To: MTB Distribution

From: Jim Gray

Date: March 17, 1981

Subject: Effects of Security on the MRDS Interface.

Send Comments by:

Multics mail on System M to JGray.Multics

Telephone HVN 341-7463 or 602-249-7463

Continuum Meeting mrds_security, link to transaction 392.

1. INTRODUCTION

This MTB collects the changes in the behavior of MRDS due to the new security approach in a single document, and in simple terms.

The new security approach is outlined in [1]. That MTB references other MTB's that give a more detailed explanation than provided here.

2. DEFINITION OF NEW SECURITY TERMS

A database is said to be secured, if the version 4 (MR8 MRDS release) database has had the new command secure_mrds_db run against it.

A submodel is said to be secure, if it is a version 5 (attribute level control capability) submodel, and it resides under the database in the secure.submodels directory.

A DBA (database administrator) is a person that holds "sma" access on the database directory.

A non-DBA is any other Multics user, other than a DBA. A non-DBA may or may not have had a secure submodel created for him by a DBA.

Attribute level control is the scheme described in [1]. This scheme uses secured databases, secured submodels, and non-DBA access restricted to secured submodels.

A model view, is the view of the database obtained by opening the database using the actual database pathname in the call to dsl_$open.

A submodel view is obtained when the call to dsl_$open uses a submodel pathname. This view is usually a subset of the full model view.

3. INTERFACES RESTRICTED TO THE DBA

The commands secure_mrds_db (SMDB), adjust_mrds_db (AMDB), and quiesce_mrds_db(QMDB) are usable only by a DBA. This is the case regardless of the secured state of the database.

The AMDB and QMDB commands are restricted because malicious use of these commands can lock out database users, or destroy concurrency control.

Obviously SMDB can not be allowed to be used by just any non-DBA, to secure or un-secure the database as he sees fit, if true security is to be provided.

4. INTERFACES RESTRICTED TO THE DBA FOR SECURED DATABASES

Once the database is secured, non-DBAs must only be able to access the database through a secured submodel for the attribute level control scheme to work. Thus interfaces to the model, rather than a submodel, must be restricted to a DBA. These include the command display_mrds_dm, and the subroutine interfaces dmd_, and mmi_ (the new MRDS_model_interface).

In addition, secure submodels must not be able to be counterfeited, allowing non-DBAs unauthorized access, that was not strictly granted by a DBA. Since submodels not in the secure.submodels directory can not be used against a secure database, they present no problem. However, the ability to create a secure submodel via either the create_mrds_dsm -install option, or having append access on the secure.submodels directory must be restricted. Thus CMDSM will only be usable by a DBA, once the database is secured, and the granting of append access on the submodel directory will be advised against.

The command update_mrds_db_version will not be usable against a database created with the -secure option for create_mrds_db, or that has had it's secure bit set via secure_mrds_db. This is because UMDV operates with model openings, and a secured database requires submodel openings only.

5. SECURED DATABASE NON-DBA INTERFACES

Those interfaces that a non-DBA is restricted to once the database is secured include the subroutine interfaces dsmd_ (obsolete) and msmi_ (new).

Of course, some interfaces are available to the non-DBA but change behavior, as noted below. These include msmi_, the linus list_db request, display_mrds_dsm, display_mrds_db_status, create_mrds_dm_include, create_mrds_dm_table, display_mrds_db_access, and the open and scope setting interfaces.

6. INTERFACES CHANGING BEHAVIOR FOR SECURED DATABASES

Non-DBA available interfaces that allow submodel views, but also work on model views, must be usable only through secured submodels once the database is secured. This is because allowing these interfaces to look at the full model view, would allow knowledge to the non-DBA, of information not in his secured submodel view. These include the commands create_mrds_dm_include, create_mrds_dm_table, display_mrds_db_status (extended to submodel views for this release), and the new command display_mrds_db_access. Also included is the linus list_db request, which can not show model information for a secured database. The same applies to the model information normally returned by the msmi_ (new) and dsmd_ (obsolete) interfaces for un-secured databases.

