2. Which of the following is true about the Session Initiation Protocol (SIP)?
   - **A.** Used to establish virtual private network (VPN) sessions
   - **B.** Framework for authenticating network connections
   - **C.** Session layer protocol for out-of-band statistics
   - **D.** Application layer protocol used in online gaming communications
3. Which of the following is not considered a best practice for securing multimedia collaboration platforms?
   - **A.** Don't record meetings unless necessary
   - **B.** Use consumer-grade products
   - **C.** Use AES 256-bit encryption
   - **D.** Restrict participants' sharing of their screens or cameras as appropriate
4. How could you best protect a unified communications (UC) platform?
   - **A.** Protect it as you would any other systems
   - **B.** Enable Password Authentication Protocol (PAP)
   - **C.** Use the Session Initiation Protocol (SIP) for every new session
   - **D.** Ensure the hub is protected against physical and logical threats

Use the following scenario to answer Questions 5–7. You are the CISO of a research and development company that is transitioning to a 100 percent remote workforce, so your entire staff will be working from home. You don't have enough laptops for all your staff, so those without one will be using their personal computers and printers for work. Your VPN concentrators are sufficient to support the entire workforce, and you will be requiring all staff members to connect to the VPN.

5. Which authentication protocol would be best for your VPN connections?
   - **A.** Password Authentication Protocol (PAP)
   - **B.** Challenge Handshake Authentication Protocol (CHAP)
   - **C.** Extensible Authentication Protocol (EAP)
   - **D.** Session Initiation Protocol (SIP)
6. Which of the following additional VPN configurations should you also enable?
   - **A.** Split tunneling
   - **B.** Full tunneling
   - **C.** VPN kill switch
   - **D.** Hybrid tunneling

7. Which of the following will best protect the confidentiality of your sensitive research data?
   - **A.** Secure Shell (SSH)
   - **B.** Virtualized networks
   - **C.** Virtual desktop infrastructure (VDI)
   - **D.** Remote Procedure Calls (RPC)
8. During a recent review of your enterprise architecture, you realize that many of your mission-critical systems rely on Remote Procedure Call (RPC). What measures should you take to ensure remote procedure calls are secured?
   - **A.** Implement ITU standard H.323
   - **B.** Tunnel RPC through Transport Layer Security (TLS)
   - **C.** Use the Password Authentication Protocol (PAP) for authentication
   - **D.** Enforce client-side authentication
9. Which of the following is not an advantage of virtual desktops?
   - **A.** Reduced user downtime during incident response
   - **B.** Support for both persistent and nonpersistent sessions
   - **C.** Support for both physical and remote logins
   - **D.** Better implementation of data retention standards

Answers

1. **D.** The SS7 protocol is used in a PSTN to set up, control, and disconnect calls.
2. **D.** SIP is an application layer protocol used for call setup and teardown in IP telephony, video and multimedia conferencing, instant messaging, and online gaming.
3. **B.** Consumer-grade products almost always lack the security controls and management features that we need to properly secure multimedia collaboration platforms.
4. **D.** Securing UC involves similar security controls that we would apply to any other communications platform, but with a couple of important caveats. Unified communications rely on a central hub that integrates, coordinates, and synchronizes the various technologies. You want to ensure that this hub is adequately protected against physical and logical threats.
5. **C.** EAP is considered much more secure than both PAP (which is not secure at all) and CHAP. SIP does not provide authentication mechanisms at all.
6. **A.** Because your staff will be using printers on their home networks, you will have to enable split tunneling, which allows some traffic to be sent over the VPN and other traffic to go to the local network or to the Internet directly.

7. **C.** VDI allows your sensitive data to remain in your protected network even as users are able to work with it over a virtual desktop. Properly configured, this infrastructure prevents any sensitive research data from being stored on the remote user's computer.
8. **B.** Since many implementations of RPC lack security controls, many organizations require TLS for authenticating hosts and encrypting RPC traffic.
9. **C.** VDI is particularly helpful in regulated environments because of the ease with which it supports data retention, configuration management, and incident response through persistent and nonpersistent sessions. However, since VDI relies on VMs in a data center, there is not a computer at which a user could physically log in.

PART V

Identity and Access Management

This page intentionally left blank

CHAPTER 16

Identity and Access Fundamentals

This chapter presents the following:

- Identification, authentication, authorization, and accountability
- Credential management
- Identity management
- Federated identity management with a third-party service

The value of identity of course is that so often with it comes purpose.

—Richard Grant

The concept of identity is foundational to controlling access to our assets because everyone (and everything) that touches them must have a legitimate purpose in doing so. What makes access control tricky is that most of us have multiple identities that depend on the context in which we find ourselves. A person could simultaneously be an asset owner, custodian, and processor (roles we discussed in Chapter 5), depending on which asset we consider and at what time. On top of the challenge of handling multiple identities, we also have to ensure that each identity belongs to the person claiming it.

In this chapter, we discuss the fundamentals of user identification, authentication, and authorization. We do this while considering a variety of real-world contexts, such as complex enterprise environments and the interaction with third parties. Of course, we must be able to verify that things are being done correctly, so we also talk about accountability in these efforts. This all sets the stage for the next chapter, in which we delve into how to actually manage identities and access.

Identification, Authentication, Authorization, and Accountability

For users to be permitted to access any resource, they first must prove they are who they claim to be, have the necessary credentials, and have been given the necessary rights or privileges to perform the actions they are requesting. Once these steps are completed successfully, it is necessary to track users' activities and enforce accountability for their

actions. *Identification* describes a method by which a subject (user, program, or process) claims to have a specific identity (username, account number, or e-mail address). *Authentication* is the process by which a system verifies the identity of the subject, usually by requiring a piece of information that only the claimed identity should have. This piece could be a password, passphrase, cryptographic key, personal identification number (PIN), physiological characteristic, or token. Together, the identification and authentication information (for example, username and password) make up the subject's *credentials*. These credentials are compared 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 authenticated, 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 may look at an 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, authorization, and accountability 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 may not have the authorization to access certain 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 16-1 illustrates the four steps that must happen for a subject to access an object.

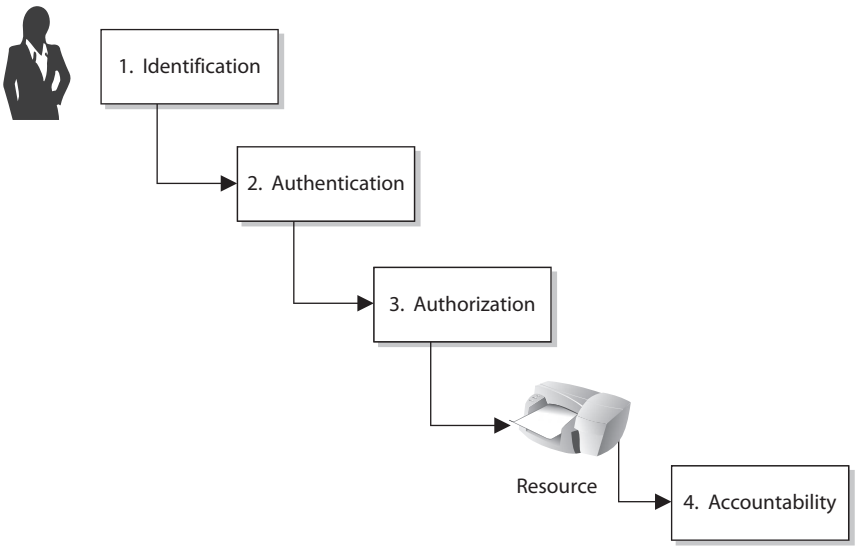


Figure 16-1 Four steps must happen for a subject to access an object: identification, authentication, authorization, and accountability.

Race Condition

A *race condition* occurs when processes carry out their tasks on a shared resource in an incorrect order. A race condition is possible when two or more processes use a shared resource, such as 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 would be much different than if process 1 carried out its tasks on the data before process 2.

In software, when the authentication and authorization steps are split into two functions, there is a possibility 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 occurs when two or more processes use the same resource and the sequence of steps within the software can be carried out in an improper order, something that 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.

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 technical 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 all vulnerabilities are covered without producing overlaps of functionality. However, if it were easy, security professionals would not be getting paid the big bucks!

EXAM TIP 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 must 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, or PIN). Entering public information is the identification step, while 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.

Identification and Authentication

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. Three main types of factors can be used for authentication: *something a person knows*, *something a person has*, and *something a person is*. Sometimes, these factors are combined with two additional factors: *somewhere a person is* (logical or physical location) and *something a person does* (behavioral factor). These location and behavioral factors may not be all that strong by themselves, but when combined with other factors they can significantly improve the effectiveness of the authentication process.

Something a person knows (knowledge-based authentication [KBA]) can be, for example, a password, PIN, mother's maiden name, or the combination to a lock. Authenticating a person by something that she knows is usually the least expensive method to implement. The downside to this method is that another person may acquire this knowledge and gain unauthorized access to a resource.

Something a person has (ownership-based authentication) 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 (biometric authentication) becomes a bit more interesting. This is not based on whether the person is an American, a geek, or an athlete—it is based on a physical attribute. Authenticating a person's identity based on a unique physical attribute is referred to as biometrics.

Strong authentication contains two or all 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 into a keypad before the biometric scan is performed. This satisfies the "something the person knows" category. Conversely, the person could be required to swipe a magnetic card through a

One-to-One and One-to-Many

Verification 1:1 is the measurement of an identity against a single claimed identity. The conceptual question is, "Is this person who he claims to be?" So if Bob provides his identity and credential set, this information is compared to the data kept in an authentication database. If they match, we know that it is really Bob. If the identification is *1:N (many)*, the measurement of a single identity is compared against multiple identities. The conceptual question is, "Who is this person?" An example is if fingerprints were found at a crime scene, the cops would run them through their database to identify the suspect.

reader prior to the biometric scan. This would satisfy the "something the person has" category. Whatever identification system is used, for strong authentication to be in the process, it must include multiple factors.

