Chapter 13
Managing Identity and Authentication

THE CISSP TOPICS COVERED IN THIS CHAPTER INCLUDE:

- ✓ **Domain 5.0: Identity and Access Management (IAM)**
  - 5.1 Control physical and logical access to assets
    - 5.1.1 Information
    - 5.1.2 Systems
    - 5.1.3 Devices
    - 5.1.4 Facilities
    - 5.1.5 Applications
    - 5.1.6 Services
  - 5.2 Design identification and authentication strategy (e.g., people, devices, and services)
    - 5.2.1 Groups and Roles
    - 5.2.2 Authentication, Authorization and Accounting (AAA) (e.g., multi-factor authentication (MFA), password-less authentication)
    - 5.2.3 Session management
    - 5.2.4 Registration, proofing, and establishment of identity
    - 5.2.5 Federated Identity Management (FIM)
    - 5.2.6 Credential management systems (e.g., Password vault)
    - 5.2.7 Single Sign On (SSO)
    - 5.2.8 Just-In-Time
  - 5.3 Federated identity with a third-party service
    - 5.3.1 On-premise
    - 5.3.2 Cloud

- 5.3.3 Hybrid
- 5.5 Manage the identity and access provisioning lifecycle
  - 5.5.1 Account access review (e.g., user, system, service)
  - 5.5.2 Provisioning and deprovisioning (e.g., on/off boarding and transfers)
  - 5.5.3 Role definition and transition (e.g., people assigned to new roles)
  - 5.5.5 Service accounts management

The Identity and Access Management (IAM) domain focuses on issues related to granting and revoking privileges to access data or perform actions on systems. A primary focus is on identification, authentication, authorization, and accounting. In this chapter and Chapter 14, "Controlling and Monitoring Access," we discuss all the objectives in the Identity and Access Management domain. Be sure to read and study the materials from both chapters to ensure complete coverage of this domain's essential material.

Controlling Access to Assets

Controlling access to assets is one of the central themes of security, and you'll find that many different security controls work together to provide access control. Note that assets can be tangible or intangible. Tangible assets refer to things you can touch, such as physical equipment, whereas intangible assets refer to information and data, such as intellectual property. In addition to personnel, technology assets can be information, systems, devices, facilities, applications, or services:

Information An organization's information includes all of its data. Data is stored in simple files on servers, computers, and smaller devices. It can also be stored in databases within a server farm or the cloud. It can even be paper records maintained in a

file cabinet. Logical access controls attempt to prevent unauthorized access to information.

Systems An organization's systems include any IT systems that provide one or more services. For example, a simple file server that stores user files is a system. Additionally, a web server working with a database server to provide an e-commerce service is a system. Permissions assigned to user and system accounts control system access.

Devices Devices refer to any computing equipment, including networking devices (routers and switches), storage devices (SAN and NAS), computing devices (servers, desktop computers, portable laptop computers, tablets, and smartphones), and external devices such as printers and scanners. Organizations have increasingly adopted policies allowing employees to connect their personally owned devices (such as smartphones or tablets) to an organization's network. Although the employees may own the devices, organizational data stored on the devices is still an asset of the organization.

Facilities An organization's facilities include any physical location that it owns or rents. This could be individual rooms, entire buildings, or whole complexes of several buildings. Physical security controls help protect facilities.

Applications Applications frequently provide access to an organization's data. Controlling access to applications provides an additional layer of control for the organization's data. Permissions are an easy way to restrict logical access to applications and be assigned to specific users or groups.

Services Services offered by an organization may include printing capabilities, network capacity, end-user support, and a variety of other offerings. Access control systems ensure that only authorized users gain access to these services.

Controlling Physical and Logical Access

In addition to understanding what assets need to be protected, you must know how to protect them. You can do so with physical

security controls and logical access controls.

Chapter 10, “Physical Security Requirements,” discusses physical security controls in depth. In general, a physical security control is one you can touch, such as perimeter security controls (fences, gates, guards, and turnstiles) and environmental controls such as heating, ventilation, and air-conditioning (HVAC) systems and fire suppression.

Physical security controls protect systems, devices, and facilities by controlling access and controlling the environment. As an example, organizations often have a server room where servers are running, and it's common for server rooms to include routers and switches. The benefit is that server rooms have increased security, such as cipher locks controlling entry into the server room. Desktop computers typically aren't as valuable as servers, but regular physical security controls such as locks provide protection.

Servers store important information (data), and also many servers host applications accessed by employees throughout the organization. These applications and data enjoy the same benefits from the other physical security controls protecting these servers.

Logical access controls are the technical controls used to protect access to information, systems, devices, and applications. They include authentication, authorization, and permissions. Combined, they help prevent unauthorized access to data and configuration settings on systems and other devices. For example, only people who can authenticate on a system or network can access data. Permissions help ensure only authorized entities can access data. Similarly, logical access controls restrict access to configuration settings on systems and network devices to only authorized individuals. Many of these logical access controls can apply to resources on-site or in the cloud.

The CIA Triad and Access Controls

One of the primary reasons an organization implements access control mechanisms is to prevent losses. There are three

categories of IT loss: loss of *confidentiality*, *integrity*, and *availability* (CIA). Protecting against these losses is so integral to IT security that they are frequently referred to as the *CIA Triad* (or sometimes the AIC Triad or Security Triad). Chapter 1, "Security Governance Through Principles and Policies," covers these in more depth. The following list identifies them in the context of access control:

Confidentiality Access controls help ensure that only authorized subjects can access objects. When unauthorized entities can access systems or data, it results in a loss of confidentiality.

Integrity Integrity ensures that data or system configurations are not modified without authorization, or if unauthorized changes occur, security controls detect the changes. If unauthorized or unwanted changes to objects occur, the result is a loss of integrity.

Availability Authorized requests for objects must be granted to subjects within a reasonable amount of time. In other words, systems and data should be available to users and other authorized subjects when they are needed. If the systems are not operational or the data is not accessible, the result is a loss of availability.

The AAA Model

The core functions of identity and access management systems are:

- *Authenticating* users, systems, services, and other subjects to confirm they are who they claim to be
- *Authorizing* actions attempted by those entities
- *Accounting* for activity by maintaining an audit trail

Together, these three core functions are described as the *AAA (or "Triple-A") model* of access control.

Identification and Authentication Strategy

Identification is the process of a subject claiming, or professing, an identity. A subject must provide an identity to a system to start the authentication, authorization, and accounting processes. Providing an identity might entail typing a username, swiping a smartcard, speaking a phrase, or positioning your face, hand, or finger in front of a camera or in proximity of a scanning device. A core identification principle is that all subjects must have unique identities.

Authentication verifies the subject's identity by comparing one or more factors against a database of valid identities, such as user accounts. The authentication information used to verify identity is private and needs to be protected. As an example, passwords are rarely stored in cleartext within a database. Instead, authentication systems store hashes of passwords in the authentication database.

Chapter 6, “Cryptography and Symmetric Key Algorithms,” covers hashing in more depth.

Identification and authentication occur together as a single two-step process. Providing an identity is the first step, and providing the authentication information is the second step. Without both, a subject cannot gain access to a system.

In contrast, imagine a user claims an identity (such as with a username of john.doe@sybex.com) but doesn't prove the identity (with a password). This username is for the employee named John Doe. However, if a system accepts the username without the password, it has no proof that the user is John Doe. Anyone who knows John's username can impersonate him.

Each authentication technique or factor has benefits and drawbacks. Thus, it is important to evaluate each mechanism in the context of the environment where it is deployed. For example, a facility that processes Top Secret materials requires very strong

authentication mechanisms. In contrast, authentication requirements for students within a classroom environment are significantly less.

While identification and authentication methods authenticate people, they also authenticate devices and services. The "Device Authentication" and "Service Authentication" sections, later in this chapter, explain devices and services in more depth.

You can simplify identification and authentication by thinking about a username and a password. Users identify themselves with usernames and authenticate (or prove their identity) with passwords. Of course, there are many more identification and authentication methods, but this simplification helps you keep the terms clear.

Comparing Subjects and Objects

Access control addresses more than just controlling which users can access which files or services. It is about the relationships between entities (subjects and objects). Access is the transfer of information from an object to a subject, which makes it important to understand the definition of both subject and object. Chapter 8, "Principles of Security Models, Design, and Capabilities," covers subjects and objects in more depth. The following provides a short reminder:

Subject A *subject* is an active entity that accesses a passive object to receive information from, or data about, an object. Subjects can be users, programs, processes, services, computers, or anything else that can access a resource. When authorized, subjects can modify objects.

Object An *object* is a passive entity that provides information to active subjects. Examples of objects are files, databases, computers, programs, processes, services, printers, and storage media.

You can often simplify the access control topics by substituting the word *user* for *subject* and the word *file* for *object*. For example, instead of *a subject accesses an object*, you can think of it as *a user accesses a file*. However, it's also important to remember that subjects include more than users and that objects include more than just files.

