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 and/or risk team
- The testing methodology is my own
- The scripts used in the presentation are working in a test environment (I have used them in my previous positions to execute audit testing)

Learning objectives

- SQL Server architecture
- Data access model for SQL Server
- Understanding of …
  - Defaults
  - 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 database 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 (container of databases)
  - Disk files
  - Memory
  - Network connections
  - User processes

Database

- Logical container
- Security boundary
Principal
- Entities that can request and use resources
- 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 authorizes and 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
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 supports OLAP by allowing 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 allows you to design, create and manage multidimensional structures that contain data aggregated from other data sources such as relational databases.

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 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 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.

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 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.

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

![Single instance diagram]

#### Clustered instance

![Clustered instance diagram]
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

Installed instances found in the registry
HKLM\SOFTWARE\Microsoft\Microsoft SQL Server\Instance Names\SQL2019\SQLExpress

Disk files

- All data in a SQL Server instance is written to data files on the file system (local or SAN/NAS)
- Locations of data files specified in the data dictionary
- Check locations
- Check permissions
- Validate access

Data file access
Data file 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 in SQL Server
  - TCP/IP
  - Named Pipes
  - Shared Memory
  - VIA

Common ports

<table>
<thead>
<tr>
<th>Description</th>
<th>Protocol</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Engine (default instance)</td>
<td>TCP</td>
<td>1433</td>
</tr>
<tr>
<td>Database Engine (default instance)</td>
<td>UDP</td>
<td>1434</td>
</tr>
<tr>
<td>Database Mail</td>
<td>SMTP</td>
<td>25</td>
</tr>
<tr>
<td>Database Mirroring</td>
<td>TCP</td>
<td>No official default port, but examples tend to use 5022.</td>
</tr>
<tr>
<td>Dedicated Administrative Connection (default instance)</td>
<td>TCP</td>
<td>1434</td>
</tr>
<tr>
<td>Filestream</td>
<td>TCP</td>
<td>139 and 445</td>
</tr>
<tr>
<td>Service Broker</td>
<td>TCP</td>
<td>No official default port, but examples tend to use 4022.</td>
</tr>
<tr>
<td>SQL Server Browser Service</td>
<td>UDP/TCP</td>
<td>UDP: 1434</td>
</tr>
<tr>
<td>SQL Server Integration Services</td>
<td>TCP</td>
<td>135</td>
</tr>
<tr>
<td>TSQl Debugger</td>
<td>TCP</td>
<td>135</td>
</tr>
</tbody>
</table>
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 a SQL Server instance is created
- sa
  - Server-level principal
- INFORMATION_SCHEMA & sys
  - Appear as users in catalog views
- Required by SQL Server
- Not principals
- Cannot be modified, renamed, 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

<table>
<thead>
<tr>
<th>System database</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>master Database</td>
<td>Records all the system-level information for an instance of SQL Server.</td>
</tr>
<tr>
<td>msdb Database</td>
<td>Used by SQL Server Agent for scheduling alerts and jobs.</td>
</tr>
<tr>
<td>model Database</td>
<td>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.</td>
</tr>
<tr>
<td>Resource Database</td>
<td>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.</td>
</tr>
<tr>
<td>tempdb Database</td>
<td>Workspace for holding temporary objects or intermediate result sets.</td>
</tr>
</tbody>
</table>
Default file locations

- Shared files for all instances
- `<drive>:\Program Files\Microsoft SQL Server\MSSQL1.4\SQLEXPRESS\MSSQL\DATA`, 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

