Chapter 17

LOGICAL ACCESS CONTROL

Logical access controls provide a technical means of controlling what information users can utilize, the programs they can run, and the modifications they can make.

On many multiuser systems, requirements for using (and prohibitions against the use of) various computer resources[114] vary considerably. Typically, for example, some information must be accessible to all users,[115] some may be needed by several groups or departments, and some should be accessed by only a few individuals. While it is obvious that users must have access to the information they need to do their jobs, it may also be required to deny access to non-job-related information. It may also be important to control the *kind of access* that is afforded (e.g., the ability for the average user to execute, but not change, system programs). These types of access restrictions enforce policy and help ensure that unauthorized actions are not taken.

The term *access* is often confused with *authorization* and *authentication*.

Access is the *ability* to do something with a computer resource. This usually refers to a technical ability (e.g., read, create, modify, or delete a file, execute a program, or use an external connection).

Authorization is the *permission* to use a computer resource. Permission is granted, directly or indirectly, by the application or system owner.

Authentication is proving (to some reasonable degree) that users are who they claim to be.

Access is the ability to do something with a computer resource (e.g., use, change, or view). *Access control* is the means by which the ability is explicitly enabled or restricted in some way (usually through physical and system-based controls). Computer-based access controls are called *logical access controls*. Logical access controls can prescribe not only who or what (e.g., in the case of a process) is to have access to a specific system resource but also the type of access that is permitted. These controls may be built into the operating system, may be incorporated into applications programs or major utilities (e.g., database management systems or communications systems), or may be implemented through add-on security packages. Logical access controls may be implemented internally to the computer system being protected or may be implemented in external devices.

[114] The term *computer resources* includes information as well as system resources, such as programs, subroutines, and hardware (e.g., modems, communications lines).

[115] *Users* need not be actual human users. They could include, for example, a program or another computer requesting use of a system resource.

Logical access controls can help protect:

- operating systems and other system software from unauthorized modification or manipulation (and thereby help ensure the system's integrity and availability);
- the integrity and availability of information by restricting the number of users and processes with access; and
- confidential information from being disclosed to unauthorized individuals.

Controlling access is normally thought of as applying to human users (e.g., will technical access be provided for user JSMITH to the file "payroll.dat") but access can be provided to other computer systems. Also, access controls are often incorrectly thought of as only applying to *files*. However, they also protect other system resources such as the ability to place an outgoing long-distance phone call though a system modem (as well as, perhaps, the information that can be sent over such a call). Access controls can also apply to specific functions within an application and to specific fields of a file.

This chapter first discusses basic criteria that can be used to decide whether a particular user should be granted access to a particular system resource. It then reviews the use of these criteria by those who set policy (usually system-specific policy), commonly used *technical mechanisms* for implementing logical access control, and issues related to administration of access controls.

17.1 Access Criteria

In deciding whether to permit someone to use a system resource logical access controls examine whether *the user is authorized for the type of access requested.* (Note that this inquiry is usually distinct from the question of whether the user is authorized to use the system *at all*, which is usually addressed in an identification and authentication process.)

When determining what kind of technical access to allow to specific data, programs, devices, and resources, it is important to consider who will have access and what kind of access they will be allowed. It may be desirable for everyone in the organization to have access to some information on the system, such as the data displayed on an organization's daily calendar of nonconfidential meetings. The program that formats and displays the calendar, however, might be modifiable by only a very few system administrators, while the operating system controlling that program might be directly accessible by still fewer.

The system uses various criteria to determine if a request for access will be granted. They are typically used in some combination. Many of the advantages and complexities involved in implementing and managing access control are related to the different kinds of user accesses supported.

17.1.1 Identity

It is probably fair to say that the majority of access controls are based upon the identity of the user

(either human or process), which is usually obtained through identification and authentication (I&A). (See Chapter 16.) The identity is usually unique, to support individual accountability, but can be a group identification or can even be anonymous. For example, public information dissemination systems may serve a large group called "researchers" in which the individual researchers are not known.

17.1.2 Roles

Access to information may also be controlled by the job assignment or function (i.e., the *role*) of the user who is seeking access. Examples of roles include data entry clerk, purchase officer, project leader, programmer, and technical editor. Access rights are grouped by role name, and the use of resources is restricted 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 only a single role at a time. Changing roles may require logging out and then in again, or entering a role-changing command. Note that use of roles is *not* the same as shared-use accounts. An individual may be assigned a standard set of rights of a shipping department data entry clerk, for example, but the account would still be tied to that individual's identity to allow for auditing. (See Chapter 18.)