TIP Strong authentication is also sometimes referred to as *multifactor authentication (MFA)*, which just means that more than one authentication method is used. While two-factor authentication (2FA) is common, *three-factor authentication* (for example, smart card, PIN, and retinal scan) is sometimes used.

Identity is a complicated concept with many varied nuances, ranging from the philosophical to the practical. A person may have multiple digital identities. For example, a user could be JPublic in a Windows domain environment, JohnP on a Unix server, JohnPublic on the mainframe, JJP in instant messaging, JohnCPublic in the certification authority, and JohnnyPub on Facebook. If the organization that employs that user wants to centralize all of its access control, these various identity names for the same person may cause the security administrator undue stress.

NOTE *Mutual authentication* is when the two communicating entities must authenticate to each other before passing data. For example, an authentication server may be required to authenticate to a user's system before allowing data to flow back and forth.

While most of this chapter deals with user authentication, it is important to realize system-based authentication is possible also. Computers and devices can be identified, authenticated, monitored, and controlled based upon their hardware addresses (media access control) and/or Internet Protocol (IP) addresses. Networks may have network access control (NAC) technology that authenticates systems before they are allowed access to the network. Every network device has a hardware address that is integrated into its network interface card (NIC) and a software-based address (IP) that either is assigned by a Dynamic Host Configuration Protocol (DHCP) server or locally configured.

Identification Component Requirements

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

- Each identifier 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.

Knowledge-Based Authentication

We start off our discussion of authentication methods by looking at the most commonly used approach: using something that a person knows. This knowledge-based approach typically uses a password, passphrase, or cognitive password. Let's take a closer look at each.

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, and in order to ensure its effectiveness for authentication, it must be kept secret.

Password Policies Although passwords are prevalent, they are also considered one of the weakest security mechanisms available. Why? Users usually choose passwords that are easily guessed (a spouse's name, a user's birth date, or a dog's name), or tell others their passwords, and many times write the passwords down on a sticky note and 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, that is. Then security is all the rage.

This is where password policies step 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 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, nondictionary words to help users remember them so they aren't tempted to write them down.

If 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, and not be easily guessable. The operating system can keep track of the passwords a specific user generates so as to ensure no passwords are reused. In March of 2020 the National Institute of Standards and Technology (NIST) updated its guidelines concerning passwords in SP 800-63B. These include the following recommendations:

- **Increased password length** The longer the password, the harder it is to guess. The recommended minimum password length is 8 characters for user-selected ones and 6 characters for computer-generated passwords. The maximum recommended length is 64 characters.
- **Allow special characters** Users should be allowed to use any special character, and even emojis, in their passwords. Special characters, however, should not be required.
- **Disallow password hints** On the surface, password hints may seem to make sense because they allow users to remember complex passwords and reduce reliance on password resetting features. However, they mostly help attackers.

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** Comparing files of thousands of words to the user's password until a match is found.
- **Social engineering** Falsely convincing an individual that she has the necessary authorization to access specific resources.
- **Rainbow table** Using a table that contains all possible passwords already in a hash format.

Certain techniques 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 whether there were any unsuccessful logon attempts. This alerts the user to any suspicious activity and whether anyone has attempted to log on using his credentials. An administrator can set system 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, for example, 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 both successful and unsuccessful logon attempts. This audit information should include the date, time, user ID, and workstation the user logged in from.

NOTE *Clipping level* is an older term that just means threshold. If the number of acceptable failed login attempts is set to three, three is the threshold (clipping level) value.

Policies can also specify other conditions that make passwords more difficult to exploit. Many organizations maintain a password history so users cannot reuse passwords within a certain timeframe. A variation on this is having the system remember the last *n* (where *n* is some number greater than or equal to one) passwords to prevent their reuse. Policies can also specify maximum age (that is, expiration) and minimum age (so the password can't be changed immediately to bypass the other policies) requirements.

As with many things in life, education is the key. Password requirements, protection, and generation should be addressed in security awareness programs so 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.

NOTE Rainbow tables contain passwords already in their hashed format. The attacker just compares a captured hashed password with one that is listed in the table to uncover the plaintext password. This takes much less time than carrying out a dictionary or brute-force attack.

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 used to discover users' passwords. Many times the same tools employed by an attacker 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 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 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 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 Message Digest 5 (MD5) or Secure Hash Algorithm (SHA), to ensure passwords are not sent in cleartext.

Although some people think the world is run by Microsoft, other types of operating systems are 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 "shadow." This shadow file does not contain passwords in cleartext; instead, your password is run through a hashing algorithm, 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 added to passwords prior to hashing to add more complexity and randomness. The more randomness entered into the hashing 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 several thousand different hashes. This makes it much more difficult for an adversary to attack the passwords in your system using approaches like rainbow tables.

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.

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 are required by the application. (For example, an application may require your virtual password to be 128 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 a user creates a password the first time he logs on to 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.

Cognitive Password

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 the first person they kissed, the name of their best friend in 8th grade, or their favorite cartoon character. 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.

EXAM TIP Knowledge-based authentication means that a subject is authenticated based upon something she knows. This could be a PIN, password, passphrase, cognitive password, personal history information, or through the use of a CAPTCHA, which is the graphical representation of data. A CAPTCHA is a skewed representation of characteristics a person must enter to prove that the subject is a human and not an automated tool as in a software robot.

Biometric Authentication

Biometrics verifies an individual's identity by analyzing a unique personal characteristic, 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. Biometric systems typically

base authentication decisions on physical attributes (such as iris, retina, or fingerprint), which provides more accuracy because physical attributes typically don't change, absent some disfiguring injury, and are harder to impersonate.

Biometrics is typically broken up into two different categories:

- **Physiological** This category of biometrics uses physical attributes unique to a specific individual to verify that person's identity. Fingerprints are a common example of a physiological trait used in biometric systems. Physiological is "what you are."
- **Behavioral** This approach is based on something an individual does uniquely to confirm her identity. An example is signature dynamics. Behavioral is "what you do."

A biometric system scans a person's physiological attribute or behavioral trait and compares it to a record 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 must be extremely sensitive. The system must perform accurate and repeatable measurements of anatomical or behavioral characteristics. This type of sensitivity can easily cause false positives or false negatives. The system must be calibrated so 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 [FRR]). When the system accepts impostors who should be rejected, it is called a *Type II error* (false acceptance rate [FAR]). 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 FRR equals the FAR. 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 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 the rep 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 a vendor about their product's 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 rate 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 typically 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 it would have to accept a higher rate of Type I errors.

Biometric authentication is the most expensive method of verifying a person's identity, and it faces other barriers to becoming widely accepted. These include user acceptance, enrollment timeframe, and throughput. Many people are reluctant to let a machine read the pattern of their retina or scan the geometry of their hand. 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 is unable to get a clear reading of an iris scan or cannot 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.

During enrollment, the user provides the biometric data (e.g., fingerprint, voice print, or retina scan), and the biometric reader converts this data into binary values. Depending on the system, the reader may create a hash value of the biometric data, or it may encrypt the data, or do both. The biometric data then goes from the reader to a backend authentication database where the user's account has been created. When the user needs to later authenticate to a system, she provides the necessary biometric data, and the binary format of this information is compared to what is in the authentication database. If they match, then the user is authenticated.

In Figure 16-2, we see that biometric data can be stored on a smart card and used for authentication. Also, you might notice that the match is 95 percent instead of 100 percent. Obtaining a 100 percent match every time is very difficult because of the level of sensitivity of the biometric systems. A smudge on the reader, oil on the person's finger, and other small environmental issues can stand in the way of matching 100 percent. If your biometric system was calibrated so it required 100 percent matches, this would mean you would not allow any Type II errors and that users would commonly not be authenticated in a timely manner.

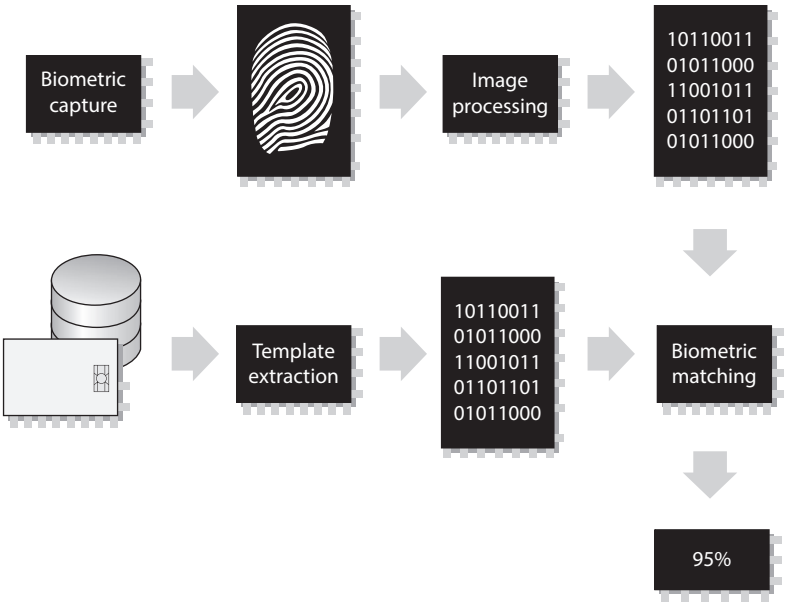


Figure 16-2 Biometric data is turned into binary data and compared for identity validation.

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.

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

Fingerprint

Fingerprints are made up of ridge endings and bifurcations exhibited by friction ridges and other detailed characteristics 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 shape, 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 is unique for each person. A camera is used to project a beam inside the eye and capture the pattern and compare it to a reference file recorded previously.