You may have noticed that some examples, such as programs, services, and computers, are listed as both subjects and objects. This is because the roles of subject and object can switch back and forth. In many cases, when two entities interact, they perform different functions. Sometimes they may be requesting information and other times providing information. The key difference is that the subject is always the active entity that receives information about, or data from, the passive object. The object is always the passive entity that provides or hosts the information or data.

As an example, consider a common web application that provides dynamic web pages to users. Users query the web application to retrieve a web page, so the application starts as an object. The web application then switches to a subject role as it queries the user's computer to retrieve a cookie and then queries a database to retrieve information about the user based on the cookie. Finally, the application switches back to an object as it sends dynamic web pages back to the user.

Registration, Proofing, and Establishment of Identity

Within an organization, new employees prove their identity with appropriate documentation during the hiring process. Acceptable documentation for in-person identity proofing includes using physical documents such as a passport, driver's license, birth certificate, and more. This documentation establishes the identity of the new employee for the employer.

After verifying the documents are authentic, employees within a human resources (HR) department begin the registration process. This process can be as simple as creating an account for the new employee and having the new employee set a password. If the organization uses more secure authentication methods, such as biometrics, the registration process is more complex. For example, if the organization uses fingerprinting as a biometric method for authentication, registration includes capturing the new employee's fingerprints.

Online organizations often use knowledge-based authentication (KBA) for identity proofing of someone new, such as a new customer. For example, if you create an online savings account, the bank will ask you a series of multiple-choice or fill-in-the-blank questions that only you should know. Here are a few examples:

- Which of the following vehicles have you recently purchased?
- How much is your car payment?
- How much is your mortgage (or rental) payment?
- Have you lived at any of the following addresses?
- What is your driver's license number?

The organization queries independent and authoritative sources, such as credit bureaus or government agencies, before creating these questions. It also gives users a limited amount of time to answer the questions.

Some organizations use a *cognitive password* (also known as security questions) when a known user is trying to change a password. Authentication systems collect the answers to these questions during the account's initial registration, but they can be collected or modified later. As an example, the subject might see the following questions when creating an account:

- What is your favorite sport?
- What is the color of your first car?

- What is the name of your first pet?
- What is the name of your first boss?
- What is your mother's maiden name?
- What is the name of your best friend in grade school?

Later, the system uses these questions for authentication. If the user answers all the questions correctly, the system authenticates the user. Cognitive passwords often assist with password management using self-service password reset systems or assisted password reset systems. For example, if users forget their original password, they can ask for help. The password management system then challenges the user with one or more of these cognitive password questions, presumably known only by the user.

One of the flaws associated with cognitive passwords is that the information is often available on social media sites or with internet searches. If a user includes some or all of the same information in an online profile, attackers may use the information to change the user's password. The National Institute of Standards and Technology's NIST SP 800-63B—Digital Identity Guidelines: Authentication and Life Cycle Management discourages using these static questions.

Authorization and Accounting

Two additional security elements in an access control system are *authorization* and *accounting*:

Authorization Subjects are granted access to objects based on proven identities. For example, administrators grant users access to files based on the user's proven identity.

Accounting Users and other subjects can be held accountable for their actions when auditing is implemented. Auditing tracks subjects and logs when they access objects, creating an audit trail

in one or more audit logs. For example, auditing can record when a user reads, modifies, or deletes a file. Auditing provides accountability.

Additionally, assuming the user has been properly authenticated, audit logs provide nonrepudiation. The user cannot believably deny doing something that is recorded in the audit logs.

An effective access control system requires strong identification and authentication mechanisms, in addition to authorization and accountability elements. Subjects have unique identities and prove their identity with authentication. Administrators grant access to subjects based on their identities, providing authorization. Logging user actions based on their proven identities provides accountability.

In contrast, if users didn't need to log on with credentials, then all users would be anonymous. It isn't possible to restrict authorization to specific users if everyone is anonymous. Logging could still record events, but it would not be able to identify which users performed any actions.

Authorization

Authorization indicates who is trusted to perform specific operations. If the action is allowed, the subject is authorized; if disallowed, the subject is not authorized. As a simple example, if a user attempts to open a file, the authorization mechanism checks to ensure that the user has at least read permission on the file.

It's important to realize that just because users or other entities can authenticate to a system, that doesn't mean they have access to anything and everything. Instead, subjects are authorized to access specific objects based on their proven identity. The process of authorization ensures that the requested activity or object access is possible based on the privileges assigned to the subject. Administrators grant users only the privileges they need to perform their jobs following the principle of least privilege.

Identification and authentication are “all-or-nothing” aspects of access control. Either a user's credentials prove a professed

identity, or they don't. In contrast, authorization occupies a wide range of variations. For example, a user may be able to read a file but not delete it, or they may be able to print a document but not alter the print queue.

Accounting

Auditing, logging, and monitoring provide accounting services by ensuring that subjects can be held accountable for their actions. Auditing is the process of tracking and recording subject activities within logs. Logs typically record who took an action, when and where the action was taken, and what the action was. One or more logs create an *audit trail* that researchers or investigators can use to reconstruct events and identify security incidents. When they review audit trails' contents, they can provide evidence to hold people accountable for their actions, such as violating security policy rules. These audit trails also help verify user compliance with policies.

There's a subtle but important point to stress about accountability. Accountability relies on effective identification and authentication, but it does not require effective authorization. In other words, after identifying and authenticating users, accountability mechanisms such as audit logs can track their activity, even when they try to access resources that they aren't authorized to access.

Authentication Factors Overview

There are three primary authentication factors:

Something You Know The *something you know* factor of authentication includes memorized secrets such as a password, personal identification number (PIN), or passphrase. Older documents refer to this as a *Type 1 authentication factor*.

Something You Have The *something you have* factor of authentication includes physical objects that a user possesses and can help them provide authentication. Examples include a smartcard, hardware token, smartphone running an

authentication application, or Universal Serial Bus (USB) drive. Older documents refer to this as a *Type 2 authentication factor*.

Something You Are The *something you are* factor of authentication uses physical characteristics of a person and is based on biometrics. Examples in the something you are category include fingerprints, face scans, retina patterns, iris patterns, palm scans, and voice pattern recognition. Older documents refer to this as a *Type 3 authentication factor*.

Single-factor authentication uses only one authentication factor. Multifactor authentication uses two or more authentication factors.

These types are progressively stronger when implemented correctly, with something you know being the weakest and something you are the strongest. In other words, passwords are the weakest form of authentication, and a fingerprint is stronger than a password. However, attackers can still bypass some biometric authentication factors. For example, an attacker can create a duplicate, or counterfeit, fingerprint on a gummy bear candy and fool a fingerprint reader.

In addition to the three primary authentication factors, attributes are sometimes used for additional authentication. These include the following:

Somewhere You Are The somewhere you are factor identifies a subject's location based on a specific computer or device, a geographic location identified by an Internet Protocol (IP) address, or a phone number identified by Caller ID. Controlling access by physical location forces a subject to be present somewhere. Geolocation technologies can identify a user's location based on the IP address, and some authentication systems use geolocation.

Somewhere You Aren't

Many IAM systems use geolocation technologies to identify suspicious activity. For example, imagine that a user typically logs on with an IP address in Virginia Beach. If the IAM detects a user trying to log on to the same account from India, it can block the access even if the user has the correct username and password. This isn't 100 percent reliable, though. A dedicated overseas attacker can use online virtual private network (VPN) services to change the IP address used to connect with an online server.

Context-Aware Authentication Many mobile device management (MDM) systems use *context-aware authentication* to identify mobile device users. It can identify multiple attributes such as the user's location, the time of day, and the mobile device. Organizations frequently allow users to access a network with a mobile device, and MDM systems can detect details on the device when a user attempts to log on. If the user meets all the requirements (location, time, and type of device in this example), it allows the user to log on using the other methods, such as with a username and password.

Many mobile devices support the use of gestures or finger swipes on a touchscreen. As an example, Microsoft Windows 11 supports picture passwords, allowing users to authenticate by moving their fingers across the screen using a picture of their choice. Similarly, Android devices support Android Lock, allowing users to swipe the screen connecting dots on a grid. These methods are sometimes referred to as something you do.

Something You Know

The most common authentication technique is the *password*, a string of characters entered by a user. Passwords are typically static. A *static password* stays the same for a length of time, such as 60 days, but static passwords are the weakest form of

authentication. Passwords are weak security mechanisms for several reasons:

- Users often choose passwords that are easy to remember and, therefore, easy to guess or crack.
- Randomly generated passwords are hard to remember, causing many users to write them down.
- Users often share their passwords or forget them.
- Attackers detect passwords through many means, including observation, sniffing networks, and stealing databases.
- Passwords are sometimes transmitted in cleartext or with easily broken encryption protocols. Attackers can capture these passwords with network sniffers.
- Password databases are sometimes stored in publicly accessible online locations.
- Passwords are subject to many types of attack, including brute-force guessing, dictionary attacks, password spraying, credential stuffing, and others. You'll learn about these attacks in Chapter 14.

