SQL*Net Encryption & Authentication

Urh Srečnik <urh.srecnik@abakus.si>
DBA Team Lead @ Abakus Plus d.o.o.
OCP DBA, OCP J2SE, OCA OCI, OCIS Exadata

Oracle ACE Associate

ORACLE | Partner

https://blog.srecnik.info/

Proxy Users & Schema Only Accounts

Jan 20, 2025 5 min read

Remember those MYAPP/MYAPP@MYDB logons? With all the security buzz and auditors out there, I thought of those as a thing of the past. This blog won't...

Oracle Native Network Encryption

Dec 22, 2024 4 min read

As I come across environments where SQL*Net is not encrypted I sometimes also notice that there are two common misconceptions about it: First, some...

Introduction to JOIN LATERAL on Oracle

Dec 7, 2024 4 min read

Lateral join joins a subquery, which is executed as many times as there are rows in the leading table. Consider it as a kind of for-each loop. Most...

Reading Execution Plans

Oct 13, 2024 5 min read

Order of Operations · I routinely do a "rehearsal" of my presentations with our DBA team and any other willing participants (they tend to be developers)...

utlfixdirs.sql

Sep 16, 2024 2 min read

It's a script that comes with every $ORACLE_HOME and it's located in $ORACLE_HOME/rdbms/admin/ folder. It recreates all DBA_DIRECTORIES that are...

Collecting SQL Plan Baselines

Sep 9, 2024 7 min read

on Standard Edition · Ever thought about collecting baselines for all queries on your database? Let's explore the benefits and the methods of doing...

Abakus plus

Abakus Plus d.o.o.

- Infrastructure Team
  - HW & Services
  - OS & NET admin
  - DBA, Programming
- Applications
  - APPM
  - Deja Vu
  - Arbiter
- Development Team
  - Enterprise Applications
  - Document Management
  - Newspaper Distribution
  - Flight Information System

References


Gorenjska Banka
GENERALI Zavarovalnica
Ljubljana Airport
Iskra
REPUBLIKA SLOVENIJA
MINISTRSTVO ZA OBRAMBO
NOVA BANKA
MILENIJUM® OSIGURANJE
DRINA OSIGURANJE
MM KARTON
studio ritem
Hofra
Mestna občina Ljubljana
skbbanka
otp group
PH
triglav
GOODYEAR
TRELLEBORG
UNIVERZITETNA PSIHIATRIČNA KLINIKA LJUBLJANA
University Psychiatric Clinic Ljubljana
BANKA SLOVENIJE
SAVA INFOND
PRVA
ELES
Фонд здравственог осигурања Републике Српске
Blubit
EKDIS
CENTROSINERGIJA
SAVA HOTELS & RESORTS
NLB Vita
Življenjska zavarovalnica
MAGNETIK
Mercator
KONTROLA ZRAČNEGA PROMETA SLOVENIJE
MERKUR
SPL d.d.
terme čatež
NLB Skladi
ZAVOD ZA ŠPORT RS PLANICA
editel
NFOTRANS
LASERLINE
PRONET
CHOOSE THE FUTURE
hit alpinea
Kranjska Gora
ORACLE
ROS d.o.o.
sij acroni
ANDRITZ
jata emona
LJUBLJANA
ADRIA ANKARAN
HOTEL & RESORT
plus

Oracle Native Network Encryption

Licence Requirements

“Note: Network encryption (native network encryption, network data integrity, and SSL/TLS) and strong authentication services (Kerberos, PKI, and RADIUS) are no longer part of Oracle Advanced Security and are available in all licensed editions of all supported releases of Oracle Database.”

--

Database Licensing Information User Manual, Release 23 (&19c)

Oracle Native Network Encryption

… it's the easiest way of enabling encryption

- does not require any certificates
- the only configuration file involved is `sqlnet.ora`

so, let's start with this one

Server Side – sqlnet.ora

```
SQLNET.ENCRYPTION_SERVER = REQUESTED
SQLNET.CRYPTO_CHECKSUM_SERVER = REQUESTED


# allowed values:
# accepted | rejected | requested | required
```

