Prospect on Security Paradigms

Leonard J. LaPadula
The MITRE Corporation
202 Burlington Road
Bedford, Massachusetts 01730

Abstract

This paper examines some of the beginnings of paradigm shifts in computer security. It focuses on formal models of computer security from an historical perspective. Surprisingly perhaps, the historical perspective reveals dramatic shifts. These shifts take the form of extensions, in several directions, to early formal modeling. From them we believe we can learn much and discern new growth directions that have a solid basis in the current technology of trusted computer systems. We do not present a complete list of formal computer security models; for example, the formal model for the Multinet Gateway [1] is not discussed. It may be that the framework we use is applicable to them, or at least partly applicable. More likely, though, the framework would be enriched by a careful study of these other models. Viewing a number of models within the same framework can give us insights into a particular model's characteristics that may not be discernible when viewing a model in isolation.

1 What is this Paper About?

The workshop call for papers refered to shifts in computer security paradigms. The shifts, as we see them, are taking place along various "axes" of interest, among which are functionality, assurance, verification, and modeling.

Functionality. A shift from one kind of security policy to many kinds, from single-policy systems to multiple-policy systems

Assurance. A shift from bundled to unbundled functionality and assurance; a shift away from equating high assurance with formal verification toward developing assurance by a combination of various methods.

Verification. A shift from using only verification tools approved by the National Computer Security Center to vendor-developed tools and methods.

Modeling. A shift from state-machine representation of internal rules of operation toward new forms and wider scope of modeling for secure systems.

Two questions focus our discussion of several formal models.

- *Have we modeled the policies we should model?*
- *Have we extended the uses of modeling?*

Following the historical perspective based on these two questions, we offer comments on where we can go from here.

2 Have We Modeled The Policies We Should Model?

People make assumptions about the context of the Trusted Computer Systems Evaluation Criteria (TCSEC) [2] when they ask this; they frequently ask it rhetorically, assuming that the answer is "No." A case can be made for the claim that the TCSEC's proper business is to protect classified information, so that modeling non-disclosure, to the practical exclusion of all other policies, seems appropriate for it. Nevertheless, many people feel that a "richer" set of policies for trusted systems would better serve the needs of military and classified government activities. Information labeling in the Compartmented Mode Workstation [3] is one worked example of this. The security literature [4, 5] offers improved schemes using access control lists to provide stronger need-to-know controls than does current DAC.

In the larger context of the new Federal Criteria project, a number of people seem to be saying that

the answer is "No, we are not modeling the policies we should be modeling." We do not think this is unanimous by any means, but enough opinion favors new ways of doing business that we should take seriously the challenge to apply formal modeling technology to a wide range of policies. In the largest context of users and vendors, there can be no question that their answer is "No."

Yet, there are specific cases where people have modeled the policies that were appropriate to their interests. In doing this they have broken new ground in trusted systems and given us examples to study and, in some instances, to emulate. We know of four such efforts: in chronological order, secure military message system model [6], labeling policy for compartmented mode workstations [3], Clark-Wilson policy for integrity in commercial systems [7], and the ORGCON policy [8].

SECURE MILITARY MESSSAGE SYSTEM MODEL

This was the first major departure from the TCSEC's style for trusted systems. First, it is application oriented. It takes advantage of this fact to enrich its model semantics by defining the type of object called "message," in addition to distinguishing between "objects" and "containers" as information entities. Second, it expresses its security policy as a set of secure transforms* rather than a set of secure rules of operation. This avoids choosing a particular implementation, so the model can be applied to trusted message systems built on various platforms. Contrast this with the Multics-oriented Bell-LaPadula model, whose rules of operation represent kernel calls. Finally, the model deals with both internal and external interface requirements. This makes clear what the relationship is between the trusted message system and its users.

LABELING POLICY FOR COMPARTMENTED MODE WORKSTATIONS

This is an interesting non-access-control policy for dynamic assignment of labels to information, where the labels include dissemination control, handling markings, and code words—generally the components needed for appropriate external labeling of information products. It gets its requirements from an application environment-an intelligence analyst receiving, correlating, fusing, and disseminating information using a workstation. Its corresponding formal model is a direct descendant of the BLM-class of models. It demonstrates the feasibility of including an auxiliary policy to TCSEC MAC and DAC.

