A Complete Protection Model

Luke C. Dion

Ford Aerospace & Communications Corporation
Palo Alto, California 94303

ABSTRACT

The Bell & LaPadula computer security model is, to date, the most successful data flow restriction security model. Specifically, it prevents unauthorized downgrading of data. K.J. Biba strengthened the Bell & LaPadula security model by adding security's mathematical dual (integrity) to prevent unauthorized upgrading of data. The security and integrity constraints are overly restrictive in some cases and not restrictive enough in others. This paper describes another extension to the Bell & LaPadula security model (with integrity) to better accommodate secure systems designers and implementers.

INTRODUCTION

There are several ways to enforce computer security in a multi-user system. The methods used can be divided into three classes: isolation, stratification, and limited flow restrictions. The term isolation refers to the separation of users into groups. Users are only allowed to communicate and affect state information of other users in their group. Furthermore, groups will generally be restricted to operating within a single security classification. Note that isolation requires that users cannot communicate with users in other groups even when the groups are at the same security classification. The term stratification refers to the separation of users into security classifications. Users are only allowed to communicate and affect state information of other users at their security classification. Stratification is a less restricted version of isolation. The term flow restrictions refers to the establishment of rules by which communication between security classifications is governed. A system with isolation or stratification is less useful than one with flow restrictions since duplication of mechanism and tools is required.

As the preceding paragraph suggests, a "safe" or "correct" protection (security) model does not necessarily imply that it is useful. Isolation prevents all types of security violations quite admirably. However, it is not desirable since an extreme amount of redundancy (in software) is required to operate the system since no inter-group sharing is allowed. Similarly, stratification is generally undesirable since inter-classification sharing is not allowed. The most desirable method is the application of flow restrictions. The most successful flow restriction model yet developed is an extended Bell & LaPadula model[1].

The goal of this paper is to further extend the Bell & LaPadula protection model to improve its usefulness. The extended protection model is known as the "Complete Protection Model" or CPM. This paper is divided into 8 sections.

1. This introduction containing background information.
2. Object/object computational model.
3. Migration/corruption levels.
4. Read/write Levels.
5. Integrated view.
6. Examples.
7. Conclusion.
8. References.

Definitions

Definitions will be introduced where needed. However, the following definitions are needed for a ground level understanding of the terms.

Operating System: An operating system (or just system) is a computer hardware/software executive designed to manage global resources.

Object: An object is any data storage entity in the system.

Process/Subject: The terms process and subject are synonymous and represent a program executing on the system acting on behalf of some system user.

Segment: A segment is an object whose contents can be manipulated by a subject without direct system software mediation.

Background

Security The Bell & LaPadula security model [1][2] enforces security by restricting the types of data flow in an operating system. The classical restrictions prevent data from being willfully compromised by unauthorized disclosure of information. These restrictions prevent:

a. a subject reading data residing at a higher security level; and

b. a subject writing data to a lower security level.

For example, a subject residing at the level of SECRET cannot read a TOP SECRET object nor write an UNCLASSIFIED one.

These restrictions apply to all system subjects except those required to perform administrative actions. In the case of these administrative operations, the restrictions are either circumvented or ignored.

Definitions

Security Level: All objects and subjects of the system have an associated security level. The set of all possible security levels is partially ordered.

Simple Security Violation: A simple security violation occurs when data at a high security level is read from a lower security level.

*-property Security Violation: A *-property security violation occurs when data at a high security level is written to a lower security level.

Integrity To guarantee that certain pieces of data are not corrupted or otherwise tampered with by unauthorized subjects, the military security policy has been to require classification of the data. Unfortunately, classifying the data is not always the best way of dealing with the problem. Integrity considerations and restrictions (introduced by K.J. Biba [3]) solves some of the corruption and tampering problems.

In particular, the Bell & LaPadula model does not prevent unauthorized writing to higher security levels. This lack of restrictions has an undesirable side effect: namely, corruption and tampering of high level data is allowed from lower security levels. The integrity extension to the Bell & LaPadula model adds restrictions which prevent:

a. a subject writing to a higher integrity level; and

b. a subject reading from a lower integrity level.

The judicious use of integrity levels can prevent corruption and tampering problems but can lead to stratification. Hence, integrity restrictions (though helpful) are not always desirable.