Server Side sqlnet.ora (strict)

```
SQLNET.ENCRYPTION_SERVER = REQUIRED

SQLNET.CRYPTO_CHECKSUM_SERVER = REQUIRED

SQLNET.ENCRYPTION_TYPES_SERVER = (AES256)

SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (SHA512)

SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS = FALSE
```

Supported Encryption Algorithms in 19c

- DES (40 – 56 bit) // deprecated
- 3DES (112 – 168 bit) // deprecated
- RC4 (40 – 256 bit) // deprecated
- **AES** (128 – 256 bit)

Supported Hash Algorithms in 19c

- MD5 // deprecated
- **SHA-1**
- **SHA-2** (256 – 512 bits)

Client Configuration - sqlnet.ora

- Same settings can be used on client, just replace `_SERVER` with `_CLIENT`.

```
SQLNET.ENCRYPTION_CLIENT = REQUESTED

SQLNET.CRYPTO_CHECKSUM_CLIENT = REQUESTED
```

Is it working?

```
SQL> SELECT network_service_banner
  2  FROM v$session_connect_info
  3  WHERE sid=sys_context('userenv','sid');



NETWORK_SERVICE_BANNER
----------------------------------------------------------------------------------
…
AES256 Encryption service adapter for Linux: Version 19.0.1.0.0 - Production
SHA1 Crypto-checksumming service adapter for Linux: Version 19.0.1.0.0 - Production
…
```

Wireshark!

	TNS	114	Request, Data (6), Piggy back function follow
	TNS	449	Request, Data (6), User OCI Functions
	TNS	75	Request, Data (6), User OCI Functions
	TNS	469	Request, Data (6), Piggy back function follow
	TNS	86	Request, Data (6), Piggy back function follow
	TNS	67	Request, Data (6), User OCI Functions
	TNS	374	Request, Data (6), User OCI Functions
	TNS	75	Request, Data (6), User OCI Functions
	TNS	67	Request, Data (6), User OCI Functions
	TNS	64	Request, Data (6)

```
bc 24 11 1b db d5 bc 24  11 41 08 51 08 00 45 00   ·$·····$ ·A·Q··E·
01 b3 8f a6 40 00 80 06  87 8f c1 8a 2f b8 c1 8a   ····@··· ····/···
2f 42 fc ee 05 f1 a3 00  d6 08 83 44 ec eb 50 18   /B····· ···D··P·
00 ff e3 b4 00 00 00 00  01 8b 06 00 00 00 00 00   ········ ········
03 5e 06 61 80 00 00 00  00 00 00 fe ff ff ff ff   ·^·a···· ········
ff ff ff 17 01 00 00 fe  ff ff ff ff ff ff ff 0d   ········ ········
00 00 00 fe ff ff ff ff  ff ff ff fe ff ff ff ff   ········ ········
ff ff ff 00 00 00 00 01  00 00 00 00 00 00 00 00   ········ ········
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00 00 00 fe ff ff ff ff  ff ff ff 00 00 00 00 00   ········ ········
00 00 00 fe ff ff ff ff  ff ff ff fe ff ff ff ff   ········ ········
ff ff ff fe ff ff ff ff  ff ff ff 00 00 00 00 00   ········ ········
00 00 00 fe ff ff ff ff  ff ff ff fe ff ff ff ff   ········ ········
ff ff ff 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 5d   ········ ·······]
53 45 4c 45 43 54 20 44  45 43 4f 44 45 28 55 53   SELECT D ECODE(US
45 52 2c 20 27 58 53 24  4e 55 4c 4c 27 2c 20 20   ER, 'XS$ NULL',
58 53 5f 53 59 53 5f 43  4f 4e 54 45 58 54 28 27   XS_SYS_C ONTEXT('
58 53 24 53 45 53 53 49  4f 4e 27 2c 27 55 53 45   XS$SESSI ON','USE
52 4e 41 4d 45 27 29 2c  20 55 53 45 52 29 20 46   RNAME'),  USER) F
52 4f 4d 20 53 59 53 2e  44 55 41 4c 00 01 00 00   ROM SYS. DUAL····
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00 00 00 00 00 00 00 00  00 01 00 00 00 00 00 00   ········ ········
00 00 80 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
```

