

Re: Database Ownership

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.sqlserver.security/2004-01/0231.html>

From: Dan Guzman (*danguzman_at_nospam-earthlink.net*)

Date: 01/16/04

Date: Fri, 16 Jan 2004 08:25:45 -0600

> *my first question is, can I declare database ownership on more than one users, or say, can one database or its objects be owned by more than one login?*

A database may be owned by only one login.

> *Second, from a performance standpoint, there will be no 'broken link' if I am correct, and may I assume the response time will be better to users?*

An unbroken ownership chain eliminates extra security checking but I don't believe the performance difference is noticeable for most applications. A significant benefit of an unbroken ownership chain is that permissions on referenced objects are not needed. This allows you can restrict access to data through views and procedures. In a multi-database environment like yours, you could create views referencing tables in the other databases and then grant select permissions on the views.

> *Third, if all a,b,c users are all database owners, or say 'a' owns A,B,C database, and so as 'b' also owns A,B,C database, am I correct that I will lose the capability to fine tuning the permission setting on database objects (such as stored procedure exec., r/w on tables/fields) at database level on each database?*

You cannot deny permissions from the database owner. The database owner has full permissions on all objects within the database.

> *I know I should stop here but the cross database chaining concept is getting very interesting to me as a DBA/Programmer and the scenario I brought up her is all I am facing in my shop. I hope you could pardon me by allowing me to continue bring up the following of my concerns:*

>

> *If, say, I decide to integrate all three (a,b,c) applications into one, say BigBoy, this new BigBoy will have read and write functions/buttons on all A,B and C database. Now, the original users of a,b,c are now using only one application, the BigBoy. If I want to fine tuning the read and write permissions on databases without relying on the frontend applications, am I correct to remove the dbowner role from each of the login of the a,b,c user,*

and use/click the select, update,exec, etc. on the object list from the permission screen for each user?

The dbo user and the db_owner role have powerful permissions that are not normally needed for application access. A best practice is to grant needed object permissions to roles so that you can control security through user role membership. This provides more control over permissions.

> 2. *May I assume that the three users(a,b,c) I refer to above, can be replaced by or applied to Window's user defined group?*

Yes.

> 3. *My original intention is to use Role instead of group for setting the new permission scheme, but I was told the Role can not span across databases. Would you confirm on this, or if there is a workaround on Role? The reason I try to use Role because my shop has Network administration personnel and I could separate security tasks between Network Admin and DB Admin by using user defined DB Role.*

Database roles and database users are specific to a particular database but this really isn't a big deal since you can setup role permissions once and then control access through role membership. Cross-database chaining enables you to implement referencing views so that you don't need to create roles in the other databases. However the logins still need access to the other databases, either directly or via the guest user security context. The scripts below illustrates how you can set this up.

```
-- setup role security with cross-database chaining
USE A
EXEC sp_changedbowner 'MyLogin'
EXEC sp_dboption 'A', 'db chaining', true
EXEC sp_addRole 'ApplicationA'
GRANT ALL ON MyTable TO ApplicationA
GRANT ALL ON MyProc TO ApplicationA
GRANT SELECT ON MyDatabaseB_MyTable_View TO ApplicationA
GRANT SELECT ON MyDatabaseC_MyTable_View TO ApplicationA
GO
USE B
EXEC sp_changedbowner 'MyLogin'
EXEC sp_dboption 'B', 'db chaining', true
EXEC sp_addRole 'ApplicationB'
GRANT ALL ON MyTable TO ApplicationB
GRANT ALL ON MyProc TO ApplicationB
GRANT SELECT ON MyDatabaseA_MyTable_View TO ApplicationB
GRANT SELECT ON MyDatabaseC_MyTable_View TO ApplicationB
GO
USE C
EXEC sp_changedbowner 'MyLogin'
EXEC sp_dboption 'C', 'db chaining', true
EXEC sp_addRole 'ApplicationC'
```

