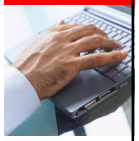
ORACLE

Oracle Advanced Security Internals

Oracle Advanced Security Internals Agenda



- Network encryption
- TDE column encryption
- TDE tablespace encryption
- TDE master encryption key storage and rotation
- Managing and storing Oracle Wallets
- Oracle Wallet: Backup and recovery
- Implementing TDE
- New in Oracle Database 11g Release 2 Patchset 1 (11.2.0.2)

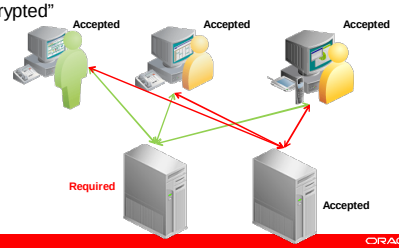
Demonstrating Network Encryption

- Starting with Oracle9*i*R2: Oracle Advanced Security is installed by default
- Modules for network encryption, integrity and authentication are running and set to **Accepted**
- Set server to 'Required'
- No changes to potentially 1,000's of clients

		Server setting			
		Rejected	Accepted	Requested	Required
Client setting	Rejected	Off	Off	Off	Failure
	Accepted	Off	Off	On	On
	Requested	Off	On	On	On
	Required	Failure	On	On	On

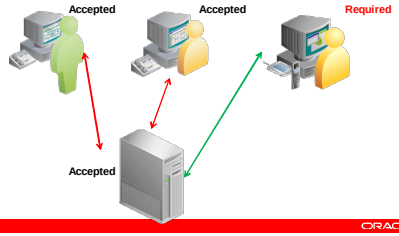
Simple solutions for recent customer scenarios

- "We need to encrypt all client connections to one server; all connections to another server are not encrypted"



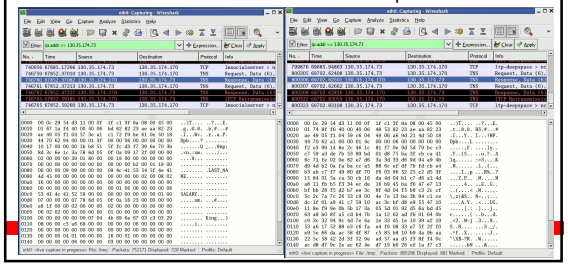
Simple solutions for recent customer scenarios

- "Few connections to a server from admin consoles are encrypted, other connections are not."



Demonstrating Network Encryption

- `V$SESSION_CONNECT_INFO` does **not change** when encryption and integrity are used, hence '`wireshark`' or similar tools are one option:



Configuring `sqlnet.ora`

- A more elaborate way is to configure clients and servers so that they search for the first match for encryption and integrity algorithms

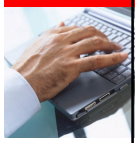
```
SQLNET.CRYPTO_CHECKSUM_SERVER = REQUIRED
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (SHA1)
SQLNET.ENCRYPTION_SERVER = REQUIRED
SQLNET.ENCRYPTION_TYPES_SERVER = (AES256,AES192,AES128)
SQLNET.CRYPTO_CHECKSUM_CLIENT = REQUIRED
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT = (SHA1)
SQLNET.ENCRYPTION_CLIENT = REQUIRED
SQLNET.ENCRYPTION_TYPES_CLIENT = (AES128,AES192)
```

- **`SQLNET.CRYPTO_SEED`** is deprecated since Oracle 9iR2, and replaced with pRNG string

ASO in Thin JDBC

	8.1.7	9.0.1	9.2	10.1	10.2	11.1	11.2
Authentication						SSL, PKI, Kerberos, RADIUS	
Encryption	DES (56 and 40 bits)						
				3DES168, 3DES112			
	RC4 (40, 56, 128, 256)						
						AES (256,192,128)	
Integrity						SHA-1	
	MD5						