NOTE Retina scans are extremely invasive and involve a number of privacy issues. Since the information obtained through this scan can be used in the diagnosis of medical conditions, it could very well be considered protected health information (PHI) subject to healthcare information privacy regulations such as HIPAA.

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. Of the biometric systems, iris scans are the most accurate. The iris remains constant through adulthood, which reduces the type of errors that can happen during the authentication process. Sampling the iris offers more reference coordinates than any other type of biometric. Mathematically, this means it has a higher accuracy potential than any other type of biometric.

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

Signature Dynamics

When a person writes a signature, usually they do so in the same manner and at the same speed each time. Writing 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.

Keystroke Dynamics

Whereas signature dynamics is a method that captures the electrical signals when a person signs a name, keystroke 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 of typing, 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

People's speech sounds and patterns have many subtle distinguishing differences. A biometric system that is programmed to capture a voice print and compare it to the information held 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 technique is used so others cannot 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.

A naïve implementation of this technology could be fooled by a photograph of the legitimate user. To thwart this approach, the scanner can perform a three-dimensional measurement of the user's face by projecting thousands of infrared dots on it. This is how Apple's Face ID works.

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

Biometric Issues and Concerns

Biometric systems are not without their own sets of issues and concerns. Because they depend upon the specific and unique traits of living things, problems can arise. Living things are notorious for not remaining the same, which means they won't present static biometric information for every login attempt. Voice recognition can be hampered by a user with a cold. Retinas can detach. Someone could lose a finger. Or all three could happen. You just never know in this crazy world.

Some biometric systems actually check for the pulsation and/or heat of a body part to make sure it is alive. So if you are planning to cut someone's finger off or pluck out someone's eyeball so you can authenticate yourself as a legitimate user, it may not work. Although not specifically stated, this type of activity definitely falls outside the bounds of the CISSP ethics you will be responsible for upholding once you receive your certification.

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.

Ownership-Based Authentication

Authentication can also be based on something that the subject has. This is almost always some sort of physical or logical token. It can be a device such as a phone, identification card, or even an implanted device. It can also be a cryptographic key, such as a private key in public key infrastructure (PKI). Sometimes, access to the token is protected by some other authentication process, such as when you have to unlock your phone to get to a software-based token generator.

One-Time Password

A *one-time password (OTP)*, also called a *dynamic password*, is used for authentication purposes and is valid only once. After the password is used, it is no longer valid; thus, it can't be reused if a hacker obtains it. The password is generated by a token device, which is something the person owns (or at least carries around). This device is the most common implementation mechanism for OTP and generates the one-time password for the user to submit to an authentication server. It is commonly implemented in three formats: as a dedicated physical device with a small screen that displays the OTP, as a smartphone application, and as a service that sends an SMS message to your phone. The following sections explain the concepts behind this technology.

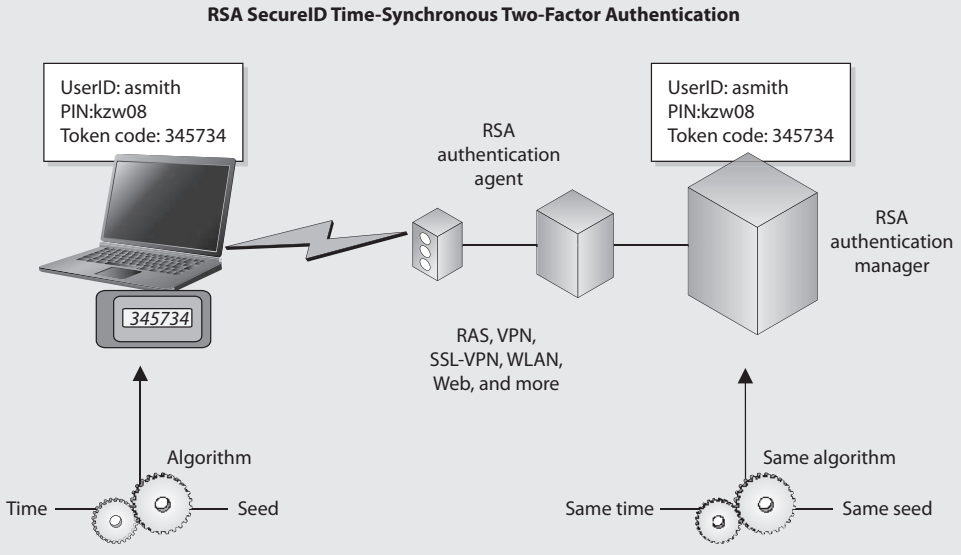
NOTE SMS was deprecated as a means of providing 2FA by the NIST in 2017. It is widely considered an insecure channel but is unfortunately still in common use.

The Token Device The token device, or password generator, is usually a handheld device that has a display and possibly a keypad. This hardware is separate from the computer the user is attempting to access. The token device and authentication service must 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 on to a computer. Only the token device and authentication service know the meaning of these characters. Because the two are synchronized, the token device presents 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* requires the device and the authentication service to advance to the next OTP in sync with each other. This change can be triggered by time (e.g., every 30 seconds a new OTP is in play) or by simply going down a pre-agreed sequence of passwords, each of which is used only once before both the device and the server advance to the next one. The device displays the OTP to the user, who then enters this value and a user ID. The authentication service decrypts credentials and compares the OTP to the value it expects. If the two match, the user is authenticated and allowed to access the system.

RSA SecurID

RSA SecurID, from RSA Security LLC, is a well-known time-based token. One version of the product generates the OTP 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.

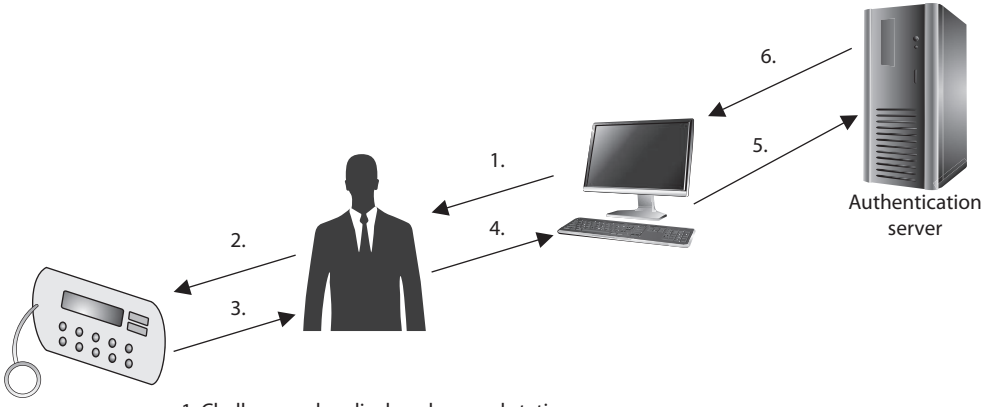


EXAM TIP Synchronous token-based OTP generation can be time-based or counter-based. Another term for counter-based is event-based. Counter-based and event-based are interchangeable terms, and you could see either or both on the CISSP exam.

Asynchronous A token device using an *asynchronous token*–generating method employs 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 the user uses as an OTP. 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 sent earlier, the user is authenticated, as shown in Figure 16-3.

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



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. AS sends an "allow access" response.

Figure 16-3 Authentication using an asynchronous token device includes a workstation, token device, and authentication service.

If the user has to enter a password or PIN into the token device before it provides an OTP, 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 OTPs.

Cryptographic Keys

Another way to prove one's identity is to use asymmetric cryptography and let the users' private keys show they are who they claim to be. Recall that the private key is kept secret by an individual and should never be shared. So, if the authentication server has (or gets a hold of) the user's public key, it can use that key to encrypt a challenge and send it to the user. Only the person owning the corresponding private key would be able to decrypt it and respond to it. Ideally, the user then encrypts the response using the server's public key to provide mutual authentication. This approach is commonly used in Secure Shell (SSH) instead of passwords, which are the weakest form of authentication and can be easily sniffed as they travel over a network.

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 the user only needs to type in a user ID or PIN and present the memory card, and if the data that the user enters 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 backend 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 slide the ATM card (or memory card) through the reader and enter the correct PIN.

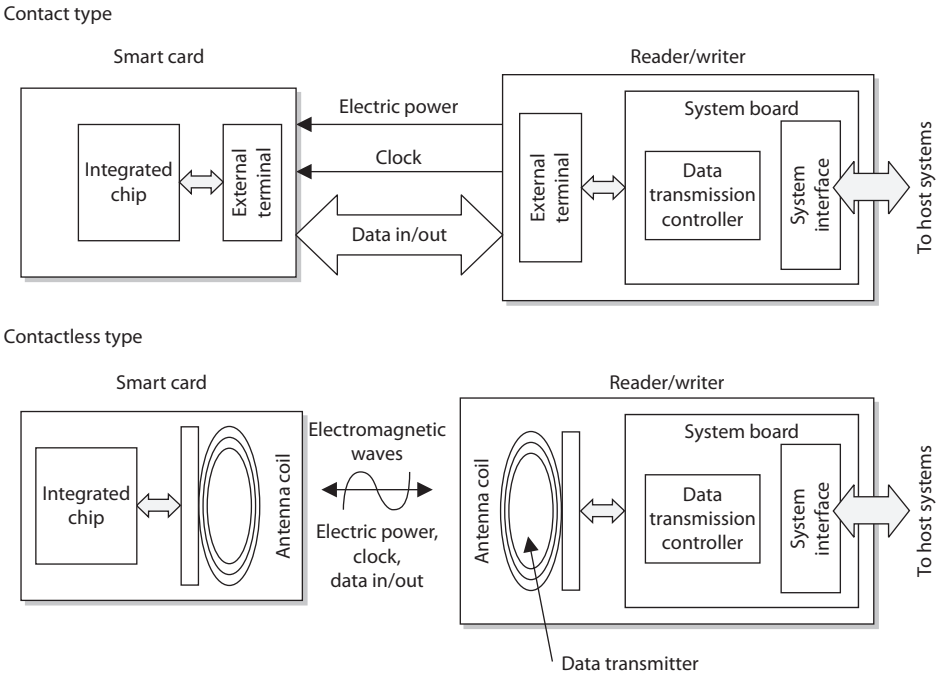
Memory cards can be used with computers, but they require a reader to process the information. The reader adds cost to the process, especially when one is needed per computer, and card generation adds cost and effort to the whole authentication process. Using a memory card provides a more secure authentication method than using a password because the attacker would need to obtain the card and know the correct PIN. Administrators and management must 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

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 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 supplies 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 employing a challenge/response value, or by providing the user's private key if it is used within a PKI environment.



