Übertragen von Benutzern mit ihren Kennwörtern zwischen SQL Server Instanzen

Kathegorie
SQL Server

Aaron
Priesterroth

Wenn man die SQL Server Instanz wechselt, beispielsweise im Rahmen einer Migration, ist es von großem Vorteil alle Benutzer und Benutzer-Gruppen der Quelle zu übernehmen um allen Benutzern der alten Instanz den Zugriff auf die neue Instanz wie gewohnt zu ermöglichen.

Besonders einfach funktioniert das Übertragen, wenn man sich die zwei von Microsoft bereitgestellten gespeicherten Prozeduren zur Hilfe nimmt.

Prozedur 1: sp_hexadecimal

Die erste Prozedur, **sp_hexadecimal**, ist eine Unterstützung für die eigentliche Prozedur **sp_help_revlogin** und sieht wie folgt aus:

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

```
USE master
GO
IF OBJECT_ID ('sp_hexadecimal') IS NOT NULL
  DROP PROCEDURE sp_hexadecimal
GO
CREATE PROCEDURE sp_hexadecimal
    @binvalue varbinary(256),
    @hexvalue varchar (514) OUTPUT
AS
DECLARE @charvalue varchar (514)
DECLARE @i int
DECLARE @length int
DECLARE @hexstring char(16)
SELECT @charvalue = '0x'
SELECT @i = 1
SELECT @length = DATALENGTH (@binvalue)
SELECT @hexstring = '0123456789ABCDEF'
WHILE (@i <= @length)
BEGIN
  DECLARE @tempint int
  DECLARE @firstint int
  DECLARE @secondint int
  SELECT @tempint = CONVERT(int, SUBSTRING(@binvalue,@i,1))
  SELECT @firstint = FLOOR(@tempint/16)
  SELECT @secondint = @tempint - (@firstint*16)
  SELECT @charvalue = @charvalue +
    SUBSTRING(@hexstring, @firstint+1, 1) +
    SUBSTRING(@hexstring, @secondint+1, 1)
  SELECT @i = @i + 1
END

SELECT @hexvalue = @charvalue
GO
```

Prozedur 2: sp_help_revlogin

Die zweite Prozedur, **sp_help_revlogin**, ist die Prozedur die zum Übertragen der Benutzer ausgeführt wird. Dabei werden die Benutzer mit samt ihrer Passwörter in verschlüsselter Form aus der Instanz extrahiert und ein ausführbares Statement zum erstellen des ausgelesenen Benutzers erstellt. Die Prozedur hat folgende Gestalt:

```
Use master
GO
IF OBJECT_ID ('sp_help_revlogin') IS NOT NULL
  DROP PROCEDURE sp_help_revlogin
GO
CREATE PROCEDURE sp_help_revlogin @login_name sysname = NULL AS
DECLARE @name sysname
DECLARE @type varchar (1)
DECLARE @hasaccess int
DECLARE @denylogin int
DECLARE @is_disabled int
DECLARE @PWD_varbinary  varbinary (256)
DECLARE @PWD_string  varchar (514)
DECLARE @SID_varbinary varbinary (85)
DECLARE @SID_string varchar (514)
DECLARE @tmpstr  varchar (1024)
DECLARE @is_policy_checked varchar (3)
DECLARE @is_expiration_checked varchar (3)

DECLARE @defaultdb sysname

IF (@login_name IS NULL)
  DECLARE login_curs CURSOR FOR

      SELECT p.sid, p.name, p.type, p.is_disabled, p.default_database_name,
l.hasaccess, l.denylogin FROM
sys.server_principals p LEFT JOIN sys.syslogins l
      ON ( l.name = p.name ) WHERE p.type IN ( 'S', 'G', 'U' ) AND p.name <> 'sa'
ELSE
```


Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

```
  DECLARE login_curs CURSOR FOR


      SELECT p.sid, p.name, p.type, p.is_disabled, p.default_
database_name, l.hasaccess, l.denylogin FROM
sys.server_principals p LEFT JOIN sys.syslogins l
      ON ( l.name = p.name ) WHERE p.type IN ( 'S', 'G', 'U' ) AND
p.name = @login_name
OPEN login_curs

FETCH NEXT FROM login_curs INTO @SID_varbinary, @name, @type, @
is_disabled, @defaultdb, @hasaccess, @denylogin
IF (@@fetch_status = -1)
BEGIN
  PRINT 'No login(s) found.'
  CLOSE login_curs
  DEALLOCATE login_curs
  RETURN -1
END
SET @tmpstr = '/* sp_help_revlogin script '
PRINT @tmpstr
SET @tmpstr = '** Generated ' + CONVERT (varchar, GETDATE()) + ' on
' + @@SERVERNAME + ' */'
PRINT @tmpstr
PRINT ''
WHILE (@@fetch_status <> -1)
BEGIN
  IF (@@fetch_status <> -2)
  BEGIN
    PRINT ''
    SET @tmpstr = '-- Login: ' + @name
    PRINT @tmpstr
    IF (@type IN ( 'G', 'U'))
    BEGIN -- NT authenticated account/group
      SET @tmpstr = 'CREATE LOGIN ' + QUOTENAME( @name ) + ' FROM WINDOWS WITH
DEFAULT_DATABASE = [' + @defaultdb + ']'
    END
    ELSE BEGIN -- SQL Server authentication
        -- obtain password and sid
            SET @PWD_varbinary = CAST( LOGINPROPERTY( @name, 'PasswordHash' ) AS
varbinary (256) )
        EXEC sp_hexadecimal @PWD_varbinary, @PWD_string OUT
        EXEC sp_hexadecimal @SID_varbinary,@SID_string OUT

        -- obtain password policy state
        SELECT @is_policy_checked = CASE is_policy_checked WHEN 1 THEN 'ON' WHEN
0 THEN 'OFF' ELSE NULL END FROM sys.sql_logins WHERE name = @name
        SELECT @is_expiration_checked = CASE is_expiration_checked WHEN 1 THEN
'ON' WHEN 0 THEN 'OFF' ELSE NULL END FROM sys.sql_logins WHERE name = @name

            SET @tmpstr = 'CREATE LOGIN ' + QUOTENAME( @name ) + ' WITH PASSWORD
= ' + @PWD_string + ' HASHED, SID = ' + @SID_string + ', DEFAULT_DATABASE = [' +
@defaultdb + ']'

        IF ( @is_policy_checked IS NOT NULL )
        BEGIN
          SET @tmpstr = @tmpstr + ', CHECK_POLICY = ' + @is_policy_checked
        END
        IF ( @is_expiration_checked IS NOT NULL )
        BEGIN
          SET @tmpstr = @tmpstr + ', CHECK_EXPIRATION = ' + @is_expiration_
checked
        END
    END
    IF (@denylogin = 1)
    BEGIN -- login is denied access
      SET @tmpstr = @tmpstr + '; DENY CONNECT SQL TO ' + QUOTENAME( @name )
    END
    ELSE IF (@hasaccess = 0)
```


Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

```
    BEGIN -- login exists but does not have access
      SET @tmpstr = @tmpstr + '; REVOKE CONNECT SQL TO ' +
QUOTENAME( @name )
    END
    IF (@is_disabled = 1)
    BEGIN -- login is disabled
      SET @tmpstr = @tmpstr + '; ALTER LOGIN ' + QUOTENAME( @name )
+ ' DISABLE'
    END
    PRINT @tmpstr
  END

  FETCH NEXT FROM login_curs INTO @SID_varbinary, @name, @type, @
is_disabled, @defaultdb, @hasaccess, @denylogin
   END
CLOSE login_curs
DEALLOCATE login_curs
RETURN 0
GO
```

Hinweis: Beide Prozeduren sind auch direkt auf der Website von Microsoft hier zu finden.

