DATA SECURITY MANAGEMENT

SECURITY MODELS FOR OBJECT-ORIENTED DATA BASES

James Cannady

INSIDE: BASICS OF DATA BASE SECURITY:
Discretionary vs. Mandatory Access Control Policies; Securing a RDBMS vs. OODBMS; Relational DBMS Security; OODBMS Characteristics; Proposed OODBMS Security Models

INTRODUCTION
Object-oriented (OO) programming languages and OO analysis and design techniques influence data base systems design and development. The inevitable result is the object-oriented data base management system (OODBMS).

Many of the established data base vendors are incorporating object-oriented concepts into their products in an effort to facilitate data base design and development in the increasingly object-oriented world of distributed processing. In addition to improving the process of data base design and administration, the incorporation of object-oriented principles offers new tools for securing the information stored in the data base. This article explains the basics of data base security, the differences between securing relational and object-oriented systems, and some specific issues related to the security of next-generation OODBMSs.

BASICS OF DATA BASE SECURITY
Data base security is primarily concerned with the secrecy of data. Secrecy means protecting a data base from unauthorized access by users and software applications.

Secrecy, in the context of data base security, includes a variety of threats incurred through unauthorized access. These threats range...
from the intentional theft or destruction of data to the acquisition of information through more subtle measures, such as inference. There are three generally accepted categories of secrecy-related problems in data base systems:

- The improper release of information from reading data that was intentionally or accidentally accessed by unauthorized users. Securing data bases from unauthorized access is more difficult than controlling access to files managed by operating systems. This problem arises from the finer granularity that is used by data bases when handling files, attributes, and values. This type of problem also includes the violations to secrecy that result from the problem of inference, which is the deduction of unauthorized information from the observation of authorized information. Inference is one of the most difficult factors to control in any attempts to secure data. Because the information in a data base is semantically related, it is possible to determine the value of an attribute without accessing it directly. Inference problems are most serious in statistical data bases, where users can trace back information on individual entities from the statistical aggregated data.

- The improper modification of data. This threat includes violations of the security of data through mishandling and modifications by unauthorized users. These violations can result from errors, viruses, sabotage, or failures in the data that arise from access by unauthorized users.

- Denial-of-service threats. Actions that could prevent users from using system resources or accessing data are among the most serious. This threat has been demonstrated to a significant degree recently with the SYN flooding attacks against network service providers.

### Discretionary vs. Mandatory Access Control Policies

Both traditional relational data base management system (RDBMS) security models and object-oriented data base models make use of two general types of access control policies to protect the information in multilevel systems. The first of these policies is the discretionary policy. In the discretionary access control (DAC) policy, access is restricted based on the authorizations granted by the owner to the user.

The mandatory access control (MAC) policy secures information by assigning sensitivity levels, or labels, to data entities. MAC policies are more secure than DAC policies and they are used in systems in which security is critical, such as military applications. However, the price that is usually paid for this tightened security is reduced performance of the data base management system. MAC policies also incorporate DAC measures as well.
The development of secure models for object-oriented DBMSs has obviously followed on the heels of the development of the data bases themselves. The theories that are currently being researched and implemented in the security of object-oriented data bases are also influenced heavily by the work that has been conducted on secure relational data base management systems.

**Relational DBMS Security**

The principal methods of security in traditional RDBMSs are through the appropriate use and manipulation of views and the SQL (structured query language) GRANT and REVOKE statements. These measures are reasonably effective because of their mathematical foundation in relational algebra and relational calculus.

**View-based Access Control.** Views allow the data base to be conceptually divided into pieces in ways that allow sensitive data to be hidden from unauthorized users. In the relational model, views provide a powerful mechanism for specifying data-dependent authorizations for data retrieval.

Although the individual user who creates a view is the owner and is entitled to drop the view, he or she may not be authorized to execute all privileges on it. The authorizations that the owner may exercise depend on the view semantics and on the authorizations that the owner is allowed to implement on the tables directly accessed by the view. For the owner to exercise a specific authorization on a view that he or she creates, the owner must possess the same authorization on all tables that the view uses. The privileges the owner possesses on the view are determined at the time of view definition. Each privilege the owner possesses on the tables is defined for the view. If, later on, the owner receives additional privileges on the tables used by the view, these additional privileges will not be passed onto the view. In order to use the new privileges within a view, the owner will need to create a new view.

The biggest problem with view-based mandatory access controls is that it is impractical to verify that the software performs the view interpretation and processing. If the correct authorizations are to be assured, the system must contain some type of mechanism to verify the classification of the sensitivity of the information in the data base. The classification must be done automatically, and the software that handles the classification must be trusted. However, any trusted software for the automatic classification process would be extremely complex. Furthermore, attempting to use a query language such as structured query language (SQL) to specify classifications quickly becomes convoluted and complex. Even when the complexity of the classification scheme is
overcome, the view can do nothing more than limit what the user sees—it cannot restrict the operations that may be performed on the views.