Many systems already support a small number of special-purpose roles, such as System Administrator or Operator. For example, an individual who is logged on in the role of a System Administrator can 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, a user in a company could have an Order Taking role, and 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 Shipping role, could then be responsible for shipping products and updating the inventory. To provide additional security, constraints could be imposed so a single 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 can be a very effective way of providing access control. The process of defining roles should be based on a thorough analysis of how an organization operates and should include input from a wide spectrum of users in an organization.

17.1.3 Location

Access to particular system resources may also be based upon physical or logical location. For example, in a prison, all users in areas to which prisoners are physically permitted may be limited to read-only access. Changing or deleting is limited to areas to which prisoners are denied physical access. The same authorized users (e.g., prison guards) would operate under significantly different logical access controls, depending upon their physical location. Similarly, users can be restricted based upon network addresses (e.g., users from sites within a given organization may be permitted greater access than those from outside).

17.1.4 Time

Time-of-day or day-of-week restrictions are common limitations on access. For example, use of confidential personnel files may be allowed only during normal working hours – and maybe denied before 8:00 a.m. and after 6:00 p.m. and all day during weekends and holidays.

17.1.5 Transaction

Another approach to access control can be used by organizations handling transactions (e.g., account inquiries). Phone calls may first be answered by a computer that requests that callers key in their account number and perhaps a PIN. Some routine transactions can then be made directly, but more complex ones may require human intervention. In such cases, the computer, which already knows the account number, can grant a clerk, for example, access to a particular account *for the duration of the transaction*. When completed, the access authorization is terminated. This means that users have no choice in which accounts they have access to, and can reduce the potential for mischief. It also eliminates employee browsing of accounts (e.g., those of celebrities or their neighbors) and can thereby heighten privacy.

17.1.6 Service Constraints

Service constraints refer to those restrictions that depend upon the parameters that may arise during use of the application or that are preestablished by the resource owner/manager. For example, a particular software package may only be licensed by the organization for five users at a time. Access would be denied for a sixth user, even if the user were otherwise authorized to use the application. Another type of service constraint is based upon application content or numerical thresholds. For example, an ATM machine may restrict transfers of money between accounts to certain dollar limits or may limit maximum ATM withdrawals to $500 per day. Access may also be selectively permitted based on the type of service requested. For example, users of computers on a network may be permitted to exchange electronic mail but may not be allowed to log in to each others' computers.

17.1.7 Common Access Modes

In addition to considering criteria for *when* access should occur, it is also necessary to consider the *types* of access, or *access modes*. The concept of access modes is fundamental to access control. Common access modes, which can be used in both operating or application systems, include the following:[116]

---

[116] These access modes are described generically; exact definitions and capabilities will vary from implementation to implementation. Readers are advised to consult their system and application documentation.

Read access provides users with the capability to view information in a system resource (such as a file, certain records, certain fields, or some combination thereof), but not to *alter* it, such as delete from, add to, or modify in any way. One must assume that information can be copied and printed if it can be read (although perhaps only manually, such as by using a print screen function and retyping the information into another file).

Write access allows users to add to, modify, or delete information in system resources (e.g., files, records, programs). Normally user have read access to anything they have write access to.

Execute privilege allows users to run programs.

Delete access allows users to erase system resources (e.g., files, records, fields, programs).[117] Note that if users have write access but not delete access, they could overwrite the field or file with gibberish or otherwise inaccurate information and, in effect, delete the information.

Other specialized access modes (more often found in applications) include:

Create access allows users to create new files, records, or fields.

Search access allows users to list the files in a directory.

Of course, these criteria can be used in conjunction with one another. For example, an organization may give authorized individuals write access to an application at any time from within the office but only read access during normal working hours if they dial-in.

Depending upon the technical mechanisms available to implement logical access control, a wide variety of access permissions and restrictions are possible. No discussion can present all possibilities.

17.2 Policy: The Impetus for Access Controls

Logical access controls are a technical means of implementing *policy decisions*. Policy is made by a management official responsible for a particular system, application, subsystem, or group of systems. The development of an access control policy may not be an easy endeavor. It requires balancing the often-competing interests of security, operational requirements, and user-friendliness. In addition, technical constraints have to be considered.

