Chapter 2
Understanding Identity and Access Management

CompTIA Security+ objectives covered in this chapter:

1.6 Explain the impact associated with types of vulnerabilities.

- Improperly configured accounts

2.3 Given a scenario, troubleshoot common security issues.

- Unencrypted credentials/clear text, Authentication issues

2.6 Given a scenario, implement secure protocols.

- Protocols (LDAPS)

3.3 Given a scenario, implement secure systems design.

- Operating systems (Disable default accounts/passwords)

4.1 Compare and contrast identity and access management concepts.

- Identification, authentication, authorization and accounting (AAA), Multifactor authentication (Something you are, Something you have, Something you know, Somewhere you are, Something you do), Federation, Single sign-on, Transitive trust

4.2 Given a scenario, install and configure identity and access services.

- LDAP, Kerberos, SAML, OpenID Connect, OAUTH, Shibboleth, Secure token, NTLM

4.3 Given a scenario, implement identity and access management controls.

- Access control models (MAC, DAC, ABAC, Role-based access control, Rule-based access control), Biometric factors (Fingerprint scanner, Retinal scanner, Iris scanner, Voice recognition, Facial recognition, False acceptance rate, False rejection rate, Crossover error rate), Tokens (Hardware, Software, HOTP/TOTP), Certificate-based authentication (PIV/ CAC/smart card)

4.4 Given a scenario, differentiate common account management practices.

- Account types (User account, Shared and generic accounts/credentials, Guest accounts, Service accounts, Privileged accounts), General Concepts (Least privilege, Time-of-day restrictions, Recertification, Standard naming convention, Account maintenance, Group-based access control, Location- based policies), Account policy enforcement (Credential management, Group policy, Password complexity, Expiration, Recovery, Disablement, Lockout, Password history, Password reuse, Password length)

6.1 Compare and contrast basic concepts of cryptography.

- Common use cases (Supporting authentication)

**

Domain 4.0 in the CompTIA Security+ objectives is Identity and Access Management, which admittedly is a mouthful. In simpler language, users claim an identity with a username and prove their identity by authenticating (such as with a password). They are then granted access to resources based on their proven identity. In this chapter, you'll learn about various authentication concepts and methods, along with some basic security principles used to manage accounts. This chapter closes with a comparison of some access control models.

Exploring Authentication Concepts

Authentication proves an identity with some type of credentials, such as a username and password. For example, ***identification*** occurs when users claim (or profess) their identity with identifiers such as usernames or email addresses. Users then prove their identity with ***authentication***, such as with a password. In this context, a user's credentials refer to both a claimed identity and an authentication mechanism.

At least two entities know the credentials. One entity, such as a user, presents the credentials. The other entity is the authenticator that verifies the credentials. For example, Marge knows her username and password, and an authenticating server knows her username and password. Marge presents her credentials to the authenticating server, and the server authenticates her.

The importance of authentication cannot be understated. You can't have any type of access control if you can't identify a user. In other words, if everyone is anonymous, then everyone has the same access to all resources.

Also, authentication is not limited to users. Services, processes, workstations, servers, and network devices all use authentication to prove their identities. Many computers use mutual authentication, where both parties authenticate to each other.

Comparing Identification and AAA

Authentication, authorization, and accounting (***AAA***) work together with identification to provide a comprehensive access management system. If you understand identification (claiming an identity, such as with a username) and authentication (providing the identity, such as with a password), it's easier to add in the other two elements of AAA—authorization and accounting.

If users can prove their identity, that doesn't mean that they are automatically granted access to all resources within a system. Instead, users are granted ***authorization*** to access resources based on their proven identity. This can be as simple as granting a user permission to read data in a shared folder. Access control systems include multiple security controls to ensure that users can access resources they're authorized to use, but no more.

Accounting methods track user activity and record the activity in logs. As an example, audit logs track activity and administrators use these to create an audit trail. An ***audit trail*** allows security professionals to re-create the events that preceded a security incident.

Effective access control starts with strong authentication mechanisms, such as the use of robust passwords, smart cards, or biometrics. If users can bypass the authentication process, the authorization and accounting processes are ineffective.