One way of strengthening a password is by using a *passphrase*. This is a string of characters similar to a password but has a unique meaning to the user. As an example, a passphrase can be "I earned my CISSP certification." Many authentication systems do not support spaces, so this passphrase can be modified to "IEarnedMyCISSPCertification."

Using a passphrase has several benefits. It is easy to remember, and it encourages users to create longer passwords. Longer passwords are more difficult to crack using a brute-force tool. Encouraging users to create passphrases also helps ensure that they don't use common, predictable passwords such as "password" and "123456."

Personal identification numbers (PINs) are also in the something you know category. PINs are typically four, six, or eight numbers long.

IT personnel have been trying to force users into creating and maintaining secure passwords using password policies. However, users always seem to find a way around these policies, creating passwords that attackers can easily crack. As a result, security personnel often seek new solutions. The following sections identify several basic password policy components, followed by some of the recommendations by different entities.

Password Policy Components

Organizations often include a written *password policy* in the overall security policy. IT security professionals then enforce the policy with technical controls such as a technical password policy that enforces the password restriction requirements. The following list includes some common password policy settings:

Maximum Age This setting requires users to change their password periodically, such as every 45 days. Some documents refer to this as password expiration.

Password Complexity Password complexity refers to how many character types it includes. The different character types are lowercase letters, uppercase letters, numbers, and special characters. A simple password, such as 123456789, contains only one character type (numbers). Complex passwords use three or four character types.

Password Length The length is the number of characters in the password, such as at least eight characters long. When using the same character types in a password, shorter passwords are easier to crack and longer passwords are harder to crack.

Minimum Age This setting prevents users from changing their password again until a certain time has passed. Password policies enforcing password history typically have a minimum age of one day.

Password History Many users get into the habit of rotating between two passwords. A password history remembers a certain number of previous passwords and prevents users from reusing passwords. Combined with a minimum age of one or more days,

it prevents users from changing their password multiple times in one sitting until they return to their original password.

Authoritative Password Recommendations

Password recommendations are changing, and so far, there isn't a consensus that everyone is following. Depending on what source you use, you'll find different suggestions for passwords. Several authoritative sources are worth mentioning. All of these sources are updated regularly, but the following versions were active when this book was published:

- NIST SP 800-63B—Digital Identity Guidelines: Authentication and Life Cycle Management
- Payment Card Industry Data Security Standard (PCI DSS) version 4.0

Chapter 4, "Laws, Regulations, and Compliance," covers PCI DSS in more depth.

NIST Password Recommendations

NIST SP 800-63B provided new recommendations on passwords that are quite different from past recommendations. The following list summarizes the changes recommended by NIST:

Passwords must be hashed. Passwords should never be stored or transmitted in cleartext.

Passwords should not expire. Users should not be required to change their passwords regularly, such as every 30 days. Users often changed a single character when forced to change their password. For example, they would change Password1 to Password2. Although this complies with the requirement to change the password, it doesn't add to security. Attackers use the same methods when guessing passwords. Users should only be forced to change their password if there is evidence that their current password was compromised.

Users should not be required to use special characters. Requiring users to include special characters often challenged users' memory, and they wrote these passwords down. Further, NIST analyzed breached password databases and discovered that special characters in passwords didn't provide the desired benefits.

Users should be able to copy and paste passwords. Password managers allow users to create and store complex passwords. Users enter one password into the password manager to access stored passwords. They can then copy passwords from the password manager and paste passwords into the password text box. When copy and paste is restricted, users must retype the password and typically default to easier passwords.

Users should be able to use all characters. Password storage mechanisms have commonly rejected spaces and some special characters. By allowing spaces, users can create longer passwords that are easier to remember. Systems sometimes reject special characters to prevent attacks (such as a SQL injection attack), but properly hashing the password masks these characters.

Password length should be at least eight characters and as many as 64 characters. A longer length allows users to create passphrases that are meaningful to them.

Password systems should screen passwords. Before accepting a password, password systems should check them against a list of commonly used passwords, such as 123456 or password.

PCI DSS Password Requirements

The PCI DSS (version 4.0) has the following requirements, which differ from NIST SP 800-63B:

- Passwords expire at least every 90 days.
- Passwords must be at least 12 characters long.
- Passwords must contain both numeric and alphabetic characters.

- Passwords may not be the same as any of the user's previous four passwords.

If organizations need to comply with a specific standard, such as PCI DSS, they should follow at least the minimum requirements from that standard.

Something You Have

Smartcards and hardware tokens are both examples of the Type 2, or something you have, factor of authentication. They are rarely used by themselves but are commonly combined with another authentication factor, providing multifactor authentication.

Smartcards

A *smartcard* is a credit card–sized ID or badge and has an integrated circuit chip embedded in it. Smartcards contain information about the authorized user that is used for identification and/or authentication purposes. Most current smartcards include a microprocessor and one or more certificates. The certificates are used for asymmetric cryptography such as encrypting data or digitally signing emails, as discussed in Chapter 7, "PKI and Cryptographic Applications." Smartcards are tamper-resistant and provide users with an easy way to carry and use complex encryption keys.

Users insert the card into a smartcard reader when authenticating. It's common to require users to also enter a PIN or password as a second authentication factor with the smartcard.

Note that smartcards can provide both identification and authentication. However, because users can share or swap smartcards, they aren't effective identification methods by themselves. Most implementations require users to use another authentication factor, such as a PIN or username and password.

Authenticators

A *device authenticator*, or token, is an authentication secret-generating device or application that users can carry with them. Common authenticators include a display showing a six- to eight-digit number, known as the *one-time password (OTP)*. An authentication server stores the details of the authenticator, so at any moment, the server knows what number is displayed on the user's authenticator.

Authenticators are typically combined with another authentication mechanism. For example, users might enter a username and password (in the something you know factor of authentication) and then enter the number displayed on the authenticator (in the something you have factor of authentication). This provides multifactor authentication.

Figure 13.1 shows an example of using a dedicated hardware device from RSA as an authenticator. Figure 13.2 shows an example of using Google Authenticator as a software-based authenticator running on a smartphone.

Source: Kevin/Adobe Stock Photos

FIGURE 13.1 Hardware authenticator

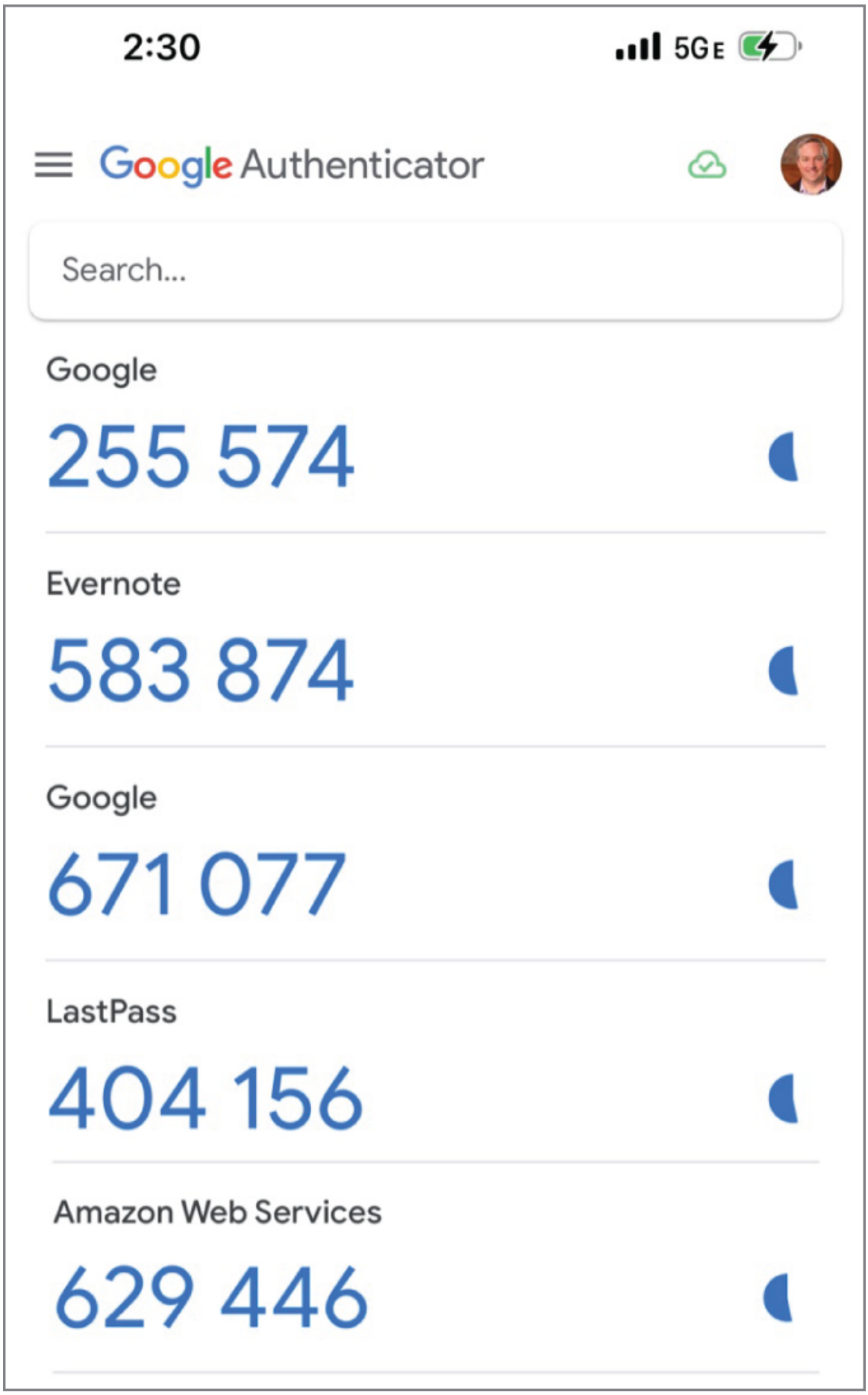
2:30
5GE
Google Authenticator
Search...
Google
255 574
Evernote
583 874
Google
671 077
LastPass
404 156
Amazon Web Services
629 446

