SQL Server Database Security Audit

ISACA Denver Chapter
January 2016

Agenda

- Introduction
- Learning objectives
- Definitions
- SQL Server architecture
- Defaults
- Permissions model
- Security standards
- Authentication
- Performing an audit
- Q&A

Introduction - Who am I?

- Sr IT Audit Leader - Wells Fargo Audit Services
- Past life …
  - Information system security officer (ISSO)
  - Database and application security engineer
  - Oracle/SQL Server/Sybase DBA and developer
  - C language programmer
    - CP/M
    - MS-DOS
    - VAX/VMS
    - Mac OS 7
    - UNIX

Caveats

- Always get permission to run any scripts in your environment
- Always test scripts in a non-production environment before using them in production
  - Vet the scripts with your IT DBA team
- The testing methodology is my own
- The scripts used in the presentation are working in a test environment (I also use them in my present position to execute audit testing)

Learning objectives

- SQL Server architecture
- Data access model for SQL Server
- Understanding of
  - Security containers model
  - Server and database level roles
  - Sweeping security roles
- Database audit trail

Preliminaries

The big picture

- Database security is all about …
  - Data security and access
- How data is protected by …
  - The installation and configuration of the software
  - Access to the data and log files that contain data through operating system permissions
  - Patching
  - System privileges
  - RBAC application design
  - Direct (and denied) access
  - Auditing and monitoring

Definitions

Database instance

- SQL Server is a Windows service which manages a group of databases
  - Disk files
  - Memory
  - Network connections
  - User processes

Database

- Logical container
- Security boundary

Principal

- Entities that can request and use resources
- Can be arranged in a hierarchy
  - Windows-level
    - Windows domain login
    - Windows local login
  - SQL Server-level
    - SQL Server login
  - Database-level
    - Database user
    - Database role
    - Application role

Securables

Server scope

- Access to resources which SQL Server authorization regulates
  - End point
  - Login
  - Database

Database scope

- User
- Role
- Application role
- Assembly
- Message type
- Route
- Service
- Remote Service binding
- Fulltext catalog
- Certificate
- Key (asymmetric, symmetric)
- Contract
- Schema

Schema scope

- Type
- XML Schema Collection
- Object

Object scope

- Aggregate
- Function
- Procedure
- Queue
- Synonym
- Table
- View

SQL Server Architecture

Categories

Category	Description
Database Engine	The Database Engine is the core service for storing, processing and securing data. The Database Engine provides controlled access and rapid transaction processing to meet the requirements of the most demanding data consuming applications within your enterprise. The Database Engine also provides rich support for sustaining high availability.
Analysis Services - Multidimensional Data	Analysis Services supports OLAP by allowing you to design, create, and manage multidimensional structures that contain data aggregated from other data sources such as relational databases.
Analysis Services - Data Mining	Analysis Services enables you to design, create and visualize data mining models. These mining models can be constructed from other data sources by using a wide variety of industry-standard data mining algorithms.
Integration Services	Services is a platform for building high performance data integration solutions including packages that provide extract, transform, and load (ETL) processing for data warehousing.
Master Data Services	Master Data Services is the source of master data for your organization. By integrating disparate operational and analytic systems with Master Data Services you ensure that all applications across the organization rely on a central accurate source of information. Using Master Data Services you create a single source of master data and maintain an auditable record of that data as it changes over time.
Replication	Replication is a set of technologies for copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency. By using replication you can distribute data to different locations and to remote or mobile users by means of local and wide area networks dial-up connections wireless connections and the Internet.
Reporting Services	Reporting Services delivers enterprise Web-enabled reporting functionality so you can create reports that draw content from a variety of data sources publish reports in various formats and centrally manage security and subscriptions.
SharePoint Integration	SQL Server 2008 R2 offers new self-service business intelligence capability through integration with SharePoint products and technologies. In this release both Analysis Services and Reporting Services support deployment in a SharePoint farm.
Service Broker	Service Broker helps developers build scalable secure database applications. This new Database Engine technology provides a message-based communication platform that enables independent application components to perform as a functioning whole. Service Broker includes infrastructure for asynchronous programming that can be used for applications within a single database or a single instance and also for distributed applications.

Database engine components

- Relational engine
- Storage engine
- SQLOS

Relational engine

- Query processor
- All components to determine resources needed to process queries
  - Query processing
  - Memory management
  - Thread and task management
  - Buffer management
  - Distributed query processing

Storage engine

- Responsible for storage and retrieval of data to the disk storage system
- Mapped over set of operating system files
- Three types of files
  - Primary data file
  - Secondary data files
  - Log files

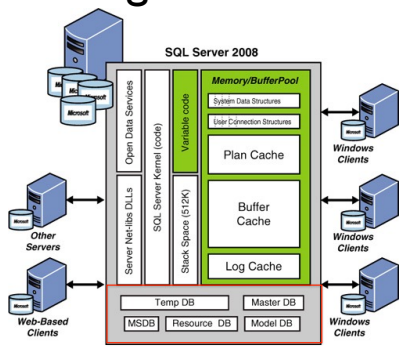
SQLOS

- Interface (API) SQL Server and Windows host operating system
- Query engine and query optimizer abstraction layer
- No special privileges or priority
- Does not bypass Windows OS
  - Memory management
  - Buffer pools
  - Log buffer
  - Deadlock detection
  - Exception handling
  - Common language runtime (CLR)
  - Scheduling

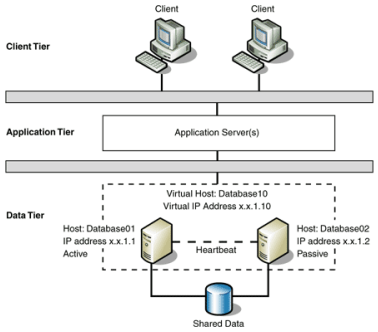
SQL Server instances

- Two types of SQL Server database engine instances
  - Single instance
  - Clustered instance
- Default instance
- Named instance

