Logical Access Control

NOTE

This file is a DRAFT chapter intended to be part of the NIST Computer Security Handbook. The chapters were prepared by different parties and, in some cases, have not been reviewed by NIST. The next iteration of a chapter could be SUBSTANTIALLY different than the current version. If you wish to provide comments on the chapters, please email them to roback@ecf.ncsl.gov or mail them to Ed Roback/Room B154, Bldg 225/NIST/Gaithersburg, MD 20899.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT

Introduction

IT systems today can process and store a wide variety of information and provide access to it to a large number of users. It is not unusual for a system in a large organization to contain some information that must be accessible to all users, some that is needed by several groups or departments, as well as some that should be accessed by only a few individuals. Having information reside centrally on a system used by everyone contributes to cost effective and efficient information sharing and processing.

Information residing on a system that is accessed by many users, however, can also create problems. A significant concern is ensuring that users have access to information that they need but do not have inappropriate access to information that is sensitive. It is also important to ensure that certain items, though readable by many users, can only be changed by a few.

Logical access controls are a means of addressing these problems. Logical access controls are protection mechanisms that limit users' access to information and restrict their forms of access on the system to only what is appropriate for them. Logical access controls are often built into the operating system, or may be part of the "logic" of applications programs or major utilities, such as Database Management Systems. They may also be implemented in add-on security packages that are installed into an operating system; such packages are available for a variety of systems, including PCs and mainframes. Additionally, logical access controls may be present in specialized components that regulate communications between computers and networks.

Some rudimentary forms of automated access controls have been available for many years, but today there are increasingly sophisticated and cost-effective methods that managers will find well worth investigating. This chapter will discuss some of the advantages provided by logical access control and issues to be considered when investigating logical access control. It will also provide an introduction to common forms of logical access control available today.

Background Information

As noted above, logical access control limits users' access to information, and it can restrict the capabilities or modes of access they have. It can therefore help promote efficiency and IT security at the same time, but there are potential drawbacks that should be weighed and considered. While logical access controls can be of great benefit to an organization, adding them to a system does not automatically make the system more secure. A poorly chosen or improperly configured control mechanism can have a detrimental effect, as can inadequate understanding of the complexities involved in implementing and managing the technology. Following is general information and background on logical access control, and an introduction to some of the associated issues.

Types of Access Restrictions

Many of the advantages as well as many of the complexities involved in implementing and managing logical access control are related to the different kinds of user accesses supported. Not only are the types of accesses allowed an important consideration, but so are the kinds of data, programs, devices, and services. Some information on the system, such as the data displayed on an organization's daily calendar of nonsensitive meetings, should be readable by literally everyone in the organization. The program that formats and displays the calendar, however, might be modifiable by only a very few IT system administrators, while the operating system controlling that program might be accessible by still fewer.

Access Modes

The concept of access modes is fundamental to logical access control. The effect of many types of logical access control is to permit or deny access by specific individuals to specific information resources in specific access modes. An introduction to the common access modes follows.

Read only: This provides users with the capability to view, copy, and usually print information but not to do anything to alter it, such as delete from, add to, or modify it in any way. Read- only accesses are probably the most widely allowed to data files on IT systems.

Read and Write: Users are allowed to view and print as well as add, delete, and modify information. Logical access control can further refine the read/write relationship such that a user has read-only ability for one field of information but the ability to write to a related field. An example would be a project Action Item program that allows a user read-only ability for the assigned action items and permits responses to be written in the space below an action item.

Execute: The most common activity performed by users in relation to applications programs on a system is to execute them. A user executes a program each time he or she uses a word processor, spreadsheet, database, etc. Users would not ordinarily be given read or write capabilities for an application, however, since it would appear in a format that is unintelligible to most users. It might be desirable, though, for software programming specialists to be able to read and write applications.

Successfully refining, implementing, and managing these different access modes have resulted in greatly improved information sharing, both for government and industry as well as for the general public. There are systems, for example, referred to as public access systems, whose purpose is to disseminate information to the public at large. The ability to read from these systems, therefore, has been made widely available. With logical access control, the crucial requirement of preserving the integrity of the information being disseminated that is, protecting it against improper modification, can be met while the information remains available for all to view.

