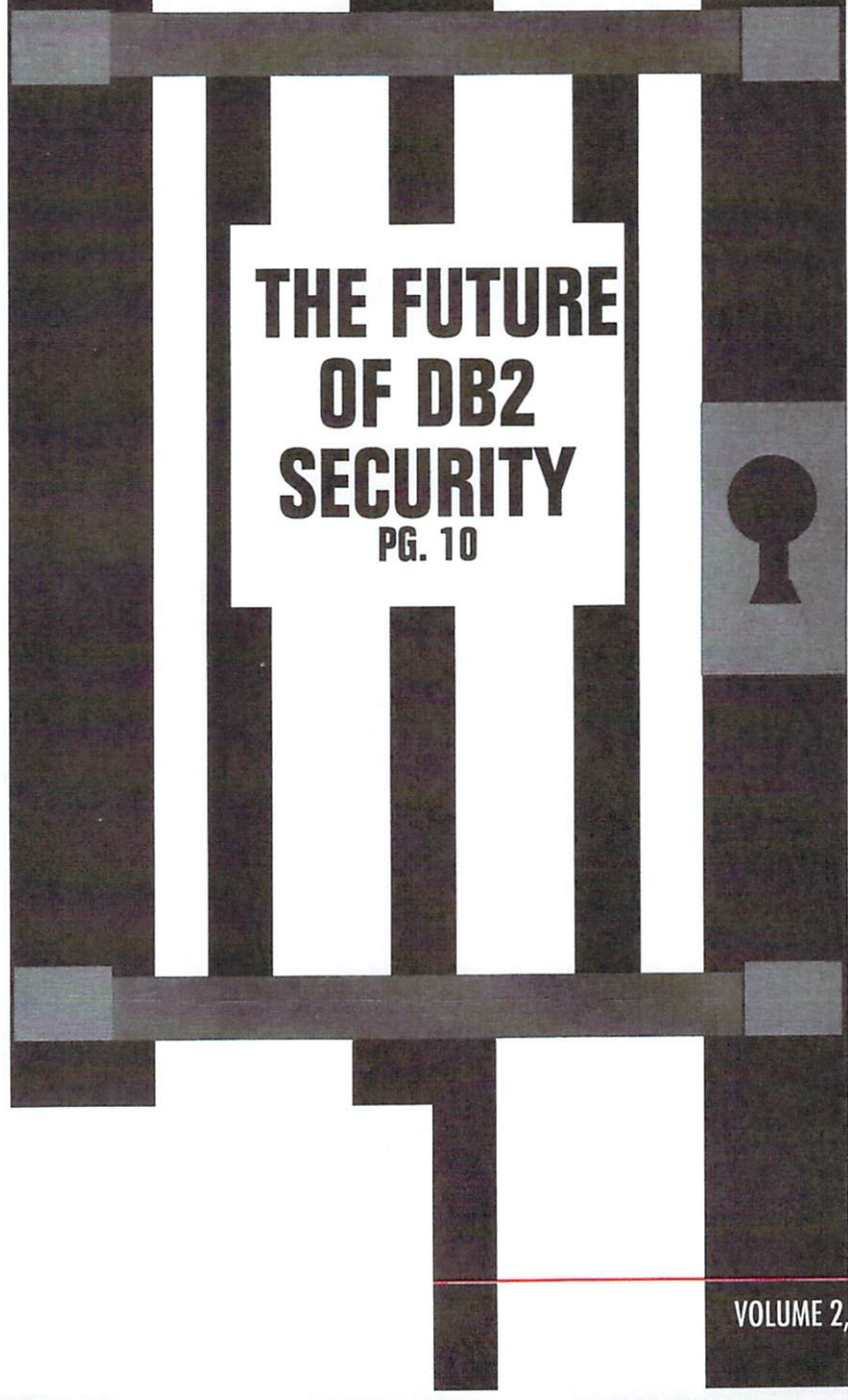
DATA BASE

APPLICATION DEVELOPMENT

MANAGEMENT



THE FUTURE OF DB2 SECURITY

PG. 10

VOLUME 2, NUMBER 7, JULY 1992

By Craig S. Mullins

THE FUTURE OF DB2 SECURITY: PART I

Access to data base objects and data is controlled by the security features internal to the data base management system that conforms to the relational model. This is often touted as an advantage for users of such a system, yet the security features of DB2 have been roundly criticized for many reasons ever since the first release of DB2. Some of this criticism is warranted and some is not.

Much of the criticism of DB2's security facilities stems from users' unfamiliarity with a new authorization scheme. Older, non-relational DBMS products do not handle data base security using internal DBMS commands. It is foreign for a DBMS to manage who is permitted to access its data. Most MIS personnel are accustomed to handling security via external software packages such as Computer Associates' ACF2 and TOP SECRET or IBM's RACF. DB2's security mechanism does not resemble the mechanisms found in these types of security packages.