Single instance



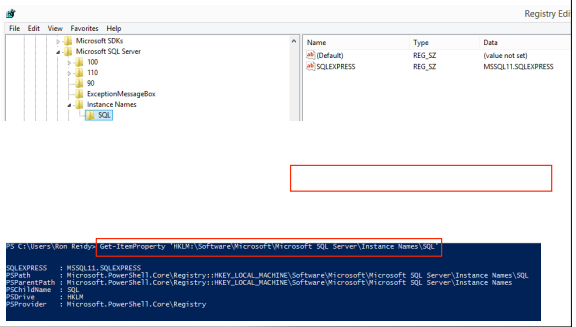
Clustered instance



Default vs. named instance

- Default instance
  - One default instance per server
  - Connect by specifying server only; port TCP: 1433
- Named instance
  - Many per server
  - Connect by specifying server and instance (Server\Instance)
  - SQL Browser service identifies and returns the port the named instance listens on

Installed instances



Disk files

- All data in a SQL Server instance is written to data files on disk
- Locations of data files specified in the data dictionary
- Check locations
- Check permissions
  - Validate access

Networking

- Ports & Protocols
- Listening port
- Browser service

Ports & protocols

- Networking communications are defined by both the port and protocol using the port
- Protocols supported
  - TCP/IP
  - Named Pipes
  - Shared Memory
  - VIA

Common ports

Description	Protocol	Port
Database Engine (default instance)	TCP	1433
Database Engine (default instance)	UDP	1434
Database Mail	SMTP	25
Database Mirroring	TCP	No official default port, but examples tend to use 5022.
Dedicated Administrative Connection (default instance)	TCP	1434
Filestream	TCP	139 and 445
Service Broker	TCP	No official default port, but examples tend to use 4022.
SQL Server Browser Service	UDP/TCP	UDP: 1434 TCP: 2382
SQL Server (default instance) over HTTP	TCP	80
SQL Server (default instance) over HTTPS	TCP	443
SQL Server Integration Services	TCP	135
TSQL Debugger	TCP	135

Listening port

- The port used by client programs to connect to the database engine.
  - Default is TCP:1433
  - Used by the default instance

Browser service

- Windows service
- Listens for incoming connection requests
  - Provides information about SQL Server instances on the server (instance name and version)

Defaults

Logins

- Many accounts created when SQL Server instance created
- sa
  - Server-level principal
- INFORMATION_SCHEMA & sys
  - Appear as users in catalog views
  - Required by SQL Server
  - Not principals
  - Cannot be modified or dropped

Certificate-based server logins

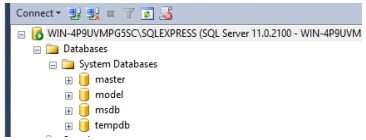
- Names enclosed in double "#"
- Created from certificates
- Should not be deleted
  - ##MS_SQLResourceSigningCertificate##
  - ##MS_SQLReplicationSigningCertificate##
  - ##MS_SQLAuthenticatorCertificate##
  - ##MS_AgentSigningCertificate##
  - ##MS_PolicyEventProcessingLogin##
  - ##MS_PolicySigningCertificate##
  - ##MS_PolicyTsqlExecutionLogin##

Service accounts

- Service accounts created during installation
  - Startup accounts used to start and run SQL Server can be domain user accounts, local user accounts, managed service accounts, virtual accounts, or built-in system accounts.
- Others can be created for use by applications

Databases

- The databases can be seen from SQL Server Management Studio
  - Expand Databases->System Databases



Default Databases

System database	Description
master Database	Records all the system-level information for an instance of SQL Server.
msdb Database	Is used by SQL Server Agent for scheduling alerts and jobs.
model Database	Is used as the template for all databases created on the instance of SQL Server. Modifications made to the model database, such as database size, collation, recovery model, and other database options, are applied to any databases created afterward.
Resource Database	Is a read-only database that contains system objects that are included with SQL Server. System objects are physically persisted in the Resource database, but they logically appear in the sys schema of every database.
tempdb Database	Is a workspace for holding temporary objects or intermediate result sets.

Default file locations

- Shared files for all instances
- <*drive*>:\Program Files\Microsoft SQL Server\100\, where <*drive*> is the drive letter where components are installed. The default is drive C.
  - Use of the C:\ drive is not advised and should be avoided.

Default Event logs

- Log size - 102400 KB
  - Default size = 102400 KB
  - Maximum size = 15168 KB
- Number of logs = 12

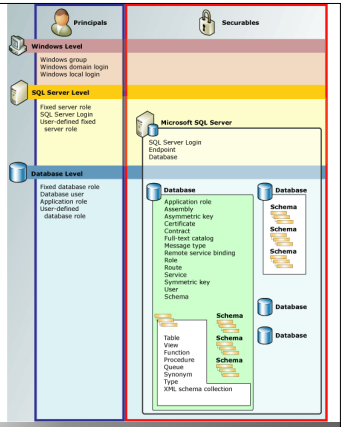
Leading practice - All of these values are too small for a production environment

Permissions model

Overview

- The permissions model is very granular
- Roles
  - Server-level roles
  - Database-level roles
    - Allow
    - Deny
  - Application roles
- Direct grants
  - Allow
  - Deny

Permissions relationships



Server-level roles

- Manage permissions of the server
  - Security principals that group other principals
- Server wide in scope
  - Similar to groups in Windows operating system
- You can add server level principals to server-level roles
  - SQL Server logins
  - Windows accounts
  - Windows groups
- Permissions granted to the server-level roles cannot be changed (fixed)

Starting in SQL Server 2012, user-defined server roles can be created

Server-level role name	Description
sysadmin	Members of the sysadmin fixed server role can perform any activity in the server.
serveradmin	Members of the serveradmin fixed server role can change server-wide configuration options and shut down the server.
securityadmin	Members of the securityadmin fixed server role manage logins and their properties. They can GRANT, DENY, and REVOKE server-level permissions. They can also GRANT, DENY, and REVOKE database-level permissions if they have access to a database. Additionally, they can reset passwords for SQL Server logins.
processadmin	Members of the processadmin fixed server role can end processes that are running in an instance of SQL Server.
setupadmin	Members of the setupadmin fixed server role can add and remove linked servers.
bulkadmin	Members of the bulkadmin fixed server role can run the BULK INSERT statement.
diskadmin	The diskadmin fixed server role is used for managing disk files.
dbcreator	Members of the dbcreator fixed server role can create, alter, drop, and restore any database.
public **NOTE: Implemented differently than other roles.**	Every SQL Server login belongs to the public server role. When a server principal has not been granted or denied specific permissions on a securable object, the user inherits the permissions granted to public on that object.

Members of the SYSADMIN and SECURITYADMIN roles should be treated as equally powerful.