Other Restrictions

In addition to restrictions based on access mode, logical access controls may deny or permit access based on a number of other factors.

Access may be permitted only during particular hours of the day, or only from particular terminals or network locations.

Access may be permitted or denied based on information content or numerical thresholds. For example, an ATM machine may restrict transfers of money between accounts to certain dollar limits. A supervisor may be allowed to read salary or other personnel information, but only for employees whom he or she supervises.

Access may be permitted selectively based on the type of service requested. For example, users of computers on a network may be permitted to exchange electronic mail but might not be allowed to log in to each others' computers.

Relationship to Identification & Authentication

The subject of identification and authentication (I&A) is discussed in more detail in Chapter 16. The basic relationship between I&A and logical access control is included here because I&A forms the basis for logical access control. I&A is the process by which anyone attempting to interact with a system establishes his/her identity to the system, for example, by use of a password or token. The logical access control process then associates the appropriate information and permissible forms of accesses with that identity. This means that logical access control can only be as effective as the I&A process employed for the system. If users tell one another or write down passwords, both I&A and logical access control for the system are compromised.

Relationship to Physical Access Control

Before logical access controls were widely available, physical access control was the main means of protecting information on an IT system. Access to information was controlled solely by controlling access to the system, for example, by keeping the system in a locked room or having a guard on duty to restrict admittance to a facility. Once logged onto a system, though, a user could generally access all of its data. In some environments, this is not a problem. Physical access control may be sufficient in environments where all users of a system need to access to all of the information on it and need to perform all of the same types of accesses in relation to it (read it, add to it, delete it, etc). In environments where not all information resources on a system should be equally available to all users, a more precise control is necessary.

Logical access control can enhance the security provided by physical access control by acting as an additional guard against unauthorized access to or use of the system's resources. It can also augment physical access control by providing added precision, since different users are able to perform different functions. An example would be a team of scientists who all need access to up-to-the minute information in a field of research. Everyone in the group could be given physical access to a system where the information is being posted and the ability to read all information. Senior scientists might also be able to add comments on the information, while perhaps only the head of the research effort might be able to add and delete files.

Administration of Logical Access Controls

Administration is the most complex and challenging aspect of logical access control. Administration of logical access controls involves implementing, monitoring, modifying, testing, and terminating user accesses on the system and can be a demanding task. Administration typically does not include making the actual decisions as to who may have access to what and be given which capabilities. Those decisions are usually the data owner's responsibility, perhaps made in conjunction with management. Decisions regarding accesses should be guided by organizational policy, employee job descriptions and tasks, information sensitivity, user "need to know" determinations, and many other factors. Procedures and forms for the request and approval process are also typically developed.

Regardless of how and at whose discretion the decisions on user accesses are made, implementation and management are accomplished through an administrative function. There are three basic approaches to administration: centralized, decentralized, or a combination. Each has relative advantages and disadvantages, and which is best will depend upon the needs and complexity of the particular organization.

Centralized Administration

Centralized administration means that one element (usually a group in large organizations, an individual in small ones) is responsible for configuring access controls so that users can access data and perform the activities they need to. As users' information processing needs change, their accesses can be modified only through the central administration, usually after requests have been approved through an established procedure and by the appropriate authority.

The main advantage of centralized administration is that very strict control over information can be maintained because the ability to make changes resides with a very few persons. Each user's account can be centrally monitored, and closing all accesses for any user can be easily accomplished if that individual leaves the organization. Consistent and uniform procedures and criteria are usually not difficult to enforce, since relatively few individuals oversee the process.

A major disadvantage, though, is that the change process can be constant, due to employees being hired, terminated, and reassigned. Constant changes can make the task of administration time-consuming and costly in terms of staffing and equipment. Also, when changes are needed quickly in order for users to complete important tasks, going through central administration can be time-consuming. Another problem that can arise is that permissions for access can be too limited. This can interfere with users' ability to get work done.

