PROJECT MAC
Computer Systems Research Division

May 21, 1973
Request for Comments No. 21

INSTANCE TAGS IN MULTICS

by Michael D. Schroeder

In this RFC a proposal is made to use the instance tag portion of a Multics principal identifier to add a new capability to the Multics access control mechanisms. The instance tag portion of a Multics principal identifier currently is used to differentiate interactive and absentee processes. It is not clear that this application of instance tags meets any interesting protection needs. An alternative application for instance tags, which seems to patch a deficiency in the current Multics access control mechanisms, is to allow a user to specify at login the instance tag portion of the principal identifier to be associated with the process being created. User-specified instance tags would allow each user to create several protection compartments for his files. Such compartments appear useful in limiting the access of borrowed procedures to the stored information of the borrower.

It is implicit in the design of Multics that a borrowed procedure assumes the access privileges of the borrower's principal identifier. Thus, every borrowed procedure is a potential Trojan horse, i.e., it could contain code to alter or steal information accessible from the borrowing process. If a user has particularly sensitive information accessible to his principal identifier, then he is effectively constrained from borrowing procedures from all but the most trusted sources.

When the possibility of a general mechanism for supporting user-constructed protected subsystem is contemplated*, it becomes apparent that this problem could prevent a user who constructs and maintains a protected

---

* Dick Bratt and I currently are working on a proposal for a minimal set of modifications to Multics to allow all users to construct protected subsystems and to allow each user potentially to borrow the protected subsystem of any other user.

---

subsystem from ever borrowing a protected subsystem provided by another user. The borrowed protected subsystem would assume the access privileges of the borrower's principal identifier and thus could alter or steal the borrower's protected subsystem.

One method of protecting sensitive information from borrowed procedures is to allow users to login under different principal identifiers at different times. A user then could restrict access to sensitive information to one of his principal identifiers and be very careful when logged-in under that identifier not to borrow questionable procedures. Another principal identifier, which did not have access to the sensitive information, could be used when borrowing potentially dangerous procedures. The availability of multiple principal identifiers to a user also would provide the user with means to prevent himself from erroneously altering sensitive information while performing unrelated computations.

My proposal is that, with appropriate restrictions, users be allowed to specify at login the instance tag portion of the principal identifier to be associated with the process being created. Because the instance tag allows many distinct principal identifiers to be created for a given person/project combination, the several protection compartments for a given user are created without upsetting the use of person names and project names as unique identifiers for persons and projects, respectively, and without the requirement for clumsy administrative intervention on a per user, per compartment basis.

The Default ACL Term

In order for the proposed new application for instance tags to be most useful, a second change to the system is required as well. Currently the append_branch and make_seg gates to the supervisor cause a default ACL term of the form

Person_name.Project_name.* <mode>

to be included automatically in the ACL of the new segment. Always using a third component of "*" in this default ACL term defeats the purpose of user-specified instance tags, for it makes all new segments a user creates accessible from all of the user's protection compartments, as defined by different instance tags. In order for user-specified instance tags to work properly, some control over the default ACL term is required. At the very

least, when a principal identifier corresponding to a protection compartment for sensitive information is in use, then the default ACL term should use a third component that precisely matches the instance tag. When a principal identifier for information of low sensitivity is used, then a third component of "*" can be used in the default ACL term.

The Proposal in More Detail

Detailed below are a minimal set of changes and additions to the system for implementing protection compartments for users, as discussed above.

1. Add a control argument to the login command to allow optional user specification of the instance tag portion of the principal identifier that will be associated with the process being created. If no instance tag is specified then the value "a" is the default.

2. Provide means in the Project Master File to control user specification of instance tags at login. The control mechanism should allow specification of whether or not a particular user can specify the instance tag when he logs in, and if he can should allow specification of the particular instance tags that are acceptable.

3. Provide a repository in ring 0 for the default ACL term of a process. Recorded there when a process is created should be the precise person, project, and instance tag values for this term. Determining the initial value for this term could be done in a variety of ways. A simple method that seems to cover most cases is to use "person_name.project_name.*" if the instance tag portion of the process's principal identifier is "a", else use "person_name.project_name.instance_tag". The supervisor gates hcs_$append_branch and hcs_$make_seg should generate the ACL for a newly created segment by adding this default ACL term, with the mode bits specified by the caller, to the appropriate initial ACL. Note that many standard service system procedures create new segments via the supervisor gate hcs_$append_branchx. In this case the default ACL term is not used. Rather, the caller of hcs_$append_branchx supplies an entire ACL term to be added to the appropriate initial ACL. In most cases, the term to be added is generated by a call to get_group_id$tag_star, which provides a term of the form "person_name.project_name.*". If ignored, the continued use of get_group_id$tag_star could make the use of user specified instance tags very awkward by frequently (and quietly) making access to newly

created segments available to all protection compartments of a user. In general, all calls to get_group_id$tag_star, as well as its companion, get_group_id, should become calls to a new function get_default_acl_term. This function should return the default ACL term as recorded in ring 0. In addition, new functions get_principal_id and get_principal_id$tag_star should be made available for those (few) situations where special ACL manipulations are really required. Note that get_principal_id is the same as the current get_group_id. Changing its name will force all present uses of get_group_id to be reconsidered and mapped to get_default_acl_term or get_principal_id, whichever is appropriate. It is expected that get_default_acl_term will be appropriate in almost all cases.

