Access Control and Authorization 9

9.1 Definitions

Access control is a process to determine "Who does what to what," based on a policy.

One of the system administrator's biggest problems, which can soon turn into a nightmare if it is not well handled, is controlling access of who gets in and out of the system and who uses what resources, when, and in what amounts. Access control is restricting this access to a system or system resources based on something other than the identity of the user. For example, we can allow or deny access to a system's resources based on the name or address of the machine requesting a document.

Access control is one of the major cornerstones of system security. It is essential to determine how access control protection can be provided to each of the system resources. To do this, a good access control and access protection policy is needed. According to Raymond Panko, such a policy has benefits including the following [1]:

- It focuses the organization's attention on security issues, and probably, this attention results in resource allocation toward system security.
- It helps in configuring appropriate security for each system resource based on role and importance in the system.
- It allows system auditing and testing.

As cyberspace expands and the forces of globalization push e-commerce to the forefront of business and commercial transactions, the need for secure transactions has propelled access control to a position among the top security requirements, which also include authorization and authentication. In this chapter, we are going to discuss access control and authorization; authentication will be discussed in the next chapter.

J.M. Kizza, *Guide to Computer Network Security*, Computer Communications
and Networks, DOI 10.1007/978-3-319-55606-2_9

9.2 Access Rights

To provide authorization, and later as we will see authentication, system administrators must manage a large number of system user accounts and permissions associated with those accounts. The permissions control user access to each system resource. So, user A who wants to access resource R must have permission to access that resource based on any one of the following modes: read, write, modify, update, append, and delete. Access control regimes and programs, through validation of passwords and access mode permissions, let system users get access to the needed system resources in a specified access mode.

Access control consists of four elements: subjects, objects, operations, and a reference monitor. In the normal operation, seen in Fig. 9.1, the subject, for example, a user, initiates an access request for a specified system resource, usually a passive object in the system such as a Web resource. The request goes to the reference monitor. The job of the reference monitor is to check on the hierarchy of rules that specify certain restrictions. A set of such rules is called an *access control list* (ACL). The access control hierarchy is based on the URL path for Web access or the file path for a file access such as in a directory. When a request for access is made, the monitor or server goes in turn through each ACL rule, continuing until it encounters a rule that prevents it from continuing and results in a request rejection or comes to the last rule for that resource, resulting into access right being granted.

Subjects are system users and groups of users, while objects are files and resources such as memory, printers, and scanners including computers in a network. An access operation comes in many forms including Web access, server access, memory access, and method calls. Whenever a subject requests to access an object, an access mode must be specified. There are two access modes: observe and alter. In the observe mode, the subject may only look at the content of the object; in the alter mode, the subject may change the content of the object. The observe mode is the typical read in which a client process may request a server to read from a file.

Access rights refer to the user's ability to access a system resource. There are four access rights: *execute, read, append,* and *write.* The user is cautioned not to confuse access rights and access modes. The difference lies in the fact that you can perform any access right within each access mode. Figure 9.2 shows how this can be done. Note that according to the last column in Fig. 9.2, there are X marks in both rows because in order to write, one must observe first before altering. This prevents the operating system from opening the file twice, one for the read and another for a write.

Access rights can be set individually on each system resource for each individual user and group. It is possible for a user to belong to a few groups and enjoy those groups' rights. However, user access rights always take precedence over group access rights regardless of where the group rights are applied. If there are inherited group access rights, they take precedence over user default access rights. A user has default rights when the user has no assigned individual or group rights from the root down to the folder in question. In the cascading of access rights application, user

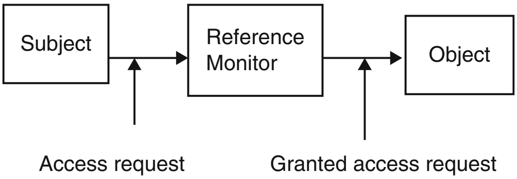

Fig. 9.1 Access control administration

Fig. 9.2 Access modes and access rights (Ref. [2])

	execute	append	read	write
observe			X	X
alter		X		X

access rights that are closest to the resource being checked by the monitor take precedence over access rights assignments that are farther away.

We have so far discussed access rights to resources. The question that still remains to be answered is: Who sets these rights? The owner of the resource sets the access rights to the resource. In a global system, the operating systems own all system resources and therefore set the access rights to those resources. However, the operating system allows folders and file owners to set and revoke access rights.

9.2.1 Access Control Techniques and Technologies

Because a system, especially a network system, may have thousands of users and resources, the management of access rights for every user per every object may become complex. Several control techniques and technologies have been developed to deal with this problem; they include access control matrix, capability tables, access control lists, role-based access control, rule-based access control, restricted interfaces, and content-dependent access control.

