CHAPTER 4

Access Control

This chapter presents the following:

- Identification methods and technologies
- Authentication methods, models, and technologies
- Discretionary, mandatory, and nondiscretionary models
- Accountability, monitoring, and auditing practices
- Emanation security and technologies
- Intrusion detection systems
- Possible threats to access control practices and technologies

A cornerstone in the foundation of information security is to control how resources are accessed so that they can be protected from unauthorized modification or disclosure. The controls that enforce access control can be technical, physical, or administrative in nature.

Access Controls Overview

Access controls are security features that control how users and systems communicate and interact with other systems and resources. They protect the systems and resources from unauthorized access and can be a component that participates in determining the level of authorization after an authentication procedure has successfully completed. Although we usually think of a user as the entity that requires access to a network resource or information, there are many other types of entities that require access to other network entities and resources that are subject to access control. It is important to understand the definition of a subject and an object when working in the context of access control.

Access is the flow of information between a subject and an object. A ***subject*** is an active entity that requests access to an object or the data within an object. A subject can be a user, program, or process that accesses an object to accomplish a task. When a program accesses a file, the program is the subject and the file is the object. An ***object*** is

a passive entity that contains information. An object can be a computer, database, file, computer program, directory, or field contained in a table within a database. When you look up information in a database, you are the active subject and the database is the passive object. Figure 4-1 illustrates subjects and objects.

Access control is a broad term that covers several different types of mechanisms that enforce access control features on computer systems, networks, and information. Access control is extremely important because it is one of the first lines of defense used to fight against unauthorized access to systems and network resources. When a user is prompted for a username and password to be able to use a computer, this is access control. Once the user logs in and later attempts to access a file, that file may have a list of users and groups that have the right to access it. If the user is not on this list, the user is denied. This is another form of access control. The users' permissions and rights may be based on their identity, clearance, and/or group membership. Access controls give organizations the ability to control, restrict, monitor, and protect resource availability, integrity, and confidentiality.

Security Principles

The three main security principles for any type of security control are

- Availability
- Integrity
- Confidentiality

These principles, which were touched upon in Chapter 3, will be a running theme throughout this book because each core subject of each chapter approaches these prin-

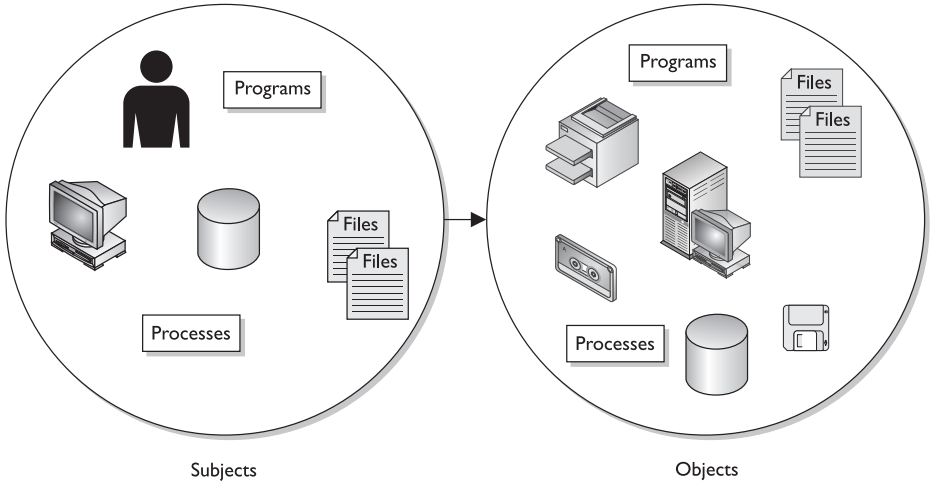


Figure 4-1 Subjects are active entities that access objects; objects are passive entities.

ciples in a unique way. In Chapter 3, you read that security management procedures include identifying threats that can negatively affect the availability, integrity, and confidentiality of the company's assets and finding cost-effective countermeasures that will protect them. This chapter looks at the ways that the three principles can be affected and protected through access control methodologies and technologies.

Every control that is used in computer and information security provides at least one of these security principles. It is critical that security professionals understand all of the possible ways that these principles can be provided and circumvented.

Availability

Hey, I'm available. Response: But no one wants you.

Information, systems, and resources need to be available to users in a timely manner so productivity will not be affected. Most information needs to be accessible and available to users when it is requested so that they can carry out tasks and fulfill their responsibilities. Accessing information does not seem that important until it is inaccessible. Administrators experience this when a file server goes offline or a highly used database is out of service for one reason or another. Fault tolerance and recovery mechanisms are put into place to ensure the continuity of the ***availability*** of resources. User productivity can be greatly affected if requested data is not readily available.

Information has various attributes, such as accuracy, relevance, timeliness, and privacy. It may be extremely important for a stockbroker to have information that is accurate and timely, so that he can buy and sell stocks at the right times at the right prices. The stockbroker may not necessarily care about the privacy of this information, only that it is readily available. A soft drink company that depends on its soda pop recipe would care about the privacy of this trade secret, and the security mechanisms in place need to ensure this secrecy.

Integrity

Information must be accurate, complete, and protected from unauthorized modification. When a security mechanism provides ***integrity***, it protects data, or a resource, from being altered in an unauthorized fashion. If some type of illegitimate modification does occur, the security mechanism must alert the user in some fashion. One example is when a user sends a request to her online bank account to pay her $24.56 water utility bill. The bank needs to be sure that the integrity of that transaction was not altered during transmission, so the user does not end up paying the utility company $240.56 instead. Integrity of data is very important. What if a confidential e-mail was sent from the Secretary of State to the President of the United States and was intercepted and altered without a security mechanism in place that disallows this or alerts the President that this message has been altered? Instead of receiving a message reading, "We would love for you and your wife to stop by for drinks tonight," the message could be altered to say, "We have just bombed Libya." Big difference.

Confidentiality

This is my secret and you can't have it. Response: I don't want it.

Confidentiality is the assurance that information is not disclosed to unauthorized individuals, programs, or processes. Some information is more sensitive than other information and requires a higher level of confidentiality. Control mechanisms need to be in place to dictate who can access data and what the subject can do with it once they have accessed it. These activities need to be controlled, audited, and monitored. Examples of information that could be considered confidential are health records, financial account information, criminal records, source code, trade secrets, and military tactical plans. Some security mechanisms that would provide confidentiality are encryption, logical and physical access controls, transmission protocols, database views, and controlled traffic flow.

It is important for a company to identify the data that needs to be classified, so that the company can ensure that a top priority of security protects this information and keeps it confidential. If this information is not singled out, too much time and money can be spent on implementing the same level of security for critical and mundane information alike. It may be necessary to configure virtual private networks (VPNs) between organizations and use the IPSec encryption protocol to encrypt all messages passed when communicating about trade secrets, sharing customer information, or making financial transactions. This takes a certain amount of hardware, labor, funds, and overhead. The same security precautions are not necessary when communicating that today's special in the cafeteria is liver and onions with a roll on the side. So, the first step in protecting data's confidentiality is to identify which information is sensitive and to what degree, and then implement security mechanisms to protect it properly.

Different security mechanisms can supply different degrees of availability, integrity, and confidentiality. The environment, the classification of the data that is to be protected, and the security goals need to be evaluated to ensure that the proper security mechanisms are bought and put into place. Many corporations have wasted a lot of time and money not following these steps and instead buying the new "gee whiz" product that recently hit the market.

Identification, Authentication, and Authorization

For a user to be able to access a resource, he first must prove that he is who he claims to be, has the necessary credentials, and has been given the necessary rights or privileges to perform the actions he is requesting. Once these steps are completed successfully, the user can access and use network resources; however, it is necessary to track the user's activities and enforce accountability for his actions. ***Identification*** describes a method of ensuring that a subject (user, program, or process) is the entity it claims to be. Identification can be provided with the use of a username or account number. To be properly ***authenticated***, the subject is usually required to provide a second piece to the credential set. This piece could be a password, passphrase, cryptographic key, personal identification number (PIN), anatomical attribute, or token. These two credential items are com-

Race Condition

A race condition is possible when two or more processes use a shared resource, as in data within a variable. It is important that the processes carry out their functionality in the correct sequence. If process 2 carried out its task on the data before process 1, the result will be much different than if process 1 carried out its tasks on the data before process 2. A race condition is when processes carry out their tasks on a shared resource in an incorrect order.

In software, when the authentication and authorization steps are split into two functions, there is a possibility that an attacker could use a race condition to force the authorization step to be completed *before* the authentication step. This would be a flaw in the software that the attacker has figured out how to exploit. A *race condition* is when two or more processes use the same resource and the sequences of steps within software can be carried out in improper order, which can drastically affect the output. So, an attacker can force the authorization step to take place before the authentication step and gain unauthorized access to a resource.

pared to information that has been previously stored for this subject. If these credentials match the stored information, the subject is authenticated. But we are not done yet.

Once the subject provides its credentials and is properly identified, the system it is trying to access needs to determine if this subject has been given the necessary rights and privileges to carry out the requested actions. The system will look at some type of access control matrix or compare security labels to verify that this subject may indeed access the requested resource and perform the actions it is attempting. If the system determines that the subject may access the resource, it ***authorizes*** the subject.

Although identification, authentication, and authorization have close and complementary definitions, each has distinct functions that fulfill a specific requirement in the process of access control. A user may be properly identified and authenticated to the network, but he may not have the authorization to access the files on the file server. On the other hand, a user may be authorized to access the files on the file server, but until she is properly identified and authenticated, those resources are out of reach. Figure 4-2 illustrates the three steps that must happen for a subject to access an object.

The subject needs to be held accountable for the actions taken within a system or domain. The only way to ensure accountability is if the subject is uniquely identified and the subject's actions are recorded.

Logical access controls are tools used for identification, authentication, authorization, and accountability. They are software components that enforce access control measures for systems, programs, processes, and information. The logical access controls can be embedded within operating systems, applications, add-on security packages, or database and telecommunication management systems. It can be challenging to synchronize all access controls and ensure that all vulnerabilities are covered without producing overlaps of functionality. However, if it were easy, security professionals would not be getting paid the big bucks!

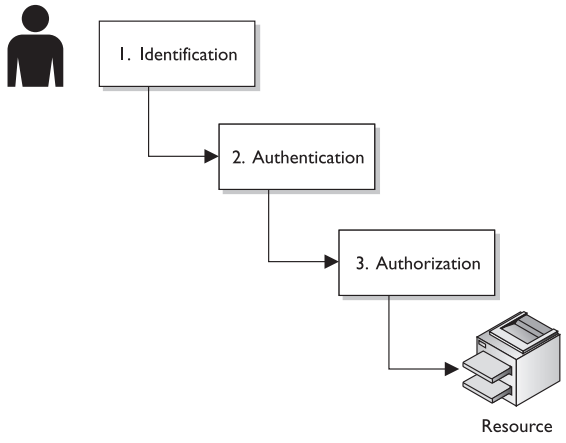


Figure 4-2 There are three steps that must happen for a subject to access an object: identification, authentication, and authorization.

NOTE The words "logical" and "technical" can be used interchangeably in this context. It is conceivable that the CISSP exam would refer to logical and technical controls interchangeably.

An individual's identity needs to be verified during the authentication process. Authentication usually involves a two-step process: entering public information (a username, employee number, account number, or department ID) and then entering private information (a static password, smart token, cognitive password, one-time password, PIN, or digital signature). Entering public information is the identification step and entering private information is the authentication step of the two-step process. Each technique used for identification and authentication has its pros and cons. Each should be properly evaluated to determine the right mechanism for the correct environment.

NOTE A cognitive password is based on a user's opinion or life experience. The password could be a mother's maiden name, a favorite color, or a dog's name.

References

- **Secure programmer: Prevent race conditions** www-128.ibm.com/developerworks/library-combined/l-sprace.html
- **"What Are Race Conditions and Deadlocks?" Microsoft Knowledge Base Article 317723** http://support.microsoft.com/default.aspx?scid=kb;en-us;Q317723

Identification and Authentication

Now, who are you again?

Once a person has been identified, through the user ID or a similar value, she must be authenticated, which means she must prove she is who she says she is. There are three general factors that can be used for authentication: something a person knows, something a person has, and something a person is.

Something a person knows can be, for example, a password, PIN, mother's maiden name, or combination to a lock. Authenticating a person by something that she knows is usually the least expensive to implement. The downside to this method is that another person may aquire this knowledge and gain unauthorized access to a system or facility.

Something a person has can be a key, swipe card, access card, or badge. This method is common for accessing facilities, but could also be used to access sensitive areas or to authenticate systems. A downside to this method is that the item can be lost or stolen, which could result in unauthorized access.

Something specific to a person becomes a bit more interesting. This is not based on if the person is a Republican, a martian, or a moron—it is based on a physical attribute. Authenticating a person's identity based on a unique physical attribute is referred to as biometrics. (For more information, see the upcoming section, "Biometrics.")

Strong authentication contains two out of these three methods: something a person knows, has, or is. Using a biometric system by itself does not provide strong authentication because it provides only one out of the three methods. Biometrics supplies what a person is, not what a person knows or has. For a strong authentication process to be in place, a biometric system needs to be coupled with a mechanism that checks for one of the other two methods. For example, many times the person has to type a PIN number into a keypad before the biometric scan is performed. This satisfies the "what the person knows" category. Conversely, the person could be required to swipe a magnetic card through a reader prior to the biometric scan. This would satisfy the "what the person has" category. Whatever identification system is used, for strong authentication to be in the process, it must include two out of the three categories. This is also referred to as *two-factor authentication*.

Identification Component Requirements

When issuing identification values to users, the following should be in place:

- Each value should be unique, for user accountability.
- A standard naming scheme should be followed.
- The value should be nondescriptive of the user's position or tasks.
- The value should not be shared between users.

Identity Management

There are too many of you who want to access too much stuff. Everyone just go away!

Identity management is a broad term that encompasses the use of different products to identify, authenticate, and authorize users through automated means. Selling identity management products is now a flourishing market that focuses on reducing administrative costs, increasing security, and improving upon service levels throughout enterprises. The continual increase in complexity and diversity of networked environments only increases the complexity of keeping track of who can access what and when. Users usually access several different types of systems throughout their daily tasks, which makes controlling access and providing the necessary level of protection on different data types difficult and full of obstacles. This complexity usually results in unforeseen and unidentified holes in asset protection, overlapping and contradictory controls, and policy and regulation noncompliance. It is the goal of identity management technologies to simplify the administration of these tasks and bring sanity to chaos.

The following are many of the common problems that enterprises deal with today in controlling access to assets:

- **Various types of users need different levels of access** Internal users, contractors, outsiders, partners, etc.
- **Resources have different classification levels** Confidential, internal use only, private, public, etc.
- **Diverse identity data must be kept on different types of users** Credentials, personal data, contact information, work-related data, digital certificates, cognitive passwords, etc.
- **The corporate environment is continually changing** Business environment needs, resource access needs, employee roles, actual employees, etc.

Despite all of these challenges, access control is expected to be consistent, efficient, transparent, reliable, and secure. (Sigh…)

The traditional identity management process has been manual, using directory services with permissions and profiles. This approach has proven incapable of keeping up with complex demands and thus has been replaced with the use of newly arrived automated applications that are rich in functionality, including enterprise-wide products and single sign-on solutions. The following are some of the services that these types of products supply:

- User provisioning
- Password synchronization and resetting
- Self service for users on specific types of activities
- Delegation of administrative tasks
- Centralized auditing and reporting
- Integrated workflow and increase in business productivity

- Decrease in network access points
- Regulatory compliance

The following sections explain the various types of authentication methods commonly used and integrated in many identity management processes and products today.

References

- **"Securing Identity: Part 3 – Infrastructure Versus Management," by META Group (March 13, 2003)** www.entrust.com/resources/pdf/metagroupreport.pdf
- **"Identity Management in the Real World," by Deborah Radcliff, CSOonline .com (Nov. 2004)** www.csoonline.com/read/110104/idmgmt.html

Biometrics

I would like to prove who I am. Please look at the blood vessels at the back of my eyeball. Response: Gross.

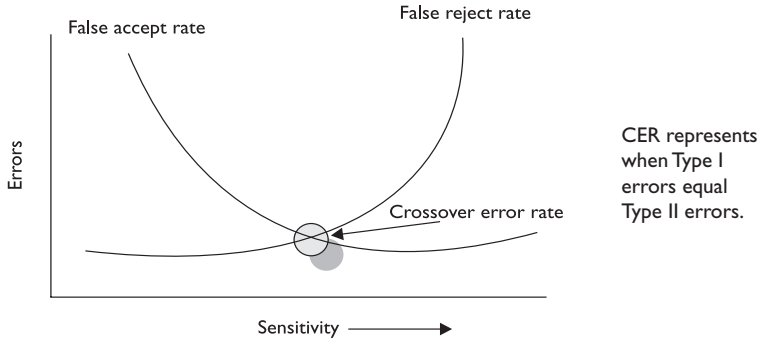
Biometrics verifies an individual's identity by analyzing a unique personal attribute or behavior, which is one of the most effective and accurate methods of verifying identification. Biometrics is a very sophisticated technology; thus, it is much more expensive and complex than the other types of identity verification processes. A biometric system can make authentication decisions based on an individual's behavior, as in signature dynamics, but these can change over time and possibly be forged. Biometric systems that base authentication decisions on physical attributes (iris, retina, fingerprint) provide more accuracy, because physical attributes typically don't change, absent some disfiguring injury, and are harder to impersonate.

A biometric system scans an attribute or behavior of a person and compares it to a record that was created in an earlier enrollment process. Because this system inspects the grooves of a person's fingerprint, the pattern of someone's retina, or the pitches of someone's voice, it has to be extremely sensitive. The system must perform accurate and repeatable measurements of anatomical or physiological characteristics. This type of sensitivity can easily cause false positives or false negatives. The system must be calibrated so that these false positives and false negatives occur infrequently and the results are as accurate as possible.

When a biometric system rejects an authorized individual, it is called a ***Type I error*** (false rejection rate). When the system accepts impostors who should be rejected, it is called a ***Type II error*** (false acceptance rate). The goal is to obtain low numbers for each type of error, but Type II errors are the most dangerous and thus the most important to avoid.

When comparing different biometric systems, many different variables are used, but one of the most important metrics is the ***crossover error rate (CER)***. This rating is stated as a percentage and represents the point at which the false rejection rate equals the false acceptance rate. This rating is the most important measurement when determining

the system's accuracy. A biometric system that delivers a CER of 3 will be more accurate than a system that delivers a CER of 4.



NOTE Crossover error rate (CER) is also called equal error rate (EER).

What is the purpose of this CER value anyway? Using the CER as an impartial judgment of a biometric system helps to create standards by which products from different vendors can be fairly judged and evaluated. If you are going to buy a biometric system, you need a way to compare the accuracy between different systems. You can just go by the different vendors' marketing material (they all say they are the best), or you can compare the different CER values of the products to see which one really is more accurate than the others. It is also a way to keep the vendors honest. One vendor may tell you, "We have absolutely no Type II errors." This would mean that their product would not allow any imposters to be improperly authenticated. But what if you asked the vendor how many Type I errors their product had and she sheepishly replied, "We average around 90 percent of Type I errors." That would mean that 90 percent of the authentication attempts would be rejected, which would negatively affect your employees' productivity. So you can ask about their CER value, which represents when the Type I and Type II errors are equal, to give you a better understanding of the product's overall accuracy.

Individual environments have specific security level requirements, which will dictate how many Type I and Type II errors are acceptable. For example, a military institution that is very concerned about confidentiality would be prepared to accept a certain number of Type I errors, but would absolutely not accept any false accepts (Type II errors). Because all biometric systems can be calibrated, if you lower the Type II error rate by adjusting the system's sensitivity, it will result in an increase in Type I errors. The military institution would obviously calibrate the biometric system to lower the Type II errors to zero, but that would mean that it would have to accept a higher rate of Type I errors.

Biometrics is the most expensive method of verifying a person's identity, and it faces other barriers to becoming widely acceptable. These include user acceptance, enrollment timeframe, and throughput. Many times people are reluctant to let a machine

Processing Speed

When reviewing biometric devices for purchase, one component to take into consideration is the length of time it takes to actually authenticate users. From the time a user inserts data until she receives an accept or reject response should take five to ten seconds.

read the pattern of their retina or scan the geometry of their hand. This lack of enthusiasm has slowed down the widespread use of biometric systems within our society. The enrollment phase requires an action to be performed several times to capture a clear and distinctive reference record. People are not particularly fond of expending this time and energy when they are used to just picking a password and quickly typing it into their console. When a person attempts to be authenticated by a biometric system, sometimes the system will request an action to be completed several times. If the system was unable to get a clear reading of an iris scan or could not capture a full voice verification print, the individual may have to repeat the action. This causes low throughput, stretches the individual's patience, and reduces acceptability.

There are many types of biometric systems that examine different personal characteristics. With each type of system, the individual must go through an enrollment process, which captures the biometric data and stores it in a reference file. This reference file is used later when the person attempts to be authenticated.

The following is an overview of the different types of biometric systems and the physiological characteristics they examine.

Fingerprint Fingerprints are made up of ridge endings and bifurcations exhibited by the friction ridges and other detailed characteristics that are called *minutiae.* It is the distinctiveness of these minutiae that gives each individual a unique fingerprint. An individual places his finger on a device that reads the details of the fingerprint and compares this to a reference file. If the two match, the individual's identity has been verified.

NOTE Fingerprint systems store the full fingerprint, which is actually a lot of information that takes up hard drive space and resources. The finger-scan technology extracts specific features from the fingerprint and stores just that information, which takes up less hard drive space and allows for quicker database lookups and comparisons.

Palm Scan The palm holds a wealth of information, and has many aspects that are used to identify an individual. The palm has creases, ridges, and grooves throughout that are unique to a specific person. The palm scan also includes the fingerprints of each finger. An individual places his hand on the biometric device, which scans and captures this information. This information is compared to a reference file and the identity is either verified or rejected.

Hand Geometry The shape of a person's hand (the length and width of the hand and fingers) defines hand geometry. This trait differs significantly between people and is used in some biometric systems to verify identity. A person places her hand on a device that has grooves for each finger. The system compares the geometry of each finger, and the hand as a whole, to the information in a reference file to verify that person's identity.

Retina Scan A system that reads a person's retina scans the blood-vessel pattern of the retina on the backside of the eyeball. This pattern has shown to be extremely unique between different people. A camera is used to project a beam inside the eye and capture the pattern and compare it to a reference file recorded previously.

Iris Scan The iris is the colored portion of the eye that surrounds the pupil. The iris has unique patterns, rifts, colors, rings, coronas, and furrows. The uniqueness of each of these characteristics within the iris is captured by a camera and compared with the information gathered during the enrollment phase.

NOTE When using an iris pattern biometric system, the optical unit must be positioned so that the sun does not shine into the aperture; thus, when implemented, it must have proper placement within the facility.

Signature Dynamics When a person signs a signature, usually they do so in the same manner and speed each time. Signing a signature produces electrical signals that can be captured by a biometric system. The physical motions performed when someone is signing a document create these electrical signals. The signals provide unique characteristics that can be used to distinguish one individual from another. Signature dynamics provides more information than a static signature, so there are more variables to verify when confirming an individual's identity and more assurance that this person is who he claims to be.

Signature dynamics is different from a digitized signature. A digitized signature is just an electronic copy of someone's signature and is not a biometric system that captures the speed of signing, the way the person holds the pen, and the pressure the signer exerts to generate the signature.

Keyboard Dynamics Whereas signature dynamics is a method that captures the electrical signals when a person signs a name, keyboard dynamics captures electrical signals when a person types a certain phrase. As a person types a specified phrase, the biometric system captures the speed and motions of this action. Each individual has a certain style and speed, which translate into unique signals. This type of authentication is more effective than typing in a password, because a password is easily obtainable. It is much harder to repeat a person's typing style than it is to acquire a password.

Voice Print There are many subtle distinguishing differences in people's speech sounds and patterns. A biometric system that is programmed to capture a voice print

and compare it to the information captured in a reference file can differentiate one individual from another. During the enrollment process, an individual is asked to say several different words. Later, when this individual needs to be authenticated, the biometric system jumbles these words and presents them to the individual. The individual then repeats the sequence of words given. This is used as a deterrent so that others will not attempt to record the session and play it back in hopes of obtaining unauthorized access.

Facial Scan A system that scans a person's face takes many attributes and characteristics into account. People have different bone structures, nose ridges, eye widths, forehead sizes, and chin shapes. These are all captured during a facial scan and compared to an earlier captured scan held within a reference record. If the information is a match, the person is positively identified.

Hand Topography Whereas hand geometry looks at the size and width of an individual's hand and fingers, hand topology looks at the different peaks and valleys of the hand, along with its overall shape and curvature. When an individual wants to be authenticated, she places her hand on the system. Off to one side of the system, a camera snaps a side-view picture of the hand from a different view and angle than that of systems that target hand geometry, and thus captures different data. This attribute is not unique enough to authenticate individuals by itself and is commonly used in conjunction with hand geometry.

References

- **Michigan State University Biometrics Research web site** biometrics.cse.msu.edu
- **The Biometric Consortium home page** www.biometrics.org

Passwords

User identification coupled with a reusable password is the most common form of system identification and authorization mechanisms. A password is a protected string of characters that is used to authenticate an individual. As stated previously, authentication factors are based on what a person knows, has, or is. A password is something the user knows.

Passwords are one of the most used authentication mechanisms employed today. It is important that the passwords are strong and properly managed.

Password Management Although passwords are the most commonly used authentication mechanisms, they are also considered one of the weakest security mechanisms available. Why? Users usually choose passwords that are easily guessed (spouse's name, user's birth date, or dog's name), tell others their passwords, and many times write the passwords down on a sticky note and cleverly hide it under the keyboard. To

most users, security is usually not the most important or interesting part of using their computers—except when someone hacks into their computer and steals confidential information. Then security is all the rage.

This is where password management steps in. If passwords are properly generated, updated, and kept secret, they can provide effective security. Password generators can be used to create passwords for users. This ensures that a user will not be using "Bob" or "Spot" for a password, but if the generator spits out "kdjasijew284802h," the user will surely scribble it down on a piece of paper and safely stick it to the monitor, which defeats the whole purpose. If a password generator is going to be used, the tools should create uncomplicated, pronounceable, non-dictionary words to help users remember them so that they aren't tempted to write them down.

If the users can choose their own passwords, the operating system should enforce certain password requirements. The operating system can require that a password contain a certain number of characters, unrelated to the user ID, include special characters, include upper- and lowercase letters, and not be easily guessable. The operating system can keep track of the passwords a specific user generates, to ensure that no passwords are reused. The users should also be forced to change their passwords periodically. All of these factors make it harder for an attacker to guess or obtain passwords within the environment.