Now consider interfaces that are available to both the DBA and non-DBA's. The database opening interfaces must be restricted to secured submodels once the database is secured. These include the LINUS request of open, the mrds_call command function of open, and the dsl_$open subroutine.

Interfaces that work with security display must change from working off of strictly Multics acl's, to adding MRDS access modes of attribute level control, once the database is secured. These include the commands display_mrds_dsm, and display_mrds_db_access.

The same considerations can be applied to interfaces that *detect* security violations. They must change from Multics acl's only to adding MRDS access. These include the mrds_call set_scope function, the LINUS set_scope request, and the dsl_$set_scope subroutine. (see [1] on why most data access violations are detected at scope setting time, rather than at data reference time)

7. INTERFACES NOT AFFECTED BY SECURED STATE OR DBA-NESS

Most of the remaining MRDS and LINUS interfaces are unaffected by the new security approach. These include display_mrds_db_version, display_mrds_open_dbs, display_mrds_temp_dir, set_mrds_temp_dir, display_mrds_scope_settings (new), dsl_ entries related to the above other than open and set_scope, mrds_call functions related to the above, other than open and set_scope, LINUS requests related to the above other than open, set_scope, list_scope, and list_dbs.

The command create_mrds_db is not affected, except for the addition of a -secure option, to secure the database at creation time.

8. DETAILED CHANGES

The scope display interfaces, and the scope documentation will be changed to refer to the modes append_tuple, delete_tuple, read_attr, and modify_attr, in order to agree with the MRDS attribute level control access modes. This affects the interfaces dealing with scope including mrds_call get_scope, and the command display_mrds_scope_settings.

[1] references other MTB's giving the details of the required interface changes for security purposes, as well as other planned interface changes related to bug fixes, and improvements.

10.0 REFERENCES

[1] MTB-501, The New MRDS Security Apporach, J. Gray

[2] MTB-503, Changes to the MRDS Command Interface, J. Gray

[3] MTB-504, Changes to the MRDS dsl_ Subroutine interface, J. Gray

[4] MTB-505, Changes to the MRDS dmd_ Subroutine interface, J. Gray

[5] MTB-506, Extension to the create_mrds_dsm and display_mrds_dsm Commands for MRDS security, N. Davids

[6] MTB-496, Proposed Changes in the MRDS Submodel Interface, N. Davids

APPENDIX - TABLE OF EFFECTS

DBA ONLY

adjust_mrds_db
create_mrds_dsm -install option
secure_mrds_db
quiesce_mrds_db

SECURED DB - DBA ONLY

create_mrds_dsm
display_mrds_dm
dmd_
mmi_

SECURED DB - RESTRICTED TO SECURED SUBMODELS FOR NON-DBA

create_mrds_dm_include
create_mrds_dm_table
display_mrds_db_status
dsl_$open
linus open
mrds_call open

SECURED DB - ACCESS VIOLATION DETECTION/DISPLAY CHANGE

display_mrds_db_access
display_mrds_dsm
dsl_$open
dsl_$set_scope
linus list_db
linus open
linus set_scope
msmi_
mrds_call open
mrds_call set_scope

UNAFFECTED

display_mrds_db_version
display_mrds_open_dbs
display_mrds_scope_settings
display_mrds_temp_dir
dsl_ (other than open or set_scope)
dsmd_
linus (other than list_db, open, or set_scope)
mrds_call (other than open or set_scope)
set_mrds_temp_dir

NEW OPTIONS

create_mrds_db -secure option
create_mrds_dsm -install option

SCOPE DISPLAY CHANGES

display_mrds_db_status
display_mrds_scope_settings
dsl_$get_scope
linus list_scope
mrds_call get_scope

UNUSABLE ON A SECURED DB

update_mrds_db_version