Many of the techniques and technologies we are going to discuss below are new in response to the growth of cyberspace and the widespread use of networking. These new techniques and technologies have necessitated new approaches to system access control. For a long time, access control was used with user- or group-based access control lists, normally based in operating systems. However, with Web-based network applications, this approach is no longer flexible enough because it does not scale in the new environment. Thus, most Web-based systems employ newer techniques and technologies such as role-based and rule-based access control, where access rights are based on specific user attributes such as their role, rank, or organization unit.

9.2.1.1 Access Control Matrix

All the information needed for access control administration can be put into a matrix with rows representing the subjects or groups of subjects and columns representing the objects. The access that the subject or a group of subjects is

Fig. 9.3 Access matrix

Objects → Subjects/groups | V	R1	R2	R3	R4
A	W	R	R	W
B	R			
Group G1	W			
Group G2		W		
C				R

permitted to the object is shown in the body of the matrix. For example, in the matrix shown in Fig. 9.2, user A has permission to write in file R4. One feature of the access control matrix is its sparseness. Because the matrix is so sparse, storage consideration becomes an issue, and it is better to store the matrix as a list.

9.2.1.2 Access Control Lists

In the access control lists (ACLs), groups with access rights to an object are stored in association to the object. If you look at the access matrix shown in Fig. 9.2, each object has a list of access rights associated with it. In this case, each object is associated with all the access rights in the column. For example, the ACL for the matrix shown in Fig. 9.3 is shown in Fig. 9.4.

ACLs are very fitting for operating systems as they manage access to objects [2].

9.2.1.3 Access Control Capability

A capability specifies that "the subject may do operation O on object X."

Unlike the ACLs, where the storage of access rights between objects and subjects is based on columns in the access control matrix, capabilities access control storage is based on the rows. This means that every subject is given a capability, a forgery-proof token that specifies the subject's access rights [2].

From the access matrix shown in Fig. 9.3, we can construct a capability as shown in Fig. 9.5.

9.2.1.4 Role-Based Access Control

The changing size and technology of computer and communication networks are creating complex and challenging problems in the security management of these large networked systems. Such administration is not only becoming complex as technology changes and more people join the networks, it is also becoming extremely costly and prone to error when it is solely based on access control lists for each user on the system individually.

System security in role-based access control (RBAC) is based on roles assigned to each user in an organization. For example, one can take on a role as a chief executive officer, a chief information officer, or chief security officer. A user may

Fig. 9.4 Access control list (ACL)

Object	Access rights	Subjects
R1	W	A
	R	B
	W	Group G1
R2	R	A
	W	Group G2
R3	R	A
R4	R	A
	R	C

Subject	Object 1/Access	Object 2/Access	Object 3/Access	Object 4/Access
A	R1/W	R2/R	R3/R	R4/R
B	R1/R			
Group G1	R1/W			
Group G2		R2/W		
C				R4/R

Fig. 9.5 Access control capability lists

be assigned one or more roles, and each role is assigned one or more privileges that are permitted to users in that role. Access decisions are then based on the roles that individual users have as part of an organization. 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.

Access rights are grouped by role name, and the use of resources is restricted to individuals authorized to assume the associated role. A good example to illustrate the role names and system users who may assume more than one role and play those roles while observing an organization's security policy is the following given in the NIST/ITL Bulletin, of December 1995. "Within a hospital system the role of doctor can include operations to perform diagnosis, prescribe medication, and order laboratory tests, and the role of researcher can be limited to gathering anonymous clinical information for studies" [3].

Accordingly, users are granted membership into roles based on their competencies and responsibilities in the organization. The types of operations that a user is permitted to perform in the role he or she assumes are based on that user's role. User roles are constantly changing as the user changes responsibilities and functions in the organizations, and these roles can be revoked. Role associations can be established when new operations are instituted, and old operations can be deleted as organizational functions change and evolve. This simplifies the administration and management of privileges; roles can be updated without updating the privileges for every user on an individual basis.

Like other types of access control, RBAC is also based on the concept of *least privilege* that requires identifying the user's job functions, determining the minimum set of privileges required to perform that function, and restricting the user to a domain with those privileges and nothing more. When a user is assigned a role, that user becomes associated with that role, which means that user can perform a certain and specific number of privileges in that role. Although the role may be associated with many privileges, individual users associated with that role may not be given more privileges than are necessary to perform their jobs.

Although this is a new technology, it is becoming very popular and attracting increasing attention, particularly for commercial applications, because of its potential for reducing the complexity and cost of security administration in large networked applications.

9.2.1.5 Rule-Based Access Control

Like other access control regimes, rule-based access control (RBAC), also known as *policy-based access control* (PBAC), is based on the least privilege concept. It is also based on policies that can be algorithmically expressed. RBAC is a multipart process, where one process assigns roles to users just like in the role-based access control techniques discussed above. The second process assigns privileges to the assigned roles based on a predefined policy. Another process is used to identify and authenticate the users allowed to access the resources.