The security features of DB2 are far from exhaustive, however. The valid criticism levied against DB2 focuses on the security features that do not exist, as well as those that were awkwardly or incompletely implemented. This article discusses these issues and suggests solutions.

No information on DB2 security can be complete without recognizing the new security features implemented for DB2's latest release, V2.3. These features address some of the inadequacies in DB2's security implementation. Therefore, let's first investigate the enhancements that are available in DB2 V2.3.

DB2 V2.3 Security Enhancements

V2.3 provides four major extensions to DB2's security implementation. These extensions deal with:

- assigning authority;
- environment execution control;
- DDL control; and
- BIND control.

There were a few additional levels of authorization control added to administer the new features of DB2 V2.3, such as packages and log archival upon command.

Assigning authority is the ability to enable a security control agent to GRANT and REVOKE privileges for other users, without themselves having the particular authority being granted or revoked. In all prior releases of DB2, any user issuing a GRANT or REVOKE statement had to have the explicit authorization

Figure 1: Environments That Can Be Specifically Enabled or Disabled

Environment	Description
BATCH	TSO Batch
CICS	CICS
DB2CALL	Call Attach Facility
DLIBATCH	DL/I Batch (IMS)
IMSBMP	IMS/DC BMP
IMSMPP	IMS/DC MPP
REMOTE	Remote Package

that s/he was granting or revoking. This did not follow the authorization features of the relation model V2 (RMV2)[1]. The DB2 system privileges BINDAGENT and SYSCTRL implement assigning authority.

A user who is granted the BINDAGENT authority can bind, rebind and free plans and packages specifying an OWNER other than her/his primary or secondary authorization ID. The BINDAGENT can accomplish this without having any explicit ability to execute the plan or package, or to access the underlying data in the tables referenced by the plan or package. (A package is a new feature of DB2 V2.3 that effectively breaks a plan into separately bindable components called packages.)

A user who is granted the SYSCTRL authority will have the same authority as a SYSADM with one major difference: The user will have no authority to access the underlying table data. This should reduce the number of SYSADMs granted for DB2 subsystems. Many technicians who require blanket authority to create DB2 objects and execute DB2 commands and utilities, but require minimal access to application data, can now be assigned SYSCTRL instead of the more powerful, and potentially dangerous, SYSADM authority.

Keep in mind that SYSADMs can access and update every table in a DB2 subsystem. This increases the likelihood of data integrity violations due to erroneous updates or unauthorized accesses prohibited by your EDP auditing department such as payroll or billing inquiries.

Environment execution control is implemented as an extension to the BIND command. The ENABLE and DISABLE feature provides the capability to establish the environments in which a plan or package is permitted to execute. For example, if a given plan is to be run in CICS only, this can be ensured by binding the plan specifying that only CICS is enabled. In this way, it can be ensured that plans targeted for a spe-

Figure 2: DDL for a RLST

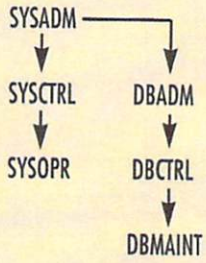
```
CREATE DATA BASE DSNRLST;

CREATE TABLESPACE DSNRLSxx
       IN DSNRLST;

CREATE TABLE authid.DSNRLSTxx
       (AUTHID      CHAR(8)      NOT NULL WITH DEFAULT,
       PLANNAME     CHAR(8)      NOT NULL WITH DEFAULT,
       ASUTIME      INTEGER,
       LUNAME       CHAR(8)      NOT NULL WITH DEFAULT,
       RLFFUNC      CHAR(1)      NOT NULL WITH DEFAULT,
       RLFBIND      CHAR(7)      NOT NULL WITH DEFAULT,
       RLFCOLLN     CHAR(18)     NOT NULL WITH DEFAULT
       )
       IN DSNRLST.DNSRLSxx;

CREATE UNIQUE INDEX authid.DSNARLxx
       ON authid.DSNRLSTxx
       (AUTHID, PLANNAME, LUNAME)
       CLUSTER CLOSE NO;
```