Permissions of server-level roles

Fixed server role	Server-level permission
bulkadmin	Granted: ADMINISTER BULK OPERATIONS
dbcreator	Granted: CREATE ANY DATABASE
diskadmin	Granted: ALTER RESOURCES
processadmin	Granted: ALTER ANY CONNECTION, ALTER SERVER STATE
securityadmin	Granted: ALTER ANY LOGIN
serveradmin	Granted: ALTER ANY ENDPOINT, ALTER RESOURCES, ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN, VIEW SERVER STATE
setupadmin	Granted: ALTER ANY LINKED SERVER
sysadmin	Granted with GRANT option: CONTROL SERVER

Database-level roles

- Manage permissions of the databases
  - Security principals that group other principals
- Database wide in scope and exist in each database
- Any database account and any server role can be added into database-level roles
- Any member of a database-level role can add other logins to that same role
- Three different types of roles
  - Administration roles
  - Data access roles
  - Roles restricted to the msdb database

Database roles should **never** be members of fixed roles. This could enable unintended privilege escalation.

Administration roles

Database-level role name	Description
db_owner	Members of the **db_owner** fixed database role can perform all configuration and maintenance activities on the database, and can also drop the database.
db_securityadmin	Members of the **db_securityadmin** fixed database role can modify role membership and manage permissions. **Adding principals to this role could enable unintended privilege escalation.**
db_accessadmin	Members of the **db_accessadmin** fixed database role can add or remove access to the database for Windows logins, Windows groups, and SQL Server logins.
db_backupoperator	Members of the **db_backupoperator** fixed database role can back up the database.
db_ddladmin	Members of the **db_ddladmin** fixed database role can run any Data Definition Language (DDL) command in a database.

Data access roles

Database-level role name	Description
db_datawriter	Members of the **db_datawriter** fixed database role **can** add, delete, or change data in all user tables.
db_datareader	Members of the **db_datareader** fixed database role **can** read all data from all user tables.
db_denydatawriter	Members of the **db_denydatawriter** fixed database role **cannot** add, modify, or delete any data in the user tables within a database.
db_denydatareader	Members of the **db_denydatareader** fixed database role **cannot** read any data in the user tables within a database.

msdb specific roles

msdb role name	Description
db_ssisadmin db_ssisoperator db_ssisltduser	Members of these database roles can administer and use SSIS.
dc_admin dc_operator dc_proxy	Members of the **db_datareader** fixed database role can read all data from all user tables.
PolicyAdministratorRole	Members of the **db_ PolicyAdministratorRole** database role can perform all configuration and maintenance activities on Policy-Based Management policies and conditions.
ServerGroupAdministratorRole ServerGroupReaderRole	Members of these database roles can administer and use registered server groups.
dbm_monitor	Created in the **msdb** database when the first database is registered in Database Mirroring Monitor.

Members of the db_ssisadmin role and the dc_admin role may be able to elevate their privileges to sysadmin. This elevation of privilege can occur because these roles can modify Integration Services packages and Integration Services packages can be executed by SQL Server using the sysadmin security context of SQL Server Agent.

PUBLIC

- All principals belong to the database role PUBLIC
  - When a user has not been granted or denied specific permissions on a securable object, the user inherits the permissions granted to **public** on that object.

High risk system privileges

- All server-level roles are granted specific (fixed) privileges
- Many other high risk privileges

Permission Name	Permission State	Principal Name	Principal Type	Principal Is Disabled
ALTER ANY AVAILABILITY GROUP	GRANT	NT AUTHORITY\SYSTEM	WINDOWS_LOGIN	FALSE
CONNECT SQL	GRANT	BUILTIN\Users	WINDOWS_GROUP	FALSE
CONNECT SQL	GRANT	NT AUTHORITY\SYSTEM	WINDOWS_LOGIN	FALSE
CONNECT SQL	GRANT	NT Service\MSSQL$SQLEXPRESS	WINDOWS_LOGIN	FALSE
CONNECT SQL	GRANT	NT SERVICE\SQLWriter	WINDOWS_LOGIN	FALSE
CONNECT SQL	GRANT	NT SERVICE\Winmgmt	WINDOWS_LOGIN	FALSE
CONNECT SQL	GRANT	sa	SQL_LOGIN	TRUE
CONNECT SQL	GRANT	WIN-4P9UVMPG5SC\Ron Reidy	WINDOWS_LOGIN	FALSE
VIEW SERVER STATE	GRANT	NT AUTHORITY\SYSTEM	WINDOWS_LOGIN	FALSE

Application roles

- Allows application to run with user-like permissions
- Enable access to specific data
  - Contain no members
  - Inactive by default
  - Access other databases through permissions granted in those databases to the guest account
- Not associated with server-level principals

User-defined roles

- Database-level securable
- Can be assigned an AUTHORIZATION
  - Database user or role that owns the role
  - Default is the user that creates the role
- Requires CREATE ROLE or the DB_SECURITYADMIN database-level role
  - To assign to another user requires IMPERSONATE permission on that user
  - To assign to another role requires ALTER permission on that role
  - To assign to an application role requires ALTER permission on the application role

Direct grants - Allow

- Grants permissions on a table, view, table-valued function, stored procedure, extended stored procedure, scalar function, aggregate function, service queue, or synonym
  - Tables/views (INSERT, UPDATE, DELETE, SELECT, REFERENCES, etc.)
  - Stored procedures (EXECUTE)
- WITH GRANT OPTION
  - Principal may grant the privilege to other principals

Direct grants - Deny

- Denies permissions on a member of the OBJECT class of securables. These are the members of the OBJECT class: tables, views, table-valued functions, stored procedures, extended stored procedures, scalar functions, aggregate functions, service queues, and synonyms.
- DENY takes precedence over grant
  - Exception - table-level DENY does not take precedence over column-level GRANT

Security standards

Security standard

- Something considered by an authority or by general consent as a basis of comparison; an approved model.
- Base security stance
- Most organizations have technical standards

Free security benchmarks

- CIS SQL Server Benchmark
  http://benchmarks.cisecurity.org/downloads/browse/index.cfm?category=benchmarks.servers.database.mssql
- DISA STIG
  http://iase.disa.mil/stigs/app-security/database/Pages/sql.aspx
- Microsoft
  http://blogs.technet.com/b/secguide/archive/2014/03/24/sql-server-2012-baselines-are-now-live.aspx

Vendor tools

- CIS
- Tenable Nessus
- Trustwave AppDetective Pro
- Microsoft SQLRAP

Authentication

Modes

- Windows authentication (local accounts and Active Directory managed accounts)
- SQL Server authentication

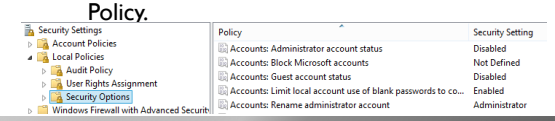
Windows authentication

- Account name and password validated using the Windows principal token (OS layer)
- Identity confirmed by the OS
  - Default mode
  - Uses Kerberos
    - Password policy enforcement
    - Account lockout
    - Password expiration
- Windows groups can be used at the domain level
  - Logins added to the group
  - Simplified administration

Preferred method and leading practice

Determine password policy on computer

- On the Start menu, click Run.
- In the Run dialog box, type secpol.msc, and then click OK.
- In the Local Security Settings application, expand Security Settings, expand
- Account Policies, and then click Password Policy.