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

Overview
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

Permissions of server-level roles

<table>
<thead>
<tr>
<th>Server-level role name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysadmin</td>
<td>Members of the sysadmin fixed server role can perform any activity in the server.</td>
</tr>
<tr>
<td>serveradmin</td>
<td>Members of the serveradmin fixed server role can change server-wide configuration options and shut down the server.</td>
</tr>
<tr>
<td>securityadmin</td>
<td>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.</td>
</tr>
<tr>
<td>processadmin</td>
<td>Members of the processadmin fixed server role can end processes that are running in an instance of SQL Server.</td>
</tr>
<tr>
<td>setupadmin</td>
<td>Members of the setupadmin fixed server role can add and remove linked servers.</td>
</tr>
<tr>
<td>bulkadmin</td>
<td>Members of the bulkadmin fixed server role can run the BULK INSERT statement.</td>
</tr>
<tr>
<td>diskadmin</td>
<td>The diskadmin fixed server role is used for managing disk files.</td>
</tr>
<tr>
<td>dbcreator</td>
<td>Members of the dbcreator fixed server role can create, alter, drop, and restore any database.</td>
</tr>
<tr>
<td>public</td>
<td>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.</td>
</tr>
</tbody>
</table>

Members of the SYSAADMIN and SECURITYADMIN roles should be treated as equally powerful.
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 (server-level) roles. This could enable unintended privilege escalation.

<table>
<thead>
<tr>
<th>Database-level role name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>db_owner</td>
<td>Members of the db_owner fixed database role can perform all configuration and maintenance activities on the database, and can also drop the database.</td>
</tr>
<tr>
<td>db_securityadmin</td>
<td>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.</td>
</tr>
<tr>
<td>db_accessadmin</td>
<td>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.</td>
</tr>
<tr>
<td>db_backupoperator</td>
<td>Members of the db_backupoperator fixed database role can back up the database.</td>
</tr>
<tr>
<td>db_ddladmin</td>
<td>Members of the db_ddladmin fixed database role can run any Data Definition Language (DDL) command in a database.</td>
</tr>
</tbody>
</table>

Data access roles

<table>
<thead>
<tr>
<th>Database-level role name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>db_datawriter</td>
<td>Members of the db_datawriter fixed database role can add, delete, or change data in all user tables.</td>
</tr>
<tr>
<td>db_datareader</td>
<td>Members of the db_datareader fixed database role can read all data from all user tables.</td>
</tr>
<tr>
<td>db_denydatawriter</td>
<td>Members of the db_denydatawriter fixed database role cannot add, modify, or delete any data in the user tables within a database.</td>
</tr>
<tr>
<td>db_denydatareader</td>
<td>Members of the db_denydatareader fixed database role cannot read any data in the user tables within a database.</td>
</tr>
</tbody>
</table>

msdb specific roles

<table>
<thead>
<tr>
<th>msdb role name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>db_ssisadmin</td>
<td>Members of these database roles can administer and use SSIS.</td>
</tr>
<tr>
<td>db_ssisoperator</td>
<td>Members of the db_datareader fixed database role can read all data from all user tables.</td>
</tr>
<tr>
<td>db_ssisltuser</td>
<td>Members of the db_datareader fixed database role can read all data from all user tables.</td>
</tr>
<tr>
<td>dc_admin</td>
<td>Members of the db_datareader fixed database role can read all data from all user tables.</td>
</tr>
<tr>
<td>dc_operator</td>
<td>Members of the db_datareader fixed database role can read all data from all user tables.</td>
</tr>
<tr>
<td>dc_proxy</td>
<td>Members of the db_datareader fixed database role can read all data from all user tables.</td>
</tr>
<tr>
<td>PolicyAdministratorRole</td>
<td>Members of the db_PolicyAdministratorRole database role can perform all configuration and maintenance activities on Policy-Based Management policies and conditions.</td>
</tr>
<tr>
<td>ServerGroupAdministratorRole</td>
<td>Members of these database roles can administer and use registered server groups.</td>
</tr>
<tr>
<td>ServerGroupReaderRole</td>
<td>Members of these database roles can administer and use registered server groups.</td>
</tr>
<tr>
<td>dbm_monitor</td>
<td>Created in the msdb database when the first database is registered in Database Mirroring Monitor.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Permission Name</th>
<th>Permission State</th>
<th>Principal Name</th>
<th>Principal Type</th>
<th>Principal Is Defined</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER ANY Availability Group</td>
<td>GRANT</td>
<td>NT AUTHORITY\SYSTEM</td>
<td>WINDOWS\LOGIN</td>
<td>FALSE</td>
</tr>
<tr>
<td>CONNECT SQL</td>
<td>GRANT</td>
<td>NT AUTHORITY\SYSTEM</td>
<td>WINDOWS\LOGIN</td>
<td>FALSE</td>
</tr>
<tr>
<td>CONNECT SQL</td>
<td>GRANT</td>
<td>NT SERVICE\IISWMSGS\SQLExpress</td>
<td>WINDOWS\LOGIN</td>
<td>FALSE</td>
</tr>
<tr>
<td>CONNECT SQL</td>
<td>GRANT</td>
<td>NT SERVICE\SQLWitro</td>
<td>WINDOWS\LOGIN</td>
<td>FALSE</td>
</tr>
<tr>
<td>CONNECT SQL</td>
<td>GRANT</td>
<td>NT SERVICE\SrvrWro</td>
<td>WINDOWS\LOGIN</td>
<td>FALSE</td>
</tr>
<tr>
<td>CONNECT SQL</td>
<td>GRANT</td>
<td>sa</td>
<td>SQL\LOGIN</td>
<td>TRUE</td>
</tr>
<tr>
<td>CONNECT SQL</td>
<td>GRANT</td>
<td>WIN\ASPMIGO\Ron Reidy</td>
<td>WINDOWS\LOGIN</td>
<td>FALSE</td>
</tr>
<tr>
<td>VIEW SERVER STATE</td>
<td>GRANT</td>
<td>NT AUTHORITY\SYSTEM</td>
<td>WINDOWS\LOGIN</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