PART V

TIP Two types of contactless smart cards are available: 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 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. Essentially, a smart card is a kind of computer, and because of that it has many of the operational challenges and risks that can affect a computer.

Smart cards have several different capabilities, and as the technology develops and memory capacities increase for storage, they will gain even more. They can store personal information in a storage manner that is tamper resistant. This also gives them the capability to isolate security-critical computations within themselves. They can be used in encryption systems to store keys and have a high level of portability as well as security. The memory and integrated circuit also provide the capacity to use encryption algorithms on the actual card and use them for secure authorization that can be utilized throughout an entire organization.

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 the industry throws at them. Over the years, criminals have become very inventive in the development of various ways to attack smart cards. Smart card attacks tend to be special cases of the cryptanalysis techniques we discussed in Chapter 8. For example, attackers have introduced computational errors into smart cards with the goal of uncovering the encryption keys 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 may allow an attacker to reverse-engineer the encryption process, with the hope 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. 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

Interoperability

In the industry today, lack of interoperability is a big problem. 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

intrusive measures. Some examples of side-channel attacks that have been carried out on smart cards are *differential power analysis* (examining the power emissions released during processing), *electromagnetic analysis* (examining the frequencies emitted), and *timing* (how long a specific process takes to complete). These types of attacks are used to uncover sensitive information about how a component works without trying to compromise any type of flaw or weakness. They are commonly used for data collection. Attackers monitor and capture the analog characteristics of all supply and interface connections and any other electromagnetic radiation produced by the processor during normal operation. They can also collect the time it takes for the smart card to carry out its function. From the collected data, the attacker can deduce specific information she is after, which could be a private key, sensitive financial data, or an encryption key stored on the card.

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 into the card instructions that will allow 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.

A more intrusive smart card attack is called *microprobing*, which uses needleless and ultrasonic vibration to remove the outer protective material on the card's circuits. Once this is completed, data can be accessed and manipulated by directly tapping into the card's ROM chips.

Near Field Communications

Near Field Communication (NFC) is a short-range (i.e., a few centimeters) radio frequency (RF) communications technology that provides data communication on a base frequency of 13.56 MHz. Manufacturers of NFC devices abide by ISO/IEC 18092 for international interoperability. While this technology is perhaps best known for contactless payments using mobile phones, it is also used for contactless smart cards.

Credential Management

Credential management deals with creating user accounts on all systems, assigning and modifying the account details and privileges when necessary, and decommissioning the accounts when they are no longer needed. In many environments, the IT department creates accounts manually on the different systems, users are given excessive rights and permissions, and when an employee leaves the organization, many or all of the accounts stay active. This typically occurs because a centralized credential management technology has not been put into place.

Credential management products attempt to attack these issues by allowing an administrator to manage user accounts across multiple systems. When there are multiple directories containing user profiles or access information, the account management software allows for replication between the directories to ensure each contains the same up-to-date information. This automated workflow capability not only reduces the potential errors that can take place in account management, it also logs and tracks each step (including account approval). This allows for accountability and provides documentation for use in backtracking if something goes wrong. Automated workflow also helps ensure that only the necessary amount of access is provided to the account and that there are no "orphaned" accounts still active when employees leave the organization. In addition, these types of processes are the kind your auditors will be looking for—and we always want to make the auditors happy!

NOTE These types of credential management products are commonly used to set up and maintain internal accounts. Web access control management is used mainly for external users.

Enterprise credential management products are usually expensive and can take time to properly roll out across the enterprise. Regulatory requirements, however, are making more and more organizations spend the money for these types of solutions—which the vendors love! In the following sections, we'll explore the many facets of a good credential management solution.

Password Managers

Two of the best practices when it comes to password-based authentication are to use complex passwords/passphrases and to have a different one for each account; accomplishing both from memory is a tall order for most of us. A popular solution to address this challenge is to use software products that remember our credentials for us. These products, known as *password managers* or *password vaults*, come in two flavors: as a stand-alone application or as a feature within another application (such as a web browser). In either case, the application stores user identifiers and passwords in a password-encrypted data store. The user need only remember this master password and the application maintains all others. These products typically provide random password generation and allow the user to store other information such as URLs and notes. Most modern web browsers also provide features that remember the user identifiers and passwords for specific websites.

An obvious problem with using password vaults is that they provide one-stop-shopping for malicious actors. If they can exploit this application, they gain access to all of the user's credentials. Developers of these applications go to great lengths to ensure they are secure, but as we all know there is no such thing as a 100 percent secure system. In fact, there have been multiple documented vulnerabilities that allowed adversaries to steal these (supposedly secure) credentials.

Password Synchronization

Another approach to credential management is to use password synchronization technologies that can allow a user to maintain just one password across multiple systems. The product synchronizes the password to other systems and applications, which happens transparently to the user. The goal is to require the user to memorize only one password, which enables the organization to enforce more robust and secure password requirements. If a user needs to remember only one password, he is more likely to not have a problem with longer, more complex strings of values. This reduces help-desk call volume and allows the administrator to keep her sanity for just a little bit longer.

One criticism of this approach is that since only one password is used to access different resources, the hacker only has to figure out one credential set to gain unauthorized access to all resources. But if the password requirements are more demanding (12 characters, no dictionary words, three symbols, upper- and lowercase letters, and so on) and the password is changed out regularly, the balance between security and usability can be acceptable.

Self-Service Password Reset

Some products are implemented to allow users to reset their own passwords. This does not mean that the users have any type of privileged permissions on the systems to allow them to change their own credentials. Instead, during the registration of a user account, the user can be asked to provide several personal questions (first car, favorite teacher, favorite color, and so on) in a question-and-answer form. When the user forgets his password, he may be required to provide another authentication mechanism (smart card, token, etc.) and to answer these previously answered questions to prove his identity.

Products are available that allow users to change their passwords through other means. For example, if you forgot your password, you may be asked to answer some of the questions answered during the registration process of your account (i.e., a cognitive password). If you do this correctly, an e-mail is sent to you with a link you must click. The password management product has your identity tied to the answers you gave to the questions during your account registration process and to your e-mail address. If you do everything correctly, you are given a screen that allows you to reset your password.

CAUTION The product should not ask for information that is publicly available, as in your mother's maiden name, because anyone can find that out and attempt to identify himself as you.

Assisted Password Reset

Some products are created for help-desk employees who need to work with individuals when they forget their password. The help-desk employee should not know or ask the individual for her password. This would be a security risk since only the owner of the password should know the value. The help-desk employee also should not just change a password for someone calling in without authenticating that person first. This can allow social engineering attacks where an attacker calls the help desk and indicates she is someone who she is not. If this were to take place, an attacker would have a valid employee password and could gain unauthorized access to the organization's jewels.

The products that provide assisted password reset functionality allow the help-desk individual to authenticate the caller before resetting the password. This authentication process is commonly performed through the use of cognitive passwords described in the previous section. The help-desk individual and the caller must be identified and authenticated through the password management tool before the password can be changed. Once the password is updated, the system that the user is authenticating to should require the user to change her password again. This would ensure that only she (and not she and the help-desk person) knows her password. The goal of an assisted password reset product is to reduce the cost of support calls and ensure all calls are processed in a uniform, consistent, and secure fashion.

Just-in-Time Access

You probably don't want your general users having administrative privileges on their computers. However, if you apply the security principle of least privilege (described in Chapter 9), your users will probably lack the authorization to perform many functions that you would like them to be able to perform in certain circumstances. From having their laptops "forget" wireless networks to which they may have connected, to updating software, there are many scenarios in which a regular user may need administrative (or otherwise elevated) credentials. The traditional approach is to have the user put in a ticket and wait for an IT administrator to perform the action for the user. This is a costly way of doing business, particularly if you have a large organization.

Just-in-time (JIT) access is a provisioning methodology that elevates users to the necessary privileged access to perform a specific task. This is a way to allow users to take care of routine tasks that would otherwise require IT staff intervention (and possibly decrease user productivity). This approach mitigates the risk of privileged account abuse by reducing the time a threat actor has to gain access to a privileged account. JIT access is usually granted in a granular manner, so that it applies to a specific resource or action in a given timeframe. For example, if users need administrative rights to allow a conferencing application access to their desktop, they can be granted one-time access to change that particular setting in their systems and then it's gone.

Registration and Proofing of Identity

Now let's think about how accounts are set up. In many environments, when a new user needs an account, a network administrator sets up the account(s) and provides some type

of privileges and permissions. But how would the network administrator know what resources this new user should have access to and what permissions should be assigned to the new account? In most situations, she doesn't—she just wings it. This is how users end up with too much access to too many resources. What should take place instead is implementation of a workflow process that allows for a request for a new user account. Since hardly anyone in the organization likely knows the new employee, we need someone to vouch for this person's identity. This process, sometimes called *proofing of identity*, is almost always carried out by human resources (HR) personnel who would've had to verify the new employee's identity for tax and benefit purposes. The new account request is then sent to the employee's manager, who verifies the permissions that this person needs, and a ticket is generated for the technical staff to set up the account(s).