It is based on a set of rules that determine users' access rights to resources within an organization's system. For example, organizations routinely set policies on the access to the organizations' Web sites on the organizations' intranet or Internet. Many organizations, for example, limit the scope and amount, sometimes the times, employees, based on their ranks and roles, can retrieve from the site. Such limits may be specified based on the number of documents that can be downloaded by an employee during a certain time period and on the limit of which part of the Web site such an employee can access.

The role of ACLs has been diminishing because ACLs are ineffective in enforcing policy. When using ACLs to enforce a policy, there is usually no distinction between the policy description and the enforcement mechanism (the policy is essentially defined by the set of ACLs associated with all the resources on the network). Having a policy being implicitly defined by a set of ACLs makes the management of the policy inefficient, error prone, and hardly scalable up to large enterprises with large numbers of employees and resources. In particular, every time an employee leaves a company or even just changes his/her role within the company, an exhaustive search of all ACLs must be performed on all servers, so that user privileges are modified accordingly.

In contrast with ACLs, policy-based access control makes a strict distinction between the formal statement of the policy and its enforcement. It makes rules explicit, and instead of concealing them in ACLs, it makes the policy easier to manage and modify. Its advantage is based on the fact that it administers the concept of least privilege justly because each user can be tied to a role which in turn can be tied to a well-defined list of privileges required for specific tasks in the role. In addition, the roles can be moved around easily and delegated without explicitly de-allocating a user's access privileges [4].

9.2.1.6 Restricted Interfaces

As the commercial Internet grows in popularity, more and more organizations and individuals are putting their data into organization and individual databases and restricting access to it. It is estimated that 88% of all cyberspace data is restricted data or what is called hidden data [5].

For the user to get access to restricted data, the user has to go via an interface. Any outside party access to restricted data requires a special access request, which many times requires filling in an online form. The interfaces restrict the amount and quality of data that can be retrieved based on filter and retrieval rules. In many cases, the restrictions and filters are instituted by content owners to protect the integrity and proprietary aspects of their data. The Web site itself and the browser must work in cooperation to overcome the over-restriction of some interfaces. Where this is impossible, hidden data is never retrievable.

9.2.1.7 Content-Dependent Access Control

In content-dependent access control, the decision is based on the value of the attribute of the object under consideration. Content-dependent access control is very expensive to administer because it involves a great deal of overhead resulting from the need to scan the resource when access is to be determined. The higher the level of granularity, the more expensive it gets. It is also extremely labor intensive.

9.2.1.8 Other Access Control Techniques and Technologies

Other access control techniques and technologies include those by the US Department of Defense (DoD) that include discretionary access control (DAC), mandatory access control (MAC), context-based access control (CBAC), view-based access control (VBAC), and user-based access control (UBAC).

DAC permits the granting and revoking of access control privileges to be left to the discretion of the individual users. A DAC mechanism departs a little bit from many traditional access control mechanisms where the users do not own the information to which they are allowed access. In DAC, users own the information and are allowed to grant or revoke access to any of the objects under their control.

Mandatory access control (MAC), according to DoD, is "a means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity." [3].

Context-based access control (CBAC) makes a decision to allow access to a system resource based not only on who the user is, which resource it is, and its content but also on its history, which involves the sequence of events that preceded the access attempt.

View-based access control (VBAC), unlike other notions of access control which usually relate to tangible objects such as files, directories, and printers, takes the system resource itself as a collection of sub-resources, which are the views. This allows all users to access the same resource based on the view they have of the resource. It makes an assumption that the authentication of the source has been done by the authentication module.

User-based access control (UBAC), also known as identity-based access control (IBAC), is a technique that requires a system administrator to define permissions for each user based on the individual's needs. For a system with many users, this technique may become labor intensive because the administrator is supposed to know precisely what access each and every user needs and configure and update permissions.

9.3 Access Control Systems

In Sect. 2.3.1, we briefly discussed system access control as part of the survey of system security services. The discussion then was centered on both hardware and software access control regimes. Let us now look at these services in a more detailed form.

9.3.1 Physical Access Control

Although most accesses to an organization systems are expected to originate from remote sites and therefore access the system via the network access points, in a limited number of cases, system access can come from intruders physically gaining access to the system itself, where they can install password-cracking programs. Studies have shown that a great majority of system break-ins originate from inside the organization. Access to this group of users who have access to the physical premises of the system must be appropriate.

