



Scenario 1: Data Discovery and Classification

You have successfully migrated your company's databases to Azure SQL Managed Instance, and the IT Director has asked to you need to ensure sensitive data in these newly migrated databases are classified and labelled according company regulations as specified in the chart below.

Database	Table	Column	Information Type	Sensitivity Label
TeamXX_LocalMasterDataDB	Users	Username	Name	Confidential – GDPR
TeamXX_TenantDataDB	Users	Username	Name	Confidential – GDPR
TeamXX_TenantDataDB	UserTransactions	TranAmount	Financial	Highly Confidential

LAB INSTRUCTIONS

1. Using the data discovery and classification service within the azure portal set the information type and the sensitivity label, for the columns in the table above.

HINT: Check this at the database level and not the server level.

Scenario 2: Auditing

Users accessing the UserTransactions table needs to be audited because it contains sensitive information. You have been asked to ensure all DML operations on TeamXX_TenantDataDB are audited and saved in blob storage.

HINT:

You will need the following storage account for your audit logs.

'https://<Replace with Storage Account>.blob.core.windows.net/auditlogs'

For Storage Account please find C:_SQLHACK_\LABS\03-Security\StorageAccount.txt

LAB INSTRUCTIONS:

1. Using SSMS, create an Audit for you Managed Instance and define blob storage as the target path.

```
USE master ;
GO
-- Create the server audit.
CREATE SERVER AUDIT [<your_server_audit_name>]
TO URL (PATH = '<container_url>' )
GO
```

2. Create a Database Audit specification that maps to the Audit created in Step 1.

-- Create the database audit specification.

```
CREATE DATABASE AUDIT SPECIFICATION [<your_specificatio_name>]
FOR SERVER AUDIT [<your_audit_name>]
ADD (SELECT, INSERT
      ON database_name BY dbo)
WITH (STATE = ON);
GO
```

3. Enable the Audit.

-- Enable the server audit.

```
ALTER SERVER AUDIT [<your_audit_name>]
WITH (STATE = ON);
GO
```

4. Check the audit logs are being saved in the Blob storage account.

Scenario 3: Dynamic Data Masking

One of the new developers in your team, Peter, has been tasked to make changes and would need to access the database in order to test the changes. Your company policy states **only** members of the Accounting team should have visibility of the data in the TranAmount column.

Perform the steps required to mask the data in the TranAmount column from Peter.

LAB INSTRUCTIONS:

1. Connect to your Managed Instance from Management Studio.
2. Run the below script to mask the transaction amount in the UserTransactions table using the default masking function

```
ALTER TABLE [UserTransactions]
ALTER COLUMN [TranAmount] [decimal](18, 2) MASKED WITH (FUNCTION = 'DEFAULT()')
```

3. Check the table to ensure you are still able to see **ALL** the data

```
SELECT * FROM UserTransactions
GO
```

4. Create a user for the developer and grant the read only access on the UserTransactions table

```
CREATE USER Peter WITHOUT LOGIN
GRANT SELECT ON dbo.UserTransactions TO Peter;
```

5. Run the query below to ensure we have been able to prevent the developer from accessing customers' privacy information

```
EXECUTE AS USER = 'Peter';
SELECT * FROM dbo.UserTransactions;
```

```
REVERT;
```

Scenario 4: Vulnerability Assessment

Examine the risk security vulnerabilities and deviations from best practices for the databases you have migrated to SQL Managed Instance by utilizing the Vulnerability Assessments service. To complete this section you will need to assess some the highlighted vulnerabilities.

LAB INSTRUCTIONS:

Part 1: Transparent Data Encryption

1. Review the vulnerability assessment for TEAMXX_LocalMasterDataDB.
2. Using SSMS run the following to enable TDE on TEAMXX_LocalMasterDataDB

```
USE MASTER
GO
ALTER DATABASE [TEAMXX_LocalMasterDataDB]
SET ENCRYPTION ON
GO
```

Part 2: CLR assemblies

1. Review the vulnerability assessment results. The application uses CLR which has been approved by security. Using the vulnerability assessment mark the CLR result as acceptable by clicking on **“Approve as Baseline”**.