Definitions

Integrity Level: All objects and subjects of the system have an associated integrity level. The set of all possible integrity levels is partially ordered.

Simple Integrity Violation: A simple integrity violation occurs when data at a low integrity level is read from a higher integrity level.

*-property Integrity Violation: A *-property integrity violation occurs when data at a low integrity level is written to a higher integrity level.

KSOS The integrity extended Bell & LaPadula model was used on the Kernelized Secure Operating System (KSOS) [4][5] project. In addition, the KSOS model included privileges. Privileges allow controlled downgrading of information by circumventing the model. Once given the privilege to circumvent the KSOS model, a subject can violate it to any degree desired.

The KSOS computational model assumes that subjects (i.e., processes) are data storage objects. When checking the security and integrity constraints, only the object involved, and the process itself, are checked. This object behavior of subjects introduces a couple of undesirable side effects. First, the interprocess communication (IPC) mechanism permits writing to processes. This functionality causes object tranquility problems. Second, if a process is allowed to change levels, then it may have segments of widely differing security and integrity levels in its address space at the same time.

OBJECT/OBJECT MODEL

To avoid the subject/object computational model complexity (as encountered in KSOS), an object/object computational model is introduced. The object/object computational model states that no subjects ever behave as objects, i.e., subjects are never objects. Data flow, by definition, is allowed only between objects; never between objects and subjects.

Definitions

Explicit Connection: When a subject wishes to transfer data between object O1 and object O2, it must establish an explicit connection between the objects. To establish an explicit connection, access checks to O1 and O2 must be made. Once an explicit connection is established between two objects, a subject can then cause data to flow by making the appropriate system calls. For the sake of simplicity, all explicit connections are unidirectional.

Implicit Connection: If a subject is allowed to simultaneously maintain more than one segment, then when adding a new segment an implicit connection must be made between the new segment and every segment from which (or to which) data may flow. That is, no subject

is allowed to artificially construct a connection (using segments) which bypasses (explicit) connection establishment protocol.

Implications

There are several implications of an object/object (with connections) computational model.

1. In terms of security and integrity, only connections must be modeled.

2. Connection establishment may be independent of the security and integrity levels of the objects with respect to the security and integrity levels of the subject.

3. Data passing through an explicit connection is free from subject eavesdropping or modification.

4. All interprocess communication will be through shared segments.

The object/object model is a much better model of data flow than the subject/object one. Furthermore, it is sufficiently general to allow great freedom of choice for the connection establishment protocol.

MIGRATION AND CORRUPTION

The unrestricted flow of information upward under the KSOS protection model causes two very subtle security violations. The first violation is caused by inadequately restricted write up. For example, a user can destroy or subtlely modify data at a higher security level. This action is clearly intolerable in a multi-level secure system. It can be limited somewhat by judicial usage of integrity levels, but it cannot be entirely eradicated since many applications will require limited types of write up.

The second violation is caused by unrestricted read down. Since no restrictions exist on the ability to read down under the KSOS protection model, then a user can execute a program which resides at a lower security level. Since the program is constructed and modified at the lower security level, it could be used to modify or destroy highly classified material instead of its purported functionality. Again, these types of actions can be limited somewhat by judicial usage of integrity levels. However, as in the write up case, there are applications where the ability to read down cannot (or should not) be prevented.

In the CPM, the restrictions on the upward data flow between objects are enforced by migration and corruption constraints.

Definitions

Each system object will have three security and three integrity levels associated with it. They are migration security and integrity levels, absolute security and integrity levels, and corruption security and integrity levels.

Migration Level: The migration level (ML) of an object is the highest security level (MSL) and the lowest integrity level (MIL) to which data in the object may flow.

Absolute Level: The absolute level (AL) of an object is the security level (ASL) and the integrity level (AIL) at which the data in the object is classified.

Corruption Level: The corruption level (CL) of an object is the lowest security level (CSL) or the highest integrity level (CIL) from which data may flow into that object.