Remember this

Identification occurs when a user claims an identity such as with a username or email address. Authentication occurs when the user proves the claimed identity (such as with a password) and the credentials are verified. Access control systems provide authorization by granting access to resources based on permissions granted to the proven identity. Logging provides accounting.

Comparing Authentication Factors

Authentication is often simplified as types, or factors, of authentication. A use case of supporting authentication may require administrators to implement one factor of authentication for basic authentication, two factors for more secure authentication, or more factors for higher security. As an introduction, the factors are:

- Something you know, such as a password or personal identification number (PIN)
- Something you have, such as a smart card or USB token
- Something you are, such as a fingerprint or other biometric identification
- Somewhere you are, such as your location using geolocation technologies
- Something you do, such as gestures on a touch screen

Something You Know

The ***something you know*** authentication factor typically refers to a shared secret, such as a password or even a PIN. This factor is the least secure form of authentication. However, you can increase the security of a password by following some simple guidelines. The following sections provide more details on important password security concepts.

Password Complexity

One method used to make passwords more secure is to require them to be complex and strong. A strong password is of sufficient length, doesn't include words found in a dictionary or any part of a user's name, and combines at least three of the four following character types:

- Uppercase characters (26 letters A–Z)
- Lowercase characters (26 letters a–z)
- Numbers (10 numbers 0–9)
- Special characters (32 printable characters, such as !, $, and *)

A complex password uses multiple character types, such as Ab0@. However, a complex password isn't necessarily strong. It also needs to be sufficiently long. It's worth noting that recommendations for the best length of a strong password vary depending on the type of account. As of January 2016, Microsoft began recommending a best practice of setting the minimum password length to at least 14 characters. Organizations often require

administrators to create longer passwords. A key point is that longer passwords using more character types are more secure and short passwords of 4 or 5 characters are extremely weak.

The combination of different characters in a password makes up the key space, and you can calculate the key space with the following formula: C^N (C^N). *C* is the number of possible characters used, and *N* is the length of the password. The ^ character in C^N indicates that *C* is raised to the *N* power.

For example, a 6-character password using only lowercase letters (26 letters) is calculated as 26^6 (26^6), or about 308 million possibilities. Change this to a 10-character password and the value is 26^10 (26^{10}), or about 141 trillion possibilities. Although this looks like a high number of possibilities, there are password-cracking tools that can test more than 20 billion passwords per second on desktop computers with a high-end graphics processor. An attacker can crack a 10-character password using only lowercase characters (141 trillion possibilities) in less than two hours.

However, if you use all 94 printable characters (uppercase, lowercase, numbers, and special characters) with the same 6- and 10-character password lengths, the values change significantly: 94^6 (94^6) is about 689 billion possibilities, and 94^10 (94^{10}) is about 53 quintillion. That's 53 followed by 18 zeroes.

You probably don't come across quintillion very often. The order is million, billion, trillion, quadrillion, and then quintillion. The password-cracking tool that cracks a lowercase password in two hours will take years to crack a 10-character password using all four character types.

Security experts often mention that if you make a password too complex, you make it less secure.

Read that again. It is not a typo.

More complexity equates to less security. This is because users have problems remembering overly complex passwords such as 4%kiElNsB* and they are more likely to write them down. A password written on paper or stored in a file on a user's computer significantly reduces security. Instead, users are encouraged to use passphrases. Instead of nonsensical strings of characters, a passphrase is a long string of characters that has meaning to the user. A few examples of strong passphrases are IL0veSecurity+, IL0veThi$B00k, and IWi11P@$$. Note that these examples include all four character types—uppercase letters, lowercase letters, one or more numbers, and one or more special characters. These passwords are also known as

passphrases because they are a combination of words that are easier to remember than a nonsensical string
of characters such as 4*eiRS@<].

Strong passwords never include words that can be easily guessed, such as a user's name, words in a dictionary (for any language), or common key combinations.

Remember this

Complex passwords use a mix of character types. Strong passwords use a mix of character types and have a minimum password length of at least 14 characters.

Training Users About Password Behaviors

Common user habits related to password behaviors have historically ignored security. Many users don't understand the value of their password, or the potential damage if they give it out. It's important for an organization to provide adequate training to users on password security if

they use passwords within the organization. This includes both the creation of strong passwords and the importance of never giving out their passwords.