9.3.2 Access Cards

Cards as access control devices have been in use for sometime now. Access cards are perhaps the most widely used form of access control system worldwide. Initially, cards were used exclusively for visual identification of the bearer. However, with advanced digital technology, cards with magnetic strips and later with embedded microchips are now very common identification devices. Many companies require their employees to carry identity cards or identity badges with a photograph of the card holder or a magnetic strip for quick identification. Most hotels now have done away with metal keys in favor of magnet stripe keys. Access cards are used in most e-commerce transactions, payment systems, and in services such as health and education. These types of identification are also known as electronic keys.

Access control systems based on an embedded microprocessor, known as smart cards, have a number of advantages including the ability to do more advanced and sophisticated authentication because of the added processing power, storing large quantities of data, usually personal data, and smaller sizes. Smart cards also have exceptional reliability and extended life cycle because the chip is usually encased in tamper-resistant materials like stainless steel. The cards, in addition, may have built-in unique security identifier numbers called personal identification numbers (PINs) to prevent information falsification and imitations.

A cousin of the smart card is the proximity card. Proximity cards are modern, prestigious, and easy-to-use personal identifiers. Like magnetic and smart cards, proximity cards also have a great deal of embedded personal information. However, proximity cards have advantages the other cards do not have. They can be used in extreme conditions and still last long periods of time. They can also be read from a distance such as in parking lots where drivers can flash the card toward the reader while in a car and the reader still reads the card through the car window glass.

9.3.3 Electronic Surveillance

Electronic surveillance access control consists of a number of activity frame captures such as video recordings, system logs, keystroke and application monitors, screen-capture software commonly known as activity monitors, and network packet sniffers.

Video recordings capture the activities at selected access points. Increasingly these video cameras are now connected to computers and actually a Web, a process commonly now referred to as webcam surveillance. Webcam surveillance consists of a mounted video camera, sometimes very small and embedded into some object, camera software, and an Internet connection to form a closed-circuit monitoring system. Many of these cameras are now motion-activated, and they record video footage shot from vantage points at the selected points. For access control, the selected points are system access points. The video footage can be viewed live or stored for later viewing. These captures can also be broadcast over the Internet or transmitted to a dedicated location or sent by e-mail.

Keystroke monitors are software or hardware products that record every character typed on keyboards. Software-based keystroke monitors capture the signals that move between keyboard and computer as they are generated by all human-computer interaction activities that include the applications ran, chats, and e-mails sent and received. The captures are then sent live onto a closed-circuit recording system that stores them to a file for future review or sends them by e-mail to a remote location or user. Trojan horse spyware such as Back Orifice and Netbus are good examples of software-based monitoring tools [6].

Packet sniffers work at a network level to sniff at network packets as they move between nodes. Depending on the motives for setting them, they can motive all packets, selected packets, or node-originating and node-bound traffic. Based on the analysis, they can monitor e-mail messages, Web browser usage, node usage, traffic into a node, nature of traffic, and how often a user accesses a particular server, application, or network [6].

9.3.4 Biometrics

Biometric technology, based on human attributes, something you are, aims to confirm a person's identity by scanning a physical characteristic such as a fingerprint, voice, eye movement, facial recognition, and others. Biometrics came into use because we tend to forget something we have. We forget passwords, keys, and cards. Biometric has been and continues to be a catch-all and buzz word for all security control techniques that involve human attributes. It has probably been one of the oldest access control techniques. However, during the past several years and with heightened security, biometric technology has become increasingly popular. The technology, which can be used to permit access to a network or a building, has become an increasingly reliable, convenient, and cost-effective means of security.

Current technology has made biometric access control much more practical than it has ever been in the past. Now, a new generation of low-cost yet accurate fingerprint readers is available for most mobile applications so that screening stations can be put up in a few minutes. Although biometrics is one of those security control techniques that have been in use the longest, it does not have standards as yet. There is an array of services on the market for biometric devices to fit every form of security access control.

Technological advances have resulted in smaller, high-quality, more accurate, and more reliable devices. Improvements in biometrics are essential because bad biometric security can lull system and network administrators into a false sense of safety. In addition, it can also lock out a legitimate user and admit an intruder. So, care must be taken when procuring biometric devices.

Before a biometric technique can be used as an access control technique for the system, each user of the system first has his or her biometric data scanned by a biometric reader, processed to extract a few critical features, and then those few features stored in a database as the user's template. When a user requests access to a system resource and that user must be authenticated, the biometric readers verify customers' identities by scanning their physical attributes, such as fingerprints, again. A match is sought by checking them against prints of the same attributes previously registered and stored in the database.

One of the advantages that has made biometrics increasingly popular is that while other methods of access control such as authentication and encryption are crucial to network security and provide a secure way to exchange information, they are still expensive and difficult to design for a comprehensive security system. Other access control techniques such as passwords, while inexpensive to implement, are easy to forget and easy to guess by unauthorized people if they are simple and too complex to be of any use if they are complex.