4. Change the absentee process creation algorithm so that the principal identifier of an absentee process is identical to or functionally related to the principal identifier of the creating interactive process. The important point is that the absentee process' principal identifier not be specifiable by the user procedures which cause the creation of the absentee process. If this principal identifier were program specified, then a borrowed procedure could gain unauthorized access to a protection compartment of the borrowing user simply by submitting an absentee process creation while executing in a process of the borrowing user. The request would specify the principal identifier for the sensitive protection compartment and specify the execution of a malicious procedure in the absentee process.

An Example

Now consider an example of the use of user-specified instance tags. First of all note that users unconcerned with sensitive information perceive no change in the system, unless they look hard at the instance tag for absentee processes. A user with sensitive information to protect enlists the cooperation of his (friendly) project administrator to be provided with two directories immediately inferior to the project directory, as in Figure 1. One, named "Schroeder" in the figure, is the normal working directory of the user, and the ACL of this directory gives full "sma" access to all of the user's protection compartments. In the subtree rooted by this directory appear all normal directories and segments of the user. All are normally accessible to all protection compartments associated with the user.

The second directory, "Schroeder_b" in the figure, is the repository for especially sensitive information. It is accessible only to protection compartment "b" of the user. When dealing with such information, the user logs in under instance tag "b" and uses this directory as his working directory. Note that the default ACL term in this case will give access only to the user's "b" compartment. The user must be careful when logged in as "b" not to borrow untrusted procedures.

When logged in under the instance tag "a", however, borrowed procedures cannot get at sensitive segment, unless, of course, their ACL's specifically allow such access. Also, the ACL's of sensitive segments cannot be altered from protection compartment "a" because of the ACL term on directory "Schroeder_b".

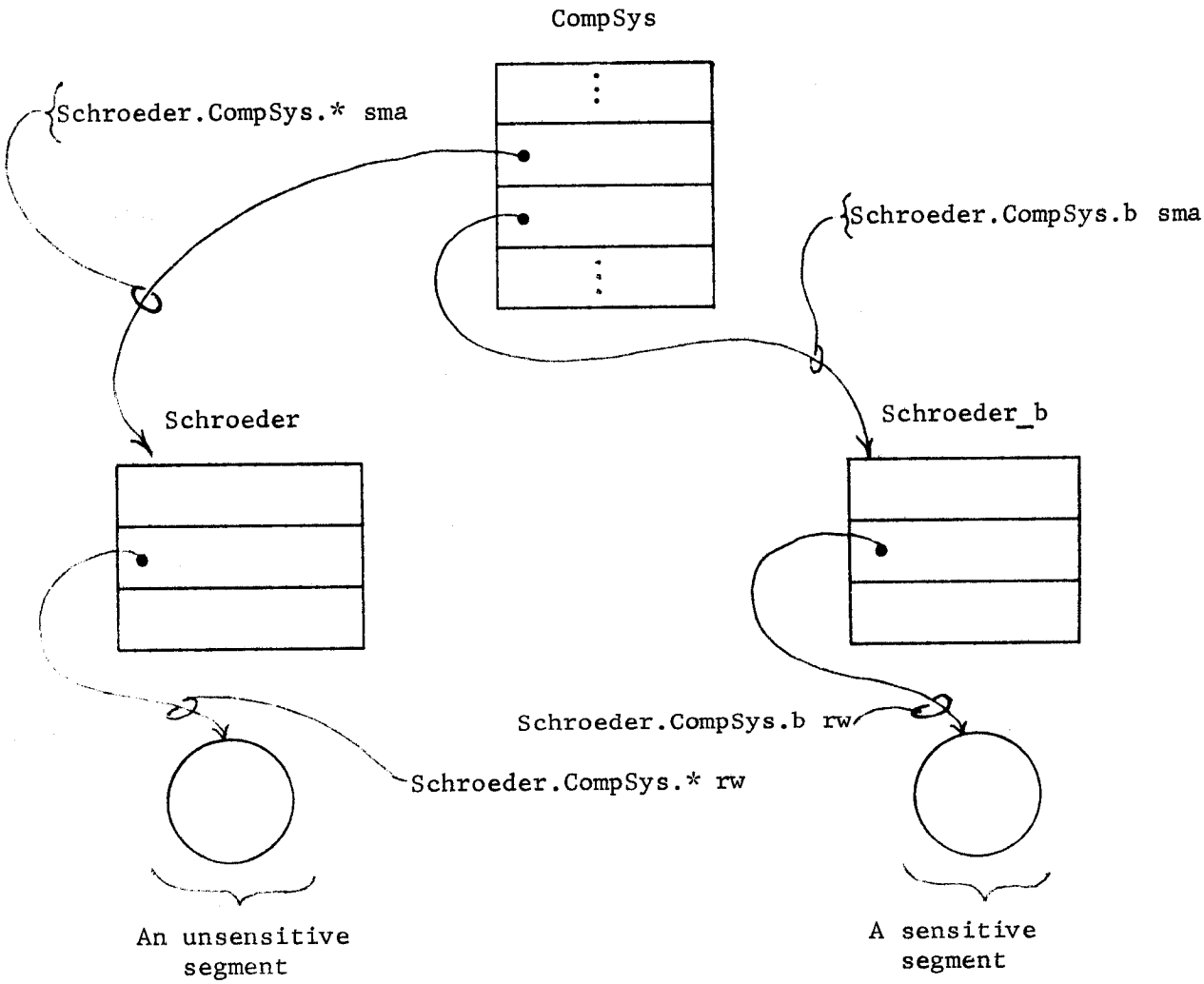

Figure 1