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

- 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
- DISA STIG
- Microsoft

Vendor tools

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

Authentication

- Windows authentication (local accounts and Active Directory managed accounts) **DEFAULT**
- Mixed mode supports authentication both by Windows and by SQL Server. User name and password pairs are maintained within SQL Server
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

SQL Server password policies

- User must change password at next login
  Requires the user to change the password the next time 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 organizations.
• 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. This is an additional attack vector.

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

Audit types - management

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

Dedicated server
- SQL Server should be on a dedicated server
- Never on a domain controller
- Never on a web server
- Never with security software
- Never on an email server
- etc.

Version and service pack information
- Ensure SQL Server is a current supported version.
- Patches must be current

List databases
- The databases can be seen from SQL Server Management Studio
- Expand Databases->System Databases
Location of software

- Should never be installed on the system partition
- 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
- Data files
- Exception!

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

Enabled protocols should be specified in the system security plan. Detective controls should be created and enabled to ensure no other protocols are enabled.

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 in the instance
- Security principals that group other principals
- Instance 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

<table>
<thead>
<tr>
<th>Server-level role name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysadmin</td>
<td>Members of the sysadmin fixed server role can perform any activity in the server.</td>
</tr>
<tr>
<td>serveradmin</td>
<td>Members of the serveradmin fixed server role can change server-wide configuration options and shut down the server.</td>
</tr>
<tr>
<td>securityadmin</td>
<td>Members of the securityadmin fixed server role manage logins and their properties. 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.</td>
</tr>
<tr>
<td>processadmin</td>
<td>Members of the processadmin fixed server role can end processes that are running in an instance of SQL Server.</td>
</tr>
<tr>
<td>setupadmin</td>
<td>Members of the setupadmin fixed server role can add and remove linked servers.</td>
</tr>
<tr>
<td>bulkadmin</td>
<td>Members of the bulkadmin fixed server role can run the BULK INSERT statement.</td>
</tr>
<tr>
<td>diskadmin</td>
<td>The diskadmin fixed server role is used for managing disk files.</td>
</tr>
<tr>
<td>dbcreator</td>
<td>Members of the dbcreator fixed server role can create, alter, drop, and restore any database.</td>
</tr>
<tr>
<td>public</td>
<td>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.</td>
</tr>
</tbody>
</table>

Members of the SYSADMIN and SECURITYADMIN roles should be treated as equally powerful.
Identify accounts with server-level roles

Ensure all accounts are identified in the system security plan. Validate these accounts are periodically certified to have this level of access (refer to your local policies).