HINT: You will need to click on the CLR result to accept the approve as baseline option.

Part 3: Password

1. Review the vulnerability assessment results for “Login password should not be easily guessed”. Use the remediation script to change the password to a more secure password of your choice.
2. Re-run the vulnerability assessment ensuring the “Login password should not be easily guessed” has been removed.
3. Update the transaction reporting application with the new password.



TASK 1: DISCOVER


Data discovery & classification (currently in preview) provides advanced capabilities built into Azure SQL Database for **discovering, classifying, labeling & protecting** the sensitive data in your databases. It comes with a built-in set of sensitivity labels and a built-in set of information types and discovery logic.


In this lab, we will be exploring how to implement discovery, classification and labelling data in our managed instance database we migrated in the previous section.

1. Go to the Azure portal
2. Navigate to the Resource group created for the Hack “SCOTLANDHACK”
3. Search for the Managed Instance Databases you have migrated in the previous exercise.

<input type="checkbox"/>		TEAM01_LocalMasterDataDB (scotl...	Managed database	UK South	...
<input type="checkbox"/>		TEAM01_SharedMasterDataDB (sco...	Managed database	UK South	...
<input type="checkbox"/>		TEAM01_TenantDataDb (scotlandda...	Managed database	UK South	...


4. Select the TeamXX_LocalMasterDataDB database by clicking on it.
5. Navigate to **Advanced Data Security** under the Security heading in your SQL Database pane. Click to enable advanced data security, and then click on the **Data discovery & classification (preview)** card as shown below:

 **Data Discovery & Classification (preview)**



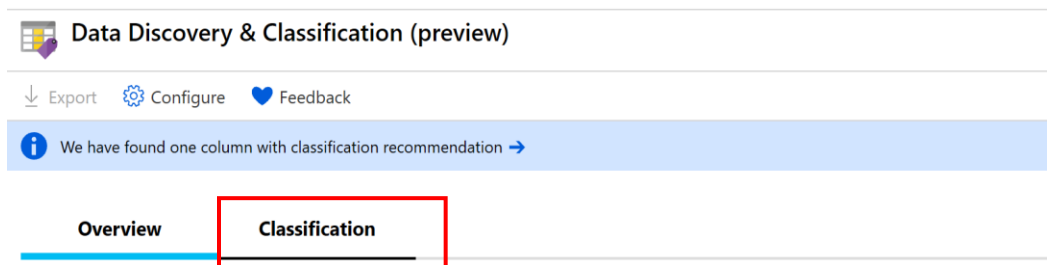
0
TOTAL

Recommended columns to classify

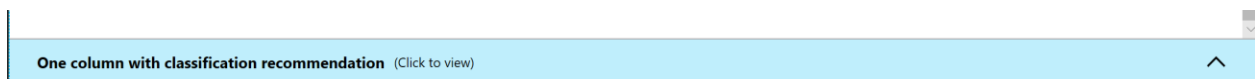
COLUMN	SENSITIVITY LABEL
 UserName	Confidential