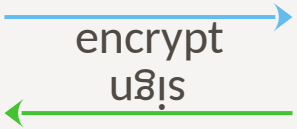
Destination	Protocol	Length	Info
	TNS	322	Request, Data (6), unknown
	TNS	530	Response, Data (6), unknown
	TNS	1394	Request, Data (6), unknown
	TNS	2322	Response, Data (6), Function Comp
	TNS	146	Request, Data (6), unknown
	TNS	194	Response, Data (6), unknown
	TNS	482	Request, Data (6), unknown
	TNS	418	Response, Data (6), unknown
	TNS	98	Request, Data (6), unknown
	TNS	194	Response, Data (6), unknown
	TNS	498	Request, Data (6), unknown
	TNS	274	Response, Data (6), unknown
	TNS	114	Request, Data (6), unknown
	TNS	98	Response, Data (6), unknown
	TNS	98	Request, Data (6), unknown
	TNS	98	Response, Data (6), unknown
	TNS	402	Request, Data (6), unknown
	TNS	354	Response, Data (6), unknown

```
bc 24 11 1b db d5 bc 24  11 41 08 51 08 00 45 00   ·$·····$ ·A·Q··E·
01 e4 90 48 40 00 80 06  86 bc c1 8a 2f b8 c1 8a   ···H@··· ····/···
2f 42 fd be 05 f1 67 21  29 95 6c f4 8b 71 50 18   /B····g! )·l··qP·
00 fd e3 e5 00 00 00 00  01 bc 06 20 00 00 00 00   ········ ··· ····
fd 93 24 8b d0 fa 66 13  bf 1c 81 a4 57 6d fc 12   ··$···f· ····Wm··
ed 7f 73 8b 63 ac c4 b6  a6 36 ab 2a 43 4f 22 a3   ··s·c··· ·6·*CO"·
75 59 a5 d6 8d d5 78 cd  9a b1 7a 30 c8 b9 da ab   uY····x· ··z0····
5b 8e 3f 43 37 9f 5a 3d  e2 3b f1 8b 48 d5 40 00   [·?C7·Z= ·;··H·@·
ad 1b 61 33 0a 1d 5e 5d  4e 11 ab 3e ba 0b 36 83   ··a3··^] N··>··6·
2e 25 43 84 9e 1b 66 92  15 e2 aa 08 f7 1c e8 c5   .%C···f· ········
92 87 34 f8 56 60 d8 13  02 69 01 9b 7f dd 04 1f   ··4·V`·· ·i······
9c dc 8f 44 ef cb a9 a3  ce d0 09 e4 6e 51 b9 84   ···D···· ····nQ··
a1 1c 3b ec cb 24 ac c4  58 ba e3 ca 08 32 67 83   ··;··$·· X····2g·
75 71 b4 d5 45 ab 52 a0  38 55 13 b2 ff 9d 4c ba   uq··E·R· 8U····L·
6a f8 95 2d d3 b5 29 88  77 77 a9 7c 62 17 96 fd   j··-··)· ww·|b···
f9 47 6f ed bb 7c d2 f9  24 8e 3f 1b 02 7f 27 b7   ·Go··|·· $·?···'·
9a 56 65 0f 45 93 18 12  1a 78 eb 3f e1 22 9b 39   ·Ve·E··· ·x·?·"·9
58 28 cc 80 fa 93 8e d0  93 4e f4 13 bf 00 b2 96   X(····· ·N······
4a 97 ec f6 37 00 cc 6e  db c5 4c 7f c0 5b 15 7d   J···7··n ··L··[·}
f9 df 75 a1 8f 1e ec 4a  bf 11 8b 51 ac e5 d4 46   ··u····J ···Q···F
39 f8 a6 25 1c 37 b9 e6  b8 01 11 e1 9c e0 6d e0   9··%·7·· ······m·
ce d7 69 cc 29 26 d4 de  11 63 1b 54 8f 19 7a ca   ··i·)&·· ·c·T··z·
47 aa fc 1a 94 3a a7 ce  8f 16 4c 95 7f c7 f5 4c   G····:·· ··L····L
07 09 05 3c 73 80 32 2c  0b fd e6 a8 18 76 53 3e   ···<s·2, ·····vS>
d9 61 ef 45 17 27 07 8d  80 9c 57 52 08 86 8c 0c   ·a·E·'·· ··WR····
a7 2f de 7c f4 8e 08 3e  f6 99 a1 73 1c f1 4c 29   ·/·|···> ···s··L)
50 d0 c8 96 67 df 7a 77  e0 09 33 a8 2b 15 a4 a6   P···g·zw ··3·+···
62 8a 24 3e 1b ab e4 0a  e4 e9 7e 3a 01 aa 33 f9   b·$>···· ··~:··3·
97 66 a6 7f d3 d8 ed 4d  6d d7 96 9b 04 57 54 0f   ·f·····M m····WT·
06 1e b8 d0 a1 ff 6f 70  44 e2 3d b8 96 6d 9d ed   ·····op D·=··m··
df a1 5c 89 fe a9 63 d9  0c ce 67 b8 b5 5e 65 91   ··\···c· ··g··^e·
```