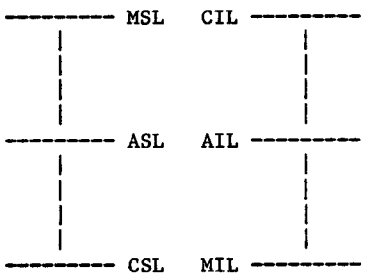
Implications

The addition of migration and corruption levels to the integrity extended Bell & LaPadula model adds enforced upward (for security) and downward (for integrity) data flow restrictions. The migration and corruption levels can be used to both allow and restrict migration of information in a system. For example, some unclassified system users may be permitted to write general purpose programs for everyone to use while others may not.

To further restrict the domain of this discussion, the following relations are required for all objects:

$$MSL \geq ASL \geq CSL$$
$$MIL \leq AIL \leq CIL$$

(Hereafter, if any of the inequalities are undefined due to the partial ordering of security or integrity levels, then the inequality is defined to be false.) These inequalities require that the absolute levels for security and integrity be bracketed by the migration and corruption levels. Pictorially:



(Note: this picture is for intuitive understanding only. No relationship between security and integrity levels should be derived from it.)

READ/WRITE LEVELS

One of the flaws of the KSOS protection model is that controlled security violations (such as downgrading) can only be performed by privileged subjects. The trouble with privileges is that they are pervasive. Given the privilege to write down, a subject can write data downward to any lower security level. That is, either a subject is totally trusted or it is totally untrusted. Therefore, if an application requires the ability to perform limited security violations, then the application must be trusted not to exceed its limitations. Furthermore, distributing the ability to perform controlled security violations has to be very carefully controlled.

A solution to this problem is to incorporate controlled violations into the security model. In addition, limited violations should be feasible without verification of self imposed limits. Subject read and write levels comprise the solution.

Definitions

Each subject in the system will have three security and integrity levels associated with it. They are read security and integrity levels, absolute security and integrity levels, and write security and integrity levels.

Read Level: The read level (RL) of a subject is the highest security level (RSL) and lowest integrity level (RIL) from which the subject is allowed to read.

Absolute Level: The absolute level (AL) of a subject is the security level (ASL) and integrity level (AIL) given the subject upon creation. Typically this level will be the level of the user on behalf of whom the subject is acting.

Write Level: The write level (WL) of a subject is the lowest security level (WSL) and highest integrity level (WIL) to which the subject is allowed to write.

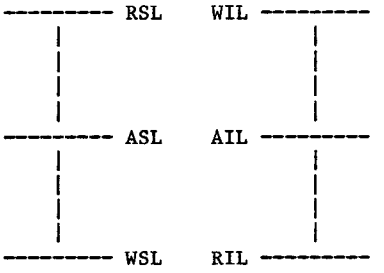
Implications

The addition of read and write levels to the integrity extended Bell & LaPadula model permits limited and controlled security violations. For example, if the read security level of a subject is higher than its absolute security level, or if the write security level of a subject is less than its absolute security level, then the subject possesses the ability to perform controlled security violations.

To further restrict the domain of this discussion, the following relations are required for all subjects:

$$RSL \geq ASL \geq WSL$$
$$RIL \leq AIL \leq WIL$$

These inequalities require that the read and write levels bracket the absolute level of the subject. Pictorially:



(Note: this picture is for intuitive understanding only. No relationship between security and integrity levels should be derived from it.)

INTEGRATED VIEW

Introduction

In previous paragraphs, the object/object model, migration/corruption levels, and read/write levels were introduced. This section will integrate these concepts into a complete flow restriction model of protection (CPM).

Combining the concepts of object/object data flow, migration/corruption levels, and read/write levels allows a user to flexibly utilize the system. Furthermore, since the object/object model

simplifies data flow models, and since migration/corruption and read/write levels simplify the protection model, then proving security and integrity properties about the system becomes easier.

Implications

Connections The basic model of data flow of the CPM incorporates connections. Remember, connections are established between objects to perform data transfers. When a connection is established, the system must perform the appropriate security and integrity checks to guarantee adequate protection model enforcement. Once a connection has been established, no further protection model checks are needed to cause data flow.

The following assumptions about the system are necessary to properly discuss the rules of connection establishment.

