MITRE
M83-17

Proceedings of the Workshop on Implementing DoD Multilevel Security Policy on Capability-Based Operating Systems

Susan A. Rajunas

October 1982
Bedford, MA

DoD Computer Security Center
Project No. 8420
Contract No. AF19628-82-C-0001

DRAFT NRL SMMS TECHNICAL MEMORANDUM

23 March 1982

INTRODUCTION

A message system is secure if it can adequately protect the information it processes against unauthorized disclosure, unauthorized modification, and the unauthorized withholding of it (also referred to as denial of service). We say "adequately" because no practical system can achieve these goals without qualification; security is inherently a relative concept. A system is multilevel secure if it can protect information of different classification levels from users with different clearances and some users of the system are not cleared for some of the information processed by the system.

A model of the concept of security embodied by a system is needed for several reasons: so that users can understand how to operate it, so that implementors can build it correctly, and so that certifiers can determine whether its concept is consistent with the relevant policies and directives and whether the implementation matches the concept [Land81]. For the past several years, a single security model (called the Bell and LaPadula model [Bell75,Feie77]) has dominated attempts to build secure systems. A system that enforces this model can protect against the unauthorized disclosure of classified information.

Unfortunately, a system that strictly enforces this model is also impractical: in real systems, users frequently need to perform operations that, although they do not violate security, do violate the constraints of the model [Wils79]. For example, a user may wish to extract an unclassified paragraph from a confidential document and use it in an unclassified document. A system that strictly enforces the Bell and LaPadula model would have to prohibit this operation. Consequently, systems developed based on this model usually contain mechanisms for allowing some operations that are prohibited by it (e.g., the mechanisms for privileges and trusted processes in KSOS, SCOMP, and SIGMA [McCa79,Bonn81,Stot79]). The presence of these mechanisms makes it much more difficult to determine what the actual security policy enforced by the system is and tends to complicate the user interface.

Instead of adopting the Bell and LaPadula model as the top-level security model for a military message system, we develop a security model based more closely on the application. It is our

intent that this model be the common ground for the designers, implementors, and users of the system. Consequently it must be developed carefully, it must use terms that are accessible to users, and it must be precise without being overly formal. The next section documents our effort to meet these goals, and a final section discusses the implications of the model in particular areas of system operation. A key property of the model is that it includes the concept of a multilevel object: that is, an entity that has a classification itself, but may also contain other entities with different classification levels.

SECURITY MODEL

In this section we define some terms, use them to specify a model of how a user views the system's operation, and state assumptions and assertions based on the terms and the model that are intended to be sufficient to assure the security of the system. The security model includes the definitions, the model of operation, the assumptions, and the assertions. It is a revision of earlier work [Land80] based partly on the comments of Schaefer, Cooper, Miller, and Resnick [Coop81, Mill81].

Definitions

The definitions given below correspond in most cases to those in general use and are given here simply to establish an explicit basis for the model. We distinguish between objects, which are single-level, and containers, which are multilevel. We also introduce the concept of user roles, which correspond to particular job-related sets of privileges.

Classification: a designation attached to information that reflects the damage that could be caused by unauthorized disclosure of that information. A classification includes a sensitivity level (unclassified, confidential, secret or top secret) and a set of zero or more compartments (NATO, NUCLEAR, etc.). The set of classifications, together with the relation defining the allowed information flows between levels, forms a lattice [Denn76].

Clearance: the degree of trust associated with a person or device. For a person, this is established on the basis of background investigations and on the basis of the functions required of the individual (need-to-know). It is expressed in the same way as classifications are, as a sensitivity level and a (possibly null) compartment set. In a secure MMS, each user will have a clearance, and functions performed by the MMS for that user may check the user's clearance and the classifications of objects to be operated on. Devices such as disks, printers, tape drives, and the screen of a user's terminal may also

have clearances that define the highest classification of information that the device may store.

Effectives clearance: a clearance level for a device that is less than or equal to the clearance of the device. The effective clearance level may be set lower than the actual clearance in order to prevent more highly classified information from appearing on the device (even though the device is authorized for such information).