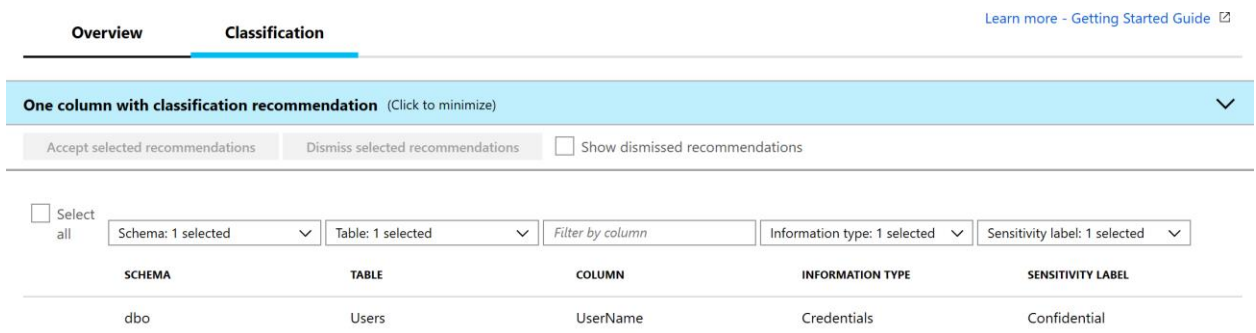
- The **Overview** tab includes a summary of the current classification state of the database, including a detailed list of all classified columns, which you can also filter to view only specific schema parts, information types and labels.
- To begin classifying your data, click on the **Classification tab** at the top of the window.



- The classification engine scans your database for columns containing potentially sensitive data and provides a list of **recommended column classifications**. To view and apply classification recommendations click on the classification recommendation as shown below:



- To view the list of recommended column classifications, click on the recommendation panel at the bottom of the window:



- Review the list of recommendations.

Accept the recommendation and modify the information type and sensitivity. To accept the recommendation for the username column in the Users table, check the checkbox in the left column of the Users row and select **Accept selected recommendations**.



11. Once the recommendation has been accepted it will show as a classified column. Change the **Information type** to **Name** and the **Sensitivity** to **Confidential – GDPR** as shown below:


Overview **Classification**





One classified column

Schema: 1 selected Table: 1 selected Filter by column Information type: 1 selected Sensitivity label: 1 selected

SCHEMA	TABLE	COLUMN	INFORMATION TYPE	SENSITIVITY LABEL
dbo	Users	UserName	Name	Confidential - GDPR

12. Click save button to commit your changes.

 Data Discovery & Classification (preview)

 Save  Discard  Add classification  Feedback

13. Perform steps 3-12 for the following additional classifications:

Database	Table	Column	Information Type	Sensitivity Label
TeamXX_LocalMasterDataDB	Users	Email	Contact Info	Confidential
TeamXX_TenantDataDB	Users	Username	Name	Confidential – GDPR
TeamXX_TenantDataDB	UserTransactions	TranAmount	Financial	Highly Confidential



Database Modernisation Workshop: Auditing

TASK 2: AUDIT

Auditing in Managed Instance is done at the server level and stores .XEL files in Azure Blob storage. In this task, we will be auditing access to the sensitive data in [TEAMXX_TenantDatabaseDB] and [TEAMXX_LocalMasterDataDB] databases which we classified and label in the previous task.

1. Create an audit for your Managed Instance and define the target

```
-- Create the server audit.  
-- Change the path to a path to Blob Storage Path for auditing  
CREATE SERVER AUDIT [TEAMXX_ScotlandDataAuditLogs]  
TO URL (PATH = 'https://scotlandauditlogs.blob.core.windows.net/auditlogs')  
GO
```

2. Create a database audit specification that maps to the audit by running the following query in Management Studio

```
USE [TEAMXX_TenantDataDb]  
GO  
-- Create the database audit specification.  
CREATE DATABASE AUDIT SPECIFICATION TEAMXX_Audit_Data --change xx to team number  
FOR SERVER AUDIT [TEAMXX_ScotlandDataAuditLogs] --change to server audit name  
created in step 1  
ADD (INSERT, UPDATE, DELETE, SELECT  
ON SCHEMA::DBO BY dbo )  
WITH (STATE = ON);  
GO
```

3. Enable the Audit

```
-- Enable the server audit.  
ALTER SERVER AUDIT [TEAMXX_ScotlandDataAuditLogs]  
WITH (STATE = ON);  
GO
```

5. Check the audit logs are being saved in the Blob storage account.



Data Modernisation Workshop: Dynamic Data Masking

TASK 3: PROTECT

Dynamic data masking helps prevent unauthorized access to sensitive data by enabling customers to designate how much of the sensitive data to reveal with minimal impact on the application layer.

