Toward a Universal Integrity Model

Leonard J. LaPadula
James G. Williams

The MITRE Corporation
Burlington Road
Bedford, Massachusetts 01730

Abstract

This position paper establishes the framework for a working group exercise to address the interesting problem "What is an acceptable definition of Clark-Wilson integral—having Clark-Wilson integrity—in terms of external-interface requirements for a trusted system?". We want a characterization of the Clark-Wilson policy, used as a paradigm for its class of integrity policy, useful for demonstrating that a system incorporating Clark-Wilson integrity in accordance with a model of this trust policy satisfies "external" notions of Clark-Wilson integral and, thus, meets the corresponding integrity control objectives.

Introduction

Table 1 presents a taxonomy of stages in the elaboration of the security requirements for a trusted system. This taxonomy was developed to serve as a framework within which to locate formal models of trusted systems. It serves here as both background and frame of reference for our working group exercise.

Trust Objective. A trust objective specifies what is to be achieved by proper design and use of a computing system. It constrains the relationship between the system and its environment.

External-Interface Requirement. An external-interface requirement applies a trust objective to the system's external-interface, that is, the system's interface to its environment. It constrains the interactions of the system with agents in its environment, such as users or network connections.

Internal Requirement. Internal requirements constrain relationships among system components and, in the case of a TCB-oriented design, among controlled entities.

Rules of Operation. Rules of operation explain how internal requirements are enforced by specifying the access checks and related behaviors that guarantee satisfaction of the internal requirements.

Table 1. Stages of Elaboration of Security Requirements

<table>
<thead>
<tr>
<th>Stage of Elaboration and Characterization</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Trust Objectives</td>
<td>• TCSEC mandatory security objective [11]</td>
</tr>
<tr>
<td></td>
<td>• Clark-Wilson integrity objectives [3]</td>
</tr>
<tr>
<td>2 External-Interface Requirements</td>
<td>• Noninterference [5]</td>
</tr>
<tr>
<td></td>
<td>• Millen Universal Multilevel Model [9]</td>
</tr>
<tr>
<td>3 Internal Requirements</td>
<td>• Bell-LaPadula model *-property [2]</td>
</tr>
<tr>
<td></td>
<td>• CMW model accessible expression [10]</td>
</tr>
<tr>
<td>4 Rules of Operation</td>
<td>• Open-file access checks [2]</td>
</tr>
<tr>
<td></td>
<td>• Read-file label float [10]</td>
</tr>
<tr>
<td>5 Functional Design</td>
<td>• Open-file functional description; for example, UNIX open system call [1]</td>
</tr>
</tbody>
</table>

Functional Design. Like the rules of operation, the functional design specifies behavior of system components and controlled entities, but is a complete functional description.
Discussion

The traditional Bell-LaPadula Model (BLM) [2] addresses the third and fourth levels of elaboration. The simple security property and the no-write-down property (the \( * \)-property) are two axioms of the model that express the mandatory access control policy as constraints on a trusted system's operation. These properties are defined as internal requirements at the third stage of elaboration in our taxonomy. The rules of operation of the BLM elaborate the behavior of the trusted system at the next lower level of abstraction, level 4 in the stages of elaboration shown in the table. The proof of security in the BLM consists in showing that the rules of operation satisfy the axioms or, in the terms of table 1, that the rules of operation developed in stage 4 are a correct elaboration of the internal requirements developed in stage 3.

The more recent Compartmented Mode Workstation (CMW) Model (CMWM) [10] essentially addresses the same levels of elaboration as the BLM. Corresponding to simple security property and \( * \)-property are the static read property and the static write property, two axioms of the model given as state invariants. The CMWM has more formally stated axioms than the BLM, giving greater elaboration of the internal requirements, stage 3 in the taxonomy. We draw attention to the fact that neither the BLM nor the CMWM incorporates external-interface requirements (stage 2 in the taxonomy), nor is there any provision within the frameworks of these models to relate the internal requirements to any external-interface requirements. This characteristic has been noted, as a drawback to this type of model, by McLean [8]. McLean has pointed out that a system may be "secure" with respect to the basic security theorem of the BLM while being manifestly insecure according to commonly accepted notions of security. Stated in terms of our taxonomy, McLean's observation is that a BLM-based system may be "internally" secure with respect to internal requirements (level 3) while being "externally" insecure — failing to satisfy trust objectives (level 1) and their elaboration in the form of external-interface requirements (level 2).