---

Database level (recap)

- Manage permissions of the databases
- Security principals that group other principals
- 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.

---

Identify accounts with database-level roles

Overrides DB_DATAREADER and DB_DATAWRITER

Validate these accounts are periodically certified to have this level of access (refer to your local policies).
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

Ensure all accounts are identified in the system security plan. Validate these accounts are periodically certified to have this level of access (refer to your local policies). Identify ownership of service accounts.

Service accounts
- Valid service accounts
- Local user
- Domain user
- NetworkService
- Local System
Administrative logins

- Ensure all administrative SQL logins have password settings enabled.

<table>
<thead>
<tr>
<th>login</th>
<th>Access Method</th>
<th>is_expiration_checked</th>
<th>is_security_policy_checked</th>
<th>is_disabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysadmin membership</td>
<td>FALSE</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>sa</td>
<td>FALSE</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>sa</td>
<td>FALSE</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>securityadmin membership</td>
<td>FALSE</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
</tbody>
</table>

Ensure all accounts are identified in the system security plan. Validate these accounts are periodically certified to have this level of access (refer to your local policies). Identify ownership of service accounts.

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

Built-in administrators

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

Security configurations

Ensure all accounts are identified in the system security plan. Validate these accounts are periodically certified to have this level of access (refer to your local policies). Identify ownership of service accounts.
Server configuration

- Stored in the data dictionary
- Can be modified

<table>
<thead>
<tr>
<th>name</th>
<th>value_configured</th>
<th>value_in_use</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Hfc Distributed Queues</td>
<td>FALSE</td>
<td>FALSE</td>
<td>Enable or disable All Hfc Distributed Queues</td>
</tr>
<tr>
<td>C2 audit mode</td>
<td>FALSE</td>
<td>FALSE</td>
<td>C2 audit mode</td>
</tr>
<tr>
<td>CLR enabled</td>
<td>FALSE</td>
<td>FALSE</td>
<td>CLR user code execution enabled in the server</td>
</tr>
<tr>
<td>contained database auth</td>
<td>FALSE</td>
<td>FALSE</td>
<td>Enable contained databases and contained authentication</td>
</tr>
<tr>
<td>cross db ownership chaining</td>
<td>FALSE</td>
<td>FALSE</td>
<td>Allow cross db ownership chaining</td>
</tr>
<tr>
<td>Database Mail XP</td>
<td>FALSE</td>
<td>FALSE</td>
<td>Enable or disable Database Mail XP</td>
</tr>
<tr>
<td>default trace enabled</td>
<td>TRUE</td>
<td>TRUE</td>
<td>Enable or disable the default trace</td>
</tr>
<tr>
<td>flashlight access level</td>
<td>FALSE</td>
<td>FALSE</td>
<td>Set the FLIGHTSCAM access level</td>
</tr>
<tr>
<td>DWE Automation Procedures</td>
<td>FALSE</td>
<td>FALSE</td>
<td>Enable or disable DWE Automation Procedures</td>
</tr>
<tr>
<td>remote access</td>
<td>TRUE</td>
<td>TRUE</td>
<td>Allow remote access</td>
</tr>
<tr>
<td>remote admin connections</td>
<td>TRUE</td>
<td>TRUE</td>
<td>Dedicated Admin connections are allowed from remote clients</td>
</tr>
<tr>
<td>remote login timeout (in)</td>
<td>TRUE</td>
<td>TRUE</td>
<td>remote login timeout</td>
</tr>
<tr>
<td>scan for startup procs</td>
<td>FALSE</td>
<td>FALSE</td>
<td>scan for default startup procedures</td>
</tr>
<tr>
<td>show advanced options</td>
<td>FALSE</td>
<td>TRUE</td>
<td>show advanced options</td>
</tr>
<tr>
<td>SMO and DRO XP</td>
<td>TRUE</td>
<td>TRUE</td>
<td>Enable or disable SMO and DRO XP</td>
</tr>
<tr>
<td>user instance timeout (in)</td>
<td>TRUE</td>
<td>TRUE</td>
<td>The timeout of the user instance after no connection is made on the server</td>
</tr>
<tr>
<td>&quot;sqlcmd&quot;</td>
<td>FALSE</td>
<td>FALSE</td>
<td>Enable or disable &quot;sqlcmd&quot;</td>
</tr>
</tbody>
</table>

