

# Lesson 8 Transcript: Database Security

## Slide 1: Cover

Welcome to Lesson 8 of the DB2 on Campus Series. Today we are going to talk about database security. My name is Raul Chong, and I am the DB2 on Campus Program Manager.

## Slide 2: Agenda

This is the agenda for today.

## Slide 3: Agenda

We will start with DB2 security overview.

## Slide 4: DB2 Security Overview

So, on this slide, we show that security in DB2 is performed in two steps. The first step is called "authentication" and that is normally performed by the operating system, or any external security facility outside of DB2. The second step is called "authorization" and that part is performed by DB2. So for example, let's say, here at the bottom, we see a user called Bob, and he is issuing a connect to the sample database, and he's using "USER bob using password" as pwd, his password is "pwd". So when he issues this connect statement, the statement will first go to DB2, it will check, first of all, authentication, and then, depending on the value of that parameter, it will perform authentication either at the client machine or the server machine, but it will be performed by the operating system at that machine or at that server.

So the authentication, again, is not performed by DB2, but by the operating system normally, or by an external facility, or even you can write your own code to perform the authentication. And this is some difference that we have with respect to other relational database management systems, because these other systems normally can define users within their database systems, but in DB2 you really cannot define a user within DB2. The users are defined outside DB2; the same thing with groups. They will be defined outside of DB2. Normally this would be operating system users or operating system groups as well. And once Bob is able to connect to... so once the operating system confirms that Bob and "pwd" is matching, so there is a user "bob" with password "pwd", then the operating system will say ok, yes... will tell DB2 that, yes, this person is OK to connect. So he will connect to DB2, and then he can issue, for example, a: **select \* from mytable**. So now, here, when the users start executing operations in the database, then DB2 will check, in this case, that user bob has "select" privilege and he can do it on the table, "mytable". So this part of the authorization, which is the second part, is performed by DB2.

## Slide 5: Agenda

Ok, so moving on to the next section of this presentation, we have authentication.

### **Slide 6: Authentication**

So, as I said before, authentication is performed by an external facility, which is normally the operating system. Now, where is it going to be performed? Is it going to be performed at the client machine where I'm issuing the "connect" statement, or is it going to be performed at the server machine where the DB2 database resides? So that is determined based on a parameter called AUTHENTICATION, which is set in the dbm cfg, it's at the instance level, basically.

### **Slide 7: Authentication (cont'd)**

So, for example, if I issue or if my AUTHENTICATION is set to "SERVER" –My "AUTHENTICATION" is set to SERVER, in this example, and a client that issues a connect statement using "user1" password "pwd1", it will go to the DB2 server and the user1/pwd1 has to exist at the DB2 server, because AUTHENTICATION=SERVER. If, on the other hand, AUTHENTICATION is set to "CLIENT", then "user1", "pwd1" can exist at the client, the authentication will be performed at the client and then I can connect this way.

### **Back to Slide 6: Authentication**

Now going back to the previous slide: there are different valid values for AUTHENTICATION: SERVER, CLIENT, SERVER\_ENCRYPT is the same as SERVER but it would be encrypting the IDs and passwords when you send them through the network; for KERBEROS as well, SQL\_AUTHENTICATION\_DATAENC and SQL\_AUTHENTICATION\_DATAENC\_CMP, and the GSSPLUGIN as well. This last one, GSSPLUGIN, allows you to create your own authentication mechanism, as a plugin to DB2. So basically, let's say, you don't like to define many users in the operating system to use DB2, so you could write your own authentication, and you can maybe list only the users that are allowed to access DB2 and then through this plugin only those people in this list will be able to connect to DB2.

### **Slide 8: Agenda**

Ok. Moving on to the next section, which is authorization.

### **Demo**

Actually before we move to authorization, I just want to show you from the Control Center, if you want to change the parameter related to authentication, you right-click on the database, and choose "Configure"... I'm sorry... right-click on the instance, and choose "Configure Parameters", and then from there you would change "AUTHENTICATION", and you can click on this button and then you can change to any of these options. Ok, now, going back to the presentation, we have authorization. As I said before, authorization is performed by DB2.

