

Re: Database security design with ASP.net and form-based authentication

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.sqlserver.security/2006-03/msg00037.html>

- *From:* "Dan Guzman" <guzmanda@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 8 Mar 2006 21:12:33 -0600
-

Since you already have forms-based security, why not use a single SQL login for all database access?

—

Hope this helps.

Dan Guzman
SQL Server MVP

"Diane Y" <diane.yocom@xxxxxxxxxxxxxxxxxxxx> wrote in message
news:OUiKBOwOGHA.5500@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

I'm setting up an ASP.Net intranet application with a SQL Server 2000 database. The application uses form-based authentication which is supported by the following tables: User, Role, UserRole (where each user is assigned specific roles). The system will have several different roles and users can belong to multiple roles. As an example, let's say I have the following roles: data entry, guest/view only, admin, report viewer. I'm guessing now the system will have about 20 unique users. I've figured out how to implement the role-based part on ASP.Net, but I'm stuck trying to decide the best way to secure my database tables and stored procedures.

We're on a Novell network, so I'm using SQL Server authentication. At it's simplest, I could just have one login for my database and lock down all the tables and stored procedures to that one login. I'd like to have the security a little tighter, though, so that only users who belong to the administrative role can access the administrative procedures, only data entry members can access the data entry procedures, etc.

I've thought of the following scenarios, but none makes me happy:

- 1) Create a SQL Server login for each user of the application and assign

Re: Database security design with ASP.net and form-based authentication

them to roles. Then lock the tables and procedures down to the appropriate roles.

I don't want to do this because I want an administrative user to be able to create new application users through the Web application. This wouldn't be possible as I don't have rights to create new SQL Server logins. I'd have to go to my DB Admin each time we want to add a new user, which isn't really acceptable.

2) Use SQL application roles to secure tables and procedures. We've used these in other applications, but I'd like to stay away from them since connection pooling doesn't work with them.

3) Use a set number of SQL Logins for each pre-defined role (data entry, guest, admin, report viewer) and grant those logins permission to tables and procedures as appropriate. I think this is my favorite method right now, but then I'm not sure how to manage the multiple usernames and passwords. Where do I store them and how does the application decide which one to use?

This is where maybe this question is more appropriate in an ASP.Net group, but I thought I'd try here first.

I'm wondering what other people have done in this scenario?

Thanks,
Diane Y.