UserID: a character string used to denote a user of the system. To use the MMS, a person must present a userID to the system, and the system must authenticate that the user is the person corresponding to that userID. This procedure is called logging in. Since clearances are recorded on the basis of one per userID, each user should have a unique userID.

User: a person who is authorized to use the MMS.

Role: the function the user is performing, such as downgrader, releaser, etc. To act in a given role the user must be authorized for it. Some roles may only be assumed by one user at a time. With each role comes the ability to perform certain operations. A user may change roles without logging out and logging in again.

Object: an abstraction implemented by an MMS. An object is the smallest unit of information in the system to which a classification is explicitly attached. An object thus contains no other objects -- it is not multilevel. There are many kinds of objects; an example is the date-time group of a message.

Container: an abstraction implemented by an MMS. A container has a classification and may contain objects (each with its own classification) and/or other containers. Message files and messages are containers. Some fields of a message (such as the text field) may be containers as well. The distinction between an object and a container is based on its type, not its current contents: a container is still a container even if it is empty or if all the objects it contains are classified at the same level as the container itself.

Entity: either a container or an object.

Minimum clearance: an attribute of some containers. For some containers, it is important to require a minimum clearance, so that if a user does not have at least this clearance, he cannot view any of the entities within the container. The classification of such containers is marked with the attribute "Minimum Clearance (MC)". For example, a user with only a confidential clearance could be prohibited from viewing just the confidential paragraphs of a message classified top secret. On the other hand, a message file might contain both top secret and

confidential messages, and it would be acceptable to allow the user in question to view the confidential ones, even though the container (message file) as a whole is classified top secret.

UID: unique identifier. Every entity is named by a unique identifier.

Direct reference: a reference to an entity is direct if the entity's UID is used to name it.

Indirect reference: a reference to an entity is indirect if a sequence of entity names is used to name it.

Operation: a function that can be applied to an entity. It may simply return information from that entity (e.g., display a message) or it may alter the entity (forward a message), or both (compose a message).

Access List: a set of pairs (userID or role, operation) that is attached to an entity. The operations that may be specified for a particular entity depend on the type of that entity. For messages, operations might include create, destroy, update, reply, forward, etc. The existence of a particular pair on the access list implies that the user corresponding to the specified userID or role is authorized to invoke the specified operation on the entity to which the list is attached.

Message: a particular kind of container. A message may include a subject field, date-time group, addressee list, drafter identification, releaser identification, text, comments, and additional fields as well. Whether a given field is implemented as an object or a container may vary from one MMS family member to another.

User's View of MMS Operation

We present the following as a model of the use of a secure MMS. Terms defined above are printed in upper case.

People initiate use of the system by logging in. To log in, a person presents a USERID and the system performs authentication, using passwords, fingerprint recognition, or any appropriate technique. Following a successful authentication, the USER invokes OPERATIONS to perform the functions of the message system. The OPERATIONS a USER may invoke depend on his USERID and his current ROLE; by applying OPERATIONS, the USER may view or modify OBJECTS or CONTAINERS. The system enforces the security assertions listed below (i.e., it prevents the user from performing OPERATIONS that would contradict these assertions).

Security Assumptions

It will always be possible for a valid user of a message system to compromise the information to which he has legitimate access. To make the dependence of system security on the behavior of its users explicit, we list the following assumptions. These assumptions are really security assertions that can only be enforced by the users of the system.

A1. The System Security Officer (SSO) is assumed to assign clearances and roles properly to users.

A2. The user is assumed to enter the appropriate classification when composing, editing, or reclassifying information.

A3. Within a classification, the user is assumed to address messages and control access lists for entities he creates so that only users with a valid need-to-know can view the information.

A4. The user is assumed to control properly information extracted from containers marked with minimum clearance levels (i.e., to exercise discretion in moving that information to entities that may not have minimum clearance levels specified).

