SQL Server and Security

Arun Jebakumar

SQL Server DBA

PASS Professional Association for SQL Server

www.chnsqlug.co.cc

www.sql-articles.com

What's in this Session

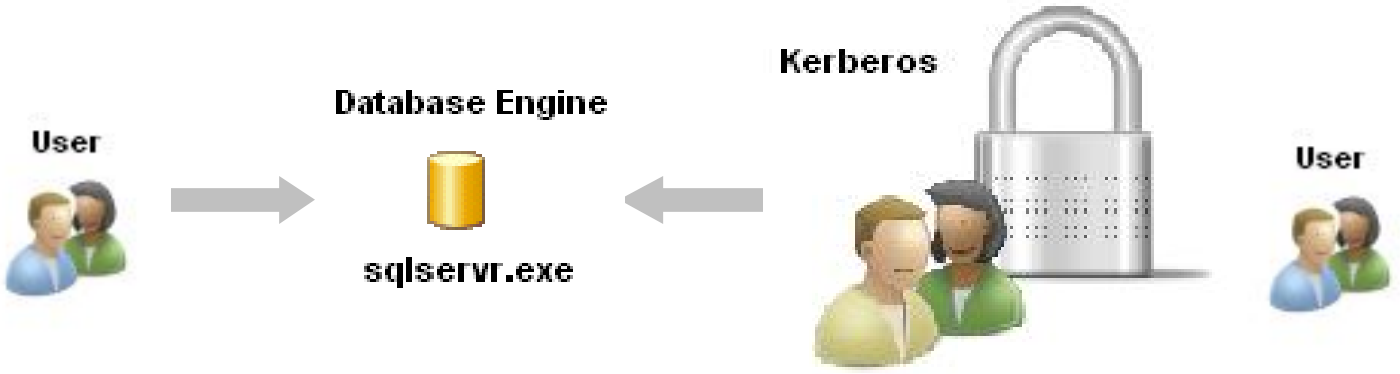
- Instance level security
  - Authentication
  - Windows logins
  - Service account
  - Securables
  - Endpoints

What's in this Session

- Database level security
  - Security model
  - Schemas
  - Database roles
  - Application roles
  - Execution context

Authentication

- Default – Integrated with Windows security
- Mixed mode – SQL logins



Authentication (continued)

- Kerberos security protocol
  - Strong password
  - Account lockout
  - Password expiration
- SQL logins
  - Enforce password policy option
  - Passwords stored in system databases

Authentication -considerations

- Always Windows authentication
- SQL logins
  - Unknown and untrusted domains
  - Non Windows clients
  - Only for backward compatibility

Windows Logins

- SQL 2008 – Secure by design
- BUILTIN\Administrators
- NT Authority\System
  - Service packs, hotfixes
  - SQL Server VSS writer service
- SQL 2005 service account groups
  - MSSQLUser, SQLAgentUser groups

Service account

- NT Authority\System
- NT Authority\Network Service
- NT Authority\Local Service
- Domain User Account
- Always use dedicated service account with minimal privileges

Service account –minimal permissions

- Log on as a service
- Replace process level token
- Bypass traverse checking
- Adjust memory quotas for a process
- Start permissions SQL AD helper, Writer
- Read event log and RPC service
- Read/write access to SQL folders

Securables

- Logins, Databases, Endpoints
- CREATE LOGIN.. WITH options
- FROM Windows, Certificates, Keys
- Default database
- Login SID, ID and Windows SID

Endpoints

- TDS packets
- TDS endpoints for all protocols
- Protocols installed with Windows
- SQL Network configuration utility
- Default protocols
  - TCP/IP, Named pipes, VIA, Shared memory

Endpoints (continued)

- SQL Server Native Client
- Network interface Protocol layer
  - Server side implemented in sqlservr.exe
  - Client side through Native Client or older SQL server protocols
  - Encapsulation as TDS packets

Endpoints (continued)

- Dedicated Administrator Connection
- Mirroring endpoints
- Endpoint states
  - Started, Stopped, Disabled

Database security model

- Securables
  - All database objects (Tables, Views, Procedures, Functions, Triggers, etc.)
- Principals
  - Users and Roles
- One login mapped to one user

Schemas

- SQL 2000 Security model
  - User and schema names are same
  - Users cannot be dropped if they own objects
  - Code needs to be changed after changing owner
- SQL 2005 and later
  - Users do not own objects
  - Objects are contained in schemas

Schemas (continued)

- Every object is part of a schema
- User can own one or more schemas in a database
- Every user will have a default schema
- Default users and schemas
  - Dbo, sys, guest, information_schema

Roles

- Fixed-server roles
  - Sysadmin, serveradmin, securityadmin
  - Processadmin, setupadmin, bulkadmin
  - Diskadmin, dbcreator
- Database roles
- Application roles

Database Roles

- Fixed database roles
  - Db_accessadmin, db_securityadmin
  - Db_datareader, db_datawriter
  - Db_denydatareader, db_denydatawriter
  - Db_backupoperator
  - Db_ddladmin
  - Db_owner
- Flexible database roles

Application Roles

- Role includes password
- Password specified by application
- Different roles can be used by parts of the application without user intervention
- Activated using sp_setapprole
- Reverted using sp_unsetapprole
- @fCreateCookie and @cookie parameters

Metadata permissions

- View server state
- View database state
- View definition
- View any definition
- View any database

Execution context

- Modules (Procedures, Functions, Triggers), Views
- Ownership chaining
  - Owner of module should be same as referenced
- Broken ownership chains
  - Caller should have permissions on the object

Execution context (continued)

- SQL 2008 execution context
  - Mark part of modules to use different users
  - Cannot drop a user if used for execution context
- Execute as Caller – no impersonation
- Execute as ‘User1’ – no ownership chain
- Execute as Self
- Execute as Owner
- REVERT statement

Covered in this Session

- SQL Database Engine security
  - Instance level security
  - Database level security

Other Security features

- Surface area configuration
- SQL Agent security
- File level security
- Encryption
- Auditing
- BI security

QUESTIONS??

Contact me @ **arunjeba@hotmail.com**
Questions @ **chnsqlug@googlegroups.com**

THANK YOU!!