Biba Security Model Comparison

By:
Nathan Balon
Ishraq Thabet

April 7, 2004
Winter 2004
CIS 576

Introduction

Security models define the basis for creating a security policy. Many different security models have been created to address different security concerns. The goal of all security models is to define the authorized and unauthorized states of a system and to restrict the system to moving into an unauthorized state. Models are based on either mandatory or discretionary access control. In turn, a security model produces a software-independent, conceptual model, which is used to enforce security. Security models have been developed based on the type of system that they will be used on. Some models are used to provide security for operating systems, while other are used for a specific application such as a database. This paper will explore some of the similarities and difference between different security models. The Biba security model will be used as the baseline model of the paper. The Biba model will be compared first to the Take-Grant model and then to the Sea-View model.

Biba Model

The motivation for creating the Biba model was for preserving the integrity in a computer system. The model prevents the unauthorized modification of data and maintains the consistency of the data (Bishop 2003). The Biba model assigns the subjects and objects an integrity label. The integrity label is used to tell the degree of confidence that may be placed in the data (RFC 1457). The Biba model contains four access modes: modify, observe, invoke, and execute.

The Biba model defines a family of different policies that can be used to enforce integrity. The Biba model supports five mandatory policies: strict integrity policy, low-watermark policy for subjects, low-watermark for objects, low-watermark integrity audit policy, and ring policy. Three discretionary policies have been established: access control lists, object hierarchy, and ring. In practice the mandatory policies of the Biba model are most often used.

The most restrictive of the policies of the Biba model is the strict integrity property. The strict integrity property is essentially a reverse of the Bell-LaPadula model. The strict integrity model is composed of three properties. First, the simple integrity condition allows a subject to observe (read) an object only if the integrity level of the subject is less than the integrity level of the object. The simple integrity condition enforces "no read down". Second, the integrity star property states that a subject can modify (write to) an object only if the integrity level of the object is less than the integrity level of the object. The integrity star property enforces "no write up". Last, in the invocation property, a subject can invoke another subject only if the integrity level of the subject being invoked is less than the subject that invoked it. The figures below show the simple integrity condition (Fig. 1) and the integrity star property (Fig. 2).

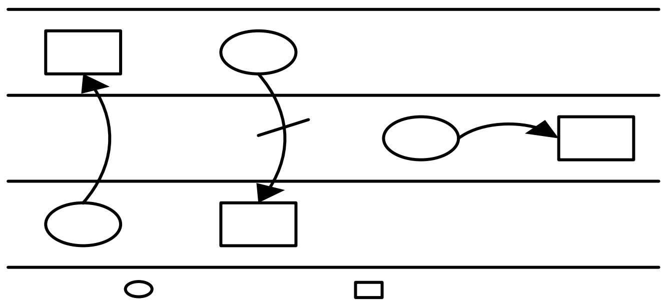
Fig. 1 Simple Integrity Condition

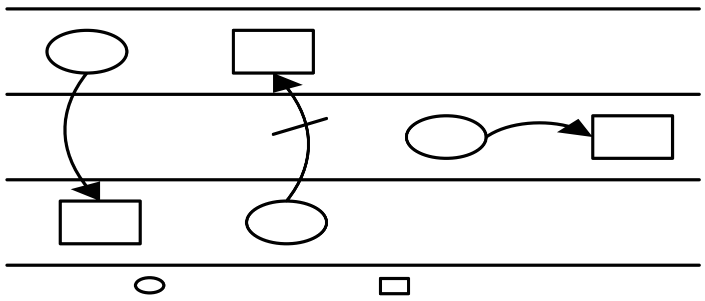
Fig. 2 Integrity Star Property

Based on the strict integrity condition, a number of other policies were established. Each of the other policies is less restrictive than the strict integrity policy. The model has a number of dynamic policies that lower the integrity levels of either subjects or objects based on the access mode operation.