Oracle Advanced Security Internals Agenda



- Network encryption
- TDE column encryption
- TDE tablespace encryption
- TDE master encryption key storage and rotation
- Managing and storing Oracle Wallets
- Oracle Wallet: Backup and recovery
- Implementing TDE
- New in Oracle Database 11g Release 2 Patchset 1 (11.2.0.2)

TDE column encryption internals

- Performs encryption and decryption in PGA user session memory
- Data blocks are fetched into the SGA's buffer cache encrypted and remain encrypted until result rows are returned via PGA
  - Only normal (B-tree) indexes are allowed on encrypted columns, and only for equality searches
  - Foreign key columns cannot be encrypted

Data types and lengths supported by TDE column encryption:

Most likely formats to store sensitive data	
varchar2 (< 3933 characters)	nvarchar2 (< 1967 characters)
char (< 1933 characters)	nchar (< 967 characters)
number	raw
binary_float	binary_double
timestamp	date
SecureFile (11.1.0.6 and later)	

TDE column encryption internals

- Please use 10.2.0.4/5, 11.1.0.7, 11.2.0.1/2
- Improvements in 10.2.0.4 and 11.1.0.7:
  - Introduction of the 'nomac' option
    - Reduce storage overhead by 20 bytes, fewer CPU cycles
    - Always use 'nomac' unless customer is concerned about tampering with encrypted values
    - Oracle database has integrity checks built-in, this is an additional protection
  - Install highly recommended patches
    - 7639262 (10.2.0.4)
    - 8421211 (11.1.0.7)

TDE column encryption internals

Storage considerations

- Mandatory:
  - Padding: Every encrypted field needs to be padded out to the next full 16 bytes (with AES), or 8 bytes (with 3DES)
  - Example:
    - 19 bytes + 13 bytes = 32 bytes encrypted with AES
    - 19 bytes + 5 bytes = 24 bytes encrypted with 3DES
- Optional:
  - SALT: + 16 byte
  - Message authentication code (MAC): + 20 byte
- Total: Between 1 and 52 bytes more *per value*

TDE column encryption

Encryption settings

- Table-wide settings (apply to all encrypted columns in this table):
  - Encryption algorithm
    - `AES256`
    - `AES192` (default)
    - `AES128`
    - `3DES168`
  - '`NOMAC`' (reverse command and default is '`SHA1`')
- Settings per column:
  - `NO SALT` (reverse command and default is '`SALT`')
  - A table can have salted and unsalted columns
  - `NO SALT` is mandatory for indexed columns

TDE column encryption

Residual clear text data

- Existing content in 'Recycle Bin' cannot be encrypted
  - Disable the Recycle Bin feature with:

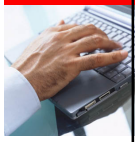
```
SQL> alter session set recyclebin = OFF;
SQL> alter system set recyclebin = OFF scope = SPFILE;
```

  - Purge tables from the Recycle Bin
- Data files can contain clear text version of encrypted data
  - '`alter table ... move` to itself' to overwrite stale data copies
  - Move table to other tablespace and delete old datafile on the OS level with '`sdelete`' or '`shred`'

How table keys are cached

- Table key is not cached, but deleted from memory when query ends
- With the Oracle Wallet:
  - '`open wallet`' command loads current **and** retired TDE master encryption keys into DB memory
    - Stored obfuscated until wallet is closed
  - Do not use PKI based TDE master encryption key, these are 100 … 500 times slower than symmetric keys
- With Hardware Security Module:
  - TDE master encryption never leaves the HSM
  - Table keys are sent to HSM when needed, and returned decrypted via secure connection

Oracle Advanced Security Internals Agenda



- Network encryption
- TDE column encryption
- TDE tablespace encryption
- TDE master encryption key storage and rotation
- Managing and storing Oracle Wallets
- Oracle Wallet: Backup and recovery
- Implementing TDE
- New in Oracle Database 11g Release 2 Patchset 1 (11.2.0.2)

TDE tablespace encryption internals

- Performs encryption and decryption at the block layer
- Data blocks in buffer cache are decrypted
  - All index types are supported
  - Equality and range searches are allowed
  - All data types are supported
  - Execution plans do not change
- 100% application transparent
- No additional storage required