Figure 3: Description of the RLST Columns

Column	Description
AUTHID	Specify the primary authorization ID to whom the limit set by this row applies. If blank, this row will apply to all primary authorization IDs at the location specified by the LUNAME column.
PLANNAME	Specify the plan name for which the limit set by this row applies. If blank, this row will apply to all plan names at the location specified by the LUNAME column. Valid only when RLFFUNC is blank. If RLFFUNC contains any other value, the column must be blank or the entire row will be ignored.
ASUTIME	Specify the maximum number of CPU service units permitted for any single dynamic SQL statement. If NULL, this row will apply no limit. If less than or equal to 0, this row will indicate that no dynamic SQL is to be permitted.
LUNAME	The logical unit name of the site where the request originated. If blank, this row will apply to the local site. If PUBLIC, this row will apply to all sites.
RLFFUNC	Indicates the type of resource that this row is limiting. Blank . . . row governs dynamic SQL by plan name 1 ..... row governs BIND for plans or packages in collections 2 ..... row governs dynamic SQL by collection and package names If any other values are encountered in this column, the entire row will be ignored.
RLFBIND	Indicates if the BIND command is permitted. The value 'N' indicates that BIND is not allowed; any other value means that the BIND command is allowed. Valid only when RLFFUNC equals 1.
RLFCOLLN	Specify the name of the collection to which this RLF row applies. If blank, this row will apply to all packages at the location specified by the LUNAME column. Valid only when RLFFUNC equals 2. If RLFFUNC <> 2, the column must be blank or the entire row will be ignored.
RLFPKG	Specify the package name for which the limit set by this row applies. If blank, this row will apply to all packages at the location specified by the LUNAME column. Valid only when RLFFUNC equals 2. If RLFFUNC <> 2, the column must be blank or the entire row will be ignored.

Figure 4: Group Level DB2 Privileges

Hierarchy of Group Level Privileges



System Privileges

SYSOPR: DISPLAY, RECOVER, STOPALL, TRACE

SYSCTRL: SYSOPR + DBCTRL + DBMAINT + BIND, BINDADD, BSDS, CREATEDBA, CREATEDBC, CREATESG, EXECUTE, MONITOR1, MONITOR2, STOSPACE, CREATEIN, ARCHIVE, USE BUFFERPOOL, USE TABLESPACE, USE STOGROUP

SYSADM: SYSCTRL + DBADM

System privileges span all objects. Therefore, someone granted a system privilege has access to all objects in the system that are controlled by the given authority.

Data Base Privileges

DBMAINT: CREATETAB, CREATETS, DISPLAYDB, IMAGCOPY, STARTDB, STATS, STOPDB

DBCTRL: DBMAINT + DROP, LOAD, RECOVERDB, REORG, REPAIR

DBADM: DBCTRL + ALTER, DELETE, INDEX, INSERT, SELECT, UPDATE

Data base privileges apply at the data base level. This means that a user with a data base privilege has the given authority only on those data bases explicitly granted.

cific environment will be run in that environment only. Valid target environments are outlined in the chart shown in Figure 1.

The addition of DDL control establishes controls for prohibiting non-registered programs from executing DDL statements. A program that can issue DDL should be very closely monitored to ensure that it is not consuming too many system resources and is not being executed during an undesirable time frame (e.g., during a period of heavy online transaction processing). By establishing an application registration table that contains only those programs that are permitted to issue DDL, this level of control can be systematically enforced. The primary purpose of this table is to ensure that only known and registered programs can execute DDL. Also, an object registration table can be implemented in tandem with the application registration table. An object registration table lists objects that can be affected by DDL issued from an application program. This effectively limits both the programs that can issue DDL and the objects that can be impacted by the DDL. DDL control is optional and is activated by DB2 installation options.

BIND control is implemented through an extension of the Resource Limit Facility (RLF). By coding RLF table entries, BIND activity can be explicitly prohibited by LUNAME, collection ID, package name, plan name, primary authorization ID or any combination thereof. This is implemented using a Resource Limit Specification Table (RLST) as depicted in Figure 2.

The exact functionality of the RLF is beyond the scope of this article. By placing entries into an RLST, the RLF can limit:

- dynamic SQL access by plan and/or package; and
- execution of the BIND command.

A brief synopsis of the RLST columns and their functionality is provided in Figure 3.

Finally, DB2 V2.3 provides two miscellaneous authorization extensions that manage access to the new features of V2.3:

- CREATEIN controls the ability of users to bind packages in specific collections; and
- ARCHIVE controls who can execute the new ARCHIVE LOG command, which is used to explicitly request that the current active DB2 log be archived.

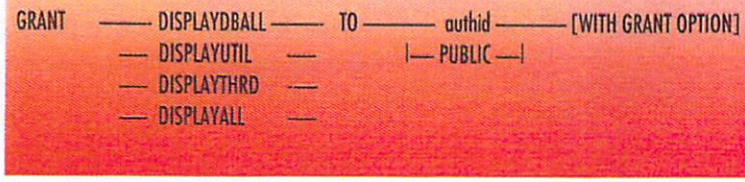
What Is Still Missing?

Even with the extensive security enhancements that have been implemented for DB2 V2.3, areas for improvement still abound. The following sections outline problems that still exist with DB2 security. In addition, a solution will be proposed for each problem that is addressed. I have made every effort to conform to the specifications of RMV2 when formulating these proposed solutions.