FIGURE 13.2 Software authenticator

Each authenticator uses one of two different techniques to generate one-time passwords:

Time-Based One-Time Passwords *Time-based one-time passwords (TOTPs)* are generated by devices and applications that are synchronized with an authentication server. They generate a new OTP periodically, such as every 60 seconds. This requires the authenticator and the server to have accurate and synchronized clocks. For this reason, TOTP approaches are also known as synchronous authenticators.

Hash-Based One-Time Passwords *HMAC-based one-time passwords (HOTP)* do not use a clock. Instead, the hardware authenticator generates OTPs based on an algorithm and an incrementing counter. When using an incrementing counter, the user clicks a button, causing the authenticator to create a dynamic one-time password that stays the same until it is used for authentication. For this reason, HOTP approaches are also known as asynchronous authenticators.

Hardware authenticators provide strong authentication, but they do have failings. If the battery dies or the device breaks or is lost, the user won't be able to gain access to services requiring their use.

Something You Are

Another common authentication and identification technique is the use of *biometrics*. *Biometric factors* fall into the Type 3, something you are, authentication category.

Biometric factors can be used as an identifying technique, an authentication technique, or both. They do not provide authorization or accountability. Using a biometric factor instead of a username or account ID as an identification factor requires a search of the offered biometric pattern against a stored database of enrolled and authorized patterns.

Using a biometric factor as an authentication technique requires a one-to-one match of the offered biometric pattern against a

stored pattern for the claimed subject identity. In other words, the user claims an identity, and the authentication system checks the biometric factor to see if the person matches the claimed identity.

Physiological biometric methods include fingerprints, face scans, retina scans, iris scans, palm scans (also known as palm topography or palm geography), and voice patterns:

Fingerprints *Fingerprints* are the visible patterns on the fingers and thumbs of people. They are unique to an individual and have been used for decades in physical security for identification. Fingerprints have loops, whorls, ridges, and bifurcations (also called minutiae), and fingerprint readers match the minutiae to data within a database. Fingerprint readers are commonly used on laptop computers, keyboards, mice, security keys, and USB flash drives to identify and authenticate users. It usually takes less than a minute to capture a user's fingerprint during the registration process.

Face Scans *Face scans* use the geometric patterns of faces for detection and recognition. Many smartphone, tablet, and computer operating systems support face identification to unlock the device. Casinos use it to identify card cheats. Law enforcement agencies have been using it to catch criminals at borders and in airports. Face scans are also used to identify and authenticate people before allowing them to access secure spaces such as a secure vault.

Retina Scans *Retina scans* focus on the pattern of blood vessels at the back of the eye. They are the most accurate form of biometric authentication and can differentiate between identical twins. However, some privacy proponents object to their use because they can reveal medical conditions, such as high blood pressure and pregnancy. Additionally, retina scanners typically require users to be as close as three inches from the scanner.

Iris Scans Focusing on the colored area around the pupil, *iris scans* are the second-most accurate form of biometric authentication. Like the retina, the iris remains relatively unchanged throughout a person's life (barring eye damage or

illness). Users consider iris scans less intrusive than retina scans because scans can occur from distances of 20 to 40 feet. However, some scanners can be fooled with a high-quality image in place of a person's eye. Additionally, the accuracy of iris scans may be affected by changes in lighting and the usage of some glasses and contact lenses.

Palm Scans *Palm scanners* scan the palm of the hand for identification. They use near-infrared light to measure vein patterns in the palm, which are as unique as fingerprints. Individuals simply place their palm over a scanner for a few seconds during the registration process. Later, they place their hand over the scanner again for identification. For example, some testing providers use palm vein readers to prevent people from taking exams for others and ensure that the same person reenters the testing room after a break.

Voice Pattern Recognition This type of biometric authentication relies on the characteristics of a person's speaking voice, known as a *voiceprint*. The user speaks a specific phrase, which is recorded by the authentication system. To authenticate, they repeat the same phrase, and it is compared to the original. *Voice pattern* recognition is sometimes used as an additional authentication mechanism but is rarely used by itself.

Speech recognition is commonly confused with voice pattern recognition, but they are different. Speech recognition software, such as dictation software, extracts communications from sound. In other words, voice pattern recognition differentiates between one voice and another for identification or authentication, whereas speech recognition differentiates between words with any person's voice.

The use of biometrics promises universally unique identification for every person on the planet. Unfortunately, biometric technology has yet to live up to this promise. However,

technologies that focus on physical characteristics are very useful for authentication.

Biometric Factor Error Ratings

The most important aspect of a biometric device is its accuracy. When using biometrics for identification, a biometric device must detect minute differences in information, such as variations in the blood vessels in a person's retina or differences in a person's veins in their palm. Because most people are similar, biometric methods often result in false negative and false positive authentications. Biometric devices are rated for performance by examining the different types of errors they produce:

False Rejection Rate A false rejection occurs when an authentication system does not authenticate a valid user. As an example, say Dawn has registered her fingerprint and used it for authentication previously. Imagine that she uses her fingerprint to authenticate herself today, but the system incorrectly rejects her fingerprint, indicating it isn't valid. This is sometimes called a false negative authentication. The ratio of false rejections to valid authentications is known as the *false rejection rate (FRR)*. False rejection is sometimes called a *Type I error*.

False Acceptance Rate A false acceptance occurs when an authentication system authenticates someone incorrectly. This is also known as a false positive authentication. As an example, imagine that Joe doesn't have an account and hasn't registered his fingerprint. However, he uses his fingerprint to authenticate, and the system recognizes him. This is a false positive or a false acceptance. The ratio of false positives to valid authentications is the *false acceptance rate (FAR)*. False acceptance is sometimes called a *Type II error*.

Most biometric devices have a sensitivity adjustment. When a biometric device is too sensitive, false rejections (false negatives) are more common. When a biometric device is not sensitive enough, false acceptances (false positives) are more common.

You can compare the overall quality of biometric devices with the *crossover error rate (CER)*, also known as the equal error rate

(ERR). Figure 13.3 shows the FRR and FAR percentages when a device is set to different sensitivity levels. The point where the FRR and FAR percentages are equal is the CER, and the CER is used as a standard assessment value to compare the accuracy of different biometric devices. Devices with lower CERs are more accurate than devices with higher CERs.