The Biba model is not without its problems. The model does not support the granting and revocation of authorizations. Another problem is that the model is used strictly for integrity; it does nothing to enforce confidentiality. For these reasons, to use the Biba model, it should be combined with other security models. The Lipner Integrity model is one model that combines the Biba model and the Bell-LaPadula models together to enforce both integrity and confidentiality.

Take-Grant Model

The Take-Grant model is a discretionary model. This is in contrast to the Biba model, which is mostly used to support the mandatory policies. Like the Biba model, the Take-Grant model also describes security based on subjects and objects. The Take-Grant model uses a directed graph to describe the protected states. In the graph, the subjects and objects are represented as nodes. In the graph: a ● represents a subject, a ○ represents an object, and a ⊗ can be either a subject or an object (Bishop 1981). The access modes are represented by an edge, which connects nodes such as a subject to an object. The edge also has a label which states the right associated with it. The Take-Grant model uses a graph to model access control; the model is fundamentally an access matrix model (Landwehr).

The access modes supported by the Take-Grant model are read, write, take and grant. The arcs between nodes are labeled with these access rights. The read and write are comparable to observe and modify in the Biba model. The take and grant access modes support operation that are not supported by the Biba model. The take and grant operations shown below are publicized in the paper "Formal Models for Computer Security". Suppose there are two nodes in a graph, node A and node B, the graph can be used to show the results of the operations. For instance, if nodes A were to take the read right that node B has on node C, the graph would be as follows:

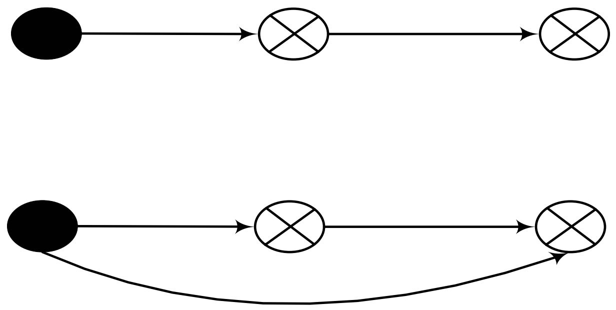

Fig. 3 Take Right

The grant right allows a subject to grant any writes that the subject posses to another subject or object. The grant operation can be shown as follows. If subject A posses the grant right to B it can grant any of the other rights that A posses to B.

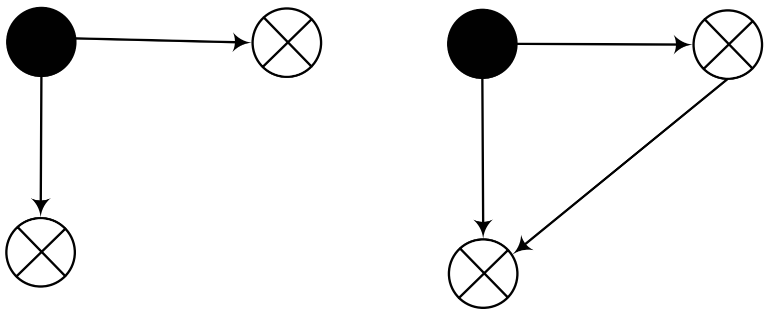

Fig. 4 Grant Right

The take and grant rights, also known as transfer rights, support the modification of system authorizations, which allow rights to be transferred to other objects. The transfer rights allow a subject to give or take away the rights of an object. This overcomes one of the problems of the Biba model, which is that it does not provide any administrative options for granting and revoking authorizations (Castano).

The authorization graph can be modified by four operations: take, grant, create and remove (Slomka, & Upendranathan). Create and remove provide two additional administrative operations. First, create allows a subject to create another object (or subject). The result of the operation is a new node in the graph with a set of rights assigned to it. Second, remove allows a subject to revoke rights from an object (or subject). The result of this operation is the removal of rights from the set of rights possessed by the object. If no rights remain in the set of rights of the object node, it is deleted from the graph.