TLS
Encryption & Authentication

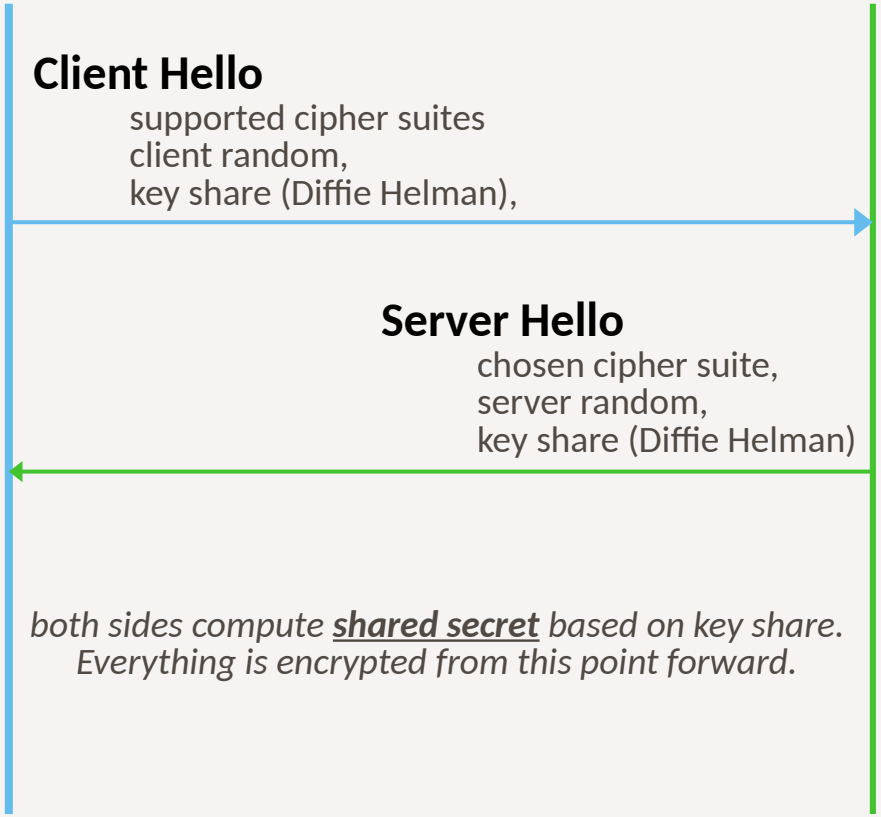
PKI (Certificates)

- **public key**
- CA signature
- “metadata”
  - Issued To:
  - Issued By:
  - Validity Period:
  - Common Name:



- **private key**
  - ***not*** a part of certificate
  - public key is derived from this one

TLS 1.3 – Handshake (concept)