CLARK-WILSON POLICY FOR INTEGRITY IN COMMERCIAL SYSTEMS

The Clark-Wilson integrity model (CWIM) gives a new view of how a trusted computer system and people could cooperate to maintain an accurate computer-based model of a real-world enterprise. Clark-Wilson integrity requirements derive from objectives such as separation of duty and correspondence of computer data to real-world information. The model directly deals with external consistency issues. Although it does not explicitly articulate an external model (i.e., one that shows the relationship of the computer to its users), it does describe an internal model—one that shows how the computer system meets the external requirements allocated to it—that can support a class of external models. This class is exemplified by an accounting enterprise. The CWIM, like the SMMSM, takes advantage of its application orientation by distinguishing integrity-controlled objects (constrained data items - CDIs) from ordinary objects (unconstrained data items—UDIs).

We developed a rule-set model of Clark-Wilson integrity policy for a UNIX System V/MLS system [9]. This model carries the elaboration of requirements into greater detail, giving implementation-dependent rules of operation and functional specification. Like the Bell-LaPadula model, though, it does not explicitly give an external model. The SMMSM approach, however, appears suited to this need. Using the SMMSM modeling paradigm, we might be able to articulate an explicit external model for Clark-Wilson integrity.

ORGCON POLICY

This ORCON-like policy† was defined, prototyped, and modeled in the GFAC research project [8]. The model for this policy uses a policy rule set and a separate finite state machine representation of the target system. The finite state machine representation is a BLM-type construct while the rule set provides a separate definition of

*A *transform* is a system function from $UI \times I \times S$ into S, UI is a set of userIDs, I is the set of well-formed system requests, and S is the set of possible system states. A transform is secure when it is access secure, copy secure, CCR (container clearance required) secure, translation secure, set secure, downgrade secure, and release secure.

†The ORCON policy, defined in Director of Central Intelligence Directive 1/7, describes controlled dissemination of information belonging to an organization.

the policies enforced by the state machine. This model has some interesting aspects for this workshop. First, it deals with a policy that is neither traditional MAC nor traditional DAC. The computer's policy directly supports a real-world document handling policy for limited distribution. Second, it describes in detail how a UNIX system can support this policy, down to the system call level. Third, it explicitly models trusted processes. And, finally, issues that arose in defining the model influenced the definition of the policy for computer implementation. The final point is an example of one way to extend the uses of modeling.

3 Have We Extended The Uses Of Modeling?

We believe we have extended the uses of modeling somewhat-the instances we cited above are examples. We will characterize the extensions we see, but, to start, we need a basis for identifying extensions. A generally familiar basis we can use is the Bell-LaPadula (Multics) model. We need also a framework for characterizing extensions. For this we can use the Williams-LaPadula taxonomy that we introduced last year at the Computer Security Foundations Workshop IV [10]. The taxonomy identifies stages in the development of requirements for a trusted system, from objectives to detailed functional specification. The short form of the taxonomy is displayed in Table 1.

In Table 2 we have "located" the BLM and the four examples cited above, in chronological order from left to right. The way we have located each model is by markings in the stages that we think each model deals with. For each stage, the number of X's are a rough indication of the attention paid by the model on the particular stage; the greater the number of X's, the greater the focus of the model on the particular stage.

Table 2 shows that we have extended the uses of formal modeling compared to the early days of trusted systems. Here are observations about what the tabulation tells us.

- The SMMSM and CWIM have much in common with each other and little in common with the BLM. One reason for this is that they both, but especially the SMMSM, avoid implementation dependencies. And they place heavy emphasis on relating the computer policy to real-world activities. They are application-oriented while the BLM is operating system-oriented.

- The depiction of the CMWM in table 2 looks like what it was intended to be—a demonstration that the floating label policy could be added to a Bell-LaPadula-type model.

- The ORGCON policy modeling spans the most stages, intentionally as part of the research objectives, but completely skipped stage 3, internal requirements. This is arguably a significant deficiency, not of the effort, which was constrained by time and budget, but of the resulting model if it were to be used for a real implementation and assurance assessment. The research effort defined the ORGCON policy, stages 1 and 2 description, and prototyped it, for which the formal modeling effort (stages 4 and 5) provided design guidance for a particular implementation. If we look at this in light of Table 2, we can see that an implementation-independent formalization of ORGCON would require elaboration of the requirements at stage 3 while stages 4 and 5 could be eliminated.

One striking facet we see here is that modeling is not just for assurance in the TCSEC sense. We think there is a tendency closely to couple formal models with assurance of that kind for trusted systems. In fact, a formal model may have nothing to do with TCSEC-assurance but much to do with utility, functionality, correctness, or other aspects of an application or enterprise. The implementation-independent models in table 2, having no elaboration at levels 4 and 5, are of this type. They naturally do not deal with TCSEC-assurance that the computer will do what its FTLS says it will do, although, of course, they can lead to this through successive elaboration of requirements at the lower levels.