TDE tablespace encryption internals

- Improvements in 11.2.0.1 over 11.1.0.7
  - Unified master encryption key (used for TDE column encryption and TDE tablespace encryption)
  - Full re-key support for unified master encryption key regardless if stored in Oracle Wallet or HSM
  - Unified master encryption key can be migrated from Wallet to HSM
  - Support for HSM partitions (slots) with patch 9453959 for 11.1.0.7 and 9229896 for 11.2.0.1

```
SQL> alter system set encryption wallet open
identified by "HSM_auth_string|<slot_name>";
```

ORACLE

19

TDE tablespace encryption internals

- Limitations
  - SYSTEM and SYSAUX tablespaces cannot be encrypted
  - UNDO, TEMP tablespaces and redo logfiles cannot be encrypted, but content temporarily stored there **is** encrypted.
  - Clear text tablespaces cannot be 'altered' to encrypted tablespaces
  - This also implies that re-keying at the tablespace-level is not possible

ORACLE

20

TDE tablespace encryption

Encryption settings

- Encryption algorithms
  - AES256
  - AES192
  - AES128 (default)
  - 3DES168
- 'SALT' is not optional and generated from:
  - Relative data block address (RDBA)
  - System change number (SCN)
- Multiple encrypted tablespaces created from the same clear text tablespace **are** different

ORACLE

21

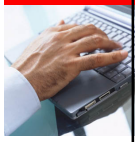
How tablespace keys are cached

- With the Oracle Wallet:
  - '`open wallet`' command loads TDE master encryption key (current and retired keys) into DB memory
    - Stored obfuscated until Wallet is closed
- With Hardware Security Module:
  - TDE master encryption never leaves the HSM
  - Tablespace keys are sent to HSM when first needed, and returned decrypted via secure connection
- Tablespace keys needed by the system **are kept in DB memory** until shutdown
- Tablespace keys for user operations are not cached

ORACLE

22

Oracle Advanced Security Internals Agenda



- Demonstrating Network
- TDE column encryption
- TDE tablespace encryption
- TDE master encryption key storage and rotation
- Managing and storing Oracle Wallets
- Oracle Wallet: Backup and recovery
- Implementing TDE
- New in Oracle Database 11g Release 2 Patchset 1 (11.2.0.2)

ORACLE

23

TDE master key storage and rotation

- TDE master key storage and rotation options

DB release	Enc. Type	Wallet	HSM	Re-key / Migrate
10.2.x	Column	**Yes**	No	**Yes / n/a**
11.1.0.6	Column	**Yes**	**Yes**	**Yes**
	Tablespace	**Yes**	No	No
11.1.0.7	Column	**Yes**	**Yes**	**Yes**
	Tablespace	**Yes**	**Yes**	No
11.2.0.1/2	Unified MK	Yes	Yes	Yes

- Migrating the TDE master key is a re-key operation; this explains why the TDE master key for TDE tablespace encryption in 11.1.0.7 cannot be migrated from a wallet to HSM

ORACLE

24

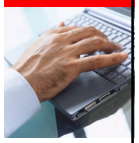
Encrypting files outside of the database

- TDE column encryption allows to encrypt columns in external tables
- DBFS provides OS-like access to files in a 'directory' that is stored in the database, without the user knowing it
  - Encryption with SecureFiles is an installation option
  - Deselect encryption with SecureFiles, put the entire directory in an encrypted tablespace
  - Files are under transactional controls of the database, can be versioned, compressed, and are backed-up professionally

'Undo' TDE

- "I want TDE, I don't want TDE, I want TDE"
  - Create wallet with TDE master encryption key
  - Encrypt data
  - Decrypt data
  - Drop wallet
  - Re-enable TDE:
    - 'typed master key not found'
  - Install patch 8682102 and perform log switches to cycle through all log files
  - Create new wallet and re-enable TDE