For example, the password “123456” frequently appears on lists as the most common password in use. The users who are creating this password probably don’t know that it’s almost like using no password at all. Also, they probably don’t realize that they can significantly increase the password strength by using a simple passphrase such as “ICanCountTo6.” A little training can go a long way.

Check out the online lab Using John the Ripper available at *http://gcgapremium.com/501labs/.* It shows how easy it can be to crack weak passwords.

Ashley Madison Passwords (Sidebar)

You might think that users never use weak passwords anymore, but that simply isn’t true. As an example, attackers hacked into the Ashley Madison site in 2015 and posted information on 36 million accounts. Ashley Madison is an online dating service marketed to people who are married or in committed relationships, but still want to date others. Users have a vested interest in keeping their information private, but many were still using weak passwords.

Here’s a list of some of the passwords in the top 10 list that

users had created for their accounts: 123456, 12345, password, DEFAULT, 123456789, qwerty, 12345678, abc123, and 1234567. Over 120,000 users had 123456 as their password.

Password Expiration

In addition to using strong passwords, users should change their passwords regularly, such as every 45 or 90 days. In most systems, technical password policies require users to change their passwords regularly. When the password expires, users are no longer able to log on unless they first change their password.

I can tell you from experience that if users are not forced to change their passwords through technical means, they often simply don't. It doesn't matter how many reminders you give them. On the other hand, when a password policy locks out user accounts until they change their password, they will change it right away.

Pass word Recovery

It's not uncommon for users to occasionally forget their password. In many organizations, help-desk professionals or other administrators reset user passwords.

Before resetting the password, it's important to verify the user's identity. Imagine that Hacker Harry calls into the help desk claiming to be the CEO and asks for his password to be reset. If the help- desk professional does so, it locks the CEO out of the account. Worse, depending on the process, it might give Hacker Harry access to the CEO's account. Organizations use a variety of different methods of identification before resetting a user's account to prevent these vulnerabilities.

In some systems, help-desk professionals manually change the user's password. This causes a different problem. Imagine a user calls the help desk and asks for a password reset. The help- desk professional changes the password and lets the user know the new password. However, at this point, two people know the password. The help-desk professional could use the password and impersonate the user, or the user could blame the help-desk professional for impersonating the user.

Instead, the help-desk professional should set the password as a temporary password that expires upon first use. This requires the user to

change the password immediately after logging on and it maintains password integrity.

Instead of an IT professional spending valuable time resetting passwords, a self-service password reset or password recovery system automates the process. For example, many online systems include a link, such as “Forgot Password.” If you click on this link, the system might send you your password via email, or reset your password and send the new password via email.

Some systems invoke an identity-proofing system. The identity-proofing system asks you questions that you previously answered, such as the name of your first dog, the name of your first boss, and so on. Once you adequately prove your identity, the system gives you the opportunity to change your password.

Many password reset systems send you a code, such as a six-digit PIN, to your mobile phone or to an alternate email address that you’ve preconfigured. When you receive this PIN, you can enter it and then change your password.

Remember this

Before resetting passwords for users, it’s important to verify the user’s identity. When resetting passwords manually, it’s best to create a temporary password that expires upon first use.

Password History and Password Reuse

Many users would prefer to use the same password forever simply because it’s easier to remember. Even when technical password policies force users to change their passwords, many users simply change them back to the original password. Unfortunately, this significantly weakens password security.

A password history system remembers past passwords and prevents users from reusing passwords. It’s common for password policy settings to remember the last 24 passwords and prevent users from reusing these until they’ve used 24 new passwords.

Group Policy

Windows domains use Group Policy to manage multiple users and

computers in a domain. Group Policy allows an administrator to configure a setting once in a Group Policy Object (***GPO***) and apply this setting to many users and computers within the domain. Active Directory Domain Services (AD DS) is a directory service Microsoft developed for Windows domain networks. It is included in most Windows Server operating systems as a set of processes and services. Administrators implement domain Group Policy on domain controllers.

Although you can implement Group Policy on single, stand-alone Windows computers, the great strength of Group Policy comes when you implement it in a Microsoft domain. As an example, if you want to change the local Administrator password on all the computers in your domain, you can configure a GPO once, link the GPO to the domain, and it changes the local Administrator password for all the computers in the domain. The magic of Group Policy is that it doesn't matter if you have five systems or five thousand systems. The policy still only needs to be set once to apply to all systems in the domain.

