

Exploring Organizational Security and Auditing

# Contents

Organizational security and audit	3
Scenario 1: SQL Server audit	3
Scenario 2: Contained Database Authentication	16
Scenario 3: User Defined Server Roles	22
Scenario 4: New Permissions	28
Terms of use	40

# Organizational security and audit

# Estimated time to complete lab is 30 minutes

Today, businesses require anyone within an organization to be able to access information so that they are able to best perform their role. As businesses become more closely aligned with their suppliers and customers this requirement is extending to theses as well. This places requirements to ensure that users are only able to access the appropriate information and that user access is monitored and recorded

You want to be sure that to be sure that users have access only to the data and features that they need to perform their roles. In order to meet this requirement you are going to use the following features on SQL 2014:

- Audit resilience and security
- Contained database authentication
- User defined server roles
- The new server and database permissions

You will be checking that the audit features implemented to meet compliance requirements. As a new application has been recently deployed, you are going to check that the user security has been correctly implemented as per you security policies. As part of the implementation of the security policy database permissions have been defined for the roles within the data administration team. You are going to check that these have been correctly implemented

In this exercise, you will use the following new auditing features in SQL Server 2014:

- **Audit Resilience** so that audit logs are not lost during temporary file and network issues during a failover
- **User Defined Audit** events to allow applications to write custom information to the audit log
- Audit Filtering to improve filtering to simplify audit reporting

#### **Audit Resilience**

The resilience features implemented in SQL Server Auditing means that the audit logging is now tolerant to loss of connectivity to the target directory and will recover automatically once the network connection is re-established.

# Scenario 1: SQL Server audit

You will implement a policy is that processing should continue in the event of an audit log failure. The audit process is also able to record that auditing has been paused or stopped. The policy has been implemented on a new server and Richard has been tasked to confirm that it has been implemented correctly and to check that the audit log correctly records when auditing is disabled and then re-enabled.

# **Connect to SQLONE computer**

1. Click on **SQLONE** button on right side of the screen to connect to the **SQLONE** computer. If you see the following in the lower right corner of the screen, you can jump to step 5 below to set your screen resolution.

SQLONE Logged in as LABUSER

> Windows Server 2012 R2 Datacenter Build 9600

- Click Send Ctrl-Alt-Del for SQLONE computer and then click Switch user.
- 3. Click **Send Ctrl-Alt-Del** for **SQLONE** computer again and then click **Other user**.
- Log on to SQLONE computer as labuser with password pass@word1

Note, if you have a monitor that supports a larger screen resolution than 1024 x 768, you can change the screen resolution for the lab to go as high as 1920 x 1080. By going to a higher screen resolution, it will be easier to use SQL Server Management Studio.

- 5. Right click on the desktop and click on **Screen resolution**.
- 6. Select **1366 x 786** (a good minimum screen size for using SSMS) and click **OK**.
- 7. Click Keep Changes.

8. Resize the client **holLaunchPad Online** window for the lab to fit your screen resolution.

## **Create an Audit called Monitor**

- Open SQL Server 2014 Management Studio with a connection to the SQLONE server and ensure Windows Authentication is selected before clicking Connect.
- 2. From the File menu, select Open and then File...
- 3. Browse to C:\SQLSCRIPTS\E2 and select E2C-1-#5-Audit
- 4. Click Open



5. Click **Execute** 

# Confirm that the audit was created

1. Expand Security in the Object Explorer and then Audits

NOTE: you may need to click on Refresh 🖻 to see the Audit

2. Right-click on Monitor and select Properties

Here Richard can see the properties of this audit. We see that it is set to continue operation of the server in case of an audit log failure. The audit destination (File) and file path are also defined here

🧏 Audit Properties 📃 🗖							
🕕 Ready							
Select a page	🔄 Script 🔹 📑 Help						
Filter	Audit name: Queue delay (in milliseconds): On Audit Log Failure: Audit destination: File path: Audit File Maximum Limit:	Monitor         1000					
Connection	Maximum file size:	0 ♠ ⊛ MB ○ GB ○ TB					
왕 SQLONE [contoso\labuser]	Reserve disk sp	Dace					
View connection properties							
Progress							
Ready							
		OK Cancel Help					

3. Click OK

# Temporarily disable the audit

This event will be written to the audit log, but the server will remain in operation

1. Right-click on **Monitor** and select **Disable Audit** 

8	Disab	le Audit			x
Success	1 To 1 Si	otal uccess	0	Error Warning	
Details:		Chatura		Magaza	
<ul> <li>Disable Audit 'Monitor'</li> </ul>		Success		message	
				Clos	e

2. Click Close

Wait a few seconds

3. Right-click on Monitor and select Enable Audit

٩	Enable Audit	_ <b>□</b> X
Success	1 Total 1 Success	0 Error 0 Warning
Details:		
Action	Status	Message
Enable Audit 'Monitor'	Succes	s
		Close

