Database Security Challenges By Kevin Harianto

# PART-A-(30 POINTS-3 POINTS EACH)

***Prerequisites for this activity:***

*Keystore should be created, opened with the setting of master key.*

As a Database Security Administrator, you are required to create the encrypted tablespaces and ass encrypted columns to an existing tablespace. In your database if the keystore is already created and opened then jump to questions, if not then first set up the keystore(that includes creating the keystore, opening it and setting up the master key) using the lesson slides and labs and as demonstrated by me in the hands-on lab during the class.

* 1. Create the hr.client\_payment table with encrypted columns. Use the following schema to create this table. This table should be created from the sys user account in hr schema. The table name can be prefixed with hr schema name.

|  |  |
| --- | --- |
| Column Name | Data Definitions |
| f\_name | Varchar2(10) |
| l\_name | Varchar2(15) |
| balance | Number(10) |
| credit\_card\_no | Varchar2(15), encrypt |
| c\_id | Number(5), primary key |

The table is created in the default tablespace of the user that issues this command. The credit\_card\_no column is encrypted. All data entered for the credit\_card\_no column would be encrypted on disk. Any user with access to the credit\_card\_no data can view the decrypted data. A database user need not be aware if the contents of a particular column are encrypted on the disk.

Graphical user interface, text, application

Description automatically generated

* 1. You can now enter data into the table. The following example shows some sample data to be inserted into the hr.client\_payment table:

Row 1: ('Joe', 'Oldfield', 10000, '5446959708812',1);

Row 2: ('Chris', 'White', 15000, '5122358046082',2);

Row 3: ('Alex', 'Jones', 20000, '5595968943757',3);

Row 4: ('Mike', 'Anderson', 30000, '4929889576357',4);

Row 5: ('Ann', 'Hetmyer', 10005, '4556988708236',5);

Row 6: ('Ellen', 'Meyer', 10006, '374366599711',6);

Row 7: ('Celine', 'Rhodes', 10007, '4716898533213',7);

Row 8: ('Steve', 'Smith', 10008, '3409759003762',8);

Row 9: ('Albert', 'Bairstow', 10009, '3106543054125',9);

Graphical user interface, text

Description automatically generated

* 1. You can create an index on an encrypted column if it has been encrypted. Let us create an index on the credit\_card\_no column. The following command creates an index on the credit\_card\_no column:

CREATE INDEX hr.client\_payment\_idx ON hr.client\_payment (credit\_card\_no);

Graphical user interface, text, application, email

Description automatically generated

* 1. Create a new table hr.operations with the following column definitions:

Name

Null? Type

FIRST\_NAME

VARCHAR2(11)

LAST\_NAME

VARCHAR2(10)

EMP\_SIN

VARCHAR2(9)

DEPT\_ID

VARCHAR2(20)

Graphical user interface, text, application

Description automatically generated

* 1. You can use the ALTER TABLE command to alter an existing table. Let us alter a table called hr.operations with no encrypted columns. Encrypt the emp\_sin column in the hr.operations table.

Graphical user interface, text, application, email

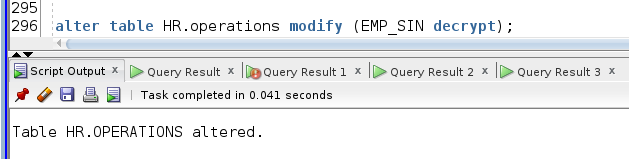
Description automatically generated

* 1. Verify the schema of a table to show the encrypted column. All existing data in the emp\_ssn column will now be encrypted on the disk. Data would be transparently decrypted for database users, who otherwise have access to the data.