Oracle Advanced Security Internals Agenda



- Network encryption
- TDE column encryption
- TDE tablespace encryption
- TDE master encryption key storage and rotation
- Managing and storing Oracle Wallets
- Oracle Wallet: Backup and recovery
- Implementing TDE
- New in Oracle Database 11g Release 2 Patchset 1 (11.2.0.2)

Managing and storing Oracle Wallets

- Encryption wallet (`ewallet.p12`)
  - Encrypted with the wallet password (➔ PKCS#5)
  - Needs to be opened manually for the database to encrypt and decrypt data
  - NEVER delete the encryption wallet
- Auto-open wallet (`cwallet.sso`)
  - Wallet is opened automatically when database accesses encrypted data for the first time
  - NEVER backup `cwallet.sso` together with database files!
- Local auto-open wallet (`cwallet.sso`)
  - Only auto-opens on the server is was created on.

Oracle Wallet basics and essentials

- Use strong password to protect the wallet
- Never forget the wallet password
- Restrict wallet file and directory permissions
- Store the Oracle Wallet outside of the '`$ORACLE_BASE`' tree to avoid accidentally backing up the wallet with your database; for example in:

```
/etc/ORACLE/WALLETS/oracle
```


If the wallet is lost

- When the Encryption Wallet is lost, encrypted data cannot be recovered
  - Encrypting data and destroying the encryption key is the most reliable way to delete data
- Attempts to re-create the wallet will fail, because the master encryption key will not be the same

Protect the Oracle Wallet

- Example: Store the Oracle Wallet in `/etc/ORACLE/WALLETS/oracle`
- Create directories (`/etc` is owned by '`root`'):

```
# cd /etc
# mkdir –pv ./ORACLE/WALLETS/oracle
mkdir: created directory 'ORACLE'
mkdir: created directory 'ORACLE/WALLETS'
mkdir: created directory 'ORACLE/WALLETS/oracle'
```

- Change owner and set directory access rights

```
# chown –R oracle:oinstall ORACLE/*
# chmod –R 700 ORACLE/*
```

ORACLE

31

Prepare database, create wallet and master key

- Oracle Enterprise Manager
  - Retrieve default wallet location from `v$encryption_wallet`
  - **Either** create sub-directory '`wallet`' directory in
    - `$ORACLE_BASE/admin/$ORACLE_SID/`
  - **Or** define directory in '`sqlnet.ora`', overwrites value in `v$encryption_wallet`
  - Create wallet and TDE master encryption key from within TDE homepage in Enterprise Manager
- In Enterprise Manager, with Oracle Database Vault
  - User who manages TDE needs to be participant or owner in the 'Data Dictionary Realm'

ORACLE

32

Prepare database, create wallet and master key

- Create directory
- Add entry to `$TNS_ADMIN/sqlnet.ora`:

```
ENCRYPTION_WALLET_LOCATION =
  (SOURCE = (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /etc/ORACLE/WALLETS/oracle)))
```

- Use `SQL*Plus` to create wallet and TDE master key:

```
SQL> alter system set encryption key identified by
"<strong_password>";
```

ORACLE

33

Lock down the wallet

- Initially, the wallet is created with the default privileges of the `oracle` user, for example 644:

```
$ cd /etc/ORACLE/WALLETS/oracle
$ ls –l
$ -rw-r--r--          ewallet.p12
```

- This should be reduced to:

```
$ chmod 600 ewallet.p12
```

- In order to protect the wallet from accidental deletion, make it read only:

```
$ chmod 400 ewallet.p12
```

- This also disables updates to the wallet (Master key re-key operations), returns error message

ORACLE

34

Changing the Wallet password

- Changing the Wallet password does not change the TDE master encryption key; they are independent
- Copy the wallet before changing the password
- Change the password
- Close the wallet, and try to open it with the new password
- If new password is not accepted, restore wallet from copy and try changing the password again
- If successful, continue to use the new wallet
- Backup the wallet after changing the password
- Never forget the wallet password