- 4. Click Close
- 5. Right-click on **Monitor** and select **View Audit Logs** then close after looking at the results



*Here you can see that the audit log recorded that auditing was disabled and then re-enabled* 

Server Principal Name Server Principal SID 0x15000 Database Principal Name	contoso\LabUser 00521000173184113751982117423124667208881400
Target Server Principal Name	
Target Server Principal SID	NULL
Target Database Principal Na	me
Database Name	
Schema Name	
Object Name	
Statement	
Additional Information	<pre>&lt;action_info xmlns="http://sch_emas.microsoft.com/sqlserver/20&lt;/pre&gt;</pre>
[CDATA[Monitor\$A]]> <td>n&gt;<action>event enabled</action><startup_type>manual</startup_type></td>	n> <action>event enabled</action> <startup_type>manual</startup_type>
[audit_event]]> <td>ion info&gt;</td>	ion info>
File Name C:\SQLA	UDITS\Monitor_BCE18AF0-8533-4CAD-B950-E4BBDFAA5A5D_
File Offset 5120	
User Defined Event ID	0

6. Click **Close**.

# **User Defined Audit**

The new User-defined audit events allow applications to write custom information to the audit log

As part of an auditing policy you want to be able to log employee's whose salary is increased by more than 20%. You decide to create a User-Defined Audit event. The event will be triggered whenever an employee's salary is increased by more than 20%

#### Create an Audit called LargePayIncrease

- In SQL Server 2014 Management Studio open the File menu, select Open and then File...
- 2. Browse to C:\SQLSCRIPTS\E2 and select E2C-1-#6-Audit
- 3. Click Open

```
E2C-1-#6-Audit.sql...ntoso\labuser (51)) × E2C-1-#5-Audit.sql...ntoso\labuser (84))
     1 USE [master]
     2 GO
     3 ⊡CREATE SERVER AUDIT [TestingUserDefinedEvents]
     4 TO FILE
     5 ( FILEPATH = N'C:\SQLAUDITS'
     6 ,MAXSIZE = 5 MB
     7 ,MAX_ROLLOVER_FILES = 5
     8 ,RESERVE_DISK_SPACE = OFF
     9
         )
    10
        WITH
    11 ( QUEUE_DELAY = 1000
        ,ON_FAILURE = CONTINUE
    12
    13
         )
    14
         GO
    15 ALTER SERVER AUDIT [TestingUserDefinedEvents] WITH (STATE = ON);
    16
        GO
    17 USE [AdventureWorks2012]
    18
        GO
    19 
CREATE DATABASE AUDIT SPECIFICATION [LargePayIncrease]
    20 FOR SERVER AUDIT [TestingUserDefinedEvents]
    21 ADD (USER_DEFINED_AUDIT_GROUP)
    22 WITH (STATE = ON)
    23
        GO
```

- 4. Click Execute
- Expand Databases and then AdventureWorks2012, Security, Database Audit Specifications in the Object Explorer

You will see that the Audit Report called LargePayIncrease was created.



6. Right Click on LargePayIncrease and select Properties

You will notice that the Audit Action Type is set to USER\_DEFINED\_AUDIT\_GROUP. This group tracks events raised by using the **sp\_audit\_write** stored procedure

5	Database Audit Specification Properties					
🕕 Ready						
Select a page	🕵 Script 👻 🛐 Help					
' <sup>ው</sup> General	Name: LargePayIncrease Audit: TestingUserDefinedEvents Actions:	~				
	Audit Action Type	Object Class				
	▶ 1 USER_DEFINED_AUDIT_GROUP V	<b>~</b>				
	*2	× .				
Connection						
물 SQLONE [contoso\labuser]						
View connection properties						
Progress						
C Ready	< m	>				
	OK Cancel	Help				

7. Click on **OK** 

## Use a trigger to write an audit record using sp\_audit\_write

Create a trigger for when an employee's salary is increased by 20%. This trigger uses the **sp\_audit\_write** stored procedure, hence an event will be logged when it is used and recorded in the LargePayIncrease audit

- 1. From SQL Server 2014 Management Studio select File and then Open and then File...
- 2. Browse to C:\SQLSCRIPTS\E2 and select E2C-1-#1
- 3. Click Open



4. Click **Execute** 

# Test the trigger

Now that you have created a trigger, you can now test the user defined audit event is working by running the following Script, which virtually doubles employee 4's pay rate

- From SQL Server 2014 Management Studio select File and then Open and then File...
- 2. Browse to C:\SQLSCRIPTS\E2 and select E2C-1-#2
- 3. Click Open



4. Click Execute

# Check that the pay increase was recorded

 In the SQL Server 2014 Management Studio in Object Explorer, expand Security and then Audits (if this node is already open, refresh it by right-clicking on the node and selecting Refresh)