If an attacker is after a password, she can try a few different techniques:

- **Electronic monitoring** Listening to network traffic to capture information, especially when a user is sending her password to an authentication server. The password can be copied and reused by the attacker at another time, which is called a *replay attack*.
- **Access the password file** Usually done on the authentication server. The password file contains many users' passwords and, if compromised, can be the source of a lot of damage. This file should be protected with access control mechanisms and encryption.
- **Brute force attacks** Performed with tools that cycle through many possible character, number, and symbol combinations to uncover a password.
- **Dictionary attacks** Files of thousands of words are used to compare to the user's password until a match is found.
- **Social engineering** An attacker falsely convinces an individual that she has the necessary authorization to access specific resources.

There are techniques that can be implemented to provide another layer of security for passwords and their use. After each successful logon, a message can be presented to a user indicating the date and time of the last successful logon, the location of this logon, and if there were any unsuccessful logon attempts. This alerts the user if suspicious activity has been going on, and if anyone has attempted to log on using his credentials. An administrator can set operating parameters that allow a certain number of failed logon attempts to be accepted before a user is locked out; this is a type of *clipping*

level. The user can be locked out for five minutes or a full day after the threshold (or clipping level) has been exceeded. It depends on how the administrator configures this mechanism. An audit trail can also be used to track password usage and successful and unsuccessful logon attempts. This audit information should include the date, time, user ID, and workstation the user logged on from.

A password's lifetime should be short but practical. Forcing a user to change a password on a more frequent basis provides more assurance that the password will not be guessed by an intruder. If the lifetime is too short, however, it causes unnecessary management overhead and users may forget which password is active. A balance between protection and practicality needs to be decided upon and enforced.

As with many things in life, education is the key. Password requirements, protection, and generation should be addressed in security-awareness programs so that users understand what is expected of them, why they should protect their passwords, and how passwords can be stolen. Users should be an extension to a security team, not the opposition.

Password Checkers Several organizations test user-chosen passwords using tools that perform dictionary and/or brute force attacks to detect the weak passwords. This helps make the environment as a whole less susceptible to dictionary and exhaustive attacks that are used to discover users' passwords. Many times the same tools used to attempt to crack a password are used by a network administrator to make sure the password is strong enough. Most security tools have this dual nature. They are used by security professionals and IT staff to test for vulnerabilities within their environment in the hope of uncovering them and fixing them before an attacker finds the vulnerabilities. An attacker uses the same tools to uncover vulnerabilities to exploit before the security professional can fix them. It is the never-ending cat-and-mouse game.

If a tool is called a *password checker*, it is a tool used by a security professional to test the strength of a password. If a tool is called a *password cracker,* it is usually used by a hacker; however, most of the time, these tools are one and the same.

You need to obtain management's approval before attempting to test (break) employees' passwords with the intent of identifying weak passwords. Explaining that you are trying to help the situation, not hurt it, *after* you have uncovered the CEO's password is not a good situation to be in.

Password Hashing and Encryption In most situations, if an attacker sniffs your password from the network wire, she still has some work to do before she actually knows your password value, because most systems hash the password with a hashing algorithm, commonly MD4 or MD5, to ensure that passwords are not sent in cleartext.

NOTE The hashing process and hashing algorithms will be covered in Chapter 8.

In a Windows environment the passwords are stored in a Security Accounts Management (SAM) database in their hashed version. For extra protection, administrators can use a Syskey utility, which encrypts the database that stores the passwords with a locally stored system key. Syskey can work in three modes, each increasing in the protection provided:

- **Mode 1** System key is generated, encrypted, and stored locally. Computer can restart and work normally with no user interaction.
- **Mode 2** System key is generated, encrypted, and stored locally but is password protected. When the computer restarts, the administrator must enter the password to "unlock Syskey," and this password is not stored locally.
- **Mode 3** System key is generated, encrypted, and stored on a floppy disk or CD-ROM. The computer cannot start up properly without a user providing the floppy disk.

Although some people think the world is run by Microsoft, there are also other types of operating systems out there, such as Unix and Linux. These systems do not use registries and SAM databases, but contain their user passwords in a file cleverly called passwd. Now, this passwd file does not contain passwords in cleartext; instead, your password is used to encrypt a block of bits with a one-way function and the resulting value is stored in this file. Unix-type systems zest things up by using salts in this process. *Salts* are random values that are added to the encryption process to add more complexity. The more randomness entered into the encryption process, the harder it is for the bad guy to decrypt and uncover your password. The use of a salt means that the same password can be encrypted into 4096 different formats, which makes it much more difficult for an attacker to uncover the *right* format for your system.

Password Aging Many systems enable administrators to set expiration dates for passwords, forcing users to change them at regular intervals. The system may also keep a list of the last five to ten passwords (password history) and not let the users revert back to previously used passwords.

Limit Logon Attempts A threshold can be set to allow only a certain number of unsuccessful logon attempts. After the threshold is met, the user's account can be locked for a period of time or indefinitely, which requires an administrator to manually unlock the account. This protects against dictionary and other exhaustive attacks that continually submit credentials until the right combination of username and password is discovered.

Cognitive Passwords

What is your mother's name? Response: Shucks, I don't remember. I have it written down somewhere.

Cognitive passwords are fact- or opinion-based information used to verify an individual's identity. A user is enrolled by answering several questions based on her life experiences. Passwords can be hard for people to remember, but that same person will

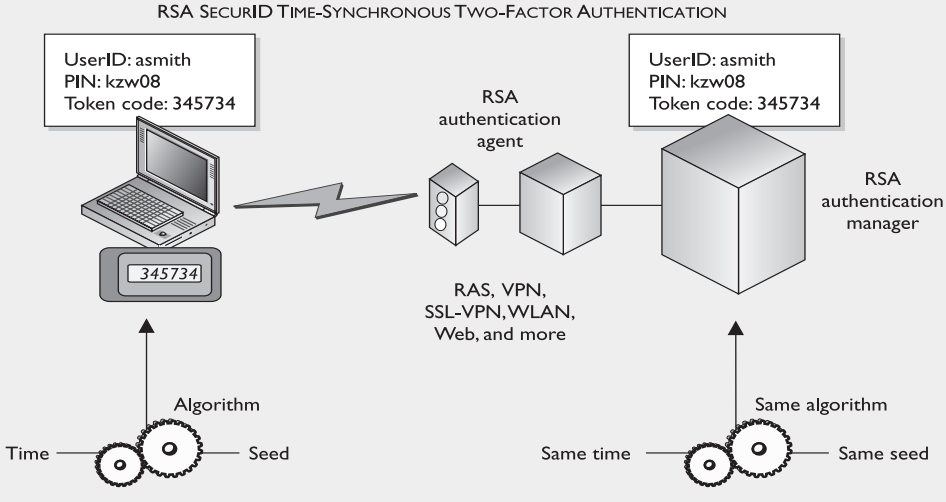
not likely forget her mother's maiden name, favorite color, dog's name, or the school she graduated from. After the enrollment process, the user can answer the questions asked of her to be authenticated, instead of having to remember a password. This authentication process is best for a service the user does not use on a daily basis because it takes longer than other authentication mechanisms. This can work well for help-desk services. The user can be authenticated via cognitive means. This way, the person at the help desk can be sure he is talking to the right person, and the user in need of help does not need to remember a password that may be used once every three months.

One-Time Passwords

A ***one-time password*** is also called a dynamic password. It is used for authentication purposes and is only good once. After the password is used, it is no longer valid; thus, if a hacker obtained this password, it could not be reused. This type of authentication mechanism is used in environments that require a higher level of security than static passwords provide. There are two general types of one-time password generating tokens: synchronous and asynchronous. The token device generates the one-time password for the user to submit to an authentication server. The following sections explain these concepts.

SecureID

SecureID, from RSA Security, Inc., is one of the most widely used time-based tokens. One version of the product generates the one-time password by using a mathematical function on the time, date, and ID of the token card. Another version of the product requires a PIN to be entered into the token device.



Token Device The *token device*, or password generator, is usually a handheld device that has an LCD display and possibly a keypad. This hardware is separate from the computer the user is attempting to access. The token device and authentication service need to be synchronized in some manner to be able to authenticate a user. The token device presents the user with a list of characters to be entered as a password when logging onto a computer. Only the token device and authentication service know the meaning of these characters. Because the two are synchronized, the token device will present the exact password the authentication service is expecting. This is a one-time password, also called a token, and is no longer valid after initial use.

Synchronous A *synchronous token device* synchronizes with the authentication service by using time or a counter as the core piece of the authentication process. If the synchronization is *time based*, the token device and the authentication service must hold the same time within their internal clocks. The time value on the token device and a secret key are used to create the one-time password, which is displayed to the user. The user enters this value and a user ID into the computer, which then passes them to the server running the authentication service. The authentication service decrypts this value and compares it to the value that it expected. If the two match, the user is authenticated and allowed to use the computer and resources.

If the token device and authentication service use *counter-synchronization*, the user will need to initiate the logon sequence on the computer and push a button on the token device. This causes the token device and the authentication service to advance to the next authentication value. This value and a base secret are hashed and displayed to the user. The user enters this resulting value along with a user ID to be authenticated. In either time- or counter-based synchronization, the token device and authentication service must share the same secret base key used for encryption and decryption.

Asynchronous A token device that is using an ***asynchronous token***-generating method uses a challenge/response scheme to authenticate the user. In this situation, the authentication server sends the user a challenge, a random value also called a *nonce*. The user enters this random value into the token device, which encrypts it and returns a value that the user uses as a one-time password. The user sends this value, along with a username, to the authentication server. If the authentication server can decrypt the value and it is the same challenge value that was sent earlier, the user is authenticated, as shown in Figure 4-3.

NOTE The actual implementation and process that these devices follow can differ between different vendors. What is important to know is that asynchronous is based on challenge/response mechanisms and synchronous is based on time- or counter-driven mechanisms.

Both token systems can fall prey to masquerading if a user shares his identification information (ID or username) and the token device is shared or stolen. The token device can also have battery failure or other malfunctions that would stand in the way of

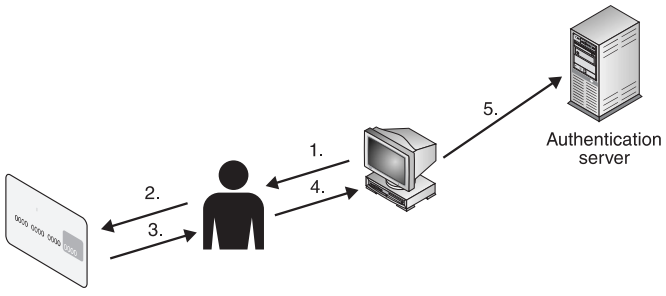


Figure 4-3 Authentication using an asynchronous token device includes a workstation, token device, and authentication service.

1. Challenge value displayed on workstation.
2. User enters challenge value and PIN into token device.
3. Token device presents a different value to the user.
4. User enters new value into the workstation.
5. Value sent to authentication service on server.
6. Authentication service is expecting a specific value.
7. User is authenticated and allowed access to workstation.

a successful authentication. However, this type of system is not vulnerable to electronic eavesdropping, sniffing, or password guessing.

If the user has to enter a password or PIN into the token device before it provides a one-time password, then strong authentication is in effect because it is using two factors—something the user knows (PIN) and something the user has (the token device).

NOTE One-time passwords can also be generated in software, in which case a piece of hardware such as a token device is not required. These are referred to as *soft tokens* and require that the authentication service and application contain the same base secrets, which are used to generate the one-time passwords.

References

- **"A One-Time Password System," by N. Haller, et al., IETF Network Working Group (Feb. 1998)** www.ietf.org/rfc/rfc2289.txt
- **RSA SecurID Authentication home page** www.rsasecurity.com/node.asp?id=1156
- **"A Token of Our Esteem," by Timothy O'Shea (Sept. 1999)** www.networkcomputing.com/1018/1018f1.html

Cryptographic Keys

Another way to prove one's identity is to use a private key or generate a digital signature. A private key or digital signature could be used in place of using a password. Passwords are the weakest form of authentication and can be easily sniffed as they travel over a network. Private keys and digital signatures are forms of authentication

used in environments that require higher security protection than what is provided by passwords.

A private key is a secret value that should be in the possession of one person, and one person only. It should never be disclosed to an outside party. A digital signature is a technology that uses a private key to encrypt a hash value (message digest). The act of encrypting this hash value with a private key is called *digitally signing* a message. A digital signature attached to a message proves that the message originated from a specific source, and that the message itself was not changed while in transit.

A public key can be made available to anyone without compromising the associated private key; this is why it is called a public key. We explore private keys, public keys, digital signatures, and public key infrastructure (PKI) in Chapter 8, but for now, understand that a private key and digital signatures are other mechanisms that can be used to authenticate an individual.

Passphrase

A ***passphrase*** is a sequence of characters that is longer than a password (thus a "phrase") and, in some cases, takes the place of a password during an authentication process. The user enters this phrase into an application and the application transforms the value into a ***virtual password***, making the passphrase the length and format that is required by the application. (For example, an application may require your virtual password to be 64 bits to be used as a key with the AES algorithm.) If a user wants to authenticate to an application, such as Pretty Good Privacy (PGP), he types in a passphrase, let's say StickWithMeKidAndYouWillWearDiamonds. The application converts this phrase into a virtual password that is used for the actual authentication. The user usually generates the passphrase in the same way that a user creates a password the first time he logs onto a computer. A passphrase is more secure than a password because it is longer, and thus harder to obtain by an attacker. In many cases the user is more likely to remember a passphrase than a password.

Memory Cards

The main difference between memory cards and smart cards is their capacity to process information. A ***memory card*** holds information but cannot process information. A ***smart card*** holds information and has the necessary hardware and software to actually process that information. A memory card can hold a user's authentication information, so that the user only needs to type in a user ID or PIN and present the memory card, and if the data that the user entered matches the data on the memory card, the user is successfully authenticated. If the user presents a PIN value, then this is an example of two-factor authentication—something the user knows, and something the user has. A memory card can also hold identification data that is pulled from the memory card by a reader. It travels with the PIN to a back-end authentication server. An example of a memory card is a swipe card that must be used for an individual to be able to enter a building. The user enters a PIN and swipes the memory card through a card reader. If this is the correct combination, the reader flashes green and the individual can open the door and enter the building. Another example is an ATM card. If Buffy wants to withdraw $40 from her checking account, she needs to enter the correct PIN and slide the ATM card (or memory card) through the reader.

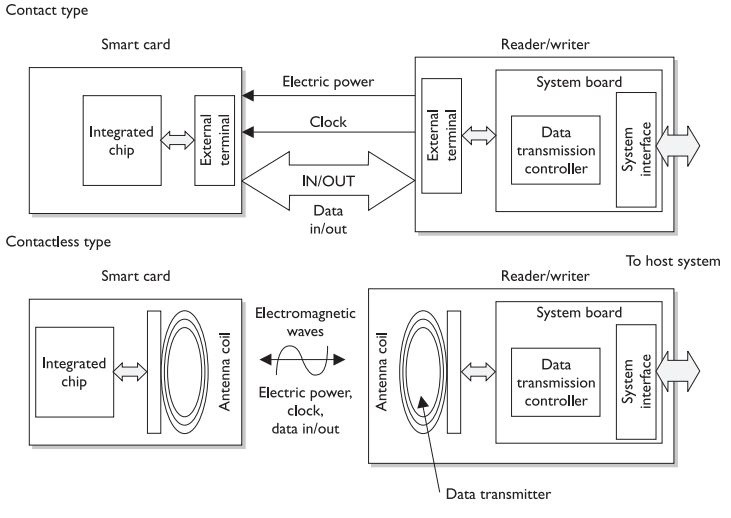
Memory cards can be used with computers, but they require a reader to process the information. The reader adds cost to the process, especially when one is needed per computer, and card generation adds additional cost and effort to the whole authentication process. Using a memory card provides a more secure authentication method than using a password because the attacker would need to obtain the card and know the correct PIN. Administrators and management need to weigh the costs and benefits of a memory token-based card implementation to determine if it is the right authentication mechanism for their environment.

Smart Card

My smart card is smarter than your memory card.

A smart card has the capability of processing information because it has a microprocessor and integrated circuits incorporated into the card itself. Memory cards do not have this type of hardware and lack this type of functionality. The only function they can perform is simple storage. A smart card, which adds the capability to process information that is stored on it, can also provide a two-factor authentication method because the user may have to enter a PIN to unlock the smart card. This means the user must provide something she knows (PIN) and something she has (smart card).

Two general categories of smart cards are the contact and the contactless types. The ***contact*** smart card has a gold seal on the face of the card. When this card is fully inserted into a card reader, electrical fingers wipe against the card, in the exact position that the chip contacts are located. This will supply power and data I/O to the chip for authentication purposes. The ***contactless*** smart card has an antenna wire that surrounds the perimeter of the card. When this card comes within an electromagnetic field of the reader, the antenna within the card generates enough energy to power the internal chip. Now, the results of the smart card processing can be broadcast through the same antenna, and the conversation of authentication can take place. The authentication can be completed by using a one-time password, by using a challenge/response value, or by providing the user's private key if it is used within a PKI environment.



Contact and contactless smart card architectures

NOTE There are two types of contactless smart cards: hybrid and combi. The hybrid card has two chips, with the capability of utilizing both the contact and contactless formats. A combi card has one microprocessor chip that can communicate to contact or contactless readers.

The information held within the memory of a smart card is not readable until the correct PIN is entered. This fact and the complexity of the smart token make these cards resistant to reverse-engineering and tampering methods. If George loses the smart card he uses to authenticate to the domain at work, the person who finds the card would need to know his PIN to be able to do any real damage. The smart card can also be programmed to store information in an encrypted fashion, as well as detect any tampering with the card itself. In the event that tampering is detected, the information stored on the smart card can be automatically wiped.

The drawbacks to using a smart card are the extra cost of the readers and the overhead of card generation, as with memory cards, although this cost is decreasing. The smart cards themselves are more expensive than memory cards because of the extra integrated circuits and microprocessor.

Smart Card Attacks Smart cards are more tamperproof than memory cards, but where there is sensitive data there are individuals who are motivated to circumvent any countermeasure that the industry throws at them.

Over the years people have become very inventive in the development of various ways to attack smart cards. For example, individuals have introduced computational errors into smart cards with the goal of uncovering the encryption keys that are being used and stored on the cards. These "errors" are introduced by manipulating some environmental component of the card (changing input voltage, clock rate, temperature fluctuations). The attacker reviews the result of an encryption function after introducing an error to the card and also reviews the correct result, which the card performs when no errors are introduced. Analysis of these different results allows an attacker to reverse-engineer the encryption process, with the hopes of uncovering the encryption key. This type of attack is referred to as ***fault generation***.

Side-channel attacks are nonintrusive and are used to uncover sensitive information about how a component works without trying to compromise any type of flaw or weakness. As an analogy, suppose that you want to figure out what your boss does each day at lunch time but you feel too uncomfortable to ask her. So you follow her, and you see that she enters a building holding a small black bag and exits exactly 45 minutes later with the same bag and her hair not looking as great as when she went in. You keep doing this day after day and come to the conclusion that she must be working out. Now you could have just read the sign on the building that said "Gym," but we will give you the benefit of the doubt here but not call you for any further private investigator work.

So a noninvasive attack is one in which the attacker watches how something works and how it reacts in different situations instead of trying to "invade" it with more intrusive measures. Some examples of side channel attacks that have been carried out on smart cards are *differential power analysis* (examining the power emissions that are released

Interoperability

An ISO/IEC standard, 14443, outlines the following items for smart card standardization:

- **ISO/IEC 14443-1** Physical characteristics
- **ISO/IEC 14443-2** Radio frequency power and signal interface
- **ISO/IEC 14443-3** Initialization and anticollision
- **ISO/IEC 14443-4** Transmission protocol

In the industry today, lack of interoperability is a big problem. Although vendors claim to be "compliant with ISO/IEC 14443," many have developed technologies and methods in a more proprietary fashion. The lack of true standardization has caused some large problems because smart cards are being used for so many different applications. In the United States, the DoD is rolling out smart cards across all of their agencies, and NIST is developing a framework and conformance testing programs specifically for interoperability issues.

during processing), *electromagnetic analysis* (examining the frequencies that are emitted), and timing (how long a specific process takes to complete).

Software attacks are also considered noninvasive attacks. A smart card has software just like any other device that does data processing, and anywhere there is software there is the possibility of software flaws that can be exploited. The main goal of this type of attack is to input instructions into the card that will allow for the attacker to extract account information, which he can use to make fraudulent purchases. Many of these types of attacks can be disguised by using equipment that looks just like the legitimate reader.

If you would like to be more intrusive in your smart card attack, you could give ***microprobing*** a try. Microprobing uses needles to remove the outer protective material on the card's circuits, by using ultrasonic vibration. Once this is completed, then data can be accessed and manipulated by directly tapping into the card's ROM chips.

References

- **NIST Smart Card Standards and Research web page** http://smartcard.nist.gov/
- **Smart Card Alliance home page** www.smartcardalliance.org
- **"Smart Cards: A Primer," by Rinaldo Di Giorgio, *JavaWorld* (Dec. 1997)** www.javaworld.com/javaworld/jw-12-1997/jw-12-javadev.html
- **"What Is a Smart Card," HowStuffWorks.com** http://electronics.howstuffworks.com/question332.htm

- *Securing Java: Getting Down to Business with Mobile Code*, **Chapter 8, Section 5, "How Secure Are Smart Cards?" by Edward Felten and Gary McGraw (Wiley, 1999)** www.securingjava.com/chapter-eight/chapter-eight-5.html

Authorization

Now that I know who you are, let's see if I will let you do what you want.

Although authentication and authorization are quite different, together they comprise a two-step process that determines whether an individual is allowed to access a particular resource. In the first step, authentication, the individual must prove to the system that he is who he claims to be—a permitted system user. After successful authentication, the system must establish whether the user is authorized to access the particular resource and what actions he is permitted to perform on that resource.

Authorization is a core component of every operating system, but applications, security add-on packages, and resources themselves can also provide this functionality. For example, suppose that Marge has been authenticated through the authentication server and now wants to view a spreadsheet that resides on a file server. When she finds this spreadsheet and double-clicks on the icon, she will see an hourglass instead of a mouse pointer. At this stage, the file server is seeing if Marge has the rights and permissions to view the requested spreadsheet. It also checks to see if Marge can modify, delete, move, or copy the file. Once the file server searches through an access matrix and finds that Marge does indeed have the necessary rights to view this file, the file opens up on Marge's desktop. The decision of whether or not to allow Marge to see this file was based on access criteria. Access criteria is the crux of authentication.

Access Criteria

You can perform that action only because we like you, and you wear a funny hat.

We have gone over the basics of access control. This subject can get very granular in its level of detail when it comes to dictating what a subject can or cannot do to an object or resource. This is a good thing for network administrators and security professionals, because they want to have as much control as possible over the resources they have been put in charge of protecting, and a fine level of detail enables them to give individuals just the precise level of access that they need. It would be frustrating if access control permissions were based only on full control or no access. These choices are very limiting, and an administrator would end up giving everyone full control, which would provide no protection. Instead, there are different ways of limiting access to resources and, if they are understood and used properly, they can give just the right level of access desired.

Granting access rights to subjects should be based on the level of trust a company has in a subject and the subject's need to know. Just because a company completely trusts Joyce with its files and resources does not mean she fulfills the need-to-know criteria to access the company's tax returns and profit margins. If Maynard fulfills the need-to-know criteria to access employees' work histories, it does not mean that the

company trusts him to access all of the company's other files. These issues need to be identified and integrated into the access criteria. The different access criteria can be broken up into roles, groups, location, time, and transaction types.

Using ***roles*** is an efficient way to assign rights to a type of user who performs a certain task. This role is based on a job assignment or function. If there is a position within a company for a person to audit transactions and audit logs, the role that this person would fill would only need a read function to those types of files. This role would not need full control, modify, or delete privileges.

Using ***groups*** is another effective way of assigning access control rights. If several users require the same type of access to information and resources, putting them into a group and then assigning rights and permissions to that group is easier to manage than assigning rights and permissions to each and every individual separately. If a specific printer is available only to the accounting group, when a user attempts to print to it, the group membership of the user will be checked to see if she is indeed in the accounting group. This is one way that access control is enforced through a logical access control mechanism.

Physical or logical location can also be used to restrict access to resources. Some files may be available only to users who can log on interactively to a computer. This means the user must be physically at the computer and enter the credentials locally versus logging on remotely from another computer. This restriction is implemented on several server configurations to restrict unauthorized individuals from being able to get in and reconfigure the server remotely.

Logical location restrictions are usually done through network address restrictions. If a network administrator wants to ensure that status requests of an intrusion detection management console are accepted only from certain computers on the network, the network administrator can configure this within the software.

Time of day, or temporal isolation, is another access control mechanism that can be used. If a security professional wants to ensure that no one is accessing payroll files between the hours of 8:00 P.M. and 4:00 A.M., that configuration can be implemented to ensure that access at these times is restricted. If the same security professional wants to ensure that no bank account transactions happen during days on which the bank is not open, she can indicate in the logical access control mechanism that this type of action is prohibited on Sundays.

Transaction-type restrictions can be used to control what data is accessed during certain types of functions and what commands can be carried out on the data. An online banking program may allow a customer to view his account balance, but may not allow the customer to transfer money until he has a certain security level or access right. A bank teller may be able to cash checks up to $2000, but would need a supervisor's access code to retrieve more funds for a customer. A database administrator may be able to build a database for the human resources department, but may not be able to read certain confidential files within that database. These are all examples of transaction-type restrictions to control access to data and resources.

Authorization Creep

As employees work at a company over time and move from one department to another, they often are assigned more and more access rights and permissions. This is commonly referred to as *authorization creep*. It can be a large risk for a company, because too many users have too much privileged access to company assets. Users' access needs and rights should be periodically reviewed to ensure that the principle of least privilege is being properly enforced.

Default to No Access

If you're unsure, just say no.

Access control mechanisms should default to no access, to provide the necessary level of security and ensure that no security holes go unnoticed. A wide range of access levels is available to assign to individuals and groups, depending on the application and/or operating system. A user can have read, change, delete, full control, or no access permissions. The statement that security mechanisms should default to no access means that if nothing has been specifically configured for an individual or the group she belongs to, that user should not be able to access that resource. If access is not explicitly allowed, it should be implicitly denied. Security is all about being safe, and this is the safest approach to practice when dealing with access control methods and mechanisms. In other words, all access controls should be based on the concept of starting with zero access, and building on top of that. Instead of giving access to everything, and then taking away privileges based on need to know, the better approach is to start with nothing and add privileges based on need to know.

Most access control lists (ACLs) that work on routers and packet-filtering firewalls default to no access. Figure 4-4 shows that traffic from Subnet A is allowed to access Subnet B, traffic from Subnet D is not allowed to access Subnet A, and Subnet B is allowed to talk to Subnet A. All other traffic transmission paths not listed here are not allowed by default. Subnet D cannot talk to Subnet B because such access is not explicitly indicated in the router's ACL.