**GRANT and REVOKE Privileges.** Although view mechanisms are often regarded as security “freebies” because they are included within structured query language (SQL) and most other traditional relational data base managers, views are not the sole mechanism for relational data base security. GRANT and REVOKE statements allow users to selectively and dynamically grant privileges to other users and subsequently revoke them if necessary. These two statements are considered to be the principal user interfaces in the authorization subsystem.

There is, however, a security-related problem inherent in the use of the GRANT statement. If a user is granted rights without the GRANT option, he or she should not be able to pass GRANT authority on to other users. However, the system can be subverted by a user by simply making a complete copy of the relation. Because the user creating copy is now the owner, he or she can provide GRANT authority to other users. As a result, unauthorized users are able to access the same information that had been contained in the original relation. Although this copy is not updated with the original relation, the user making the copy could continue making similar copies of the relation, and continue to provide the same data to other users.

The REVOKE statement functions similarly to the GRANT statement, with the opposite result. One of the characteristics of the use of the REVOKE statement is that it has a cascading effect. When the rights previously granted to a user are subsequently revoked, all similar rights are revoked for all users who may have been provided access by the originator.

**Other Relational Security Mechanisms.** Although views and GRANT/REVOKE statements are the most frequently used security measures in traditional RDBMSs, they are not the only mechanisms included in most security systems using the relational model. Another security method used with traditional relational data base managers, which is similar to GRANT/REVOKE statements, is the use of query modification.

This method involves modifying a user’s query before the information is retrieved, based on the authorities granted to the user. Although query modification is not incorporated within SQL, the concept is supported by the Cobb-Date relational data base model.

Most relational data base management systems also rely on the security measures present in the operating system of the host computer. Traditional RDBMSs such as DB2 work closely with the operating system to ensure that the data base security system is not circumvented by permitting access to data through the operating system. However, many
operating systems provide insufficient security. In addition, because of the portability of many newer data base packages, the security of the operating system should not be assumed to be adequate for the protection of the wealth of information in a data base.

Object-Oriented DBMS Characteristics

Unlike traditional RDBMSs, secure object-oriented DBMSs (or OODBMSs) have certain characteristics that make them unique. Furthermore, only a limited number of security models have been designed specifically for object-oriented data bases. The proposed security models make use of the concepts of encapsulation, inheritance, information hiding, methods, and the ability to model real-world entities that are present in object-oriented environments.

The object-oriented data base model also permits the classification of an object's sensitivity through the use of class (of entities) and instance. When an instance of a class is created, the object can automatically inherit the level of sensitivity of the superclass. Although the ability to pass classifications through inheritance is possible in object-oriented data bases, class instances are usually classified at a higher level within the object's class hierarchy. This prevents a flow control problem, where information passes from higher to lower classification levels.

Object-oriented DBMSs also use unique characteristics that allow these models to control the access to the data in the data base. They incorporate features such as flexible data structure, inheritance, and late binding. Access control models for OODBMSs must be consistent with such features. Users can define methods, some of which are open for other users as public methods. Moreover, the OODBMS may encapsulate a series of basic access commands into a method and make it public for users, while keeping basic commands themselves away from users.

Proposed OODBMS Security Models

Currently only a few models use discretionary access control measures in secure object-oriented data base management systems.

Explicit Authorizations. The ORION authorization model permits access to data on the basis of explicit authorizations provided to each group of users. These authorizations are classified as positive authorizations because they specifically allow a user access to an object. Similarly, a negative authorization is used to specifically deny a user access to an object.

The placement of an individual into one or more groups is based on the role that the individual plays in the organization. In addition to the positive authorizations that are provided to users within each group,
there are a variety of implicit authorizations that may be granted based on the relationships between subjects and access modes.

**Data-Hiding Model.** A similar discretionary access control secure model is the data-hiding model proposed by Dr. Elisa Bertino of the Universita di Genova. This model distinguishes between public methods and private methods.

The data-hiding model is based on authorizations for users to execute methods on objects. The authorizations specify which methods the user is authorized to invoke. Authorizations can only be granted to users on public methods. However, the fact that a user can access a method does not automatically mean that the user can execute all actions associated with the method. As a result, several access controls may need to be performed during the execution, and all of the authorizations for the different accesses must exist if the user is to complete the processing.

Similar to the use of GRANT statements in traditional relational database management systems, the creator of an object is able to grant authorizations to the object to different users. The “creator” is also able to revoke the authorizations from users in a manner similar to REVOKE statements. However, unlike traditional RDBMS GRANT statements, the data-hiding model includes the notion of protection mode. When authorizations are provided to users in the protection mode, the authorizations actually checked by the system are those of the creator and not the individual executing the method. As a result, the creator is able to grant a user access to a method without granting the user the authorizations for the methods called by the original method. In other words, the creator can provide a user access to specific data without being forced to give the user complete access to all related information in the object.