2. Right Click on **TestingUserDefinedEvents** and select **View Audit Logs** 

You can see that the User-defined event was recorded in the logs for employee 4

	Log File Viewer -	SQLONE		_ 🗆 X			
Select logs	📴 Load Log 👌 Export 👔	Refresh 🍸 Filter	🔍 Search 🔳 Stop	📑 Help			
	Log file summary: No filter applied	ł					
TestingUserDefinedEvents	Date ⊽	Event Time	Server Instance Name	Action ID			
	✓ 7/1/2014 8:18:56 PM	20:18:56.1042284	SQLONE	USER DEFINED A			
	✓ 7/1/2014 8:10:50 PM	20:10:50.7446538	SQLONE	AUDIT SESSION			
	III       Selected row details:       Target Database Principal ID 0       Object ID 7       Session Server Principal Name contoso\LabUser       Server Principal Name dob       Database Principal Name dob       Target Server Principal Name       Target Server Principal SID       Target Server Principal SID       NULL						
Status	Schema Name Object Name Advertureur	dra2012					
Last Refresh: 7/1/2014 1:19:46 PM Filter: None View filter settings	Statement E @succeeded = 1 .@user_defined_information = @ Additional Information dta !Addrentureworks2012*schema_m 'SalaryMonitor'> File Name C:\SQLAUD 972D84C58929.0_1304871905	mszurz (EC sp_audit_write @u msg; (aj_stack>⊲frame nest_ ame = 'HumanResourc ITS\TestingUserDefine 06970000.sqlaudit	ser_defined_event_id = 27 level = '2' database_name es' object_name = dEvents_89C7D5EA-6F3/	'. = A-4A48-9401- ≡			
Progress	File Unset 6 144						
Done (2 records).	User Defined Information En Message	nployee 4 pay rate incre	eased more than 20%				
				Close			

3. Click Close.

# **Audit Filtering**

Audit Filtering allows the filtering of unwanted audit events before they are written to the audit log

As part of an auditing policy, you need to be able to log which users are accessing selected tables which contain sensitive data. To do this the you will create an Audit Filter which records the table, user, date and time when the table was accessed

First, you will create an Audit called Payole\_Security\_Audit

- In SQL Server 2014 Management Studio from the File menu, select Open and then File...
- 2. Browse to C:\SQLSCRIPTS\E2 and select E2C-1-#4-Audit
- 3. Click Open

```
E2C-1-#4-Audit.sql...ntoso\labuser (55)) × E2C-1-#2.sql - SQL...ntoso\labuser (56))
     1 USE master ;
     2 GO
     3 -- Create the server audit
     4 ⊡CREATE SERVER AUDIT Payrole Security Audit
     5 TO FILE (FILEPATH = 'C:\SQLAUDITS' );
     6
        GO
     7 -- Enable the server audit
     8 ALTER SERVER AUDIT Payrole_Security_Audit WITH (STATE = ON);
     9 GO
    10 -- Move to the target database
    11 USE AdventureWorks2012 ;
    12 GO
    13
        -- Create the database audit specification
    14 CREATE DATABASE AUDIT SPECIFICATION Audit_Pay_Tables
    15 FOR SERVER AUDIT Payrole Security Audit
    16 ADD (SELECT , INSERT ON HumanResources.EmployeePayHistory BY dbo )
    17 WITH (STATE = ON) ;
    18 60
```

4. Click **Execute** 

Confirm that the audit was created.

 Expand Databases and then expand AdventureWorks2012, Security, Database Audit Specifications in the Management Studio Object Explorer (if this node is already open, refresh it by right-clicking on the node and clicking Refresh)

There's an Audit Report called Audit\_Pay\_Tables



6. Right Click on Audit\_Pay\_Tables and select Properties

*Richard notices that if any user uses SELECT in the EmployeePayHistory table it will be recorded in the Audit logs* 

Name:		Audit_Pay_Tables						
Audit:		Payrole_Security_Audit						
Actions								
	Audit	Action Type		Object Clas	s	Object	Object Name	
1	INSERT		~	OBJECT	~	HumanResou	EmployeePay	
▶ 2	SELECT		~	OBJECT	~	HumanResou	EmployeePay	
*3			~		~			

7. Click OK

Test the Audit is working by running the following SELECT query

- 8. From **SQL Server 2014 Management Studio** select **File** and then **Open** and then **File...**
- 9. Browse to C:\SQLSCRIPTS\E2 and select E2C-1-#3

This script selects information from the EmployeePayHistory table, so should cause an entry in the audit log.