Decentralized Administration

In contrast to centralized administration, decentralized administration means that access to information is controlled by the owners or creators of the files, whoever or wherever those individuals may be. An advantage of decentralized administration is that control is in the hands of the individuals most accountable for the information, most familiar with it, and best able to judge who should be able to do what in relation to it. One disadvantage, however, is that there may not be consistency among owners/creators as to procedures and criteria for granting user accesses and capabilities. Another is that when requests are not processed centrally, it may be much more difficult to form a system-wide composite view of all user accesses on the system at any given time. Different data owners may inadvertently implement combinations of accesses that introduce conflicts of interest or that are in some other way not in the organization's best interest. It may also be difficult to ensure that accesses are properly terminated when an employee transfers within or leaves an organization.

Hybrid Approach

In a hybrid approach, centralized control is exercised for some information and decentralized is allowed for other information. One typical arrangement is that central administration is responsible for the broadest and most basic accesses, and the owners/creators of files control types of accesses or changes in users' abilities for the files under their control. For example, when a new employee is hired into a department, a central administrator might provide him with a set of accesses, perhaps based on the functiona l element he is assigned to, his job classification, and a specific task he was hired to work on. He might have read-only access to an organizationwide bulletinboard and to project status report files, but read and write privileges to his department's weekly activities report. Over time, was assigned to other projects, the project managers could modify his capabilities on their respective files to include the ability to write information in project files such as project status reports. Also, if he left a particular project, the project manager could close the employee's access to that file.

The main disadvantage to a hybrid approach is adequately defining which accesses should be assignable locally and which centrally.

Super Users@user: privileges]

Regardless of the type of administration chosen, the prevailing needs of adequate user access plus maintenance of IT system security need to be ensured. To contribute to meeting these needs, all logical access control schemes allow for "super user" capabilities for some individual or small group. This enables all user and administrator activities to be changed or superseded immediately when necessary. Consider the possibility that an employee with very select accesses or capabilities for data in a department is unexpectedly absent, due to a personal emergency or illness. A super user could provide someone else the same accesses and capabilities. Such emergency changes are usually governed by policy and subject to close scrutiny, to ensure limited implementation. Super users also typically have capabilities for accessing and interacting with critical system programs, such as the operating system, not accessible by others. This type of access is necessary for maintenance and upgrades.

Because super users have sufficient privileges to bypass or modify logical access controls, super user capabilities present a potential vulnerability and must be guarded carefully. Organizations should stringently minimize the number of individuals who are authorized to act as super users. Furthermore, additional I&A precautions, such as ensuring that super users' passwords are robust and changed regularly, are important to minimize opportunities for unauthorized individuals to gain super user access to the system.

Integration

Uniform enforcement of logical access control in IT systems is made more complicated because of the pervasiveness of networks and applications. No longer is a single operating system responsible for enforcing all access control decisions. Many applications or utilities run by the operating system, such as Database Management Systems (DBMS), also enforce logical access control, but at a different level than the operating system. The degree to which the logical access control performed by an operating system and that performed by an application are integrated can vary significantly. It is important in any event that they do not conflict.

Returning to the example of a DBMS will provide an illustration. A DBMS manages a collection of information called a database. The DBMS is responsible for controlling who can access the data in the database. Databases are frequently stored in files, and operating systems are responsible for enforcing protection on files. In order for the DBMS logical access control to be effective, the underlying operating system has to ensure that no user or program other than the DBMS can access the database. This is a minimal, but necessary, form of logical access control integration between an operating system and a DBMS.

Integration issues also arise in a network environment. Instead of coordinating access control decisions between the operating system and applications on one host, coordination needs to take place across a collection of hosts. It is generally considered desirable for information to be protected in a uniform manner, regardless of the particular location where it is stored. This requires coordination among the administrators of the various hosts comprising an organization's IT system and comparable access control mechanisms on each host.

When Logical Access Control Is Not Necessary

While logical access control can greatly increase the flexibility and ease with which information can be shared on an IT system, it is not always necessary. As noted earlier, logical access controls are best suited for situations where multiple users of a system should not all have the same form of access to all of the information on the system. A personal computer used solely by one person, for example, does not necessarily need logical access control, nor does a multi- user system in an environment where all users should have access to all of the data and have all of the same forms of accesses.

There are also environments where logical access control would be appropriate and beneficial but may not be cost effective. Logical access control might be quite useful, for example, to a small company for tightly restricting access to personnel salary information, if that data were stored on a multi-user system. However, the costs of the technology and administration might be higher than the cost and operational impact of keeping the salary data on a separate, isolated system within a locked office.

