New Security Options in DB2 for z/OS Release 9 and 10

IBM has added several security improvements for DB2 (IBM's mainframe strategic database software) in these releases. Both Data Security Officers and IS Auditors will want to know about them. We cover the major ones briefly in the next few paragraphs, and then provide some specifics on how they work. We include descriptions of some earlier security functions within DB2, as background.

One of the biggest improvements is separation of data access from security administration and from system administration. Before this, the SYSADM privilege gave the DBA total access to data, the ability to manage security with the GRANT statement, and all power needed to administer the DB2 system. Now if a switch named **SEPARATE_SECURITY** is set to YES, the SYSADM privilege only gives the power to administer the system. It also activates new privileges such as SECADM for security administration and DATAACCESS for the ability to access data.

Role based access control is a feature which introduces the concept of a role, that is a collection of privileges, perhaps corresponding to a job function. AuthIDs (the name DB2 calls its userids) can be made members of a role, in which case they inherit the privileges of the role. And now, privileges can be granted to an authID or to a role.

Role based access control in DB2 depends on another new concept called trusted context. A trusted context is a combination of information describing a specific remote or local connection. This information can include: IP address, domain name, and the job name. Trusted contexts can be used to assign an authID to a connection without the need for a password to prove the user's identity.

A new type of object is the security object. Examples of security objects include roles and trusted contexts. They do not exist until created with the CREATE statement.

Another new feature lets you optionally turn off an older feature that many people have considered a bug. DB2 privileges are stored in several security tables stored in the DB2 catalog. If a user grants a privilege to another user, the result of the grant is the automatic insertion of a row describing the privilege into one of these security tables. (If you are using your security software (RACF, ACF2, or TopSecret) instead of these internal DB2 security tables, then the feature we are about to describe may be irrelevant for you.)

The original version of DB2 included a feature called cascading revoke. Imagine that, for example, USERA granted a privilege to USERB who granted it to USERC who

granted it to USERD. If the privilege is revoked (taken away) from USERA, then the others automatically lose the privileges too. This cascading revoke feature can be disabled with a new option called **REVOKE_DEP_PRIVILEGES.** It can then be re-enabled on selective REVOKE statements by adding the phrase *INCLUDING DEPENDENT PRIVILEGES* or *NOT INCLUDING DEPENDENT PRIVILEGES*..

Note: Some of these features came with DB2 for z/OS Release 9, some with Release 10, and some even earlier. Since we will all shortly be on at least Release 10, this article does not detail which feature came with which release. If you need this information, your DBA can provide the details.

Some Specifics:

We describe details of major DB2 security options in these sections:

- Earlier Security Options as Background Briefing
- How Users Are Identified
- Control Over Access to a Given DB2 Sub-System
- New Security Options Specified in DSNZPARM
- New Security Privileges for AuthIDs and Roles
- Some Additional Security Mechanisms
- Automation of Data Integrity

Earlier Security Options as Background Briefing

Each DB2 sub-system has one file where its options, including security options, are specified by the DBA. The name of this file is DSNZPARM. If you want to do a security review or audit of DB2, you need to have a source copy of this to read. Some of the earlier security options specified there include:

AUTH = YES or NO (default is YES) indicates whether security checking is active or not

SYSADM = (default is SYSADM) specifies an authID with the SYSADM privilege (which gives complete power for data access, database administration, and security administration). Note that in the future if the switch **SEPARATE_SECURITY** is set to YES, then the SYSADM privilege does not grant access to data nor the ability to perform security administration.

SYSADM2 (default is SYSADM) specifies a second authID with the powerful SYSADM privilege. These two authIDs together are called the "*install SYSADMs*" and are meant

to be used only for installing DB2 and for recovering from emergencies.

SYSOPR (default is SYSOPR) specifies an authID with the SYSOPR privilege, which is a subset of SYSADM.

SYSOPR2 (default is SYSOPR) specifies a second authID with the SYSOPR privilege. These two are called together are called the "*install SYSOPRs*".

DDF = NO or AUTO or COMMAND (Default is NO) Whether or not DDF is started automatically, or by command, or not at all. DDF stands for Distributed Data Facility, the ability of a DB2 system to talk over a network with other software on other types of computer. This is commonly used for example when a Windows or UNIX server wishes to query a DB2 database on the mainframe. The network connections can be TCP/IP or SNA.

TCPALVER = YES or NO or CLIENT or SERVER or SERVER_ENCRYPT. (Default is NO.) Stands for **TCP AL**ready **VER**ified and indicates whether DDF connections over TCP/IP are considered already to have a userid assigned to them or not. If set to YES or CLIENT, then userids are accepted without passwords or other proof of identity. If set to NO or SERVER, then users cannot connect over TCP/IP to this DB2 unless they provide both a userid and password (or other proof of identity). If set to SERVER_ENCRYPT, then in addition to userids and passwords being required, the userids and passwords must be either AES encrypted or sent over a secure port encrypted with AT-TLS (Application Transparent -Transport Layer Security).