1. Each system object has a migration level, a corruption level, and an absolute level associated with it.

2. Each subject has a read level, a write level, and an absolute level associated with it.

3. No subject or object will ever change absolute levels.

In terms of the protection model, only the connection establishment rules need to be investigated. Given objects O1 and O2 and a subject P, the following security and integrity levels inequalities must be true for P to establish a connection from O1 to O2:

$$O1MSL \geq O2MSL \quad (S1)$$
$$O1CSL \geq O2CSL \quad (S2)$$
$$PRSL \geq O1ASL \quad (S3)$$
$$O2ASL \geq PWSL \quad (S4)$$
$$PASL \geq O2CSL \quad (S5)$$
$$O1MSL \geq PASL \quad (S6)$$

$$O1CIL \leq O2CIL \quad (I1)$$
$$O1MIL \leq O2MIL \quad (I2)$$
$$PRIL \leq O1AIL \quad (I3)$$
$$O2AIL \leq PWIL \quad (I4)$$
$$PAIL \leq O2CIL \quad (I5)$$
$$O1MIL \leq PAIL \quad (I6)$$

The meaning of these inequalities will be discussed in the next section.

Consistency There are six types of consistency which must be enforced. They are:

1. the migration properties of object O1;

2. the corruption properties of object O2;

3. the security properties of P with respect to O1 and O2;

4. the integrity properties of P with respect to O1 and O2;

5. the corruption properties of O2 with respect to P; and

6. the migration properties of O1 with respect to P.

Migration Properties To ensure the soundness of the migration properties, it is essential that the migration properties of the object O2 be at least as restrictive as the migration properties of object O1. In the security level case, if the migration security level of object O2 is greater than the migration security level of object O1, then the migration security property of object O1 has been circumvented. Therefore, to guarantee the migration properties of object O1, the following inequalities are required:

$$O1MSL \geq O2MSL \quad (S1)$$
$$O1MIL \leq O2MIL \quad (I1)$$

Corruption Properties To ensure the soundness of the corruption properties, it is essential that the corruption properties of the object O1 be at least as restrictive as the corruption properties of object O2. In the security level case, if the corruption security level of object O1 is lower than the corruption security level of object O2, then the corruption security level of object O2 has been circumvented. Therefore, to guarantee the corruption properties of object O2, the following inequalities are required:

$$O1CSL \geq O2CSL \quad (S2)$$
$$O1CIL \leq O2CIL \quad (I2)$$

Security Properties To ensure the soundness of the security requirements of the system, the subject P must have read access to the object O1 and have write access to the object O2. To guarantee the security properties of the system, the following inequalities are required:

$$PRSL \geq O1ASL \quad (S3)$$
$$O2ASL \geq PWSL \quad (S4)$$

Integrity Properties To ensure the soundness of the integrity requirements of the system, the subject P must have read access to the object O1 and have write access to the object O2. To guarantee the integrity properties of the system,

the following inequalities are required:

$$PRIL \leq O1AIL \quad (I3)$$
$$O2AIL \leq PWIL \quad (I4)$$

Write/corruption Properties To guarantee the corruption properties of object 02 with respect to the subject P, the following inequalities are required:

$$PASL \geq O2CSL \quad (S5)$$
$$PAIL \leq O2CIL \quad (I5)$$

Read/migration Properties To guarantee the migration properties of object 01 with respect to the subject P, the following inequalities are required:

$$O1MSL \geq PASL \quad (S6)$$
$$O2MIL \leq PAIL \quad (I6)$$

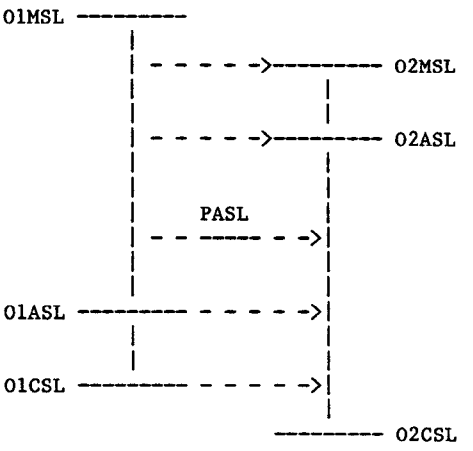
Special Cases There are two special cases of the CPM. They are: the case where the subject is trusted and the case where the subject is untrusted. A subject (or process) is trusted whenever its write level or its read level is not equal to its absolute level. The following discussions are for security only. The integrity discussion is analogous.

