2. R.J. Feiertag, K.N. Levitt, and L. Robinson. Proving Multilevel Security of a System Design. Proceedings of Sixth ACM Symposium on Operating Systems Principles, ACM, November, 1977.

3. R.J. Feiertag. A Technique for Proving Specifications are Multilevel Secure. CSL-109, SRI International, January, 1980.

4. B. Silverberg, L. Robinson, and K. Levitt. HDM Handbook, Volume II: The Languages and Tools of HDM. SRI International, June, 1979.

5. L. Robinson. The HDM Handbook, Volume I: The Foundations of HDM. SRI International, June, 1979.

6. R. Boyer and J Moore, A Computational Logic, Academic Press, 1979.

ASSERTIONS FOR VERIFICATION OF MULTI-LEVEL SECURE MILITARY MESSAGE SYSTEMS

Carl Landwehr
Naval Research Laboratory
Washington, D.C. 20375

As a developer and user of future multi-level secure (MLS) computer systems, the Navy has a considerable interest in verification technology. Other types of systems (e.g., highly reliable systems, systems with "guaranteed" performance characteristics) may also benefit from software verification techniques. Questions of importance in the development of such systems include:

- □ What properties of the system are to be verified?
- □ How much confidence can be generated through such a verification?
- □ How can the confidence established by a verification be carried over to the implementation and operation of the system?
- □ What influence do the verification goals have on the system design?
- □ What kinds of assertions can be proven about specifications using current automated aids for verification?

This note focuses on the first of these questions, as applied to a multi-level secure military message system. Currently, virtually all systems that are being developed for use in multi-level secure applications are based on extensions and reformulations of the original Bell and LaPadula model (1). In their efforts to build security kernel based operating systems (2,3), both Honeywell and Ford Aerospace and Communications Corporation (FACC) are using a reformulation of that model by Feiertag, et. al. (4,5). The SIGMA message system (6,7), developed by USC-ISI for the Military Message Experiment (MME (8)), employs a user interface consistent with the Bell and LaPadula model. "Consistent" here means that any actions that would violate the model require explicit confirmation by the user.

Use of models based on Bell and LaPadula's has had several effects on the development of these three systems. First, it freed the developers from the task of formulating new security models. It also means that these systems will be similar enough with respect to security constraints that users of one system should have a good idea of what the security-relevant behavior of the others will be. At a review of a preliminary SIGMA design, the sponsors, designers, and prospective users concluded that strict enforcement of the simple security condition and the *-property of the Bell and LaPadula model would result in an unacceptable user interface (8, p.10). Consequently, the user interface for SIGMA is based on use of "trusted processes" that are, with user confirmation, allowed to violate the axioms of the security model. In the MME, users found the required confirmations to trusted processes annoying, but not so much as to cause them to reject the system. As with SIGMA, the Honeywell and Ford efforts are likely to produce systems that will require applications to depend heavily on the use of trusted processes.

We note that, in the final version of their model, Bell and LaPadula did include trusted processes. What is not included in their exposition is a technique for establishing when a process may be trusted. The definition of an appropriate set of assertions for verifying that a process can be trusted has proven to be a knotty problem. Verification of the trusted processes to be used in one of the first operational applications of KSOS (the GUARD project (9)) has consumed far more resources than originally planned. Part of the difficulty in verifying trusted processes is that security kernel verification is concerned only with the security properties of the kernel, not with its correctness. Secure behavior on the part of trusted processes, however, often requires that the kernel perform correctly as well as securely. Thus, verification of the trusted process may require new assertions about kernel correctness be proven as well.

Instead of adopting the Bell and LaPadula model as the top-level security model for a military message system, we advocate an approach based more closely on the intended application. We propose to develop a simple model of the application and to state a number of assertions concerning it. These assertions would form the basis of the verification effort for the system. It is of course possible that this application-based model might in fact turn out to be a refinement of the Bell and LaPadula model (or some other security model), but we do not require such a relationship a priori. It seems clear to us that imposing such a requirement would not guarantee a cleaner design or implementation or a more useful set of assertions to be verified about the specification.

By defining a simple model of the application and deriving the security assertions from it, we plan to arrive at a set of assertions stated in terms understood by message systems designers, developers, and users. This approach is not without risk; some abstraction from the initial set of assertions may be required for verification purposes, and the security actually provided by enforcement of the chosen assertions will have to be examined carefully. Nevertheless, we believe this approach is more likely to lead to a system with a convenient user interface that provides both the required functionality and security than would a strictly top down approach based on refinements of the Bell and LaPadula model.

DEFINITIONS

First, we provide a set of definitions of terms to be used in the model. In most cases, these definitions correspond to those generally accepted. We have broadened the notion of classification to encompass both security and integrity, and we distinguish between objects, which are single level, and containers, which are (potentially) multi-level. Work on the definitions, the model, and the assertions given below is continuing. All are subject to change, but all have received at least a preliminary review by members of the Military Message System (MMS) project.

Classification: security level. A security level has two parts: one represents the damage that could be caused by unauthorized disclosure of the information (called the DISCLOSURE LEVEL) and the other represents the damage that could be caused by the unauthorized modification of the information (called the MODIFICATION LEVEL). Disclosure levels range from unclassified (low) to top secret (high) and include compartments. Modification levels are user (low), operator (medium) and system administrator (high). Modification compartments are also allowed; they may be used to restrict certain data-modifying operations, such as the release of formal messages, to specific groups of users. The disclosure and modification levels have often been referred to as the security and integrity levels, respectively, in other publications.