**Other DAC Models for OODBMS Security.** Rafiul Ahad has proposed a similar model that is based on the control of function evaluations. Authorizations are provided to groups or individual users to execute specific methods. The focus in Ahad’s model is to protect the system by restricting access to the methods in the database, not the objects. The model uses proxy functions, specific functions, and guard functions to restrict the execution of certain methods by users and enforce content-dependent authorizations.

Another secure model that uses authorizations to execute methods has been presented by Joel Richardson. This model has some similarity to the data-hiding model’s use of GRANT/REVOKE-type statements. The creator of an object can specify which users may execute the methods within the object.

A final authorization-dependent model emerging from OODBMS security research has been proposed by Dr. Eduardo B. Fernandez of Florida Atlantic University. In this model the authorizations are divided into
positive and negative authorizations. The Fernandez model also permits the creation of new authorizations from those originally specified by the user through the use of the semantic relationships in the data.

Dr. Naftaly H. Minsky of Rutgers University has developed a model that limits unrestricted access to objects through the use of a view mechanism similar to that used in traditional relational systems database management systems. Minsky’s concept is to provide multiple interfaces to the objects within the database. The model includes a list of laws, or rules, that govern the access constraints to the objects. The laws within the database specify which actions must be taken by the system when a message is sent from one object to another. The system may allow the message to continue unaltered, block the sending of the message, send the message to another object, or send a different message to the intended object.

Although the discretionary access control models do provide varying levels of security for the information within the database, none of the DAC models effectively addresses the problem of the authorizations provided to users. A higher level of protection within a secure object-oriented database model is provided through the use of mandatory access control.

**MAC Methods for OODBMS Security.** Dr. Bhavani Thuraisingham of MITRE Corp. proposed in 1989 a mandatory security policy called SORION. This model extends the ORION model to encompass mandatory access control. The model specifies subjects, objects, and access modes within the system, and it assigns security/sensitivity levels to each entity. Certain properties regulate the assignment of the sensitivity levels to each of the subjects, objects, and access modes. In order to gain access to the instance variables and methods in the objects, certain properties that are based on the various sensitivity levels must be satisfied.

A similar approach has been proposed in the Millen-Lunt model. This model, developed by Jonathan K. Millen of MITRE Corp. and Teresa Lunt of SRI/DARPA (Defense Advanced Research Projects Agency), also uses the assignment of sensitivity levels to the objects, subjects, and access modes within the database. In the Millen-Lunt model, the properties that regulate the access to the information are specified as axioms within the model. This model further attempts to classify information according to three different cases:

- The data itself is classified.
- The existence of the data is classified.
- The reason for classifying the information is also classified.
These three classifications broadly cover the specifics of the items to be secured within the data base; however, the classification method also greatly increases the complexity of the system.

**The SODA Model.** Dr. Thomas F. Keefe of Penn State University proposes a model called Secure Object-Oriented Data Base (SODA). The SODA model was one of the first models to address the specific concepts in the object-oriented paradigm. It is often used as a standard example of secure object-oriented models from which other models are compared.

The SODA model complies with MAC properties and is executed in a multilevel security system. SODA assigns classification levels to the data through the use of inheritance. However, multiple inheritance is not supported in the SODA model.

Similar to other secure models, SODA assigns security levels to subjects in the system and sensitivity levels to objects. The security classifications of subjects are checked against the sensitivity level of the information before access is allowed.

**Polyinstantiation.** Unlike many current secure object-oriented models, SODA allows the use of polyinstantiation as a solution to the multiparty update conflict. This problem arises when users with different security levels attempt to use the same information. The variety of clearances and sensitivities in a secure data base system result in conflicts between the objects that can be accessed and modified by the users.

Through the use of polyinstantiation, information is located in more than one location, usually with different security levels. Obviously, the more sensitive information is omitted from the instances with lower security levels.

Although polyinstantiation solves the multiparty update conflict problem, it raises a potentially greater problem in the form of ensuring the integrity of the data within the data base. Without some method of simultaneously updating all occurrences of the data in the data base, the integrity of the information quickly disappears. In essence, the system becomes a collection of several distinct data base systems, each with its own data.

**CONCLUSION**
The move to object-oriented DBMSs is likely to continue for the foreseeable future. Because of the increasing need for security in distributed processing environments, the expanded selection of tools available for securing information in this environment should be used fully to ensure that the data is as secure as possible. In addition, with the continuing dependence on distributed data the security of these systems must be fully
The techniques that are ultimately used to secure commercial OODBMS implementations will depend in large part on the approaches promoted by the leading database vendors. However, the applied research that has been conducted to date is also laying the groundwork for the security components that will in turn be incorporated in the commercial OODBMSs.

James Cannady is a research scientist at Georgia Tech Research Institute. For the past seven years he has focused on developing and implementing innovative approaches to computer security in extremely sensitive networks and systems in military, law enforcement, and commercial environments.