ORACLE

35

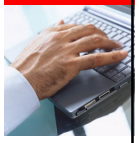
Changing the Wallet password

- Prior to 11.1.0.7, Oracle Wallet Manager was needed
- With 11.1.0.7 and later, '`orapki`' can be used as well
- Changing the wallet password requires knowledge of the current password
  - Also when (local) auto-open wallet is present
- Select a strong password!
- Never forget the wallet password

ORACLE

36

Oracle Advanced Security Internals Agenda



- Demonstrating Network encryption
- TDE column encryption internals
- TDE tablespace encryption internals
- TDE master encryption key storage and rotation
- Managing and storing Oracle Wallets
- Oracle Wallet: Backup and recovery
- Implementing TDE
- New in Oracle Database 11g Release 2 Patchset 1 (11.2.0.2)

ORACLE

37

Wallet backup – why and how often

- Backup the wallet before and after each modification
  - Immediately following the initial creation of the Wallet
  - Before each master key re-key operation
  - Before changing the wallet password
- Lost the wallet, forgotten wallet password, and no backup?
  - No way to retrieve encrypted data
  - Encrypting data and destroying (loosing) the encryption key is the most reliable method of deleting data

ORACLE

38

Key recovery

- The master encryption key for TDE column encryption (also the unified MK in 11.2 when TDE tablespace encryption is **not** used) can be generated as a PKI public/private key pair.
  - Public key (even though it's not public) **encrypts** table keys
  - Private key **decrypts** table keys
  - 100 … 500 times slower than symmetric keys
- Table keys are not cached

ORACLE

39

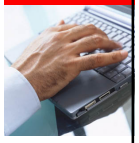
Key recovery

- Key recovery and key escrow involves a trusted third party (Certificate Authority) which is a controversial topic and involves technology with 100% auditing

ORACLE

40

Oracle Advanced Security Internals Agenda



- Network encryption
- TDE column encryption
- TDE tablespace encryption
- TDE master encryption key storage and rotation
- Managing and storing Oracle Wallets
- Oracle Wallet: Backup and recovery
- Implementing TDE
- New in Oracle Database 11g Release 2 Patchset 1 (11.2.0.2)

ORACLE

41

Implementing TDE column encryption

Existing table

- Encrypt existing column in place
  - `alter table <table> modify (<column> encrypt [using '<algorithm>'] [no salt] ['nomac']);`
  - While existing values are encrypted, table remains READ accessible due to table lock
- If column has in index:
  - Extract index DDL (`dbms_metadata.get_ddl`), save to file
  - Drop index
  - Encrypt column
  - Rebuild index with command retrieved from file
  - 2 .. 3 times faster than encrypting column with index in place

ORACLE

42

Implementing TDE column encryption

Existing table using Online Table Redefinition

- If table needs to remain fully accessible
  - Use Online Table Redefinition
    - Transparent to applications and users
    - No downtime, no data loss
  - Extract table DDL (`dbms_metadata.get_ddl`), save to file
  - Modify table name in stored DDL command, for example:
    - `customers_int`
  - Add encryption parameters to sensitive columns
  - Create empty interim table by running SQL from file
  - Follow documentation to complete Online Table Redefinition

Implementing TDE tablespace encryption

During application install time

- Add '`ENCRYPTION_WALLET_LOCATION`' parameter to `sqlnet.ora` file
- Create master encryption key either in Oracle Wallet or Hardware Security Module
- Verify if Wallet is open
- Locate the part of the application installation script that generates the application tablespaces
  - `CREATE TABLESPACE AMAPP DATAFILE '/opt/oracle/oradata/psft/amapp.dbf' SIZE 90M EXTENT MANAGEMENT LOCAL AUTOALLOCATE ENCRYPTION using 'AES256' DEFAULT STORAGE(ENCRYPT)`
- Run the installation script without any other changes