Administrators also use Group Policy to target specific groups of users or computers. For example, in a Microsoft domain, administrators organize user accounts and computer accounts in organizational units (OUs). They can then create a GPO, link it to a specific OU, and the GPO settings only apply to the users and computers within the OU. These settings do not apply to users and computers in other OUs.

Remember this

Group Policy is implemented on a domain controller within a domain. Administrators use it to create password policies, implement security settings, configure host-based firewalls, and much more.

Using a Password Policy

A common group of settings that administrators configure in Group Policy is the Password Policy settings. Password policies typically start as a written document that identifies the organization's security goals related to passwords. For example, it might specify that passwords must be at least 14 characters long, complex, and users should change them every 45 days. Administrators then implement these requirements with a technical control

such as a technical Password Policy within a GPO.

Figure 2.1 shows the Local Group Policy Editor with the Password Policy selected in the left pane. The right pane shows the password policy for a Windows system and the following text explains these settings:

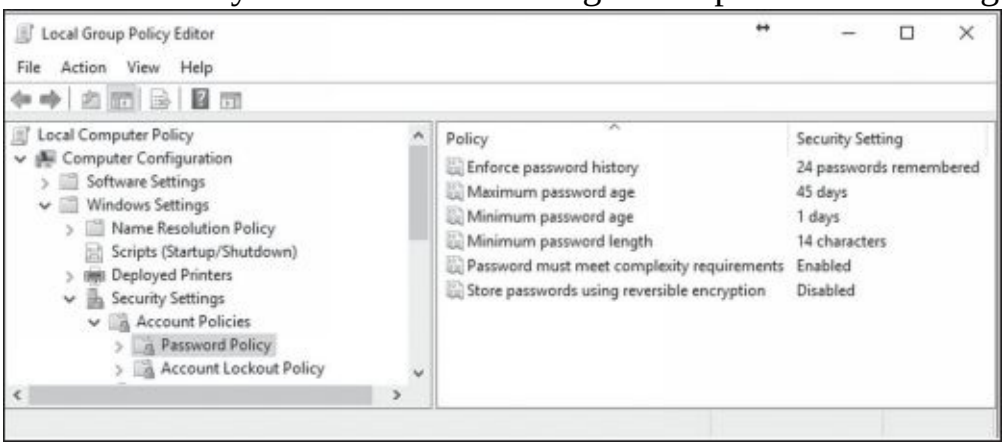


Figure 2.1: Password Policy in Windows

- **Enforce password history.** Some users will go back and forth between two passwords that they constantly use and reuse. However, password history remembers past passwords and prevents the user from reusing previously used passwords. For example, setting this to 24 prevents users from reusing passwords until they've used 24 new passwords.
- **Maximum password age.** This setting defines when users must change their password. For example, setting this to 45 days causes the password to expire after 45 days. This forces users to reset their password to a new password on the 46th day.
- **Minimum password age.** The minimum password age defines how long users must wait before changing their password again. If you set this to 1 day, it prevents users from changing their passwords until 1 day has passed. This is useful with a password history to prevent users from changing their password multiple times until they get back to the original password. If the password history is set to 24 and the minimum password age is set to 1 day, it will take a user 25 days to get back to the original password. This is enough to discourage most users.
- **Minimum password length.** This setting enforces the character length of the password. It's common to require users to have passwords at least 14 characters long, but some organizations require administrators to have longer passwords.

- **Password must meet complexity requirements.** This setting requires users to have complex passwords that include at least three of the four character types (uppercase letters, lowercase letters, numbers, and special characters).
- **Store passwords using reversible encryption.** Reversible encryption stores the password in such a way that the original password can be discovered. This is rarely enabled.

Remember this

Password policies include several elements. The password history is used with the minimum password age to prevent users from changing their password to a previously used password. Maximum password age causes passwords to expire and requires users to change their passwords periodically. Minimum password length specifies the minimum number of characters in the password. Password complexity increases the key space, or complexity, of a password by requiring more character types.