10. Click Open

```
E2C-1-#3.sql - SQL...ntoso\labuser (58)) × E2C-1-#4-Audit.sql...ntoso\labuser (55))
      1 USE Adventureworks2012
      2
         GO
     3 
SELECT TOP 10 [BusinessEntityID]
               ,[RateChangeDate]
     4
      5
                ,[Rate]
      6
                ,[PayFrequency]
      7
          ,[ModifiedDate]
      8 FROM [Adventureworks2012].[HumanResources].[EmployeePayHistory]
      9
         60
```

11. Click **Execute** 

Check that the query was recorded in the Audit logs

 Under the SQLONE connection, expand Security and then Audits (if this node is already expanded, right-click and select Refresh)



13. Right-click on Payrole\_Security\_Audit and select View Audit Logs

Log file summary: No filter applied							
Date 🔻		Event Time	Server Instance Name	Action ID			
✓ 7/1/2014 8:3	34:46 PM	20:34:46.4350102	SQLONE	SELECT			
7/1/2014 8:2	29·32 PM	20.29.32 0081008	SQLONE	AUDIT SESSION O			
		20.20.02.000.000	00,20112				
< 111				>			
Selected row details:							
Date 7/	/1/2014 8:34	:46 PM					
Log Au	udit Collection	n (Payrole_Security_A	udit)				
Event Time 20	<u>):34:46.4350</u>	102					
Server Instance Name	SQL	ONE					
Action ID St		16					
Sequence Number 1	IAD	LC					
Succeeded	Тлие						
Permission Bit Mask Ov	,0000000000	000001					
Column Permission Tr	100000000000000000000000000000000000000	1000001					
Session ID	58						
Server Principal ID 26	50			≡			
Database Principal ID	1						
Target Server Principal	ID Ó						
Target Database Princ	ipal ID 0						
Object ID 14	493580359						
Session Server Princip	al Name	contoso\LabUs	er				
Server Principal Name	cont	oso\LabUser					
Server Principal SID 0x	(1500000521	0001/3184113/5198	321174231246672088814	400			
Database Principal Na	ime dbo						
Target Server Principal	I Name						
Target Server Principal	I SID NUL	L					
Target Database Princ	ipal Name						
Database Name Ad	dventurework	cs2012					
Schema Name Hu	umanResour	ces					
Dbject Name En	nployeePayF	listory	E IN INI				
otatement SELECT TOP 10 [BusinessEntityID]							
,[RateChangeDate	1						
,[Kate]							
[PayFrequency]							
InvolutiedDate1				Ť			

You can see that when the SELECT query was run on the Table EmployeePayHistory, the user, date, and time was recorded

14. Close everything without saving

# Scenario 2: Contained Database Authentication

A contained database includes all database settings and metadata required to define the database and has no configuration dependencies on the instance of the Database Engine where the database is installed. Users can connect to the database without authenticating a login at the Database Engine level. Isolating the database from the Database Engine makes it possible to easily move the database to another instance of SQL Server

In this exercise, you will create a new Contained Database that user John has read only access.

# **Enable server for contained databases**

- Ensure you have SQL Server 2014 Management Studio open with a connection to the SQLONE server (if you do not, open SQL Server 2014 Management Studio from the Windows Start screen, enter Database Engine... as the server type, SQLONE as the Server name and ensure Windows Authentication is selected before clicking Connect.)
- Right-click on SQLONE in the Object Explorer and select Properties
- 3. Select **Advanced** and then change **Enable Contained Databases** to **True**

8	Server Properties - SQLONE	_ <b>□</b> ×
Select a page	🕄 Corint 📼 🔀 Holp	
😭 General		
Memory Processors Security		
Connections	Containment     Enable Contained Databases     Take	
🚰 Database Settings		
Advanced	FILESTREAM Access Level     Full access enabled	
Permissions	FILESTREAM Share Name MSSOL SERVER	
	4 Miscellaneous	
	Allow Triggers to Fire Others True	
	Blocked Process Threshold 0	
	Cursor Threshold -1	
	Default Full-Text Language 1033	
	Default Language English	=
	Full-Text Upgrade Option Import	
	Max Text Replication Size 65536	
	Optimize for Ad hoc Workloads False	
	Scan for Startup Procs False	
	Two Digit Year Cutoff 2049	
Connection	△ Network	
Connection	Network Packet Size 4096	
Server:	Remote Login Timeout 10	
SQLONE	4 Parallelism	
Connection:	Cost Threshold for Parallelism 5	
contosoNabuser	Locks 0	×
View connection properties	Enable Contained Databases Enables or disables contained databases and authentication across the s	service instance.
Progress		
C) Ready	Configured values     O Running values	
		OK Cancel

4. Click OK

5. Right-click on **Databases** in the **Object Explorer** and select **New Database** 

3		New Database 📃 🗖 🗖								
Select a page	🔄 Script 🔻 🛐	Script 🔻 🖪 Help								
Filegroups	Database name:			ContainedD	)atabase					
	Owner:			<default></default>						
	Use full-text in	dexing								
	Database files:									
	Logical Name	File Type	Fileg	group	Initial Size (MB)	Autogrowth / N	laxsize	;		
	ContainedD	ROWS	PR	MARY	4	By 1 MB, Unlim	ited			
	ContainedD	LOG	Not	Applicable	1	By 10 percent,	Unlimi	ted		
Connection										
Server: SQLONE										
Connection: contosoVabuser										
View connection properties										
Progress										
Ready	٢	Ш			Add		Rem	ove	>	
						ОК		Cance	<mark>ار</mark>	

6. In Database name enter **ContainedDatabase** 

- 7. Click OK
- 8. Click on **Refresh in Object Explorer**.