The essence of these assumptions is that when there is no other source of information about the classification of something, or the clearance of somebody, the user is assumed to provide information that is correct.

Security Assertions

The following statements are to be demonstrated to hold for a multilevel secure MMS:

Authorization 1. A user can only invoke operations on an entity if the user's userID or current role appears on the entity's access list with the operation that is being invoked.

Classification hierarchy 2. The classification of any container is always at least as high as the maximum of the classifications of the entities it contains.

Viewing 3. A user can only view (on some output medium) an entity with a classification less than or equal to the greatest lower bound of the user's clearance and the effective clearance of the output medium.
(This assertion applies to entities referenced either directly or indirectly.)

Viewing MC entities	4.	A user can only view an indirectly referenced entity within a container marked "Minimum Clearance" if the user's clearance is greater than or equal to the classification of that container.
Extracting information	5.	Information removed from an object inherits the classification of that object.
Clearance-setting	6.	The clearance recorded for a userID or device can only be set or changed by a user with the role of system security officer.
Effective clearances	7.	The effective clearance of a device is always less than or equal to its clearance.
Downgrading	8.	No classification marking can be downgraded except by a user with the role of downgrader.
Releasing	9.	No message can be released except by a user with the role of releaser. The userID of the releaser must be recorded in the "releaser" field of the message.

DISCUSSION

Although we view the model as defined above to complete, a discussion of its application in some specific cases should clarify its effects.

1. What prevents a user from copying a classified entity to an unclassified entity?

   The classification of the entity being copied accompanies the data. Moving classified data to an unclassified entity is a violation of assertion 8 (unless the user requesting the operation is the downgrader), since the classification of the data in question is effectively changed by the operation.

2. What about copying a part of an object into another object?

   A part of an object inherits the classification of the whole object (assertion 5). Thus moving part of an object into another object is disallowed by assertions 2 and 5 unless the objects have the same classification. Note that this constraint does not affect the user's ability to remove an unclassified paragraph (an object) from a confidential document (a container) and use it in an unclassified document (another container).

3. Does a user have a "login level" (i.e., a classification less than or equal to the user's clearance, determined at login, that defines his maximum effective clearance for this session)?

   Login level is not explicitly part of the model, but the effect of a login level is obtainable through the effective clearance of the user's terminal. The clearance of the terminal acts as an upper bound on the classification of information that can be displayed on it (assertion 3). The effective clearance of the terminal can be used to enforce a more stringent restriction if the user desires (assertion 7).

4. Processes do not appear in the model, but surely will be present in the implementation. How will their activities be constrained?

   Operations, rather than processes or programs, are in the model because they are closer to the user's view of the system. To the user, the system offers functions that may be invoked by typing strings of characters, pushing function keys, etc. Each function can be understood by the user as an operation. In the implementation, processes are constrained to prevent any function that would contradict the assertions.

5. Which entities in a particular message system will be containers and which will be objects?

   This decision is really part of the next lower level of detail from the stated model. Some likely choices are that messages and message files will be containers and that the date-time group will be an object. It is not necessary that all message systems in the family make the same choices. If two message systems that make different choices communicate, of course, some method of mapping between those things that are objects in one system and containers in the other must be defined.

6. How does a user refer to an object or a container?

   Each entity has a unique identifier (UID) that is system-generated. A user (or a program he invokes) can refer to an entity by its UID or by a symbolic name (a pathname; for example, "the third object in the container called "MSG1"). The former is called a direct reference and the latter an indirect reference.

7. What policy governs access to an object in a container? (Is the classification of the container or of the contents tested, and with what is it compared)?

The answer to this question depends on the type of access (the operation invoked) and whether the reference is direct or indirect. If the object is referenced directly for viewing, assertion 3 gives appropriate restriction. If the reference is indirect, there are two cases depending on whether or not the object is within a container marked MC. If it is, both assertions 3 and 4 have an affect, otherwise, only assertion 3 is relevant. If the operation has effects other than the viewing of objects, the other assertions may impose constraints. Assertion 1 always requires that the user (or his role) be on the access list for the entity-operation pair specified.