If there is a request for a change to the permissions on the account or if an account needs to be decommissioned, it goes through the same process. The request goes to a manager (or whoever is delegated with this approval task), the manager approves it, and the changes to the various accounts take place.

Over time, this new user will commonly have different identity attributes, which will be used for authentication purposes, stored in different systems in the network. When a user requests access to a resource, all of his identity data has already been copied from other identity stores and the HR database and held in this centralized directory (sometimes called the *identity repository*). When this employee parts with the organization for any reason, this new information goes from the HR database to the directory. An e-mail is automatically generated and sent to the manager to allow this account to be decommissioned. Once this is approved, the account management software disables all of the accounts that had been set up for this user.

User provisioning refers to the creation, maintenance, and deactivation of user objects and attributes as they exist in one or more systems, directories, or applications, in response to business processes. User provisioning software may include one or more of the following components: change propagation, self-service workflow, consolidated user administration, delegated user administration, and federated change control.

Authoritative System of Record

The authoritative source is the "system of record," or the location where identity information originates and is maintained. It should have the most up-to-date and reliable identity information. An *authoritative system of record (ASOR)* is a hierarchical tree-like structure system that tracks subjects and their authorization chains. Organizations need an automated and reliable way of detecting and managing unusual or suspicious changes to user accounts and a method of collecting this type of data through extensive auditing capabilities. The ASOR should contain the subject's name, associated accounts, authorization history per account, and provision details. This type of workflow and accounting is becoming more in demand for regulatory compliance because it allows auditors to understand how access is being centrally controlled within an environment.

User objects may represent employees, contractors, vendors, partners, customers, or other recipients of a service. Services may include e-mail, access to a database, access to a file server or database, and so on.

Great. So we create, maintain, and deactivate accounts as required based on business needs. What else does this mean? The creation of the account also is the creation of the access rights to organizational assets. It is through provisioning that users either are given access or have access taken away. Throughout the life cycle of a user identity, access rights, permissions, and privileges should change as needed in a clearly understood, automated, and audited process.

Profile Update

Most companies do not just contain the information "Bob Smith" for a user and make all access decisions based on this data. There can be a plethora of information on a user that is captured (e-mail address, home address, phone number, and so on). When this collection of data is associated with the identity of a user, it is called a *profile*.

Profiles should be centrally located to enable administrators to efficiently create, edit, or delete these profiles in an automated fashion when necessary. Many user profiles contain nonsensitive data that users can update themselves (called *self-service*). So, if George moved to a new house, there should be a profile update tool that allows him to go into his profile and change his address information. Now, his profile may also contain sensitive data that should not be available to George—for example, his access rights to resources or information that he is going to be laid off on Friday.

You have interacted with a profile update technology if you have requested to update your personal information on any e-commerce website. These companies provide you with the capability to sign in and update the information they allow you to access. This could be your contact information, home address, purchasing preferences, or credit card data. They then use this information to update their customer relationship management (CRM) systems so they know where to send you their junk mail advertisements and spam messages!

Session Management

A *session* is an agreement between two parties to communicate interactively. Think of it as a phone call: you dial your friend's number, she decides whether to answer, and if she does then you talk with each other until something happens to end the call. That "something" could be that you (or her) are out of time and have to go, or maybe one of you runs out of things to say and there's an awkward silence on the line, or maybe one of you starts acting weird and the other is bothered and hangs up. Technically, the call could go on forever, though in practice that doesn't happen.

Information systems use sessions all the time. When you show up for work and log onto your computer, you establish an authenticated session with the operating system that allows you to launch your e-mail client. When that application connects to the mail server, it establishes a different authenticated session (perhaps using the same credentials you used to log onto your computer). So, a session, in the context of information systems security, can exist between a user and an information system or between two

information systems (e.g., two running programs). If the session is an authenticated one, as in the previous two examples, then authentication happens at the beginning and then everything else is trusted until the session ends.

That trust is the reason we need to be very careful about how we deal with our sessions. Threat actors often try to inject themselves into an authenticated session and hijack it for their own purposes. Session management is the process of establishing, controlling, and terminating sessions, usually for security reasons. The session establishment usually entails authentication and authorization of one or both endpoints. Controlling the session can involve logging the start and end and anything in between. It could also keep track of time, activity, and even indicia of malicious activity. These are three of the most common triggers for session termination:

- **Timeout** When sessions are established, the endpoints typically agree on how long they will last. You should be careful to make this time window as short as possible without unduly impacting the organization. For example, a VPN concentrator could enforce sessions of no more than eight hours for your teleworkers.
- **Inactivity** Some sessions could go on for very long periods of time, provided that the user is active. Sessions that are terminated for inactivity tend to have a shorter window than those that are triggered only by total duration (i.e., timeout). For example, many workstations lock the screen if the user doesn't use the mouse or keyboard for 15 minutes.
- **Anomaly** Usually, anomaly detection is an additional control added to a session that is triggered by timeouts or inactivity (or both). This control looks for suspicious behaviors in the session, such as requests for data that are much larger than usual or communication with unusual or forbidden destinations. These can be indicators of session hijacking.

Accountability

Auditing capabilities ensure 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 deter wrongdoing, 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 enabled 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 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.

Audit trails can also be used to provide alerts about any suspicious activities that can be investigated at a later time. In addition, they can be valuable in determining exactly how far an attack has gone and the extent of the damage that may have been caused. It is important to make sure a proper chain of custody is maintained to ensure any data collected can later be properly and accurately represented in case it needs to be used for later events such as criminal proceedings or investigations.

Keep the following in mind when dealing with auditing:

- Store the audits securely.
- Use audit tools that keep the size of the logs under control.
- Protect the logs from any unauthorized changes in order to safeguard data.
- Train staff to review the data in the right manner while protecting privacy.
- Make sure the ability to delete logs is only available to administrators.
- Configure logs to contain activities of all high-privileged accounts (root, administrator).

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 has been violated. If the environment does not require such levels of security, the events analyzed would be fewer, with less-demanding thresholds.

Without proper oversight, 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 is 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 applications

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 must be deliberately configured. For example, an administrator can audit each logon attempt or just each failed logon attempt. System performance can look 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, the IDS alerts the administrator or staff member so 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 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

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 retention information should be stated in the organization's security policy and procedures.

Reviewing audit information manually can be overwhelming. Fortunately, 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.

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.

Today, more organizations are implementing *security information and event management (SIEM)* systems. These products gather logs from various devices (servers, firewalls, routers, etc.) and attempt to correlate the log data and provide analysis capabilities. Reviewing logs manually looking for suspicious activity in a continuous manner is not only mind-numbing; it is close to impossible to be successful. So many packets and network communication data sets are passing along a network, humans cannot collect all the data in real or near real time, analyze it, identify current attacks, and react—it is just too overwhelming.

Organizations also have different *types* of systems on a network (routers, firewalls, IDS, IPS, servers, gateways, proxies) collecting logs in various proprietary formats, which requires centralization, standardization, and normalization. Log formats are different per product type and vendor. The format of logs created by Juniper network device systems is different from the format of logs created by Cisco systems, which in turn is different from the format created by Palo Alto and Barracuda firewalls. It is important to gather logs from various different systems within an environment so that some type of situational awareness can take place. Once the logs are gathered, intelligence routines need to be processed on them so that data mining can take place to identify patterns. The goal is to piece together seemingly unrelated event data so that the security team can fully understand what is taking place within the network and react properly.

NOTE Situational awareness means that you understand the current environment even though it is complex, dynamic, and made up of seemingly unrelated data points. You need to be able to understand each data point in its own context within the surrounding environment so that you can make the best possible decisions.

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 and stored on a remote host.

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, hashing 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* (optical discs) 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 are not compromised in any way.

NOTE We cover investigative techniques and evidence handling in Chapter 22.

Identity Management

Identity management (IdM) is a broad term that encompasses the use of different products to identify, authenticate, and authorize users through automated means. It usually includes user account management, access control, credential management, single sign-on (SSO) functionality, managing rights and permissions for user accounts, and auditing and monitoring all of these items. It is important for security professionals to understand all the technologies that make up a full enterprise IdM solution. IdM requires managing uniquely identified entities, their attributes, credentials, and entitlements. IdM allows organizations to create and manage digital identities' life cycles (create, maintain, terminate) in a timely and automated fashion. An enterprise IdM solution must meet business needs and scale from internally facing systems to externally facing systems. In this section, we cover many of these technologies and how they work together.

NOTE Identity and access management (IAM) is another term that is used interchangeably with IdM, though ISC² considers IdM to be a subset of IAM.