Problem #1: *Undesirable and Imprecise Definition of SYSOPR*

The SYSOPR group level authority contains the STOPALL privilege. Refer to Figure 4 for a listing of the DB2 group level authorities. The STOPALL privilege controls who can issue the -STOP DB2 command. -STOP DB2 brings down the DB2 address spaces. The problem is that SYSOPR also contains the implicit authorization to issue the -TERM UTIL command. When utilities abend and they can not be restarted, they must be terminated before they can be rerun. This duty typically falls to an operations control clerk or sometimes to a production job that can be submitted by an operations

Figure 5: Requisite Display Privilege Commands and Syntax



control clerk. However, there is no explicit means of granting the authority to terminate utilities. Therefore, the SYSOPR privilege must be granted to the clerk or job performing the -TERM UTIL command, thereby giving them the ability to shut down DB2. Not only is this confusing (that there is no explicit TERM UTIL privilege), but it is a probable audit exception for most shops.

Solution #1: *Create TERMUTIL Privilege and Redefine SYSOPR*

The solution is to create a new privilege, say TERMUTIL. It can be used to grant only the ability to issue the -TERM UTIL command. The proposed syntax follows:

```
GRANT TERMUTIL TO — authid —[WITH GRANT OPTION]
                 — PUBLIC —
```

The TERMUTIL privilege should still be a component of the group level SYSOPR privilege.

Problem #2: *Undesirable Definition of the DISPLAY Privilege*

The DISPLAY privilege allows the grantee to issue the following commands:

```
-DISPLAY THREAD
-DISPLAY UTIL
-DISPLAY DATA BASE (for all data bases)
```

There is no mechanism within DB2's security structure to explicitly grant the first two privileges. This is troubling. There is a DISPLAYDB privilege that can be granted on a data base by data base level, but there is no explicit privilege granting the ability to display all data bases. This is a confusing and restrictive structure.

Solution #2: *Create New Display Privileges*

The capability to grant the use of each of these commands should be provided by DB2. Furthermore, each of these privileges should be able to be granted singularly and explicitly, or as a group level that encompasses all types of displays. Figure 5 is a proposal for the requisite display privilege commands and syntax.

DISPLAYALL authority will act the same as the current DISPLAY authority, permitting all three types of display commands to be issued. I feel that DISPLAYALL would be more indicative of the authority encapsulated within the privilege than simply naming the privilege, DISPLAY. When DISPLAYDBALL is granted to a user, it will permit her/him to display the status of all data bases known to the DB2 subsystem. DISPLAYUTIL and DISPLAYTHRD control who can display the status of utilities and threads, respectively.

The second part of this article will focus on the remaining three specific problem areas surrounding the current implementation of DB2 security. Each problem will be defined and solutions will be proposed. So don't miss the next installment!

[1]Dr. E.F. Codd very precisely defines the features of RMV2 in his excellent book *The Relational Model for Database Management Version 2* (Addison-Wesley, ISBN 0-201-14192-2).

Craig S. Mullins is vice president and cofounder of Asset, Inc., a Pittsburgh-based consulting and software development firm.

Was this article of value to you? If so, please let us know by circling Reader Service No. 31.

Input/Output

We Want You!

If you enjoy reading the informative articles in ***Data Base Management*** magazine, how about adding your expertise to our pages?

Who Me?

As a member of the technical world, you have valuable experience to share with your peers. Here's your opportunity to assist the people who are responsible for the management of data bases. Maybe you'd like to share your trials and tribulations of solving a particular migration problem, designing a new data base or explain the results of evaluating a particular application generator product. We are looking for articles covering any topic related to IMS, IDMS, SQL/DS, DB2, DATACOM, ADABAS, TOTAL, ORACLE, applications development and other related data base concerns and their respective associated services, drivers and software.

But I've Never Written Before...

Even if you've never written for publication before, consider taking the time to record your experiences and knowledge for our pages. Don't worry, the ***Data Base Management*** editorial staff can always polish up your writing for you. If you need additional guidelines, the Writer's Kit provides valuable information regarding style, content and format of article submissions. Call or write for your guide!

How Should I Submit My Article?

Send a copy of the article along with a 5.25" or 3.5" diskette in either ASCII or WordPerfect 5.0 format. Your article will promptly be reviewed for an upcoming issue based on editorial topic. The deadline for articles is three months prior to the targeted issue (i.e., articles for November must be received by July 31).

So show us your stuff and become an active part of the information exchange in ***Data Base Management*** magazine!

Send submissions or Writer's Guide requests to:

Data Base Management magazine
4811 S. 76th St., Suite 210
Milwaukee, WI 53220
Attn: Editor, Tom Sprague