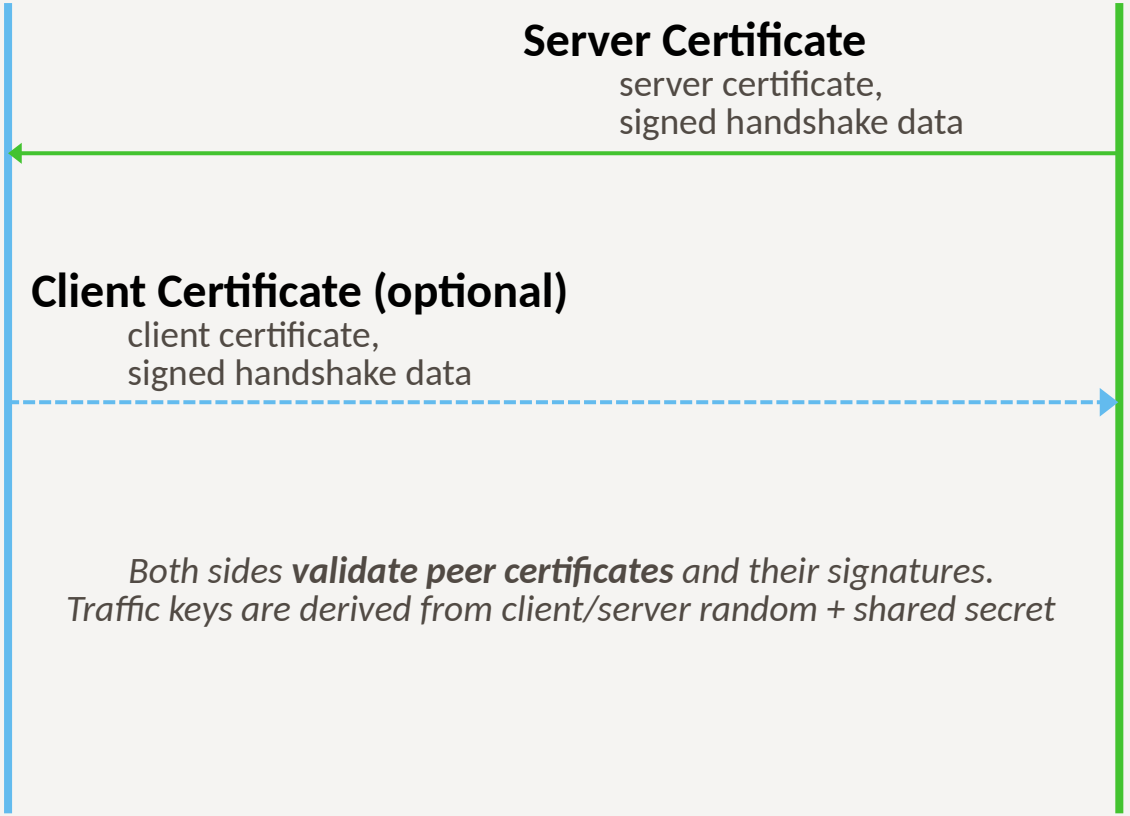


19c <= TLS 1.2

23ai <= TLS 1.3

+ forward secrecy
+ encrypted handshake
+ single round-trip
+ ...

TLS 1.3 – Handshake (concept)



Skip to client

What certificates do we need?

		Certificate	Private Key	Common Name
CA	CA	ca.**crt**	ca.**key**	*ca.example.com*
SE	Server	db-server.**crt**	db-server.**key**	*db-server.example.com*
CL	Client	db-client.**crt**	db-client.**key**	*db-client.example.com*

Let’s setup our own CA!

- EasyRSA
  https://github.com/openvpn/easy-rsa/

```
$ tar zxvf EasyRSA-3.2.2.tgz
$ ln -s /opt/EasyRSA-3.2.2 /opt/easy-rsa
$ export PATH="/opt/easy-rsa/:$PATH"
$ export EASYRSA_PKI='/opt/pki'
```

* *the exported* $PATH *contains one single bash script executable called* ***easyrsa***

Generate CA Certificate

```
$ easyrsa init-pki      # creates folder /opt/pki
  # see vars.example inside


$ easyrsa build-ca  # creates CA certificate:
                        # /opt/pki/ca.crt
                        # /opt/pki/private/ca.key
```

SE

Oracle Wallet Creation

```
$ orapki wallet create \
  -wallet /oracle/network/wallet \
  -pwd '<password>' \
  -auto_login


# creates: ewallet.p12
#  cwallet.sso (for -auto_login)
```

Abakus plus

SE

Oracle Wallet: Add CA Certificate

```
$ orapki wallet add \
   -wallet /oracle/network/wallet \
   -trusted_cert \
   -cert /path/to/ca.crt \
   -pwd '<password>'
```

