To: Distribution

From: Jerold C. Whitmore

Date: July 19, 1974

Subject: Multics Data Security and Access Control

I. Requirements for Data Security

Computer data security is an important topic today at the federal, state, and local levels of government, and in the commercial environment. The needed security precautions affect all aspects of the operation of a computer system, from computer room locks and backup tape vaults to internal access controls on data and correctness of the operating system. Work is being done toward making computers "secure." However, true data security can be achieved only when all aspects of the system operation can be certified secure; achievement of this goal is unfortunately still years ahead.

In the meantime, some current efforts are aimed at defining the access control mechanisms within an operating system which will provide the necessary degree of control over direct access to information. Three abstract mechanisms have been found to be required for providing this degree of control:

a. Multiple state hardware to isolate the access control mechanisms (and the operating system) from unauthorized actions of users.

b. Discretionary access controls which allow individual users to grant specific access to other users.

c. Administrative access controls which will limit the effect of the discretionary access controls throughout the system.

Fortunately, controlled sharing of information among users has been one of the fundamental design goals of Multics, making satisfaction of the above requirements relatively easy. The Multics ring and access control list mechanisms, which meet the first two requirements, are evidence that the data security issues which faced the Multics development community were

---

successfully resolved. The need for the third requirement was not obvious until after Multics had been announced as a standard product.

When looking for a solution for the third requirement, notice that the ring access control mechanism is the only one administered on a system-wide basis. However, the limit of only eight rings with their hierarchically-ordered privileges does not provide the degree of control needed to satisfy the requirement. Also, rings do not generalize well for the protection of other objects in the system, such as terminals.

Therefore, to satisfy this third requirement, the Multics standard product is being provided with a different type of access control which will serve to contain the actions of a user. Within the boundaries of this containment, the Multics system will look the same as it does today. The containment concept will be very useful at many installations. For example, a service bureau may wish to sell the ability to isolate its major customers, so that the customer's employees cannot accidently or deliberately "give away" access to proprietary data.

The basic internal access control mechanism has been described in several MTBs on the security controls enhancements. These MTBs have stressed the military application of this new access control mechanism. However, even though the military security system has provided a model to work from, the implementation goal is to provide a mechanism which will prove useful for many applications other than the military. Thus, it is misleading to use military terms in its description.

The following sections describe the new access control mechanism in terms which are both descriptive of their data security functions and chosen to stress their wide application. This description is not intended to be exhaustive, but rather to introduce the new terms in context. I propose that these terms be used in all future coding and documentation.

II. Multics Data Security Administration

Data security on Multics depends on the correct operation of its access control mechanisms and on the correct administration of these controls. The correct operation is a reliability issue and is not of primary concern here. The administration of access control has been a function of the system administrator in setting the initial ring for projects and ensuring correct ACLs on gates and system control data segments.

These security related functions are logically distinct from system administration, though inseparable under the current implementation. It is desirable to have these functions

separated so that a site can have the option of delegating the responsibilities of security and resource administration to different persons. In the future we may be able to separate security administration from resource administration completely, but for now it is useful to separate the logical roles of the System Administrator (SA) from the System Security Administrator (SSA). This is possible since normal SA administrative functions are controlled by a limited subsystem to ensure correct actions in security sensitive areas. It is the responsibility of the SSA to verify the correctness of the limited subsystem used by the SA. Currently a user of the "SysAdmin" project, but without the limited subsystem, is an SSA.

The administration of the new access control mechanism will be another duty of the System Security Administrator. During the upcoming implementation, there will be no binding of the new SSA functions to the "SysAdmin" project, as a first step toward total separation of the SA and SSA.

III. The Multics Access Isolation Mechanism

The new access control mechanism is called the Multics Access Isolation Mechanism (AIM). The Multics AIM provides an administrative control over all users of the system to ensure that an individual cannot give users access to information when these users are not authorized to see it.

Authorizations

The process is the active agent of the user on Multics. The user's system wide authorization to access information is assigned to his process at creation time in the form of:

<max_access_authorization> and

<access_authorization>.

The <max_access_authorization> of a process is the least authorization from: the <access_authorization> assigned by the SSA for the personid and projectid; and the project administrator defined user <access_authorization>. The <max_access_authorization> is the greatest <access_authorization> which can be allowed for the user (process_group_id).

