May 17, 2011

SQL SERVER SECURITY

GRANTING, CONTROLLING, AND AUDITING DATABASE ACCESS

Mike Fal - www.mikefal.net

Denver SQLUG
SQL SERVER USER GROUP

Mike Fal

Working with SQL Server since MSSQL 7.

Currently supporting 100+ servers with varying requirements.

Blog – www.mikefal.net

Twitter - @Mike_Fal

SpeakerRate - http://speakerrate.com/speakers/15287-mike-fal

The importance of security

Primary goal – Protecting the data!

Security – Tools that control access to the data.

Risk – Can someone gain unauthorized access?
How likely is it?

Scope

- How do we manage access?
  - – Grant/Revoke/Deny
  - – Authentication types
  - – Server roles
  - – Database roles
- How can we audit login access?
  - – Views
  - – Queries

Getting Access

How do we control database logins?

Logins and Users

Access is managed on two levels

- Logins – Access to the server
- Users – Access to a database

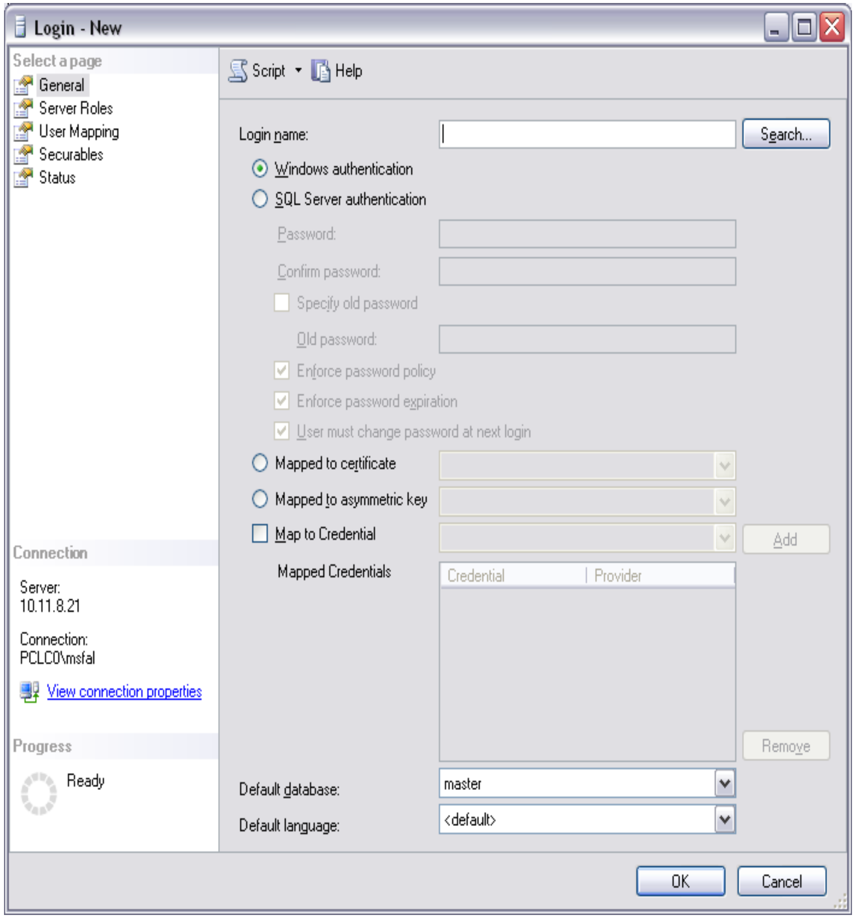
Authentication Types

Windows pass-through

- Uses Active Directory accounts
- Passwords controlled by domain policy

Direct Database Login

- Accounts used only by SQL Server.
- Passwords controlled by local computer policy
- Can override policy and expiration enforcement



Login - New

Select a page
- General
- Server Roles
- User Mapping
- Securables
- Status

Script ▾ Help

Login name: | Search...