SE

Oracle Wallet: Generate CSR

```
$ orapki wallet add \
  -wallet /oracle/network/wallet \
  -dn "CN=db-server.example.com" \
  -keysize 2048 \
  -pwd "<password>"

$ orapki wallet export \
  -wallet /oracle/network/wallet \
  -dn "CN=db-server.example.com" \
  -request /tmp/db-server.csr \
  -pwd "<password>"
```

CSR includes:

- public key
- metadata (DN)
- signature

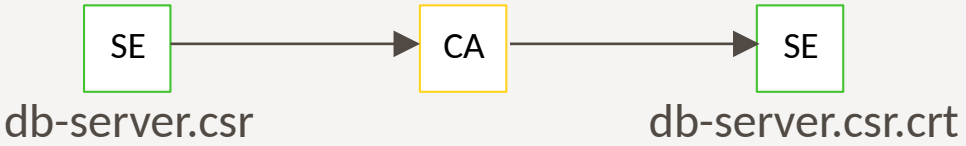
but ***without*** private key!

Abakus plus

CA

Sign CSR with CA

```
$ mv db-server.csr /opt/pki/reqs/db-server.csr.req

$ easyrsa sign-req server db-server.csr

    # produces ./issued/db-server.csr.crt
```



SE

Oracle Wallet: import signed certificate

```
$ orapki wallet add \
   -wallet /oracle/network/wallet \
   -user_cert \
   -cert /path/to/db-server.csr.crt \
   -pwd "<password>"
```

SE

Oracle Wallet: double-check

```
$ orapki wallet display \
  -wallet /oracle/network/wallet

User Certificates:
Subject:        CN=db-server.example.com
Trusted Certificates:
Subject:        CN=ca.example.com
```

SE

listener.ora

```
LISTENER =
    (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP) (HOST = db-server.example.com) (PORT = 1521))
        (ADDRESS = (PROTOCOL = TCPS) (HOST = db-server.example.com) (PORT = 1522)))

SSL_CLIENT_AUTHENTICATION = FALSE

WALLET_LOCATION=
  (SOURCE=
      (METHOD=file)
      (METHOD_DATA=
         (DIRECTORY=/oracle/network/wallet)))
```

Abakus plus

SE

sqlnet.ora

```
SSL_CLIENT_AUTHENTICATION = FALSE

WALLET_LOCATION=
  (SOURCE=
      (METHOD=file)
      (METHOD_DATA=
         (DIRECTORY=/oracle/network/wallet)))
```

(yes, they're the same two directives as used in listener.ora)

Abakus plus

CL

tnsnames.ora

```
db-server =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCPS)(HOST = db-server.example.com)(PORT = 1522))
    (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = orac)))
```

```
$ sqlplus x/y@db-server
…
ERROR:
ORA-29024: Certificate validation failure
```

← Why?

(answer is on the next slide)

Abakus plus

Client needs to trust server's CA

- #tls_handshake
- **sqlnet.ora** should point to a wallet with CA certificate

CL

sqlnet.ora

- orapki is not available with Oracle Instant Client

```
# sqlnet.ora:

WALLET_LOCATION=
  (SOURCE=
      (METHOD=file)
      (METHOD_DATA=

(DIRECTORY=/path/to/wallet)))
```

Abakus plus

CL

Create Client Wallet with CA

```
$ orapki wallet create \
  -wallet /oracle/client_wallet \
  -pwd <password> \
  -auto_login

$ orapki wallet add \
  -wallet /oracle/client_wallet \
  -trusted_cert \
  -cert /path/to/ca.crt\
  -pwd <password>
```

Client Test

Notice that the password is still being used

```
$ sqlplus scott/tiger@db-server

...

SQL> select
        sys_context('userenv','network_protocol') as protocol
        from dual;


PROTOCOL

----------------------

tcps
```

TLS
~ Client Authentication ~

Generate client certificate

```
$ easyrsa build-client-full db-client
nopass

# Certificate & private key created at:
# - /opt/pki/issued/db-client.crt
# - /opt/pki/private/db-client.key
```

CL

Import client certificate into wallet

- We could follow the same procedure as for the server
- Alternatively, note that we can also import existing pkcs12 wallet into Oracle wallet:
  - *How to Create a New Wallet from an Existing Private Key and Certificates using OpenSSL and orapki (Doc ID 2769138.1)*