The <access_authorization> of a process is the least authorization from: <max_access_authorization>; the <access_authorization> assigned by SSA for the terminal; and the user's login option (or default <access_authorization>).

Thus, the <access_authorization> of a process is its current authorization and is static for the life of the process. It can be controlled by the user at login time but must be in the range of system-low authorization through <max_access_authorization>.

Protection of Objects

The Multics AIM is designed to provide access isolation or containment for the set of objects it protects. Currently this set of objects includes: segments, directories, processes, ipc messages, and message segment messages. Protected objects in the system are given an attribute,

<access_class>,

which describes the <access_authorization> needed to perform operations, such as read, write, send, receive and execute, on the object.

The <access_authorization> of a process is compared to the <access_class> of the object it is attempting to reference, to verify that the operations to be performed are allowed within the Multics AIM. When the object of an operation is a process, as in the case of sending an ipc wakeup, the <access_authorization> of the target process is taken to be its <access_class>.

The Framework of the Multics AIM

The fundamental concept of the Multics AIM is a general administrative access control mechanism. The implementation has provided for both the <access_authorization> of a process and the <access_class> of an object to be represented by identical bit strings. Only three system modules understand the interpretation of these bit strings and the entry points of the three modules are described by their access control functions. The Multics ring access control mechanism protects the <access_authorization> and <access_class> data used for access control decisions. This framework provides for the evolution of the decision algorithm as customer needs change.

The Multics AIM has been designed with suitable defaults to ensure that the mechanism is totally invisible to users when its features are not applied at a given installation; there will be no measureable performance change. When the mechanism is applied, users will have a few new commands, but the existing user interface will not change except for the desired access restriction. The mechanism will only affect users with a higher authorization than the default. Thus, it can be applied slowly as requirements expand with no noticeable effect on other users.

IV. Isolation Strategy

Interpretation of Access Attributes

An <access_authorization> or an <access_class> contains two components, a <category_set> and a <sensitivity_level>, defined as follows:

<category_set> := <access_category(1)>....
...<access_category(18)>

<access_category(i)> := 1|0, 1 indicates inclusion in the <category_set>

<sensitivity_level> := 0|1|2|3|4|5|6|7

Each <access_category> in the <category_set> represents a logical protection compartment. The meaning assigned to each of the 18 <access_category>s is defined by the installation, as is the meaning of each of the 8 <sensitivity_level>s.

Objects are considered to be more highly protected if they require a higher <access_authorization> to read them. The term "higher" indicates more <access_categories> in the <category_set> and/or a larger value of <sensitivity_level>. The empty <category_set> and a <sensitivity_level> of zero is the system-low authorization.

The relationships between access attributes can be best described if the authorization or class identity is temporarily ignored. Consider two access attributes, A and B, each with a <category_set> and a <sensitivity_level>. The four possible relationships between A and B are defined as follows:

1. A is equal_to B when the <category_set>s are identical and the <sensitivity_level>s are equal.

2. A is less_than B when A's <category_set> is a subset of B's <category_set> and when A's <sensitivity_level> is not larger than B's <sensitivity_level> and A and B are not equal.

3. A is greater_than B when A's <category_set> is a superset of B's <category_set> and A's <sensitivity_level> is not smaller than B's <sensitivity_level> and A and B are not equal.

4. A is isolated_from B when A is neither equal to, less than, nor greater than B.

With these relationships defined, we see that the term "higher" is the same as "greater than" and the term "lower" is the same as "less than". Two other terms are frequently used to describe changes to an <access_class>. The term "upgrade" means to make

the <access_class> of an object greater than it was. The term "downgrade" means to make the <access_class> of an object less than it was. The "upgrade" and "downgrade" operations are not directly available to users other than the SSA as mentioned later.

The Isolation Decision Algorithm

The system procedures which must make decisions about what type of access to allow for a process will call a new procedure, giving the <access_authorization> and <access_class> as arguments, to determine the operations that are allowed by the Multics AIM. The operations are described as follows:

1. READ operations such as read, status, execute, and load are allowed if the <access_authorization> is greater than or equal to the <access_class>.