# Set the database as being contained

- 1. Expand **Databases**
- 2. Right-click on **ContainedDatabase** in the **Object Explorer** and then select **Properties**

+	🔋 Archive (Recovery	Archive (Recovery Pending)					
+	Contained Databas	Contained Database					
+	ContainedDatabas	P					
+	🧻 ContosoPhoneTw		New Database				
+	间 ContosoRetailDW		New Query				
+	间 ContosoRetailDWE		Script Database as				
+	🧻 ContosoRetailExpo						
+	🧻 DataAssets		Tasks 🕨				
+	间 MDS_Demo		Policies •				
+	间 RemoteBlobStorag		F i				
+	间 ReportServerDB		Facets				
+	间 ReportServerDBTe		Start PowerShell				
+	🧻 Sandbox						
+	Ifs_Configuration		Reports •				
+	Ifs_DefaultCollect		Rename				
+	🧃 Tfs_LabuserProjec		Delete				
+	间 Tfs_Warehouse						
+	Security		Refresh				
+	Server Objects		Properties				
+	Replication						

- 3. Click on the **Options** page
- 4. Change **Containment type** to **Partial**

Database Properties - ContainedDatabase			x
🔄 Script 🔻 📑 Help			
Collation:	SQL_Latin 1_General_CP1_CI_AS		~
Recovery model:	Full		~
Compatibility level:	SQL Server 2014 (120)		~
Containment type:	Partial		~
Other options:	None Partial		

5. Click OK

# Add the user John to the database

- 1. Expand the **ContainedDatabase** node in the **Object Explorer** and then **Security** and then **Users**
- 2. Right-click on **Users** and select **New User**
- In User name enter John and in Password enter pass@word1.
   Ensure the User type is SQL user with password and enter the same password in Confirm Password dialog

Database User - New	-	x
🔄 Script 🔻 🛐 Help		
User type:		
SQL user with password		~
User name:		 
John		
Password:		
•••••		
Confirm password:		
•••••		
Specify old password		

4. Click on Membership and tick db\_datareader

Ū	Database User - New
Select a page General Owned Schemas Membership Securables Extended Properties	Database User - New         Script
	<ul> <li>db_ddladmin</li> <li>db_denydatareader</li> <li>db_denydatawriter</li> <li>db_owner</li> <li>db_securityadmin</li> </ul>

# 5. Click OK

You now created a contained Database that John has read access to NOTE: John does not have a SQL login account, he only has a logon account to the database called 'ContainedDatabase'

# Test John's user with the contained database

- Click on **Connect** at the top of the **Object Explorer** and then Database Engine...
- 2. Ensure Authentication is set to **SQL Server Authentication** and the Server Name is **SQLONE**
- 3. In Login enter John and in Password enter pass@word1

s <sup>ji</sup>	Connect to Server	x
Microsoft	SQL Server 2014	
Server type:	Database Engine	~
Server name:	SQLONE	~
Authentication:	SQL Server Authentication	~
Login:	John	~
Password:	*****	
	Remember password	
Con	nnect Cancel Help Op	tions >>

- 4. Click on **Options**
- 5. Select the Tab **Connection Properties** and in **Connect to database** enter **ContainedDatabase**

Connect to Server					x
Microsoft SQL Server 2014					
Login	Connection Properties	Addition	al Connectio	on Parameters	
Туре	or select the name of the	databas	e for the con	nection.	
Conn	ect to database:		Contained	Database	-
Netw	vork				_
1	Network protocol:		<default></default>		-
1	Network packet size:		4096	; bytes	
Conn	ection				-
0	Connection time-out:		15	seconds	
E	Execution time-out:		0	seconds	
	Encrypt connection				
	Use custom color:			Select	
				Reset All	
	Connect	Cancel	Н	elp Options <	<

## 6. Click **Connect**

7. Expand Databases

Notice how John was able to access the Contained Database without having a SQL login account



8. Close all windows

# Scenario 3: User Defined Server Roles

User-Defined Server Roles increases flexibility and manageability, and it facilitates compliance through clearer separation of duties. It allows creation of server roles to suit different organizations that separate administrative duties according to roles. Roles also can be nested to allow more flexibility in mapping to hierarchical structures in organizations. User-Defined Server Roles also helps prevent organizations from using sysadmin (sa) for database administration. For example, a special database administration role can be created for common database administration without the ability to access user data

You will create a server role called PerfCheck for user Peter who needs to monitor SQL server performance. This role is allowed to see all the databases but not the data in them.

#### **Create the new role PerfCheck**

- Ensure you have SQL Server 2014 Management Studio open with a connection to the SQLONE server (if you do not, open SQL Server 2014 Management Studio from the Windows Start screen, enter Database Engine... as the server type, SQLONE as the Server name and ensure Windows Authentication is selected before clicking Connect.)
- 2. Expand Security in the Object Explorer
- 3. Right click on Server Roles and select New Server Roles...