Need to Know

If you need to know, I will tell you. If you don't need to know, leave me alone.

The ***need-to-know*** principle is similar to the ***least-privilege*** principle. It is based on the concept that individuals should be given access only to the information that they absolutely require in order to perform their job duties. Giving any more rights to a user just asks for headaches and the possibility of that user abusing the permissions assigned to him. An administrator wants to give a user the least amount of privileges she can, but just enough for that user to be productive when carrying out tasks. Management will decide what a user needs to know, or what access rights are necessary, and the administrator will configure the access control mechanisms to allow this user to have that level of access and no more, and thus the least privilege.

For example, if management has decided that Dan the copy boy needs to know where the files that he needs to copy are located and needs to know how to print them, this fulfills Dan's need-to-know criteria. Now, an administrator could give Dan full

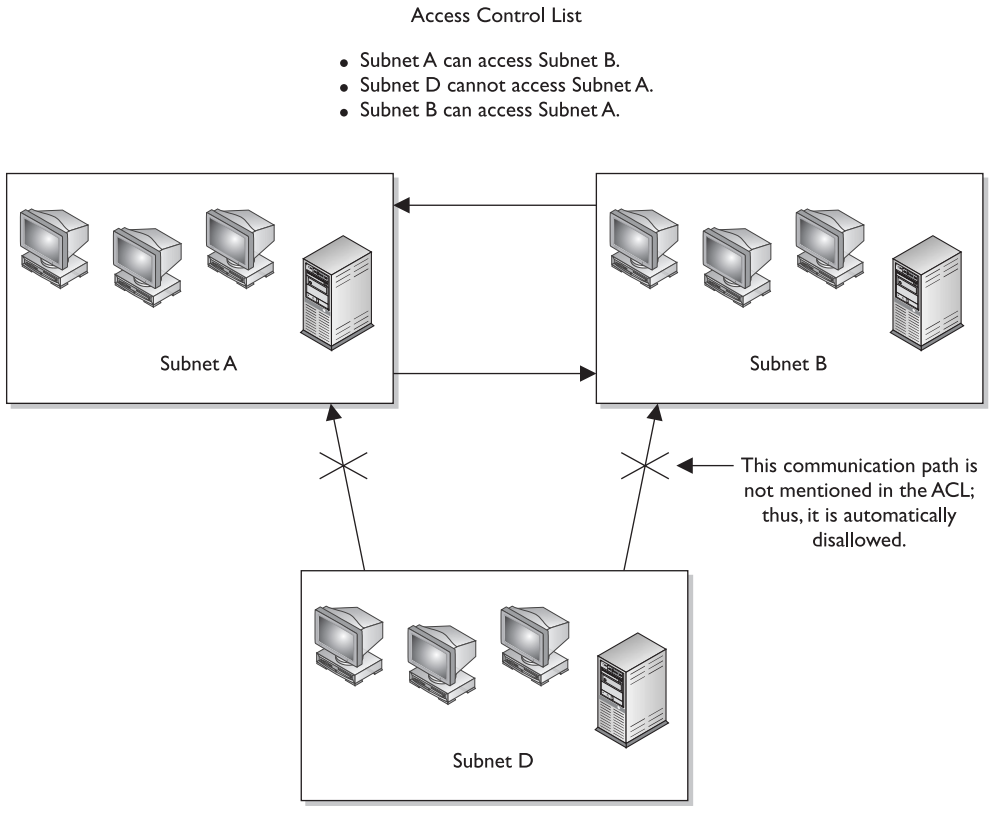


Figure 4-4 What is not explicitly allowed should be implicitly denied.

control of all the files he needs to copy, but that would not be practicing the least-privilege principle. The administrator should restrict Dan's rights and permissions to only allow him to read and print the necessary files, and no more. Besides, if Dan accidentally deletes all the files on the whole file server, whom do you think management will hold ultimately responsible? Yep, the administrator.

It is important to understand that it is management's job to determine the security requirements of individuals and how access is authorized. The security administrator configures the security mechanisms to fulfill these requirements, but it is not her job to determine security requirements of users. Those should be left to the owners. If there is a security breach, management will ultimately be held responsible, so it should make these decisions in the first place.

Single Sign-On

I only want to have to remember one username and one password for everything in the world!

Many times employees need to access many different computers, servers, databases, and other resources in the course of a day to complete their tasks. This often requires the employees to remember multiple user IDs and passwords for these different

computers. In a utopia, a user would need to enter only one user ID and one password to be able to access all resources in all the networks this user is working in. In the real world, this is hard to accomplish for all system types.

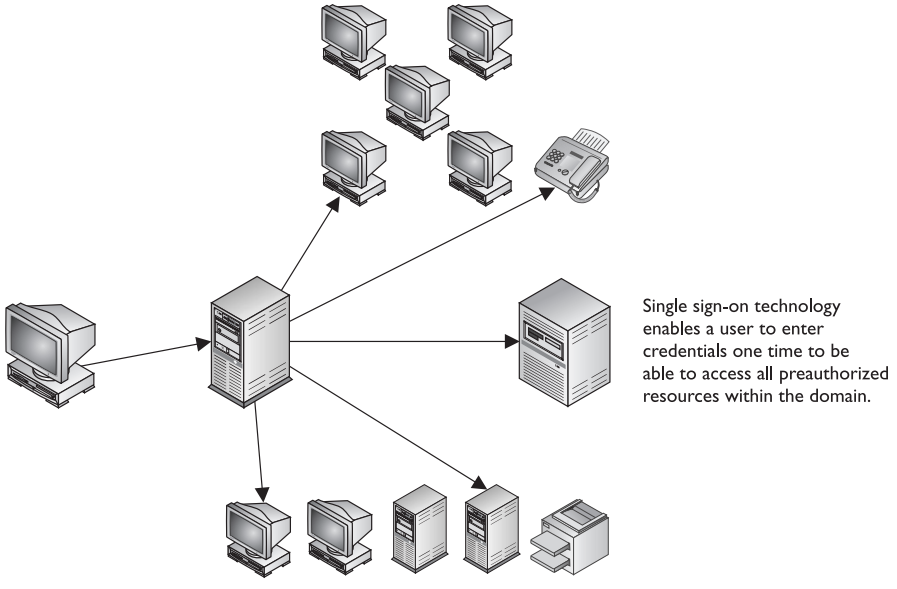
Because of the proliferation of client/server technologies, networks have migrated from centrally controlled networks to heterogeneous, distributed environments. The propagation of open systems and the increased diversity of applications, platforms, and operating systems have caused the end user to have to remember several user IDs and passwords just to be able to access and use the different resources within his own network. Although the different IDs and passwords are supposed to provide a greater level of security, they often end up compromising security (because users write them down) and causing more effort and overhead for the staff that manages and maintains the network.

As any network staff member or administrator can attest to, too much time is devoted to resetting passwords for users who have forgotten them. More than one employee's productivity is affected when forgotten passwords have to be reassigned. The network staff member who has to reset the password could be working on other tasks, and the user who forgot the password cannot complete his task until the network staff member is finished resetting the password. Many help-desk employees report that a majority of their time is spent on users forgetting their passwords. System administrators have to manage multiple user accounts on different platforms, which all need to be coordinated in a manner that maintains the integrity of the security policy. At times the complexity can be overwhelming, which results in poor access control management and the generation of many security vulnerabilities because the user IDs and passwords are usually written down and posted on the side of the monitor or underneath the user's keyboard. This fact defeats the purpose of using passwords in the first place. A lot of time is spent on multiple passwords, and in the end, they do not provide us with more security.

The increased cost of managing a diverse environment, security concerns, and user habits, coupled with the users' overwhelming desire to remember one set of credentials, has brought about the idea of ***single sign-on (SSO)*** capabilities. These capabilities would allow a user to enter credentials one time and be able to access all resources in primary and secondary network domains. This reduces the amount of time users spend authenticating to resources and it enables the administrator to streamline user accounts and better control access rights. It improves security by reducing the probability that users will write down passwords and reduces the administrator's time spent on adding and removing user accounts and modifying access permissions. If an administrator needs to disable or suspend a specific account, she can do it uniformly instead of having to alter configurations on each and every platform.

So that is our utopia: log on once and you are good to go. What bursts this bubble? Mainly interoperability issues. For SSO to actually work, every platform, application, and resource needs to accept the same type of credentials, in the same format, and interpret their meanings the same. When Steve logs onto his Windows NT 4.0 workstation and gets authenticated by a mixed-mode Windows 2000 domain controller, it needs to

authenticate him to the resources he needs to access on the Apple computer, the Unix server running VisionFS, the mainframe host server, the MICR print server, and the Windows XP computer in the secondary domain that has the plotter connected to it. Nice idea, until reality hits.



There is also a security issue to consider in an SSO environment. Once an individual is in, he is in. If an attacker was able to uncover one credential set, he would have access to every resource within the environment that the compromised account has access to.

There are different types of SSO technologies. Each has its own advantages and disadvantages, shortcomings, and quality features. It is rare to see a real SSO environment; rather, you will see a cluster of computers and resources that accept the same credentials, whereas other resources still require more work on the administrator or user side to access the systems. The SSO technologies that may be addressed in the CISSP exam are described in the next sections.

Kerberos

Sam, there is a three-headed dog in front of the server!

Kerberos is the name of a three-headed dog that guards the entrance to the underworld in Greek mythology. This is a great name for a security technology that provides authentication functionality, with the purpose of protecting a company's assets. Kerberos is an authentication protocol and was designed in the mid-1980s as part of MIT's Project Athena. It works in a client/server model and is based on symmetric key cryptography. The protocol has been used for years in Unix systems and is currently the default authentication method for Windows 2000 and 2003 operating systems.

Commercial products supporting Kerberos are becoming more frequent, so this one might be a keeper.

Kerberos is an example of a single sign-on system for distributed environments, and is a de facto standard for heterogeneous networks. Kerberos incorporates a wide range of security technologies, which gives companies much more flexibility and scalability when they need to provide an encompassing security architecture. However, this open architecture also invites interoperability issues. When vendors have a lot of freedom to customize a protocol, it usually means that no two vendors will customize it in the same fashion. This creates interoperability and incompatibility issues.

Kerberos uses symmetric key cryptography and provides end-to-end security. Although it allows the use of passwords for authentication, it was designed specifically to eliminate the need to transmit passwords over the network. Most Kerberos implementations work with shared secret keys.

Main Components in Kerberos The *Key Distribution Center (KDC)* is the most important component within a Kerberos environment. The KDC holds all users' and services' secret keys. It provides an authentication service, as well as key distribution functionality. The clients and services trust the integrity of the KDC, and this trust is the foundation of Kerberos security.

The KDC provides security services to ***principals***, which can be users, applications, or network services. The KDC must have an account for, and share a secret key with, each principal. For users, a password is transformed into a secret key value. The secret key is used to send sensitive data back and forth between the principal and the KDC, and is used for user authentication purposes.

A ***ticket*** is generated by the KDC and given to a principal when that principal, let's say a user, needs to authenticate to another principal, let's say a print server. The ticket enables one principal to authenticate to another principal. If Emily needs to use the print server, she needs to prove to the print server she is who she claims to be and that she is authorized to use the printing service. So Emily requests a ticket from the KDC. The KDC gives Emily the ticket, and in turn, Emily passes this ticket on to the print server. If the print server approves of this ticket, Emily is allowed to use the print service.

A KDC provides security services for a set of principals. This set is called a ***realm*** in Kerberos. The KDC is the trusted authentication server for all users, applications, and services within a realm. One KDC can be responsible for one realm or several realms. Realms are used to allow an administrator to logically group resources and users.

So far, we know that principals (users and services) require the KDC's services to authenticate to each other; that the KDC has a database filled with information about each and every principal within its realm; that the KDC holds and delivers cryptographic keys and tickets; and that tickets are used for principals to authenticate to each other. So how does this process work?

Kerberos Authentication Process The user and the KDC share a secret key; the service and the KDC share a different secret key. The user and the requested service do not share a symmetric key in the beginning. The user trusts the KDC because they share a secret key. They can encrypt and decrypt data they pass between each other, and thus have a protected communication path. Once the user authenticates to the service,

they too will share a symmetric key (session key) that will enable them to encrypt and decrypt the information they need to pass to each other. This is how Kerberos provides data transmission protection.

Here are the exact steps:

1. Emily comes in to work and enters her username and password into her workstation at 8 A.M.
2. The Kerberos software on Emily's computer sends the username to the authentication service (AS) on the KDC, which in turn sends Emily a ticket granting ticket (TGT) that is encrypted with Emily's password (secret key).
3. If Emily has entered her correct password, then this TGT is decrypted and Emily gains access to her local workstation desktop.
4. When Emily needs to send a print job to the print server, her system sends the TGT to the ticket granting service (TGS), which runs on the KDC. (This allows Emily to prove that she has been authenticated and allows her to request access to the print server.)
5. The TGS creates and sends a second ticket to Emily, which she will use to authenticate to the print server. This second ticket contains two instances of the same session key, one encrypted with Emily's secret key and the other encrypted with the print server's secret key. The second ticket also contains an ***authenticator***, which contains identification information on Emily, her system's IP address, and a timestamp.
6. Emily's system receives the second ticket, decrypts and extracts the session key, adds a second authenticator set of identification information to the ticket, and sends the ticket onto the print server.
7. The print server receives the ticket, decrypts and extracts the session key, and decrypts and extracts the two authenticators in the ticket. If the printer server can decrypt and extract the session key, it knows that the KDC created the ticket, because only the KDC has the secret key that was used to encrypt the session key. If the authenticator information that the KDC and the user put into the ticket matches, then the print server knows that it received the ticket from the correct principal.
8. Once this is completed, it means that Emily has been properly authenticated to the print server and the server prints her document.

This is an extremely simplistic overview of what is going on in any Kerberos exchange, but it gives you an idea of the dance that is taking place behind the scenes whenever you interact with any network service in an environment that uses Kerberos. Figure 4-5 provides a simplistic view of this process.

The authentication service is the part of the KDC that authenticates a principal, and the TGS is the part of the KDC that makes the tickets and hands them out to the principals. TGTs are used so that the user does not have to enter his password each time he needs to communicate with another principal. After the user enters his password, it is

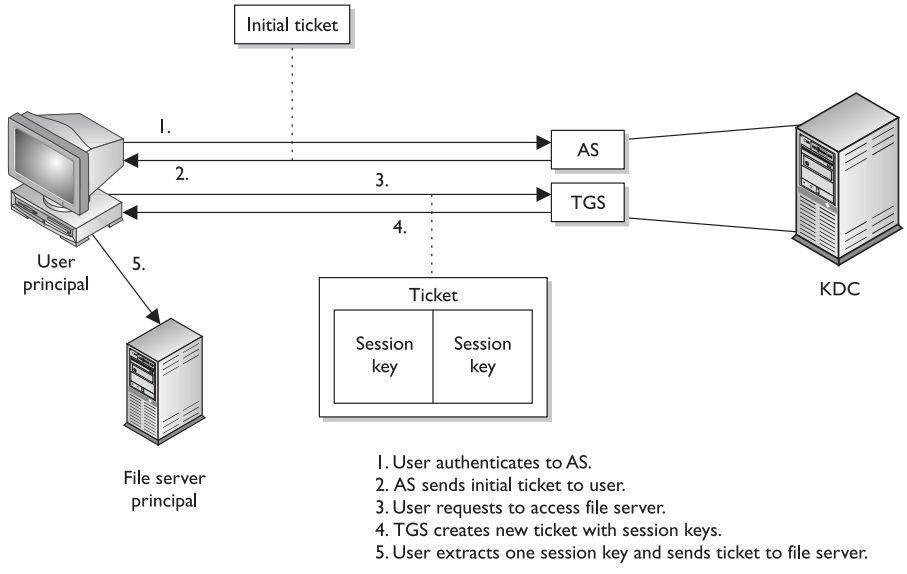


Figure 4-5 The user needs to receive a ticket from the KDC before being able to use the requested resource.

temporarily stored on his system, and any time the user needs to communicate with another principal, he just reuses the TGT.

Be sure you understand that a session key is different from a secret key. A secret key is shared between the KDC and a principal and is static in nature. A session key is shared between two principals and is generated when needed and destroyed after the session is completed.

If a Kerberos implementation is configured to use an *authenticator,* the user sends to the printer server her identification information and a timestamp encrypted with the session key they share. The printer server decrypts this information and compares it with the identification data the KDC sent to it about this requesting user. If the data is the same, the printer server allows the user to send print jobs. The timestamp is used to help fight against replay attacks. The printer server compares the sent timestamp with its own internal time, which helps to determine if the ticket has been sniffed and copied by an attacker and submitted at a later time in hope of impersonating the legitimate user and gaining unauthorized access.

The primary reason to use Kerberos is that the principals do not trust each other enough to communicate directly. In our example, the print server will not print anyone's print job without that entity authenticating itself. So none of the principals trust each other directly; they only trust the KDC. The KDC creates tickets to vouch for the individual principals when they need to communicate. Suppose that I need to communicate directly with you, but you do not trust me enough to listen and accept what I am saying. If I first give you a ticket from something you do trust (KDC), this basically says, "Look, the KDC says I am a trustworthy person. The KDC asked me to give

this ticket to you to prove it." Once that happens, *then* you will communicate directly with me.

The same type of trust model is used in PKI environments. (More information on PKI is presented in Chapter 8.) In a PKI environment, users do not trust each other directly, but they all trust the certificate authority (CA). The CA vouches for the individuals' identities by using digital certificates, the same as the KDC vouches for the individuals' identities by using tickets.

So why are we talking about Kerberos? Because it is one example of a single sign-on technology. The user enters a user ID and password one time and one time only. The tickets have time limits on them that administrators can configure. Many times, the lifetime of a TGT is eight to ten hours, so when the user comes in the next day, he will have to present his credentials again.

Weaknesses of Kerberos Kerberos is an authentication protocol that can provide confidentiality of the sensitive data being passed back and forth over a network. The following are some of the potential weaknesses of Kerberos:

- The KDC can be a single point of failure. If the KDC goes down, no one can access needed resources. Redundancy is necessary for the KDC.
- The KDC must be able to handle the number of requests it receives in a timely manner. It must be scalable.
- Secret keys are temporarily stored on the users' workstations, which means it is possible for an intruder to obtain these cryptographic keys.
- Session keys are decrypted and reside on the users' workstations, either in a cache or in a key table. Again, an intruder can capture these keys.
- Kerberos is vulnerable to password guessing. The KDC does not know if a dictionary attack is taking place.
- Network traffic is not protected by Kerberos if encryption is not enabled.

Kerberos needs to be transparent (work in the background without the user needing to understand it), scalable (work in large, heterogeneous environments), reliable (distributed server architecture to ensure that there is no single point of failure), and secure (provide authentication and confidentiality).

Kerberos and Password-Guessing Attacks

Just because an environment uses Kerberos does not mean that the systems are vulnerable to password-guessing attacks. The operating system itself will (should) provide the protection of tracking failed login attempts. The Kerberos protocol does not have this type of functionality, so another component must be in place to counter these types of attacks. No need to start ripping Kerberos out of your network environment after reading this section; your operating system provides the protection mechanism for this type of attack.

References

- **Kerberos FAQ** www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html
- **MIT Kerberos Papers and Documentation web page** www.mit.edu/afs/athena.mit.edu/astaff/project/kerberos/www/papers.html

SESAME

The *Secure European System for Applications in a Multi-vendor Environment (SESAME)* project is a single sign-on technology that was developed to extend Kerberos functionality and improve upon its weaknesses. SESAME uses symmetric and asymmetric cryptographic techniques to protect exchanges of data and to authenticate subjects to network resources.

NOTE Kerberos is a strictly symmetric key–based technology, whereas SESAME is based on both asymmetric and symmetric key cryptography.

Kerberos uses tickets to authenticate subjects to objects, whereas SESAME uses Privileged Attribute Certificates (PACs), which contain the subject's identity, access capabilities for the object, access time period, and lifetime of the PAC. The PAC is digitally signed so that the object can validate that it came from the trusted authentication server, which is referred to as the Privileged Attribute Server (PAS). The PAS holds a similar role to that of the KDC within Kerberos. After a user successfully authenticates to the authentication service (AS), he is presented with a token to give to the PAS. The PAS then creates a PAC for the user to present to the resource he is trying to access. Figure 4-6 shows a basic overview of the SESAME process.

NOTE Kerberos and SESAME can be accessed through the Generic Security Services Application Programming Interface (GSS-API), which is a generic API for client-to-server authentication. Using standard APIs enables vendors to communicate with and use each other's functionality and security. Kerberos Version 5 and SESAME implementations allow any application to use their authentication functionality as long as the application knows how to communicate via GSS-API.

References

- **SESAME in a Nutshell** www.cosic.esat.kuleuven.ac.be/sesame/html/sesame_what.html
- **SESAME links** www.cosic.esat.kuleuven.ac.be/sesame/html/sesame_links.html

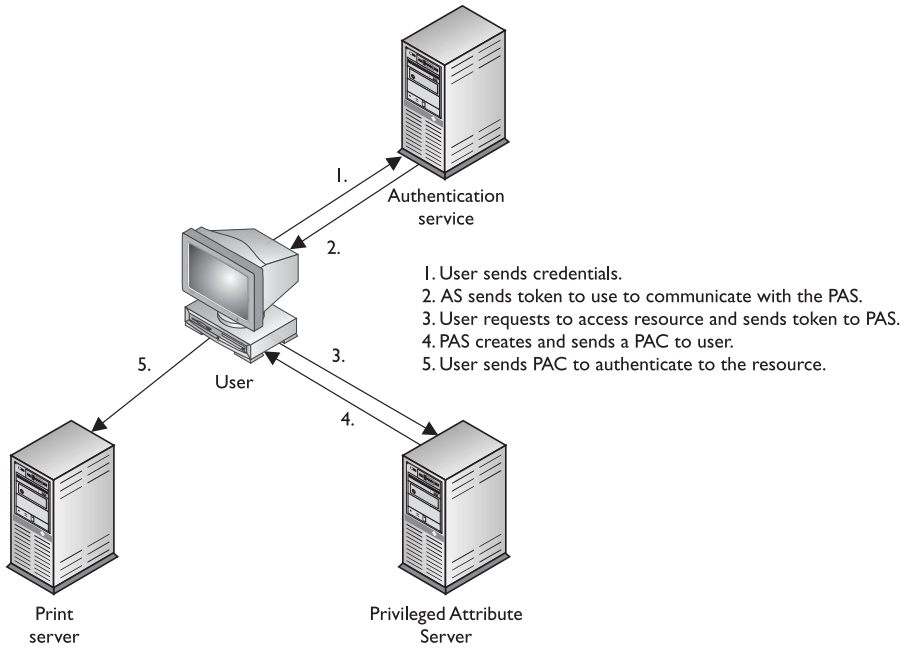


Figure 4-6 SESAME is very similar to Kerberos.

Security Domains

I am highly trusted and have access to many resources. Response: So what.

The term "domain" has been around a lot longer than Microsoft, but when people hear this term, they often think of a set of computers and devices on a network segment being controlled by a server that runs Microsoft software, referred to as a domain controller. A domain is really just a set of resources that is available to a subject. Remember that a subject can be a user, process, or application. Within an operating system, a process has a domain, which is the set of system resources that is available to the process to carry out its tasks. These resources can be memory segments, hard drive space, operating system services, and other processes. In a physical network environment, a domain is a set of physical and logical resources that is available, which can include routers, file servers, FTP service, web servers, and so forth.

The term *security domain* just builds upon the definition of domain by adding the fact that resources within this logical structure (domain) are working under the same security policy and managed by the same group. So, a network administrator may put all of the accounting personnel, computers, and network resources in Domain 1 and all of the management personnel, computers, and network resources in Domain 2. These items fall into these individual containers because they not only carry out similar types of business functions, but also, and more importantly, have the same type of trust level. It is this common trust level that allows entities to be managed by one single security policy.

The different domains are separated by logical boundaries, such as firewalls with ACLs, directory services making access decisions, and objects that have their own ACLs indicating which individuals and groups can carry out operations on them. All of these security mechanisms are examples of components that enforce the security policy for each domain.

Domains can be architected in a hierarchical manner that dictates the relationship between the different domains and the ways in which subjects within the different domains can communicate. Subjects can access resources in domains of equal or lower trust levels. Figure 4-7 shows an example of hierarchical network domains. Their communication channels are controlled by security agents (firewalls, router ACLs, directory services), and the individual domains are isolated by using specific subnet mask addresses.

Remember that a domain does not necessarily pertain only to network devices and segmentations, but can also apply to users and processes. Figure 4-8 illustrates how users

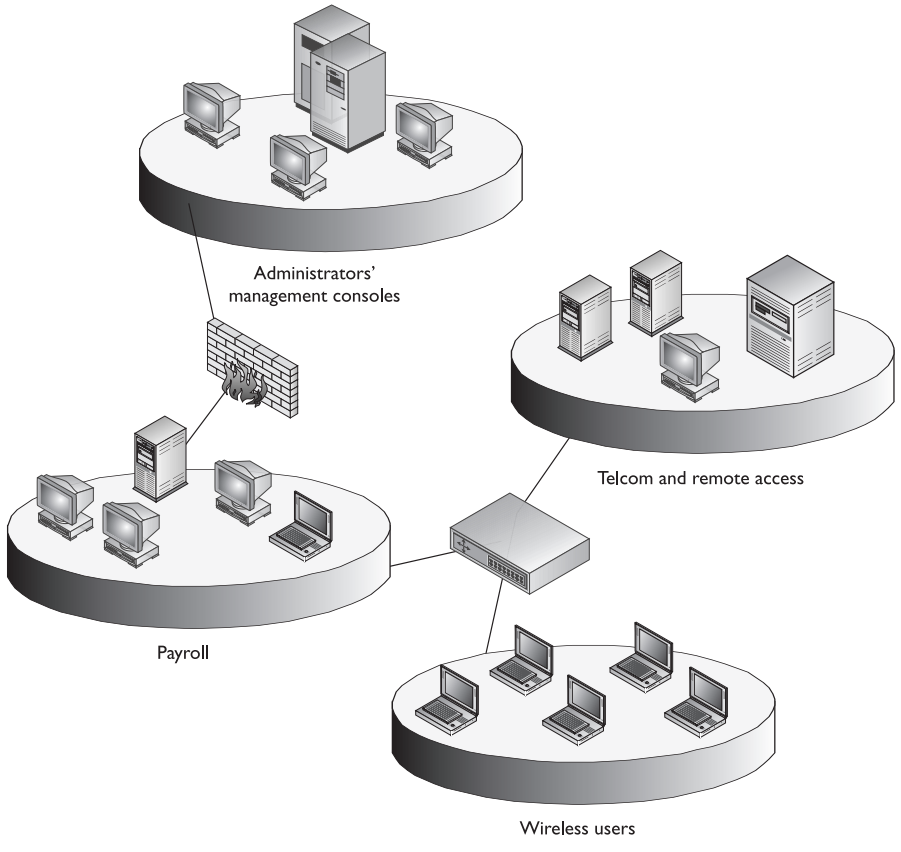


Figure 4-7 Network domains that are used to separate different network segments

and processes can have more granular domains assigned to them individually based on their trust level. Group 1 has a high trust level and can access both a domain of its own trust level (Domain 1) and a domain of a lower trust level (Domain 2). User 1, who has a lower trust level, can access only the domain at his trust level and nothing higher. The system enforces these domains with access privileges and rights provided by the file system and operating system security kernel.