9.3.4.1 Fingerprint Readers

Fingerprint recognition technology is perhaps one of the oldest biometric technologies. Fingerprint readers have been around for probably hundreds of years. These readers fall into two categories: mice with embedded sensors and stand-alone units. Mice are the latest 3D imaging developments and are threatening the stand-alone because they can play a dual role; they can be used on a desktop and also as network authentication stations. This is leading to the bundling of fingerprint recognition devices with smart cards or some other security token.

Although fingerprint technology is improving with current technology, making it possible to make a positive identification in a few seconds, fingerprint identification is susceptible to precision problems. Many fingerprints can result in false positives due to oil and skin problems on the subject's finger. Also, many of the latest fingerprint readers can be defeated by photos of fingerprints and 3D fingers from latent prints such as prints left on glass and other objects [1].

9.3.4.2 Voice Recognition

Although voice recognition technology is a biometric that is supposed to authenticate the user based on what the use is, voice imprint is based on something the user does, itself based on who the user is. Voice recognition has been around for years; however, its real-life application has been slow because of the difficulties in deployment. In order for voiceprint technology to work successfully, it needs to be deployed by first developing the front end to capture the input voice and connect it to the back-end systems which process the input and do the recognition.

The front end of the voiceprint authentication technology works much the same as other biometric technologies, by creating a digital representation of a person's voice using a set of sophisticated algorithms. Those attributes are stored in a database, part of the back end, which is prompted to make a match against the user's voice when the online system is accessed.

To set it up initially, each user is required to record and leave his or her voiceprint, which is stored in the system's database to be activated whenever the user requests access to the protected facility through a physical system input. The user is then prompted to speak into a computer's microphone to verify his or her identity.

Current systems use two servers to perform these functions. The first server runs the front-end system, and the second server then stores the database and does the processing for a recognition from the input server.

Voice recognition is not a safe authentication technique because it can be fooled by the types of recording.

9.3.4.3 Hand Geometry

Hand geometry is an authentication technology that uses the geometric shape of the hand to identify a user. The technique works by measuring and then analyzing the shape and physical features of a user's hand, such as finger length and width and palm width. Like fingerprints, this technique also uses a reader. To initiate the device, all users' hands are read and measured and the statistics are stored in a database for future recognition. To activate the system, the user places the palm of his or her hand on the surface of the reader. Readers usually have features that guide the user's hand on the surface. Once on the surface, the hand, guided by the guiding features, is properly aligned for the reader to read off the hand's attributes. The reader is hooked to a computer, usually a server, with an application that provides a live visual feedback of the top view and the side view of the hand. Hand features are then taken as the defining feature vector of the user's hand and are then compared with the user features stored in the database.

Although hand geometry is simple, human hands are not unique; therefore, individual hand features are not descriptive enough for proper identification. The technique must be used in conjunction with other methods of authentication.

9.3.4.4 Iris Scan

The human iris is the colored part of the human eye and is far more complex and probably more precise than a human fingerprint; thus, it is a good candidate for authentication. According to Panko, iris authentication is the gold standard of all biometric authentications [1]. Iris scan technology, unlike the retinal scan, does not have a long history. In fact, the idea of using iris patterns for personal identification was first mooted in 1936 by ophthalmologist Frank Burch. By the mid-1980s, the idea was still a science fiction appearing only in James Bond films. The technology came into full use in the 1990s [7].

Iris technology is an authentication technology that uses either regular or infrared light into the eye of the user to scan and analyze the features that exist in the colored tissue surrounding the pupil of the user's eye. Like the previous biometric technologies, iris technology also starts off by taking samples of the user eye features using a conventional closed-circuit digital (CCD) or video camera that can work through glasses and contacts. The camera scans the tissue around the pupils for analysis features. Close to 200 features can be extracted from this tissue surrounding the pupil and used in the analysis. The tissue gives the appearance of dividing the iris in a radial fashion. The most important of these characteristics in the tissue is the trabecular meshwork visible characteristic. Other extracted visible characteristics include rings, furrows, freckles, and the corona.

The first readings are stored in a database. Whenever a user wants access to a secure system, he or she looks in an iris reader. Modern iris readers can read a user's eye up to 2 ft away. Verification time is short and it is getting shorter. Currently it stands at about 5 s, although the user will need to look into the device only for a couple moments. Like in other eye scans, precautions must be taken to prevent a wrong person's eyes from fooling the system. This is done by varying the light shone into the eye and then pupil dilations are recorded.

The use of iris scans for authentication is becoming popular, although it is a young technology. Its potential application areas include law enforcement agencies and probably border patrol and airports. There is also potential use in the financial sector, especially in banking.