CL

(cont.)

```
$ openssl pkcs12 -export \
  -in /path/to/db-client.crt \
  -inkey /path/to/db-client.key \
  -name 'db-client' \
  -out /tmp/keystore.p12


$ orapki wallet import_pkcs12 \
  -wallet /oracle/client_wallet \
  -pkcs12file /tmp/keystore.p12
```

SE

Enable Client Authentication

- `SSL_CLIENT_AUTHENTICATION = TRUE`
  - sqlnet.ora & listener.ora

- `SQL> alter user SCOTT identified externally as 'CN=db-client';`

CL

Final Test

```
$ sqlplus /@db-server

SQL> select user from dual;


USER

----------------------

SCOTT
```

~ additional options ~

sqlnet.ora – Additional Options

- SSL_VERSION = 1.2
  - force/accept specific SSL version
- SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS = TRUE
  - to avoid ORA-12696 Double Encryption Turned On

Debugging Client (sqlnet.ora)

```
TRACE_LEVEL_CLIENT = 16

TRACE_DIRECTORY_CLIENT = /oracle/sqlnet_log

TRACE_TIMESTAMP_CLIENT = ON

TRACE_UNIQUE_CLIENT = ON

DIAG_ADR_ENABLED = OFF
```

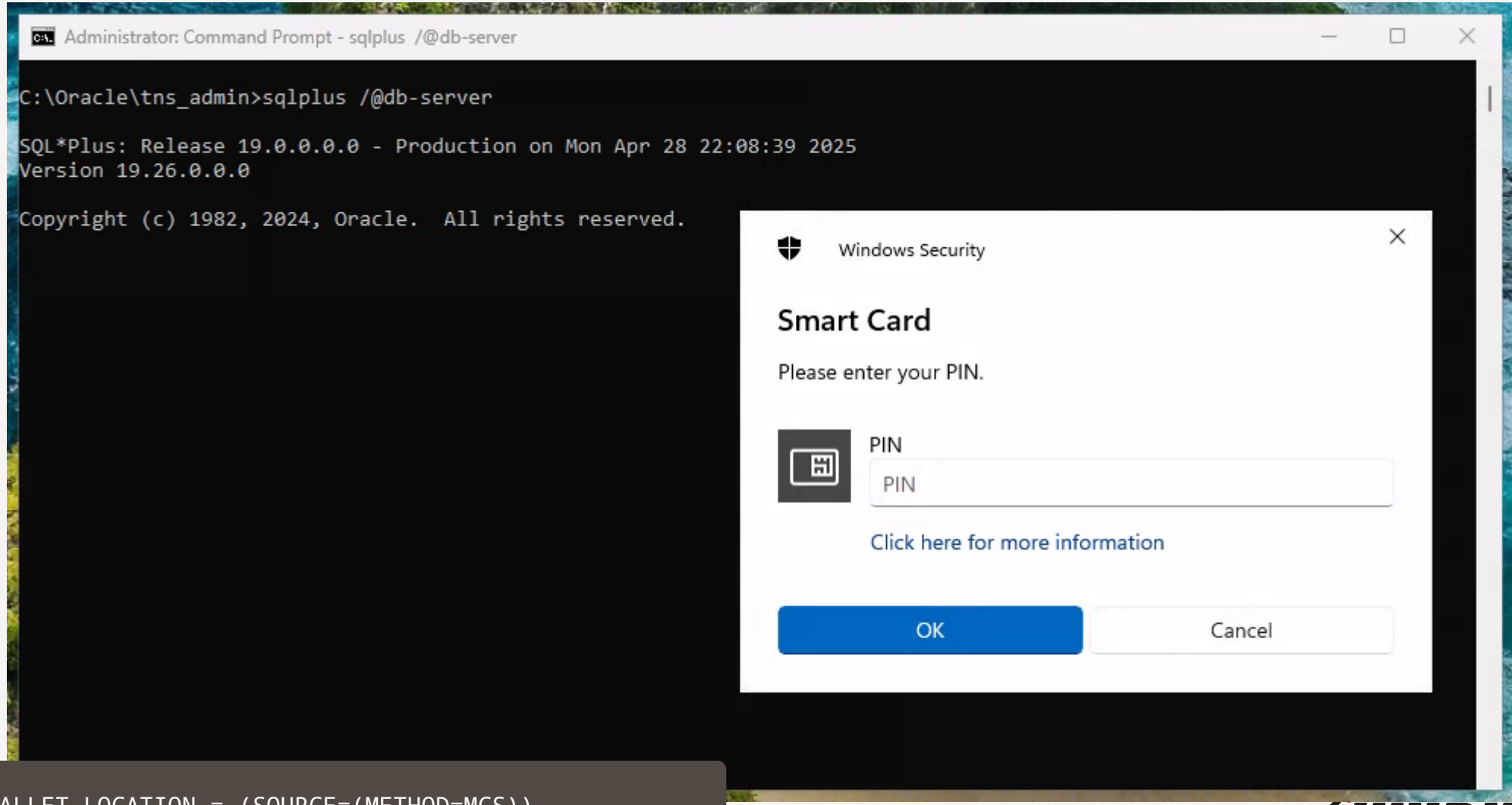
Debugging Server (sqlnet.ora)

```
TRACE_LEVEL_SERVER = 16

TRACE_DIRECTORY_SERVER = /oracle/network/log/

TRACE_TIMESTAMP_SERVER = ON

TRACE_UNIQUE_SERVER = ON

DIAG_ADR_ENABLED = OFF
```