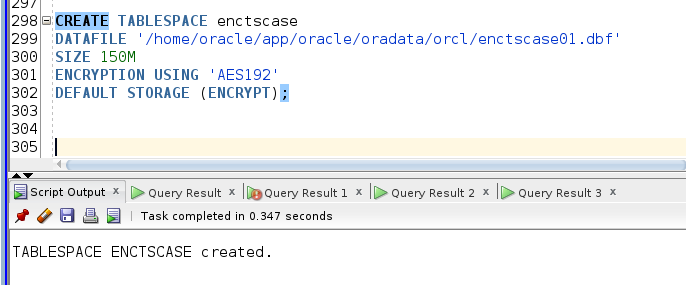
Graphical user interface, text

Description automatically generated

* 1. Turn off the Column encryption of emp\_ssn column



* 1. Let us create an encrypted tablespace to store encrypted tables. Create an encrypted tablespace called enctscase of size 150m with a datafile enctscase01.dbf. You can use any absolute path/location in your OS for creating the Tablespace.



* 1. If we create a table in an encrypted tablespace, then all data in the table is stored in encrypted form on the disk. Create a table called, cust\_in\_pay in an encrypted tablespace called, enctscase using the following schema:

Col 1: first\_name VARCHAR2(11), Col 2: last\_name VARCHAR2(10), Col 3: order\_number NUMBER(5),

Col 4: credit\_card\_number VARCHAR2(16), Col 5: active\_card VARCHAR2(3))

TABLESPACE enctscase;

Graphical user interface, text, application

Description automatically generated

* 1. Write down the SQL command syntax to perform the following tasks:

|  |  |
| --- | --- |
| Task | SQL Command |
| Create table and encrypt column |  |
| Encrypt unencrypted existing column |  |
| Master encryption key: set or reset |  |
| Wallet: open to access master encryption  key |  |
| Add encrypted column to existing table |  |

# PART-B(Research Based Questions)(20 points-2 Points each) Answer for each question should have at least 8 to 10 lines.

1. List the different kinds of Oracle DB 12c features/objects that TDE column encryption does not support.

1.Does not support post encryption compression. Unable to encrypt and compress columns further to make it harder to unlock the zip objects and to unlock again after extending using a different key.

2.Does not support easy migration of master key and backup without a third party software

3. Does not support easy critical data identification and threat response for better manageability on what objects/columns should be idententified for encryption.

4.Does not support encryption during data backup.

5.Does not support encryption while a snapshot is made.

6.Does not support flexibility for migrating a keystore in united mode.

7.Does not support automatic encryption tuning to showcase data overhead immediately when data is encrypted.

8.Does not support software based cryptographic acceleration with the use of easy remote cloud processesing frameworks.

9.Does not support automatic partitioning of the columns so that less data is scanned in the whole column encryption.

10.Does not support automatic parallization for column encryption to help reduce overhead on systems when encrypting and decrypting large columns.

1. In Oracle databases, there are possibilities of creating the local auto open wallet. How can I create one? List all of the prerequisites needed for creating the local wallet and procedure to create it.

1.Requires a secure folder for wallet.

2.Requires permissions for execution

3.Requires the secure folder path to utilize the folder.

4.Configuration of the Keystore type to be open.

5.Requires the system’s pfile to automatically accept the keystore wallet.

6.Requires the database to restart.

7.Requirement for the configuration of the actual keystore to utilize the folder.

8.guarantee the wallet is open and that the file is accessible.

9.Requires the keystore to be opened forcefully.

10.Finally, requires the wallet to use the TDE encryption Master Key to use the secure folder path and the restart the database again to finalize all changes.

1. The auditor has requested to present proof of encryption, List down all of the views and their purpose which shows the encryption level statistics and you could present it to the auditor for attestation.

1. DESC Encryption summary view to show all objects that contains encrypted data as well as statistics about these data.

2.V$ENCRYPTION\_KEYS to show the levels of encryption in terms of the keystore itself.

3. V$ENCRYPTED\_TABLESPACES to view the encrypted tablespaces and their statistics which includes the algorithm used for encryption.

4.V$DBA\_ENCRYPTED\_COLUMNS to view the encrypted columns more in depth and determine wether or not the columns are critical enough to be encrypted.

5.V$ENCRYPTION\_WALLET to check the location in terms of where the encryption keys are stored at and to present to the auditor in case the location is not secure enough.