SQL Server authentication

- Logins **NOT** based on Windows logins
- Username and password are created and stored in SQL Server
  - Strong passwords must be set for all accounts

'sa' must always have a password

name	is_security_policy_checked	is_expiration_checked	is_disabled
sa	TRUE	FALSE	TRUE
##MS_PolicyEventProcessingLogin##	TRUE	FALSE	TRUE
##MS_PolicyTsqlExecutionLogin##	TRUE	FALSE	TRUE

SQL Server password policies

- User must change password at next login

Requires the user to change the password the next time that the user connects. The ability to change the password is provided by SQL Server Management Studio.

- Enforce password expiration

The maximum password age policy of the computer is enforced for SQL Server logins.

- Enforce password complexity

The Windows password policies of the computer are enforced for SQL Server logins. This includes password length and complexity.

Advantages

- Allows SQL Server to support older applications and applications provided by third parties that require SQL Server Authentication.
- Allows SQL Server to support environments with mixed operating systems, where all users are not authenticated by a Windows domain.
- Allows users to connect from unknown or untrusted domains.
- Allows SQL Server to support Web-based applications where users create their own identities.
- Allows software developers to distribute their applications by using a complex permission hierarchy based on known, preset SQL Server logins.

Disadvantages

- If a user is a Windows domain user who has a login and password for Windows, he must still provide another (SQL Server) login and password to connect. Keeping track of multiple names and passwords is difficult for many users.
- SQL Server Authentication cannot use Kerberos security protocol.
- Windows offers additional password policies that are not available for SQL Server logins.
- The encrypted SQL Server Authentication login password, must be passed over the network at the time of the connection. Some applications that connect automatically will store the password at the client. These are additional attack points.

Using SQL Server Authentication does not limit the permissions of local administrators on the computer where SQL Server is installed.

Performing the audit

Audit types - management

- Software installation
- Roles
- Administration
- Security configurations
- Configuration management
- Application audit

Software installation

Dedicated server

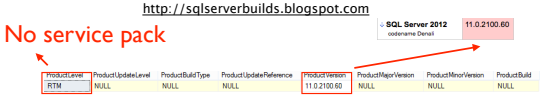
- SQL Server should be in a dedicated server
  - Never on a domain controller
  - Never with a web server
  - No security software
  - No email server
  - etc.

Version and service pack information

- Ensure SQL Server is a current supported version.

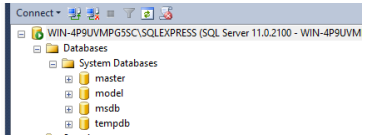
http://sqlserverbuilds.blogspot.com

No service pack



List databases

- The databases can be seen from SQL Server Management Studio
  - Expand Databases->System Databases



List databases

- The databases can be obtained from the database data dictionary

	name	database_id	create_date
1	master	1	2003-04-08 09:13:36.390
2	tempdb	2	2015-11-12 05:06:39.440
3	model	3	2003-04-08 09:13:36.390
4	msdb	4	2012-02-10 21:02:17.770

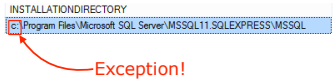
As with management studio, the Resource DB is hidden.

Location of software

- Should not be on the system partition

```
PS C:\> $env:systemdrive
C:
```

- Location is in the registry
  HKLM\SOFTWARE\Microsoft SQL Server\MSQL11.<instance name>\Setup

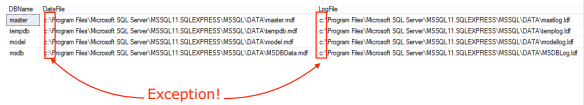


Location of data files

- Data directories must not be on the system partition

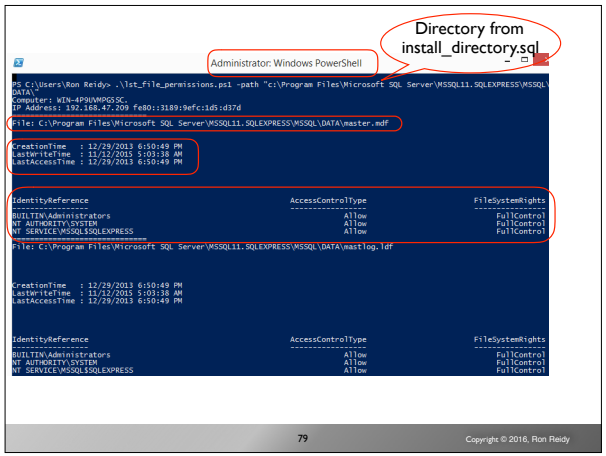
```
PS C:\Users\Ron Reidy> $env:systemdrive
C:
```

- Data files



OS Permissions to database files

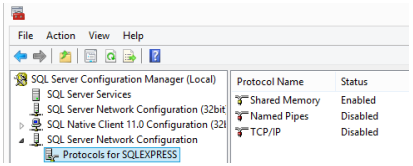
- Data files can contain sensitive application data as well as password hashes for SQL logins
- Log files contain transaction information
- Test permissions on all folders containing data or log files
- Validate access is appropriate