DFLTID specifies a default userid to be assigned to batch jobs. Since the default for this default is the userid IBMUSER, RACF installations may want to change it.

How Users Are Identified

Just as important as understanding security options and privileges is an understanding of how users are identified. DB2 has a variety of ways to establish a user's identity, some of which do not always verify the identity of the user. If these ways are not properly implemented, it may be possible for a user to impersonate some other user, in order to take advantage of that other user's privileges.

In DB2 users are assigned authIDs (similar to userids). AuthIDs may correspond to userids in the security software. They may also correspond to constructs in the security software (groups in RACF, source groups in ACF2, IBMCLASS resources in TopSecret).

On the other hand they may correspond to the name of a CICS transaction or other value specified by the system programmer.

DB2 has no "list of all the authIDs and their passwords". Instead, when the user connects to DB2, two assembler language programs create the list of authIDs which identifies the user. These two assembler language programs are named **DSN3@ATH** and **DSN3@SGN**, and it is their logic which determines how users are identified**.** System programmers can modify these programs, for example to add the install SYSADM to the list of authIDs for their own connections.

For connections over TCP/IP (the DDF feature described above), the TCPALVER switch may be set to accept userids without requiring passwords. In this case you would want to know that there are compensating controls to prevent spoofing of users' identities.

A new way of identifying users is the trusted context described above. This lets you define a communication path and have DB2 accept whatever userid is provided without necessarily requiring a password. Trusted contexts can be defined with various levels of encryption as well. Again, you may want to see compensating controls to prevent spoofing of identities.

Identity propagation is a new way of identifying a user who logs onto a distributed platform and from there connects to DB2. DB2 can use the security software to trust another platform (think of Active Directory on Windows) to identify a user. Then if a user proves who he is to Active Directory and then connects to DB2 on z/OS, your security administrator can tell RACF, ACF2, or TopSecret to trust that Active Directory for that user. The user then needs to log on only once, to Active Directory. DB2 uses the security software to trust that that Active Directory has verified the user's identity already.

Control Over Access to a Given DB2 Sub-System

When a user tries to connect to DB2, DB2 calls the security software (RACF, ACF2, or TopSecret) using the resource class DSNR to ask whether that user should be permitted to connect to that DB2. There can be separate rules for connections from TSO and batch, from CICS, from IMS, and through DDF.

New Security Options Specified in DSNZPARM

Perhaps the greatest improvement of these is the separation between the system administrator's privileges (SYSADM) and the security administrator's privilege (SECADM).

SEPARATE_SECURITY = YES or NO (default is NO). When this is set to YES, the SYSADM privilege no longer includes access to data nor the ability to do security administration. In addition, the SECADM, DATAACCESS, and ACCESSCTRL privileges become active.

REVOKE_DEP_PRIVILEGES = YES or NO or SQLSTMT (Default is SQLSTMT). If set to YES, the cascading revoke feature is active (except for when the privileges ACCESSCTRL, DATAACCESS and system DBADM are revoked). If set to NO, then cascading revoke is not active. If set to SQLSTMT, DB2 decides whether to invoke cascading revoke based on whether the REVOKE statement includes the phrase *INCLUDING DEPENDENT PRIVILEGES* or *NOT INCLUDING DEPENDENT PRIVILEGES*.. (Default is INCLUDING DEPENDENT PRIVILEGES.)

SECADM1= (default is SECADM) specifies an authID or role which can perform security administration if the SEPARATE_SECURITY switch is set to YES

SECADM2= (default is SECADM) specifies a second authID or role with the SECADM privilege if the SEPARATE_SECURITY switch is set to YES. These two authIDs or roles together are called the "*install SECADMs*".

SECADM1TYPE = AUTHID or ROLE (default is AUTHID) specifies whether SECADM1 is an authID or a role

SECADM2TYPE = AUTHID or ROLE (default is AUTHID) specifies whether SECADM2 is an authID or a role

New Security Privileges for AuthIDs and Roles

SECADM gives the authID or role the ability to do security administration, but only if the switch SEPARATE_SECURITY is equal to YES. This does not give access to data, only the ability to do GRANT and REVOKE commands.

ACCESSCTRL gives the authID or role the ability do most security administration, but only if the switch SEPARATE_SECURITY is equal to YES. This does not let you do

everything the SECADM can do. (ACCESSCTRL does not let you grant: CREATE_SECURE_OBJECT, DBADM, DATAACCESS, nor ACCESSCTRL.)

DATAACCESS gives the authID or role complete access to all data in user tables, but only if the switch SEPARATE_SECURITY is equal to YES. It does not give the ability to perform security administration.

System DBADM gives an authID or role the ability to administer all databases on the system, without granting access to the data in the databases. (You can give an authID or role which has System DBADM additional privileges, for example ACCESSCTRL or DATAACCESS or even SELECT authority on specific tables.) The System DBADM privilege does not give the ability to issue GRANT and REVOKE statements.