The modeling approach under development by LaPadula [7] addresses the noted deficiency by suggesting that a formal model be shown to satisfy some accepted definition of security. In the referenced paper, a multilevel model at the level of abstraction of the external-interface requirements, like the model developed by Millen [9], is proposed as the definition of multilevel secure. In the terms of our taxonomy, LaPadula has suggested that a model developed for the third and fourth stages of elaboration should be shown to satisfy the external-interface requirements elaborated at the second stage. This need not be restricted to a single trust policy. Generally, one can attempt to show that a model incorporating several trust objectives, multilevel security being only one possibility, satisfies formally defined external-interface requirements for each of its trust objectives.

One of the more interesting possibilities for trust requirements other than multilevel security is the Clark-Wilson Integrity policy [3]. Clark-Wilson Integrity (CWI) provides for both external and internal consistency of data. Measures for external consistency, such as their Integrity Verification Procedures (IVPs), ensure that the data stored in the computer system correctly models the state of the real-world situation it relates to. Measures for internal consistency ensure that data in a valid state is modified in such a way that the resulting state of the data is again valid. Some of the CWI rules deal with external consistency, some with internal consistency, and some with the relationship between these two aspects of the data that is integrity-controlled.

Working Group Exercise

Now, the interesting problem arises of what is an acceptable definition of "Clark-Wilson integral" — having Clark-Wilson integrity — at the second stage of elaboration. What is wanted is a definition for Clark-Wilson integral, expressed in a formal language, at the second stage of elaboration. Such a definition would make it possible to demonstrate that a system incorporating Clark-Wilson integrity in accordance with a formal model of this trust policy satisfies "external" notions of Clark-Wilson integral and, thus, meets the corresponding control objectives of this policy. This challenging problem is the focus of the Computer Security Foundations Workshop IV working group exercise Toward a Universal Integrity Model.

Annotated List of References

Its basic security theorem states that a system is secure (according to the intra-model definition of security) if at each state transition the access pairs (Subject, Object) added to the current access set are such that the security level of the subject dominates the security level of the object, and the security levels of the subjects and objects in the system do not change (or, in another version of the theorem, the security levels of subjects increase or remain the same while the security levels of objects decrease or remain the same).


The external consistency requirements are of particular relevance to this working group, as are their IVPs. An unformalized aspect of their IVPs appears to be that the verification of a CDI by an IVP involves a comparison of the CDI to an independently gathered view of the modeled real-world situation; this is one reason why separation of duty is necessary.


The presented notion of noninterference, as applied to multilevel security says that system outputs at a given classification level must depend only on inputs at or below that level. Noninterference is defined only for deterministic state machines; for these, it apparently rules out all storage channels (and timing channels as well, if the passage of time is considered to be among the system's inputs). An analogue of the basic security theorem, known as the "unwinding theorem," is presented in [6].


This multilevel model is a state-machine expanded, first to include a notion of information flow, and then to include object-labels. It allows a form of the basic security theorem and is general enough to cover floating-label policies as well as more traditional policies satisfying tranquility. Its basic security theorem, however, does not rule out label-based covert channels in the case of floating label policies.


The CMW Model is a formal model of the policy required for a compartmented mode workstation as defined in [4]. Besides the mandatory and discretionary access control policies typically defined in a security policy model, the CMW Model also addresses information labels with markings, an internal floating label mechanism, and least privilege. Its axioms include subject- and object-tranquility, stated as state-transition constraints. This model, in our opinion, is the modern archetypal example of a trusted system formal model employing the state-transition machine approach. Its "maccessible" expression implements the mandatory access control policy requirements for the CMW.