In this task we will mask the transaction amount from Peter who is a developer.

6. Connect to your Managed Instance from Management Studio.
7. Run the below script to mask the transaction amount in the UserTransactions table using the default masking function

```
ALTER TABLE [UserTransactions]
ALTER COLUMN [TranAmount] [decimal](18, 2) MASKED WITH (FUNCTION = 'DEFAULT()')
```

8. Check the table to ensure you are still able to see **ALL** the data

```
SELECT * FROM UserTransactions
GO
```

9. Create a user for the developer and grant the read only access on the UserTransactions table

```
CREATE USER Peter WITHOUT LOGIN
GRANT SELECT ON dbo.UserTransactions TO Peter;
```

10. Run the query below to ensure we have been able to prevent the developer from accessing customers' privacy information

```
EXECUTE AS USER = 'Peter';
SELECT * FROM dbo.UserTransactions;
REVERT;
```




Data Modernisation Workshop: Vulnerability Assessment

Task 4: DETECT

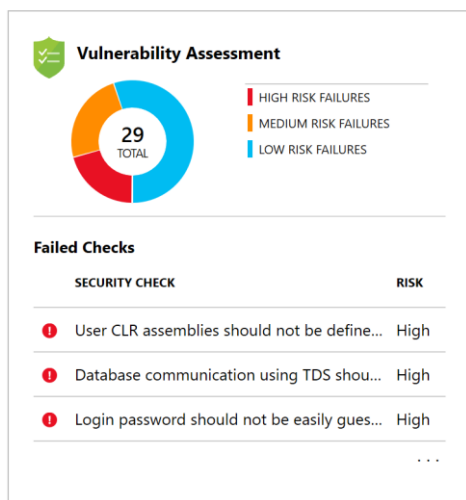
SQL Vulnerability Assessment (VA) is a service that provides visibility into your security state, and includes actionable steps to resolve security issues, and enhance your database security.

In this exercise we will be examining some of the risk security vulnerabilities and deviations from best practices.

1. Go to the Azure portal
2. Navigate to the Resource group created for the Hack "SCOTLANDHACK"
3. Search for the Managed Instance Database you have migrated in the previous exercise.

<input type="checkbox"/>		TEAM01_LocalMasterDataDB (scotl...	Managed database	UK South	...
<input type="checkbox"/>		TEAM01_SharedMasterDataDB (sco...	Managed database	UK South	...
<input type="checkbox"/>		TEAM01_TenantDataDb (scotlandda...	Managed database	UK South	...

4. Select the TeamXX_TenantDataDB database.
5. Navigate to **Advanced Data Security** under the Security heading in your SQL Database pane. Then click on the **Vulnerability assessment** card as shown below:



6. View the Vulnerability Assessment report. It would look similar to the below.



Vulnerability Assessment

Scan Export Scan Results Scan History Feedback

Total failing checks
29

Total passing checks
164

Risk summary

High Risk 6
Medium 7
Low Risk 16

Last scan time
Mon, 19 Aug 2019 18:52:18 UTC

Learn more
[SQL Security Center](#)
Best Practices for SQL Security

Failed (29)

Passed (164)