Trusted Subjects Consider the case where the write security level of a subject is less than its absolute security level, which in turn is less than its read security level. In this case, the subject can read objects at higher security levels and can write objects at lower security levels. Note that two of the six security inequalities for establishing a connection do not involve subject security levels. These properties are independent of the "trusted" nature of the subject. Those two inequalities comprise the migration and corruption properties. (The term "trusted" becomes less important under the CPM since the "trusting" is not absolute.)

Untrusted Subjects Consider the case where the read and write levels of a subject are equal to its absolute level, i.e., subjects are not trusted. In this case, connection establishment inequalities for security become:

$$O1MSL \geq O2MSL$$
$$O1CSL \geq O2CSL$$
$$O2ASL \geq PASL \geq O1ASL$$

Pictorially:



Note that when the O1CSL=O2CSL=system low and O1MSL=O2MSL=system high, the resulting access is similar to that allowed by the KSOS protection model.

EXAMPLES

Below are several examples of how the CPM can be utilized. Note that each of the applications below can be utilized in the same system!

ACCAT GUARD

Consider the problem of inter-security level mail. This problem is directly addressed by the ACCAT GUARD project[6]. In particular, ACCAT GUARD is responsible for secure transference of information between a high security level network and low security level network. The heart of the ACCAT GUARD design is the Security Watch Officer (SWO) who is responsible for downward data transfers. The SWO has three options when downgrading messages:

1. He can downgrade the message without modifications.

2. He can sanitize the message and then downgrade it.

3. He can deny the downgrade request.

The analogous problem in a multi-level secure mail facility is much harder since the number of possible levels through which a message needs to pass can be very large. The probability that one SWO can handle the bulk of the traffic is very low. Furthermore, the probability that one SWO is able to determine whether or not an arbitrary message contains classified information is also very low. The solution (using CPM) is to have several SWO's whose sphere of influence (levels

of responsibility) are limited. This can be done by appropriate assignments of a SWO's read and/or write levels.

Executable Images

In a general purpose multi-level operating system, there typically is little control over program development. Since any program which can be read can often be executed, then care must be taken to prevent Trojan horses from being inserted into the system. One approach to preventing these types of Trojan horses is through the proper use of migration levels.

Every user of the multi-level system should be assigned a login migration level. The user will be able to create objects with migration levels less than or equal to his login migration level. If a user is not cleared to any level, his login migration level should be set to system low to prevent cleared users from reading and/or executing his programs.

Isolation

To implement an isolation subsystem using CPM, all one needs to do is:

i. make the read and write levels of each subsystem subject equal to its absolute level;

ii. make the migration and corruption levels of each subsystem object equal to its absolute level; and

iii. restrict all discretionary modes of access to members of the subsystem.

Stratification

To implement a stratification subsystem using CPM, one can implement several isolation subsystems with little or no discretionary access modes restriction between them.

CONCLUSION

The Complete Protection Model closes up some obvious holes in the KSOS security/integrity model. However, more work in this area is needed if security models will ever accurately reflect military security policies. More investigation needs to be done to determine the working attributes of CPM.

REFERENCES

1. Bell, D.E. and LaPadula, L.J., "Secure Computer Systems", ESD-TR-73-278, Volume I-III, The MITRE Corporation, Bedford, MA (November 1973 - June 1974)

2. Bell, D.E. and LaPadula, L.J., "Computer Security Model: Unified Exposition and Multics Interpretation," ESD-TR-75-306, The MITRE Corporation, Bedford, MA (June 1975)

3. Biba, K.J., "Integrity Considerations for Secure Computer Systems", MTR-3153, MITRE Corporation, Bedford, MA (June 1975)

4. "KSOS Computer System Specification (Type A)", WDL-TR7808, Ford Aerospace & Communications Corporation, Palo Alto, CA (July 1978)

5. "KSOS Security Kernel Computer Program Development Specification (Type B5)", WDL-TR7932, Ford Aerospace & Communications Corporation, Palo Alto, CA (September 1978)

6. "ACCAT GUARD Computer Program Development Specification (Type B5)", ARPA-78C0323-01, LOGICON, San Diego, CA (February 1979)