

SQL Server Security

Granting, Controlling, and Auditing Database Access

September 10, 2011

Thank you to our sponsors

Gold



SQL Solutions GROUP
Knowledgeable, Reliable, and Innovative Database Engineers

idera™ **redgate®**
TOOLS FOR SQL SERVER,
SHAREPOINT, AND POWERSHELL

Blog



Shannon Lowder
Database Engineer

Prize

Apress

SSWUG.org™
your database answers are here

SQL SERVER
magazine

Bronze



CozyRoc
► Go to the next level

SQLRNNR
DATABASE SERVICES

Mike Fal

Working with SQL Server since MSSQL 7.

Experience with different industries.

No cool letters after my name. ;-;

Blog – www.mikefal.net

Twitter – @Mike_Fal

The importance of security

Primary goal – Protecting the data!

Security – Tools and processes that control access to the data.

Risk – How we evaluate the strength of our security.

Scope

- ▶ How do we manage access?
 - 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

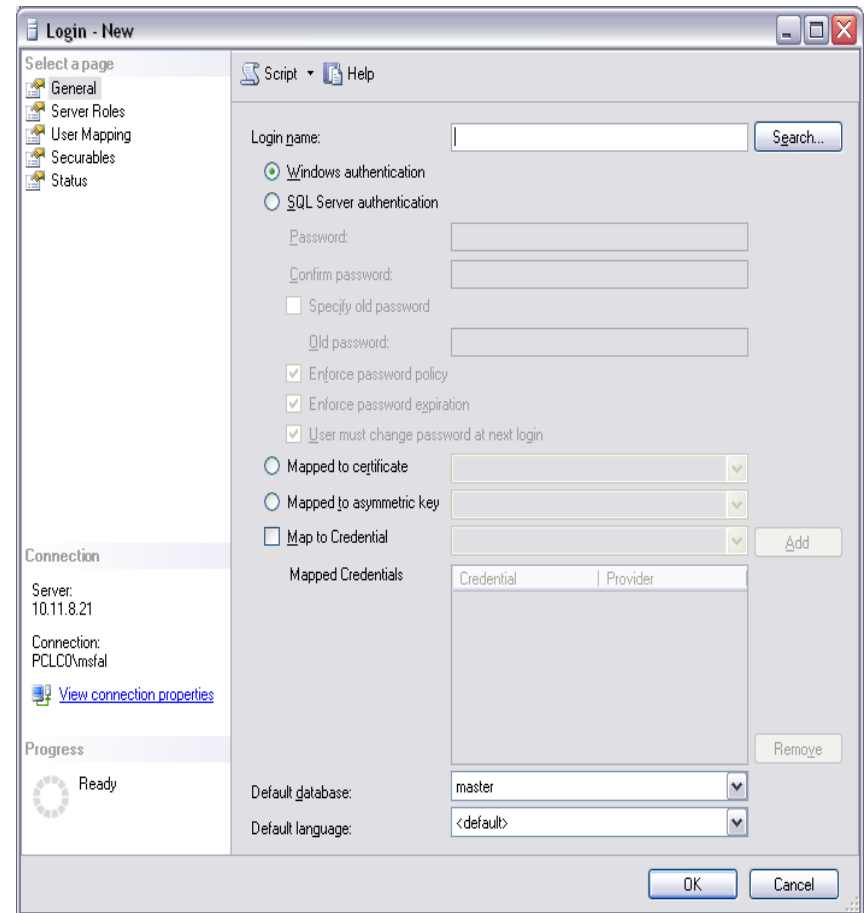
Authentication Types

Windows pass-through

- Uses Active Directory accounts
- Passwords controlled by domain policy

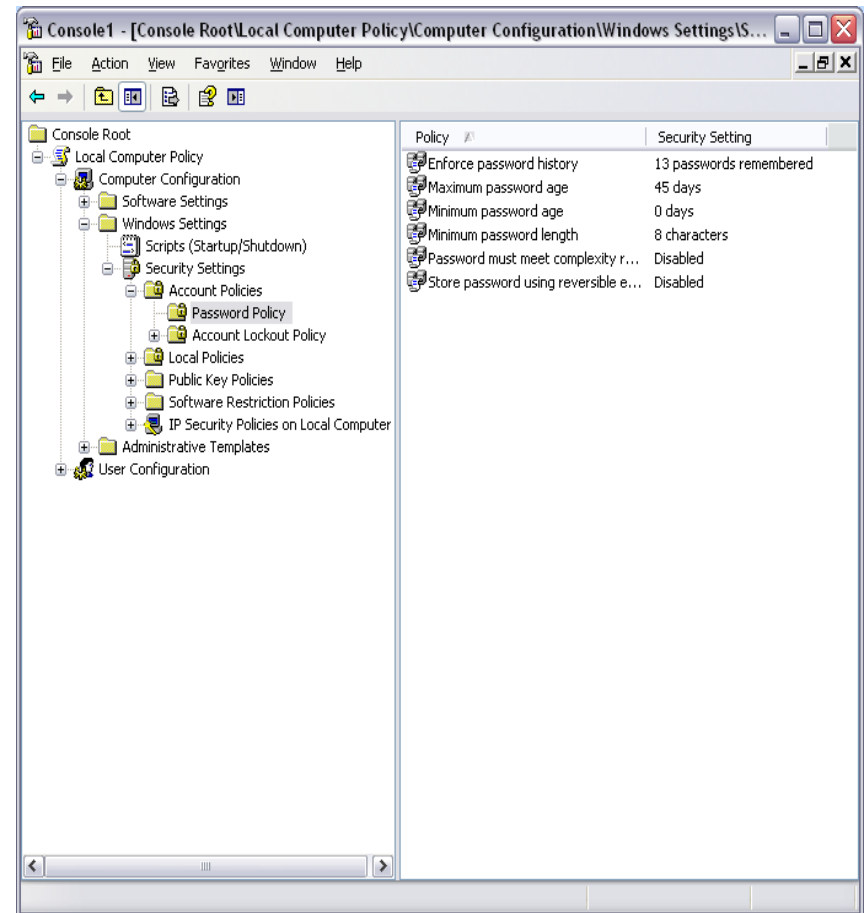
Direct Database Login

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



Editing Password Policies

Local Policy Editor
Administrative tools –
>
Local Security Policy



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?

Managing Access

Understand your business needs.

Keep access as restrictive as possible.

Access Levels

Server Level (logins)

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

Database Level (users)

- Query and modify data
- Create objects

Explicit permissions

Assumption is no access unless granted

GRANT –give user privileges on an object

Does not override explicit 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 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 11 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.

<http://msdn.microsoft.com/en-us/library/ms189121.aspx>

Database Roles

Create your own roles to define specific access.

```
create role report_writer  
grant select on customers to report_writer  
exec sp_addrolemember 'report_writer', 'test'
```

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/redundant 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
- xp_logininfo allows us to AD query group membership

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

www.mikefal.net