So, we think we can make a case for the usefulness of modeling that does not give an implementation and is not oriented toward TCSEC-assurance. Then, looking at table 2 in another way, one can see the possibility of extending modeling even further toward implementation than has been traditional. The ORGCON model shows this—it includes new UNIX system calls, in the form of rules of operation of a finite state machine, to support control of file copying. In this approach, each system call corresponds exactly with one rule of operation and each rule of operation includes a wealth of detail that can support traceability from requirements to implementation, bring formal methods closer to complete functional design and coding.

Table 1: Requirements taxonomy

Stage of Elaboration	Description
1 - Trust Objectives	A *trust objective* specifies what is to be achieved by an information-processing enterprise, an important component of which is a computing system.
2 - External-Interface Requirements Model	An *external-interface requirements model* describes the behaviors of the computing system, its users, and other entities in the system's environment in such a way as to allocate responsibilities for achieving the identified trust objectives, thereby showing how the system supports the identified trust objectives.
3 - Internal Requirements Model	An *internal requirements model* describes, in an abstract manner, how the system responsibilities given in the external model are met with the system.
4 - Rules of Operation	*Rules of operation* explain how the internal requirements developed in the model are enforced.
5 - Functional Design	A *functional design*, like the rules of operation, specifies the behavior of system components and controlled entities, but is a complete functional description.

4 Where Can We Go From Here?

We see two ways to extend the taxonomy we described earlier.

- Additional Stages of Elaboration
- Application to Informal Description and Verification

These extensions suggested themselves to me as a result of two activities: documenting [11] the results of the Integrity Working Group Exercise [10] and participation in the "Fundamental Questions on Formal Methods" panel at the Computer Security Foundations Workshop V [12].

The additional stages of elaboration are

ENTERPRISE DESCRIPTION
: A model of an enterprise, describing the activities, responsibilities, and methods by which the goals of the enterprise are carried out, such as a description of roles, responsibilities, interactions with customers, and data flow at the SmallTown Branch of BigCity Bank. The enterprise description gives the needed background for stating the trust objectives for the enterprise.

HARDWARE & SOFTWARE
: Detailed models of hardware and software features and mechanisms to support the security features of the functional design.

In table 3 1 have used the expanded taxonomy to show the traditional emphasis in computer security modeling, what other stages have been more recently modeled, and what are areas most in need of development. Dobson's work [13] is an example of recent work in enterprise modeling

Here the markings for what has already been done is used as follows:

None: the stage has gotten only broad-brush treatment at best.

X: the stage has been explored but much remains to be done.

Table 2: A view of five models

Requirements Elaboration Stage	**BLM**	**SMMSM**	**CWIM**	**CMWM** (floating labels)	**ORGCON**
Objective	XX	XXX	XXX	XX	XXX
External-Interface Requirements		XXX	XX		XXX
Internal Requirements	XX	XX	XXX	XX	
Rules of Operation	XXX			XXX	XXX
Functional Design					XX

XX: the stage has been thoroughly developed from at least one point of view or in at least one technological context.

while the boxes for areas for development are checked as follows:

None: the stage has been done within some technological framework.

$\surd$: the stage has been explored but much remains to be done.

$\surd\surd$: the stage has not yet been explored in an significant way.

Another way to expand the scope of the taxonomy is to apply it to description and verification, in addition to modeling. Verification would demonstrate correct mapping of an (n-1)-stage elaboration to its successor n-stage elaboration. As an example of using the expanded scope, table 4 depicts the requirements of the TCSEC.

This expanded taxonomy can provide a common basis for various views of and proposals for paradigm shifts. As an example, several of the positions discussed in the "Fundamental Questions on Formal Methods" panel at the Computer Security Foundations Workshop V [12] are commented in terms of the taxonomy.

POSITION STATEMENT: Formal methods of modeling and verification have failed to provide assurance—we need new ways to provide assurance.

COMMENT: This position suggests that we may need to develop better descriptions and add more columns to the taxonomy table.

POSITION STATEMENT: Formal methods would be more cost-effective if applied early in the definition of requirements.

COMMENT: This position suggests that the best pay-off is in the upper rows of the taxonomy table.

POSITION STATEMENT: Non-disclosure is the wrong policy to apply formal methods to; we need to start to deal with other policies that are relevant to the real world outside of DOD.

COMMENT: This position suggests that a version of the taxonomy table may be needed for each general policy of interest, with each version placing varying emphasis on the several stages and columns of the taxonomy.

POSITION STATEMENT: Formal methods need to encompass applications and application software; it is here that the semantics of a trust policy for an enterprise are known.