microsoft.public.sqlserver.security: Re: Database Ownership

```
GRANT ALL ON MyTable TO ApplicationC
GRANT ALL ON MyProc TO ApplicationC
GRANT SELECT ON MyDatabaseA_MyTable_View TO ApplicationC
GRANT SELECT ON MyDatabaseB_MyTable_View TO ApplicationC
GO
```

```
-- user setup with cross-database chaining without guest user
EXEC A..sp_grantdbaccess 'MyDomain\UserA'
EXEC A..sp_grantdbaccess 'MyDomain\UserB'
EXEC A..sp_grantdbaccess 'MyDomain\UserC'
EXEC A..sp_addrolemember 'ApplicationA', 'MyDomain\UserA'
EXEC B..sp_grantdbaccess 'MyDomain\UserA'
EXEC B..sp_grantdbaccess 'MyDomain\UserB'
EXEC B..sp_grantdbaccess 'MyDomain\UserC'
EXEC B..sp_addrolemember 'ApplicationB', 'MyDomain\UserB'
EXEC C..sp_grantdbaccess 'MyDomain\UserA'
EXEC C..sp_grantdbaccess 'MyDomain\UserB'
EXEC C..sp_grantdbaccess 'MyDomain\UserC'
EXEC C..sp_addrolemember 'ApplicationC', 'MyDomain\UserC'
GO
```

```
-- user setup with cross-database chaining and guest user in each database
GO
EXEC A..sp_grantdbaccess 'MyDomain\UserA'
EXEC A..sp_addrolemember 'ApplicationA', 'MyDomain\UserA'
EXEC B..sp_grantdbaccess 'MyDomain\UserB'
EXEC B..sp_addrolemember 'ApplicationB', 'MyDomain\UserB'
EXEC C..sp_grantdbaccess 'MyDomain\UserC'
EXEC C..sp_addrolemember 'ApplicationC', 'MyDomain\UserC'
GO
```

```
--
Hope this helps.
Dan Guzman
SQL Server MVP
"Martin" <anonymous@discussions.microsoft.com> wrote in message
news:2E80551C-9C11-43E6-9457-98EA275B4A2E@microsoft.com...
> Dear Support,
>
> Upon knowing the cross database chaining option in SP3 on SQL2000 Server,
I finally understood why I had troubles on our applications last year. I
took a 'giant' step to work around the issue last time and it is about time
I should make it right now. I am hoping if you could share some thought and
have your comments on the following live scenario.
>
> 1. I have 3 databases(say A,B and C) and they are required together to
serve 3 applications or 3 login users through Access's ADPs(say a,b and c).
Each frontend application is designed and programmed to update only on its
own database, but they are allowed to pull data from the other two
databases. For example, 'a' could read/write on A database, but readonly on
B and C database.
> My SQL Server is in Windows Authentication mode. Say, if I make changes
to three users(again a,b,c) of their database access setting on database A,B
and C by declaring all of them(a,b,c) to be the owner (dbo) of all three
databases, my first question is, can I declare database ownership on more
```

microsoft.public.sqlserver.security: Re: Database Ownership

than one users, or say, can one database or its objects be owned by more than one login?

> Second, from a performance standpoint, there will be no 'broken link' if I am correct, and may I assume the response time will be better to users?

> Third, if all a,b,c users are all database owners, or say 'a' owns A,B,C database, and so as 'b' also owns A,B,C database, am I correct that I will lose the capability to fine tuning the permission setting on database objects (such as stored procedure exec., r/w on tables/fields) at database level on each database?

>

> I know I should stop here but the cross database chaining concept is getting very interesting to me as a DBA/Programmer and the scenario I brought up here is all I am facing in my shop. I hope you could pardon me by allowing me to continue bring up the following of my concerns:

>

> If, say, I decide to integrate all three (a,b,c) applications into one, say BigBoy, this new BigBoy will have read and write functions/buttons on all A,B and C database. Now, the original users of a,b,c are now using only one application, the BigBoy. If I want to fine tuning the read and write permissions on databases without relying on the frontend applications, am I correct to remove the dbowner role from each of the login of the a,b,c user, and use/click the select, update,exec, etc. on the object list from the permission screen for each user?

>

> 2. May I assume that the three users(a,b,c) I refer to above, can be replaced by or applied to Window's user defined group?

>

> 3. My original intention is to use Role instead of group for setting the new permission scheme, but I was told the Role can not span across databases. Would you confirm on this, or if there is a workaround on Role? The reason I try to use Role because my shop has Network administration personnel and I could separate security tasks between Network Admin and DB Admin by using user defined DB Role.

>

> Thank you for looking into this matter.

>

> Martin

>