---

[117] "Deleting" information does not necessarily physically remove the data from the storage media. This can have serious implications for information that must be kept confidential. See "Disposition of Sensitive Automated Information," CSL Bulletin, NIST, October 1992.

This chapter discusses issues relating to the technical implementation of logical access controls – not the actual policy decisions as to who *should* have what type of access. These decisions are typically included in system-specific policy, as discussed in Chapters 5 and 10.

Once these policy decisions have been made, they will be *implemented* (or *enforced*) through logical access controls. In doing so, it is important to realize that the capabilities of various types of technical mechanisms (for logical access control) vary greatly.[118]

A few *simple* examples of *specific policy issues* are provided below; it is important to recognize, however, that *comprehensive system-specific policy* is significantly more complex.

1. The director of an organization's personnel office could decide that all clerks can update all files, to increase the efficiency of the office. Or the director could decide that clerks can only view and update specific files, to help prevent information browsing.

2. In a disbursing office, a single individual is usually prohibited from both requesting and authorizing that a particular payment be made. This is a *policy decision* taken to reduce the likelihood of embezzlement and fraud.

3. Decisions may also be made regarding access to the system itself. In the government, for example, the senior information resources management official may decide that agency systems that process information protected by the Privacy Act may not be used to process public-access database applications.

17.3 Technical Implementation Mechanisms

Many mechanisms have been developed to provide internal and external access controls, and they vary significantly in terms of precision, sophistication, and cost. These methods are not mutually exclusive and are often employed in combination. Managers need to analyze their organization's protection requirements to select the most appropriate, cost-effective logical access controls.

17.3.1 Internal Access Controls

Internal access controls are a logical means of separating what defined users (or user groups) can or cannot do with system resources. Five methods of internal access control are discussed in this section: passwords, encryption, access control lists, constrained user interfaces, and labels.

17.3.1.1 Passwords

Passwords are most often associated with user authentication. (See Chapter 16.) However, they are also used to protect data and applications on many systems, including PCs. For instance, an accounting application may require a password to access certain financial data or to invoke a

[118] Some policies may not be technically implementable; appropriate technical controls may simply not exist.

restricted application (or function of an application).[119]

The use of passwords as a means of access control can result in a proliferation of passwords that can reduce overall security.

Password-based access control is often inexpensive because it is already included in a large variety of applications. However, users may find it difficult to remember additional application passwords, which, if written down or poorly chosen, can lead to their compromise. Password-based access controls for PC applications are often easy to circumvent if the user has access to the operating system (and knowledge of what to do). As discussed in Chapter 16, there are other disadvantages to using passwords.

17.3.1.2 Encryption

Another mechanism that can be used for logical access control is encryption. Encrypted information can only be decrypted by those possessing the appropriate cryptographic key. This is especially useful if strong physical access controls cannot be provided, such as for laptops or floppy diskettes. Thus, for example, if information is encrypted on a laptop computer, and the laptop is stolen, the information cannot be accessed. While encryption can provide strong access control, it is accompanied by the need for strong key management. Use of encryption may also affect availability. For example, lost or stolen keys or read/write errors may prevent the decryption of the information. (See the cryptography chapter.)

17.3.1.3 Access Control Lists

Access Control Lists (ACLs) refer to a register of: (1) users (including groups, machines, processes) who have been given permission to use a particular system resource, and (2) the types of access they have been permitted.

ACLs vary considerably in their capability and flexibility. Some only allow specifications for certain pre-set groups (e.g., owner, group, and world) while more advanced ACLs allow much more flexibility, such as *user-defined* groups. Also, more advanced ACLs can be used to explicitly *deny* access to a particular individual or group. With more advanced ACLs, access can be at the discretion of the policymaker (and implemented by the security administrator) or individual user, depending upon how the controls are technically implemented.

Elementary ACLs. Elementary ACLs (e.g., "permission bits") are a widely available means of providing access control on multiuser systems. In this scheme, a short, predefined list of the access rights to files or other system resources is maintained.

---

[119] Note that this password is normally *in addition* to the one supplied initially to log onto the system.

Elementary ACLs are typically based on the concepts of *owner, group,* and *world*. For each of these, a set of access modes (typically chosen from read, write, execute, and delete) is specified by the owner (or custodian) of the resource. The owner is usually its creator, though in some cases, ownership of resources may be automatically assigned to project administrators, regardless of the identity of the creator. File owners often have all privileges for their resources.