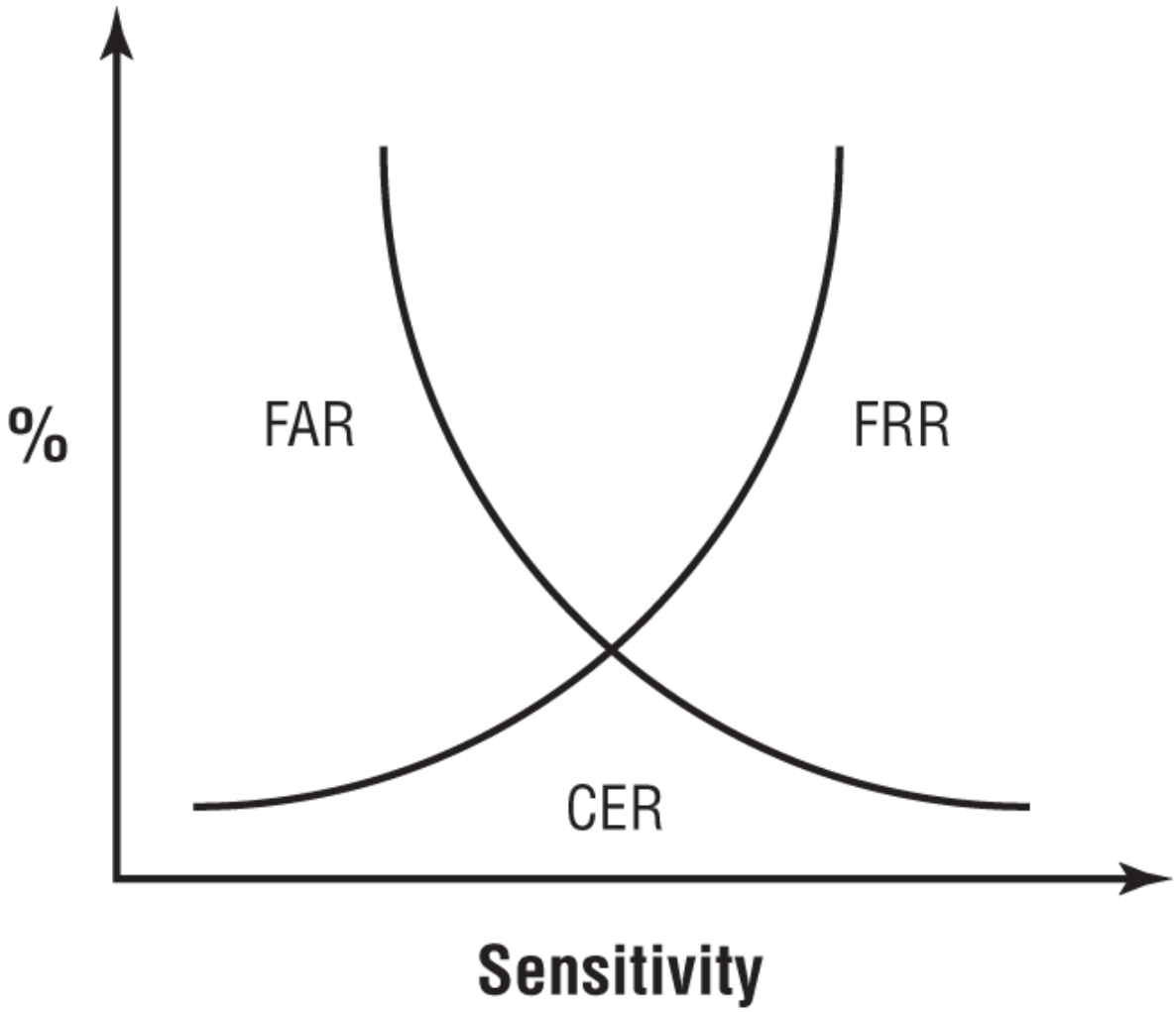

FIGURE 13.3 Graph of FRR and FAR errors indicating the CER point

It's not necessary, and often not desirable, to operate a device with the sensitivity set at the CER level. For example, an organization may use a facial recognition system to allow or deny access to a secure area because they want to ensure that unauthorized individuals are never granted access. In this case, the organization would set the sensitivity very high, so there is little chance of a false acceptance (false positive). This may result

in more false rejections (false negatives), but a false rejection is more acceptable than a false acceptance in this scenario.

Biometric Registration

Biometric devices can be ineffective or unacceptable due to factors known as enrollment time, throughput rate, and acceptance. For a biometric device to work as an identification or authentication mechanism, enrollment (or registration) must occur. During enrollment, a subject's biometric factor is sampled and stored in the device's database. This stored sample of a biometric factor is the *reference profile* (also known as a *reference template*).

The time required to scan and store a biometric factor depends on which physical or performance characteristic is measured. Users are less willing to accept the inconvenience of biometric methods that take a long time. In general, enrollment times over 2 minutes are unacceptable. If you use a biometric characteristic that changes over time, such as a person's voice tones, facial hair, or signature pattern, users must enroll again at regular intervals, adding an inconvenience.

The *throughput rate* is the amount of time the system requires to scan a subject and approve or deny access. The more complex or detailed a biometric characteristic, the longer processing takes. Subjects typically accept a throughput rate of about 6 seconds or faster.

Multifactor Authentication (MFA)

Multifactor authentication (MFA) is any authentication using two or more factors. *Two-factor authentication (2FA)* requires two different proofs of identity to provide authentication. In contrast, any authentication method using only one factor is *single-factor authentication*. For example, smartcards typically require users to insert their card into a reader and enter a PIN. The smartcard is in the something you have factor, and the PIN is in the something you know factor. As a general rule, additional factors result in more secure authentication.

Multifactor authentication must use multiple types or factors, such as the something you know factor and the something you have factor. In contrast, requiring users to enter a password and a PIN is not multifactor authentication because both methods are from a single authentication factor (something you know).

When two authentication methods of the same factor are used together, the authentication strength is no greater than it would be if just one method was used because the same attack that could steal or obtain one could also obtain the other. For example, using two passwords together is no more secure than using a single password because a password-cracking attempt could discover both in a single successful attack.

In contrast, when two or more different factors are employed, two or more different attack methods must succeed to collect all relevant authentication elements. For example, suppose a token, a password, and a biometric factor are all used for authentication. In that case, a physical theft, a password crack, and a biometric duplication attack must all succeed simultaneously to allow an intruder to gain entry into the system.

NIST Deprecates SMS for 2FA

Another method of two-factor authentication uses the Short Message Service (SMS) to send users a text message with the OTP. This method is better than just using a password, but it has problems. NIST SP 800-63B has pointed out several vulnerabilities with using SMS for two-step authentication and deprecated its use for federal agencies.

Smartphones and tablets display texts on the lock screen without the user logging on. If an attacker stole the smartphone or tablet, they would have access to the OTP sent via SMS.

Attackers may be able to convince a mobile operator to redirect SMS messages to an attacker's devices. This is sometimes possible via subscriber identity module (SIM) card fraud. If successful, attackers may be able to intercept SMS messages.

Passwordless Authentication

There is a growing trend toward passwordless authentication. As mentioned previously, static passwords are the weakest form of authentication. Worse, as IT departments attempt to force users into creating longer and more complex passwords with expiration dates, users engage in risky behavior such as writing their passwords down or creating weaker passwords that are easier to remember.

Passwordless authentication allows users to log into systems without entering a password (or any other memorized secret). As an example, many smartphones and tablets support biometric authentication. If you've enabled facial recognition on your smartphone, all you need to do is look at it to get beyond the login screen. Similarly, if you've enabled fingerprint recognition on a tablet, all you need to do is place your finger on the sensor.

Once you get past the logon screen, many internal applications use the same authentication methods to access sensitive data. As an example, imagine you use an app on a tablet to access an online bank. The first time you access it, the app prompts you to save your credentials, and you agree. The next time you access the app, the app prompts you to authenticate with your fingerprint again.

The Fast Identity Online (FIDO) Alliance is an open industry association with a stated mission of reducing the over-reliance on passwords. FIDO has created recommended frameworks and protocol standards for passwordless authentication. These revolve around the use of hardware passkeys, such as the YubiKey device shown in Figure 13.4.

Source: gguy / Adobe Systems Incorporated

FIGURE 13.4 YubiKey passkey

Device Authentication

Historically, users have only been able to log into a network from a company-owned system such as a desktop PC. For example, in a Windows domain, user computers join the domain and have

computer accounts (sometimes called system accounts) and passwords similar to user accounts and passwords. If the computer hasn't joined the domain, or its credentials are out of sync with a domain controller, users cannot log on from the computer.

Today, more and more employees are bringing their own mobile devices to work and hooking them up to the network. Some organizations embrace this but implement security policies as a measure of control. These devices aren't necessarily able to join a domain, but it is possible to implement device identification and authentication methods.

One method is device fingerprinting. Users can register their devices with the organization and associate them with their user accounts. During registration, a device authentication system captures the characteristics of the device. This is often accomplished by having the user access a web page with the device. The registration system then identifies the device using attributes such as the operating system and version, web browser, browser fonts, browser plug-ins, time zone, data storage, screen resolution, cookie settings, and HTTP headers.

When the user logs on from the device, the authentication system checks the user account for a registered device. It then verifies the characteristics of the user's device with the registered device. Even though some of these characteristics change over time, this has proven to be a successful device authentication method.

As mentioned previously, many MDM systems use context-aware authentication methods to identify devices. They typically work with network access control (NAC) systems to check the device's health and grant or restrict access based on requirements configured within the NAC system.

802.1X is another method used for device authentication. It can be used for port-based authentication on some routers and switches. Additionally, it is often used with wireless systems, forcing users to log on with an account before being granted access to a network. Many MDM and NAC solutions implement

802.1X solutions to control user access from mobile devices. If the device or user cannot authenticate through the 802.1X system, they cannot access the network.

Service Authentication

Many services also require authentication, and they typically use a username and password. A service account is simply a user account that an administrator created for a service or application instead of a person.