9.3.5 Event Monitoring

Event monitoring is a cousin of electronic monitoring in which the focus is on specific events of interest. Activities of interest can be monitored by video camera, webcam, digital or serial sensors, or a human eye. All products we discussed in Sects. 9.3.3 and 9.3.4.2 can be used to capture screenshots, monitor Internet activity, and report a computer's use, keystroke by keystroke, and human voice, including human movement. The activities recorded based on selected events can be stored, broadcast on the Internet, or sent by e-mail to a selected remote location or user.

9.4 Authorization

This is the determination of whether a user has permission to access, read, modify, insert, or delete certain data, or to execute certain programs. In particular, it is a set of access rights and access privileges granted to a user to benefit from a particular system resource. Authorization is also commonly referred to as access permissions, and it determines the privileges a user has on a system and what the user should be allowed to do to the resource. Access permissions are normally specified by a list of possibilities. For example, Unix allows the list {read, write, execute} as the list of possibilities for a user or group of users on a Unix file.

We have seen above that access control consists of defining an access policy for each system resource. The enforcement of each one of these access policies is what is called authorization. It is one thing to have a policy in place, but however good a policy is, without good enforcement, the policy serves no purpose. The implementation of mechanisms to control access to system resources is, therefore, a must for an effective access control regime.

The process of authorization itself has traditionally been composed of two separate processes: authentication, which we are going to discuss in the next chapter, and access control. To get a good picture, let us put them together. In brief, authentication deals with ascertaining that the user is who he or she claims he or she is. Access control then deals with a more refined problem of being able to find out "what a specific user can do to a certain resource." So authorization techniques such as the traditional centralized access control use ACL as a dominant mechanism to create user lists and user access rights to the requested resource. However, in more modern and distributed system environments, authorization takes a different approach from this. In fact, the traditional separation of authorization process into authentication and access control also does not apply [8].

As with access control, authorization has three components: a set of objects we will designate as O, a set of subjects designed as S, and a set of access permissions designated as S. The authorization rule is a function f that takes the triple (s, o, a), where $s \in S, o \in O, a \in A$ and maps then into a binary-value T, where $T = \{\text{true}, \text{false}\}$ as $f: S \times O \times A \rightarrow (True, False)$. When the value of the function f is true, this signals that the request for subject *s* to gain access to object *o* has been granted at authorization level *a*.

The modern authentication process is decentralized to allow more system independence and to give network services providers more control over system resource access. This is also the case in yet more distributed systems, since in such systems, it is hard and sometimes impossible to manage all users and resources in one central location. In addition, many servers actually do not need to know who the user is in order to provide services.

The capability mechanism so central in the traditional process, however, still plays a central role here, providing for decentralization of authorization through providing credentials to users or applications whenever it receives requests for resource access. Each user or application keeps a collection of capabilities, one for each resource they have access to, which they must present in order to use the requested resource. Since every resource maintains its own access control policy and complete proof of compliance between the policy and credentials collected from the user or application, the server receiving the request need not consult a centralized ACL for authorization [8].

9.4.1 Authorization Mechanisms

Authorization mechanisms, especially those in database management systems (DBMSs), can be classified into two main categories: discretionary and mandatory.

9.4.1.1 Discretionary Authorization

This is a mechanism that grants access privileges to users based on control policies that govern the access of subjects to objects using the subjects' identity and authorization rules, discussed in Sect. 9.3 above. These mechanisms are discretionary in that they allow subjects to grant other users authorization to access the data. They are highly flexible, making them suitable for a large variety of application domains.

However, the same characteristics that make them flexible also make them vulnerable to malicious attacks, such as Trojan horses embedded in application programs. The reason is that discretionary authorization models do not impose any control on how information is propagated, and once used, they have been accessed by users authorized to do so.

But in many practical situations, discretionary policies are preferred since they offer a better trade-off between security and applicability. For this reason, in this chapter, we focus on discretionary access control mechanisms. We refer the reader to [4] for details on mandatory access control mechanisms.

9.4.1.2 Mandatory Access Control

Mandatory policies, unlike the discretionary ones seen above, ensure a high degree of protection in that they prevent any illegal flow of information through the enforcement of multilevel security by classifying the data and users into various security classes. They are, therefore, suitable for contexts that require structured but graded levels of security such as the military. However, mandatory policies have the drawback of being too rigid in that they require a strict classification of subjects and objects in security levels and are therefore applicable only to very few environments [4].

9.5 Types of Authorization Systems

Before the creation of decentralized authorization systems, authorization was controlled from one central location. Operating system authorization, for example, was centrally controlled before the advent of network operating systems (NOSs). The birth of computer networks and therefore NOS created the decentralized authorization systems.