Example of Elementary ACL for the file "payroll":

Owner: PAYMANAGER
Access: Read, Write, Execute, Delete

Group: COMPENSATION-OFFICE
Access: Read, Write, Execute, Delete

"World"
Access: None

In addition to the privileges assigned to the owner, each resource is associated with *a named group of users*. Users who are members of the group can be granted modes of access distinct from nonmembers, who belong to the rest of the "world" that includes all of the system's users. User groups may be arranged according to departments, projects, or other ways appropriate for the particular organization. For example, groups may be established for members of the Personnel and Accounting departments. The system administrator is normally responsible for technically maintaining and changing the membership of a group, based upon input from the owners/custodians of the particular resources to which the groups may be granted access.

As the name implies, however, the technology is not particularly flexible. It may not be possible to explicitly deny access to an individual who is a member of the file's group. Also, it may not be possible for two groups to easily share information (without exposing it to the "world"), since the list is predefined to only include one group. If two groups wish to share information, an owner may make the file available to be read by "world." This may disclose information that should be restricted. Unfortunately, elementary ACLs have no mechanism to easily permit such sharing.

Since one would presume that no one would have access without being granted access, why would it be desirable to explicitly deny access? Consider a situation in which a group name has already been established for 50 employees. If it were desired to exclude five of the individuals from that group, it would be easier for the access control administrator to simply grant access to that group and take it away from the five rather than grant access to 45 people. Or, consider the case of a complex application in which many groups of users are defined. It may be desired, for some reason, to prohibit Ms. *X* from generating a particular report (perhaps she is under investigation). In a situation in which group names are used (and perhaps modified by others), this explicit denial may be a safety check to restrict Ms. *X*'s access – in case someone were to redefine a group (with access to the report generation function) to include Ms. *X*. She would still be denied access.

Advanced ACLs. Like elementary ACLs, advanced ACLs provide a form of access control based upon a logical registry. They do, however, provide *finer precision* in control.

Advanced ACLs can be very useful in many complex information sharing situations. They provide a great deal of flexibility in implementing system-specific policy and allow for customization to meet the security requirements of functional managers. Their flexibility 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 security administrators. When such systems are introduced, they should be coupled with training to ensure their correct use.

Example of Advanced ACL for the file "payroll"

PAYMGR:	R,	W,	E,	D
J. Anderson:	R,	W,	E,	-
L. Carnahan:	-,	-,	-,	-
B. Guttman:	R,	W,	E,	-
E. Roback:	R,	W,	E,	-
H. Smith:	R,	-,	-,	-
PAY-OFFICE:	R,	-,	-,	-
WORLD:	-,	-,	-,	-

17.3.1.4 Constrained User Interfaces

Often used in conjunction with ACLs are *constrained user interfaces*, which restrict users' access to specific functions by never allowing them to request the use of information, functions, or other specific system resources for which they do not have access. Three major types exist: (1) *menus*, (2) *database views*, and (3) *physically constrained user interfaces.*

Menu-driven systems are a common constrained user interface, where different users are provided different menus on the same system.

Constrained user interfaces can provide a form of access control that closely models how an organization operates. Many systems allow administrators to restrict users' ability to use the operating system or application system directly. Users can only execute commands that are provided by the administrator, typically in the form of a *menu*. Another means of restricting users is through restricted *shells* which limit the system commands the user can invoke. The use of menus and shells can often make the system easier to use and can help reduce errors.

Database views is a mechanism for restricting user access to data contained in a database. It may be necessary to allow a user to access a database, but that user may not need access to all the data in the database (e.g., not all fields of a record nor all records in the database). Views can be used to enforce complex access requirements that are often needed in database situations, such as those based on the content of a field. For example, consider the situation where clerks maintain personnel records in a database. Clerks are assigned a range of clients based upon last name (e.g., A-C, D-G). Instead of granting a user access to all records, the view can grant the user access to the record based upon the first letter of the last name field.

Physically constrained user interfaces can also limit a user's abilities. A common example is an ATM machine, which provides only a limited number of physical buttons to select options; no

alphabetic keyboard is usually present.