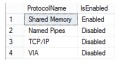


Networking

Enabled network protocols

- Identify all SQL Server networking protocols enabled

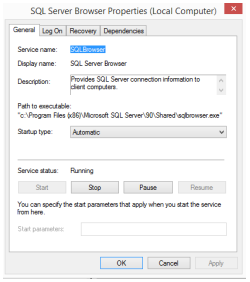


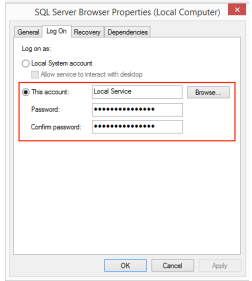


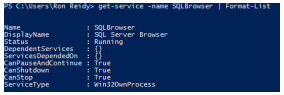
	ProtocolName	IsEnabled
1	Shared Memory	Enabled
2	Named Pipes	Disabled
3	TCP/IP	Disabled
4	VIA	Disabled

Browser service

- Windows service
- Listens for incoming connection requests
  - Provides information about SQL Server instances on the server (instance name and version)







Roles

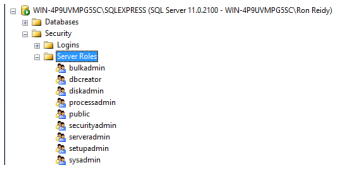
Server level (recap)

- Manage permissions of the databases
  - Security principals that group other principals
- Database wide in scope and exist in each database
- Any database account and any server role can be added into database-level roles
- Any member of a database-level role can add other logins to that same role
- Three different classes of roles (we will discuss later)
  - Administration roles
  - Data access roles
  - Roles restricted to the msdb database

Server-level role name	Description
sysadmin	Members of the sysadmin fixed server role can perform any activity in the server.
serveradmin	Members of the serveradmin fixed server role can change server-wide configuration options and shut down the server.
securityadmin	Members of the securityadmin fixed server role manage logins and their properties. They can GRANT, DENY, and REVOKE server-level permissions. They can also GRANT, DENY, and REVOKE database-level permissions if they have access to a database. Additionally, they can reset passwords for SQL Server logins.
processadmin	Members of the processadmin fixed server role can end processes that are running in an instance of SQL Server.
setupadmin	Members of the setupadmin fixed server role can add and remove linked servers.
bulkadmin	Members of the bulkadmin fixed server role can run the BULK INSERT statement.
diskadmin	The diskadmin fixed server role is used for managing disk files.
dbcreator	Members of the dbcreator fixed server role can create, alter, drop, and restore any database.
public **NOTE: Implemented differently than other roles.**	Every SQL Server login belongs to the public server role. When a server principal has not been granted or denied specific permissions on a securable object, the user inherits the permissions granted to public on that object.

Members of the SYSADMIN and SECURITYADMIN roles should be treated as equally powerful.

List all server-level roles



Identify accounts with server-level roles

SERVERNAME	LOGINNAME	SYSADMIN	SECURITYADMIN	SETUPADMIN	PROCESSADMIN	DISKADMIN	DBCREATOR	BULKADMIN
Server1\PROD_DB	[redacted]\Prod_DBA	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Server1\PROD_DB	[redacted]\Auditors	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
Server1\PROD_DB	[redacted]\Dev_DBA	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
Server1\PROD_DB	[redacted]\Windows_Admins	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
Server1\PROD_DB	[redacted]\Windows_Admins_NonPriv	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

Database level (recap)

- Manage permissions of the databases
  - Security principals that group other principals
- Database wide in scope and exist in each database
- Any database account and any server role can be added into database-level roles
- Any member of a database-level role can add other logins to that same role
- Three different types of roles
  - Administration roles
  - Data access roles
  - Roles restricted to the msdb database

Database roles should **never** be members of fixed roles. This could enable unintended privilege escalation.

List all database-level roles

```
1  exec sp_helpdbfixedrole;
```

100 %

Results | Messages

	DbFixedRole	Description
1	db_owner	DB Owners
2	db_accessadmin	DB Access Administrators
3	db_securityadmin	DB Security Administrators
4	db_ddladmin	DB DDL Administrators
5	db_backupoperator	DB Backup Operator
6	db_datareader	DB Data Reader
7	db_datawriter	DB Data Writer
8	db_denydatareader	DB Deny Data Reader
9	db_denydatawriter	DB Deny Data Writer

Identify accounts with database-level roles

SERVERNAME	DBNAME	USERNAME	DB_OWNER	DB_ACCESSADMIN	DB_SECURITYADMIN	DB_DDLADMIN	DB_DATAREADER	DB_DATAWRITER	DB_DENYDATAREADER	DB_DENYDATAWRIT
Server1\PROD_DB	Fin	dbo	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Server1\PROD_DB	Purch	[redacted]\Prod_DBA	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
Server1\PROD_DB	HR	[redacted]\Dev_DBA	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
Server1\PROD_DB	Stage	[redacted]\Prod_DBA	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE
Server1\PROD_DB	WH1	Ron	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
Server1\PROD_DB	WH2	[redacted]\Windows_Admins	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE
Server1\PROD_DB	WH1	[redacted]\Windows_Admin_NonPriv	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE

Overrides DB_DATAREADER and DB_DATAWRITER

Administration

Windows OS administrative accounts

- Windows-authenticated accounts that have administrative access to the SQL Server
  - Windows built-in accounts
  - Local Windows groups
  - Active Directory groups
- Password trust obtained from the operating system

Identify all OS groups and accounts

- Verify they are authorized administrators

```
PS C:\Users\Ron Reidy> Get-WmiObject win32_groupuser |
>> Where-Object { $_.GroupComponent -match 'administrators' } |
>> ForEach-Object {[wmi]$_.PartComponent } | export-csv admins.csv
>>
PS C:\Users\Ron Reidy>
```

	A	B	C	D	E	F
1	#TYPE System.Management.ManagementObject#root\cimv2\Win32_UserAccount					
2	PSComputerName	__GENUS	__CLASS	__SUPERCLASS	__DYNAST	__RELPATH
3	WIN-4P9UVMPG5SC	2	Win32_Us	Win32_Account	CIM_Mana	Win32_UserAccount.Domain="WIN-4P9UVMPG5SC",Name="Administrator"
4	WIN-4P9UVMPG5SC	2	Win32_Us	Win32_Account	CIM_Mana	Win32_UserAccount.Domain="WIN-4P9UVMPG5SC",Name="Ron Reidy"

Service accounts

- Valid service accounts
  - Local user
  - Domain user
  - NetworkService
  - Local System

```
WIN-4P9UVMPG5SC  name                 state    startname
---------------  ----                 -----    ---------
                 MSSQL$SQLEXPRESS     Running  NT Service\MSSQL$SQLEXPRESS
                 SQLAgent$SQLEXPRESS  Stopped  NT AUTHORITY\NETWORKSERVICE
                 SQLBrowser           Running  NT AUTHORITY\LOCALSERVICE
                 SQLWriter            Running  LocalSystem
```