Implementing TDE tablespace encryption

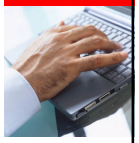
In an existing installation

- If the application needs to remain fully accessible
  - Use Online Table Redefinition
    - Transparent to applications and users
    - No downtime, no data loss
  - Extract DDL, save to file, for:
    - Application tablespaces that are encryption candidates
    - All tables (and indexes) in these tablespaces
  - Add encryption keywords to tablespace DDL
  - Create new encrypted tablespaces
  - Change tablename and tablespace name in table DDL
  - Create empty interim tables in encrypted tablespaces
  - Follow documentation to complete Online Table Redefinition

Setup TDE in Real Application Clusters and Exadata

- Oracle RAC 11gR2:
  - Wallet in shared location (ACFS) recommended
    - ACFS available for Windows, Linux, Solaris, AIX
    - Wallet open/close and master encryption key re-key operations are synchronized across all instances
  - If wallet is copied to all instances, only wallet open/close is synchronized:
- Older versions (and Exadata V1, V2 and X2)
  - Create wallet on first node
  - Copy wallet to all other nodes
  - 'Open wallet' on all other nodes

Oracle Advanced Security Internals Agenda

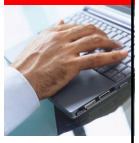


- Network encryption
- TDE column encryption
- TDE tablespace encryption
- TDE master encryption key storage and rotation
- Managing and storing Oracle Wallets
- Oracle Wallet: Backup and recovery
- Implementing TDE
- HSM update
- New in Oracle Database 11g Release 2 Patchset 1 (11.2.0.2)

Update on HSM support

- RSA refuses to complete the integration test against Oracle Database 11g Release 2
- Safenet sometimes competes against TDE with DataSecure
- Thales released new software version 11.40 which includes their part to our fixes for HSM partition support
- Utimaco has been acquired by Sophos, but HSM business remains an independent entity (as required by German authorities)
- BULL are most recent to certify
- Sun SCA6000 can be used as FIPS 140-2 level 3 certified local HSM to replace the Oracle Wallet
- Integration with SUN KMS announced at OW '10
- No new HSM partners unless customers specifically ask for them

Oracle Advanced Security Internals Agenda



- Network encryption
- TDE column encryption
- TDE tablespace encryption
- TDE master encryption key storage and rotation
- Managing and storing Oracle Wallets
- Oracle Wallet: Backup and recovery
- Implementing TDE
- HSM support
- New in Oracle Database 11g Release 2 Patchset 1 (11.2.0.2)

ORACLE

49

What's new in 11.2.0.2

- Automatic use of hardware-based encryption acceleration
  - Intel AES-NI in Xeon 56xx CPUs
  - Coming soon: Oracle SPARC T3 CPUs
  - Performance overhead for encryption and decryption drops ~ 80% !!
  - Hardware acceleration for encryption is disabled by default, apply patch 10080579

ORACLE

50

Lock down the wallet in ACFS (11.2.0.2 on Linux only)

- '`root`': Initialize Security for the cluster
- '`secadmin`'
  - changes password for cluster security
  - prepares each ACFS file system for Security
  - creates Realm
  - creates Ruleset
  - creates Rule to only allow Oracle binary (optional **`orapki`** and **`Oracle Wallet Manager`**) to access the wallet
  - adds Rule to Ruleset
  - adds Ruleset to Realm
- Detailed example in Dec. '10 issue of TDE best practices document

ORACLE

51

Crypto acceleration for TDE tablespace encryption in Exadata X2

Exadata Model	X2-2		X2-8	
Node	Encrypt	Decrypt	Encrypt	Decrypt
Compute	Enable hardware-acceleration with patch 10080579	Hardware acceleration enabled by default	Reduced hardware acceleration (~ 2x) through Nehalem technology in Intel® X7560, encryption needs patch 10080579	
Storage	n/a	Hardware acceleration enabled by default	n/a	Hardware acceleration enabled by default

ORACLE

52

What Yuntaa Is Saying

"I wouldn't trust the encryption of our content to anything or anyone else."

Andy Barrett, CTO, Yuntaa NV

ORACLE

53

ORACLE®