17.3.1.5 Security Labels

A security label is a designation assigned to a resource (such as a file). Labels can be used for a variety of purposes, including controlling access, specifying protective measures, or indicating additional handling instructions. In many implementations, once this designator has been set, it cannot be changed (except perhaps under carefully controlled conditions that are subject to auditing).

Data Categorization

One tool that is used to increase the ease of security labelling is categorizing data by similar protection requirements. For example, a label could be developed for "organization proprietary data." This label would mark information that can be disclosed only to the organization's employees. Another label, "public data" could be used to mark information that is available to anyone.

When used for access control, labels are also assigned to *user sessions*. Users are permitted 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. The labels of the session and those of the files accessed during the session are used, in turn, to label output from the session. This ensures that information is uniformly protected throughout its life on the system.

For systems with stringent security requirements (such as those processing national security information), labels may be useful in access control.

Labels are a very strong form of access control; however, they are often inflexible and can be expensive to administer. Unlike permission bits or access control lists, labels cannot ordinarily be changed. Since labels are permanently linked to specific information, data cannot be disclosed by a user copying information and changing the access to that file so that the information is more accessible than the original owner intended. 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 to copy Organization Proprietary Information into a file with a different label. This prevents inappropriate disclosure, but can interfere with legitimate extraction of some information.

Labels are well suited for consistently and uniformly enforcing access restrictions, although their administration and inflexibility can be a significant deterrent to their use.

17.3.2 External Access Controls

External access controls are a means of controlling interactions between the system and outside people, systems, and services. External access controls use a wide variety of methods, often including a separate physical device (e.g., a computer) that is between the system being protected and a network.

One of the most common PPDs is the *dial-back modem*. A typical dial-back modem sequence follows: a user calls the dial-back modem and enters a password. The modem hangs up on the user and performs a table lookup for the password provided. If the password is found, the modem places a return call to the user (at a previously specified number) to initiate the session. The return call itself also helps to protect against the use of lost or compromised accounts. This is, however, not always the case. Malicious hackers can use such advance functions as call forwarding to reroute calls.

17.3.2.1 Port Protection Devices

Fitted to a communications port of a host computer, a port protection device (PPD) authorizes access to the port itself, prior to and independent of the computer's own access control functions. A PPD can be a separate device in the communications stream,[120] or it may be incorporated into a communications device (e.g., a modem). PPDs typically require a separate authenticator, such as a password, in order to access the communications port.

17.3.2.2 Secure Gateways/Firewalls

Often called *firewalls*, secure gateways block or filter access between two networks, often between a private[121] network and a larger, more public network such as the Internet, which attract malicious hackers. Secure gateways allow internal users to connect to external networks and at the same time prevent malicious hackers from compromising the internal systems.[122]

Some secure gateways are set up to allow all traffic to pass through except for specific traffic which has known or suspected vulnerabilities or security problems, such as remote log-in services. Other secure gateways are set up to disallow all traffic except for specific types, such as e-mail. Some secure gateways can make access-control decisions based on the location of the requester. There are several technical approaches and mechanisms used to support secure gateways.

---

[120] Typically PPDs are found only in serial communications streams.

[121] *Private network* is somewhat of a misnomer. *Private* does not mean that the organization's network is totally inaccessible to outsiders or prohibits use of the outside network from insiders (or the network would be disconnected). It also does not mean that all the information on the network requires confidentiality protection. It does mean that a network (or part of a network) is, in some way, separated from another network.

[122] Questions frequently arise as to whether secure gateways help prevent the spread of viruses. In general, having a gateway scan transmitted files for viruses requires more system overhead than is practical, especially since the scanning would have to handle many different file formats. However, secure gateways may reduce the spread of network worms.

Because gateways provide security by restricting services or traffic, they can affect a system's usage. For this reason, firewall experts always emphasize the need for policy, so that appropriate officials decide how the organization will balance operational needs and security.

Types of Secure Gateways

There are many types of secure gateways. Some of the most common are packet filtering (or screening) routers, proxy hosts, bastion hosts, dual-homed gateways, and screened-host gateways.

In addition to reducing the risks from malicious hackers, secure gateways have several other benefits. They can reduce internal system security overhead, since they allow an organization to concentrate security efforts on a limited number of machines. (This is similar to putting a guard on the first floor of a building instead of needing a guard on every floor.)

A second benefit is the centralization of services. A secure gateway can be used to provide a central management point for various services, such as advanced authentication (discussed in Chapter 16), e-mail, or public dissemination of information. Having a central management point can reduce system overhead and improve service.