The Take Grant model can easily be applied to database security. Some models, as in the case of the Biba model, were not created to be used with database. It is difficult to use the Biba model in a DBMS. Many DBMS use the Take-Grant model for the authorization of privileges. For instance, Oracle uses the Grant command to let Oracle know that you are authorizing a user or role to have one or more system or object privileges (Theriault and Newman). Revoke, on the other hand, removes a privilege that a user has on an object. The syntax for the two commands in Oracle are:

```
grant <privilege> to <USERNAME or ROLE>
revoke <privilege> from <USERNAME>
```

The Take Grant model is not without it problems. For example, the Take Grant model does not consider the issue of integrity as the Biba model does. The Take Grant model is also vulnerable to Trojan horse attacks. A final problem is that the model is limited in the number of nodes that can be shown at one time with the model. A graph would be hard for an individual to comprehend and prove secure if there were thousands of nodes and arcs in it.

Sea-View Model

Denning et al. at the Stanford Research Institute (SRI) developed the Sea View Model (**SE**cure d**A**ta **VIEW**) model. Like the Biba model, the Sea View model uses mandatory as well as discretionary policies to govern access to data stored in the database. However, unlike the Biba model where secrecy is not addressed, the Sea View model combines both secrecy and integrity.

The Sea View model consists of two layers. The first layer, which corresponds to a reference monitor that enforces the mandatory security policy of the Bell-LaPadula model, is the Mandatory Access Control (MAC). The second layer is the Trusted Computing Base (TCB) and it “defines the concept of multilevel relations, supports discretionary controls for multilevel relations and views, and formalizes the supporting policies” (Castano, S et al.)

In the MAC model, user is given an access to classified information based on the user’s clearance (secrecy and integrity authorization) for the information. The Sea View mandatory policy is formalized in terms of subjects, objects, and access classes. It implements the same axioms used in the Bell-LaPadula and Biba models.

An access class consists of two components: secrecy and integrity. The secrecy class corresponds to the security level of the Bell-LaPadula model whereas the integrity class corresponds to the integrity level of the Bib model. Like the Biba model, the Sea View model forms a lattice of relationship. An access class C_1 dominates ($\supseteq$) an access class C_2 if, and only if, the secrecy component of C_1 dominates the secrecy component of C_2 and the integrity component of C_1 is dominated by the integrity component of C_2.

Each object of the MAC model, which is defined to be the information containers (files), is associated with a unique identifier and a unique access class. The identifier and access classes are fixed for the whole life of the object. As in the Biba model, subjects are defined as the processes acting on behalf of the users. Unlike the Biba model, however, each user in the system using the Sea View model is assigned a range of secrecy and integrity classes in which the user is allowed to operate. Each user is assigned a minimal secrecy and minimal integrity classes as well as a maximal secrecy and maximal integrity classes. These classes are referred to as minsecrecy, minintegrity, maxsecrey, and maxintegrity, respectively. The writeclass of the subject is composed of the pair minisecrecy and maxintegrity of the user and the readclass is composed of the maxsecrecy and minintegrity of the user.

The mandatory access modes of the Sea View models are: read, write, and execute. The read operation is similar to the *observe* access mode of the Biba model, which allows the subject to read information stored in an object. The write access mode allows subject to write information into an object and execute is to execute on object. These modes are comparable to the *modify* and *execute* operations of the Biba model, respectively. The read property states that a subject *i* can read an object *j* only if its readclass dominates the access class of the object. This property is the formulation of the No Read-Up secrecy of the Bell-LaPadula model, and the No Read-Down integrity of the Biba model. In the write property, a subject *i* can write an object *j* if its writeclass is dominated by the access class of the object. This property is the formulation of the No Write Down secrecy of the Bell-LaPadula model, and the No Write Up integrity of the Biba model. Finally, the execute property allows subject *i* to execute object j only if its maxintegrity is less than or equal to the integrity class of the object, and its maxsecrecy is greater than or equal to the secrecy class of the object. This property overcomes the limitation of the Biba model. Hence, the Sea View model accomplishes this by "distinguishing *execute* access from *read* access, allowing trusted subjects to read data of lesser integrity level than their maxintegrity, and restricting execute access for all subjects to programs of greater or equal integrity." (Castano, S et al.)