So why are domains in the "Single Sign-On" section? Because several different types of technologies are available today that are used to define and enforce these domains and security policies that are mapped to them: domain controllers in a Windows environment, enterprise resource management (ERM) products, Microsoft Passport, and the various products that provide SSO functionality. The goal of each of them is to allow a user (subject) to sign in one time and be able to access the different domains that are available to them without having to reenter any other credentials.

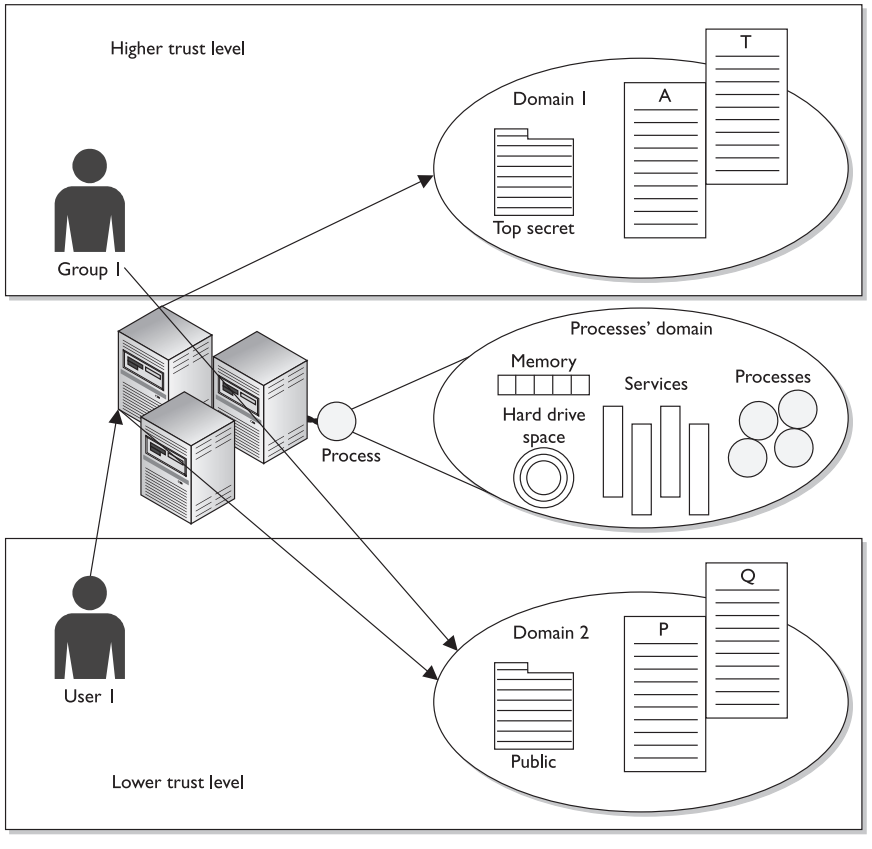


Figure 4-8 Subjects can access specific domains based on their trust levels.

References

- *Underlining Technical Models for Information Technology Security, Recommendations of the National Institute of Standards and Technology,* **by Gary Stoneburner, NIST Special Publication 800-33 (Dec. 2001)** http://csrc.nist.gov/publications/nistpubs/800-33/sp800-33.pdf
- **"New Thinking About Information Technology Security," by Marshall Abrams, PhD and Michael Joyce (first published in *Computers & Security*, Vol. 14, No. 1, pp. 57–68)** www.acsac.org/secshelf/papers/new_thinking.pdf

Directory Services

A network service is a mechanism that identifies resources (printers, file servers, domain controllers, and peripheral devices) on a network and provides a way to make them available to users and programs. A network directory service contains information about these different resources and provides a naming scheme. It also provides a hierarchical database that outlines the resources' characteristics, such as name, logical and physical location, subjects that can access them, and the operations that can be carried out on them. These directory services are held within a domain controller in a Windows environment.

Network directory services provide users access to network resources transparently, meaning that users don't need to know the exact location of the resources or the steps required to access them. The network directory services handle these issues for the user in the background. Some examples of directory services are Lightweight Directory Access Protocol (LDAP), Novell NetWare Directory Service (NDS), and Microsoft Active Directory.

Thin Clients

Hey, where's my operating system? Response: You don't deserve one.

Diskless computers, sometimes called dumb terminals or thin clients, cannot store much information because of their lack of on-board storage space and necessary resources. This type of client/server technology forces users to log onto a central server just to be able to use the computer and access network resources. When the user starts the computer, it runs a short list of instructions and then points itself to a server that will actually download the operating system, or interactive operating software, to the terminal. This enforces a strict type of access control, because the computer cannot do anything on its own until it authenticates to a centralized server, and then the server gives the computer its operating system, profile, and functionality. Thin-client technology provides another type of SSO access for users, because users authenticate only to the central server or mainframe, which then provides them access to all authorized and necessary resources.

In addition to providing an SSO solution, a thin-client technology offers several other advantages. A company can save money by purchasing thin clients instead of powerful and expensive PCs. The central server handles all application execution, pro-

Examples of Single Sign-On Technologies

- **Kerberos** Authentication protocol that uses a KDC and tickets, and is based on symmetric key cryptography
- **SESAME** Authentication protocol that uses a PAS and PACs, and is based on symmetric and asymmetric cryptography
- **Security domains** Resources working under the same security policy and managed by the same group
- **Thin clients** Terminals that rely upon a central server for access control, processing, and storage

cessing, and data storage. The thin client displays the graphical representation and sends mouse clicks and keystroke inputs to the central server. Having all of the software in one location, instead of distributed throughout the environment, allows for easier administration, centralized access control, easier updates, and standardized configurations. It is also easier to control malware infestations and the theft of confidential data because the thin clients often do not have CD-ROM, DVD, or floppy drives.

Access Control Models

An ***access control model*** is a framework that dictates how subjects access objects. It uses access control technologies and security mechanisms to enforce the rules and objectives of the model. There are three main types of access control models: discretionary, mandatory, and nondiscretionary (also called role-based). Each model type uses different methods to control how subjects access objects, and each has its own merits and limitations. The business and security goals of an organization will help prescribe what access control model it should use, along with the culture of the company and the habits of conducting business. Some companies use one model exclusively, whereas others combine them to be able to provide the necessary level of protection.

These models are built into the core or the kernel of the different operating systems and possibly their supporting applications. Every operating system has a security kernel that enforces a reference monitor concept, which differs depending upon the type of access control model that has been embedded into the system. For every access attempt, before a subject can communicate with an object, the security kernel reviews the rules of the access control model to determine whether the request is allowed.

The following sections explain these different models, their supporting technologies, and where they should be implemented.

Discretionary Access Control

It is up to me to allow you to access my files. Response: Mother, may I?

If a user creates a file, he is the owner of that file. An identifier for this user is placed in the file header. Ownership might also be granted to a specific individual. For example, a manager for a certain department might be made the owner of the files and resources within her domain. A system that uses ***discretionary access control (DAC)*** enables the owner of the resource to specify which subjects can access specific resources. This model is called discretionary because the control of access is based on the discretion of the owner.

In a DAC model, access is restricted based on the authorization granted to the users. This means that users are allowed to specify what type of access can occur to the objects they own. If an organization is using a DAC model, the network administrator can allow resource owners to control who has access to their files. The most common implementation of DAC is through ACLs, which are dictated and set by the owners and enforced by the operating system. This does not lend itself to a centrally controlled environment and can make a user's ability to access information dynamic versus the more static role of mandatory access control (MAC).

Most of the operating systems that you may be used to dealing with are based on DAC models, such as all Windows, Linux, and Macintosh systems and most flavors of Unix. When you look at the properties of a file or directory and you see the choices that allow you to control which users can have access to this resource and to what degree, you are witnessing an instance of ACLs enforcing a DAC model.

DACs can be applied to both the directory tree structure and the files it contains. The PC world has access permissions of No Access, Read (r), Write (w), Execute (x), Delete (d), Change (c), and Full Control. The Read attribute allows you to read the file but not make changes. The Change attribute allows you to read, write, execute, and delete the file but does not allow you to change the ACLs or the owner of the files. Obviously, the attribute of Full Control allows you to make any changes to the file and its permissions and ownership.

It is through the discretionary model that Sam can share his D: drive with David so that David can copy all of Sam's MP3s. Sam can also block access to his D: drive from his manager so that his manager does not know that Sam is wasting valuable time and resources by downloading MP3s and sharing them with friends.

References

- ***Security in Open Systems,*** **Node 25, "Discretionary Access Control," by Robert Bagwell, et al., NIST Special Publication 800-7 (July 1994)** http://csrc.nist.gov/publications/nistpubs/800-7/node25.html
- ***Wikipedia*** **definition of access control** http://en.wikipedia.org/wiki/Access_control

Identity-Based Access Control

DAC systems grant or deny access based on the identity of the subject. The identity can be a user identity or group membership. So, for example, a data owner can choose to allow Bob (user identity) and the Accounting group (group membership identity) to access his file.

Mandatory Access Control

In a ***mandatory access control (MAC)*** model, users and data owners do not have as much freedom to determine who can access files. The operating system makes the final decision and can override the users' wishes. This model is much more structured and strict and is based on a security label system. Users are given a security clearance (secret, top secret, confidential, and so on), and data is classified in the same way. The clearance and classification data is stored in the security labels, which are bound to the specific subjects and objects. When the system makes a decision about fulfilling a request to access an object, it is based on the clearance of the subject, the classification of the object, and the security policy of the system. The rules for how subjects access objects are made by the security officer, configured by the administrator, enforced by the operating system, and supported by security technologies.

Security labels are attached to all objects; thus, every file, directory, and device has its own security label with its classification information. A user may have a security clearance of secret, and the data that he requests may have a security label with the classification of top secret. In this case, the user will be denied because his clearance is not equivalent or does not dominate (equal or higher than) the classification of the object.

NOTE The terms "security labels" and "sensitivity labels" can be used interchangeably.

Each subject and object must have an associated label with attributes at all times, because this is part of the operating system's access-decision criteria. Each subject and object does not require a physically unique label, but can be logically associated. For example, all subjects and objects on Server 1 can share the same label of secret clearance and classification.

This type of model is used in environments where information classification and confidentiality is of utmost importance, such as a military institution. Special types of Unix systems are developed based on the MAC model. A company cannot simply choose to turn on either DAC or MAC. It has to purchase an operating system that has been specifically designed to enforce MAC rules. DAC systems do not understand security labels, classifications, or clearances, and thus cannot be used in institutions that

require this type of structure for access control. The most recently released MAC system is SE Linux, developed by the NSA and Secure Computing. Trusted Solaris is a product based on the MAC model that most people are familiar with (relative to other MAC products).

References

- *Security in Open Systems*, **Node 36, "Determining MAC Access," by Robert Bagwell, et al., NIST Special Publication 800-7 (July 1994)** http://csrc.nist.gov/publications/nistpubs/800-7/node36.html
- ***Integrating Flexible Support for Security Policies into the Linux Operating System*, Node 2, "Security Architecture," by Peter Loscocco and Stephen Smalley, National Security Agency research paper (2001)** www.nsa.gov/selinux/papers/freenix01/node2.html
- **Access Control** www.list.gmu.edu/infs762/infs762su04ng/l1-access-control.ppt

Sensitivity Labels

I am very sensitive. Can I have a label? Response: Nope.

When the MAC model is being used, every subject and object must have a sensitivity label, also called a security label. It contains a classification and different categories. The classification indicates the sensitivity level, and the categories enforce need-to-know rules. Figure 4-9 illustrates a sensitivity label.

The classifications follow a hierarchical structure, one level being more trusted than another. However, the categories do not follow a hierarchical scheme, because they represent compartments of information within a system. The categories can correspond to departments (UN, Information Warfare, Treasury), projects (CRM, AirportSecurity, 2003Budget), or management levels. In a military environment, the classifications could be top secret, secret, confidential, and unclassified. Each classification is more trusted than the one below it. A commercial organization might use confidential, proprietary, corporate, and sensitive. The definition of the classification is up to the organization and should make sense for the environment in which it is used.

The categories portion of the label enforces need-to-know rules. Just because someone has a top-secret clearance does not mean that she now has access to all top-secret information. She must also have a need to know. As shown in Figure 4-9, if Cheryl has a top-secret clearance but does not have a need to know that is sufficient to access any of the listed categories (Dallas, Max, Cricket), she cannot look at this object.

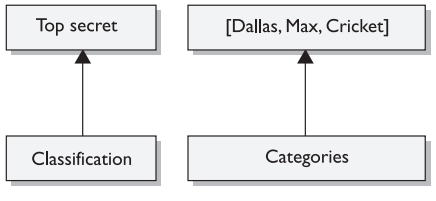


Figure 4-9 A sensitivity label is made up of a classification and categories.

NOTE In MAC implementations, the system makes access decisions by comparing the subject's clearance and need-to-know level to that of the security label. In DAC, the system compares the subject's identity to the ACL on the resource.

Role-Based Access Control

I am in charge of chalk, thus I need full control of all servers! Response: Good try.

A ***role-based access control (RBAC)*** model, also called ***nondiscretionary access control***, uses a centrally administrated set of controls to determine how subjects and objects interact. This type of model allows access to resources to be based on the role the user holds within the company. It is referred to as nondiscretionary because assigning a user to a role is unavoidably imposed. This means that if you are assigned only to the Contractor role in a company, there is nothing you can do about it. You don't have the discretion to determine what role you will be assigned.

The more traditional access control administration is based on just the DAC model, where access control is specified at the object level with ACLs. This approach is more complex because the administrator has to translate organizational authorization policy into permission when configuring ACLs. As the number of objects and users grows within an environment, users are bound to be granted unnecessary access to some objects, thus violating the least-privilege rule and increasing the risk to the company. The RBAC approach simplifies access control administration by allowing permissions to be managed in terms of user job roles.

In an RBAC model, a role is defined in terms of the operations and tasks that the role will need to carry out, whereas a DAC model outlines which subjects can access what objects.

Let's say we need a research and development analyst role. We develop this role not only to allow an individual to have access to all product and testing data, but also, and more importantly, to outline the tasks and operations that the individual can carry out on this data. When the analyst role makes a request to access the new testing results on the file server, in the background the operating system reviews the role's access levels before allowing this operation to take place.

NOTE Introducing roles also introduces the difference between rights being assigned explicitly and implicitly. If rights and permissions are assigned explicitly, it indicates that they are assigned directly to a specific individual. If they are assigned implicitly, it indicates that they are assigned to a role or group and the user inherits those attributes.

An RBAC model is the best system for a company that has high employee turnover. If John, who is mapped to the contractor role, leaves the company, then Chrissy, his replacement, can be easily mapped to this role. That way, the administrator does not need to continually change the ACLs on the individual objects. He only needs to create a role (contractor), assign permissions to this role, and map the new user to this role.

RBAC, MAC, DAC

A lot of confusion exists regarding whether RBAC is a type of DAC model or type of MAC model. Different sources claim different things, but in fact it is a model in its own right. In the 1960s and 1970s, the U.S. military and NSA did a lot of research on the MAC model. DAC, which also sprang to life in the '60s and '70s, has its roots in the academic and commercial research laboratories. The RBAC model, which started to gain popularity in the 1990s, can be used in combination with MAC and DAC systems. For the most up-to-date information on the RBAC model, go to **http://csrc.nist.gov/rbac**, which has documents that describe an RBAC standard and independent model, with the goal of clearing up this continual confusion.

References

- **Role Based Access Control** http://hissa.nist.gov/project/rbac.html
- **Role Based Access Control Case Studies** http://csrc.nist.gov/rbac
- **"Proposed Standard for Role Based Access Control," by David Ferraiolo, et al. (first published in *ACM Transactions on Information and System Security*, Vol. 4, No. 3, Aug. 2001, pp. 224–274)** http://csrc.nist.gov/rbac/rbacSTD-ACM.pdf
- **"Role-Based Access Control for the Web," by John Barkley, et al., CALS Expo International & 21st Century Commerce 1998: Global Business Solutions for the New Millennium (1998)** http://hissa.ncsl.nist.gov/rbac/cals-paper.html
- **"A Role-Based Access Control Model and Reference Implementation Within a Corporate Intranet," by John Barkley, D. Richard Kuhn, and David Ferraiolo (first published in *ACM Transactions on Information and System Security*, Vol. 2, No. 1, Feb. 1999, pp. 34–64)** www.ecs.syr.edu/faculty/chin/cse774/readings/rbac/p34-ferraiolo.pdf

Access Control Models

The main characteristics of the three different access control models are important to understand.

- **DAC** Data owners decide who has access to resources, and ACLs are used to enforce the security policy
- **MAC** Operating systems enforce the system's security policy through the use of security labels
- **RBAC** Access decisions are based on each subject's role and/or functional position

Access Control Techniques and Technologies

Once an organization determines what type of access control model it is going to use, it needs to identify and refine its technologies and techniques to support the model. The following sections describe the different access controls and technologies available to support different access control models.

Rule-Based Access Control

Everyone will adhere to my rules. Response: Who are you again?

Rule-based access control uses specific rules that indicate what can and cannot happen between a subject and an object. It is based on the simple concept of "if X then Y" programming rules, which can be used to provided finer-grained access control to resources. Before a subject can access an object in a certain circumstance, it must meet a set of predefined rules. This can be simple and straightforward, as in "if the user's ID matches the unique user ID value that is in the provided digital certificate, then the user can gain access." Or there could be a set of complex rules that must be met before a subject can access an object. For example, "if the user is accessing the system between Monday and Friday and between 8 A.M. and 5 P.M., and if the user's security clearance equals or dominates the object's classification, and if the user has the necessary need to know, then the user can access the object."

Rule-based access control is not necessarily identity-based. The DAC model is identity-based. For example, an identity-based control would stipulate that Tom Jones can read File1 and modify File2. So when Tom attempts to access one of these files, the operating system will check his identity and compare it to the values within an ACL to see if Tom can carry out the operations he is attempting. In contrast, here is a rule-based example: a company may have a policy that dictates that e-mail attachments can only be 5MB or smaller. This rule affects all users. If rule-based was identity-based, it would mean that Sue can accept attachments of 10MB and smaller, Bob can accept attachments 2MB and smaller, and Don can only accept attachments 1MB and smaller. This would be a mess and too confusing. Rule-based access controls simplify this by setting a rule that will affect all users across the board—no matter what their identity is.

Rule-based access allows a developer to define specific and detailed situations in which a subject can or cannot access an object, and what that subject can do once access is granted. Traditionally, rule-based access control has been used in MAC systems as an enforcement mechanism of the complex rules of access that MAC systems provide; today, rule-based access is used in other types of systems and applications, as well.

Many routers and firewalls use rules to determine which types of packets are allowed into a network and which are rejected. Rule-based access control is a type of compulsory control, because the administrator sets the rules and the users cannot modify these controls.

References

- **"Creating a Policy-Aware Web: Discretionary, Rule-based Access for the World Wide Web," Tim Berners-Lee, et al.** www.w3.org/2004/09/Policy-Aware-Web-acl.pdf
- **"Roles or Rules: The Access Control Debate," by John Desmond, *eSecurityPlanet* Security Advisors article (July 29, 2003)** www.esecurityplanet.com/views/article.php/2241671

Constrained User Interfaces

Constrained user interfaces restrict users' access abilities by not allowing them to request certain functions or information, or to have access to specific system resources. There are three major types of restricted interfaces: menus and shells, database views, and physically constrained interfaces.

When menu and shell restrictions are used, the options that users are given are the commands that they can execute. For example, if an administrator wants users to be able to execute only one program, that program would be the only choice available on the menu. This limits the users' functionality. A *shell* is a type of virtual environment within a system; it is the user's interface to the operating system and works as a command interpreter. If restricted shells were used, the shell would contain only the commands the administrator wants the users to be able to execute.

Many times, a database administrator will configure a database so that users cannot see fields that require a level of confidentiality. ***Database views*** are mechanisms used to restrict user access to data that is contained in databases. If the database administrator wants managers to be able to view their employees' work records but not their salary information, then the salary fields would not be available to these types of users. Similarly, when payroll employees look at the same database, they will be able to view the salary information but not the work history information. This example is illustrated in Figure 4-10.

Physically constraining a user interface can be implemented by providing only certain keys on a keypad or certain touch buttons on a screen. You see this when you

Figure 4-10 Different database views of the same tables

Harris, D	$45,000	8am-5pm
Torkelson, T	$60,000	6pm-2am
Kowtko, J	$45,000	8am-5pm
Swenson, J	$65,000	6pm-2am

Payroll database view

Harris, D	Work history	8am-5pm
Torkelson, T	Work history	6pm-2am
Kowtko, J	Work history	8am-5pm
Swenson, J	Work history	6pm-2am

Manager database view

get money from an ATM machine. This device has a type of operating system that can accept all kinds of commands and configuration changes, but you are physically constrained from being able to carry out these functions. You are presented with buttons that only enable you to withdrawal, view your balance, or deposit funds. Period.

Access Control Matrix

The matrix—let's see, should I take the red pill or the blue pill?

An ***access control matrix*** is a table of subjects and objects indicating what actions individual subjects can take upon individual objects. Matrices are data structures that programmers implement as table lookups that will be used and enforced by the operating system. Table 4-1 provides an example of an access control matrix.

This type of access control is usually an attribute of DAC models. The access rights can be assigned directly to the subjects (capabilities) or to the objects (ACLs).

Capability Tables

A ***capability table*** specifies the access rights a certain subject possesses pertaining to specific objects. A capability table is different from an ACL because the subject is bound to the capability table, whereas the object is bound to the ACL.

The capability corresponds to the subject's row in the access control matrix. In Table 4-1, Diane's capabilities are File1: read and execute; File2: read, write, and execute; File3: no access. This outlines what Diane is capable of doing to each resource. An example of a capability-based system is Kerberos. In this environment, the user is given a ticket, which is his capability table. This ticket is bound to the user and dictates what objects that user can access and to what extent. The access control is based on this ticket, or capability table, not on an ACL bound to the object, although ACLs and capability tables can be used together. Figure 4-11 shows the difference between a capability table and an ACL.

A capability can be in the form of a token, ticket, or key. When a subject presents a capability component, the operating system (or application) will review the access rights and operations that are outlined in the capability component and allow the subject to carry out just those functions. A capability component is a data structure that contains a unique object identifier and access rights the subject has to that object. The object may be a file, array, memory segment, or port. Each user, process, and application in a capability system has a list of capabilities.

User	File1	File2	File3
Diane	Read and execute	Read, write, and execute	No access
Katie	Read and execute	Read	No access
Chrissy	Read, write, and execute	Read and execute	Read
John	Read and execute	No access	Read and write

Table 4-1 Example of an Access Control Matrix

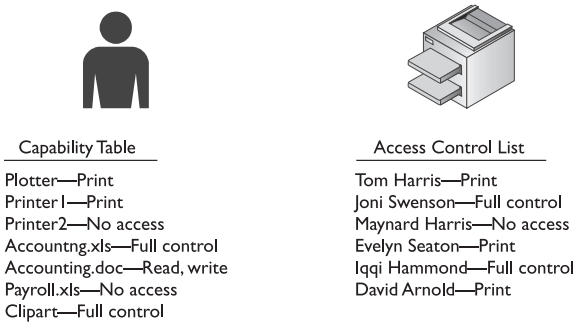


Figure 4-11 A capability table is bound to a subject, whereas an ACL is bound to an object.

Access Control Lists

Access control lists (ACLs) are used in several operating systems, applications, and router configurations. They are lists of subjects that are authorized to access a specific object and they define what level of authorization is granted. Authorization can be specified to an individual or group.

ACLs map values from the access control matrix to the object. Whereas a capability corresponds to a row in the access control matrix, the ACL corresponds to a column of the matrix. The ACL for File1 in Table 4-1 is shown in Table 4-2.

Content-Dependent Access Control

As the name suggests, with ***content-dependent access control***, access to objects is determined by the content within the object. This often is used in databases. The earlier example pertaining to database views showed how content-dependent access control can work. The content of the database fields dictates which users can see specific information within the database tables.

Let's say we give department managers access to the payroll database, but specific managers would be given access only to the records that display those employees who actually work for them. That would require the access control system to look at the contents of the data in order to decide whether to give access or not. Therefore, that would be considered content-dependent access control.

Content-dependent filtering is used when corporations employ e-mail filters that look for specific strings, such as "confidential", "social security number", "top secret", and any other types of words that the company deems unacceptable. Corporations also

Table 4-2 The ACL for File1

User	File1
Diane	Read and execute
Katie	Read and execute
Chrissy	Read, write, and execute
John	Read and execute

have this in place to control web surfing, where filtering is done to look for specific words, to try to figure out whether employees are gambling or looking at pornography.

Context-Dependent Access Control

First you kissed a parrot, then you threw your shoe, and then you did a jig. That's the right sequence, you are allowed access.

Context-dependent access control differs from content-dependent access control in that it makes access decisions based on the context of a collection of information rather than on the sensitivity of the data. A system that is using context-dependent access control "reviews the situation" and then makes a decision. For example, firewalls make context-based access decisions when they collect state information on a packet before allowing it into the network. A stateful firewall understands the necessary steps of communication for specific protocols. For example, in a TCP connection, the sender sends an SYN packet, the receiver sends an SYN/ACK, and then the sender acknowledges that packet with an ACK packet. A stateful firewall understands these different steps and will not allow packets to go through that do not follow this sequence. So, if a stateful firewall receives a SYN/ACK and there was not a previous SYN packet that correlates with this connection, the firewall understands that this is not right and disregards the packet. This is what stateful means—something that understands the necessary steps of a dialog session. And this is an example of context-dependent access control, where the firewall understands the *context* of what is going on and includes that as part of its access decision.

Access Control Administration

Once an organization develops a security policy, supporting procedures, standards, and guidelines (described in Chapter 3), it must choose the type of access control model: DAC, MAC, or role-based. After choosing a model, the organization needs to select and implement different access control technologies and techniques. Access control matrices, restricted interfaces, and content-dependent, context-dependent, and rule-based controls are just a few of the choices.

If the environment does not require a high level of security, the organization will choose discretionary and/or role-based. The DAC model enables data owners to allow other users to access their resources, so an organization should choose the DAC model only if it is fully aware of what it entails. If an organization has a high turnover rate and/or requires a more centralized access control method, the role-based model is more appropriate. If the environment requires a higher security level and only the administrator should be able to grant access to resources, then a MAC model is the best choice.