tnsnames.ora

Client doesn’t really need sqlnet.ora to specify wallet location:

```
db-server =
 (DESCRIPTION=
   (ADDRESS = (PROTOCOL = TCPS)(HOST = db-server.example.com)(PORT=1522))
   (CONNECT_DATA = (SERVR = DEDICATED) (SERVICE_NAME = orac))
   (SECURITY=(MY_WALLET_DIRECTORY=/oracle/client_wallet)))
```




```
WALLET_LOCATION = (SOURCE=(METHOD=MCS))
```

Kerberos Setup

(just a quick *conceptual** overview)

#skip

Abakus plus

SE

Generate keytab

- Generate `acme.keytab` file.
  - generated by AD administrator using ***ktpass*** command
  - contains the ***secret key*** for a Kerberos service principal (something like `orakrb/hostname@REALM`)
    - It allows the Oracle Database to authenticate (decrypt) Kerberos tickets presented by clients, by matching the encrypted ticket against the key stored in the file.

krb5.conf

```
[libdefaults]
default_realm = ACME

[realms]
ACME = {
    kdc = ad.acme.com
    admin_server = ad.acme.com
}

[domain_realm]
.acme = ACME
ACME = ACME
```

SE

SE

sqlnet.ora (server)

```
SQLNET.KERBEROS5_CC_NAME=MSLSA:
SQLNET.KERBEROS5_CONF=C:\path\to\krb5.conf
SQLNET.KERBEROS5_KEYTAB=C:\path\to\acme.keytab
SQLNET.KERBEROS5_CONF_MIT=TRUE
SQLNET.AUTHENTICATION_KERBEROS5_SERVICE=orakrb
SQLNET.AUTHENTICATION_SERVICES=(NTS, KERBEROS5)
```

Abakus plus

sqlnet.ora (client)

- Similar to the one on server, but:
  - Without SQLNET.KERBEROS5_KEYTAB directive
  - And with added:
    SQLNET.AUTHENTICATION_SERVICES = (KERBEROS5)

SE

CL

Create an user & connect

```
SQL> create user aba_test
identified externally as 'scott@ACME.COM';
```

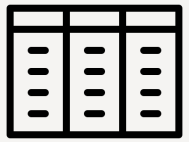
```
> okinit
> oklist
> sqlplus /@hostname:1521/service_name
```

Abakus plus

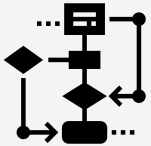
Related Ideas

Schema Only Accounts

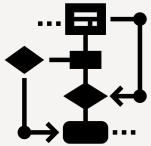
```
SQL> create user acme_owner no authentication;

User created.
```



acme_owner



acme_web_api



acme_mobile_api

Proxy Users

```
SQL> create user scott_developer identified by tiger;

SQL> alter user acme_owner
     grant connect through scott_developer;

SQL> grant create session to acme_owner;
```

```
$ sqlplus scott_developer[acme_owner]/tiger@db-server


SQL> select user from dual;

USER
--------------
ACME_OWNER
```

The Possibilities