CLR enabled

- If enabled, identify all CLR assemblies stored in the DB instance and validate they are authorized and/or needed for operations

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

Scan for startup procs

- Stored procedures that execute when SQL Server starts

<table>
<thead>
<tr>
<th>name</th>
<th>minimum</th>
<th>maximum</th>
<th>config_value</th>
<th>run_value</th>
</tr>
</thead>
<tbody>
<tr>
<td>scan for startup procs</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- If enabled, validate it should be enabled
- Validate any startup proc found are authorized or needed to run in the environment
A note about the trustworthy bit

If the database owner for a database is assigned 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. The 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

<table>
<thead>
<tr>
<th>DATABASE_NAME</th>
<th>OWNER_LOGIN</th>
<th>is_trustworthy_on</th>
<th>EXCEPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>master</td>
<td>sa</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>tempdb</td>
<td>sa</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>model</td>
<td>sa</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>msdb</td>
<td>sa</td>
<td>TRUE</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Stored procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xp_dirtree</td>
<td>Used to get a list of all the folders for the folder named in the xp</td>
</tr>
<tr>
<td>xp_fileexist</td>
<td>Determine whether a particular file exists on the disk or not</td>
</tr>
<tr>
<td>xp_fixeddrives</td>
<td>Returns the list of all hard drives and the amount of free space in Mb for each hard drive.</td>
</tr>
<tr>
<td>xp_getnetname</td>
<td>Returns the WINS name of the SQL Server that you’re connected to</td>
</tr>
<tr>
<td>xp_instance_regread</td>
<td>Reads the registry</td>
</tr>
<tr>
<td>xp_msver</td>
<td>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.</td>
</tr>
<tr>
<td>xp_regread</td>
<td>Reads the registry</td>
</tr>
<tr>
<td>xp_replposteor</td>
<td>Replication</td>
</tr>
<tr>
<td>xp_sprintf</td>
<td>Formats and stores a series of characters and values in the string output parameter.</td>
</tr>
<tr>
<td>xp_sscanf</td>
<td>Reads data from the string into the argument locations specified by each format argument.</td>
</tr>
</tbody>
</table>

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

- 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 associated 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

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

traceestatus

```
dbcc_traceestatus
```

```
<table>
<thead>
<tr>
<th>Results</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>TraceFlag</td>
<td>Status</td>
</tr>
<tr>
<td>8017</td>
<td>1</td>
</tr>
</tbody>
</table>
```

List application roles

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

```
<table>
<thead>
<tr>
<th>class_desc</th>
<th>permission_name</th>
<th>type</th>
<th>covering_permission_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION ROLE</td>
<td>VIEW DEFINITION</td>
<td>VV</td>
<td>CONTROL</td>
</tr>
<tr>
<td>APPLICATION ROLE</td>
<td>ALTER</td>
<td>AL</td>
<td>CONTROL</td>
</tr>
<tr>
<td>APPLICATION ROLE</td>
<td>CONTROL</td>
<td>CL</td>
<td></td>
</tr>
</tbody>
</table>
```
Account management

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

Server level roles

Database-level roles

dbo

Overrides DB_DATAREADER and DB_DATAWRITER
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 SQL Server 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 can be DENYed permissions to securable (tables, views, etc.)
• The “dbo” cannot be denied access

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

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

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

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

All endpoints will be granted to PUBLIC

Do not change the permissions on endpoints.

Tables

- Identify access to all tables with sensitive data
  - PHI
  - 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 event on master.sys.fn_get_audit_file
• Always audit actions of the “dbo” in all databases

Topics I did not cover

• Schema security audit
• Replication and backups
• Transparent Data Encryption (TDE)
Q&A

References

- Microsoft
- Database weekly - http://databaseweekly.com

Thank you