What is left to work out is how the organization will administer the access control model. Access control administration comes in two basic flavors: centralized and decentralized. The decision makers should understand both approaches so that they choose and implement the proper one to achieve the level of protection required.

Access Control Techniques

Access control techniques are used to support the access control models.

- **Access control matrix** Table of subjects and objects that outlines their access relationships
- **ACL** Bound to an object and indicates what subjects can access it
- **Capability table** Bound to a subject and indicates what objects that subject can access
- **Content-based access** Bases access decisions on the sensitivity of the data, not solely on subject identity
- **Context-based access** Bases access decisions on the state of the situation, not solely on identity or content sensitivity
- **Restricted interface** Limits the user's environment within the system, thus limiting access to objects
- **Rule-based** Restricts subjects' access attempts by predefined rules

Centralized Access Control Administration

I control who can touch the carrots and who can touch the peas. Response: Could you leave now?

A ***centralized access control administration*** method is basically what it sounds like: one entity (department or individual) is responsible for overseeing access to all corporate resources. This entity (security administrator) configures the mechanisms that enforce access control, processes any changes that are needed to a user's access control profile, disables access when necessary, and completely removes these rights when a user is terminated, leaves the company, or moves to a different position. This type of administration provides a consistent and uniform method of controlling users' access rights. It supplies strict control over data because only one entity (department or individual) has the necessary rights to change access control profiles and permissions. Although this provides for a more consistent and reliable environment, it can be a slow one, because all changes must be processed by one entity.

The following sections present some examples of centralized remote access control technologies. Each of these authentication protocols is referred to as an AAA protocol, which stands for authentication, authorization, and auditing. (Some resources have the last *A* stand for accounting, but it is the same functionality—just a different name.)

Depending upon the protocol, there are different ways to authenticate a user in this client/server architecture. The traditional authentication protocols are Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP),

and a newer method referred to as Extensible Authentication Protocol (EAP). Each of these authentication protocols is discussed at length in Chapter 7.

RADIUS

So, I have to run across half of a circle to be authenticated? Response: Don't know. Give it a try.

Remote Authentication Dial-In User Service (RADIUS) is a client/server authentication protocol that authenticates and authorizes remote users. A network may have access servers, a modem pool, DSL, ISDN, or T1 line dedicated for remote users to communicate through. The access server requests the remote user's logon credentials and passes them back to a RADIUS server, which houses the usernames and password values. The remote user is a client to the access server, and the access server is a client to the RADIUS server.

Most ISPs today use RADIUS to authenticate customers before they are allowed access to the Internet. The access server and customer's software negotiate, through a handshake procedure, and agree upon an authentication protocol (PAP, CHAP, or EAP). The customer provides to the access server a username and password. This communication takes place over a PPP connection. The access server and RADIUS server communicate over the RADIUS protocol. Once the authentication is completed properly, the customer's system is given an IP address and connection parameters, and is allowed access to the Internet. The access server notifies the RADIUS server when the session starts and stops, for billing purposes.

RADIUS is also used within corporate environments to provide road warriors and home users access to network resources. RADIUS allows companies to maintain user profiles in a central database. When a user dials in and is properly authenticated, a preconfigured profile is assigned to him to control what resources he can and cannot access. This technology allows companies to have a single administered entry point, which provides standardization in security and a simplistic way to track usage and network statistics.

RADIUS was developed by Livingston Enterprises for its network access server product series, but was then published as RFC 2138 and RFC 2139. This means that it is an open protocol that any vendor can use and manipulate to be able to work within its individual products. Because RADIUS is an open protocol, it can be used in different types of implementations. The format of configurations and user credentials can be held in LDAP servers, various databases, or text files. Figure 4-12 shows some examples of possible RADIUS implementations.

TACACS

Terminal Access Controller Access Control System (TACACS) has a very funny name. Not funny ha-ha, but funny "huh?" TACACS has been through three generations: TACACS, Extended TACACS (XTACACS), and TACACS+. TACACS combines its authentication and authorization processes, XTACACS separates authentication, authorization, and auditing processes, and TACACS+ is XTACACS with extended two-factor user authentication. TACACS uses fixed passwords for authentication and TACACS+ allows users to use dynamic (one-time) passwords, which provides more protection.

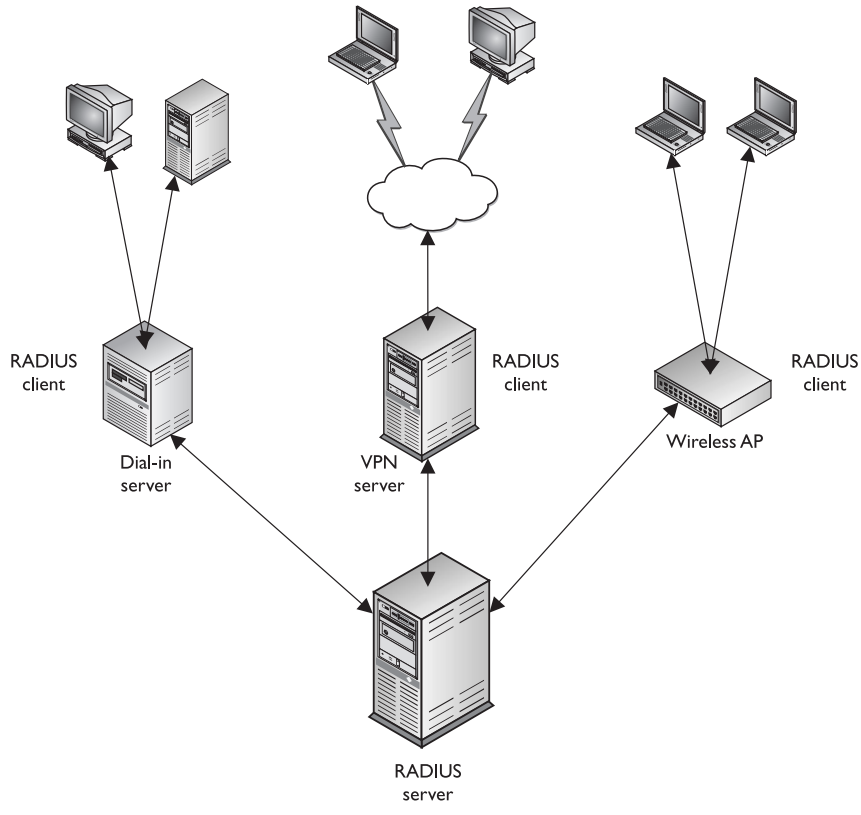


Figure 4-12 Environments can implement various different RADIUS infrastructures.

NOTE TACACS+ is really not a new generation of TACACS and XTACACS; it is a brand new protocol that provides similar functionality and shares the same naming scheme. Because it is a totally different protocol, it is not backward compatible with TACACS or XTACACS.

TACACS+ provides basically the same functionality as RADIUS with a few differences in some of its characteristics. First, TACACS+ uses TCP as its transport protocol, while RADIUS uses UDP. "So what?" you may be thinking. Well, any software that is developed to use UDP as its transport protocol has to be "fatter" with intelligent code that will look out for the items that UDP will not catch. Since UDP is a connectionless protocol, it will not detect or correct transmission errors. So RADIUS has to have the necessary code to detect packet corruption, long timeouts, or dropped packets. Since the developers of TACACS+ choose to use TCP, the TACACS+ software does not have to have the extra code to look for and deal with these transmission problems. TCP is a connection-oriented protocol, and that is its job and responsibility.

RADIUS encrypts the user's password only as it is being transmitted from the RADIUS client to the RADIUS server. Other information, as in the username, accounting, and

authorized services, is passed in cleartext. This is an open invitation for attackers to capture session information for replay attacks. Vendors who integrate RADIUS into their products need to understand these weaknesses and integrate other security mechanisms to protect against these types of attacks.

TACACS+ encrypts all of this data and thus does not have the vulnerabilities that are inherent in the RADIUS protocol.

The RADIUS protocol combines the authentication and authorization functionality. TACACS+ uses a true AAA architecture, which separates the authentication, authorization, and accounting functionalities. This gives a network administrator more flexibility in how remote users are authenticated. For example, if Tom is a network administrator and has been assigned the task of setting up remote access for users, he will have to decide between RADIUS and TACACS+. If the current environment already authenticates all of the local users through a domain controller using Kerberos, then Tom can configure the remote users to be authenticated in this same manner, as shown in Figure 4-13. Instead of having to maintain a remote access server database of remote user credentials and a database within Active Directory for local users, Tom can just configure and maintain one database. The separation of authentication, authorization, and accounting functionality provides this capability. TACACS+ also enables the network administrator to define more granular user profiles, which can control the actual commands that users can carry out.

Remember that RADIUS and TACACS+ are both protocols, and protocols are just agreed-upon ways of communication. When a RADIUS client communicates with a RADIUS server, it does so through the RADIUS protocol, which is really just a set of defined fields that will accept certain values. These fields are referred to as attribute-value pairs (AVPs). As an analogy, suppose that I send you a piece of paper that has several different boxes drawn on it. Each box has a headline associated with it: first name, last name, hair color, shoe size. You fill in these boxes with your values and send it back to me. This is basically how protocols work; the sending system just fills in the boxes (fields) with the necessary information for the receiving system to extract and process.

Since TACACS+ allows for more granular control on what users can and cannot do, TACACS+ has more AVPs, which allows the network administrator to define ACLs, filters, user privileges, and much more.

Watchdog

Watchdog timers are commonly used to detect software faults, such as a process ending abnormally or hanging. The watchdog functionality sends out a type of "heartbeat" packet to determine whether a service is responding; if it is not, the process can be terminated or reset. This guards against software deadlocks, infinite loops, and process prioritization problems. This functionality can be used in AAA protocols to determine whether packets need to be resent and whether connections that are experiencing problems need to be closed and reopened.

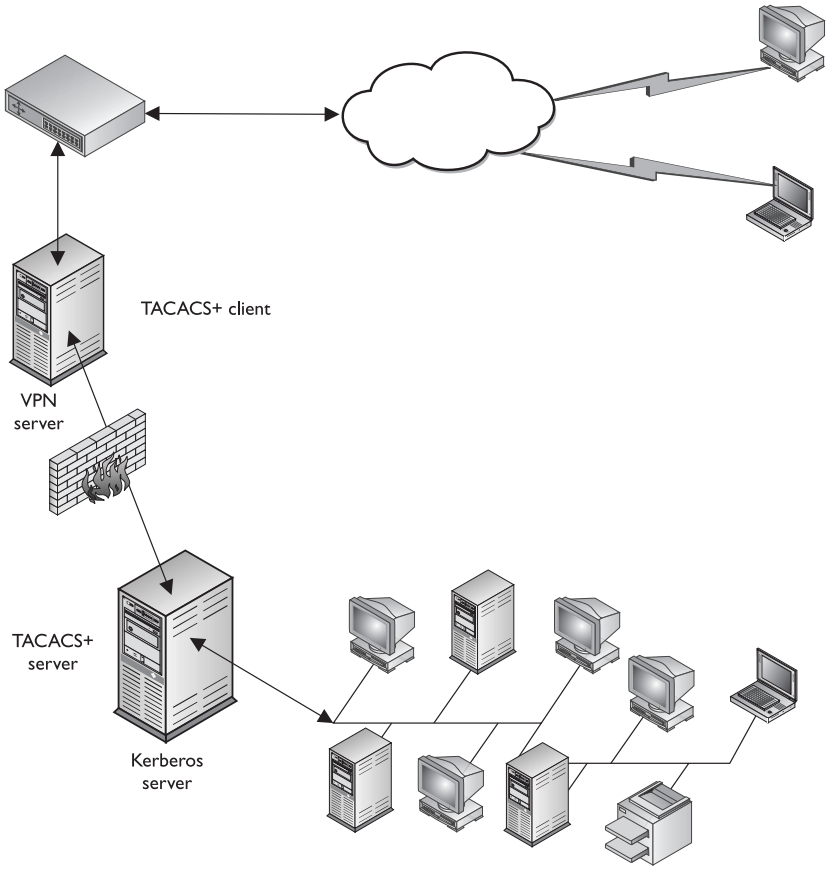


Figure 4-13 TACACS+ works in a client/server model.

So, RADIUS is the appropriate protocol when simplistic username/password authentication can take place and users only need an Accept or Deny for obtaining access, as in ISPs. TACACS+ is the better choice for environments that require more sophisticated authentication steps and tighter control over more complex authorization activities, as in corporate networks.

Diameter

If we create our own technology, we get to name it any goofy thing we want! Response: I like Snizzernoodle.

Diameter is a protocol that has been developed to build upon the functionality of RADIUS and overcome many of its limitations. The creators of this protocol decided to call it Diameter as a play on the term RADIUS, as in *the diameter is twice the radius.*

Diameter is another AAA protocol that provides the same type of functionality as RADIUS and TACACS+ but also provides more flexibility and capabilities to meet the

Mobile IP

This technology allows a user to move from one network to another and still use the same IP address. It is an improvement upon the IP protocol because it allows a user to have a *home IP address,* associated with his home network, and a *care-of address*. The care-of address changes as he moves from one network to the other. All traffic that is addressed to his home IP address is forwarded to his care-of address.

new demands of today's complex and diverse networks. At one time, all remote communication took place over PPP and SLIP connections and users authenticated themselves through PAP or CHAP. Those were simpler, happier times when our parents had to walk uphill both ways to school wearing no shoes. As with life, technology has become much more complicated and there are more devices and protocols to choose from than ever before. Today, we want our wireless devices and smart phones to be able to authenticate themselves to our networks and we use roaming protocols, Mobile IP, Ethernet over PPP, and other crazy stuff that the traditional AAA protocols cannot keep up with. So in came the smart people with a new AAA protocol, Diameter, that can deal with these issues and many more.

Up until the conception of Diameter, IETF has had individual working groups who defined how Voice over IP (VoIP), Fax over IP (FoIP), Mobile IP, and remote authentication protocols work. Defining and implementing them individually in any network can easily result in too much confusion and interoperability. It requires customers to roll out and configure several different policy servers and increases the cost with each new added service. Diameter provides a base protocol, which defines header formats, security options, commands, and AVPs. This base protocol allows for extensions to tie in other services, such as VoIP, FoIP, Mobile IP, wireless, and cell phone authentication. So Diameter can be used as an AAA protocol for all of these different uses.

As an analogy, consider a scenario in which ten people all need to get to the same hospital, which is where they all work. They all have different jobs (doctor, lab technician, nurse, janitor, and so on), but they all need to end up at the same location. So, they can either all take their own cars and their own routes to the hospital, which takes up more hospital parking space and requires the gate guard to authenticate each and every car, or they can take a bus. The bus is the common element (base protocol) to get the individuals (different services) to the same location (networked environment). Diameter provides the common AAA and security framework that different services can work within, as illustrated in Figure 4-14.

NOTE Roaming Operations (ROAMOPS) allows PPP users to gain access to the Internet without the need of dialing into their home service provider. The individual service providers, who have roaming agreements, carry out cross-authentication for their customers, so users can dial into any service provider's point of presence and gain Internet access.

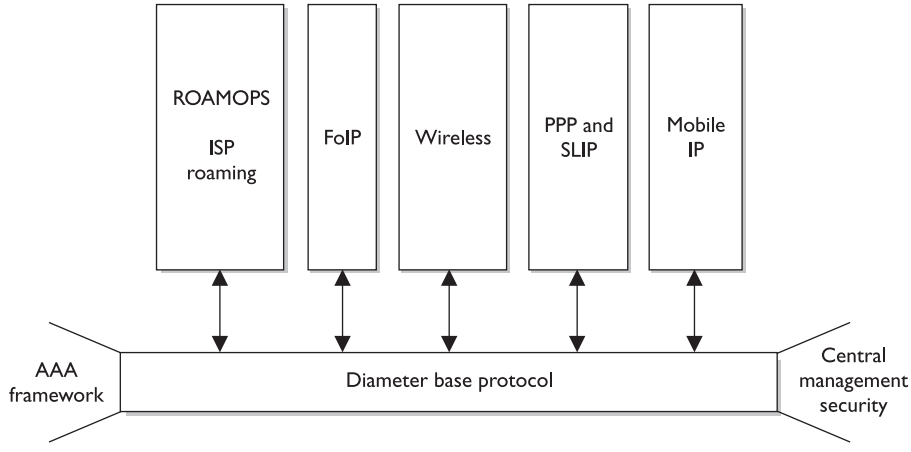


Figure 4-14 Diameter provides an AAA architecture for several services.

RADIUS and TACACS+ are client/server protocols, which means that the server portion cannot send unsolicited commands to the client portion. The server portion can only speak when spoken to. Diameter is a peer-based protocol that allows either end to initiate communication. This functionality allows the Diameter server to send a message to the access server to request the user to provide another authentication credential if she is attempting to access a secure resource. This functionality also allows the Diameter server to disconnect the user if necessary for one reason or another.

Diameter is backward compatible with RADIUS, uses UDP and AVPs, and provides proxy server support. It has better error detection and correction functionality and fail-over properties than RADIUS, thus provides better network resilience. Diameter also provides end-to-end security through the use of IPSec or TLS, which is not available in RADIUS.

Diameter has the functionality and ability to provide the AAA functionality for other protocols and services because it has a large AVP set. RADIUS has 2^8 (256) AVPs and Diameter has 2^{32} (a whole bunch). Recall from earlier in the chapter that AVPs are like boxes drawn on a piece of paper that outline how two entities can communicate back and forth. So, more AVPs allow for more functionality and services to exist and communicate between systems. Diameter provides the following AAA functionality:

- **Authentication**
  - PAP, CHAP, EAP
  - End-to-end protection of authentication information
  - Replay attack protection
- **Authorization**
  - Redirects, secure proxies, relays, and brokers
  - State reconciliation

  - Unsolicited disconnect
  - Reauthorization on demand
- **Accounting**
  - Reporting, ROAMOPS accounting, event monitoring

You may not be familiar with Diameter, because it is relatively new. It probably won't be taking over the world tomorrow, but it will be used by ISPs that need to provide the type of services that are being demanded of them, and then slowly seep down into corporate networks as more products are available. RADIUS has been around for a long time and has served its purpose well, so don't expect it to exit the stage any time soon.

References

- **RFC 3588 – Diameter Base Protocol** www.faqs.org/rfcs/rfc3588.html
- **RFC 2869 – RADIUS Extensions** www.faqs.org/rfcs/rfc2869.html
- **RFC 2865 – Remote Authentication Dial In User Service (RADIUS)** www.faqs.org/rfcs/rfc2865.html
- **RFC 2975 – Introduction to Accounting Management** www.faqs.org/rfcs/rfc2975.html

Decentralized Access Control Administration

Okay, everyone just do whatever you want.

A ***decentralized access control administration*** method gives control of access to the people closer to the resources—the people who may better understand who should and should not have access to certain files, data, and resources. In this approach, it is often the functional manager who assigns access control rights to employees. An organization may choose to use a decentralized model if its managers have better judgment regarding which users should be able to access different resources, and there is no business requirement that dictates that strict control through a centralized body is necessary.

Changes can happen faster through this type of administration because not just one entity is making changes for the whole organization. However, there is a possibility that conflicts of interest could arise that may not benefit the organization. Because no single entity controls access as a whole, different managers and departments can practice security and access control in different ways. This does not provide uniformity and fairness across the organization. One manager could be too busy with daily tasks and decide that it is easier to let everyone have full control over all the systems in the department. Another department may practice a more strict and detail-oriented method of control by giving employees only the level of permissions needed to fulfill their tasks.

Also, certain controls can overlap, in which case actions may not be properly proscribed or restricted. If Mike is part of the accounting group and recently has been under suspicion for altering personnel account information, the accounting manager may

restrict his access to these files to read-only access. However, the accounting manager does not realize that Mike still has full-control access under the network group he is also a member of. This type of administration does not provide methods for consistent control, as a centralized method would. Another issue that comes up with decentralized administration is lack of proper consistency pertaining to the company's protection. For example, when Sean is fired for looking at pornography on his computer, some of the groups Sean is a member of may not disable his account. So, Sean may still have access after he is terminated, which could cause the company heartache if Sean is vindictive.

Access Control Methods

Access controls can be implemented at various layers of a network and individual systems. Some controls are core components of operating systems or embedded into applications and devices, and some security controls require third-party add-on packages. Although different controls provide different functionality, they should all work together to keep the bad guys out and the good guys in, and to provide the necessary quality of protection.

Most companies do not want people to be able to walk into their building arbitrarily, sit down at an employee's computer, and access network resources. Companies also don't want every employee to be able to access all information within the company, as in human resource records, payroll information, and trade secrets. Companies want to have some assurance that employees who can access confidential information will have some restrictions put upon them so that a disgruntled employee does not have the ability to delete financial statements, tax information, and top-secret data that would put the company at risk. There are several types of access controls that prevent these things from happening, as discussed in the sections that follow.

Access Control Layers

There are three broad categories of access control: administrative, technical, and physical. Each category has different access control mechanisms that can be carried out manually or automatically. All of these access control mechanisms should work in concert with each other to protect an infrastructure and its data.

Each category of access control has several components that fall within it, as shown here:

- **Administrative Controls**
  - Policy and procedures
  - Personnel controls
  - Supervisory structure
  - Security-awareness training
  - Testing

- **Physical Controls**
  - Network segregation
  - Perimeter security
  - Computer controls
  - Work area separation
  - Data backups
  - Cabling
- **Technical Controls**
  - System access
  - Network architecture
  - Network access
  - Encryption and protocols
  - Control zone
  - Auditing

The following sections explain each of these categories and components and how it relates to access control.

Administrative Controls

Senior management must decide what role security will play in the organization, including the security goals and objectives. These directives will dictate how all the supporting mechanisms will fall into place. Basically, senior management provides the skeleton of a security infrastructure and then appoints the proper entities to fill in the rest.

The first piece to building a security foundation within an organization is a security policy. It is management's responsibility to construct a security policy and delegate the development of the supporting procedures, standards, and guidelines, indicate which personnel controls should be used, and specify how testing should be carried out to ensure that all pieces fulfill the company's security goals. These items are ***administrative controls*** and work at the top layer of a hierarchical access control model. (Administrative controls are examined in detail in Chapter 3, but are mentioned here briefly to show the relationship to logical and physical controls pertaining to access control.)

Policy and Procedures

Now, what's our overall plan?

A *security policy* is a high-level plan that states management's intent pertaining to how security should be practiced within an organization, what actions are acceptable, and what level of risk the company is willing to accept. This policy is derived from the laws, regulations, and business objectives that shape and restrict the company. The security policy provides direction for each employee and department regarding how security should be implemented and followed, and the repercussions for noncompliance.

Procedures, guidelines, and standards provide the details that support and enforce the company's security policy.

Personnel Controls

Personnel controls indicate how employees are expected to interact with security mechanisms, and address noncompliance issues pertaining to these expectations. These controls indicate what security actions should be taken when an employee is hired, terminated, suspended, moved into another department, or promoted. Specific procedures need to be developed for each situation, and many times the human resources and legal departments are involved with making these decisions.

The separation of duties and rotation of duties are also personnel controls that need to be dictated by management. The ***separation of duties*** should be enforced so that no one individual can carry out a critical task alone that could prove to be detrimental to the company. A bank teller who has to get supervisory approval to cash checks over $2000 is an example of separation of duties. For a security breach to occur, it would require ***collusion***, which means that more than one person would need to commit fraud, and their efforts would need to be concerted. The use of separation of duties drastically reduces the probability of security breaches and fraud.

Rotation of duties means that people rotate jobs so that they know how to fulfill the obligations of more than one position. In many companies, only one person knows how to work that big important machine that keeps the company in business. If this person gets hit by a bus or finds a better job, the company's productivity can drop while it frantically trains the new person. If there were a practice of rotation of duties within this company, other employees would be able to cover the duties while a replacement is being trained.

Another benefit of rotation of duties is that if an individual attempts to commit fraud within his position, detection is more likely to happen if there is another employee who knows what tasks should be performed in that position and how they should be performed.

Supervisory Structure

Management must construct a supervisory structure in which each employee has a superior to report to, and that superior is responsible for that employee's actions. This forces management members to be responsible for employees and take a vested interest in their activities. If an employee is caught hacking into a server that holds customer credit card information, that employee *and* her supervisor will face the consequences. This is an administrative control that aids in fighting fraud and enforcing proper control.

Security-Awareness Training

How do you know they know what they are supposed to know?

In many organizations, management has a hard time spending money and allocating resources for items that do not seem to affect the bottom line: profitability. This is

why training traditionally has been given low priority, but as computer security becomes more and more of an issue to companies, they are starting to recognize the value of security-awareness training.

A company's security depends upon technology and people, and people are usually the weakest link and cause the most security breaches and compromises. If users understand how to properly access resources, why access controls are in place, and the ramifications for not using the access controls properly, a company can reduce many of the types of security incidents that take place.

Testing

All security controls, mechanisms, and procedures need to be tested on a periodic basis to ensure that they properly support the security policy, goals, and objectives set for them. This testing can be a drill to test reactions to a physical attack or disruption of the network, a penetration test of the firewalls and perimeter network to uncover vulnerabilities, a query to employees to gauge their knowledge, or a review of the procedures and standards to make sure they still align with business or technology changes that have been implemented. Because change is constant and environments continually evolve, security procedures and practices should be continually tested to ensure that they align with management's expectations and stay up to date with each addition to the infrastructure. It is management's responsibility to make sure these tests take place.

Physical Controls

We will go much further into physical security in Chapter 6, but it is important to understand that there are physical controls that must support and work with administrative and technical (logical) controls to supply the right degree of access control. Examples of physical controls include having a security guard verify individuals' identities prior to entering a facility, erecting fences around the exterior of the facility, making sure server rooms and wiring closets are locked and protected from environmental elements (humidity, heat, and cold), and allowing only certain individuals to access work areas that contain confidential information. Some physical controls are introduced next, but again, these and more physical mechanisms are explored in depth in Chapter 6.

Network Segregation

I have used my LEGO set to outline the physical boundaries between you and me. Response: Can you make the walls a little higher please?

Network segregation can be carried out through physical and logical means. A network might be physically designed to have all AS400 computers and databases in a certain area. This area may have doors with security swipe cards that allow only individuals who have a specific clearance to access this section and these computers. Another section of the network may contain web servers, routers, and switches, and yet another network portion may have employee workstations. Each area would have the necessary physical controls to ensure that only the permitted individuals have access into and out of those sections.

Perimeter Security

How perimeter security is implemented depends upon the company and the security requirements of that environment. One environment may require employees to be authorized by a security guard by showing a security badge that contains picture identification before being allowed to enter a section. Another environment may require no authentication process and let anyone and everyone into different sections. Perimeter security can also encompass closed-circuit TVs that scan the parking lots and waiting areas, fences surrounding a building, lighting of walkways and parking areas, motion detectors, sensors, alarms, and the location and visual appearance of a building. These are examples of perimeter security mechanisms that provide physical access control by providing protection for individuals, facilities, and the components within facilities.