Implementing Account Lockout Policies

Accounts will typically have lockout policies preventing users from guessing the password. If a user enters the wrong password too many times (such as three or five times), the system locks the user's account. Figure 2.1 shows the Password Policy settings. The Account Lockout Policy is right below it and allows administrators to use Group Policy to implement a lockout policy.

Two key phrases associated with account lockout policies are:

- **Account lockout threshold.** This is the maximum number of times a user can enter the wrong password. When the user exceeds the threshold, the system locks the account.
- **Account lockout duration.** This indicates how long an account remains locked. It could be set to 30, indicating that the system will lock the account for 30 minutes. After 30 minutes, the system automatically unlocks the account. If the duration is set to 0, the account remains locked until an administrator unlocks it.

Changing Default Passwords

Many systems and devices start with default passwords. A basic security practice is to change these defaults before putting a system into use. As an example, many wireless routers have default accounts named "admin" or "administrator" with a default password of "admin." If you don't change the password, anyone who knows the defaults can log on and take control of the router. In that case, the attacker can even go as far as locking you out of your own network.

Changing defaults also includes changing the default name of the Administrator account, if possible. In many systems, the Administrator account can't be locked out through regular lockout policies, so an attacker can continue to try to guess the password of the Administrator account without risking being locked out. Changing the name of the Administrator account to something else, such as Not4U2Know, reduces the chances of success for the attacker. The attacker needs to know the new administrator name before he can try to guess the password.

Some administrators go a step further and add a dummy user account named "administrator." This account has no permissions. If someone does try to guess the password of this account, the system will lock it out, alerting administrators of possible illicit activity.

Something You Have

The ***something you have*** authentication factor refers to something you can physically hold. This section covers many of the common items in this factor, including smart cards, Common Access Cards, and hardware tokens. It also covers two open source protocols used with both hardware and software tokens.

Smart Cards

Smart cards are credit card-sized cards that have an embedded microchip and a certificate. Users insert the ***smart card*** into a smart card reader, similar to how someone would insert a credit card into a credit card reader. The smart card reader reads the information on the card, including the details from the certificate, which provides certificate-based authentication.

Chapter 10, "Understanding Cryptography and PKI," covers certificates in more detail, but as an introduction, they are digital files that support cryptography for increased security. The embedded certificate allows the use

of a complex encryption key and provides much more secure authentication than is possible with a simple password. Additionally, the certificate can be used with digital signatures and data encryption. The smart card provides confidentiality, integrity, authentication, and non-repudiation.

Requirements for a smart card are:

- **Embedded certificate.** The embedded certificate holds a user's private key (which is only accessible to the user) and is matched with a public key (that is publicly available to others). The private key is used each time the user logs on to a network.
- **Public Key Infrastructure (PKI).** Chapter 10 covers PKI in more depth, but in short, the PKI supports issuing and managing certificates.

Smart cards are often used with another factor of authentication. For example, a user may also enter a PIN or password, in addition to using the smart card. Because the smart card is in the something you have factor and the PIN is in the something you know factor, this combination provides dual-factor authentication.

CACs and PIVs

A Common Access Card (***CAC***) is a specialized type of smart card used by the U.S. Department of Defense. In addition to including the capabilities of a smart card, it also includes a picture of the user and other readable information. Users can use the CAC as a form of photo identification to gain access into a secure location. For example, they can show their CAC to guards who are protecting access to secure areas. Once inside the secure area, users can use the CAC as a smart card to log on to computers.

Similarly, a Personal Identity Verification (***PIV***) card is a specialized type of smart card used by U.S. federal agencies. It also includes photo identification and provides confidentiality, integrity, authentication, and non-repudiation for the users, just as a CAC does.

CACs and PIVs both support dual-factor authentication (sometimes called two-factor authentication) because users generally log on with the smart card and by entering information they know such as a password. Additionally, just as with smart cards, these cards include embedded certificates used for digital signatures and encryption.

Remember this

Smart cards are often used with dual-factor authentication where users have something (the smart card) and know something (such as a password or PIN). Smart cards include embedded certificates used with digital signatures and encryption. CACs and PIVs are specialized smart cards that include photo identification. They are used to gain access into secure locations and to log on to computer systems.