17.3.2.3 Host-Based Authentication

Host-based authentication grants access based upon the *identity of the host* originating the request, instead of the identity of the user making the request. Many network applications in use today use host-based authentication to determine whether access is allowed. Under certain circumstances it is fairly easy to masquerade as the legitimate host, especially if the masquerading host is physically located close to the host being impersonated. Security measures to protect against misuse of some host-based authentication systems are available (e.g., Secure RPC[123] uses DES to provide a more secure identification of the client host).

An example of host-based authentication is the Network File System (NFS) which allows a server to make file systems/directories available to specific machines.

17.4 Administration of Access Controls

One of the most complex and challenging aspects of access control, administration involves implementing, monitoring, modifying, testing, and terminating user accesses on the system. These can be demanding tasks, even though they typically do not include making the actual decisions as

[123] RPC, or Remote Procedure Call, is the service used to implement NFS.

to the type of access each user may have.[124] 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.

There are three basic approaches to administering access controls: centralized, decentralized, or a combination of these. Each has relative advantages and disadvantages. Which is most appropriate in a given situation will depend upon the particular organization and its circumstances.

System and Security Administration

The administration of systems and security requires access to advanced functions (such as setting up a user account). The individuals who technically set up and modify who has access to what are very powerful users on the system; they are often called system or security administrators. On some systems, these users are referred to as having *privileged accounts.*

The type of access of these accounts varies considerably. Some administrator privileges, for example, may allow an individual to administer only one application or subsystem, while a higher level of privileges may allow for oversight and establishment of subsystem administrators.

Normally, users who are security administrators have two accounts: one for regular use and one for security use. This can help protect the security account from compromise. Furthermore, additional I&A precautions, such as ensuring that administrator passwords are robust and changed regularly, are important to minimize opportunities for unauthorized individuals to gain access to these functions.

17.4.1 Centralized Administration

Using centralized administration, one office or individual is responsible for configuring access controls. As users' information processing needs change, their accesses can be modified only through the central office, usually after requests have been approved by the appropriate official. This allows very strict control over information, because the ability to make changes resides with very few individuals. 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. Since relatively few individuals oversee the process, consistent and uniform procedures and criteria are usually not difficult to enforce. However, when changes are needed quickly, going through a central administration office can be frustrating and time-consuming.

17.4.2 Decentralized Administration

In decentralized administration, access is directly controlled by the owners or creators of the files, often the functional manager. This keeps control in the hands of those most accountable for the information, most familiar with it and its uses, and best able to judge who needs what kind of

[124] As discussed in the policy section earlier in this chapter, those decisions are usually the responsibility of the applicable application manager or cognizant management official. See also the discussion of system-specific policy in Chapters 5 and 10.

access. This may lead, however, to a lack of consistency among owners/creators as to procedures and criteria for granting user accesses and capabilities. Also, when requests are not processed centrally, it may be much more difficult to form a systemwide composite view of all user accesses on the system at any given time. Different application or 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.[125] It may also be difficult to ensure that all accesses are properly terminated when an employee transfers internally or leaves an organization.

17.4.3 Hybrid Approach

A hybrid approach combines centralized and decentralized administration. 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. The main disadvantage to a hybrid approach is adequately defining which accesses should be assignable locally and which should be assignable centrally.

17.5 Coordinating Access Controls

It is vital that access controls protecting a system work together. At a minimum, three basic types of access controls should be considered: physical, operating system, and application. In general, access controls within an application are the most specific. However, for application access controls to be fully effective they need to be supported by operating system access controls. Otherwise access can be made to application resources without going through the application.[126] Operating system and application access controls need to be supported by physical access controls.

17.6 Interdependencies

Logical access controls are closely related to many other controls. Several of them have been discussed in the chapter.

Policy and Personnel. The most fundamental interdependencies of logical access control are with policy and personnel. Logical access controls are the technical implementation of system-specific and organizational policy, which stipulates *who* should be able to access what kinds of information, applications, and functions. These decisions are normally based on the principles of

---

[125] Without necessary review mechanisms, central administration does not *a priori* preclude this.

[126] For example, logical access controls within an application block User A from viewing File F. However, if operating systems access controls do not also block User A from viewing File F, User A can use a utility program (or another application) to view the file.

separation of duties and least privilege.