Administrative logins

- Ensure all administrative SQL logins have password settings enabled

name	Access_Method	is_expiration_checked	is_security_policy_checked	is_disabled
sa	sysadmin membership	FALSE	TRUE	TRUE
sa	serveradmin membership	FALSE	TRUE	TRUE
sa	securityadmin membership	FALSE	TRUE	TRUE

Built-in users

- Local windows group (BUILTIN\Users)
  - Run applications
  - Use printers
  - Shutdown and lock computer
- All accounts on server are members of this group

LoginName	type_desc	is_disabled	db_perms	srv_perms
BUILTIN\Users	WINDOWS_GROUP	FALSE	no_db_users	no_srv_permissions

Built-in administrators

- Full control over the windows server
- Access should be limited (after installation, only "Administrator" is present)

server_name	SrvRole	LoginName

Leading practice - the Administrators group should not have access to SQL Server.

Security configurations

Server configuration

- Stored in the data dictionary
- Can be modified

name	value_configured	value_in_use	description
Ad Hoc Distributed Queries	FALSE	FALSE	Enable or disable Ad Hoc Distributed Queries
c2 audit mode	FALSE	FALSE	c2 audit mode
clr enabled	FALSE	FALSE	CLR user code execution enabled in the server
contained database authentication	FALSE	FALSE	Enables contained databases and contained authentication
cross db ownership chaining	FALSE	FALSE	Allow cross db ownership chaining
Database Mail XPs	FALSE	FALSE	Enable or disable Database Mail XPs
default trace enabled	TRUE	TRUE	Enable or disable the default trace
filestream access level	FALSE	FALSE	Sets the FILESTREAM access level
Ole Automation Procedures	FALSE	FALSE	Enable or disable Ole Automation Procedures
remote access	TRUE	TRUE	Allow remote access
remote admin connections	FALSE	FALSE	Dedicated Admin Connections are allowed from remote clients
remote login timeout (s)	TRUE	TRUE	remote login timeout
scan for startup procs	FALSE	FALSE	scan for startup stored procedures
show advanced options	FALSE	TRUE	show advanced options
SMO and DMO XPs	TRUE	TRUE	Enable or disable SMO and DMO XPs
user instance timeout	TRUE	TRUE	The timeout of the user instance after no connection is made on the server
xp_cmdshell	FALSE	FALSE	Enable or disable command shell

CLR enabled

- If enabled, identify all CLR assemblies stored in the DB instance and validate them

db	owner	name	clr_name	permission_set_desc	is_visible	create_date	modify_date	is_user_defined
master	securityadmin	microsoft.sqlserver.types, version=11.0.0.0, cul...	Microsoft.SqlServer.Types	UNSAFE_ACCESS	TRUE	2012-02-10 20:15:58.843	2012-02-10 20:15:59.427	FALSE
tempdb	securityadmin	microsoft.sqlserver.types, version=11.0.0.0, cul...	Microsoft.SqlServer.Types	UNSAFE_ACCESS	TRUE	2012-02-10 20:15:58.843	2012-02-10 20:15:59.427	FALSE
model	securityadmin	microsoft.sqlserver.types, version=11.0.0.0, cul...	Microsoft.SqlServer.Types	UNSAFE_ACCESS	TRUE	2012-02-10 20:15:58.843	2012-02-10 20:15:59.427	FALSE
msdb	securityadmin	microsoft.sqlserver.types, version=11.0.0.0, cul...	Microsoft.SqlServer.Types	UNSAFE_ACCESS	TRUE	2012-02-10 20:15:58.843	2012-02-10 20:15:59.427	FALSE

Scan for startup procs

- Stored procedures that execute when SQL Server starts

name	minimum	maximum	config_value	run_value
scan for startup procs	0	1	0	0

- If enabled, validate it should be enabled
- Validate any startup proc found are authorized to run

name

A note about the trustworthy bit

- The Trustworthy bit allows database objects to access objects in other (remote) databases
- Setting this to 'off' provides protection from malicious CLR assemblies or extended procedures
- The exception to this is the 'sa' account

A note about the trustworthy bit

If the database owner for a database is assigned to the SYSADMIN server role and the database has its TRUSTWORTHY bit set to ON then a privileged database user can elevate privileges to the SYSADMIN server role and compromise the system. MSDB database is allowed to have the database owner assigned to the SYSADMIN server role and TRUSTWORTHY bit set to ON.

- The exception to this is the 'sa' account

DATABASE_NAME	OWNER_LOGIN	is_trustworthy_on	EXCEPTION
master	sa	FALSE	FALSE
tempdb	sa	FALSE	FALSE
model	sa	FALSE	FALSE
msdb	sa	TRUE	FALSE

Stored procedures

- SQL Server has many built-in stored procedures.
- CIS benchmark specifies only 'xp_cmdshell'
- Many others included which are not controlled by configuration system
  - Access given to PUBLIC

Stored procedures

Stored procedure	Description
xp_dirtree	Used to get a list of all the folders for the folder named in the xp
xp_fileexist	Determine whether a particular file exists on the disk or not
xp_fixeddrives	Returns the list of all hard drives and the amount of free space in Mb for each hard drive.
xp_getnetname	Returns the WINS name of the SQL Server that you're connected to
xp_instance_regread	Reads the registry
xp_msver	Returns version information about Microsoft SQL Server. **xp_msver** also returns information about the actual build number of the server and information about the server environment.
xp_qv	Returns information about the SKU type, licensing etc.
xp_regread	Reads the registry
xp_replposteor	Replication
xp_sprintf	Formats and stores a series of characters and values in the string output parameter.
xp_sscanf	Reads data from the string into the argument locations specified by each format argument.