As an example, it's common to create a service account for third-party tools monitoring email in Microsoft's Exchange Server. These third-party tools typically need permission to scan all mailboxes looking for spam, malware, potential data exfiltration attempts, and more. Administrators create a Microsoft domain account and give the account the necessary privileges to perform the tasks.

Some applications have built-in service accounts. For example, Microsoft's SQL Server has a built-in account known as the sa (short for system administrator) account. It is a member of the sysadmin fixed server role and has unlimited permissions on the SQL instance. It's only enabled if the instance is configured for SQL Server Authentication. In older versions, the default was a blank password, and attackers frequently check to see if the account is enabled and if it has a blank or weak password.

It's common to set the properties of the account so that the password never expires. For a regular user, you'd set the maximum password age to something like 45 days. When the password expires, the system informs the user to change the password, and the user does so. However, a service can't respond to such a message and instead is just locked out.

Because a service account has a high level of privileges, administrators configure it with a strong, complex password that is changed more often than regular users. However, administrators need to change these passwords manually. The longer a password remains the same, the more likely it will be compromised. Account access reviews can detect security issues

for service accounts. Another option is to configure the account to be noninteractive, which prevents a user from logging onto the account using traditional logon methods.

Services can be configured to use certificate-based authentication. Certificates are issued to the device running the service and presented by the service when accessing resources. Web-based services often use application programming interface (API) methods to exchange information between systems. These API methods are different depending on the web-based service. As an example, Google and Facebook provide web-based services that web developers use, but they use different implementations.

Mutual Authentication

There are many occasions when mutual authentication is needed. As an example, when a client accesses a server, both the client and the server provide authentication. This prevents a client from revealing information to a rogue server. Mutual authentication methods commonly use digital certificates.

For example, when employees are connecting to a company network while working from home, they typically connect to a virtual private network (VPN) server. Both the server and the client present digital certificates to the other endpoint, providing two-way authentication. If this mutual authentication fails, the two endpoints don't start a communication session. If an attacker redirected the traffic to a rogue VPN server, the authentication would fail, and the employee would know not to enter credentials.

Implementing Identity Management

Identity management (IdM) implementation techniques generally fall into two categories:

- *Centralized access control* implies that a single entity within a system performs all authorization verification.
- *Decentralized access control* (also known as distributed access control) implies that various entities located

throughout a system perform authorization verification.

A small team or individual can manage centralized access control. Administrative overhead is lower because all changes are made in a single location, and a single change affects the entire system. However, a vulnerability is that centralized access control potentially creates a single point of failure.

Another benefit of centralized identity management solutions is that they can scale up to support more users. For example, a Microsoft Active Directory domain can start with just a single domain controller. As the company grows, administrators can add additional domain controllers to handle the additional traffic.

Decentralized access control often requires several teams or multiple individuals. Administrative overhead is higher because changes must be implemented across numerous locations. Maintaining consistency across a system becomes more difficult as the number of access control points increases. Changes made to any individual access control point need to be repeated at every access point.

Single Sign-On

Single sign-on (SSO) is a centralized access control technique that allows a subject to be authenticated once on a system and access multiple resources without authenticating again. SSO is convenient for users, and it also has security benefits. When users have to remember multiple usernames and passwords, they often resort to writing them down, ultimately weakening security. Users are less likely to write down a single password. SSO also eases administration by reducing the number of accounts required for a subject.

The primary disadvantage to SSO is that once an account is compromised, an attacker gains unrestricted access to all of the authorized resources. However, most SSO systems include methods to protect user credentials. The following sections discuss several common SSO mechanisms.

LDAP and Centralized Access Control

Within a single organization, a centralized access control system is often used for SSO. For example, a *directory service* is a centralized database that includes information about subjects and objects, including authentication data. Many directory services are based on the Lightweight Directory Access Protocol (LDAP). For example, the Microsoft Active Directory Domain Services (AD DS) is an LDAP-based directory.

You can think of an LDAP directory as a telephone directory for network services and assets. Users, clients, and processes can search the directory service to find where a desired system or resource resides. Subjects must authenticate to the directory service before performing queries and lookup activities. Even after authentication, the directory service will reveal only certain information to a subject, based on its assigned privileges.

Multiple domains and trusts are commonly used in access control systems. A security domain is a collection of subjects and objects that share a common security policy, and individual domains can operate separately from other domains. *Trusts* are established between the domains to create a security bridge and allow users from one domain to access another domain's resources. Trusts can be one-way only, or they can be two-way.

LDAP and PKIs

A public key infrastructure (PKI) uses LDAP when integrating digital certificates into transmissions. Chapter 7 covers the topic in more depth, but in short, a PKI is a group of technologies used to manage digital certificates during the certificate life cycle. There are many times when clients need to query a certificate authority (CA) for information on a certificate, and LDAP is one of the protocols used. LDAP and centralized access control systems can be used to support SSO capabilities.

SSO and Federated Identities

SSO is common on internal networks, and it is also used on the Internet with third-party services. Many cloud-based applications

use SSO solutions, making it easier for users to access resources over the Internet. Cloud-based applications use *federated identity management (FIM)* systems, which are a form of SSO.

Identity management is the management of user identities and their credentials. A FIM system links a user's identity in one system with multiple identity management systems.

FIM extends beyond a single organization. Multiple organizations can join a federation or group, where they agree to share identity information. Users in each organization can log on once in their own organization, and their credentials are matched with a federated identity. They can then use this federated identity to access resources in any other organization within the federation.

A federation can be composed of multiple networks within a single university campus, numerous college and university campuses, multiple organizations sharing resources, or any other group that can agree on a common federated identity management system. Members of the federation match user identities within an organization to federated identities.

It's important to realize that membership in a federation doesn't automatically grant everyone access to all resources owned by other members of the federation. Instead, each organization decides what resources to share. Administrators manage these details behind the scenes, and the process is usually transparent to users. The important point is that users don't need to enter their credentials again.

A challenge with multiple companies communicating in a federation is finding a common language. They often have different operating systems, but they still need to share a common language. Chapter 14 discusses the methods used to implement federated identity management systems. These include Security Assertion Markup Language (SAML), OAuth, and OpenID Connect (OIDC).

Cloud-Based Federation

A *cloud-based federation* typically uses a third-party service to share federated identities. As an example, many corporate online

training websites use federated SSO systems. When the organization coordinates with the online training company for employee access, they also coordinate the federated access details.

A common method is to match the user's internal login ID with a federated identity. Users log on within the organization using their normal login ID. When the user accesses the training website with a web browser, the federated identity management system uses their login ID to retrieve the matching federated identity. If it finds a match, it authorizes the user access to the web pages granted to the federated identity.

On-Premises Federation

Federated identity management systems can be hosted on-premises, in the cloud, or in a combination of the two as a hybrid system.

As an example of an *on-premises federated identity management system*, imagine that Acme merges with Emca. Both companies have their own networks and SSO systems. However, management wants employees to be able to access resources in both networks without logging on twice. By creating an on-premises federated identity management system, both companies can share authentication data. This system allows users to continue to log on normally, and they will also have access to the other company's network resources. An on-premises solution provides the organization with the most control.

Hybrid Federation

A *hybrid federation* is a combination of a cloud-based solution and an on-premises solution. Imagine Acme has a cloud-based federation providing employees with online training. After the merger with Emca, they implement an on-premises solution to share identities with the two companies.

This approach doesn't automatically give employees from Emca access to the training sites. However, it is possible to integrate the existing on-premises solution with the training sites' cloud-

based solution. This creates a hybrid solution for Emca employees and, as with other federated solutions, provides SSO for Emca employees.

Just-In-Time

Some federated identity solutions support *Just-in-Time (JIT)* provisioning. These solutions automatically create the relationship between two entities so that new users can access resources. A JIT solution creates the connection without any administrator intervention.

For example, imagine Acme contracted with a third party to provide cafeteria-style benefit plans for employees. The third-party site offers benefit choices such as healthcare plans, life insurance choices, and 401K contribution amounts. Employees access the third-party site and choose the benefits they want. One way to provide employees access to the third-party site is to create separate accounts for every employee, but that can be a huge administrative burden, especially as Acme hires new employees.

With JIT provisioning, employees log on normally to their employer's network. The first time the employee accesses the benefits site, the JIT system exchanges data with the employer's network and creates the employee's account.

JIT systems commonly use SAML to exchange the required data. SAML provides entities with a lot of flexibility to exchange a wide assortment of data. The process starts with the third party verifying the user is logged onto a trusted organization's network. The employer's network then sends data on the employee, such as the username, first and last name, email address, and any other information needed by the third party.

Credential Management Systems

Credential management systems provide storage space for usernames and passwords. As an example, many web browsers can remember usernames and passwords for any site that a user has visited.