Computer Controls

Each computer can have physical controls installed and configured, such as locks on the cover so that the internal parts cannot be stolen, the removal of the floppy and CD-ROM drives to prevent copying of confidential information, or implementation of a protection device that reduces the electrical emissions to thwart attempts to gather information through airwaves.

Work Area Separation

Some environments might dictate that only particular individuals can access certain areas of the facility. For example, research companies might not want office personnel to be able to enter laboratories, so that they can't disrupt experiments or access test data. Most network administrators allow only network staff in the server rooms and wiring closets, to reduce the possibilities of errors or sabotage attempts. In financial institutions, only certain employees can enter the vaults or other restricted areas. These examples of work area separation are physical controls that are used to support access control and the overall security policy of the company.

Data Backups

Backing up data is a physical control to ensure that information can still be accessed after an emergency or a disruption of the network or a system. When a network administrator of a bank backs up that day's transaction records, account histories, and financial information, it is stored on a type of physical media (tape, drive, or CD-ROM) and usually kept in a fireproof safe or copied and transported to an offsite facility. This provides physical protection of this data and a way to recover this information in case the original data is lost.

Cabling

There are different types of cabling that can be used to carry information throughout a network. Some cable types have sheaths that protect the data from being affected by the electrical interference of other devices that emit electrical signals. Some types of cable have protection material around each individual wire to ensure that there is no crosstalk between the different wires. All cables need to be routed throughout the facility in

a manner that is not in people's way or that could be exposed to any danger of being cut, burnt, crimped, or eavesdropped upon.

Technical Controls

Technical controls, also called logical controls, are the software tools used to restrict subjects' access to objects. They are core components of operating systems, add-on security packages, applications, network hardware devices, protocols, encryption mechanisms, and access control matrixes. These controls work at different layers within a network or system and need to maintain a synergistic relationship to ensure that there is no unauthorized access to resources and that the resources' availability, integrity, and confidentiality are guaranteed. Technical controls protect the integrity and availability of resources by limiting the number of subjects that can access them and protect the confidentiality of resources by preventing disclosure to unauthorized subjects. The following sections explain how some technical controls work and where they are implemented within an environment.

System Access

Different types of controls and security mechanisms control how a computer is accessed. If an organization is using a MAC architecture, the clearance of a user is identified and compared to the resource's classification level to verify that this user can access the requested object. If an organization is using a DAC architecture, the operating system checks to see if a user has been granted permission to access this resource. The sensitivity of data, clearance level of users, and users' rights and permissions are used as logical controls to control access to a resource.

There are many types of technical controls that enable a user to access a system and the resources within that system. A technical control may be a username and password combination, a Kerberos implementation, biometrics, public key infrastructure (PKI), RADIUS, TACACS, or authentication using a smart card through a reader connected to a system. These technologies verify that the user is who he says he is by using different types of authentication methods. Once a user is properly authenticated, he can be authorized and allowed access to network resources. These technologies are addressed in further detail in future chapters, but for now understand that system access is a type of technical control that can enforce access control objectives.

Network Architecture

The architecture of a network can be constructed and enforced through several logical controls to provide segregation and protection of an environment. Whereas a network can be segregated physically by walls and location, it can also be segregated logically through IP address ranges and subnets and by controlling the communication flow between the segments. Often, it is important to control how one segment of a network communicates with another segment.

Figure 4-15 is an example of how an organization may segregate its network and determine how network segments can communicate. This example shows that the organization does not want the internal network and the demilitarized zone (DMZ) to

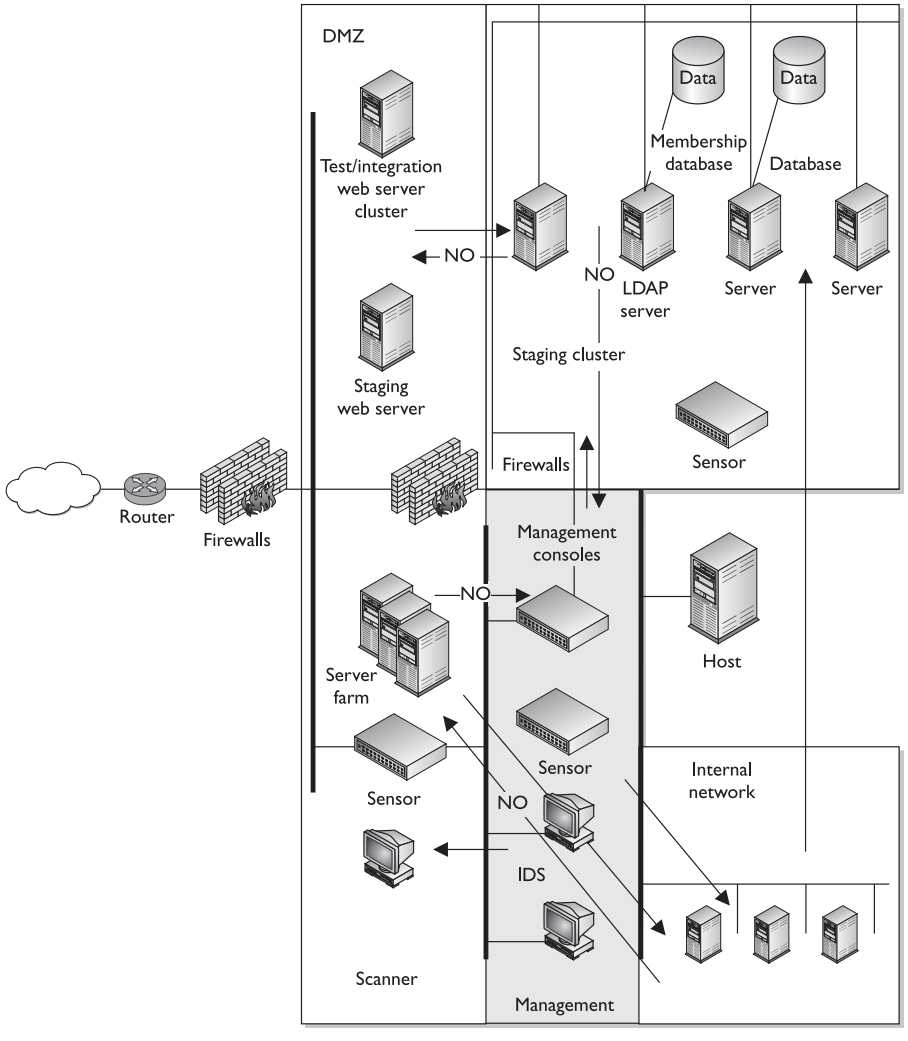


Figure 4-15 Technical network segmentation controls how different network segments communicate.

have open and unrestricted communication paths. There is usually no reason for internal users to have direct access to the systems in the DMZ, and cutting off this type of communication reduces the possibilities of internal attacks on those systems. Also, if an attack comes from the Internet and successfully compromises a system on the DMZ, the attacker must not be able to easily access the internal network, which this type of logical segregation protects against.

This example also shows how the management segment can communicate with all other network segments, but those segments cannot communicate in return. The segmentation is implemented because the management consoles that control the firewalls

and IDSs reside in the management segment, and there is no reason for users, other than the administrator, to have access to these computers.

A network can be segregated physically and logically. This type of segregation and restriction is accomplished through logical controls.

Network Access

Systems have logical controls that dictate who can and cannot access them and what those individuals can do once they are authenticated. This is also true for networks. Routers, switches, firewalls, and bridges all work as technical controls to enforce access restriction into and out of a network, and access to the different segments within the network. If an attacker from the Internet wants to gain access to a specific computer, chances are that she will have to hack through a firewall, router, and a switch just to be able to start an attack on a specific computer that resides within the internal network. Each device has its own logical controls that make decisions about what entities can access them and what type of actions they can carry out.

Access to different network segments should be granular in nature. Routers and firewalls can be used to ensure that only certain types of traffic get through to each segment.

Encryption and Protocols

Encryption and protocols work as technical controls to protect information as it passes throughout a network and resides on computers. They ensure that the information is received by the correct entity, and that it is not modified during transmission. These logical controls can preserve the confidentiality and integrity of data and enforce specific paths for communication to take place. (Chapter 8 is dedicated to cryptography and encryption mechanisms.)

Control Zone

I will ensure that no signal escapes this room by using my incredible mental abilities. Response: How about using a control zone instead?

A ***control zone*** is physical control. It is a specific area that surrounds and protects network devices that emit electrical signals. These electrical signals can travel a certain distance and can be contained by a specially made material, which is used to construct the control zone. Some companies use extra metallic material in their building walls to zone off areas that have computers that deal with top-secret information. The control zone is used to resist penetration attempts and disallow sensitive information to "escape" through the airwaves. We will address how information can be captured from emitted electrical waves in the "Emanation Security" section of this chapter, but for now understand that a control zone is used as a physical control to ensure that confidential information is contained and to hinder intruders from accessing information through the airwaves. Figure 4-16 depicts a control zone. Companies that have very sensitive information would likely protect that information by creating control zones around the systems that are processing that information.

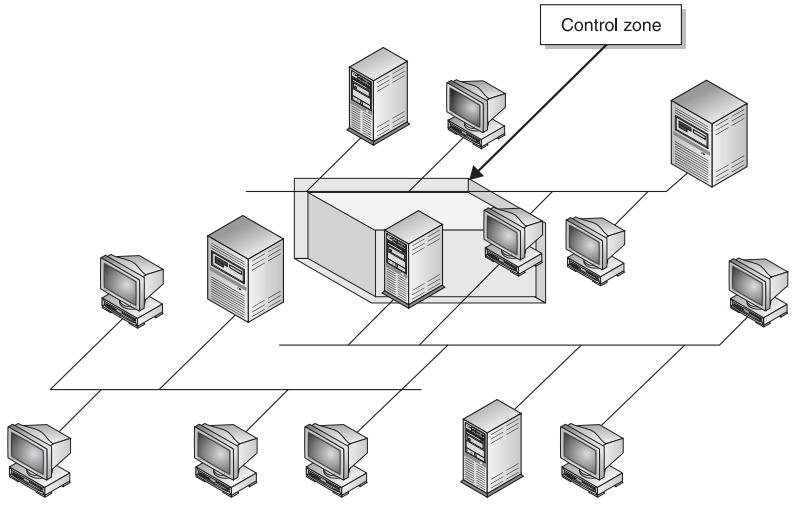


Figure 4-16 Companies can implement control zones, which control the amount of electrical signals that are emitted from sections of the facility.

Auditing

Auditing tools are technical controls that track activity within a network, on a network device, or on a specific computer. Even though auditing is not an activity that will deny an entity access to a network or computer, it will track activities so that a network administrator can understand the types of access that took place, identify a security breach, or warn the administrator of suspicious activity. This information can be used to point out weaknesses of other technical controls and help the administrator understand where changes need to be made to preserve the necessary security level within the environment.

NOTE Many of the subjects touched on in these sections will be fully addressed and explained in later chapters. What is important to understand is that there are administrative, technical, and physical controls that work toward providing access control, and you should know several examples of each for the exam.

Access Control Types

As previously stated, access control types (administrative, physical, and technical) work at different levels, but different levels of what? They work together at different levels within their own categories. A security guard is a type of control used to scare off evildoers and ensure that only authorized personnel enter a building. If an intruder gets around the security guard in some manner, he could be faced with motion detectors, locks on doors, and alarms. These layers are depicted in Figure 4-17.

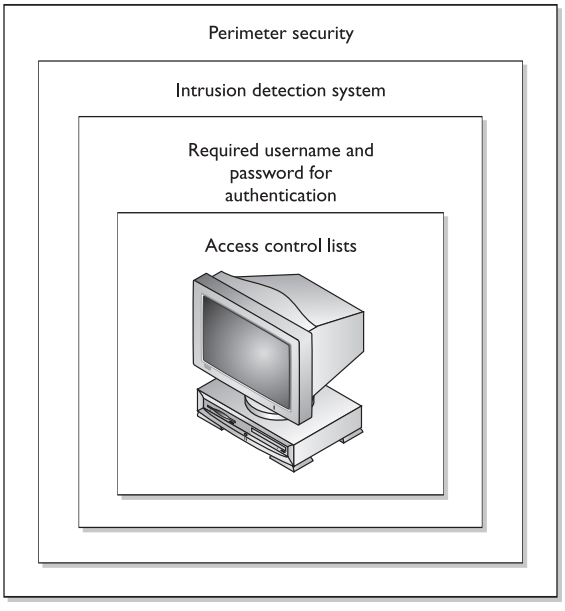


Figure 4-17 Security should be implemented in layers, which provides several barriers to attackers.

Each control works at a different level of granularity, but it can also perform different functionalities. The different functionalities of access controls are *preventive, detective, corrective, deterrent, recovery,* and *compensative.* When looking at a security structure of an environment, it is most productive to use a preventive model and then use detective, recovery, and corrective mechanisms to help support this model. Basically, you want to stop any trouble before it starts, but you need to be able to quickly react and combat trouble if it does find you. All security controls should be built on the concept of preventive security. However, it is not feasible to prevent everything; therefore, what you cannot prevent, you should be able to quickly detect. That's why preventive and detective controls should always be implemented together and should complement each other. To take this concept further, what you can't prevent, you should be able to detect, and if you detect something, it means that you weren't able to prevent it, and therefore you should take corrective action to make sure that it is indeed prevented the next time around. Therefore, all three types work together, preventive, detective, and corrective.

The control types described next (administrative, physical, and technical) are preventive in nature. These are important to understand when developing a security access control model and when taking the CISSP exam.

Preventive: Administrative

The following are the *soft* mechanisms that are put into place to enforce access control and protection for the company as a whole:

- Policies and procedures
- Effective hiring practices
- Pre-employment background checks
- Controlled termination processes
- Data classification and labeling
- Security awareness

Preventive: Physical

The following can physically restrict access to a facility, specific work areas, or computer systems:

- Badges, swipe cards
- Guards, dogs
- Fences, locks, mantraps

Preventive: Technical

The following are logical controls that are part of operating systems, third-party application add-ons, or hardware units:

- Passwords, biometrics, smart cards
- Encryption, protocols, call-back systems, database views, constrained user interfaces
- Antivirus software, ACLs, firewalls, routers, clipping levels

Table 4-3 shows how these categories of access control mechanisms perform different security functions. However, Table 4-3 does not necessarily cover all the possibilities. For example, a fence can provide preventive and deterrent measures by making it harder for intruders to access a facility, but it could also be a compensative control. If a company cannot afford a security guard, it might erect a fence to act as the compensative physical control. Each control is able to meet more requirements than what is listed in the table. Table 4-3 is only an example to show the relationship among the different controls and the security attributes they could provide.

NOTE Locks are usually considered delay mechanisms because they only delay a determined intruder. The goal is to delay access long enough to allow law enforcement or the security guard to respond to the situation.

Type of Control:	Preventive	Detective	Corrective	Deterrent	Recovery	Compensative
	Avoid undesirable events from occurring	Identify undesirable events that have occurred	Correct undesirable events that have occurred	Discourage security violations	Restore resources and capabilities	Provide alternatives to other controls
Category of Control:						
Physical						
Fences	X			X		
Locks	X			X		
Badge system	X			X		
Security guard	X	X		X		
Biometric system	X					
Mantrap doors	X			X		
Lighting	X					
Motion detectors		X				
Closed-circuit TVs		X		X		
Alarms	X	X		X		
Backups					X	
Administrative						
Security policy	X					
Monitoring and supervising	X	X		X		X
Separation of duties	X					
Job rotation		X				

Table 4-3 Services That Security Controls Provide

Type of Control:	Preventive	Detective	Corrective	Deterrent	Recovery	Compensative
Information classification	X					
Personnel procedures	X			X		X
Investigations		X				
Testing	X					
Security-awareness training	X			X		
Technical						
ACLs	X					
Routers	X					
Encryption	X					
Audit logs		X				
IDS		X				
Antivirus software	X	X	X		X	
Firewalls	X			X		
Smart cards	X					
Dial-up call-back systems	X					
Alarms and alerts		X				

Table 4-3 Services That Security Controls Provide *(continued)*

There are several types of security mechanisms, and they all need to work together. The complexity of the controls and of the environment they are in can cause the controls to contradict each other or leave gaps in security. This can introduce unforeseen holes in the company's protection that are not fully understood by the implementers. A company may have very strict technical access controls in place and all the necessary administrative controls up to snuff, but if any person is allowed to physically access any system in the facility, then clear security dangers are present within the environment. Together these controls should work in harmony to provide a healthy, safe, and productive environment.

Accountability

Auditing capabilities ensure that users are accountable for their actions, verify that the security policies are enforced, and can be used as investigation tools. There are several reasons why network administrators and security professionals want to make sure accountability mechanisms are in place and configured properly: to be able to track bad deeds back to individuals, detect intrusions, reconstruct events and system conditions, provide legal recourse material, and produce problem reports. Audit documentation and log files hold a mountain of information—the trick is usually deciphering it and presenting it in a useful and understandable format.

Accountability is tracked by recording user, system, and application activities. This recording is done through auditing functions and mechanisms within an operating system or application. Audit trails contain information about operating system activities, application events, and user actions. Audit trails can be used to verify the health of a system by checking out performance information or certain types of errors and conditions. After a system crashes, a network administrator often will review audit logs to try and piece together the status of the system and attempt to understand what events could be attributed to the disruption.

An administrator configures what actions and events are to be audited and logged. In a high-security environment, the administrator would configure more activities to be captured and set the threshold of those activities to be more sensitive. The events can be reviewed to identify where breaches of security occurred and if the security policy had been violated. If the environment does not require such level of security, the events analyzed would be fewer with less demanding thresholds.

Items and actions to be audited can become an endless list. A security professional should be able to assess an environment and its security goals, know what actions should be audited, and know what is to be done with that information after it is captured—without wasting too much disk space, CPU power, and staff time. The following gives a broad overview of the items and actions that can be audited and logged:

- **System-level events**
  - System performance
  - Logon attempts (successful and unsuccessful)

  - Logon ID
  - Date and time of each logon attempt
  - Lockouts of users and terminals
  - Use of administration utilities
  - Devices used
  - Functions performed
  - Requests to alter configuration files
- **Application-level events**
  - Error messages
  - Files opened and closed
  - Modifications of files
  - Security violations within application
- **User-level events**
  - Identification and authentication attempts
  - Files, services, and resources used
  - Commands initiated
  - Security violations

The threshold (clipping level) and parameters for each of these items needs to be configured. For example, an administrator can audit each logon attempt or just each failed logon attempt. System performance can look only at the amount of memory used within an eight-hour period, or the memory, CPU, and hard drive space used within an hour.

Intrusion detection systems (IDSs) continually scan audit logs for suspicious activity. If an intrusion or harmful event takes place, audit logs are usually kept to be used later to prove guilt and prosecute if necessary. If severe security events take place, many times the IDS will alert the administrator or staff member so that they can take proper actions to end the destructive activity. If a dangerous virus is identified, administrators may take the mail server offline. If an attacker is accessing confidential information within the database, this computer may be temporarily disconnected from the network or Internet. If an attack is in progress, the administrator may want to watch the actions taking place so that she can track down the intruder. IDSs can watch for this type of activity during real time and/or scan audit logs and watch for specific patterns or behaviors.

Review of Audit Information

It does no good to collect it if you don't look at it.

Audit trails can be reviewed manually or through automated means—either way, they must be reviewed and interpreted. If an organization reviews audit trails manually, it needs to establish a system of how, when, and why they are viewed. Usually audit logs

are very popular items right after a security breach, unexplained system action, or system disruption. An administrator or staff member rapidly tries to piece together the activities that led up to the event. This type of audit review is event-oriented. Audit trails can also be viewed periodically to watch for unusual behavior of users or systems, and to help understand the baseline and health of a system. Then there is a real-time, or near real-time, audit analysis that can use an automated tool to review audit information as it is created. Administrators should have a scheduled task of reviewing audit data. The audit material usually needs to be parsed and saved to another location for a certain time period. This information should be stated in the company's security policy and procedures.

Reviewing audit information manually can be overwhelming. There are applications and audit trail analysis tools that reduce the volume of audit logs to review and improve the efficiency of manual review procedures. A majority of the time, audit logs contain information that is unnecessary, so these tools parse out specific events and present them in a useful format. There are three main types of audit trail analysis tools: audit reduction, variance detection, and attack signature detection.

An ***audit-reduction tool*** does just what its name suggests—reduces the amount of information within an audit log. This tool discards mundane task information and records system performance, security, and user functionality information that can be useful to a security professional or administrator. A ***variance-detection*** tool can monitor computer and resource usage trends and detect variations. If an employee works from 8:00 A.M. to 5:00 P.M., which is the normal time his computer is used and resources are accessed, and recently this computer has been used at 2:00 A.M. and on the weekend, the variance-detection tool can capture this information and alert the administrator of unusual activity. If an ***attack signature–detection*** tool is used, the application will have a database of information that has been known to indicate specific attacks. This type of tool parses audit logs in search of certain patterns. If a pattern matches a pattern or signature held within its database, the tool indicates that an attack has taken place or is in progress.

Keystroke Monitoring

Oh, you typed an L; let me write that down. Oh, and a P, and a T, and an S…hey, slow down!

Keystroke monitoring is a type of auditing that can review and record keystrokes entered by a user during an active session. The person using this type of monitoring can have the characters written to an audit log to be reviewed at a later time. This type of auditing is usually done only for special cases and only for a specific amount of time, because the amount of information captured can be overwhelming. If a security professional or administrator is suspicious of an individual and his activities, she may invoke this type of monitoring. In some authorized investigative stages, a keyboard dongle may be unobtrusively inserted between the keyboard and the computer to capture all the keystrokes entered, including power-on passwords.

A hacker can also use this type of monitoring. If an attacker can successfully install a Trojan horse on a computer, the Trojan horse can install an application that captures

data as it is typed into the keyboard. Most of the time these programs are most interested in user credentials and can alert the attacker when credentials have been successfully captured.

There are privacy issues with this type of monitoring, and administrators could be subject to criminal and civil liabilities if it is done without proper notification to the employees and authorization from management. If a company wants to be able to use this type of auditing, it should state so in the security policy, address the issue in security-awareness training, and present a banner notice to the user warning that the activities at that computer may be monitored in this fashion. These steps should be taken to protect the company from violating an individual's privacy, and they should inform the users where their privacy boundaries start and stop pertaining to computer use.

Protecting Audit Data and Log Information

If an intruder breaks into your house, he will do his best to cover his tracks by not leaving fingerprints or any other clues that can be used to tie him to the criminal activity. The same is true in computer fraud and illegal activity. The intruder will work to cover his tracks. Attackers often delete audit logs that hold this incriminating information. (Deleting specific incriminating data within audit logs is called *scrubbing*.) Deleting this information can cause the administrator to not be alerted or aware of the security breach, and can destroy valuable data. Therefore, audit logs should be protected by strict access control.

Only certain individuals (the administrator and security personnel) should be able to view, modify, and delete audit trail information. No other individuals should be able to view this data, much less modify or delete it. The integrity of the data can be ensured with the use of digital signatures, message digest tools, and strong access controls. Its confidentiality can be protected with encryption and access controls, if necessary, and it can be stored on ***write-once media*** (CD-ROM) to prevent loss or modification of the data. Unauthorized access attempts to audit logs should be captured and reported.

Audit logs may be used in a trial to prove an individual's guilt, demonstrate how an attack was carried out, or corroborate a story. The integrity and confidentiality of these logs will be under scrutiny. Proper steps need to be taken to ensure that the confidentiality and integrity of the audit information is not compromised in any way.

References

- **Authentication, Authorization, and Accounting (AAA) Charter** www.ietf.org/html.charters/aaa-charter.html
- **Google authentication categories** http://directory.google.com/Top/Computers/Security/Authentication
- ***Security in Open Systems*, NIST Special Publication 800-7 (July 1994)** http://csrc.nist.gov/publications/nistpubs/800-7/node2.html

Access Control Practices

The fewest number of doors open allows the fewest number of flies in.

We have gone over how users are identified, authenticated, and authorized, and how their actions are audited. These are necessary parts of a healthy and safe network environment. You also want to take steps to ensure that there are no unnecessary open doors and that the environment stays at the same security level you have worked so hard to achieve. This means that you need to implement good access control practices. Not keeping up with daily or monthly tasks usually causes the most vulnerabilities in an environment. It is hard to put out all the network fires, fight the political battles, fulfill all the users' needs, and still keep up with small maintenance tasks. However, many companies have found that not doing these small tasks caused them the greatest heartache of all.

The following is a list of tasks that need to be accomplished on a regular basis to ensure that security stays at a satisfactory level:

- Deny access to systems by undefined users or anonymous accounts.
- Limit and monitor the usage of administrator and other powerful accounts.
- Suspend or delay access capability after a specific number of unsuccessful logon attempts.
- Remove obsolete user accounts as soon as the user leaves the company.
- Suspend inactive accounts after 30 to 60 days.
- Enforce strict access criteria.
- Enforce the need-to-know and least-privilege practices.
- Disable unneeded system features, services, and ports.
- Replace default password settings on accounts.
- Limit and monitor global access rules.
- Ensure that logon IDs are nondescriptive of job function.
- Remove redundant resource rules from accounts and group memberships.
- Remove redundant user IDs, accounts, and role-based accounts from resource access lists.
- Enforce password rotation.
- Enforce password requirements (length, contents, lifetime, distribution, storage, and transmission).
- Audit system and user events and actions and review reports periodically.
- Protect audit logs.

Even if all of these countermeasures are in place and properly monitored, there are ways that data can be lost in an unauthorized manner. The next section looks at these issues and their corresponding countermeasures.

Unauthorized Disclosure of Information

There are several ways that information can become available to others for whom it is not intended, which can bring about unfavorable results. Sometimes this is done intentionally, but it can be done unintentionally as well. Information can be disclosed unintentionally when one falls prey to attacks that specialize in causing this disclosure. These attacks are social engineering, covert channels, malicious code, and electrical airwave sniffing. Information can be disclosed accidentally through object reuse methods, which are explained next. (Social engineering was discussed in Chapter 3; covert channels are discussed in Chapter 5.)

Object Reuse

Can I borrow this floppy? Response: Let me destroy it first.

Object reuse issues pertain to reassigning to a subject media that previously contained one or more objects. Huh? This means before someone uses a hard drive, floppy disk, or tape, it should be cleared of any residual information that was on it previously. This concept also applies to objects that are reused by computer processes, such as memory locations, variables, and registers. The sensitive information that may be left by a process should be securely cleared before allowing another process the opportunity to access the object. This ensures that information not intended for this individual or any other subject is not disclosed. Many times, floppy disks are exchanged casually in a work environment. What if a supervisor lent a floppy to an employee without erasing it and it contained confidential employee performance reports and salary raises forecasted for the next year? This could prove to be a bad decision and maybe turn into a morale issue if the information was passed around. Formatting a disk or deleting files only removes the pointers to the files; it does not remove the actual files. This information will still be on the disk and available until the operating system needs that space and overwrites those files. So, for media that holds confidential information, more extreme methods should be taken to ensure that the files are actually gone, not just their pointers.