Selling identity management products is a flourishing market that focuses on reducing administrative costs, increasing security, meeting regulatory compliance, and improving upon service levels throughout enterprises. The continual increase in complexity and diversity of networked environments also increases the complexity of keeping track of who can access what and when. Organizations have different types of applications, network operating systems, databases, enterprise resource management (ERM) systems, customer relationship management (CRM) systems, directories, and mainframes—all used for different business purposes. Organizations also have partners, contractors, consultants, employees, and temporary employees. (Figure 16-4 provides a simplistic

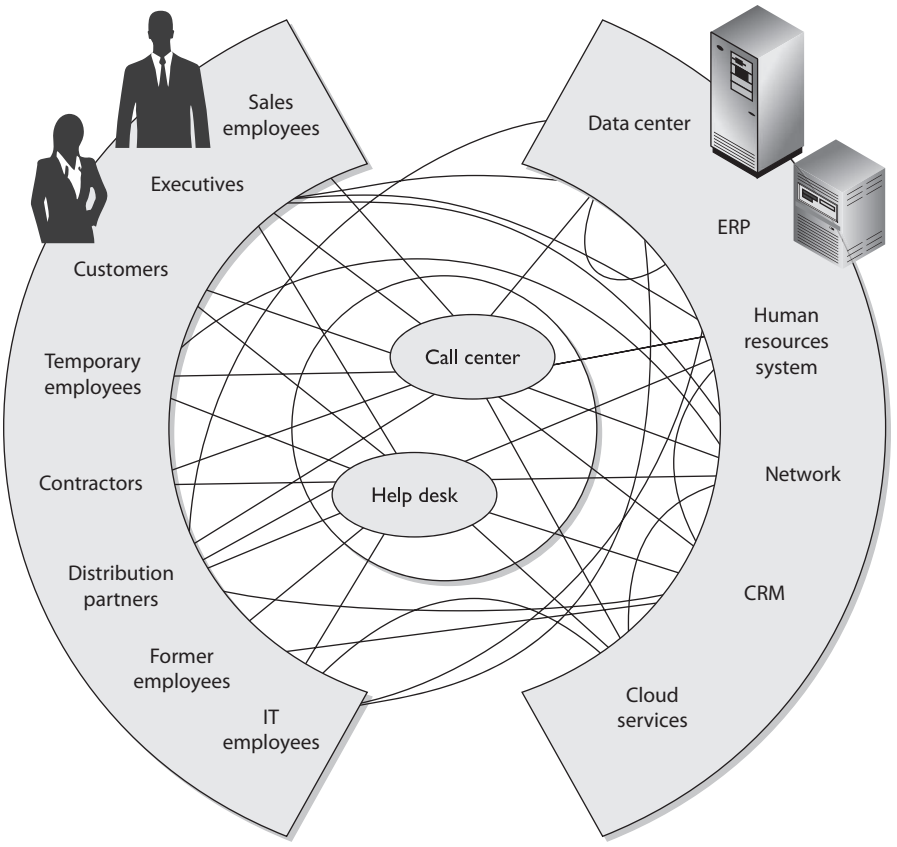


Figure 16-4 Most environments are complex in terms of access.

view of most environments.) 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 IdM technologies to simplify the administration of these tasks and bring order to chaos.

The following are some of the common questions enterprises deal with regarding IdM implementation:

- What should each user have access to?
- Who approves and allows access?
- How do the access decisions map to policies?
- Do former employees still have access?
- How do we keep up with our dynamic and ever-changing environment?

- What is the process of revoking access?
- How is access controlled and monitored centrally?
- Why do employees have eight passwords to remember?
- We have five different operating platforms. How do we centralize access when each platform (and application) requires its own type of credential set?
- How do we control access for our employees, customers, and partners?
- How do we make sure we are compliant with the necessary regulations?

The traditional identity management process has been manual, using directory services with permissions, access control lists (ACLs), and profiles. This labor-intensive approach has proven incapable of keeping up with complex demands and thus has been replaced with automated applications rich in functionality that work together to create an IdM infrastructure. The main goal of IdM technologies is to streamline the management of identity, authentication, authorization, and auditing of subjects on multiple systems throughout the enterprise. The sheer diversity of a heterogeneous enterprise makes proper implementation of IdM a huge undertaking.

Directory Services

Directory services, much like DNS, map resource names to their corresponding network addresses, allowing discovery of and communication with devices, files, users, or any other asset. 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.

Most organizations have some type of directory service that contains information pertaining to the organization's network resources and users. Most directories follow a hierarchical database format, originally established by the ITU X.500 standard but now most commonly implemented with the Lightweight Directory Access Protocol (LDAP), that allows subjects and applications to interact with the directory. Applications can request information about a particular user by making an LDAP request to the directory, and users can request information about a specific resource by using a similar request.

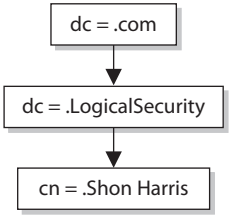
The objects within the directory are managed by a directory service. The directory service allows an administrator to configure and manage how identification, authentication, authorization, and access control take place within the network and on individual systems. The objects within the directory are labeled and identified with namespaces.

In a Windows Active Directory (AD) environment, when you log in, you are logging into a domain controller (DC), which has a hierarchical LDAP directory in its database. The database organizes the network resources and carries out user access control functionality. So once you successfully authenticate to the DC, certain network resources are available to you (print service, file server, e-mail server, and so on) as dictated by the configuration of AD.

How does the directory service keep all of these entities organized? By using *namespaces*. Each directory service has a way of identifying and naming the objects they manage. In LDAP, the directory service assigns distinguished names (DNs) to each object. Each DN

represents a collection of attributes about a specific object and is stored in the directory as an entry. In the following example, the DN is made up of a common name (cn) and domain components (dc). Since this is a hierarchical directory, .com is the top, LogicalSecurity is one step down from .com, and Shon is at the bottom.

```
dn: cn=Shon Harris,dc=LogicalSecurity,dc=com
cn: Shon Harris
```



This is a very simplistic example. Companies usually have large trees (directories) containing many levels and objects to represent different departments, roles, users, and resources.

A directory service manages the entries and data in the directory and also enforces the configured security policy by carrying out access control and identity management functions. For example, when you log into the DC, the directory service determines which resources you can and cannot access on the network.

Directories' Role in Identity Management

A directory service is a general-purpose resource that can be used for IdM. When used in this manner it is optimized for reading and searching operations and becomes the central component of an IdM solution. This is because all resource information, users' attributes, authorization profiles, roles, access control policies, and more are stored in this one location. When other IdM features need to carry out their functions (authorization, access control, assigning permissions), they now have a centralized location for all of the information they need.

A lot of the information that is catalogued in an IdM directory is scattered throughout the enterprise. User attribute information (employee status, job description, department, and so on) is usually stored in the HR database, authentication information could be in a Kerberos server, role and group identification information might be in a SQL database, and resource-oriented authentication information may be stored in Active Directory on a domain controller. These are commonly referred to as *identity stores* and are located in different places on the network.

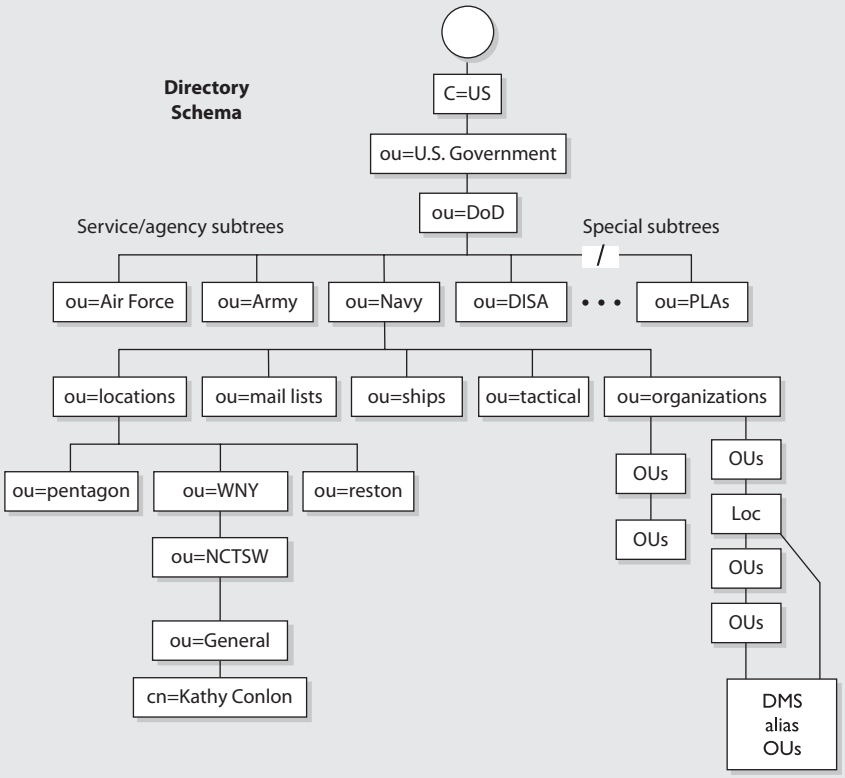
Something nifty that many IdM products do is create meta-directories or virtual directories. A *meta-directory* gathers the necessary information from multiple sources and stores it in one central directory. This provides a unified view of all users' digital identity information throughout the enterprise. The meta-directory synchronizes itself with all of the identity stores periodically to ensure the most up-to-date information is being used by all applications and IdM components within the enterprise.

Organizing All of This Stuff

In an LDAP system, the following rules are used for object organization:

- The directory has a tree structure to organize the entries using a parent-child configuration.
- Each entry has a unique name made up of attributes of a specific object.
- The attributes used in the directory are dictated by the defined schema.
- The unique identifiers are called distinguished names.

The schema describes the directory structure and what names can be used within the directory, among other things. The following diagram shows how an object (Kathy Conlon) can have the attributes of ou=General, ou=NCTSW, ou=WNY, ou=locations, ou=Navy, ou=DoD, ou=U.S. Government, and C=US. Kathy's distinguished name is made up by listing all of the nodes starting at the root of the tree (C=US) all the way to her leaf node (cn=Kathy Conlon), separated by commas.



Note that OU stands for organizational unit. OUs are used as containers of other similar OUs, users, and resources. CN stands for common name.