The World Wide Web Consortium (W3C) published the Credential Management Level 1 API in January 2019. Many web browsers have adopted the API for credential management. The API provides several benefits that developers can implement programmatically:

- Offering to store the user's credentials after logging on
- Showing a credential chooser, allowing the user to skip sign-in forms
- Automatically logging the user on in subsequent visits, unless the user signed out

Some federated identity management solutions use the Credential Management API. This allows different web applications to implement SSO solutions using a federated identity provider. As an example, if you have a Google or Facebook account, you can use one of them to sign in to Zoom.

Identity as a service (IDaaS) is a third-party service that provides identity and access management (IAM). IDaaS effectively provides SSO for the cloud and is especially useful when internal clients access cloud-based software-as-a-service (SaaS) applications. Google implements this with its motto of "One Google Account for everything Google." Users log into their Google account once, and it provides them with access to multiple Google cloud-based applications without requiring users to log in again.

As another example, Microsoft 365 provides Office applications as a combination of installed applications and SaaS applications. Users have full Office desktop applications installed on their user systems, which can also connect to cloud storage using OneDrive. This allows users to edit and share files from multiple devices. When people use Microsoft 365 at home, Microsoft provides IDaaS, allowing users to authenticate via the cloud to access their data on OneDrive.

When employees use Microsoft 365 from within an enterprise, administrators can integrate the network with a third-party

service. For example, Delinea provides third-party services that integrate with Microsoft Active Directory. Once configured, users log onto the domain and access Microsoft 365 cloud resources without logging on again.

Credential Manager Apps

Windows includes the Credential Manager applet in Control Panel. When a user enters credentials in a browser or a Windows application, Credential Manager offers to save them. It encrypts the credentials and stores them. When a user returns to the website or opens the application, it retrieves the credentials from the Credential Manager.

Third-party credential management systems, known as *password vaults*, are also available. For example, KeePass is a freeware tool that allows you to store your credentials. Credentials are stored in an encrypted database, and users can unlock the database with a master password. Once the database is unlocked, users can easily copy their passwords to paste into a website form. It's also possible to configure the app to enter the credentials automatically into the web page form. Of course, it's important to use a strong master password to protect all the other credentials.

Scripted Access

Scripted access or logon scripts establish communication links by providing an automated process to transmit login credentials at the start of a login session. Scripted access can often simulate SSO even though the environment still requires a unique authentication process to connect to each server or resource. Scripts can implement SSO in environments where true SSO technologies are not available. Scripts and batch files should be stored in a protected area because they usually contain access credentials in cleartext.

Session Management

When you're using any type of authentication system, it's important to use session management methods to prevent

unauthorized access. This includes sessions on regular computers such as desktop PCs and within online sessions with an application.

Desktop PCs and laptops include screen savers. These change the display when the computer isn't in use by displaying random patterns or different pictures or simply blanking the screen. Screen savers protected the computer screens of older computers, but new displays don't need them. However, they're still used, and screen savers have a password-protect feature that can be enabled. This feature displays the logon screen and forces the user to authenticate again before exiting the screen saver.

Screen savers have a time frame in minutes that you can configure. They are commonly set between 10 and 20 minutes. If you set it for 10 minutes, it will activate after 10 minutes. This requires users to authenticate again if the system is idle for 10 minutes or longer.

Secure online sessions will typically terminate after some time too. For example, if you establish a secure session with your bank but don't interact with the session for 10 minutes, the application will typically log you off. In some cases, the application gives you a notification saying it will log you off soon. These notifications usually allow you to click on the page so that you stay logged on. If developers don't implement these automatic logoff capabilities, it allows a user's browser session to remain open with the user logged on. Even if the user closes a browser tab without logging off, it can potentially leave the browser session open, leaving the user's account vulnerable to an attack if someone else accesses the browser.

The Open Worldwide Application Security Project (OWASP) publishes many different “cheat sheets” that provide application developers' specific recommendations. The Session Management Cheat Sheet provides information about web sessions and various methods used to secure them. URLs change, but you can find the cheat sheet by using the search feature at http://owasp.org.

Developers commonly use web development frameworks to implement session management. These are used worldwide and are regularly updated. The framework creates a session identifier at the beginning of the session. This identifier is included in every HTTP request throughout the session. It's possible to force the use of Transport Layer Security (TLS) to ensure the entire session (including the identifier) is encrypted.

These frameworks also include methods to expire sessions. Developers choose the timeout periods, but high-value applications such as applications accessing financial data typically have timeout ranges of 2 to 5 minutes. Low-value applications typically have timeout ranges of 15 to 30 minutes.

Managing the Identity and Access Provisioning Life Cycle

The *identity and access provisioning life cycle* refers to the creation, management, review/audit, and deletion of accounts. Although these activities may seem mundane, they are essential to a system's access control capabilities. Without properly defined and maintained user accounts, a system is unable to establish accurate identity, perform authentication, provide authorization, and track accountability. As mentioned previously, identification occurs when a subject claims an identity. This identity is most commonly a user account, but it also includes computer accounts and service accounts.

Provisioning and Onboarding

An organization typically has an onboarding process after hiring new employees. This includes creating the user account and provisioning it with all the privileges the employee will need in their new job.

Creating new user accounts is usually a simple process, but the process must be protected and secured via organizational security policy procedures. User accounts should not be created at an administrator's whim or in response to random requests. Rather, proper provisioning ensures that personnel follow specific procedures when creating accounts.

The initial creation of a new user account is often called an *enrollment* or registration. The only item that must be provided is a username or a unique identifier. However, based on an organization's established processes, it typically includes multiple details on the user, such as the user's full name, email address, and more. When an organization uses biometric methods of authentication, biometric data is also collected and stored during this enrollment process.

It is also critical that the new hire's identity is proved through whatever means an organization deems necessary and sufficient. Photo ID, birth certificate, background check, credit check, security clearance verification, FBI database search, and even reference checks are all valid forms of verifying a person's identity before enrolling them in any secured system.

Many organizations have automated provisioning systems. For example, once a person is hired, the HR department completes initial identification and in-processing steps and then forwards a request to the IT department to create an account. IT personnel enter information such as the employee's name and their assigned department via an application. The application then creates the account using predefined rules. Automated provisioning systems create accounts consistently, such as always creating usernames the same way and treating duplicate usernames consistently. If the policy dictates that usernames include first and last names, then the application will create a

username as `suziejones` for a user named Suzie Jones. If the organization hires a second employee with the same name, then the second username might be `suziejones2`.

If the organization is using groups (or roles), the application can automatically add the new user account to the appropriate groups based on the user's department or job responsibilities. The groups will already have appropriate privileges assigned, so this step provisions the account with appropriate privileges.

Provisioning also includes issuing hardware such as laptops, mobile devices, hardware authenticators, and smartcards to employees. It's important to keep accurate records when issuing hardware to employees.

After provisioning employees with accounts and any hardware they need, organizations follow up with onboarding processes. Chapter 2, “Personnel Security and Risk Management Concepts,” introduced onboarding processes. Onboarding processes include items such as the following:

- Having them read and sign the organization's acceptable use policy (AUP)
- Explaining security best practices, such as how to avoid malware infections from emails
- Reviewing the organization's mobile device policy, if applicable
- Ensuring that the employee's computer is operational and that the employee can log on
- Helping the employee configure a password manager, if available
- Assisting the employee with configuring 2FA, if available
- Explaining how to access help desk personnel for further assistance
- Showing the employee how to access, share, and save resources

These onboarding items help set up a new employee for a successful start. Some of them may seem unnecessary, especially for employees working with the organization for a while.

Consider an organization that uses nonpersistent virtual desktops. When the user logs off, all data and settings are lost. A new employee can spend a day creating and saving files, only to come back the next day and find that everything is gone.

Deprovisioning and Offboarding

Organizations implement deprovisioning and offboarding processes when employees leave an organization. This includes when an employee is terminated for cause, is laid off, or leaves under the best of conditions. These same processes can be used when an employee transfers to a different department or location within the same organization.

Chapter 2 covers onboarding, transfers, and termination processes in the context of security policies and procedures. This section reviews them in the context of an identity and access provisioning life cycle.

The easiest way to deprovision an account is to delete it, sometimes referred to as *account revocation*. This process removes all access that the employee had while employed. However, it may also remove access to the user's data. For example, if the user encrypted data, the user account may have the only access to the decryption key to decrypt the data.

Many organizations choose to disable the account when the employee leaves. Supervisors can then review the user's data and determine if anything is needed before deleting the account. If some data is encrypted, administrators can change the user's password and give the supervisor the new password. The supervisor can now log on as the ex-employee and decrypt the data. Organizations typically have policies in place to delete these

disabled accounts within 30 days, but the time limit can vary depending on the organization's needs.

If a terminated employee retains access to a user account after the exit interview, the risk for sabotage is very high. Even if the employee doesn't take malicious action, other employees may be able to use the account if they discover the password. Logs will record the activity in the terminated employee's name instead of the person actually performing the malicious activity.

