Functional Analysis of Basic Security Models

Nikoleta H. Hristova[1], Vencislav G. Trifonov[2], Ivailo I. Atanasov[3]

Abstract – **This paper investigates some of the basic security models. These models are examined in terms of their security properties and divided into three groups with respect to their definitions of security. It has been made a description of the areas where they could be used and some proposals of model's applicability in practise.**

Keywords – **security models, access control, security classes, information flow**

I. Introduction

Information contained in an automated system must be protected from three kinds of threats: (1) the *unauthorized disclosure* of information, (2) the *unauthorized modiftcation* of information, and (3) the *unauthortzed withholding* of information (usually called *denial of service*). To achieve protection, that fully covers these kind of threats it is very important clearly to understand and implement the system's security requirements. The purpose of a security model is to express those requirements precisely.

The term security model is used to describe any formal statements of a system's confidentiality, availability, or integrity requirements. A security model does not deal with all variables and functions of the system, but it is concerned only with security relevant ones.

In this article we try to make a brief explanation of some of the well-known models of security and to compare them with respect to motivation, approach, view of security and use in practice.

II. Basic Security Models

A *finite-state machine model* describes a system as an abstract mathematical state machine; in such a model, *state variables* represent the state of the machine, and *transition functions* or *rules of operation* describe how the variables change.

The *lattice model* [2] uses a lattice as a building base.A lattice is a finite set together with a partial ordering on its elements such that for every pair of elements there is a least upper bound and a greatest lower bound.

The *Access Matrix model* [2] is a state machine model which represents the security state of the system as a large rectangular array containing one row per subject and one column for subject and objects. Each entry specifies the modes of access the object has to a subject or to other subject. A variant of the access model is the *information flow model*, which – rather than checking a subject's access to an object – attempts to control the transfer of information from one object into another object, constrained according to the two objects' security attributes.

The *Bell and LaPadula model* [2], may be summarized in two axioms: (1) No user may read information classified above his clearance level ("No read up");(2) No user may lower the classification of information ("No write down"). The full statement of the model includes several more axioms and is quite complex.

The *high-water mark model* [1] takes its name from the "History functions" which record the highest authority assigned to the object and the union of all categories assigned to the object since its creation. The model works with four types of objects, and each object is described by an ordered triple of properties, called Authority (A), Category (C), and Franchise (F). The model also defines an ordering on these triplets that corresponds to the lattice model.

UCLA Data Secure Unix (DSU) model [1] is a finite state machine model, with the state defined by the following four components: (a) process objects; (b) protection-data objects, with values being sets of capabilities; (c) general objects (comprising both pages and devices); and (d) a current-process-name object, whose value is the name of the currently running process.

Take-grant models [1] use graphs to model access control. Although couched in the terms of graph theory, these models are fundamentally access matrix models. The protection state of a system is described by a directed graph that represents the same information found in an access matrix. Nodes in the graph are of two types, one corresponding to subjects and the other to objects. It implies a new property called "Take", which means that grants may be taken from another subject to receive rights for a given object.

Filter models imply security policies as a filter of input functions on system inputs.

Strong dependency [1] is a model build on an approach which is based on the notion, fundamental to information theory, that information is the transmission of *variety* from a sender to a receiver.

A *constraint* specifies a sequence of states that cannot occur.It may be a part of other model, or may be used a base to a separate group of models.

[1] Nikoleta H. Hristova is with the Faculty of Communications and Communications Technologies,Technical University, Kliment Ohridski 8, 1000 Sofia, Bulgaria, E-mail: nhh@suntu.vmei.acad.bg

[2] Vencislav G. Trifinov is with the Faculty of Communications and Communications Technologies,Technical University, Kliment Ohridski 8, 1000 Sofia, Bulgaria, E-mail: vgt@vmei.acad.bg

[2] Ivailo I. Atanasov is with the Faculty of Communications and Communications Technologies,Technical University, Kliment Ohridski 8, 1000 Sofia, Bulgaria, E-mail: iia@vmei.acad.bg

III. Comparison of Models

It is very hard to find rates suitable for assessing the security of a particular security model, that are either precise and general enough to imply for different types of models, so that a quantitative analysis could be made. That is a result of that all of the models define security as absolute: an operation is either secure or not secure. Therefore we can make only a qualitative comparison of models.