Filter by ID or security check

Category: All selected

Risk: All selected

ID	SECURITY CHECK	APPLIES TO	CATEGORY	RISK	ADDITIONAL INFO
VA12...	User CLR assemblies should not be defined in the database	TEAM01_Tenant...	Surface area redukti...	High	
VA12...	Database communication using TDS should be protected through TLS	master	Data protection	High	
VA21...	Login password should not be easily guessed	master	Authentication & A...	High	
VA21...	Execute permissions to access the registry should be revoked	master	Authentication & A...	High	
VA21...	Minimal set of principals should be members of high impact fixed server roles	master	Authentication & A...	High	Should set an initial...
VA2...	Minimal set of principals should be granted high impact database-scoped permissi...	msdb	Authentication & A...	High	Should set an initial...
VA12...	Transparent data encryption should be enabled	TEAM01_Tenant...	Data protection	Medium	
VA12...	Orphaned users should be removed from SQL server databases	TEAM01_Tenant...	Surface area redukti...	Medium	
VA21...	Minimal set of principals should be granted medium impact server-scoped permissi...	master	Authentication & A...	Medium	Should set an initial...

7. Click Description and impact for “Transparent data encryption should be enabled”

VA1219 - Transparent data encryption should be enabled

Approve as Baseline Clear Baseline

NAME	VA1219 - Transparent data encryption should be enabled
RISK	Medium
STATUS	FAIL
APPLIES TO	TEAM01_TenantDataDb
DESCRIPTION	Transparent data encryption (TDE) helps protect against the threat of malicious activity by performing real-time encryption and decryption of the database, associated backups, and transaction log files 'at rest', without requiring changes to the application. This rule checks that TDE is enabled on the database.
IMPACT	Transparent Data Encryption (TDE) protects data 'at rest', meaning the data and log files are encrypted when stored on disk.
BENCHMARK REFERENCES	<ul style="list-style-type: none">FedRAMP
RULE QUERY	<pre>FROM sys.databases WHERE name = db_name() AND is_encrypted = 0</pre>
MICROSOFT RECOMMENDATION	True
BASELINE	Not set
ACTUAL RESULT	False
REMEDIATION	Enable TDE on the affected database. Please follow the instructions on https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption

8. To enable TDE, connect to your Managed Instance via SQL Server Management Studio and execute the following query:



```
USE MASTER
GO
ALTER DATABASE [TEAMXX_LocalMasterDataDB]
SET ENCRYPTION ON
GO
```

9. Enable TDE for the other databases TEAMXX_SharedMasterDataDB and TEAMXX_TenantDataDB in your managed instance.
10. Navigate back to the Vulnerability Assessment report in Azure Portal.
11. ToClick on security check - “User CLR assemblies should not be defined in the database” and review description and remediation.
12. Review the assessment results, in this case we will mark specific results as being an acceptable *Baseline* in our environment by clicking on “Approve as Baseline”.

VA1256 - User CLR assemblies should not be defined in the database


✓ Approve as Baseline

✗ Clear Baseline


NAME	VA1256 - User CLR assemblies should not be defined in the database						
RISK	High						
STATUS	<div>✗</div> FAIL						
APPLIES TO	TEAM01_TenantDataDb						
DESCRIPTION	CLR assemblies can be used to execute arbitrary code on SQL Server process. This rule checks that there are no user-defined CLR assemblies in the database						
IMPACT	Using CLR assemblies can bring a security flaw to the SQL Server instance and to all other network resources accessible from it						
BENCHMARK REFERENCES	<ul style="list-style-type: none">FedRAMP						
RULE QUERY	<div>SELECT name FROM sys.assemblies WHERE is_user_defined != 0</div>						
MICROSOFT RECOMMENDATION	Empty Set						
RESULTS	<table><thead><tr><th>IN BASELINE</th><th>ASSEMBLY</th></tr></thead><tbody><tr><td>✗</td><td>CLRUFDS</td></tr><tr><td>✗</td><td>Database1</td></tr></tbody></table>	IN BASELINE	ASSEMBLY	✗	CLRUFDS	✗	Database1
IN BASELINE	ASSEMBLY						
✗	CLRUFDS						
✗	Database1						





13. Run the scan on the TEAMXX_LocalMasterDataDB




Vulnerability Assessment

 Scan

 Export Scan Results

 Scan History

 Feedback

14. The VA now reports *only* the security issues that deviate from your approved baseline state and the security issues which haven't been addressed.