Deprovisioning includes collecting any hardware issued to an employee, such as laptops, mobile devices, and authenticator devices. This process is a lot easier if an organization keeps accurate records of what they issue to employees.

It's also important to terminate employee benefits as part of the offboarding process. Without processes in place to do so, the organization may continue to pay for benefits even after employees leave. As an example, the human resource management system used by the University of Wisconsin failed to terminate health insurance premiums for 924 ex-employees several years ago. An audit discovered that they paid about $8 million before it was discovered.

Role Definition and Transition

During the lifetime of any organization, employee responsibilities will change. Many times, this is just a simple transfer to a different position. Other times an organization may create a completely different job role. When they do so, it's important to define the new role and the privileges needed by employees in the role.

As an example, imagine an organization decides to start selling items with an e-commerce site hosted on a new Linux server running Apache. Developers will write and maintain the code for the site, and administrators will manage the server. If they don't already have website developers and Linux administrators, they may decide to create two new roles to support this project. They would also define the privileges needed for these new roles and how they plan on assigning the privileges, such as with groups.

Account Maintenance

Throughout the life of a user account, ongoing maintenance is required. Organizations with static organizational hierarchies and low employee turnover or promotion will conduct significantly less account administration than an organization with a flexible or dynamic organizational hierarchy and high employee turnover and promotion rates.

Most account maintenance deals with altering rights and privileges. Procedures similar to those used when creating new accounts should be established to govern how access is changed throughout the life of a user account. Unauthorized increases or decreases in an account's access capabilities can cause serious security repercussions.

Account Access Review

Administrators periodically review accounts to ensure they don't have excessive privileges. Account reviews also check to ensure accounts comply with security policies. This includes user accounts, privileged accounts, system accounts, and service accounts. The “Device Authentication” section in this chapter discussed system accounts, such as those assigned to computers, and the “Service Authentication” section in this chapter discussed service accounts.

The local system account on computers typically has the same privileges as the local administrator account. This approach allows the computer to access other computers on the network as the computer, instead of as a user. Some applications use the local system account as the service account. This approach allows the application to run without creating a special service account, but it often grants the application more access than it needs. If an attacker exploits an application vulnerability, the attacker may gain access to the service account.

Many administrators use scripts to check for inactive accounts periodically. For example, a script can locate accounts that users have not logged onto in the past 30 days and automatically

disable them. Similarly, scripts can check group membership of privileged groups (such as administrator groups) and remove unauthorized accounts. Routine auditing procedures often include account reviews.

Privilege monitoring audits accounts that have elevated privileges. This includes any accounts with administrator privileges such as administrator accounts, root accounts, service accounts, or any account that has more privileges than a regular user.

It's important to guard against two problems related to access control: *excessive privilege* and *privilege creep*. Excessive privilege occurs when users have more privileges than their assigned work tasks dictate. If a user account has excessive privileges, administrators should revoke unnecessary privileges.

Privilege creep involves a user account accumulating additional privileges over time as job roles and assigned tasks change. As an example, imagine Karen is working in the accounting department and transfers to the sales department. She has privileges in the accounting department, and when she transfers to sales, she's granted the privileges needed in the sales department. If administrators don't remove her rights and permissions in accounting, she retains excessive privileges. Both excessive privileges and privilege creep violate the basic security principle of least privilege, and account reviews are effective at discovering these problems.

Summary

Identity and access management (IAM) covers the management, administration, and implementation aspects of granting or restricting access to assets. Assets include personnel, information, systems, devices, facilities, applications, or services. Organizations use both physical and logical access controls to protect them.

Identification is the process of a subject claiming, or professing, an identity. Authentication verifies the subject's identity by comparing one or more authentication factors against a database

holding authentication information for users. The three primary authentication factors are something you know, something you have, and something you are. Multifactor authentication uses more than one authentication factor, and it is stronger than using any single authentication factor.

Single sign-on (SSO) technologies allow users to authenticate once and access any resources in a network without authenticating again. Internal networks commonly use SSO, and SSO capabilities are also available on the Internet and via the cloud. Federated identity management (FIM) systems link user identities in one system with other systems to implement SSO.

The identity and access provisioning life cycle includes creating, managing, reviewing/auditing, and deleting accounts used by subjects. Provisioning includes creating the accounts and ensuring that they are granted appropriate access to objects and issuing employees the hardware they need for their job. Onboarding processes inform employees of organizational processes and help set up new employees for success. Deprovisioning processes disable or delete an account when employees leave, and offboarding processes ensure that employees return all the hardware an organization issued to them.

Study Essentials

Know how physical access controls protect assets. Physical access controls are those you can touch, and they directly protect systems, devices, and facilities by controlling access and controlling the environment. Indirectly, they also protect information and applications by limiting physical access.

Know how logical access controls protect assets. Logical access controls include authentication, authorization, and permissions. They limit who can access information, settings, and use of information, systems, devices, facilities, applications, and services.

Know the difference between subjects and objects. You'll find that Security documentation commonly uses the terms

subject and *object,* so it's important to know the difference between them. Subjects are active entities (such as users) that access passive objects (such as files). A user is a subject who accesses objects while performing some action or accomplishing a work task.

Know the components of the AAA model of access control. The AAA model includes three major components. Authentication confirms that a user, device, or service is who it claims to be. Authorization ensures that users, devices, and services may only perform actions that they are entitled to perform. Accounting creates an audit trail of activity that may be later verified.

Know the difference between identification and authentication. Access controls depend on effective identification and authentication. Subjects claim an identity, and identification can be as simple as a username for a user. Subjects prove their identity by providing authentication credentials such as the matching password for a username. People, devices, and services all verify their identity by giving proper credentials.

Understand the establishment of identity, registration, and proofing. New employees establish their identities with official documentation such as a passport, driver's license, or birth certificate. HR personnel then begin the registration process, which includes creating an account for new employees. When biometric authentication is used, the registration process also collects biometric data. Identity proofing includes knowledge-based authentication and cognitive passwords. These ask users a series of questions that only the user would know.

Understand the difference between authorization and accounting. After authenticating subjects, systems authorize access to objects based on their proven identity. Auditing logs and audit trails record events, including the identity of the subject that performed an action. The combination of effective identification, authentication, and auditing provides accountability.

Know the primary authentication factors. The three primary factors of authentication are something you know (such as a password or PIN), something you have (such as a smartcard or authenticator device), and something you are (based on biometrics). Multifactor authentication (MFA) includes two or more authentication factors, and using MFA is more secure than using a single authentication factor.

Understand important authentication concepts. Passwords are the weakest form of authentication, but password policies help increase their security by enforcing complexity and history requirements. Smartcards include microprocessors and cryptographic certificates, and authenticators create one-time passwords (OTPs). Biometric methods identify users based on characteristics such as fingerprints. The crossover error rate (CER) identifies the accuracy of a biometric method and shows where the false rejection rate (FRR) is equal to the false acceptance rate (FAR). Lower CERs are more accurate.

Understand single sign-on. Single sign-on (SSO) is a mechanism that allows subjects to authenticate once and access multiple objects without authenticating again.

Describe how federated identity systems are implemented. FIM systems are implemented on-premises (providing the most control), via a third-party cloud service or as a hybrid of both.

Describe Just-in-Time (JIT) provisioning. Just-in-Time provisioning creates user accounts on third-party sites the first time a user logs onto the site. JIT reduces the administrative workload.

Know about credential management systems. Credential management systems help developers easily store usernames and passwords and retrieve them when a user revisits a website. The W3C published the Credential Management API as a working draft in 2019, and developers commonly use it as a credential management system. It allows users to log on automatically to websites without entering their credentials again.

Explain session management. Session management processes help prevent unauthorized access by closing unattended sessions. Developers commonly use web frameworks to implement session management. These frameworks allow developers to ensure sessions are closed after a specific amount of inactivity, such as after 2 minutes.

Understand the identity and access provisioning life cycle. The identity and access provisioning life cycle refers to the creation, management, and deletion of accounts. Provisioning ensures that accounts have appropriate privileges based on task requirements and employees receive any needed hardware. Onboarding processes inform employees of organizational processes. Deprovisioning processes disable or delete an account when employees leave, and offboarding processes ensure that employees return all the hardware an organization issued to them.

Explain the importance of group and role definition and transition. When an organization creates new job roles, it's important to identify privileges needed by anyone in these new roles. Doing so ensures that employees in these new roles do not have excessive privileges. These roles are commonly mapped to groups in the authentication system and then privileges are assigned to those roles. When users transition from one job to another, their group membership should be modified to follow those changes

Describe the purpose of account access reviews. Account access reviews are performed on user accounts (including privileged accounts), system accounts, and service accounts. These reviews ensure that accounts don't have excessive privileges. They can often detect when accounts have excessive privileges and when unused accounts have not been disabled or deleted.

Written Lab

1. List some physical and logical access controls used to protect assets.