A small group of users dedicated to single task often indicates lack of a need for logical access control. Consider, for example, a four person technical publications group that is drafting the manual for a software product. They share a single IT system, but logical access controls may not be utilized because all of the users need to be able to access and interact with the manual as it is being written. With such a small number of users, simply scheduling assignments so that only one person is working on a given section at a time might suffice to keep team members from interfering with one another's work.

Even in circumstances where logical access control is not necessary, it may still be beneficial for preventing inadvertent errors or deletions. On the single-user PC noted above, for example, restricting access to the operating system or to very critical functions for purposes of ensuring integrity can be highly desirable. Whether or not logical access control will be worth the investment will depend on how much benefit will be derived from the expenditure.

Mechanisms

Many mechanisms have been developed to provide logical access control on IT systems, and they vary significantly in terms of precision, sophistication, and cost. This section will provide an overview of some of the methods. It should be noted that these methods are not mutually exclusive and that many systems employ a combination. Managers need to analyze their organization's information processing needs and their information's sensitivity and criticality in order to decide what is the optimal method or combination of methods.

Passwords/Keys/Tokens

Passwords are probably the most common way of protecting information on an IT system in that they are the most frequently used means for users to be identified and authenticated on the system. Thus, they are often the first line of protection afforded an IT system. In addition, passwords are also used to protect data and applications on many IT systems. Passwords are also used frequently in PC applications as a means of logical access control. For instance, an accounting application may require a password in order to access certain financial data or invoke a sensitive application.

The primary advantage of password-based logical access control is that it is provided by a large variety of PC applications and thus often does not have to be implemented as a new/separate feature on an operating system. The drawbacks of this approach center on the difficulty for users to manage even moderate numbers of passwords. As discussed in the Identification and Authentication chapter, the security of a password-based system is significantly diminished when users write down their passwords. If users need to use more than a few different passwords in the course of their work, there will be a strong likelihood that they will write them down, thus exposing the IT resources the passwords were meant to protect. Also, if passwords are the same for several different applications, then a user who learns the password for one can gain access to the others.

Encryption can also be used as a means of logical access control. Information of a certain type can be encrypted with a particular key, and possession of that key would entitle a user to access that information. Encrypting financial data from a previous year to protect it from improper modification can be part of the process of "closing the books." Tokens, as discussed in the Identification and Authentication chapter, act as an alternative for passwords or keys.

Permission Bits

Permission bits are now a widely available means of providing logical access control on multi-user IT systems. In this scheme, access rights to objects are based on the concepts of owner, group, and world; for each of these, a set of access modes (typically chosen from read, write, and execute) is specified. The owner of an object, such as a file, is typically its creator, though in some cases system or project administrators may be automatically assigned ownership of all objects regardless of who created them. The owner of an object can specify the allowed modes of access to the object.

Each object is also associated with a named group of users. Users who are members of the group associated with an object can be granted modes of access distinct from non-members, who belong to the rest of the "world" that includes all of the IT system's users. Typically user groups are arranged according to department, project, or other teaming relationships. For example, groups may be established for members of the Personnel and Accounting departments. Changing the membership of a group typically requires action by a system administrator.

As an example of the use of permission bits, consider a file that contains a personnel appraisal report. The permission bits could be set by the report's owner such that it was readable and writable by the report's owner, readable by the Personnel group, but neither writable nor readable by the rest of the organization's users.

In a system employing permission bits, access to a file is at the discretion of the file's owner. This method of access control can be quite useful in a project-oriented environment and one in which there are relatively few organizationwide restrictions for information-sharing. There are some aspects of access restriction, however, that cannot be represented using permission bits, such as explicitly denying access to an individual that is a member of the file's group. Additionally, as is the case with Access Control Lists (discussed in the next section), permission bits can not guarantee that the contents of a file will not be disclosed or modified by an unauthorized user. For example, a member of a file's group could copy the file and then set the copy's permission bits to allow world read access.