6.USER\_TABLESPACES to allow for comparison to the encrypted tablespaces for the auditor to check if the critical talespaces are dealt with in terms of being adequately protected.

7.V$DATABASE to present the current status of the master encryption key for proof of encryption.

8.V$USER\_ENCRYPTED\_COLUMNS to present which users data are encrypted to maintain compliance and security.

9.V$DICTIONARY\_CREDENTIALS\_ENCRYPT to present whether or not the credential systems tables are encrypted.

10.V$DATABASE\_KEY\_INFO to showcase the encryption algorithm and whether or not there is a need for the encryption standard to be updated to meet the requirements for complian regulations.

1. How can I find out the data which needs encryption?

1.Search throught the tables in which may contain personally identifiable information

2.Look into relevant regulations and compliance standards to determine what data needs encryption to meet the requirements for security.

3.Look into what data contains confidential business intellectual property

4.look into what has sensitive data that may make users not want publicized or make their lives harder.

5.Look into what data that should never be shown to random strangers.

6.Query throughout the tablespaces and objects for any data that resources were spent heavily on and encrypt them.

7.Question whether or not if an employee accidentally leaked the data if it would cause harm to any employees or customers and encrypt accordingly.

8.Look into whether or not the data leakage would put a constraint on the company’s ability to stay competitive in the market as well as stay reputable.

1. How can I choose between Column based encryption and Tablespace based encryption?

1.Is there a lot of forseeable sensitive data that can be easily grouped together use tablespace encryption.

2.Is there an existing object that you want to hide use column.

3.Is there going to be a lot of critical data made in the schema use tablespace encryption.

4.Is there only going to be very few objects that actually needs to be encrypted use column.

5.Do you want to create objects that you know will all be relevant and critical to your activities use tablespace encryption.

6.If the table being created only contains very few critical and sensitive objects that were just being identified as a need to be encrypted use column encryption.

7.if a large surpluss of company data will be aggregated into a central area use tablespace encryption.

8.If an existing object in a table was just determined to be sensitive and needs to be encrypted in order to meet compliance in a recent update, use column encryption to meet the minimum and to lessen overhead.

1. List all of the Oracle 12c DB supported encryption algorithms that can be used with TDE?

1.3DES triple encryption standard 168 bits.

2.AES Advanced Encryption Standard 128 buts minimum all the way to 256 bits.

3.Regional encryption algorithms ARIA. Block cipher encryption algorithm utilizing substitution in comparison of AES.

4. Regional encryption algorithms SEED 128-bit block block encryption with an equally long key using division.

5. Regional encryption algorithms GOST Block encryption algorithm for rotation of characters.

6. PKCS#12 public key cryptography standard. Uses two keys to maintain security.

7.FIPS Federal Information Processing Standard. Makes data unintelligible through cipher text.

8.Chain CBC cipher block chaining. Plaintext is exclusive or, before encryption.

1. What is the role of Oracle Database Vault in decrypting the encrypted data for all users who have been authorized to see it?

1. provides easily centralized information that can be easily managed to ensure better validation for those authorized.

2.continuously allows authorized users in while still maintaining locked security against unauthorized users.

3.Ensure multiple varied access across multiple applications for easier readability and less hassel for accessesing encrypted information.

4.Enabling secure remote access while still maintaining security and while not breaching regulatory compliance through least privilege.

5.Allows for better seperation of duties so authorized people with access of data wont accidentally cause harm in other branches of sensitive information.

6.provides a more cost efficient manner for authorized users to access information through secure centralization of controls.

7.Allows for authorized users to maintain access to data despite changing relationships to companies role, allowing for better acess to needed information without having to remake or to manually give access.

8. provides encryption controls in terms of encrypting and decrypting data for the users authorized to see it.

1. Do research to find out the impact of TDE tablespace and column based encryption on different Hardware resources like Memory and Storage.

1.TDE in terms of tablespace has an estimated performance impact around 5-8% on the CPU in proportion to the amount of data however this can scale up to 5 to 10 times slower depending on the magnitude and the parallel activities.