4. In Server Role Name enter PerfCheck

5. Place a tick in **Servers** and then expand **Servers** and make certain **SQLONE** is selected

New Server	Role - PerfCheck		x
🔄 Script 🕞 📑 Help	,		
Server role name:	PerfCheck		
Owner:		 	
Securables:			
Endpoints			
E Logins			
Servers			
SQLON			
Availability	Groups		
Server roles			

- 6. Grant the following Permissions by placing a tick in the boxes under the **Grant** title next to:
  - Alter Trace

#### Permissions for SQLONE:

Explicit				
Permission	Grantor	Grant	With Grant	~
Alter server state				
Alter settings				
Alter trace				
Authenticate server				
Connect Any Database				
<	III			~

#### • Create DDL event notification

Explicit			
Permission	Grantor	Grant	With Grant \land
Create availability group			
Create DDL event notification		<ul><li>✓</li></ul>	
Create endpoint			
Create server role			

• View Server State

Explicit			
Permission	Grantor	Grant	With Grant \land
Shutdown			
Unsafe assembly			
View any database			
View any definition			
View server state		<b></b>	
<	III		>

# 7. Click on **Members**

5	New Server Role - PerfCheck
🛈 Ready	
Select a page	🖳 Script 👻 🎼 Help
General Members Memberships	Server role name: PerfCheck
	Members of this role:
	Role Members
Connection	
V SQLONE [contoso\labuser]	
View connection properties	
Progress	
Ready	Add Remove
	OK Cancel Help

- 8. Click **Add...**
- 9. In Enter the object names to select enter Peter

٩,	Select Server Login or Role	×
Select these object types:		
Logins, Server roles		Object Types
Enter the object names to select (exar	nples):	]
Peter		Check Names
		Browse
	ОК	Cancel Help

- 10. Click **OK**
- 11. Click **OK**

# Check out the use of the new role for user Peter

Now that you have created the new role, you are going to check that it is working

- Click on **Connect** at the top of the **Object Explorer** then select **Database Engine**
- 2. Ensure Authentication is set to **SQL Server Authentication** and the Server Name is **SQLONE**
- 3. In Login enter Peter and in Password enter pass@word1

e <sup>ji</sup>	Connect to Server	x
Microsoft S	QL Server 2014	
Server type:	Database Engine	~
Server name:	SQLONE	~
Authentication:	SQL Server Authentication	~
Login:	Peter	~
Password:	********	
	Remember password	
Conne	ct Cancel Help Opt	ions >>

- 4. Click Connect
- 5. Expand Databases and then try to expand the database **ContosoSalesDB**

Notice the error message that pops up

	Microsoft SQL Server Management Studio	X
8	The database ContosoSalesDB is not accessible. (ObjectExplorer)	
Þa 🍄		ОК

The role PerfCheck is allowed to see all the databases, but is not allowed access to the data in them. However the role does allow appropriate operations to monitor SQL server performance without the user requiring sysadmin (sa) rights

6. Click OK

# Test user Peter for ability to perform a SQL Trace

- 1. From the menu click on **Tools** and then select **SQL Server Profiler**
- 2. Ensure Authentication is set to SQL Server Authentication
- 3. In Login enter **Peter** and in **Password** enter **pass@word1**

e <sup>j</sup>	Connect to Server				
Microsoft S	QL Server 2014				
Server type:	Database Engine				
Server name:	SQLONE				
Authentication:	SQL Server Authentication				
Login:	Peter 💌				
Password:					
	Remember password				
Connec	ct Cancel Help Options >>				

- 4. Click Connect
- 5. Click Run to start a Trace

Notice how the traces are captured from all databases in the server, even though the role PerfCheck does not have sysadmin (sa) rights and access to user data

<b>a</b>	Untitled - 1 (SQLONE)			- <b>-</b> ×	ĸ
EventClass	TextData	ApplicationName	NTUserName	LoginName 🗸	~
Audit Login	network protocol: LPC set quote	.Net SqlClie	NETWORK	NT AUT	
RPC:Completed	exec prc_QueryNotifications @lastEv	.Net SqlClie	NETWORK	NT AUT	
Audit Logout		.Net SqlClie	NETWORK	NT AUT	
RPC:Completed	exec sp_reset_connection	.Net SqlClie	NETWORK	NT AUT	
Audit Login	network protocol: LPC set quote	.Net SqlClie	NETWORK	NT AUT	
RPC:Completed	exec prc_QueryNotifications @lastEv	.Net SqlClie	NETWORK	NT AUT	
Audit Logout		.Net SqlClie	NETWORK	NT AUT	
RPC:Completed	exec sp_reset_connection	.Net SqlClie	NETWORK	NT AUT =	=
Audit Login	network protocol: LPC set quote	.Net SqlClie	NETWORK	NT AUT	
RPC:Completed	exec prc_QueryNotifications @lastEv	.Net SqlClie	NETWORK	NT AUT	
				~	~
<				>	
exec prc_QueryNotifications @last	tEventId=154195				7
				=	-
				×	~
<	Ш			>	
Trace is running.			Ln 19, Col 1	Rows: 19	1