Access Control Lists

Access Control Lists (ACLs) are similar to permission bits in that they provide a form of logical access control that is at the discretion of the information's owner. They do, however, provide finer precision in control. An ACL is associated with each file and specifies by name each user or group who can access the object and the type of access they are permitted. By way of example, consider a medical research experiment. The file containing experimental results could have an ACL that permitted read and write access by all the members of the research group. There could then be an additional ACL that prohibited any access by one member of the group who was responsible for conducting another experiment whose results should not be influenced by the results of the first. While the independence of the two experiments relies primarily on the researchers refraining from exchanging information via discussion, the ACL reduces the chance that independence will be compromised by snooping or inadvertent browsing of files. ACLs, however, like permission bits, can be defeated if an authorized individual copies sensitive information to another object whose ACL provides fewer access restrictions.

ACLs provide a fine grained form of logical access control that can be useful for complex information sharing situations. The flexibility provided by ACLs also makes them more of a challenge to manage. The rules for determining access in the face of apparently conflicting ACL entries are not uniform across all implementations and can be confusing to users. If such a system is introduced, it should be coupled with training to ensure that it is used correctly.

Labels

For IT systems with stringent security requirements, such as those associated with national security, labels are often used as the basis for logical access control. Systems employing labels associate an unchangeable label with each file that indicates its sensitivity. Similarly, user sessions are assigned labels that designate the degree to which access to information at different sensitivities is granted. In addition, users are authorized to initiate sessions with specific labels only. For example, a file bearing the label Organization Proprietary Information would not be accessible (readable) except during user sessions with the corresponding label. Moreover, only a restricted set of users would be able to initiate such sessions; other users would be allowed to initiate sessions at lower sensitivity levels only, and would consequently have access only to less sensitive information.

Labels are a robust form of logical access control. Unlike permission bits or access control lists, labels cannot ordinarily (e.g., accidentally) be changed, and labels for new files are automatically determined by the access control mechanism. By removing users' ability to arbitrarily designate the accessibility of files they own, opportunities for certain kinds of human errors and malicious software problems are eliminated. In the example above, it would not be possible routinely to copy Organization Proprietary Information into a file with a less sensitive label. This prevents inappropriate "leakage," but it may also interfere with legitimate extraction of less sensitive information. Label-based access controls may also be used to prevent low integrity information from leaking into and contaminating high integrity information.

Labels are well-suited for consistently and uniformly enforcing organization-wide access restrictions, sometimes called system security policies. For this reason, label-based controls can provide a level of protection not found in other approaches. Presently, labels are in relatively limited use. As more operating systems that provide labels become available, though, access controls based on labels may become more familiar and attractive to larger user populations.

Roles

A role is a job assignment or function. Examples of roles include data entry clerk, purchase officer, project leader, programmer, technical editor, etc. Logical access controls can support user roles on the IT resource. This means allowing access rights to be grouped by role name, and restricting use of those access rights to individuals authorized to assume the associated role. An individual may be authorized for more than one role, but may be required to act in a single role at a time. Changing roles may require logging out and then in again, or entry of a special role-changing command.

Many IT systems already support a small number of special purpose roles, such as System Administrator or Operator. An individual who is logged on in the role of a System Administrator can, for example, perform operations that would be denied to the same individual acting in the role of an ordinary user. Recently, the use of roles has been expanded beyond system tasks to application oriented activities. For example, in a company there could be an Order Taking Role. A user with this role would be able to collect and enter customer billing information, check on availability of particular items, request shipment of items, and issue invoices. In addition, there could be an Accounts Receivable Role which would receive payments and credit them to particular invoices. A third, Shipping Role, could then be responsible for shipping products and updating the inventory. To provide additional security, constraints could be imposed such that a single individual user would never be simultaneously authorized to assume all three roles. Constraints of this kind are sometimes referred to as separation of duty constraints.

The use of roles and the corresponding concept of a business transaction can be a very effective way of providing logical access control. The process of defining roles and their relationships should be based on a thorough analysis of the way in which an organization operates and should include input from a wide spectrum of users in an organization. Standardization of role-based access control systems, as is being done for some database management systems, will make the adoption of role-based logical access control easier. The user group mechanism described in the discussion of permission bits can in some cases support roles, but at present, more explicit support for application oriented roles in commercial operating systems is limited.