COMMENT: This position suggests that emphasis should be placed in the upper rows of the taxonomy table (enterprise and objectives), while the orientation of the lower rows (internal objectives/interface, functional design, hardware/software) should be , "shifted" away from the reference monitor orientation toward design and assurance of application software and the support needed in the underlying hardware and operating system.

5 Summary

The history of computer security over the past twenty years shows that just in the area of modeling, both formal and informal, paradigm shifting has been occurring slowly but significantly. Using the Bell-LaPadula model of 1973 as a reference, we see that later models have dealt with other and more extensive areas of computer security requirements def-

Table 3: A perspective on computer security modeling

Stage of Elaboration	Traditional Emphasis	More Recent Modeling	Areas for Development
0 - Enterprise Description		X	√
1 - Trust Objectives		X	√
External-Interface 2 - Requirements Model		X	√
Internal 3 - Requirements Model	XX	XX	
4 - Rules of Operation	XX	XX	
5 - Functional Design		X	√
6 - Hardware/Software Specification			√√

inition. The historical perspective suggests a taxonomy of requirements elaboration that spans the entire range from enterprise to hardware and software and has at least the dimensions of description, modeling, and verification. Within this territory are many opportunities for research and development, in various directions as suited to different objectives.

References

[1] Freeman, James W., R. B. Neely, and G. W. Dinolt, "An Internet System Security Policy and Formal Model", 11th National Computer Security Conference Proceedings, October 1988, pp.1019.

[2] National Computer Security Center (NCSC), Department of Defense Trusted Computer System Evaluation Criteria, DOD Standard 5200.28-STD, December 1985.

[3] Millen, Jonathan K. and D. J. Bodeau, August, 1990, A Dual-Label Model for the Compartmented Mode Workstation, MITRE Paper M90-51, The MITRE Corporation, Bedford, Massachusetts.

[4] Graubart, R. G., October 1989, "On the Need for a Third Form of Access Control," Proceedings of the 12th National Computer Security Conference, pp. 296-304.

[5] McCollum, C. J., J. R. Messing, and L. Notargiacomo, May 1990, "Beyond the Pale of MAC and DAC - Defining New Forms of Access Control," Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy.

[6] Landwehr, Carl E., C. L. Heitmeyer, and J. McLean, August 1984, "A Security Model for Military Message Systems," ACM Transactions on Computer Systems, Volume 2, Number 3, pages 1982.

[7] Clark, David D. and D. R. Wilson, April 1987, "A Comparison of Commercial and Military Computer Security Policies," Proceedings of the 1987 IEEE Symposium on Security and Privacy.

[8] Abrams, M. D., L. J. LaPadula, I. Olson, April 1992, Generalized Frameworkfor Access Control: A Formal Rule Set for the ORGCON Policy, MITRE M-Paper, The MITRE Corporation, Bedford, Massachusetts.

[9] LaPadula, Leonard J., August 1991, A Rule-Base Approach to Formal Modeling of a Trusted Computer System, MITRE Paper M91-021, The MITRE Corporation, Bedford, Massachusetts.

[10] LaPadula, Leonard J. and J. G. Williams, "Toward a Universal Integrity Model," Working Group Exercise Opening Statement, Proceedings of the Computer Security Foundations Workshop IV, June, 1991, pp. 216-218.

[11] Abrams, Marshal D., E. G. Amoroso, L. J. LaPadula, T. F. Lunt, and J. G. Williams, "Report of an Integrity Research Study Group", paper

Table 4: TCSEC requirements from the viewpoint of the taxonomy

Stage of Elaboration	**Informal Description**	**Modeling**	**Verification**
0 - Enterprise Description			
1 - Trust Objectives	**TCSEC**		
External-Interface 2 - Requirements Model			
Internal 3 - Requirements Model	**TCSEC**	**TCSEC**	
4 - Rules of Operation	**TCSEC**	**TCSEC**	**TCSEC (B2+)**
5 - Functional Design	**TCSEC (B2+)**		**TCSEC (A1)**
6 - Hardware/Software Specification	**TCSEC**		

submitted to Computers and Security September, 1992.

[12] Meadows, Catherine A., "Fundemental Questions about Formal Methods: Introduction to Panel Discussion", Proceedings of the Computer Security Foundations Workshop V, June, 1992, page 52.

[13] Dobson, J. E. and J. A. McDertuid, "Security Models and Enterprise Models", in Database Security: Status and Prospects II, ed. C. E. Landwehr, pp. 1-39, Elsevier Science Publishers, Amsterdam, 1989.