Sensitive data should be classified (secret, top secret, confidential, unclassified, and so on) by the data owners. How the data is stored and accessed should also be strictly controlled and audited by software controls. However, it does not end there; before allowing one subject to use media that was previously used, the media should be erased or degaussed. (This responsibility usually falls on the operations department.) If media holds sensitive information and cannot be purged, there should be steps on how to properly destroy it so that there is no way for others to obtain this information.

NOTE Sometimes hackers actually configure a sector on a hard drive so that it is marked as bad and unusable to an operating system, but this sector is actually fine and may hold malicious data. The operating system will not write information to this sector because it thinks it is corrupted. This is a form of *data hiding*. Some boot-sector virus routines are capable of putting the main part of their code (payload) into a specific sector of the hard drive, overwriting any data that may have been there, and then protecting it as a bad block.

Emanation Security

Quick, cover your computer and your head in tinfoil!

All electronic devices emit electrical signals. These signals can hold important information, and if an attacker buys the right equipment and positions himself in the right place, he could capture this information from the airwaves and access data transmissions as if he had a tap directly on the network wire.

There have been several incidents in which intruders have purchased inexpensive equipment and used it to intercept electrical emissions as they radiated from a computer. This equipment can reproduce data streams and display the data on the intruder's monitor, enabling the intruders to learn of covert operations, find out military strategies, and uncover and exploit confidential information. This is not just stuff found in spy novels; it really happens, so the proper countermeasures have been devised.

Tempest *Tempest* started out as a study carried out by the DoD and then turned into a standard that outlines how to develop countermeasures that control spurious electrical signals that are emitted by electrical equipment. There is special shielding that is used on equipment to suppress the signals as they are radiated from devices. Tempest equipment is implemented to prevent intruders from picking up information through the airwaves with listening devices. There are specific standards that this type of equipment must meet to be rated as providing Tempest shielding protection. Tempest refers to standardized technology that suppresses signal emanations with shielding material. Vendors who manufacture this type of equipment must be certified to this standard.

The devices (monitors, computers, printers, and so on) have an outer metal coating, referred to as a *Faraday cage.* This is made of metal with the necessary depth to ensure that only a certain amount of radiation is released. In devices that are Tempest rated, other components are also modified, especially the power supply, to help reduce the amount of electricity that is used.

There are allowable limits of emission levels that can radiate and still be considered safe. The approved products must ensure that only this level of emissions is allowed to escape the devices. This type of protection is usually needed only in military institutions, although other highly secured environments do utilize this type of safeguard.

Many military organizations are concerned with stray radio frequencies emitted by computers and other electronic equipment because an attacker may be able to pick them up, reconstruct them, and give away some secrets that were meant to stay secret.

Tempest technology is complex, cumbersome, and expensive, and therefore only used in highly sensitive areas that really need this high level of protection.

Two alternatives to Tempest exist: use white noise or use a control zone concept, both of which are explained next.

NOTE Tempest is the name of a program, and now a standard, that was developed in the late 1950s by the U.S. and British governments. It was developed to deal with electrical and electromagnetic radiation emitted from electrical equipment, mainly computers. This type of equipment is usually used by intelligence, military, government, and law enforcement agencies, and the selling of this type of equipment is under constant scrutiny.

White Noise A countermeasure used to combat intruders from extracting information from electrical transmissions is white noise. ***White noise*** is a uniform spectrum of random electrical signals. It is distributed over the full spectrum so that the bandwidth is constant and an intruder is not able to decipher real information from random noise or random information.

Control Zone Another alternative to using Tempest equipment is to use the zone concept, which was addressed earlier in this chapter. Some facilities use material in their walls to contain electrical signals. This prevents intruders from being able to access information that is emitted via electrical signals from network devices. This control zone creates a type of security perimeter and is constructed to protect against unauthorized access to data or compromise of sensitive information.

Access Control Monitoring

Access control monitoring is a method of keeping track of who attempts to access specific network resources. It is an important detective mechanism, and there are different technologies that can fulfill this need. It is not enough to invest in antivirus and firewall solutions. Companies are finding that monitoring their own internal network has become a way of life.

Intrusion Detection

Intrusion detection systems (IDSs) are different from traditional firewall products because they are designed to detect a security breach. *Intrusion detection* is the process of detecting an unauthorized use of, or attack upon, a computer, network, or telecommunications infrastructure. IDSs are designed to aid in mitigating the damage that can be caused by hacking, or breaking into sensitive computer and network systems. The basic intent of the IDS tool is to spot something suspicious happening on the network and sound an alarm by flashing a message on a network manager's screen, or possibly sending a page or even reconfiguring a firewall's ACL setting. The IDS tools can look for sequences of data bits that might indicate a questionable action or event, or monitor system log and activity recording files. The event does not need to be an intrusion to sound the alarm; any kind of "non-normal" behavior will do the trick.

Although there are different types of IDS products, they all have three common components: sensors, analyzers, and administrator interfaces. The sensors collect traffic and user activity data and send it to an analyzer, which looks for suspicious activity. If the analyzer detects an activity that it is programmed to deem as fishy, it sends an alert to the administrator's interface.

There are two main types of IDS: ***network-based***, which monitor network communications, and ***host-based***, which can analyze the activity within a particular computer system.

IDSs can be configured to watch for attacks, parse audit logs, terminate a connection, alert an administrator as attacks are happening, protect system files, expose a hacker's techniques, illustrate which vulnerabilities need to be addressed, and possibly help track down individual hackers.

Network-Based IDS

All the sailors love NIDS, because it is always in promiscuous mode.

A network-based IDS (NIDS) uses sensors, which are either host computers with the necessary software installed or dedicated appliances—each with its network interface card (NIC) in promiscuous mode. NICs watch for traffic that has the address of its host system, broadcasts, and sometimes multicast addresses. The NIC driver copies the data from the transmission medium and sends it up the network protocol stack for processing. When a NIC is put into promiscuous mode, the NIC driver captures all traffic and passes it to an analyzer to look for specific types of patterns.

A NIDS monitors network traffic and cannot "see" the activity going on inside a computer itself. To monitor the activities within a computer system, a company would need to implement a host-based IDS.

Host-Based IDS

A ***host-based IDS (HIDS)*** can be installed on individual workstations and/or servers and watch for inappropriate or anomalous activity. HIDSs are usually used to make sure users do not delete system files, reconfigure important settings, or put the system at risk in any other way. So, whereas the NIDS understands and monitors the network traffic, a HIDS's universe is limited to the computer itself. A HIDS does not understand or review network traffic, and a NIDS does not "look in" and monitor a system's activity. Each has its own job and stays out of the other's way.

In most environments, HIDS products are installed only on critical servers, not on every system on the network, because of the resource overhead and the administration nightmare that such an installation would cause.

Just to make life a little more confusing, HIDS and NIDS can be one of the following types:

- Signature based
- Statistical anomaly based

    - Protocol anomaly based
    - Traffic anomaly based
- Rule based
    - Stateful matching
    - Model based

Knowledge- or Signature-Based Intrusion Detection

Knowledge is accumulated by the IDS vendors about specific attacks and how they are carried out. Models of how the attacks are carried out are developed and called ***signatures***. Each identified attack has a signature, which is used to detect an attack in progress or determine if one has occurred within the network. Any action that is not recognized as an attack is considered acceptable.

An example of a signature is a packet that has the same source and destination IP address. All packets should have a different source and destination IP address, and if they have the same address, this means a Land attack is under way. In a Land attack, a hacker modifies the packet header so that when a receiving system responds to the packet, it is responding to its own address. Now that seems as though it should be benign enough, but vulnerable systems just do not have the programming code to know what to do in this situation, so they freeze or reboot. Once this type of attack was discovered, the signature-based IDS vendors wrote a signature that looks specifically for packets that contain the same source and destination address.

Signature-based IDS are the most popular IDS products today, and their effectiveness depends upon regularly updating the software with new signatures, as with antivirus software. This type of IDS is weak against new types of attacks because it can recognize only the ones that have been previously identified and have had signatures written for them. Attacks or viruses that have been discovered in production environments are referred to as being "in the wild." Attacks and viruses that exist but have not been released are referred to as being "in the zoo." No joke.

Statistical Anomaly–Based Intrusion Detection

Through statistical analysis I have determined that I am an anomaly in nature. Response: You have my vote.

Statistical anomaly–based IDS is a behavioral-based system. Behavioral-based IDS products do not use predefined signatures, but rather are put in a learning mode to build a profile of an environment's "normal" activities. This profile is built by continually sampling the environment's activities. The longer the IDS is put in a learning mode, in most instances, the more accurate a profile it will build and the better protection it will provide. After this profile is built, all future traffic and activities are compared to it. The same type of sampling that was used to build the profile takes place, so the same type of data is being compared. Anything that does not match the profile is seen as an attack, in response to which the IDS sends an alert. With the use of complex statistical

algorithms, the IDS looks for anomalies in the network traffic or user activity. Each packet is given an anomaly score, which indicates its degree of irregularity. If the score is higher than the established threshold of "normal" behavior, then the preconfigured action will take place.

The benefit of using a statistical anomaly–based IDS is that it can react to new attacks. It can detect "0 day" attacks, which means that an attack is new to the world and no signature or fix has been developed yet. These products are also capable of detecting the "low and slow" attacks, in which the attacker is trying to stay beneath the radar by sending a few packets at a time over a longer period of time. The IDS should be able to detect these types of attacks because they are different enough from the contrasted profile.

Now for the bad news. Since the only thing that is "normal" about a network is that it is constantly changing, developing the correct profile that will not provide an overwhelming number of false positives can be difficult. Many IT staff members know all too well this dance of chasing down alerts that end up being benign traffic or activity. In fact, some environments end up turning off their IDS because of the amount of time these activities take up. (Proper education on tuning and configuration will reduce the number of false positives.)

If an attacker detects that there is an IDS on a network, she will then try to detect the type of IDS it is so that she can properly circumvent it. With a behavioral-based IDS, the attacker could attempt to integrate her activities into the behavior pattern of the network traffic. That way, her activities are seen as "normal" by the IDS and thus go undetected. It is a good idea to ensure that there is no current attack activity that is under way while the IDS is in learning mode. If this takes place, the IDS will never alert you of this type of attack in the future because it sees this traffic as typical of the environment.

If a corporation decides to use a statistical anomaly–based IDS, it needs to ensure that the staff members who are implementing and maintaining it understand protocols and packet analysis. Because this type of an IDS sends generic alerts, compared to other types of IDSs, it is up to the network engineer to figure out what the actual issue

Attack Techniques

It is common for hackers to first identify whether an IDS is present on the network that they are preparing to attack. If one is present, that attacker may first attack the IDS to bring it off line by a denial-of-service. Another tactic is to send the IDS incorrect data, which will make the IDS send off specific alerts indicating that a certain attack is under way, when in truth it is not. The goal of these activities is either to disable the IDS or to distract the network and security individuals so that they will be busy chasing the wrong packets, while the real attack takes place.

> **What's in a Name?**
> Signature-based IDSs are also known as misuse-detection systems, and behavioral-based IDSs are also known as profile-based systems.

is. For example, a signature-based IDS reports the type of attack that has been identified, and a rule-based IDS identifies the actual rule that the packet does not comply with. In a statistical anomaly–based IDS, all the product really understands is that something "abnormal" has happened, which just means that the event does not match the profile.

Determining the proper thresholds for statistically significant deviations is really the key for the successful use of a behavioral-based IDS. If the threshold is set too low, nonintrusive activities are considered attacks (false positives). If the threshold is set too high, then malicious activities are not identified (false negatives).

Once an IDS discovers an attack, several things can take place, depending upon the capabilities of the IDS and the policy that has been assigned to it. The IDS can send an alert to a console to tell the right individuals that an attack is being carried out; send an e-mail or page to the individual who has been delegated to respond to such activities; kill the connection of the detected attack; or reconfigure a router or firewall to try to stop any further similar attacks. A modifiable response condition might include anything from blocking a specific IP address to redirecting or blocking a certain type of activity.

Protocol Anomaly–Based IDS A statistical anomaly–based IDS can use protocol anomaly–based filters. These types of IDSs have specific knowledge of each protocol that they will be monitoring. A protocol anomaly pertains to the format and behavior of a protocol. The IDS builds a model (or profile) of each protocol's "normal" usage. Keep in mind, however, that protocols have *theoretical* usage, as outlined in their corresponding RFCs, and *real-world* usage, which refers to the fact that vendors seem to always "color outside the lines" and not strictly follow the RFCs in their protocol development and implementation. So, most profiles of individual protocols are a mix between the official and real-world versions of the protocol and its usage. When the IDS is activated, it looks for anomalies that do not match the profiles built for the individual protocols.

Although there are several vulnerabilities within operating systems and applications that are available to be exploited, many more successful attacks take place by exploiting vulnerabilities in the protocols themselves. At the OSI data link layer, the Address Resolution Protocol (ARP) does not have any protection against ARP attacks where bogus data is inserted into its table. At the network layer, the Internet Control Message Protocol (ICMP) can be used in a Loki attack to move data from one place to another, when this protocol was designed to only be used to send status information—

not user data. IP headers can be easily modified for spoofed attacks. At the transport layer, TCP packets can be injected into the connection between two systems for a session hijacking attack.

NOTE When an attacker compromises a computer and loads a backdoor on the system, he will need to have a way to communicate to this computer through this backdoor and stay "under the radar" of the network firewall and IDS. Hackers have figured out that a small amount of code can be inserted into an ICMP packet, which is then interpreted by the backdoor software loaded on a compromised system. Security devices are usually not configured to monitor this type of traffic because ICMP is a protocol that is supposed to be used just to send status information—not commands to a compromised system.

Because every packet formation and delivery involves many protocols, and because more attack vectors exist in the protocols than in the software itself, it is a good idea to integrate protocol-anomaly filters in any network behavioral-based IDS.

Traffic Anomaly–Based IDS Most behavioral-based IDSs have traffic-anomaly filters, which detect changes in traffic patterns as in DoS attacks or a new service that appears on the network. Once there is a profile that is built that captures the baselines of an environment's ordinary traffic, all future traffic patterns are compared to that profile. As with all filters, the thresholds are tunable to adjust the sensitivity, to reduce the number of false positives and false negatives. Since this is a type of statistical anomaly–based IDS, it can detect unknown attacks.

Rule-Based Intrusion Detection

A rule-based IDS takes a different approach than a signature-based or statistical anomaly–based system. A signature-based IDS is very straightforward. For example, if a signature-based IDS detects a packet that has all of its TCP header flags with the bit value of 1, it knows that an xmas attack is under way—so it sends an alert. A statistical anomaly–based IDS is also straightforward. For example, if Bob has logged onto his computer at 6 A.M. and the profile indicates that this is abnormal, the IDS sends an alert, because this is seen as an activity that needs to be investigated. Rule-based intrusion detection gets a little bit more tricky, depending upon the complexity of the rules that are used.

Rule-based intrusion detection is commonly associated with the use of an expert system. An expert system is made up of a knowledge base, inference engine, and rule-based programming. Knowledge is represented as rules, and the data that is to be analyzed is referred to as facts. The knowledge of the system is written in rule-based programming (IF *situation* THEN *action*). These rules are applied to the facts, the data that comes in from a sensor, or a system that is being monitored. For example, in scenario 1 the IDS pulls data from a system's audit log and stores it temporarily in its fact database, as illustrated in Figure 4-18. Then, the preconfigured rules are applied to

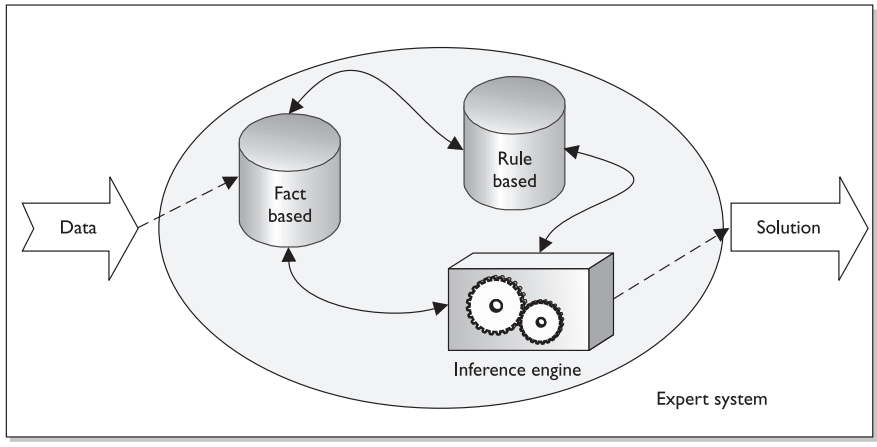


Figure 4-18 Rule-based IDS and expert system components

this data to indicate whether anything suspicious is taking place. In our scenario, the rule states "IF *a root user creates File1* AND *creates File2* SUCH THAT *they are in the same directory* THEN *there is a call to AdministrativeTool1* TRIGGER *send alert.*" This rule has been defined such that if a root user creates two files in the same directory and then makes a call to a specific administrative tool, an alert should be sent.

It is the inference engine that provides some artificial intelligence into this process. An inference engine can infer new information from provided data by using inference rules. To understand what inferring means in the first place, let's look at the following;

Socrates is a man.
All men are mortals.

Then we can infer that Socrates is mortal. If you are asking "What does this have to do with a hill of beans?" just hold on to your hat—here we go.

Regular programming languages deal with the "black and white" of life. The answer is either yes or no, not maybe this or maybe that. Although computers can carry out complex computations at a much faster rate than humans, they have a harder time guessing, or inferring, answers because they are very structured. The fifth-generation programming languages (artificial intelligence languages) are capable of dealing with the grayer areas of life and can attempt to infer the right solution from the provided data.

So, in a rule-based IDS that is based on an expert system, the IDS gathers data from a sensor or log, and the inference engine uses its preprogrammed rules on it; if the characteristics of the rules are met, then an alert or solution is provided, as illustrated in Figure 4-18.

There are two basic types of rule-based IDSs that you need to understand: state-based IDSs and model-based IDSs.

State-Based IDS Before you can get too deep into understanding how a state-based IDS works, you need to understand what the state of a system or application actually is. Every change that an operating system experiences (user logs on, user opens application, application communicates to another application, user inputs data, and so on) is considered a state transition. In a very technical sense, all operating systems and applications are just lines and lines of instructions that are written to carry out functions on data. The instructions have empty variables, which is where the data is held. So when you use the calculator program and you type in 5, there is an empty variable that has just been populated with this value. By entering that value, you change the state of that application. When applications communicate with each other, they populate empty variables that are provided in each application's instruction set. So, a state transition is when a variable's value changes, which usually happens continuously within every system.

There are specific state changes (activities) that take place for specific types of attacks. If an attacker is going to carry out a remote buffer overflow, then the following state changes will take place:

1. The remote user connects to the system.
2. The remote user sends data to an application (the data exceeds the allocated buffer for this empty variable).
3. The data is executed and overwrites the buffer and possibly other memory segments.
4. Malicious code executes.

So, *state* is a snapshot of an operating system's values in volatile, semipermanent, and permanent memory locations. In a state-based IDS, the initial state is the state prior to the execution of an attack, and the compromised state is the state after successful penetration. The IDS has rules that outline what state transition sequences should sound an alarm. The activity that takes place between the initial and compromised state is what the state-based IDS looks for, and it sends an alert if any of the state-transition sequences match its preconfigured rules.

This type of IDS scans for attack signatures in the context of a stream of activity instead of just looking at individual packets. It can only identify known attacks and requires frequent updates of its signatures.

Model-Based IDS In a model-based IDS, the product has several scenario models that represent how specific attacks and intrusions take place. The models outline how the system would behave if it were under attack, the different steps that would be carried out by the attacker, and the evidence that would be available for analysis if specific intrusions took place.

The IDS takes in the audit log data and compares it to the different models that have been developed, to see if the data meets any of the models' specifications. If the IDS finds data in an audit log that matches the characteristics in a specific model, it sends an alert.

As analogy, suppose that we have a model that states the following: if it looks like a duck, sounds like a duck, and walks like a duck—it's a duck. So, we do our in-depth analysis upon the creature in front of us. We compare each characteristic in the model and find that each matches, so we conclude that the creature is a duck.

IDS Types

It is important to understand the characteristics that make the different types of IDS technologies distinct. Here is a summary:

- **Signature based**
  - Pattern matching, similar to antivirus software
  - Signatures must be continuously updated
  - Cannot identify new attacks
- **Statistical anomaly based**
  - Behavioral-based system that learns the "normal" activities of an environment
  - Can detect new attacks
  - Two types:
    Protocol anomaly based Unusual format or behavior of protocols
    Traffic anomaly based Unusual format of traffic patterns
- **Rule based**
  - Use of IF/THEN rule-based programming within expert systems
  - Use of expert system allows for artificial intelligence characteristics
  - The more complex the rules, the more demands on software and hardware processing requirements
  - Cannot detect new attacks
  - Two types:
    Stateful matching Tracking system state changes that indicate an attack is under way
    Model based Models of attack scenarios are built and then captured data is compared to the models to uncover malicious activities

References

- **"The Science of IDS Attack Identification" (Cisco Systems white paper)** www.cisco.com/en/US/products/sw/secursw/ps976/products_white_paper09186a0080092334.shtml
- **"Intrusion Detection Terminology (Part Two)," by Andy Cuff (SecurityFocus Infocus Archive, last updated Sept. 24, 2003)** www.securityfocus.com/infocus/1733
- **Honeypots.net IDS Software links page** www.honeypots.net/ids/products
- **"State of the Practice of Intrusion Detection Technologies," by Julia Allen, et al., Software Engineering Institute, Carnegie Mellon University (Jan. 1999)** www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html

IDS Sensors

Network-based IDSs use sensors for monitoring purposes. A sensor, which works as an analysis engine, is placed on the network segment the IDS is responsible for monitoring. The sensor receives raw data from an event generator, as shown in Figure 4-19, and compares it to a signature database, profile, or model, depending upon the type of IDS. If there is some type of a match, which indicates suspicious activity, the sensor works with the response module to determine what type of activity needs to take place (alerting through instant messaging, page, e-mail, or carry out firewall reconfiguration, and so on). The sensor's role is to filter received data, discard irrelevant information, and detect suspicious activity.

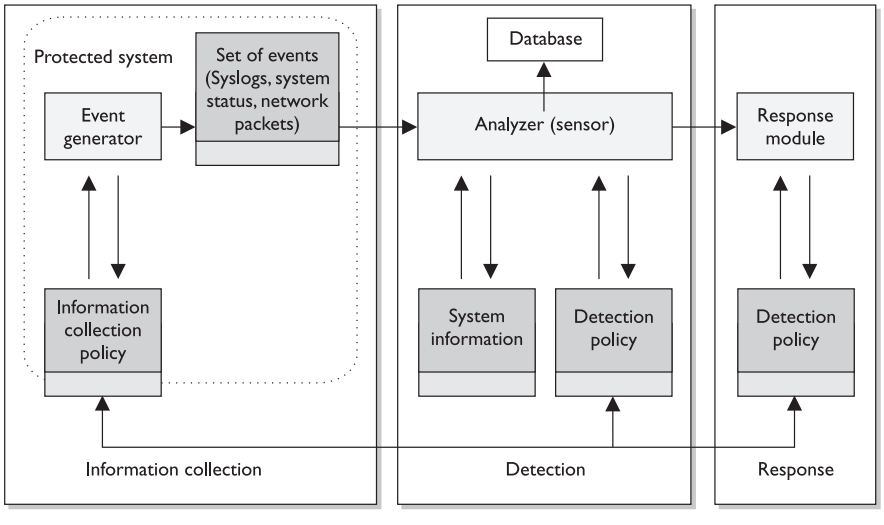


Figure 4-19 Basic architecture of an NDIS

Switched Environments

NIDSs have a harder time working on a switched network, when compared to traditional nonswitched environments, because data is transferred through independent virtual circuits and not broadcasted, as in nonswitched environments. The IDS sensor acts as a sniffer and does not have access to all the traffic in these individual circuits. So, we have to take all the data on each individual virtual private connection, make a copy of it, and put the copies of the data on one port (spanning port) where the sensor is located. This allows the sensor to have access to all the data going back and forth on a switched network.

A monitoring console monitors all sensors and supplies the network staff with an overview of the activities of all the sensors in the network. A difficulty arises in a switched environment, where traffic is forwarded through a virtual private connection and is not rebroadcast to all the ports. The switch should have a management port, enabling all traffic on that switch to be mirrored to one port, where the sensor is placed. (This is also referred to as a *spanning port,* where all traffic from all ports can be mirrored to one port.)

These are the components that enable network-based intrusion detection to actually work. Sensor placement is a critical part of configuring an effective IDS. An organization can place a sensor outside of the firewall to detect attacks and place a sensor inside the firewall (in the perimeter network) to detect actual intrusions. Sensors should also be placed in highly sensitive areas, DMZs, and on extranets. Figure 4-20 shows the sensors reporting their findings to the central console.

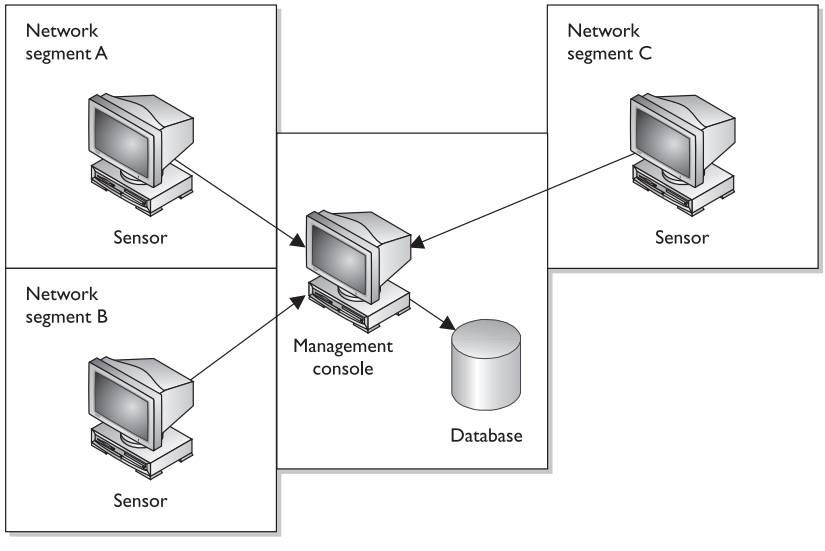


Figure 4-20 Sensors need to be placed in each network segment that is to be monitored by the IDS.