- 6. Stop the Trace by clicking on
- 7. Close the SQL Server Profiler
- 8. Close all windows

# Scenario 4: New Permissions

You are going to use the following new permissions available in SQL Server 2014 that can be used to limit the access of a person assigned the sysadmin role.:

- CONNECT ANY DATABASE
- SELECT ALL USER SECURABLES
- IMPERSONATE ANY LOGON

# **CONNECT ANY DATABASE**

A new server level permission. Grant CONNECT ANY DATABASE to a user that must connect to all databases that currently exist and to any new databases that might be created in future. Does not grant any permission in any database beyond connect.

Previously, any increase in the number of databases that the data administration team are required to support has required either granting server level role or creating a login for the administrator on each database. However now with the CONNECT ANY DATABASE this is no longer required as this grants the user the ability to connect and view any database on the server. With a new data administrator (Robert) soon to start you decide to use CONNECT ANY DATABASE to give him access to the databases on the server.

You will grant Robert the CONNECT TO ANY DATABASE permission

- Ensure you have SQL Server 2014 Management Studio open with a connection to the SQLONE server (if you do not, open SQL Server 2014 Management Studio from the Windows Start screen, enter Database Engine... as the server type, SQLONE as the Server name and ensure Windows Authentication is selected before clicking Connect.)
- 2. Expand **Security** and then **Logins** in the **Object Explorer**



- 3. Right-click on the login Robert and select Properties
- 4. Click on **Securables** and then grant the permission **Connect Any Database**

8	Login Pro	operties - Robert			-		х
Select a page	🛒 Script 🔻 📑 Help						
General	Logio parro: Robert						
Securables	Communities				Search		
Im Status	Securables.				500rom		
				lype	_		0.0
	SQLONE			Server	Г		2
Connection	Permissions for SQLONE:						
Server:	Explicit Effective						
SQLONE	Permission	Grantor	Grant	With Grant	De	eny	~
Connection:	Alter trace						
	Authenticate server						
View connection properties	Connect Any Database		<ul><li>✓</li></ul>				
-	Connect SQL						
Progress	Connect SQL	sa	✓				
Ready	Control server						
"east	Crosto sou databaso					5	
						1.4	
				ОК	С	ance	4

5. Click OK

# Test the CONNECT TO ANY DATABASE permission is working for Robert

- 1. From SQL Server 2014 Management Studio in Object Explorer, click on Connect to SQLONE and then Database Engine...
- 2. Ensure Authentication is set to **SQL Server Authentication** and in Login enter **Robert**
- 3. In Password enter Password1

e	Connect to Server			
Microsoft SC	QL Server 2014			
Server type:	Database Engine	~		
Server name:	SQLONE	¥		
Authentication:	SQL Server Authentication	¥		
Login:	Robert	*		
Password:	********			
	Remember password			
Connect	t Cancel Help Options	>>		

- 4. Click **Connect**
- 5. Expand **Databases**
- 6. Expand **ContosoSalesDB**
- 7. Expand **Tables**



Although Robert is not a sysadmin (sa) he is able to see all the Databases with the CONNECT ANY DATABASE permission, but he cannot see the data in the databases

8. Right click on **SQLONE** for Robert in the **Object Explorer** and click the **Disconnect** command.



# SELECT ALL USER SECURABLES

This is an extension of the CONNECT ANY DATABASE permission so that the user is also able to see the data in the databases.

You will use Select All User Securables permission to allow user Robert to view data.

- Right click on Robert login under Security -> Logins in Object Explorer and click on Properties.
- 2. Click on **Securables** and then grant the permission **Select All User Securables**

8	Login Propert	ies - Robert		-		x
Select a page Price General	🔄 Script 🔻 📑 Help					
Server Roles	Login name: Robert					
Status	Securables:			Searc	h	
_	Name		Туре			
	SQLONE		Serv	er		2
Converting.	President for SOLONE.					
Connection	Evaliat Fr					
SQLONE	Permission	Granter	Grant	th Gran		E a l
Connection:	Impersonate Any Login	Gianto			n.	
contoso Vabuser	Select All User Securables		✓			
View connection properties	Shutdown					
	Unsafe assembly					
Progress	View any database					
Ready	View any definition					
A CONTRACT OF A	View econociata	Ш			3	
			OK		Cance	el

- 3. Click OK
- 4. From SQL Server 2014 Management Studio in Object Explorer, click on Connect to SQLONE and then Database Engine...
- 5. Ensure Authentication is set to **SQL Server Authentication** and in Login enter **Robert**
- 6. In Password enter **Password1**

e <sup>j</sup>	Connect to Server			
Microsoft SC	QL Server 2014			
Server type:	Database Engine	~		
Server name:	SQLONE	~		
Authentication:	SQL Server Authentication	~		
Login:	Robert	~		
Password:	*******			
	Remember password			
Connect	Cancel Help Option	s >>		