Tables 1 shows a comparison of models, described above with respect of motivation, approach, view if security and use.

Table 1. Comparison of properties of models in terms of motivation

Properties	Models									
	A. M	UCLA	T-G	HWM	B+L1	B+L2	Flow	Filt	S. D	Cons
Motiwation										
Developed primarily to represent existing systems	×	$\times^b$		$\times^b$						
Developed to guide construction of future systems			×		×	×	×	×	×	×
View of security										
Models access to objects without regard to contents	×	×	×	×	×					
Models flow of information among objects						×	×			
Models inferences that can be made about project data								×	×	×
Approach										
Model focuses on system structures (files, processes)	×	×	×	×	×	×				×
Model focuses on language structures (variables, statements)							×	×	×	
Model focuses on operations on capabilities		×	×							
Model separates protection mechanism and security policy	×	×	×					×		
Systems based on or represented by this model have been implemented	×	×		×		×				

With respect to their definitions of security, the models can be divided roughly into three groups:

- those that are concerned only with controlling direct access to particular objects (access matrix model, UCLA DSU model, take-grant model);
- those that are concerned with information flows among objects assigned to security classes (information-flow model, revised Bell and LaPadula model);
- and those that are concerned with an observer's ability to deduce any information at all about particular variables (filters, strong dependency, constraints). (The high-water-mark model falls between the first and second groups, since it is concerned with the security levels of the objects a process touches over time, but it only controls direct accesses to objects.)

Models in the first category may be used in systems that require high degree if security, with addition of security policy for the concrete implementation. Those in the second group are probably the closest in structure to the requirements for telecommunication applications, but applications often require more flexibility than these models permit. The models in the third category are the least tested and would probably be the most difficult to use. Security properties of the UCLA DSU model were proved to hold for substantial portions of that system, but only the Bell and LaPadula model has been applied in more than one formally specified system. The properties specified by the high-water-mark, access matrix, and take grant models could probably be stated in a form suitable for automated verification techniques. The properties required by the constraint, strong dependency, and filter models could be expressed similarly, but actually developing a system specification in the terms required by those models requires unduly amount of work. Most of the secure system developments using the (revised) Bell and LaPadula model have been based on the concept of a security kernel, and there have been problems in extending its use beyond the operating system to application systems. The lattice model will probably fit many of the requirements for security and privacy in the private sector. An alternative to adding special models for trusted processes on top of the Bell and LaPadula model for the operating system is to develop integrated models tailored to particular applications. A security model designed for a particular application could be used as a basis for the development of an application-specific security kernel. A key problem in this approach is to ensure that the model incorporates the desired notion of security while permitting the operations required in the application. Further off, if capability-based systems are successfully developed, models more appropriate to their structures may be used. The take and grant model is a possible candidate in this area, though it would require tailoring for specific applications.

IV. Conclusions

Some of the described models are good candidates for secure system building, but even if such a system truly simulates any of the models it is impossible to say, that it is unconditionally secure, because each model defines its own concept of security, and a that system will be secure only in the sense defined by that model.

In conclusion it is clear that none of the basic models of security is suitable for the new generation complex systems, where many security requirements need to be taken into consideration. These models may be used only as a reference point when designing the security of the system.

References

[1] Shimeall T., Williams P., “Models of Information Security Trend Analysis ” – CERT® Analysis Center, Software Engineering Institute Carnegie Mellon University

[2] Amoroso E., “Fundamentals of Computer security Technology”. Prentice Hall, 1994