9.5.1 Centralized

Traditionally, every resource used to do its own local authorizations and maintained its own authorization database to associate authorizations to users. But this led to several implementation problems. For example, different resources and different software applied different rules to determine authorization for the same subject on an object. This led to the centralized authorization policy. In centralized authorization, only one central authorization unit grants and delegates access to system resources. This means that any process or program that needs access to any system resource has to request from the one omniscient central authority. Centralized authorization services allow you to set up generalized policies that control who gets access to resources across multiple platforms. For example, it is possible to set an authorization to a company's Web portal in such a way that authorization is based on either functions or titles. Those with such functions could control their organization's specially designated component of the portal, while others without functions access the general portal. This system is very easy and inexpensive to operate. A single database available to all applications gives a better more and consistent view of security. It also simplifies the process of adding, modifying, and deleting authorizations. All original operating systems have been using this authorization approach.

9.5.2 Decentralized

This differs from the centralized system in that the subjects own the objects they have created and are therefore responsible for their security, which is locally maintained. This means that each system resource maintains its own authorization process and maintains its own database of authorizations associated with all subjects authorized to access the resource. Each subject also possesses all possible rights to access every resource associated with it. Each subject may, however, delegate access rights to its objects to another subject. Because of these characteristics, decentralized authorization is found to be very flexible and easily adaptable to particular requirements of individual subjects. However, this access rights delegation may lead to the problem of cascading, and cyclic authorization may arise.

9.5.3 Implicit

In implicit authorization, the subject is authorized to use a requested system resource indirectly because the objects in the system are referenced in terms of other objects. That means that in order for a subject to access a requested object, the access must go through an access of a primary object. Using the mathematical set theoretical representation we presented earlier, in a given set of sets *(s, o, a),* a user *s* is implicitly given a type *a* authorization on all the objects of *o***.** Take, for example, a request to use a Web page; the page may have links connected to other documents. The user who requests for authorization to use the Web has also indirect authorization to access all the pages linked to the authorized original page. This is, therefore, a level of authorization called granularity. We are going to discuss this later. Notice that a single authorization here enables a number of privileges.

9.5.4 Explicit

Explicit authorization is the opposite of the implicit. It explicitly stores all authorizations for all system objects whose access has been requested. Again in a mathematical representation seen earlier, for every request for access to object *o* from subject *s* that is grantable, the triple set *(s, o, a)* is stored. All others are not stored. Recall from the last chapter that one of the problems of access control was to store a large but sparse matrix of access rights. This technique of storing only authorized triples greatly reduces the storage requirements. However, although simple, the technique still stores authorizations whether needed or not, which wastes storage.

9.6 Authorization Principles

The prime object of authorization is system security achieved through the controlled access to the system resources. The authorization process, together with access control discussed earlier, through the use of authorization data structures, clearly defines who uses what system resources and what resources can and cannot be used. The authorization process, therefore, offers undeniable security to the system through the protection of its resources. System resources are protected through principles such as least privilege and separation of duties, which eventually results in increased accountability that leads to increased system security.

9.6.1 Least Privilege

The *principle of least privilege* requires that the subject be granted authorizations based on its needs. Least privilege principle is itself based on two other principles: *less rights* and *less risk.* The basic idea behind these principles is that security is improved if subjects using system resources are given no more privileges than the minimum they require to perform the tasks that they are intended to perform and in the minimum amount of time required to perform the tasks. The least privilege principle has the ability, if followed, to reduce the risks of unauthorized accesses to the system.

9.6.2 Separation of Duties

The principle of separation of duties breaks down the process of authorization into basic steps and requires that for every request for authorization from a subject to a system resource, each step be given different privileges. It requires that each different key step in a process requires different privileges for different individual subjects. This division of labor, not only in the authorization process of one individual request but also between individual subjects, stipulates not only that one subject should never be given a blanket authorization to do all the requested functions but also that no one individual request to an object should be granted blanket access rights to an object. This hierarchical or granular authorization distributes responsibilities and creates accountability because no one subject is responsible for large processes where responsibility and accountability may slack. For example, authorization to administer a Web server or an e-mail server can be granted to one person without granting him or her administrative rights to other parts of the organization system.

9.7 Authorization Granularity

We have used the concept of granularity in the last section without officially defining it. Let us do so here. Granularity in access authorization means the level of details an authorizing process requires to limit and separate privileges. Because a single authorization may enable a number of privileges or a privilege may require multiple authorizations, when requests come into the authorizing process from subjects requiring access to system resources, the authorizing authority must pay attention and separate these two authorization privileges. These two issues may complicate the authorization process. Granularity, therefore, should be defined on functions [9].

9.7.1 Fine-Grained Authorization

As we discussed above, granularity of authorizations should not be based on either authorization requests or on granted privileges but on functions performed. Fine-grained granularity defines very specific functions that individually define specific tasks. This means that each authorization request is broken up into small but specific tasks and each one of these tasks is assigned a function.