● Windows authentication

○ SQL Server authentication

Password:

Confirm password:

□ Specify old password

Old password:

☑ Enforce password policy

☑ Enforce password expiration

☑ User must change password at next login

○ Mapped to certificate

○ Mapped to asymmetric key

□ Map to Credential | Add

Mapped Credentials

Credential	Provider

Remove

Default database: master

Default language: <default>

Connection

Server:
10.11.8.21

Connection:
PCLCO\msfal

View connection properties

Progress

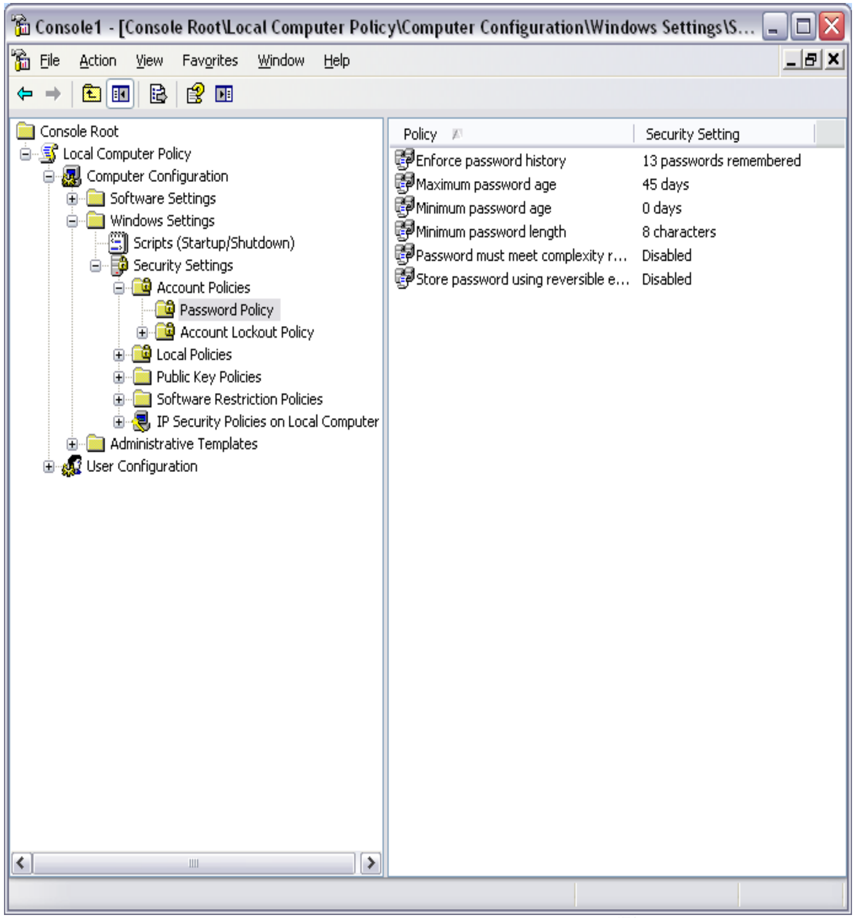
Ready

OK | Cancel

Editing Password Policies

Local Policy Editor

- Administrative tools ->
- Local Security Policy



Policy	Security Setting
Enforce password history	13 passwords remembered
Maximum password age	45 days
Minimum password age	0 days
Minimum password length	8 characters
Password must meet complexity r...	Disabled
Store password using reversible e...	Disabled

Creating a Login

Use the GUI: Security->Users->Right Click, New Login…

T-SQL:

- CREATE LOGIN <login name> FROM WINDOWS
- CREATE LOGIN <login name> WITH PASSWORD '<password>'

Creating a user

Use the GUI: Security->Users->Right Click, New User…

T-SQL:

- CREATE USER <user name> FROM LOGIN <login name>

Query Logins

Use sys.server_principals and sys.sql_logins views

```
select
  sp.name,
  sp.type_desc,
  sp.default_database_name,
  sl.is_policy_checked,
  sl.is_expiration_checked
from sys.server_principals sp
  left join sys.sql_logins sl on
   (sp.principal_id = sl.principal_id)
where sp.type not in ('R','C')
order by name
```

Controlling Access

How do you stop the monkey business?

Best Practices

Understand your business needs.

Keep access as restrictive as possible.