Configuration management

Database objects

- Objects in the database can be changed by owners of the objects
  - Logins and accounts that can impersonate the owner
  - Server-level roles (sysadmin)
  - Built-in "sa" account
  - The "dbo" of the database (more later)
  - Accounts with database-level DB_OWNER (if not explicitly denied)

Configuration management

- Identify changed stored procedures, views, and CLR assemblies

db	owner	name	type_desc	create_date	modify_date	is_ms_shipped	is_published	is_schema_published	is_visible	is_user_defined
model	sa	EventNotificationErrorsQueue	SERVICE_QUEUE	2009-04-13 12:59:08.983	2009-04-13 12:59:08.983	TRUE	FALSE	FALSE	NULL	NULL
model	sa	MSreplication_options	USER_TABLE	2012-02-10 21:14:19.113	2012-02-10 21:14:52.690	TRUE	FALSE	FALSE	NULL	NULL
model	sa	QueryNotificationErrorsQueue	SERVICE_QUEUE	2009-04-13 12:59:08.967	2009-04-13 12:59:08.967	TRUE	FALSE	FALSE	NULL	NULL
model	sa	ServiceBrokerQueue	SERVICE_QUEUE	2009-04-13 12:59:08.983	2009-04-13 12:59:08.983	TRUE	FALSE	FALSE	NULL	NULL
model	sa	sp_MSdelunusedmergepublisher	SQL_STORED_PROCEDURE	2012-02-10 21:14:20.183	2012-02-10 21:14:52.710	TRUE	FALSE	FALSE	NULL	NULL
model	sa	sp_MSreplstartup	SQL_STORED_PROCEDURE	2012-02-10 21:14:19.750	2012-02-10 21:14:52.700	TRUE	FALSE	FALSE	NULL	NULL
model	sa	spt_fallback_db	USER_TABLE	2003-04-08 09:10:01.557	2012-02-10 21:14:52.657	TRUE	FALSE	FALSE	NULL	NULL
model	sa	spt_fallback_dev	USER_TABLE	2003-04-08 09:10:02.870	2012-02-10 21:14:52.667	TRUE	FALSE	FALSE	NULL	NULL
model	sa	spt_fallback_usg	USER_TABLE	2003-04-08 09:18:04.180	2012-02-10 21:14:52.673	TRUE	FALSE	FALSE	NULL	NULL
model	sa	spt_monitor	USER_TABLE	2012-02-10 21:02:08.440	2012-02-10 21:14:52.683	TRUE	FALSE	FALSE	NULL	NULL
model	sa	spt_values	VIEW	2012-02-10 21:02:07.743	2012-02-10 21:02:07.953	TRUE	FALSE	FALSE	NULL	NULL
model	sa	sysxmitbuffer	SYSTEM_TABLE	2012-03-10 20:19:59.977	2012-03-10 20:19:59.993	TRUE	FALSE	FALSE	NULL	NULL

- Validate changes were approved

Application audit

- Roles
- Account management
- Data access

Roles

- Server level - already covered
- Database level - already covered
- Application

Application roles

- Database principal allows application to run with user-like permissions
  - Access other databases through permissions granted in those databases to the 'guest' account (if guest is disabled - no access)
- Use to enable access to data and objects
- Contain no members and inactive (by default)
- Enabled by calling 'sp_setapprole' (requires a password)
- Cannot access server-level metadata (not associates with server-level principals
  - Can be set using the 'dbcc traceon' command (global flag 4616)

Listing all trace events

- dbcc - database console command
- 4 categories of commands

Command category	Perform
Maintenance	Maintenance tasks on a database, index, or filegroup.
Miscellaneous	Miscellaneous tasks such as enabling trace flags or removing a DLL from memory.
Informational	Tasks that gather and display various types of information.
Validation	Validation operations on a database, table, index, catalog, filegroup, or allocation of database pages.

tracestatus

```
dbcc tracestatus
```

Results | Messages

TraceFlag	Status	Global	Session
8017	1	1	0

List application roles

```
select *
from sys.fn_builtin_permissions(N'APPLICATION ROLE');
```

Results | Messages

class_desc	permission_name	type	covering_permission_name
APPLICATION ROLE	VIEW DEFINITION	VW	CONTROL
APPLICATION ROLE	ALTER	AL	CONTROL
APPLICATION ROLE	CONTROL	CL	

Account management

- Account analysis
  - Identify accounts and their roles
    - Server and database level

Identify all accounts with server or database level roles

- SQL logins
- Windows accounts
  - Windows and Active Directory groups

Server level roles

SERVERNAME	LOGINNAME	SYSADMIN	SECURITYADMIN	SETUPADMIN	PROCESSADMIN	DISKADMIN	DBCREATOR	BULKADMIN
Server1\PROD_DB	[redacted]\Prod_DBA	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Server1\PROD_DB	[redacted]\Auditors	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
Server1\PROD_DB	[redacted]\Dev_DBA	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
Server1\PROD_DB	[redacted]\Windows_Admins	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
Server1\PROD_DB	[redacted]\Windows_Admins_NonPriv	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

Database-level roles

SERVERNAME	DBNAME	USERNAME	DB_OWNER	DB_ACCESSADMIN	DB_SECURITYADMIN	DB_DDLADMIN	DB_DATAREADER	DB_DATAWRITER	DB_DENYDATAREADER	DB_DENYDATAWRIT
Server1\PROD_DB	Fin	dbo	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Server1\PROD_DB	Purch	[redacted]\Prod_DBA	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
Server1\PROD_DB	HR	[redacted]\Dev_DBA	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
Server1\PROD_DB	Stage	[redacted]\Prod_DBA	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE
Server1\PROD_DB	WH1	Ron	FALSE	FALSE	FALSE	FALSE	**FALSE**	**FALSE**	TRUE	TRUE
Server1\PROD_DB	WH2	[redacted]\Windows_Admins	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE
Server1\PROD_DB	WH1	[redacted]\Windows_Admin_NonPriv	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE

Overrides DB_DATAREADER and DB_DATAWRITER

dbo

DB_OWNER database-level role vs. "dbo" account

- "dbo" is a special "pseudo" user in every database
  - **NOT** the same as DB_OWNER database-level role
  - Any account can be assigned the DB_OWNER database-level role
    - Has **complete** control of the database instance and all databases in it
  - All members of the server-level role SYSADMIN are mapped to "dbo"
    - SYSADMINs have **all** rights in **all** databases
  - "dbo" bypasses **all** permissions checks within the database
- Members of the DB_OWNER database-level role but **not** the "dbo" can be DENYied permissions to securables

dbo

db	principal_name
master	dbo
tempdb	dbo
model	dbo
msdb	dbo
AdventureWorks2012	dbo