Some Additional Security Mechanisms

While earlier sections of this paper concentrate on major new security features, the following mechanisms, are worth knowing about. Some are minor, some are not so new. We mention them briefly here for the sake of completeness.

You will see that several of them let you control access to subsets of a table, similar to defining a view on a table and then granting access to the view. Deciding when to use which of these features is beyond the scope of this paper.

Permission on rows controls access to rows in a table for an authID. Permission on rows can be used to prevent users with privileges like SYSADM, SECADM, and DBADM from accessing specific rows. If you have strong reasons to automate compliance with laws protecting sensate data, this may be useful. You use the CREATE PERMISSION DDL statement in SQL to create row permissions.

Permission on columns controls access to columns in a table based on column masks. As with permission on rows, this can be used to prevent privileged users from accessing specific sensitive data. You use the CREATE MASK DDL statement in SQL to create column masks.

Security Label Protection on columns and rows (uses security labels as defined in RACF, ACF2, or TopSecret to control access to data in tables). This lets you define a security label in the security software, permit certain users to that label, and then associate that label with a specific column in a table. Now users accessing the table can only access rows whose value in that column matches their label from the security software.

Audit Policy (stored in a table named SYSIBM.SYSAUDITPOLICIES) specifies what events are to be recorded in the audit log.

Encryption DB2 has features that let you encrypt data over the network ("data on the fly") as well as data stored in tables in memory or on disk ("data at rest"). DB2 supports TLS (Transport Layer Security) for data on the fly. For data at rest, there are a variety of hardware and software techniques to provide encryption and decryption.

Automation of Data Integrity

While data integrity is often considered separately from security, these features of DB2 make automation of data integrity much easier. We list them here because we think you should be familiar with them:

Procedures which can be associated with a column or a table. These are programs that get control whenever data is inserted or selected and can be used to convert data (for example part numbers to detailed part names). These can also be used to reject invalid data, for example refusing an invalid abbreviation of a state or province. Procedures can provide encryption and decryption of data as well.

Referential Integrity is one of the key concepts underlying the relational model on which DB2 is built. Imagine for example a table of names and addresses and other information. If you split the table into two separate tables, one containing the ZIP (postal) code and city and state, and the other table containing everything else including the ZIP code, but NOT the city and state. If you define a relation on the two tables based on the ZIP code column, DB2 will enforce referential integrity. This is a promise that for each ZIP code in the second table (the one with everything else), there will be one and only one matching ZIP code value in the first table. This means that each ZIP code can be relied upon to identify exactly one city and state combination. If you try to insert or to delete a row in either table that violates this constraint, DB2 will prevent you.

NOT NULL is a characteristic of a column that requires it to have a value

DEFAULT values can be defined on a column, so that if you insert a row and forget to specify the value for that column, DB2 will fill it in with the default value.

VIEWs defined WITH CHECK OPTION A view is a virtual table that consists of the reflection of specified columns of one or more real tables. In addition to specifying the columns to be reflected in the view, the DBA can specify that the view includes only rows where, for example, the value of a specified column equals "ABC Company". Each view can be defined WITH CHECK OPTION, which means that attempts to insert

or update data through the view will be rejected if they don't meet the constraints on the rows.

What This Means for Security Staff and IS Auditors

These new features are so attractive that at least some of them will be of real benefit to your data center. While these features are very useful and provide greater precision of security, they should of course not be implemented willy-nilly. Many of them will be desirable in your installation. But you want to have them selected, tested, and implemented carefully, after development and review of a comprehensive plan.

A reasonable stance to take for now is to learn more about these new features. Then discuss them with your DBAs to see what makes sense, which features might benefit your installation, who should conduct the evaluation, and when is a sensible time frame for them. These changes will likely lead to changes in job responsibilities, need for additional training, and development of new procedures, policies, and standards. Their implementation may be slowed by staff and budget limitations, and by the nature of the change control cycle. In the mean time, you want to be aware of the possibilities and the possible benefits.

You should recognize that the details of all the security mechanisms for DB2 and how they interact requires more information than could be presented in one paper that is brief enough to be readable. Make sure you verify any assumptions with your DBA before insisting on use of some of these features.

Summary

We have covered the major security options and privileges for DB2 for z/OS. You can learn more details from your DBA. To conduct a DB2 security review or audit, you will need some familiarity with the concepts described here. You will also want to have readable copies of DSNZPARM and the exits DSN3@ATH and DSN3@SGN. If the DB2 instance you are reviewing uses your security software (RACF, ACF2, or TopSecret) instead of the DB2 internal security tables, then you will want to be able to list the relevant resources rules from the security software. An effective security review or audit will likely answer at least these two basic questions: "***How are users identified?***" and "***Who can access the data in the DB2 tables?***" The information described here should give you a good running start.

Questions to Stu Henderson at (301) 229-7187, stu@stuhenderson.com.
More whitepapers: http://www.stuhenderson.com/XARTSTXT.HTM
Newsletters at http://www.stuhenderson.com/Newsletters-Archive.html