The IDS can be centralized, as firewall products that have IDS functionality integrated within them, or distributed, with multiple sensors throughout the network.

Network Traffic

If the network traffic volume exceeds the IDS system's threshold, attacks may go unnoticed. Each vendor's IDS product has its own threshold, and you should know and understand that threshold before it you purchase and implement the IDS.

In very high-traffic environments, multiple sensors should be in place to ensure that all packets are investigated. If necessary to optimize network bandwidth and speed, different sensors can be set up to analyze each packet for different signatures. That way, the analysis load can be broken up over different points.

Intrusion Prevention Systems

An ounce of prevention does something good. Response: Yeah, causes a single point of failure.

In the industry, there is constant frustration with the inability of existing products to stop the bad guys from accessing and manipulating corporate assets. This has created a market demand for vendors to get creative and come up with new, innovative technologies and new products for companies to purchase, implement, and still be frustrated with.

The next "big thing" in the IDS arena is the intrusion prevention system (IPS). The traditional IDS only detects that something bad may be taking place and sends an alert. The goal of an IPS is to detect this activity and not allow the traffic to gain access to the target in the first place, as shown in Figure 4-21. So, an IPS is a preventative and proactive technology, whereas an IDS is a detective and after-the-fact technology.

As of this writing, the industry is debating what the real definition of IPS is, and many vendors are attempting to be the king of the hill in this market. But the general concept of the definition is agreed upon, which is a focus on prevention rather than detection. The goal is to combine into one product the "stop the packets in their tracks" functionality that firewalls provide with the in-depth packet analysis that an IDS provides. As within the IDS arena, there are host-based IPS products and network-based IPS products. A majority of the network-based IPS products are inline devices, which means that all traffic must go through and be monitored by the IPS before going any

Intrusion Responses

Most IDSs are capable of several types of response to a triggered event. An IDS can send out a special signal to drop or kill the packet connections, at both the source and destinations. This effectively disconnects the communication, and does not allow it to be transmitted. An IDS might block a user from accessing a resource on a host system, if the threshold is set to trigger this response. An IDS can send alerts of an event trigger to other hosts, IDS monitors, and administrators. And, finally, some IDSs can reconfigure themselves to perform some predefined action.

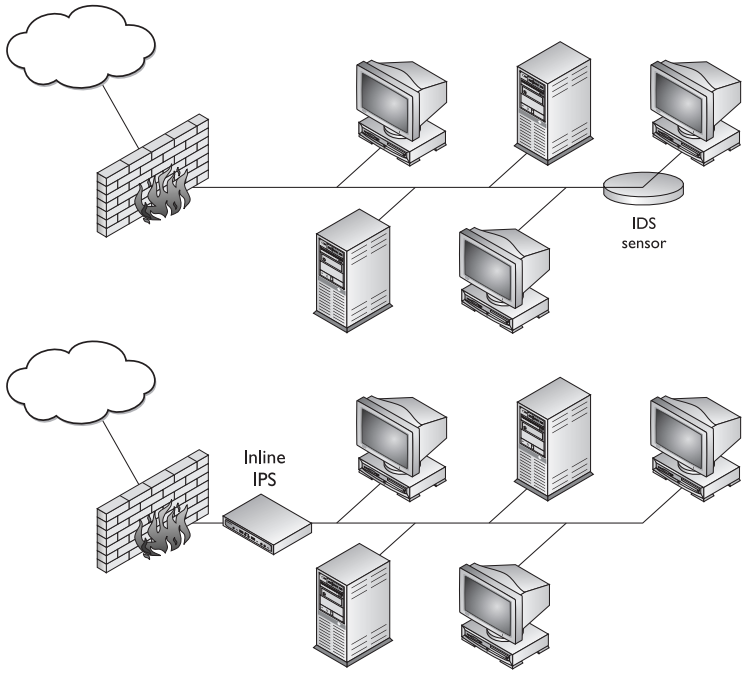


Figure 4-21 IDS vs. IPS architecture

further down its path. This is a concern for many companies because any inline device can be a traffic bottleneck, reduce performance, and pose a single point of failure. (Vendors are working hard to provide technologies to deal with each of these issues.)

Most NIPSs have two NICs, external and internal facing. Traffic arrives on the external NIC and is analyzed by the engine. If the packets are deemed safe, they go on their way out the internal-facing NIC. If the packets are deemed malicious, then the packet is discarded.

Only time will tell whether IPS products will replace IDS products. Some people feel as though it is just a new term for marketing purposes, whereas others feel as though it is the next step in our evolution of information and computer security technology advances. The rest of us should just get some popcorn and watch how this play unfolds.

References

- **"Intrusion Prevention Versus Intrusion Detection," by Markus DeShon (SecureWorks white paper)** www.secureworks.com/en/html/printer/internet/techResourceCenter/Intrusion-prevention-versus-intrusion-detection.html

- **"Intrusion Prevention Systems: The Next Step in the Evolution of IDS," by Neil Desai (SecurityFocus Infocus Archive, last updated Feb. 27, 2003)** www.securityfocus.com/infocus/1670
- **"Intrusion Prevention Systems (IPS), Part 1: Deciphering the inline Intrusion Prevention hype, and working toward a real-world, proactive security solution" (Secure Computing white paper, 2003)** www.securecomputing.com/pdf/Intru-Preven-WP1-Aug03-vF.pdf
- **"Rule-Based Intrusion Detection," from the Software Engineering Institute (SEI) Software Technology Roadmap** www.sei.cmu.edu/str/descriptions/rbid.html

Honeypot

Hey, curious, ill-willed, and destructive attackers, look at this shiny new vulnerable computer.

A ***honeypot*** is a computer set up as a sacrificial lamb on the network. The system is not locked down and has open ports and services enabled. This is to entice a would-be attacker to this computer instead of attacking authentic production systems on a network. The honeypot contains no real company information, and thus will not be at risk if and when it is attacked.

This enables the administrator to know when certain types of attacks are happening so that he can fortify the environment and perhaps track down the attacker. The longer the hacker stays at the honeypot, the more information that will be disclosed about her techniques.

It is important to draw a line between ***enticement*** and ***entrapment*** when implementing a honeypot system. There are legal and liability issues around each. If the system only has open ports and services that an attacker might want to take advantage of, this would be an example of enticement. If the system has a web page indicating that the user can download files, and once the user does this the administrator charges this user with trespassing, it would be entrapment. Entrapment is where the intruder is induced or tricked into commiting a crime. Entrapment is illegal and cannot be used when charging an individual with hacking or unauthorized activity.

References

- **The Honeynet Project** http://project.honeynet.org
- **"What Is a Honeypot?" by Loras Even, SANS (July 12, 2000)** www.sans.org/resources/idfaq/honeypot3.php

Network Sniffers

I think I smell a packet! Response: Nope. It's my feet.

A packet or network ***sniffer*** is a general term for programs or devices that are able to examine traffic on a LAN segment. Traffic that is being transferred over a network medium is transmitted as electrical signals, encoded in binary representation. The sniffer has to have a protocol-analysis capability to recognize the different protocol values to properly interpret their meaning.

The sniffer has to have access to a network adapter that works in promiscuous mode and a driver that captures the data. This data can be overwhelming, so it must be properly filtered. The filtered data is stored in a buffer, and this information is displayed to a user and/or captured in logs. Some utilities have sniffer and packet-modification capabilities, which is how some types of spoofing and man-in-the-middle attacks are carried out.

Network sniffers are used by the people in the white hats (administrators and security professionals) usually to try to track down a recent problem with the network. But the guys in the black hats (attackers and crackers) can use them to learn about what type of data is passed over a specific network segment and to modify data in an unauthorized manner. Black hats usually use sniffers to obtain credentials as they pass over the network medium.

NOTE Sniffers are dangerous because they are very hard to detect and their activities are difficult to audit.

A Few Threats to Access Control

As a majority of security professionals know, there is more risk and a higher probability of an attacker causing mayhem from within an organization than from outside the organization. However, many people within organizations do not know this fact, because they only hear stories about the outside attackers who defaced a web server or circumvented a firewall to access confidential information.

An attacker from the outside can enter through remote dial-in entry points, enter through firewalls and web servers, physically break in, or exploit a partner communication path (extranet, vendor connection, and so on). An insider has legitimate reasons for using the systems and resources, but can misuse his privileges and launch an actual attack also. The danger of insiders is that they have already been given a wide range of access that a hacker would have to work to obtain, they probably have intimate knowledge of the environment, and generally they are trusted. We have discussed many different types of access control mechanisms that work to keep the outsiders outside and restrict the insiders' abilities to a minimum and audit their actions. Now we will look at some specific attacks that are commonly carried out in environments today by insiders or outsiders.

Dictionary Attack

There are several programs that enable an attacker (or proactive administrator) to identify user credentials. This type of program is fed lists (dictionaries) of commonly used words or combinations of characters, and then compares these values to capture passwords. In other words, the program hashes the dictionary words and compares the resulting message digest with the system password file that also stores its passwords in

one-way hashed format. If the hashed values match, it means that a password has just been uncovered. Once the right combination of characters is identified, the attacker can use this password to authenticate herself as a legitimate user. Because many systems have a threshold that dictates how many failed logon attempts are acceptable, the same type of activity can happen to a captured password file. The dictionary-attack program encrypts the combination of characters and compares it to the encrypted entries in the password file. If a match is found, the program has uncovered a password.

The dictionaries come with the password cracking programs, and extra dictionaries can be found on several sites on the Internet.

NOTE Passwords should never be transmitted or stored in cleartext. Most operating systems and applications put the passwords through hashing algorithms, which result in hash values, also referred to as message digest values.

Countermeasures

To properly protect an environment against dictionary and other password attacks, the following practices should be followed:

- Do not allow passwords to be sent in cleartext.
- Encrypt the passwords with encryption algorithms or hashing functions.
- Employ one-time password tokens.
- Use hard-to-guess passwords.
- Rotate passwords frequently.
- Employ an IDS to detect suspicious behavior.
- Use dictionary cracking tools to find weak passwords chosen by users.
- Use special characters, numbers, and upper- and lowercase letters within the password.
- Protect password files.

Brute Force Attack

I will try over and over until you are defeated. Response: Okay, wake me when you are done.

There are several types of ***brute force attacks***, but overall they are attacks that continually try different inputs to achieve a predefined goal. Brute force is defined as "trying every possible combination until the correct one is identified." So in a brute force password attack, the software tool will see if the first letter is an "a" and continue through the alphabet until that single value is uncovered. Then the tool moves on to the second value and so on.

The most effective way to uncover passwords is through a hybrid attack, which combines a dictionary attack and a brute force attack. If a dictionary tool has found that a user's password starts with Dallas, then the brute force tool will try Dallas1, Dallas01,

Dallasa1, and so on until a successful logon credential is uncovered. (A brute force attack is also known as an exhaustive attack.)

These attacks are also used in ***wardialing*** efforts, in which the wardialer inserts a long list of phone numbers into a wardialing program in hopes of finding a modem that can be exploited to gain unauthorized access. A program is used to dial many phone numbers and weed out the numbers that are used for voice calls and fax machine services. The attacker usually ends up with a handful of numbers he can now try to exploit to gain access into a system or network.

So, a brute force attack perpetuates a specific activity with different input parameters until the goal is achieved.

Countermeasures

For phone brute force attacks, auditing and monitoring of this type of activity should be in place to uncover patterns that could indicate a wardialing attack:

- Perform brute force attacks to find weaknesses and hanging modems.
- Make sure only necessary phone numbers are made public.
- Provide stringent access control methods that would make brute force attacks less successful.
- Monitor and audit for such activity.
- Employ an IDS to watch for suspicious activity.
- Set lockout thresholds.

Spoofing at Logon

So, what are your credentials again?

An attacker can use a program that presents to the user a fake logon screen, which often tricks the user into attempting to log on. The user is asked for a username and password, which are stored for the attacker to access at a later time. The user does not know this is not his usual logon screen because they look exactly the same. A fake error message can appear, indicating that the user mistyped his credentials. At this point, the fake logon program exits and hands control over to the operating system, which

Phishing Attacks

Phishing attacks usually take place through e-mail schemes that request users to disclose personal or financial information. The e-mail usually appears to come from a legitimate corporation (financial institutions, PayPal, eBay, credit card companies, service providers, and so on) asking the person to update their account information or threatening to terminate accounts. The e-mail contains a link that takes the user to a web site that looks like the official site. It is at this site where the user is asked to enter their credentials, account numbers, and possibly credit card information.

prompts the user for a username and password. The user assumes he mistyped his information and doesn't give it a second thought, but an attacker now knows the user's credentials.

This has become a common attack on the Internet in phishing attacks and identity theft attempts. Hackers have created web sites that look just like www.amazon.com, www.ebay.com, and many of the other popular sites people use today in e-commerce transactions.

Countermeasures

An operating system can be configured to display the number of failed logon attempts. If the first logon attempt seemed to have failed, but was really the attacker's program capturing the entered credentials, and it was not reported at the second attempt and the user could get suspicious as to what just took place.

A guaranteed ***trusted path*** can be provided by the operating system. A trusted path is a communication link between the user and the kernel that cannot be circumvented as described in the scenario of a fake logon screen. Windows 2000 uses a sequence of CTRL-ALT-DEL to invoke the operating system's logon screen. (However, some sneaky fake programs can set themselves to be called upon by this combination of keys also.)

For countermeasures to phishing attacks, the following are common best practices:

- Be skeptical of e-mails indicating that you need to make changes to your accounts or warnings indicating that accounts will be terminated without you doing some type of activity online.
- Call the legitimate company to find out if this is a fraudulent message.
- Review the address bar to see if the domain name is correct.
- When submitting any type of financial information or credential data, an SSL connection should be set up, which is indicated in the address bar (https://) and a closed-padlock icon in the browser at the bottom-right corner.
- Do not click on an HTML link within an e-mail. Type the URL out manually instead.
- Do not accept e-mail in HTML format.

Summary

Access controls are security features that are usually considered the first line of defense in asset protection. They are used to dictate how subjects access objects, and their main goal is to protect the objects from unauthorized access. These controls can be administrative, physical, or technical in nature and can supply preventive, detective, deterrent, recovery, compensative, and corrective services.

Access control defines how users should be identified, authenticated, and authorized. These issues are carried out differently in different access control models and technologies, and it is up to the organization to determine which best fits its business and security needs.

Quick Tips

- Access is a flow of information between a subject and an object.
- A subject is an active entity that requests access to an object, which is a passive entity.
- A subject can be a user, program, or process.
- Confidentiality is the assurance that information is not disclosed to unauthorized subjects.
- Some security mechanisms that provide confidentiality are encryption, logical and physical access control, transmission protocols, database views, and controlled traffic flow.
- There are three main access control models: discretionary, mandatory, and nondiscretionary.
- Discretionary access control (DAC) enables data owners to dictate what subjects have access to the files and resources they own.
- Mandatory access control (MAC) uses a security label system. Users have clearances, and resources have security labels that contain data classifications. MAC compares these two attributes to determine access control capabilities.
- Nondiscretionary access control uses a role-based method to determine access rights and permissions.
- Role-based access control is based on the user's role and responsibilities within the company.
- There are three main types of restricted interface measurements: menus and shells, database views, and physically constrained interfaces.
- Access control lists are bound to objects and indicate what subjects can use them.
- A capability table is bound to a subject and lists what objects it can access.
- There are two main ways of administrating access control: centralized and decentralized.
- Some examples of centralized administration technologies are RADIUS, TACACS+, and Diameter.
- A decentralized administration example is a peer-to-peer working group.
- Examples of administrative controls are a security policy, personnel controls, supervisory structure, security-awareness training, and testing.
- Examples of physical controls are network segregation, perimeter security, computer controls, work area separation, data backups, and cable.
- Examples of technical controls are system access, network architecture, network access, encryption and protocols, and auditing.

- Access control mechanisms provide one or more of the following functionalities: preventive, detective, corrective, deterrent, recovery, or compensative.
- For a subject to be able to access a resource, it must be identified, authenticated, authorized, and should be held accountable for its actions.
- Authentication can be accomplished by biometrics, a password, a passphrase, a cognitive password, a one-time password, or a token.
- A Type I error in biometrics means the system rejected an authorized individual, and a Type II error means an imposter was authenticated.
- A memory card cannot process information, but a smart card can.
- Access controls should default to no access.
- Least-privilege and need-to-know principles limit users' rights to only what is needed to perform tasks of their job.
- Single sign-on requires a user to be authenticated to the network only one time.
- Single sign-on can be accomplished through Kerberos, SESAME, domains, and thin clients.
- In Kerberos, a user receives a ticket from the KDC to be able to authenticate to a service.
- The Kerberos user receives a ticket granting ticket (TGT), which allows him to request access to resources through the ticket granting service (TGS). The TGS generates a new ticket with the session keys.
- Types of access control attacks include denial of service, spoofing, dictionary, brute force, and wardialing.
- Audit logs can track user activities, application events, and system events.
- Keystroke monitoring is a type of auditing that tracks each keystroke made by a user.
- Audit logs should be protected and reviewed.
- Object reuse can unintentionally disclose information.
- Just removing pointers to files is not always enough protection for proper object reuse.
- Information can be obtained via electrical signals in airwaves. The ways to combat this type of intrusion are Tempest, white noise, and control zones.
- User authentication is accomplished by what someone knows, is, or has.
- One-time password-generating token devices can use synchronous or asynchronous methods.
- Strong authentication requires two of the three user authentication attributes (what someone knows, is, or has).
- Kerberos addresses privacy and integrity but not availability.

- The following are weaknesses of Kerberos: the KDC is a single point of failure; it is susceptible to password guessing; session and secret keys are locally stored; KDC needs to always be available; and there must be management of secret keys.
- IDSs can be statistical (monitors behavior) or signature based (watches for known attacks).
- Degaussing is a safeguard against disclosure of confidential information because it returns media back to its original state.
- Two types of statistical anomaly–based IDS are protocol anomaly based and traffic anomaly based.
- Two types of rule-based IDS are stateful and model-based IDS.

Questions

Please remember that these questions are formatted and asked in a certain way for a reason. You must remember that the CISSP exam is asking questions at a conceptual level. Questions may not always have the perfect answer, and the candidate is advised against always looking for the perfect answer. The candidate should look for the best answer in the list.

1. Which of the following statements correctly describes biometric methods?
   - **A.** They are the least expensive and provide the most protection.
   - **B.** They are the most expensive and provide the least protection.
   - **C.** They are the least expensive and provide the least protection.
   - **D.** They are the most expensive and provide the most protection.
2. What is derived from a passphrase?
   - **A.** Personal password
   - **B.** Virtual password
   - **C.** User ID
   - **D.** Valid password
3. Which of the following statements correctly describes passwords?
   - **A.** They are the least expensive and most secure.
   - **B.** They are the most expensive and least secure.
   - **C.** They are the least expensive and least secure.
   - **D.** They are the most expensive and most secure.
4. What is the reason for enforcing the separation of duties?
   - **A.** No one person can complete all the steps of a critical activity.
   - **B.** It induces an atmosphere for collusion.

C. It increases dependence on individuals.

D. It makes critical tasks easier to accomplish.

5. Which of the following is not a logical access control?

A. Encryption

B. Network architecture

C. ID badge

D. Access control matrix

6. An access control model should be applied in a _____ manner.

A. Detective

B. Recovery

C. Corrective

D. Preventive

7. Which access control policy is enforced when an environment uses a nondiscretionary model?

A. Rule based

B. Role based

C. Identity based

D. Mandatory

8. How is a challenge/response protocol utilized with token device implementations?

A. This protocol is not used; cryptography is used.

B. An authentication service generates a challenge, and the smart token generates a response based on the challenge.

C. The token challenges the user for a username and password.

D. The token challenges the user's password against a database of stored credentials.

9. Which access control method is user-directed?

A. Nondiscretionary

B. Mandatory

C. Identity based

D. Discretionary

10. Which provides the best authentication?

A. What a person knows

B. What a person is

C. What a person has

D. What a person has and knows

11. Which item is not part of a Kerberos authentication implementation?
    - **A.** Message authentication code
    - **B.** Ticket granting service
    - **C.** Authentication service
    - **D.** Users, programs, and services
12. Which model implements access control matrices to control how subjects interact with objects?
    - **A.** Mandatory
    - **B.** Centralized
    - **C.** Decentralized
    - **D.** Discretionary
13. What does authentication mean?
    - **A.** Registering a user
    - **B.** Identifying a user
    - **C.** Validating a user
    - **D.** Authorizing a user
14. If a company has a high turnover rate, which access control structure is best?
    - **A.** Role based
    - **B.** Decentralized
    - **C.** Rule based
    - **D.** Discretionary
15. A password is mainly used for what function?
    - **A.** Identity
    - **B.** Registration
    - **C.** Authentication
    - **D.** Authorization
16. The process of mutual authentication involves _____.
    - **A.** A user authenticating to a system and the system authenticating to the user
    - **B.** A user authenticating to two systems at the same time
    - **C.** A user authenticating to a server and then to a process
    - **D.** A user authenticating, receiving a ticket, and then authenticating to a service
17. Reviewing audit logs is an example of which security function?
    - **A.** Preventive
    - **B.** Detective

C. Deterrence

D. Corrective

18. In discretionary access control security, who has delegation authority to grant access to data?

A. User

B. Security office

C. Security policy

D. Owner

19. Which could be considered a single point of failure within a single sign-on implementation?

A. Authentication server

B. User's workstation

C. Logon credentials

D. RADIUS

20. What role does biometrics play in access control?

A. Authorization

B. Authenticity

C. Authentication

D. Accountability

21. What determines if an organization is going to operate under a discretionary, mandatory, or nondiscretionary access control model?

A. Administrator

B. Security policy

C. Culture

D. Security levels

22. What type of attack attempts all possible solutions?

A. Dictionary

B. Brute force

C. Man-in-the-middle

D. Spoofing

23. Spoofing can be described as which of the following?

A. Eavesdropping on a communication link

B. Working through a list of words

C. Session hijacking

D. Pretending to be someone or something else

24. Which of the following is not an advantage of a centralized access control administration?
    - **A.** Flexibility
    - **B.** Standardization
    - **C.** Higher level of security
    - **D.** No need for different interpretations of a necessary security level
25. Which of the following best describes what role-based access control offers companies in reducing administrative burdens?
    - **A.** It allows entities closer to the resources to make decisions about who can and cannot access resources.
    - **B.** It provides a centralized approach for access control, which frees up department managers.
    - **C.** User membership in roles can be easily revoked and new ones established as job assignments dictate.
    - **D.** It enforces an enterprise-wide security policy, standards, and guidelines.

Answers

1. **D.** Compared to the other available authentication mechanisms, biometric methods provide the highest level of protection and are the most expensive.
2. **B.** Most systems do not use the actual passphrase or password the user enters. Instead, they put this value through some type of encryption or hashing function to come up with another format of that value, referred to as a virtual password.
3. **C.** Passwords provide the least amount of protection, but are the cheapest because they do not require extra readers (as with smart cards and memory cards), do not require devices (as do biometrics), and do not require a lot of overhead in processing (as in cryptography). Passwords are the most common type of authentication method used today.
4. **A.** Separation of duties is put into place to ensure that one entity cannot carry out a task that could be damaging or risky to the company. It requires two or more people to come together to do their individual tasks to accomplish the overall task. If a person wanted to commit fraud and separation of duties was in place, they would need to participate in collusion.
5. **C.** A logical control is the same thing as a technical control. All of the answers were logical in nature except an ID badge. Badges are used for physical security and are considered physical controls.
6. **D.** The best approach to security is to try to prevent bad things from taking place by putting the necessary controls and mechanisms in place. Detective

controls should also be put in place, but a security model should not work from a purely detective approach.

7. **B.** Roles work as containers for users. The administrator or security professional creates the roles and assigns rights to them and then assigns users to the container. The users then inherit the permissions and rights from the containers (roles), which is how implicit permissions are obtained.
8. **B.** An asynchronous token device is based on challenge/response mechanisms. The authentication service sends the user a challenge value, which the user enters into the token. The token encrypts or hashes this value, and the user uses this as her one-time password.
9. **D.** The DAC model allows users, or data owners, the discretion of allowing other users access to their resources. DAC is implemented by ACLs, which the data owner can configure.
10. **D.** This is considered a strong authentication approach because it is two-factor—it uses two out of the possible three authentication techniques (something a person knows, is, or has).
11. **A.** Message authentication code (MAC) is a cryptographic function and is not a key component of Kerberos. Kerberos is made up of a KDC, a realm of principals (users, services, applications, devices), an authentication service, tickets, and a ticket granting service.
12. **D.** DAC is implemented and enforced through the use of access control lists (ACLs), which are held in a matrix. MAC is implemented and enforced through the use of security labels.
13. **C.** Authentication means to validate the identity of a user. In most systems, the user must submit some type of public information (username, account number) and a second credential to prove this identity. The second piece of the credential set is private and should not be shared.
14. **A.** It is easier on the administrator if she only has to create one role, assign all of the necessary rights and permissions to that role, and plug a user into that role when needed. Otherwise, she would need to assign and extract permissions and rights as each individual came and left the company.
15. **C.** As stated in a previous question, passwords are the most common authentication mechanism used today. They are used to validate a user's identity.
16. **A.** Mutual authentication means that it is happening in both directions. Instead of just the user having to authenticate to the server, the server also has to authenticate to the user.
17. **B.** Reviewing audit logs takes place after the fact, after some type of incident happens. It is detective in nature because the security professional is trying to figure out what exactly happened, how to correct it, and possibly who is responsible.

18. **D.** This question may seem a little confusing if you were stuck between user and owner. Only the data owner can decide who can access the resources she owns. She may be a user and she may not. A user is not necessarily the owner of the resource. Only the actual owner of the resource can dictate what subjects can actually access the resource.
19. **A.** In a single sign-on technology, all users are authenticating to one source. If that source goes down, authentication requests cannot be processed.
20. **C.** Biometrics is a technology that validates an individual's identity by reading a physical attribute.
21. **B.** The security policy sets the tone for the whole security program. It dictates the level of risk the management and company are willing to accept. This in turn dictates the type of controls and mechanisms that are to be put into place to ensure that this level of risk is not exceeded.
22. **B.** A brute force attack tries a combination of values in an attempt to discover the correct sequence that represents the captured password or whatever the goal of the task is. It is an exhaustive attack, meaning the attacker will try over and over again until she is successful.
23. **D.** Spoofing is the process of pretending to be another person or process with the goal of obtaining unauthorized access. Spoofing is usually done by using a bogus IP address, but it could be done by using someone else's authentication credentials.
24. **A.** A centralized approach does not provide as much flexibility as a decentralized access control administration, because one entity is making all the decisions instead of several entities that are closer to the resources. A centralized approach is more structured in nature, which means that there is less flexibility.
25. **C.** An administrator does not need to revoke and reassign permissions to individual users as they change jobs. Instead, the administrator assigns permissions and rights to a role, and users are plugged into those roles.