Constrained User Interfaces

The principle underlying constrained user interfaces is that a user should be able to access system functions for which he/she is specifically authorized. Menu driven systems are a common paradigm for constrained user interfaces, the implementation being that different users are provided different menus for the same system. A user is not given menu options for unauthorized operations and so has no means by which to invoke them. A common example of a constrained user interface is an Automated Teller Machine (ATM). An ATM presents a user with a limited list of permitted operations. The user is prevented from escaping to any other system interface and so is prevented from bypassing the logical access controls.

With an ATM machine the menu options permit a user to undertake a number of transactions, e.g., deposit, withdrawal, transfer. There is a hierarchy of menus that support these transactions. In other IT systems, a menu-based constrained user interface can similarly provide a hierarchy of menus to support arbitrarily complex transactions.

As is the case with roles, constrained user interfaces can provide a form of logical access control that closely models the way in which an organization operates. The use of menus also makes this an approach that will be easy for non-technical users to understand. The primary drawback to this approach is the cost associated with tailoring such a system to an organization.

Interdependencies

Policy

The most fundamental interdependency of logical access control is with policy. Control is performed by the system, but the decisions as to accesses are made and enforced at the discretion of individuals who must act in concert with the organization's IT security policy. Policy should specify who authorizes access to what kinds of information and provide the criteria for making access control decisions.

Audit

It is sometimes not possible to make logical access control as precise, or fine-grained, as would be ideal for an organization. Given the difficulty of configuring logical access controls in a complex IT system, there are may be occasions when a user is inadvertently allowed access to resources he should not have. In some cases, users will be granted access in case they need to act in someone's place. In addition, the policy or rules governing access may change over time, and there is a window of time between when the policy changes and when the logical access control system is updated. The net result in these cases is that it is possible for users to abuse access permissions they have. Automated auditing provides a source of information that can be used to identify users who have abused their access permissions. Audit analysis can perform such functions as checking accesses to very sensitive or critical resources, the membership of very powerful groups, verifying the consistency of rights with roles, and generating access violation reports.

Identification & Authentication

In most logical access control scenarios, the identity of the user must be established before an access control decision can be made. This is especially true with the permission bit and ACL methods. Establishing the identity of users is a necessary prerequisite for enforcing logical access control.

Data Categorization

Just as the identity of users plays a role in determining access, so does a characterization of the information being protected. At one end of the spectrum, labels are a direct representation of a data categorization and are the basis of a logical access control method. Even in the other access control methods discussed above, data categorization plays a role. For example, recall the medical experiment in which the results had a specific categorization that required additional access protection.

Assurance

By its very nature, logical access control is normally a critical component of the security provided by a system. If an IT system's logical access control does not function correctly, is not configured properly, or is not effective for the application, serious harm to the organization could result. Even in situations in which there are limited resources to provide assurance for a system, it is important to that they be directed in part towards assuring the proper functioning of the logical access control system.

Costs

Incorporating logical access control into an IT system involves both the purchase or utilization of access control mechanisms as well as a change in behavior on the part of users.

Direct Costs

Among the direct costs associated with the use of logical access control methods are the purchase and support of hardware, operating systems and applications that provide the controls, and any add-on security packages necessary or desirable. The most significant personnel cost in relation to logical access control is usually for administration. Most multi-user operating systems provide some protection mechanism such as permission bits or ACLs, so there is less acquisition cost associated with these. Support for label-based access control is available in a limited number of commercial products, but at greater cost and with less selection than for permission bits or ACLs. Role-based systems are becoming more available with time, but there is the cost of customizing these systems for particular organizational purposes. Training users to understand and use a logical access control system is a very necessary cost. If users are not comfortable in using an access control system they will attempt to configure it so that it places few or no restrictions. This may provide the organization with false confidence in the security ofits IT resources, resulting in a security situation worse than if the protection mechanisms had not been provided in the first place.

Indirect Costs

The primary indirect cost associated with introducing logical access controls into an IT system is the effect on user productivity. There are two primary dimensions to this situation. The first is the additional overhead individual users have in properly determining (when it is under their control) the protection attributes of information. This determination requires both an understanding of the relevant policy governing the treatment of the information and an understanding of the technology supporting the logical access control. The other dimension centers on the situation of users not being able to access information necessary to their jobs because the permissions were incorrectly assigned. While infrequent, this situation is familiar to most organizations that put strong emphasis on logical access control.