Problems with Security Models

There are problems with using security models in general. Dorothy Denning points out a number of problems arise from using security models.

1. Security models have theoretical limits. You cannot always prove that a model satisfies certain security conditions.
2. Security models based on strict mathematical properties can lead to systems that are totally unusable.
3. Building systems from rigorous mathematical security models is extremely time-consuming and costly. Most commercial systems will not be based on formal models.
4. Security models and formal methods do not establish security. Systems are hacked outside the models' assumptions.
5. Provable security, even if it were achievable, is not a panacea.

Even when a Security model proves that a system is secure, the system may be vulnerable. The human element and social engineering have allowed even the most secure systems to be compromised. Also, security mechanisms that are at one time thought to be secure in the future can be found to be insecure. The DES encryption algorithm is a perfect example of this. Because of the increased computing power available today DES can no longer be viewed as secure. No matter how secure a system is proved to be by a model, the system is always at risk.

Denning was one of the creators of the Sea-View model. In a speech given by Denning, she states,

> "The SeaView, grew progressively more complex as we attempted to address the real issues. By the time I left SRI in 1987, I was convinced that I would never want to use a system based on SeaView. Any hope of usability had been killed by a concept called polyinstantiation, which involved instantiating multiple data values within a single field of a record, all with different security classifications. Polyinstantiation was needed to satisfy the mathematical models of multilevel security, but it got uglier and uglier the deeper we went. I learned then that security models could lead to dreadful systems that nobody would ever use."

Conclusion

The Biba, Take-Grant and Sea-View Models were developed to address different security issues. No model is perfect in all situations. Because of this, a number of different models have been developed. Even one model may not be sufficient to support one aspect of security, such as confidentiality or integrity. Evidence of this is the Biba model which creates a number of different policies to be used based on the needs of the system. In the case of the Biba model, some polices are more restrictive than others. For example, the low-water mark policies are dynamic and lower the integrity levels of subjects or objects, which results in a model that is easier to use but less secure. The Take-Grant model on the other hand supports the authorization of privileges, which the Biba model was mute on. In many cases more than one security model will be needed to enforce security.

Computer security will always be partly a reactionary process. No matter how much active security is used and planning, problems will arise. While it is good to use a sound security model, there will always be security breaches. Even models that at one time were thought to be secure, have later been found to be vulnerable to attacks. The HRU model is one such model that was later found to have vulnerabilities, because “it is theoretically undecidable whether an arbitrary access-matrix model was safe” (Denning).

References

Bishop, M. “Hierarchical Take-Grant Protection System” Proceedings of the eighth ACM symposium on Operating systems principles. Pacific Grove, California, pgs. 109 - 122 1981.

Bishop, M. Computer Security: Art and Science, Addison Wesley, Boston, MA. 2003.

Castano, S. (et. al). Database Security, Addison Wesley, Harlow, England. 1995.

Denning, D. “The Limits of Formal Security Models”. National Computer Systems Security Award Acceptance Speech. October 18, 1999.

Landwehr, C. “Formal Models for Computer Security”, Computing Surveys, Vol. 13, No. 3, September 1981.

RFC 1457. “Security Label Framework for the Internet” http://www.ietf.org/rfc/rfc1457.txt

Slomka, P. & Upendranathan, K. “Understanding and Illustrating the Take-Grant Model”, CIS 576, University of Michigan Dearborn.

Therialut, M. & Newman, A. Oracle Security Handbook: Implementing a Sound Security-Plan in Your Oracle Environment. McGraw-Hill, Berkeley, CA. 2001.