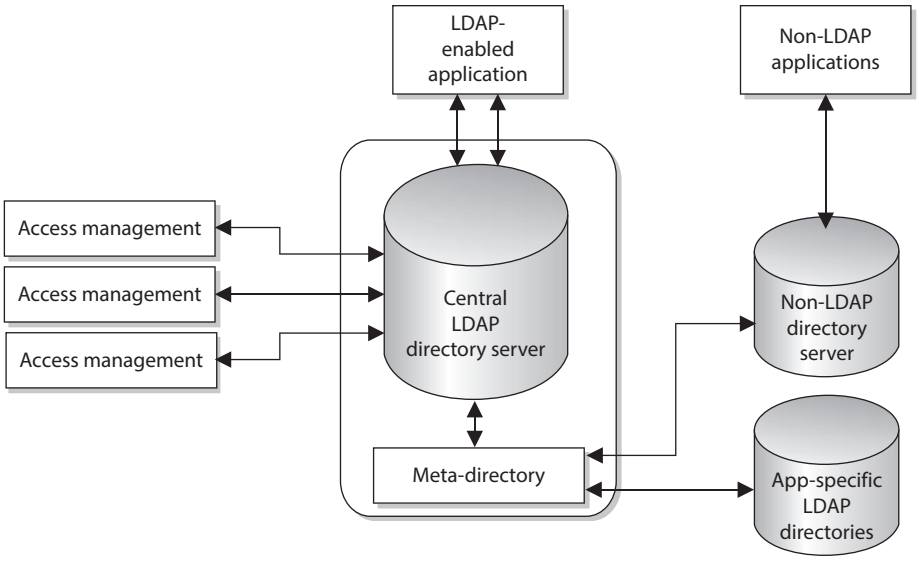


Figure 16-5 Meta-directories pull data from other sources to populate the IdM directory.

A *virtual directory* plays the same role and can be used instead of a meta-directory. The difference between the two is that the meta-directory physically has the identity data in its directory, whereas a virtual directory does not and points to where the actual data resides. When an IdM component makes a call to a virtual directory to gather identity information on a user, the virtual directory points to where the information actually lives.

Figure 16-5 illustrates a central LDAP directory that is used by the IdM services: access management, provisioning, and identity management. When one of these services accepts a request from a user or application, it pulls the necessary data from the directory to be able to fulfill the request. Since the data needed to properly fulfill these requests is stored in different locations, the metadata directory pulls the data from these other sources and updates the LDAP directory.

Single Sign-On

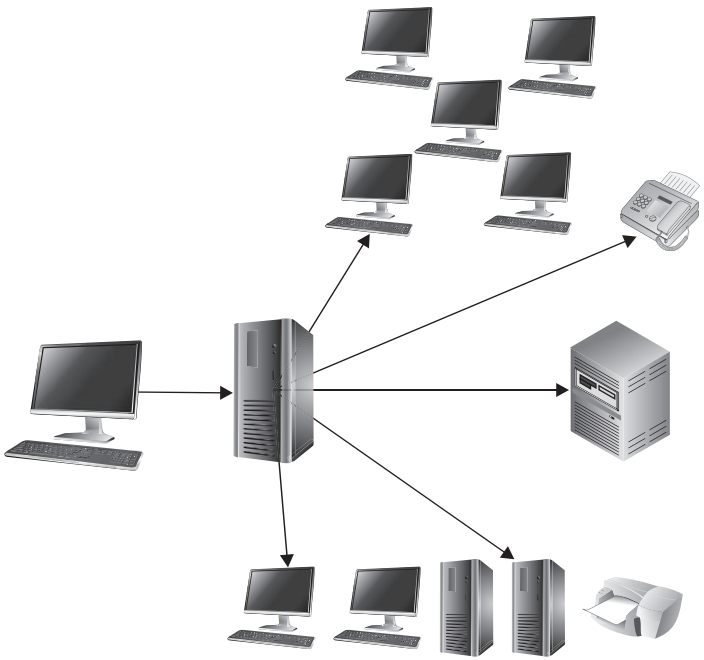
Employees typically need to access many different computers, servers, databases, and other resources in the course of a day to complete their tasks. This often requires the employees to remember multiple user IDs and passwords for these different computers. In a utopia, a user would need to enter only 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. Depending on the enterprise, between 20 percent and 50 percent of all IT help-desk calls are for password resets, according to the Gartner Group. Forrester Research estimates that each of these calls costs $70 in the United States. 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. 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 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 also 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.



Single sign-on technology enables a user to enter credentials one time to be able to access all preauthorized resources within the domain.

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 on to his Windows workstation and gets authenticated by a mixed-mode Windows domain controller, it must authenticate him to the resources he needs to access on the Apple MacBook, the Linux server running NIS, the PrinterLogic print server, and the Windows computer in a trusted domain that has the plotter connected to it. A nice idea, until reality hits.

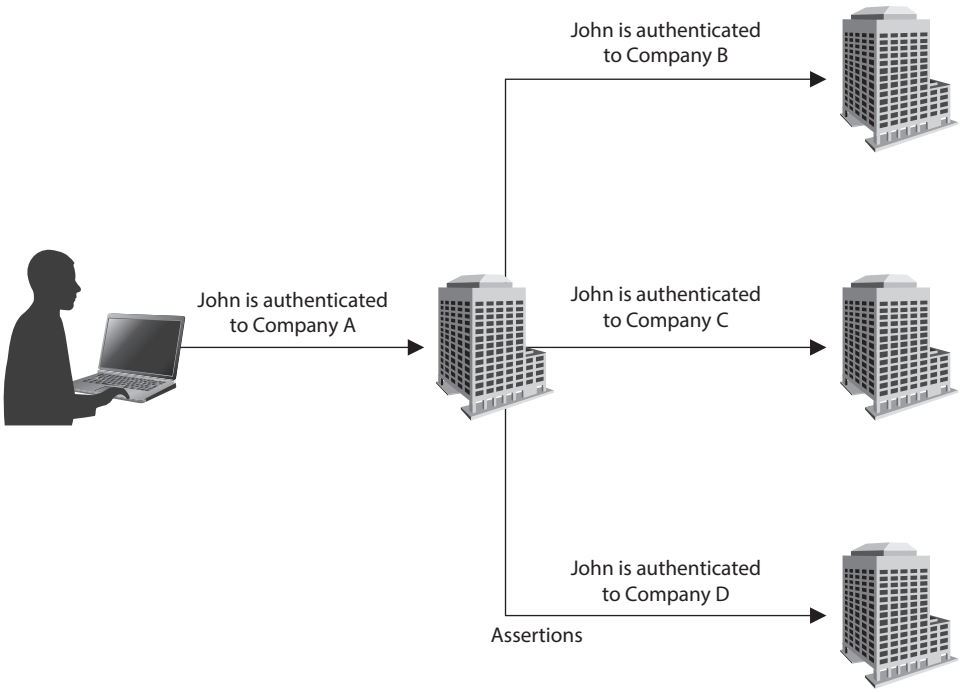
There is also a security issue to consider in an SSO environment. Once an individual is in, he is in. If an attacker is able to uncover one credential set, he has access to every resource within the environment that the compromised account has access to. This is certainly true, but one of the goals is that if a user only has to remember one password, and not ten, then a more robust password policy can be enforced. If the user has just one password to remember, then it can be more complicated and secure because he does not have nine other ones to remember also.

Federated Identity Management

The world continually gets smaller as technology brings people and companies closer together. Many times, when we are interacting with just one website, we are actually interacting with several different companies—we just don't know it. The reason we don't know it is because these companies are sharing our identity and authentication information behind the scenes. This is not done for nefarious purposes necessarily, but to make our lives easier and to allow merchants to sell their goods without much effort on our part.

For example, a person wants to book an airline flight and a hotel room. If the airline company and hotel company use a federated identity management (FIM) system, this means they have set up a trust relationship between the two companies and share customer identification and, potentially, authentication information. So when you book a flight on United Airlines, the website asks if you want to also book a hotel room. If you click Yes, you could then be brought to the Marriott website, which provides information on the closest hotel to the airport you're flying into. Now, to book a room you don't have to log in again. You logged in on the United website, and that website sent your information over to the Marriott website, all of which happened transparently to you.

A *federated identity* is a portable identity, and its associated entitlements, that can be used across business boundaries. It allows a user to be authenticated across multiple IT systems and enterprises. Identity federation is based upon linking a user's otherwise distinct identities at two or more locations without the need to synchronize or consolidate directory information. Federated identity offers businesses and consumers a more convenient way of accessing distributed resources and is a key component of e-commerce.



Web portal functions are parts of a website that act as a point of access to information. A portal presents information from diverse sources in a unified manner. It can offer various services, as in e-mail, news updates, stock prices, data access, price lookups, access to databases, and entertainment. Web portals provide a way for organizations to present one consistent interface with one "look and feel" and various functionality types. For example, you log into your company web portal and it provides access to many different systems and their functionalities, but it seems as though you are only interacting with one system because the interface is "clean" and organized. Portals combine web services (web-based functions) from several different entities and present them in one central website.

A web portal is made up of *portlets*, which are pluggable user-interface software components that present information from other systems. A portlet is an interactive application that provides a specific type of web service functionality (e-mail, news feed, weather updates, forums, etc.). A portal is made up of individual portlets to provide a plethora of services through one interface. It is a way of centrally providing a set of web services. Users can configure their view to the portal by enabling or disabling these various portlet functions.

Since each of these portlets can be provided by different entities, how user authentication information is handled must be tightly controlled, and there must be a

high level of trust between these different entities. A college, for example, might have one web portal available to students, parents, faculty members, and the public. The public should only be able to view and access a small subset of available portlets and not have access to more powerful web services (such as e-mail and database access). Students could be able to log in and gain access to their grades, assignments, and a student forum. Faculty members can gain access to all of these web services, including the school's e-mail service and access to the central database, which contains all of the students' information. If there is a software flaw or misconfiguration, it is possible that someone can gain access to something they are not supposed to.