It is important to understand, though, that through the proliferation of PCs, the decreased costs of computers, and increased use of networking, the amount and variety of information processed on shared IT systems is increasing at a rapid rate. Without the assurance provided by logical access control that information will be protected appropriately, there will be a reluctance to share that information in the most effective manner.

The result would then be a decrease in the usefulness of an IT system.

SIDEBAR NOTES:

- (1) Sec 1 para 3: Logical Access Controls are a means of controlling the types of information different users of the same system may access.
- (2) Sec 2.1.1 para 1: Logical Access Controls manage interactions among different users, different types of information, and different types of access modes.
- (3) Sec 2.2: Identification & Authentication, covered in Chapter 16, forms the basis for logical access control.
- (4) Sec 2.3 para 2: Logical access control can augment physical access control.
- (5) Sec 3 para 1: Administration is one of the most challenging aspects of logical access control.
- (6) Sec 3.1 para 1: Central administration means that one element in the organization is responsible for configuring all user access controls.
- (7) Sec 3.2: Decentralized administration means that accesses are controlled by the owners or creators of files.
- (8) Sec 3.4: "Super users" can change or supersede all user and administrator activities, when necessary, but such privileges must be monitored stringently.
- (9) Sec. 5, para 2: In some environments, although logical access controls would be beneficial, the costs might be prohibitive.
- (10) Sec. 6: A variety of logical access control mechanisms are available, and they vary in terms of precision, sophistication, and cost.
- (11) Sec. 6.3, para 1: Permission bits and Access Control Lists provide logical access control that is at the discretion of the information's owner, but ACLSs provide finer precision.

(12) Sec. 6.5, para 1: Logical access control through roles means that rights are grouped by role name and access rights are restricted to persons authorized to assume the associated role.

REFERENCES:

Caelli, William, et al. Information Security Handbook. Stockton Press, 1991, New York, NY.

Abrams, M.D., et al. A Generalized Framework for Access Control:

an Informal Description. Mitre Corporation: McLean, VA, 1990.

Baldwin, R.W. "Naming and Grouping Privileges to Simplify Security Management in Large Databases." In Proc. 1990 IEEE Symposium on Security and Privacy, pages 116-132, Oakland, CA, May 1990.

Dinkel, Charles. Secure Data Network System Access Control Documents. National Institute of Standards and Technology:

Gaithersberg, MD, 1990.

Thomsen, D.J. "Role-based Application Design and Enforcement." In Proc. of the Fourth IFIP Workshop on Database Security, Halifax, England, September 1990.

Pfleeger, Charles P. Security In Computing. Prentice-Hall, Inc.:

Englewood Cliffs, NJ, 1989.

Gasser, Morrie. Building a Secure Computer System. Van Nostrand Reinhold Company, Inc.: New York, NY, 1988.

Sandhu, R. "Transaction Control Expressions for Separation of Duty." In Fourth Annual Computer Security Applications Conference, pages 282-286, Orlando, FL, December 1988.

Clark,D. and D. Wilson. "A Comparison of Commercial and Military Computer Security Policies." In Proc. 1987 IEEE Symposium on Security and Privacy, pages 184-194, Oakland, CA, April 1987.

Bach, M.J. The Design of the Unix Operating System. Prentice-Hall, Englewood Cliffs, NJ, 1986.

Boebert, W. E. and R. Y. Kain. "A Practical Alternative to Hierarchical Integrity Policies." In Proc. 8th National Computer Security Conference, pages 18-27, Gaithersburg, MD, September 1985.

Landwehr, C., C. Heitmeyer and J. McLean. "A Security Model for Military Message Systems." In ACM Transactions on Computer Systems, Vol 2, No.3, August 1984.

"Guideline on User Authentication Techniques for Computer Network Access Control," U.S. Department of Commerce (NIST), FIPS Publication 83, September 1980.

"Guidelines for Security of Computer Applications," U.S. Department of Commerce (NIST), FIPS Publication 73, June 1980.22