9.7.2 Coarse-Grained Authorization

Coarse-grained granularity is different from fine-grained granularity in that here only the basic ability to interact with resources is focused on. Then all lower detail tasks within the large functions are ignored. These abilities can be enforced by the operating system without concern for the applications. In fact, it is this type of authorization that is enforced by most operating systems. For example, most operating systems have the following abilities or functions: delete, modify, read, write, and create. Subject requests for access authorization must then be put into one of these major functions or abilities.

9.8 Web Access and Authorization

The growth of the Internet and e-commerce has made Web application the fastest-growing client-server application model and the mainstay of the Internet. Accordingly, Web servers have also become the main targets for intruder break-ins. So, controlling access to Web-based resources has naturally become an administrative nightmare.

The Web infrastructure supports a distributed authorizing structure based on node-naming structures, where each node is known by an URL and information to be retrieved from it is accessible through protocols such as HTTP. Under this structure, authorization is based on an access control list (ACL). In a distributed environment such as this, each server node needs to either know its potential clients or there must be an authorizing server that other servers must consult before request authorization. However, both of these approaches present problems for the Web authorization because the former approach presents a client administration problem when the client population changes at a fast rate. The latter approach presents a potential performance bottleneck as the processing of a node request depends on the performance and availability of the authorization server [10].

In a fully functioning distributed Web authorization process, a coordinated authorization approach is required that grants access not only to requested document but also to all other documents linked to it. But by this writing, this is not the case.

Whether using the present authorization model or those to come, every effort must be used to control access to Web servers and minimize unauthorized access to them. In addition to access control and authorization, here are the other tips for securing servers [11]:

- Web servers should not run any other services with the exception of a carefully configured anonymous FTP.
- Periodic security scans by a trusted third party should be scheduled to identify system security weaknesses.
- Minimize system risk by never running the Web server as "root" or "administrator." Server processes should be run from a new account with no other privileges on the machine.
- For shared file system such as AFS or NFS, give the Web server only "read only" access, or separately mount a "read only" data disk.

Exercises

1. Differentiate between access and authorization.
2. What are the benefits of authorization?
3. Why is it difficult to implement distributed authorization?
4. Discuss the merits and demerits of centralized and decentralized authorization.
5. Compare the authorization model used by the network operating systems (NOSs) to that used by the old stand-alone operating systems.
6. List and discuss the most common access privileges in a computing system.
7. Discuss the three components of a global access model.
8. Physical access to resources is essential and must be the most restricted. Why?
9. Discuss four access methods, giving the weaknesses of each.
10. Discuss the many ways in which access can be abused.

Advanced Exercises

1. Is it possible to implement full distributed authorization? What will be involved?
2. Web authorization is central to the security of all Web applications. What is the best way to safeguard all Web applications and at the same time make Web access reliable and fast?
3. Consider an environment where each server does its own authorization. If an access request is made to a document that has extended links and one of the link requests is denied, should the whole document request be denied? Why or why not?
4. Discuss the benefits and problems resulting from the "least privilege" principle often used in access control.
5. Discuss the concept of global privilege. Does it work well in a distributed authorization or centralized authorization?
6. With the principle of "least privilege," is it possible to have too much authorization? What happens when there is too much authorization?

References

1. Panko RR (2004) Corporate computer and network security. Prentice-Hall, Upper Saddle River
2. Gollman D (2000) Computer security. Wiley, New York
3. An Introduction to Role-based Access Control. NIST/ITL Bulletin, December, 1995. http://csrc.nist.gov/rbac/NIST-ITL-RBAC-bulletin.html
4. Differentiating Between Access Control Terms. http://secinf.net/uplarticle/2/Access_Control_WP.pdf
5. Byers S, Freire J, Silva C. Efficient acquisition of web data through restricted query interfaces. AT&T Labs-Research. http://www10.org/cdrom/posters/p1051/
6. Bannan K. Watching you, watching me PCs are turning informant. Whose side are they on? PC Magazine, July 1, 2002, http://www.pcmag.com/article2/0,4149,342208,00.asp
7. Iris scan. http://ctl.ncsc.dni.us/biomet%20web/BMIris.html
8. NASA World Wide Web Best Practices (2000–2001) Draft version 2.0. http://nasa-wbp.larc.nasa.gov/devel/4.0/4_4.html
9. Pipkin D (2000) Information security: protecting the global enterprise. Prentice-Hall, Upper Saddle River
10. Kahan J. A distributed authorization model for WWW. May, 1995. http://www.isoc.org/HMP/PAPER/107/html/paper.html. Accessed on 5/6/2003
11. NASA. World wide web best practices 2000–2001 draft version 2.0. 8/20/2000. http://nasa-wbp.larc.nasa.gov/devel/4.0/4_4.html. 5/6/2003