2. WRITE operations such as store, modify, append, write and send are allowed if the <access_authorization> is less than or equal to the <access_class>.

3. READ/WRITE operations combined in the same process are allowed if the <access_authorization> is equal to the <access_class>.

4. A process is not allowed to perform any type of operations on an object if the <access_authorization> is isolated from the <access_class>.

The allowed operation data from the AIM is used by the access decision procedures to further restrict the access which would otherwise be granted by Multics. The allowed operations do not grant any form of access by themselves.

In some cases a WRITE operation without a READ operation allowed is not meaningful or useful, but can be destructive, e.g., for segments as opposed to ipc messages. In these cases, the access decision procedures will provide further restrictions to eliminate the possibility of WRITE only objects.

Storage System Support of the Multics AIM

All segments and directories have an <access_class>. Processes may not "modify" or "append to" directories which have an <access_class> less than or isolated from the <access_authorization> of the process. All segments created by a

process will have an <access_class> equal to that of its parent directory. A process is allowed to create a directory which has an <access_class> that is greater than the <access_class> of its parent, but not greater than the <max_access_authorization> of the process. Such a directory is called an "upgraded directory."

When attempting to initiate a segment, a process will not search a directory with an <access_class> which is greater than or isolated from the <access_authorization> of the process. This is because it would be impossible for the process to perform any operations on the segment, due to the monotonically increasing <access_class> in a subtree of the storage system.

Some Implications of the Multics AIM

Some implications of the access isolation functions provided by the Multics AIM are:

1. Since some processes may have an <access_authorization> equal to system-low and the <access_class> of directories is monotonically increasing, all system directories must have an <access_class> of system-low. This includes: the root, udd, pdd, sss, lang, doc>**, sc1, ddd, idd and others.

2. Within the universe of objects which have an <access_class> equal to the <access_authorization> of the process, the other Multics access control mechanisms operate exactly as they did before the addition of the Multics AIM.

3. If an installation is using only a single <access_category> per <category_set>, information will not be permitted to pass between <category_set>s.

4. A process is allowed to read segments within its <category_set> or any subset thereof, up to the <sensitivity_level> of its <access_authorization>, but it will only be able to write in segments when the <access_authorization> and <access_class> are equal.

5. A process will only be able to send ipc messages to processes which have an <access_authorization> equal to or greater than that of the sending process. Two way ipc communication is allowed only between processes of equal <access_authorization>.

6. The default <access_authorization> of a user registered by the System Administrator (without further authorizations provided by the SSA) is the system-low authorization.

7. At login time, a user may request that his process be created with fewer <access_category>s in its <category_set> than the

user's maximum authorization (he will not be permitted to have more). Also, he may request a lower <sensitivity_level> than his maximum authorization (he is not permitted to obtain higher). Thus, system-low is a common authorization for all users of the system.

8. A process with a given <access_authorization> can "upgrade" information of a "lower" <access_class> to one equal to its <access_authorization> by copying. A process cannot "downgrade" information since it can only write in objects which have an <access_class> equal to or greater than its <access_authorization>. Only the SSA is allowed to directly "downgrade" the <access_class> of objects, thus allowing them to be read or written by processes with a lesser <access_authorization>.

EXAMPLES:

Commercial_Environment

An obvious application of the Access Isolation Mechanism for a commercial environment is to assign each functional area within a company or each major user of a service bureau to one <access_category>. The individual users will be allowed to work up to a given <sensitivity_level> according to the proprietary nature of their positions or the degree of privacy needed within their <category_set> (as for personal data within the personnel department). The access control list on segments still provides the discretionary access control needed for personal privacy or controlled sharing within the restrictions of the Multics AIM.

Government_Environment

The Access Isolation Mechanism directly meets the requirements of the DoD Information Security Program Regulation. In this context the <access_authorization> corresponds to the clearance of the user. The <access_class> of objects corresponds to the classification of the object. The <sensitivity_level> is used as the classification level (e.g., unclassified, confidential, secret) and an <access_category> would be used for compartmented security as a formal need-to-know compartment (e.g., nuclear, crypto). The Access Control List on segments corresponds to the "owner-defined" need-to-know authorization.