Audit Trails. As discussed earlier, logical access controls can be difficult to implement correctly. Also, it is sometimes *not possible* to make logical access control as precise, or fine-grained, as would be ideal for an organization. In such situations, users may either deliberately or inadvertently abuse their access. For example, access controls cannot prevent a user from modifying data the user is authorized to modify, even if the modification is incorrect. Auditing provides a way to identify abuse of access permissions. It also provides a means to review the actions of system or security administrators.

Identification and Authentication. In most logical access control scenarios, the identity of the user must be established before an access control decision can be made. The access control process then associates the permissible forms of accesses with that identity. This means that access control can only be as effective as the I&A process employed for the system.

Physical Access Control. Most systems can be compromised if someone can physically access the machine (i.e., CPU or other major components) by, for example, restarting the system with different software. Logical access controls are, therefore, dependent on physical access controls (with the exception of encryption, which can depend solely on the strength of the algorithm and the secrecy of the key).

17.7 Cost Considerations

Incorporating logical access controls into a computer system involves the purchase or use of access control mechanisms, their implementation, and changes in user behavior.

Direct Costs. Among the direct costs associated with the use of logical access controls are the purchase and support of hardware, operating systems, and applications that provide the controls, and any add-on security packages. The most significant personnel cost in relation to logical access control is usually for administration (e.g., initially determining, assigning, and keeping access rights up to date). Label-based access control is available in a limited number of commercial products, but at greater cost and with less variety of selection. Role-based systems are becoming more available, but there are significant costs involved in customizing these systems for a particular organization. Training users to understand and use an access control system is another necessary cost.

Indirect Costs. The primary indirect cost associated with introducing logical access controls into a computer system is the effect on user productivity. There may be additional overhead involved in having individual users properly determine (when under their control) the protection attributes of information. Another indirect cost that may arise results from users not being able to immediately access information necessary to accomplish their jobs because the permissions were

incorrectly assigned (or have changed). This situation is familiar to most organizations that put strong emphasis on logical access controls.

References

Abrams, M.D., et al. *A Generalized Framework for Access Control: An Informal Description.* McLean, VA: Mitre Corporation, 1990.

Baldwin, R.W. "Naming and Grouping Privileges to Simplify Security Management in Large Databases." *1990 IEEE Symposium on Security and Privacy Proceedings.* Oakland, CA: IEEE Computer Society Press, May 1990. pp. 116-132.

Caelli, William, Dennis Longley, and Michael Shain. *Information Security Handbook.* New York, NY: Stockton Press, 1991.

Cheswick, William, and Steven Bellovin. *Firewalls and Internet Security.* Reading, MA: Addison-Wesley Publishing Company, 1994.

Curry, D. *Improving the Security of Your UNIX System, ITSTD-721-FR-90-21.* Menlo Park, CA: SRI International, 1990.

Dinkel, Charles. *Secure Data Network System Access Control Documents.* NISTIR 90-4259. Gaithersburg, MD: National Institute of Standards and Technology, 1990.

Fites, P., and M. Kratz. *Information Systems Security: A Practitioner's Reference.* New York, NY: Van Nostrand Reinhold, 1993. Especially Chapters 1, 9, and 12.

Garfinkel, S., and Spafford, G. "UNIX Security Checklist." *Practical UNIX Security.* Sebastopol, CA: O'Riley & Associates. Inc., 1991. pp. 401-413.

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

Haykin, M., and R. Warner. *Smart Card Technology: New Methods for Computer Access Control.* Spec Pub 500-157. Gaithersburg, MD: National Institute of Standards and Technology, 1988.

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

National Bureau of Standards. *Guidelines for Security of Computer Applications.* Federal

Information Processing Standard Publication 73. June 1980.

Pfleeger, Charles. *Security in Computing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1989.

President's Council on Integrity and Efficiency. *Review of General Controls in Federal Computer Systems*. Washington, DC: President's Council on Integrity and Efficiency, October 1988.

S. Salamone, "Internetwork Security: Unsafe at Any Node?" *Data Communications*. 22(12), 1993. pp. 61-68.

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

Thomsen, D.J. "Role-based Application Design and Enforcement." *Fourth IFIP Workshop on Database Security Proceedings*. International Federation for Information Processing, Halifax, England, September 1990.

T. Whiting. "Understanding VAX/VMS Security." *Computers and Security*. 11(8), 1992. pp. 695-698.