### **Slide 9: AUTHORIZATION: Authorities and Privileges**

Now, in terms of authorization, we have "authorities" and "privileges". So at the top you have authorities and these could be thought of as built-in roles, so these are roles you cannot change what they can do, it's predefined what these authorities can do. SYSADM is like God, basically, he can do whatever he wants, and then the rest will have... or there are more things they can do, but not as much as SYSADM. So we will talk more about this in the next few

slides. And then you can also have privileges, which are individual privileges to do specific operations. So you have, for example, "IMPLICIT\_SCHEMA", "BIND", "ALTER", "DELETE", etc. So, you can grant a given user a specific privilege, or a given user can be granted or be given a given authority. Now, with DB2 version 9.5, which is the latest version of DB2, there is also support for roles where you can create your own roles, which work in the same way as authorities. Authorities are predefined roles, basically. You cannot change what they can do, while roles allow you to specify what you want these roles to have in terms of what privileges they can have, and that way you can assign a role to a given user.

### **Slide 10: Other Authority Levels**

Ok, moving on to the next slide. This is the slide that shows you all the different functions or capabilities for the different authorities. So you have that SYSADM can pretty much do everything, as you can see from the list, SYSCTRL a little bit less, SYSMANT a little bit less, etc.

### **Slide 11: SYS Authorities**

OK. How do you assign a SYSADM or SYSCTRL or SYSMANT to a user? Normally by the operating system user or a group; well, there are three parameters at the instance level, the name of the parameters are SYSADM\_GROUP, SYSCTRL\_GROUP and SYSMANT\_GROUP, and you can issue an **update dbm cfg** command to assign a group, or an operating system group to these groups, or to these parameters, and then to any member in that operating system group will become immediately either SYSADM, SYSCTRL or SYSMANT.

### **Demo**

Ok, from the Control Center, again, you can right-click Configure Parameters, and then from here, if you scroll down, within the administration section you can see SYSADM\_GROUP, SYSCTRL\_GROUP, and SYSMANT\_GROUP. And from here you can specify an operating system group. Right now the values are empty. When the values are empty, by default any local administrator in Windows will become the SYSADM. And in Linux, any member of the group, that is, the group of the instance owner would be also SYSADM by default.

### **Slide 12: DBADM Authority**

Ok, moving on to the next section or slide. We have DBADM. For DBADM you cannot assign this as an instance parameter. You assign DBADM using the grant statement as shown in this example. You say **grant DBADM on database** to a given "userid". So a SYSADM could grant DBADM to a given user using the **grant** command. DBADM is a database administrator, so he basically is able to administer a given database but cannot administer things at the instance level. He cannot do that.

### **Slide 13: Launching the Table Privileges Dialog**

Now, what we show here is that you could do some administration in terms of security using the Control Center.

## Demo

So if I go back to the Control Center, so here, I already showed you that from the Control Center you can set the authentication parameter and SYSADM groups etc. But you can also do things, for example, for a given table. So let's say I choose the employee table, and from here I can right click, and I can choose Privileges. So what this allows me to do is that I can assign given privileges to a given user. Right now I have only one user selected, but I could add a user, let's say I'm going to add a user, and it's going to take information from the operating system. So let's say I pick DB2ADMIN. Right now what I can do now that I bring it to this screen, I can say ok, DB2ADMIN has "Select" privilege on table "employee", and has "Grant" privilege for "insert" on table "employee", etc. The difference between "Yes", "No" and "Grant" is, "Yes" you grant privilege for insert, "No" means you don't grant the privilege for "insert" and "Grant" means "Yes" you grant the privilege for insert and also this person can grant the same privileges to other people. Now, if we click on Show SQL, Show SQL is showing me that we are using the grant statements here for the select and a grant statement also for the insert.

So basically, again, the Control Center is a graphical tool on top of the main engine of DB2, so it's running these commands behind the scene. And by bringing a user to the Control Center we are able to make these configurations just using the GUI. Otherwise we will have to issue commands. So that's the purpose of bringing a user to the GUI. But these users come from the operating system. That means that you are not creating users from DB2.