Arbeitsablauf

Um mit den beiden oben aufgeführten Prozeduren nun die Übertragung der Benutzer zu realisieren, kann der folgender Ablauf auf der eigenen Quell-Instanz (also die Instanz, auf der die Benutzer bereits existieren) durchgeführt werden:

- Es muss SSMS (SQL Server Management Studio) gestartet und ein neues Abfrage-Fenster (engl. **New Query**) geöffnet werden.
- Es muss der Quelltext der ersten Prozedur **sp_hexadecimal** in das Abfrage-Fenster kopiert und ausgeführt werden, um die Prozedur auf der Instanz zu erstellen.
- Bei erfolgreichem Ausführen der Anweisung sollte die Nachricht "Commands completed successfully." ausgegeben werden.
- Es muss das Abfrage-Fenster geleert werden.
- Es muss der Quelltext der zweiten Prozedur **sp_help_revlogin** in das Abfrage-Fenster kopiert und ausgeführt werden, um die Prozedur auf der Instanz zu erstellen.
- Bei erfolgreichem Ausführen der Anweisung sollte erneut die Nachricht "Commands completed successfully." ausgegeben werden. Beide Prozeduren sollten nun auf der Instanz als gespeicherte Prozeduren vorhanden sein.
- Es muss das Abfrage-Fenster geleert werden.
- Es muss nun die Prozedur **sp_help_revlogin** ausgeführt werden. Dies geschieht mit folgendem Kommando:

```
EXEC sp_help_revlogin
```

- Im Ausgabe-Fenster des Abfrage-Fensters wird nun ein Statement zum erstellen der Benutzer mit ihren Passwörter generiert. Dieses muss kopiert werden.
  Vorsicht: Dabei darf die unterste Zeile "Completion time: ..." nicht mit übernommen werden.
- Ein einzelnes Statement zur Wiederherstellung eines Benutzer kann dabei wie folgt aussehen:

```
-- Login: ExampleLogin
CREATE LOGIN [ExampleLogin] WITH PASSWORD = 0x02003756AE0E-
CAF62FB1021AAFA5DC649832D2EAB95E4458D1A063683EB63DAA10138AC963A-
76B381476801A613407719D8D5E1566846BFF23F32FA6BD0E7D58DD057150F7F5
HASHED, SID = 0x5AB30D7D3B541B408CF471C0D4A70406, DEFAULT_DATA-
BASE = [master], CHECK_POLICY = ON, CHECK_EXPIRATION = ON
```

Bei dem oben aufgeführten Beispiel handelt es sich also um den Benutzer "ExampleLogin" mit seinem zuvor spezifizierten Passwort und allen weiteren benutzerspezifischen Einstellungen.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Auf der Ziel-Instanz (also die "neue" Instanz, auf der die Benutzer noch nicht zur Verfügung stehen) können nun ganz einfach die Benutzer mit Hilfe des zuvor generierten Statements erstellt werden:

- Es muss SSMS (SQL Server Management Studio) gestartet und ein neues Abfrage-Fenster (engl. **New Query**) geöffnet werden.
- In dem Abfrage-Fenster kann nun das zuvor auf der Quell-Instanz erzeugte Statement eingefügt und ausgeführt werden.
- Bei erfolgreichem Ausführen der Anweisung sollte ein letztes Mal die Nachricht "Commands completed successfully." ausgegeben werden.

Die Benutzer sind jetzt erfolgreich auf der Ziel-Instanz erstellt worden und jeder Benutzer der alten Instanz kann sich nun auf der neuen Instanz (wie gewohnt) anmelden.

Hinweise

Unterschiedliche Domänen

Befinden sich die Quell- und Ziel-Instanz in unterschiedlichen Domänen, muss das durch die zweite Prozedur erstellte Statement so abgeändert werden, dass jede Aufführung der alten Domäne durch den Namen der neuen Domäne ersetzt wird.

Beispiel:
Gehen wir davon aus, dass die Domäne der Quell-Instanz **source** und die der neuen, Ziel-Instanz **sink**. Für jeden Benutzer muss das generierte Statement also von

```
CREATE LOGIN [source\ExampleLogin] WITH PASSWORD =
0x02003756AE0ECAF62FB1021AAFA...
```

zu

```
CREATE LOGIN [sink\ExampleLogin] WITH PASSWORD =
0x02003756AE0ECAF62FB1021AAFA...
```

umgeändert werden.

SID

Für fortgeschrittene Benutzer ist es von Vorteil zu wissen, dass mit dem Benutzer und zugehörigen Kennwort als Hash zusäztlich die zuvor dem Benutzer zugeteilte SID (Security Identification) übernommen wird. Werden auf der neuen Instanz Benutzer erstellt und einer dieser Benutzer belegt zufällig die selbe SID, kann die zu Konflikten führen.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612