.kr
주 소 : 대전시 서구 문예로 137, 4층(둔산동, 케이티엔지대전빌딩)

공 급 사 : 주식회사 디에이치솔루션
등록번호 : 606-86-54064
대표이사 : 조동환
대표전화 : 051-323-0705(영업문의) / 070-4632-0869(기술지원)
이 메 일 : sales@dhsolution.kr
주 소 : 부산시 해운대구 센텀동로 71, 벽산e센텀클래스원2차 1105호

공 급 사 : 주식회사 반디데이타
등록번호 : 264-81-49402
대표이사 : 백욱인
대표전화 : 02-864-5653(영업문의, 기술지원)
이 메 일 : bandidata@bandidata.com
주 소 : 서울시 금천구 벚꽃로 278, 1503호(가산동, SJ테크노빌)