Now, also, on the same topic, there is a folder called "User and Group Objects" within the Control Center. If you expand this you will see "DB Users" and "DB Groups". If you right click here, it says "Add". But again, don't get confused whenever you find... and I talked about this when I was talking about tools in lesson number 3, don't get confused by these words. "Add" doesn't mean you are creating a new user. "Add" means you are taking an existing user from the operating system and adding it to the Control Center. And the reason for doing this, as explained before when I was showing the privileges of the table employee, is to facilitate the administration in terms of security for a given table. So rather than issuing that **grant** command or **revoke** command, you will just use a Control Center and you just pick what a given user can have in terms of privileges, in terms of security. And then the Control Center generates a grant or revoke statement for you.

So if I choose... if I want to choose "Add", what's going to happen is you have a drop-down list, and from here you will see the different users, which are actually users from the operating system, right? And it's going to show you the ones that are already not in the DB2 Control Center. So that's why it doesn't show... it doesn't show DB2ADMIN here any more because DB2ADMIN was already added from the previous step that we did for the table employee. And from here you can select all of these different options and you could specify, at different levels, where you want to provide different privileges.

## Slide 14: Table Privileges Dialog

Ok, so going back to the presentation, this I already showed you.

### **Slide 15: Agenda**

Now we are going to move on to the next section, which is the PUBLIC group.

### **Slide 16: The PUBLIC group**

The PUBLIC group is a special group where every member or any member in this PUBLIC group is basically a user from the operating system. So that means, let's say, if I'm right now working on Windows or Linux, and I add or I create a new user, a new operating system user, immediately that new operating system user will be part of the PUBLIC group. And the PUBLIC group, by default, has four privileges. One is CONNECT, the other one is CREATE TABLE, the other one is IMPLICIT\_SCHEMA, and the other one is "BINDADD". So what does this mean? If I right now go to Windows or Linux, and I create a user, that user... I logon as that user and then I do a **connect**, I would be able to connect to a database, right? Because, again, these users from the operating system are by default going to be part of the PUBLIC group, which have these "connect" privileges. If you don't like to allow any of these privileges to public at all, you can revoke these privileges.

### **Slide 17: Agenda**

So we are going to talk in the next section about the GRANT and REVOKE statements. And I already talked about some of these statements in the authorization section.

### **Slide 18: GRANT and REVOKE examples**

Basically these are standard statements from SQL to manipulate or to work on security, as to what you can grant a user to do, and what you want to revoke from a user. Now, at the bottom here you can see that you could issue these revoke statements on connect, REVOKE CONNECT, REVOKE CREATE TABLE, REVOKE IMPLICIT\_SCHEMA, and BINDADD. And you can revoke that from PUBLIC. So when you do that you are basically locking down the database from PUBLIC to do anything. So, if you... again, if you don't want any user defined from the operating system... by creating ... you could just run these revoke statements as soon as you create a database and then nobody will be able to... nobody in PUBLIC would be able to do these operations.

### **Slide 19: Agenda**

Finally we are going to cover the last section of this presentation which is Extended Security, and this applies only to Windows.

### **Slide 20: Extended Security**

So extended security, when you allow, or when you allow extended security in Windows, which can happen when you are doing installation by default, there are two groups that are created: DB2USERS.DB2ADMNS. That means basically that DB2USERS. DB2ADMNS, in there you will basically put all of the administrators of DB2, and they will be able to do pretty much everything on DB2, and they can even access the DB2 system files through the operating system. The DB2USERS groups are normally... you put users there, and they can work with DB2 objects through the operating system, but any other user that is not part of

these groups will not be able to access the DB2 system files. So they will not be able to delete DB2 system files or the code of DB2.

**Slide 21: QuickLab #9**

Ok, so, moving on. Now I will suggest that you to pause this presentation, and take a look at QuickLab #9, and start working on these labs to practice on security using grant and revoke.

**Slide 22: What's Next**

So with this we have finished this Lesson 8 on security. So congratulations for completing this lesson. And as to what is next, I would recommend you to follow with Lesson 9 on Backup and Recovery. Thank you very much, and have a good day.