Clearance: the security level associated with 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). In an MLS 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 accessed. Clearances corresponding to users are to be entered into the MMS only by the Information System Security Officer (ISSO), who will have system administrator modification level.

Object: an abstraction implemented by an MMS. The important properties of an object are that it has a classification and it does not contain any other objects. (Objects are not multi-level. (A message field, with its classification, is an example of an object.)) No. \text object).

Container: an abstraction implemented by an MMS. A container has a classification and also may contain objects (each with its own classification). Folders, message files, and messages are containers.

Process: an active entity in an MMS. A process is the abstraction that corresponds to a program in execution on the system. An MMS process has an associated classification according to its current activity, and an associated user ID (and hence a clearance), corresponding to the user on whose behalf it is functioning. A process is an object, not a container.

User ID: a character string that uniquely designates a particular user within the MMS. To use the MMS, a person must present a user ID to the system, and the system must authenticate that the user is the person corresponding to that ID. This procedure is called logging in. Since clearances are recorded on the basis of one per user ID, each user should have a unique user ID.

Access List: a list of pairs (user ID, access mode) associated with objects or containers. Possible access modes include read, write, and execute. In order for a user to have access to an object or container in a specified mode, there must be a pair on the access list with his user ID and the requested access mode.

Message: A container that may include a subject field, date-time group, addressee list, drafter identification, releaser identification, text field, comments, etc.

Operations: Actions that may be invoked by users of the message system to view or modify objects and containers. For example, operations applicable to messages include the following: compose, output, edit, update, send, release, forward, distribute, coordinate, chop, readdress, reclassify, delete, undelete, destroy, etc.

MODEL

We present the following as a simplified model of the use of a multi-level secure MMS. Terms defined above are printed in upper case. People initiate use of the system via login. At login, a person presents a user ID and the system performs authentication. (The person may also wish to authenticate the system.) Following the successful login, the person is represented in the system by a PROCESS, which has a CLASSIFICATION (security and integrity level) derived from the login request and the person's CLEARANCE, as recorded in the system. Via this PROCESS, the person may request that various OPERATIONS be performed. These operations may in general read, write, or execute 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).

ASSUMPTIONS

The system makes the following assumptions about user behavior. These are really security assertions that the system is unable to enforce.

1. The Information System Security Officer (ISSO) is assumed to assign clearances properly to system users.

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

3. Within a classification (disclosure and modification level), the user is assumed to address messages and control access lists for objects and containers he creates so that only other users with a valid need to know can view the information.

SECURITY ASSERTIONS

These are the statements to be demonstrated to hold for a multi-level secure MMS:

* Assertions concerning disclosure of information

1. Users can only view objects with disclosure level less than or equal to the user's disclosure level. For objects within containers, either the container disclosure level or the object disclosure level will be used, depending on the type of container (policy issue).

2. Users can only view messages for which they are either creator, addressee, or on the access control list with an access mode of read.

* Assertions concerning modification of information

3. User clearances can be set or changed only by a person authenticated at login as the ISSO.

4. No classification markings are downgraded except by processes executing with system administrator modification level and the downgrader modification compartment.

5. A message can be released only by a user who has release authority and is listed as releaser in the message. (Similar assertions can be designed for other functions.)

6. The disclosure level of any container is always at least as high as the maximum disclosure level it contains.

CONCLUSION

This note is an initial attempt to list security assertions for a multi-level secure MMS. The purpose of these assertions is not to be completely abstract and general, but to state assertions in terms that will be understandable to message system designers, developers, and users. These assertions are also intended to allow discrimination between security-relevant and non-security-relevant operations in message systems.

REFERENCES

1. Bell, D., and LaPadula, L.J., Secure Computer Systems. MITRE MTR-2547, Vol I-III, Nov 1973-April 1974.

2. Bonneau, C.H. Secure Communications Processor Kernel Software. Detailed Specification, Part I, Rev. D. Honeywell Inc., Avionics Division, St. Petersburg, Florida, 1980.

3. Kernelized Secure Operating System (KSOS): Security Kernel Program Development Specification (Type B5), Version 4.2. Ford Aerospace and Communications Corporation (FACC), Western Development Laboratories, Palo Alto, California, 1979.

4. Feiertag, R.J., Levitt, K.N., and Robinson, L. Proving Multilevel Security of a System Design. Proc. Sixth ACM Symp. on Op. Sys. Prin., Nov. 1977, 57-65.

5. KSOS Verification Plan. FACC/SRI International, California, 1978.

6. Stotz, R., Tugender, R., and Wilczynski, D. SIGMA - An Interactive Service for the Military Message Experiment. AFIPS Conf. Proc. Vol. 40, 1979 NCC, 839-846.

7. Ames, S.R., Jr., and Oestreicher, D.R. Design of a Message Processing System for a Multilevel Secure Environment. AFIPS Conf. Proc. Vol. 39, 1978 NCC.

8. Wilson, S.H., Kallander, J.W., Thomas, N.M., III, Klitzkie, L.C., and Bunch, J.R., Jr. MME Quick Look Report. NRL Memorandum Report 3992, Naval Research Laboratory, 1979.

9. Woodward, J.P.L. Applications for Multilevel Secure Operating Systems. AFIPS Conf. Proc. Vol. 40, 1979 NCC, 319-328.