- 7. Click **Connect**
- 8. Expand **Databases**
- 9. Expand ContosoSalesDB
- 10. Expand Tables

As Robert now has the permissions CONNECT ANY DATABASE & SELECT ALL USER SECURABLES he can now see all the data in the databases



# **IMPERSONATE ANY LOGIN**

The EXECUTE AS statement allows the execution context of a session to be switched to the specified login or user name. However for this to work the user that is executing the EXECUTE AS statement must have the sysdamin (sa) server role. The new IMPERSONATE ANY LOGON permission removes this restriction

To reduce security concerns the data administration team are keen to reduce the number of users that are members of the sysadmin (sa) role. You decide to give Robert the new IMPERSONATE ANY LOGIN permission and is also going to test that it is working

- 1. In **Object Explorer** and select the **Robert** connection.
- Expand Databases in the Object Explorer under the connection for Robert and then right-click on ContosoSalesDB and select New Query
- 3. In the query window enter the following script

Execute as user = 'Susan'

4. Click Execute

Notice the error message stating that the principal cannot be impersonated. As Robert is not a sysadmin (sa) and does not have the new IMPERSONATE ANY LOGIN permission he is not able to impersonate Susan



You will now give Robert the IMPERSONATE ANY LOGIN permission

 In the Object Explorer, under the main SQLONE connection expand Security and then Logins (if this connection does not exist, open a new connection dialog box and connect to the SQLONE database engine using Windows Authentication.)



- 6. Right-click on the user Robert and select Properties
- 7. Click on **Securables** and then Grant Robert the permission **IMPERSONATE ANY LOGIN**

B	Login Propertie	es - Robert	_ 🗆 X
Select a page Page General	🔄 Script 🔻 🛐 Help		
Server Roles	Login name: Robert		
Securables	Securables:		Search
_	Name		Туре
	SQLONE		Server 😭
Connection	Permissions for SQLONE:		
Server: SQLONE	Explicit Effective		
Connection:	Permission Granto	or Grant	With Grant Deny A
contosoVabuser	External access asse		
View connection properties	Impersonate Any Login		
	Select All User Secura		
Progress	Select All User Secura sa		
Ready	Shutdown		
No. of Concession, Name	Upasfa sasambly		
		III	
			OK Cancel

- 8. Click OK
- 9. Go back to your query window (in step 1) and Click **Execute** again

SQLQuery4.sql - SQntoso\labuser (59))* × SQLQuery3.s
1 Execute as user = 'Susan'
100 % - <
Messages
Command(s) completed successfully.

As Robert now has IMPERSONATE ANY LOGIN permission he was able to impersonate Susan without having sysadmin (sa) rights.

# Terms of use

© 2014 Microsoft Corporation. All rights reserved. By using this Hands-on Lab, you agree to the following terms:

The technology/functionality described in this Hands-on Lab is provided by Microsoft Corporation in a "sandbox" testing environment for purposes of obtaining your feedback and to provide you with a learning experience. You may only use the Hands-on Lab to evaluate such technology features and functionality and provide feedback to Microsoft. You may not use it for any other purpose. You may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this Hands-on Lab or any portion thereof.

COPYING OR REPRODUCTION OF THE HANDS-ON LAB (OR ANY PORTION OF IT) TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION IS EXPRESSLY PROHIBITED.

THIS HANDS-ONLAB PROVIDES CERTAIN SOFTWARE TECHNOLOGY/PRODUCT FEATURES AND FUNCTIONALITY, INCLUDING POTENTIAL NEW FEATURES AND CONCEPTS, IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION FOR THE PURPOSE DESCRIBED ABOVE. THE TECHNOLOGY/CONCEPTS REPRESENTED IN THIS HANDS-ON LAB MAY NOT REPRESENT FULL FEATURE FUNCTIONALITY AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK. WE ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH FEATURES OR CONCEPTS. YOUR EXPERIENCE WITH USING SUCH FEATURES AND FUNCITONALITY IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

**FEEDBACK**. If you give feedback about the technology features, functionality and/or concepts described in this Hands-on Lab to Microsoft, you give to Microsoft, without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your feedback in them. These rights survive this agreement.

MICROSOFT CORPORATION HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE HANDS-ON LAB, INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. MICROSOFT DOES NOT MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, OUTPUT THAT DERIVES FROM USE OF THE VIRTUAL LAB, OR SUITABILITY OF THE INFORMATION CONTAINED IN THE VIRTUAL LAB FOR ANY PURPOSE.

## DISCLAIMER

This lab contains only a portion of new features and enhancements in Microsoft SQL Server 2014. Some of the features might change in future releases of the product. In this lab, you will learn about some, but not all, new features.