2.Encryption overhead will be lower if most of the data is accessed from memory after loading in.

3.hardware components for faster access and execution of programs will be more constrained. This means that more memory, faster storage and a good CPU is more critical for faster querrying.

4.input and output components will have less impact in comparison to CPU as it manages how the storage is used in parallel.

5.improper cpu management for TDE has a signifincat impact as it may cause an over saturation of the CPUs and their load.

6.for reducing impact in terms of TDE for encrypting tablespaces for mass data migration, parallel processing can be used which will strain CPUs more in response to using more cores at once for encrypting and decrypting.

7.in terms of GPU load the TDE doesn’t use graphical processesing units by default as it mostly relies on CPU for parallel processing however using hardware acceleration it can relieve some of the stress that comes with encrypting and decrypting information such as SPARC m7 specifically for oracle databases.

8.In overall overhead the focus would be more memory to store the processes for fast access and better CPUs for faster parallel execution of queries and encryption/decryption of data.

1. How much is the overhead associated with TDE tablespace and column based encryption on different h/w resources?

1.Overhead on CPU is immense stemming from 5-8 percent but depending on the cpu itself and the amount of data it could relate to a performance decrease of 5 to 10 times more putting a more emphasis to only encrypt critical data.

2.Overhead on Memory is large with the need for larger memory to store large processes in the ready state for faster execution for encryption/decryption of data without having to rely on the storage.

3.Strain on the hard drives is smaller with most of the Input/Output execution occuring on memory due to how slow storage draws in data for the TDE to use efficiently.

4.in terms of encryption acceleration the overhead is subjective with it depending on how much the processing is split in between the main CPU and the hardware acceleration done in terms of TDE activities.

5.For the Input and Output components such as external drives and network storages the strain is dependant on the speed of the network and the amount of memory present on the local logical systems as less memory means more switching and moving processes in from the storage networks which would strain the bandwidth.

6.for software based 3rd party acceleration the overhead is considerably lower compared to standard consumer grade components due to the fact that they were design for efficiency in order to stay competive in the processesing and resource saving industry,

7.The overhead in relation to the different methods that being column and tablespace encryption, the difference is subjective but efficiency is dependant to how it is implemented with column encryption having a lower overhead initially but if it is used to encrypt a large amount of data than tablespace anyways than tablespace encryption would provide less overhead.

8.the overhead for software is extremely large as in previous version of oracle database the TDE encrypted through large blocks taking up a lot of processesing power despite working in parallel. This would limit performance in multiples of degradation in response to later versions that would help reduce the encryption requirements in terms of processesing itself such as reducing blocks and by compressesing data before encrypting.

1. Discuss Oracle’s powerful feature Online Table Redefinition which is used to encrypt Columns using Gigabytes and Terabytes of storage. [Hint: These tables are very large tables containing millions of rows, and they can be encrypted using Online Table Redefinition ]

1.Data is shrunk online through temporary storage allowing for the reduction of fragmentation in hard drives,

2.Allows for tables to be modified by the millions through the limit of bandwidth only for ensuring easier compression for encryption.

3.Helps for altering tables online faster through forcing by running parallel in terms of CPU loads for better distribution and use of server grade CPUs in terms of processesing cores.

4.more secure way of changing and encrypting large scale data with adequate amount of error checking to ensure that data has not been corrupted or if any critical information has been lost udring the alterations.

5.Allows for automatic creation of dependant objects and data mapping for easier manageability in encryption to ensure that sensitive data and the data it is related to are also properlly protected during the refractoring.

6.provide a more refined table as well as the statistics of the table itself to showcase what has been changed and to present the findings for better movement in terms of the companies efficiency for parsing through data for use.

7.dependant objects that may become invalidated due to the movement of the columns that it relies on are automatically highlighted to minimize possible downtime and availability outages for syntax errors on schema objects.

8.provides automatic clean up of errors and guides the debugging process after major redefining of tables for better processesing efficiency and better integrity and a reduction in reputation and lack of confidence in major company infrastructure overhauls.