Access Levels

Server Level

- Start/stop services
- Grant access
- Create databases
- Perform bulk operations

Database Level

- Query and modify data
- Create objects

Explicit Permissions

Assumption is no access unless granted

GRANT – give user privileges on an object

Does not override implicit denied permissions

Examples:

```
grant select on customers to test
grant insert on orders to test
grant delete,update on customers to test
```

Explicit Permissions

DENY – remove user privileges on an object

Overrides any implicit permission grants

Examples:

```
deny select on customers to test
deny insert on orders to test
deny delete,update on customers to test
```

Explicit Permissions

REVOKE– resets user privileges on an object

In other words, removes explicit grant or deny

Examples:

```
revoke select on customers to test
revoke insert on orders to test
revoke delete,update on customers to
test
```

Permission Types

Many different permissions to use:

- SELECT, INSERT, UPDATE, DELETE – Tables
- EXECUTE, VIEW DEFINITION – Stored Procedures
- ALTER, DROP – Objects (tables, databases, etc)

Roles provide a way to better manage permissions.

Server Roles

SYSADMIN – Perform any action on the server.

SECURITYADMIN – Manage server level permissions.

SERVERADMIN – Manage server configurations and start/stop services.

PROCESSADMIN – Kill processes running on the instance.

SETUPADMIN – Add/remove linked servers.

BULKADMIN – Able to run BULK INSERT and execute bulk operations.

DISKADMIN – Manage server disk files.

DBCREATOR – Create, alter, drop, and restore databases.

PUBLIC – Generic role that all users are a member of.

http://msdn.microsoft.com/en-us/library/ms188659.aspx

Server Roles

Access can be granted via individual GRANTs or roles.

SYSADMIN and SECURITYADMIN are the critical server roles.

SQL Denali allows you to make custom server roles.

Add logins to roles either by GUI or sp_addsrvrolemember

```
select
   r.name [Server Role],
   u.name [Login],
   u.type_desc [User Type]
from (select name,principal_id
        from sys.server_principals where type = 'R') r
   join sys.server_role_members rm
        on (r.principal_id = rm.role_principal_id)
   join (select name,type_desc,principal_id
        from sys.server_principals where type != 'R') u
        on (rm.member_principal_id = u.principal_id)
```

Database Roles

DB_OWNER - Perform all activities on the database.

DB_SECURITYADMIN – Manages role membership and permissions on the database.

DB_ACCESSADMIN – Manages login access to the database.

DB_BACKUPOPERATOR – Can backup the database.

DB_DDLADMIN – Able to run any DDL command.

DB_DATAWRITER – Able to modify data in all user tables.

DB_DATAREADER – Able to read data in all user tables.

DB_DENYDATAWRITER – Denied the ability to modify data in all user tables.

DB_DENYDATAREADER – Denied the ability to modify data in all user tables.

Database Roles

Access can be granted via individual GRANTs or roles.
Custom roles can be created within a database.
Add users to roles using GUI or sp_addrolemember.

```
select
   r.name role_name,
   u.name db_login,
   u.type_desc
from (select name,principal_id
        from sys.database_principals where type = 'R') r
   join sys.database_role_members rm
        on (r.principal_id = rm.role_principal_id)
   join (select name,type_desc,principal_id
        from sys.database_principals where type != 'R') u
        on (rm.member_principal_id = u.principal_id )
```

Auditing

Monitoring user access

General Practices

Create some basic reports – Excel or Reporting Services.

Watch out for escalating permissions (DBO and SA versus other roles).

Nested permissions:

- AD groups and changing members
- xp_logininfo

Auditing Role Access

Server and Database Role queries.

- sys.server_principals and sys.server_role_members for Server Roles
- sys.database_principals and sys. database_role_members for Database Roles

Auditing Specific Access

sys.database_permissions to show individual object grants

```
select
   pr.name,
   pe.type,
   o.name,
   o.type_desc,
   pe.permission_name,
   state_desc
from
   sys.database_principals pr
   join sys.database_permissions pe on (pr.principal_id =
   pe.grantee_principal_id)
   join sys.objects o on (pe.major_id = o.object_id)
where
   pe.state in ('W','G')
   and o.type = 'U'
order by pr.name
```

Summary

- Types of authentication – Windows pass through and Direct Database Login.
- Roles – Tools to manage access
- Auditing – Perform regular reviews of your security

Questions

HUH?

www.mikefal.net
@Mike_Fal
http://speakerrate.com/speakers/15287-mike-fal