LoginName	LoginType	SysAdmin
sa	SQL_LOGIN	TRUE
WIN-4P9UVMPG5SC\Ron Reidy	WINDOWS_LOGIN	TRUE
NT SERVICE\SQLWriter	WINDOWS_LOGIN	TRUE
NT SERVICE\Winmgmt	WINDOWS_LOGIN	TRUE
NT Service\MSSQL$SQLEXPRESS	WINDOWS_LOGIN	TRUE

db	principal_name	db_account
master	dbo	sa
tempdb	dbo	sa
model	dbo	sa
msdb	dbo	sa
AdventureWorks2012	dbo	sa

All are "dbo"

Data access

Schemas

Schema

- Logical grouping of objects within a database
  - Tables
  - Views
  - Stored procedures
- Access granted to schema applies to all objects

List schemas

Permissions on the schemas

List permissions on schemas

Object names

- Objects names are fully qualified by specifying the "containers" they reside in
- Fully qualified names consist of
  - Server name
  - Database name
  - Schema name
  - Object name
- Specified as *server.database.schema.object*
  - Remote server/database access

How object permissions are checked

- Object access tested in reverse order of fully qualified name
  - Access granted or denied directly to the object
  - Access granted or denied on the schema containing the object
  - Access granted or denied on the database containing the schema
  - Access granted or denied on the server containing the database
- During these tests, DENY permissions are checked first and access is denied if it exists at ANY of these levels
- If no specific permissions exist, access is denied

Implicit object access

- There are several ways an object can be accessed implicitly
  - Access given to server-level roles
    - public
    - SYSADMIN
  - Access given to the database-level roles
    - db_datareader - read (SELECT) access to all user tables
      - db_denydatareader denies SELECT
    - db_datawriter - INSERT, UPDATE, DELETE access to all user tables
      - db_denydatawriter denies INSERT, UPDATE, DELETE
    - db_owner - access to all user objects
    - dbo - owns the entire database
  - SQL logins with administrative access (sa and custom logins)

Access

- Access granted
  - Directly to the principal
  - Through a role (RBAC)

Permissions

Object permission	Description
ALTER	Change the properties, except ownership, of a particular securable
CONTROL	Confers ownership–like capabilities on the grantee. The grantee effectively has all defined permissions on the securable.
DELETE	Delete existing rows from a table
EXECUTE	Execute a stored procedure or CLR assembly
INSERT	Insert new rows into a table
REFERENCES	Create a FOREIGN KEY constraint that references that table.
SELECT	Select data from a table
UPDATE	Update existing rows in a table

PUBLIC role

- Server-level role (and database-level role)
  - All principals are members of the PUBLIC role
  - Access to PUBLIC cannot be revoked or denied
  - The PUBLIC role cannot be dropped

Access given to PUBLIC role

- Many built-in objects granted to PUBLIC
- Many user-defined objects can have access granted to PUBLIC
  - Stored procedures
  - CLR assemblies
  - Tables, views, etc.

db	name	type_desc	grantor	grantee
master	all_columns	VIEW	sa	public
master	all_objects	VIEW	sa	public
master	all_parameters	VIEW	sa	public
master	all_sql_modules	VIEW	sa	public
master	all_views	VIEW	sa	public
master	allocation_units	VIEW	sa	public
master	assemblies	VIEW	sa	public
master	assembly_files	VIEW	sa	public
master	assembly_modules	VIEW	sa	public
master	assembly_references	VIEW	sa	public
master	assembly_types	VIEW	sa	public
master	asymmetric_keys	VIEW	sa	public

Object access should be granted to PUBLIC when the need is fully demonstrated

All endpoints will be granted to PUBLIC

class_desc	permission_name	state_desc	grantor	name	protocol_desc	ip_address	type_desc	state_desc	is_admin_endpoint	is_dynamic_port
SERVER	VIEW ANY DATABASE	GRANT	sa	NULL	NULL	NULL	NULL	NULL	FALSE	FALSE
ENDPOINT	CONNECT	GRANT	sa	TSQL Local Machine	SHARED_MEMORY	NULL	NULL	NULL	FALSE	FALSE
ENDPOINT	CONNECT	GRANT	sa	TSQL Named Pipes	NAMED_PIPES	NULL	NULL	NULL	FALSE	FALSE
ENDPOINT	CONNECT	GRANT	sa	TSQL Default TCP	TCP	NULL	TSQL	STARTED	FALSE	TRUE
ENDPOINT	CONNECT	GRANT	sa	TSQL Default VIA	VIA	NULL	NULL	NULL	FALSE	FALSE

Tables

- Identify access to all tables with sensitive data
  - SSN
  - PII
  - Passwords
- Identify hashing or encryption (especially password fields)
  - If encrypted, look at key access/ management
  - Hashing should be "one-way

Database audit logging

Audit logging basics

- Track and log events on the database engine
  - Server events
  - Database events
- Uses extended events
- No audit is enabled by default

Audit trail

- When audit is created, logging destination is defined
- When created, it is *disabled* and must be enabled to log events
- Locations
  - Event log
    - Windows Security event log
    - Windows application event log
  - File on the OS file system
    - Restrict access to the file and its location

Audit trail leading practices

- If using the event log as an audit destination
  - Avoid using the Windows Application event log
    - Any authenticated user can read and write to this event log (less secure)
- If using an OS file as the audit destination
  - Define an audit on master.sys.fn_get_audit_file
- Always audit actions of "dbo" for all databases

Topics I did not cover

- Schema security audit
- Replication and backups
- Transparent Data Encryption (TDE)

Q&A

References

- Microsoft
  - SQL Server books online - https://msdn.microsoft.com/en-us/library/ms130214(v=sql.105).aspx
  - Securing SQL Server - https://technet.microsoft.com/en-us/library/bb283235(v=sql.110).aspx
  - Checklist - https://technet.microsoft.com/en-us/library/ff848786(v=sql.105).aspx
- Securing SQL Server 2nd edition - Denny Cherry - ISBN 978-1-59749-947-7
- Center for Internet Security benchmarks - http://benchmarks.cisecurity.org/downloads/browse/?category=benchmarks.servers.database.mssql
- DISA STIGs - http://iase.disa.mil/stigs/Pages/index.aspx
  - STIG viewer - http://iase.disa.mil/stigs/Pages/stig-viewing-guidance.aspx
- Database weekly - http://databaseweekly.com
- ISACA - http://www.isaca.org/Journal/archives/2015/Volume-1/Pages/Auditing-SQL-Server-Databases-Using-CAATs.aspx

Thank you