Federated Identity with a Third-Party Service

It should not be surprising to consider that cloud service providers are also able to provide identification services. Identity as a Service (IDaaS) is a type of Software as a Service (SaaS) offering that is normally configured to provide SSO, FIM, and password management services. Though most IDaaS vendors are focused on cloud- and web-centric systems, it is also possible to leverage their products for FIM on legacy platforms within the enterprise network. Many organizations are transitioning to IDaaS providers for compliance reasons because this approach allows them to centralize access control and monitoring across the enterprise. This, in turn reduces risk and improves auditability, meaning there's a much lower chance of getting hit with a huge General Data Protection Regulation (GDPR) fine because some obscure part of the system didn't have proper access controls.

There are three basic approaches to architecting identity management services: on-premise, cloud-based, and a hybrid of both. The first approach, on-premise, is simple because all the systems and data are located within the enterprise. In the cloud-based model, on the other hand, most or all of the systems or data are hosted by an external party in the cloud. A hybrid FIM system includes both on-premise and cloud-based IdM components, each responsible for its environment but able to coordinate with each other. Regardless of the approach, it is important to ensure that all components play nice with each other. In the following sections we will explore some of the considerations that are common to the successful integration of these services.

Integration Issues

Integration of any set of different technologies or products is typically one of the most complex and risky phases of any deployment. In order to mitigate both the complexities and risks, it is necessary to carefully characterize each product or technology as well as the systems and networks into which they will be incorporated. Regardless of whether you ultimately use an on-premise or cloud-based (or hybrid) approach, you should carefully plan how you will address connectivity, trust, testing, and federation issues. As the old carpentry adage goes, "Measure twice and cut once."

Establishing Connectivity

A critical requirement is to ensure that the components are able to communicate with one another in a secure manner. The big difference between the in-house and outsourced models here is that in the former, the chokepoints are all internal to the organization's

network, while in the latter, they also exist in the public Internet. Clearing a path for this traffic typically means creating new rules for firewalls and IDS/IPS. These rules must be restrictive enough to allow the FIM traffic, but nothing else, to flow between the various nodes. Depending on the systems being used, ports, protocols, and user accounts may also need to be configured to enable bidirectional communication.

Establishing Trust

All traffic between nodes engaged in identity services must be encrypted. (To do otherwise would defeat the whole point of this effort.) From a practical perspective, this almost certainly means that PKI in general and certificate authorities (CAs) in particular will be needed. A potential issue here is that the CAs may not be trusted by default by all the nodes. This is especially true if the enterprise has implemented its own CA internally and is deploying an outsourced service. This is easy to plan ahead of time, but could lead to some big challenges if discovered during the actual rollout. Trust may also be needed between domains.

Incremental Testing

When dealing with complex systems, it is wise to assume that some important issue will not be covered in the plan. This is why it is important to incrementally test the integration of identity services instead of rolling out the entire system at once. Many organizations choose to roll out new services first to test accounts (i.e., not real users), then to one department or division that is used as the test case, and finally to the entire organization. For critical deployments (and one would assume that identity services would fall in this category), it is best to test as thoroughly as possible in a testbed or sandbox environment. Only then should the integration progress to real systems.

Legacy Systems

Unless your entire infrastructure is in the cloud, odds are that you have at least a handful of legacy systems that don't play nice with the FIM service or provider. To mitigate this risk, you should first ensure that you have an accurate asset inventory that clearly identifies any systems (or system dependencies) that will not integrate well. Then, you should get together with all stakeholders (e.g., business, IT, security, partners) to figure out which of these systems can be retired, replaced, or upgraded. The change management process we'll discuss in Chapter 20 is a great way to handle this. Finally, for any legacy systems that must remain as they are (and hence, not integrated into FIM), you want to minimize their authorized users and put additional controls in place to ensure they are monitored in an equivalent manner as the systems that fall under IdM.

On-Premise

An *on-premise* (or *on-premises*) FIM system is one in which all needed resources remain under your physical control. This usually means that you purchase or lease the necessary hardware, software, and licenses and then use your own team to build, integrate, and maintain the system. This kind of deployment, though rare, makes sense in cases where different organizations' networks are interconnected but not directly connected to the Internet, such as those of some critical infrastructure and military organizations. Though most

on-premise FIM solution providers offer installation, configuration, and support services, day-to-day operation and management of the system falls on your team. This requires them to have not only the needed expertise but also the time to devote to managing the system's life cycle.

Cloud

Arguably, the most cost-effective and secure way to implement FIM across an enterprise is to use a cloud-only solution. The economies of scale that IDaaS providers enjoy translate into cost savings for their customers. Even if you have the talent in your workforce to implement IdM on-premises, it would almost certainly be cheaper to outsource it to one of the many established vendors in this space. The visibility that an IDaaS provider has not only across your organization but also across the entire space of its customers allows it to detect and respond to threats faster and better than might otherwise be possible. This should be a dream come true, if only your entire infrastructure were cloud-based.

Hybrid

Most likely, your organization has a combination of cloud-based and on-premise systems. Some of the latter ones probably don't lend themselves to a cloud-based FIM solution, at least not without incurring exorbitant upgrade or integration costs. So, what should you do? You can implement a hybrid approach in which you have on-premise and cloud-based FIM platforms that are integrated with each other. One would be the primary and the other would be the secondary. As long as they are interoperable and properly configured, you get to have the best of both worlds. Most major IDaaS providers have solutions that support hybrid deployments.

Chapter Review

Identification, authentication, and authorization of users and systems are absolutely essential to cybersecurity. After all, how can we differentiate good and bad actors unless we know (at least) who the good ones are? This is why we spent so much time going over knowledge-based, biometric, and ownership-based authentication techniques and technologies. These, together with credential management products and practices, allow us to ensure we know who it is that our systems are interacting with.

The purpose of this chapter was to expose you to the multiple processes and technologies that make identity management possible, both at an individual level and at aggregate enterprise scales. This all sets the stage for the next chapter, in which we will delve into how to operationalize these concepts and build on them to ensure authorized parties (and no others) have access to the right assets (and no others).

Quick Review

- Identification describes a method by which a subject (user, program, or process) claims to have a specific identity (e.g., username, account number, or e-mail address).
- Authentication is the process by which a system verifies the identity of the subject, usually by requiring a piece of information that only the claimed identity should have.
- Credentials consist of an identification claim (e.g., username) and authentication information (e.g., password).
- Authorization is the determination of whether a subject has been given the necessary rights and privileges to carry out the requested actions.
- The three main types of factors used for authentication are something a person knows (e.g., password), something a person has (e.g., token), and something a person is (e.g., fingerprint), which can be combined with two additional factors: somewhere a person is (e.g., geolocation) and something a person does (e.g., keystroke behavior).
- Knowledge-based authentication uses information a person knows, such as a password, passphrase, or life experience.
- Salts are random values added to plaintext passwords prior to hashing to add more complexity and randomness.
- Cognitive passwords are fact- or opinion-based questions, typically based on life experiences, used to verify an individual's identity.
- A Type I biometric authentication error occurs when a legitimate individual is denied access; a Type II error occurs when an impostor is granted access.
- The crossover error rate (CER) of a biometric authentication system represents the point at which the false rejection rate (Type I errors) is equal to the false acceptance rate (Type II errors).
- Ownership-based authentication is based on something a person owns, such as a token device.
- A token device, or password generator, is usually a handheld device that has a display (and possibly a keypad), is synchronized in some manner with the authentication server, and displays to the user a one-time password (OTP).
- A synchronous token device requires the device and the authentication service to advance to the next OTP in sync with each other; an asynchronous token device employs a challenge/response scheme to authenticate the user.
- 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.

- Password managers or password vaults are a popular solution to remembering a myriad of complex passwords.
- Just-in-time (JIT) access is a provisioning methodology that elevates users to the necessary privileged access to perform a specific task.
- User provisioning refers to the creation, maintenance, and deactivation of user objects and attributes as they exist in one or more systems, directories, or applications, in response to business processes.
- An authoritative system of record (ASOR) is a hierarchical tree-like structure system that tracks subjects and their authorization chains.
- User provisioning refers to the creation, maintenance, and deactivation of user objects and attributes as they exist in one or more systems, directories, or applications, in response to business processes.
- A session is an agreement between two parties to communicate interactively.
- Auditing capabilities ensure users are accountable for their actions, verify that the security policies are enforced, and can be used as investigation tools.
- Deleting specific incriminating data within audit logs is called scrubbing.
- Identity management (IdM) is a broad term that encompasses the use of different products to identify, authenticate, and authorize users through automated means.
- Directory services map resource names to their corresponding network addresses, allowing discovery of and communication with devices, files, users, or any other asset.
- The most commonly implemented directory services, such as Microsoft Windows Active Directory (AD), implement the Lightweight Directory Access Protocol (LDAP).
- Single sign-on (SSO) systems allow users to authenticate once and be able to access all authorized resources, which reduces the amount of time users spend authenticating and enables administrators to streamline user accounts and better control access rights.
- A federated identity is a portable identity, and its associated entitlements, that allows a user to be authenticated across multiple IT systems and enterprises.
- Identity as a Service (IDaaS) is a type of Software as a Service (SaaS) offering that is normally configured to provide SSO, FIM, and password management services.
- There are three basic approaches to architecting identity management services: on-premise, cloud-based, and a hybrid of both.

Questions

Please remember that these questions are formatted and asked in a certain way for a reason. Keep in mind 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. Instead, the candidate should look for the best answer in the list.

1. Which of the following statements correctly describes biometric methods of authentication?
   - **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. Which of the following statements correctly describes the use of passwords for authentication?
   - **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.
3. How is a challenge/response protocol utilized with token device implementations?
   - **A.** This type of 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.
4. 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
5. What role does biometrics play in access control?
   - **A.** Authorization
   - **B.** Authenticity
   - **C.** Authentication
   - **D.** Accountability