8. Is there anything in the system that is not (or is not part of) an entity?

   From the user's point of view, no. There may be structures in the implementation that the user is unaware of and that would be difficult to assign a legitimate classification to (such as internal operating system queues, perhaps). Anything the user can create, display, or modify, however, must be (or be part of) an entity.

9. What are the relationships among a user, an operation he invokes, and programs that the operation may invoke on his behalf? (For example, what privileges do the programs inherit, how is it determined whether a given invocation is allowed under the security policy?)

   A user has a clearance recorded in the system. When a user invokes an operation, his clearance (and his role, and the appropriate device clearances and effective clearances) control the operation.

10. There are no integrity levels or controls defined in the model. What prevents accidental/malicious modification of sensitive data?

    The reasons for omitting integrity levels have been discussed in a separate memo [Land82]. The alteration of clearance or classification data is covered in the given set of assertions. Any alteration of data must presumably be accomplished by a user's invoking an operation; his authorization to invoke that operation is required by assertion 1. Specific cases may be treated in additional assertions similar to assertion 9.

References

Bell75 Bell, D.E., and LaPadula, L.J., "Secure computer system: Unified exposition and multics interpretation," M74-244, The MITRE Corporation, Bedford, MA, July 1975.

Biba77 Biba, K.J., "Integrity considerations for secure computer systems," ESD-TR-76-372, ESD/AFSC, Hanscom AFB, Bedford, MA, April 1977 (MITRE MTR-3153, NTIS AD A039324).

Bonn81 Bonneau, C.H., "Secure communications processor kernel software, detailed specification, Part I, Rev. G," Honeywell Inc., Avionics Division, St. Petersburg, Fla., 1981.

Coop81 Cooper, D.M., "Design of a multilevel secure military message system using concepts from secure data base management systems," Final Report, System Development Corporation, TM-7662/000/00, September 1981.

Denn76 Denning, D.E., "A lattice model of secure information flow," Communications of the ACM 19, 5 (May 1976) 236-243.

Feie77 Feiertag, R.J., Levitt, K.N., and Robinson, L., "Proving multilevel security of a system design," in Proc. 6th ACM Symp. Operating Systems Principles, ACM SIGOPS Operating Systems Rev. 11, 5 (November 1977) 57-65.

Land80 Landwehr, C.E., "Assertions for verification of multilevel secure military message systems," ACM Software Engineering Notes, Vol. 5, No. 3, (July 1980) 46-47.

Land81 Landwehr, C.E., "Formal models for computer security," ACM Computing Surveys, Vol. 13, No. 3 (September 1981) 247-278.

Land82 Landwehr, C.E., "What security levels are for and why integrity levels are unnecessary," NRL Technical Memorandum, Code 7590, February 1982.

McCa79 McCauley, E.J., "KSOS: The design of a secure operating system," Proc. AFIPS National Computer Conference, AFIPS Press, Arlington, VA, Vol. 48, 335-342.

Mill81 Miller, J.S., and Resnick, R.G., "Military message systems: applying a security model," Proc. IEEE 1981 Symp. on Security and Privacy, IEEE Cat. No. 81CH1629-5, April 1981, 101-112.

Stot79 Stotz, R., Tugender, R., and Wilczynski, D., "SIGMA - an interactive message service for the military message experiment," Proc. 1979 National Computer Conference, June 1979, 839-846.

Wils79 Wilson, S.H., Killander, J.W., Thomas, N.M. III, Klitzkie, L.C., and Bunch, J.R., Jr., "Military message experiment quick look report," NRL Memorandum Rep. 3992, Naval Research Laboratory, Washington, D.C., April 1979, p. 10.

For those interested in the final version of the Military Message Systems report, the following reference is offered.

Landwehr, C.E., and Heitmeyer, C.L. Military Message Systems: Requirements and Security Model. NRL Memorandum Report 4925, Naval Research Laboratory, Washington, D.C. 20375.