Tokens or Key Fobs

A ***token*** or key fob (sometimes simply called a fob) is an electronic device about the size of a remote key for a car. You can easily carry them in a pocket or purse, or connect them to a key chain. They include a liquid crystal display (LCD) that displays a number, and this number changes periodically, such as every 60 seconds. They are sometimes called hardware tokens to differentiate them from logical, or software tokens.

The token is synced with a server that knows what the number is at any moment. For example, at 9:01, the number displayed on the token may be 135792 and the server knows the number is 135792. At 9:02, the displayed number changes to something else and the server also knows the new number.

This number is a one-time use, rolling password. It isn't useful to attackers for very long, even if they can discover it. For example, a shoulder surfing attacker might be able to look over someone's shoulder and read the number. However, the number expires within the next 60 seconds and is replaced by another one-time password.

Users often use tokens to authenticate via a web site. They enter the number displayed in the token along with their username and password. This provides dual-factor authentication because the users must have something (the token) and know something (their password).

RSA sells RSA Secure ID, a popular token used for authentication. You can Google "Secure ID image" to view many pictures of these tokens. Although RSA tokens are popular, other brands are available.

HOTP and TOTP

Hash-based Message Authentication Code (HMAC) uses a hash function and cryptographic key for many different cryptographic functions.

Chapter 1, “Mastering Security Basics,” introduced hashes. As a reminder, a hash is simply a number created with a hashing algorithm. HMAC-based One-Time Password (***HOTP***) is an open standard used for creating one-time passwords, similar to those used in tokens or key fobs. The algorithm combines a secret key and an incrementing counter, and uses HMAC to create a hash of the result. It then converts the result into an HOTP value of six to eight digits.

Imagine Bart needs to use HOTP for authentication. He requests a new HOTP number using a token or a software application. He can then use this number for authentication along with some other authentication method, such as a username and password. As soon as he uses it, the number expires. No one else is able to use it, and Bart cannot use it again either.

Here’s an interesting twist, though. A password created with HOTP remains valid until it’s used. Suppose Bart requested the HOTP number but then got distracted and never used it. What happens now? Theoretically, it remains usable forever. This presents a risk related to HOTP because other people can use the password if they discover it.

A Time-based One-Time Password (***TOTP***) is similar to HOTP, but it uses a timestamp instead of a counter. One-time passwords created with TOTP typically expire after 30 seconds.

One significant benefit of HOTP and TOTP is price. Hardware tokens that use these open source standards are significantly less expensive than tokens that use proprietary algorithms. Additionally, many software applications use these algorithms to create software tokens used within the application.

For example, Figure 2.2 shows the free VIP Access app created by Symantec and running on an iPad. It’s also available for many other tablets and smartphones. Once you configure it to work with a compatible authentication server, it creates a steady stream of one-time use passwords. The six-digit security code is the password, and the counter lets you know how much more time you have before it changes again.

Figure 2.2: VIP Access app

Similar to a hardware token, the user enters a username and password as the something you know factor, and then enters the security code from the app as the something you have factor. This provides dual-factor authentication. Many public web sites like eBay and PayPal support it, allowing many end users to implement dual-factor authentication as long as they have a smartphone or tablet device.

Remember this

HOTP and TOTP are both open source standards used to create one-time use passwords. HOTP creates a one-time use password that does not expire. TOTP creates a one-time password that expires after 30 seconds. Both can be used as software tokens for authentication.

Something You Are

The ***something you are*** authentication factor uses biometrics for authentication. Biometric methods are the strongest form of authentication because they are the most difficult for an attacker to falsify. In comparison, passwords are the weakest form of authentication.

Biometric Methods

Biometrics use a physical characteristic, such as a fingerprint, for authentication. Biometric systems use a two-step process. In the first step, users register with the authentication system. For example, an authentication system first captures a user's fingerprint and associates it with the user's identity. Later, when users want to access the system, they use their fingerprints to prove their identity. There are multiple types of biometrics, including:

- **Fingerprint scanner.** Many laptop computers include ***fingerprint scanners*** or fingerprint readers, and they are also common on tablet devices and smartphones. Similarly, some USB flash drives include a fingerprint scanner. They can store multiple fingerprints of three or four people to share access to the same USB drive. Law enforcement agencies have used fingerprints for decades, but they use them for identification, not biometric authentication.
- **Retina scanner.** ***Retina scanners*** scan the retina of one or both

eyes and use the pattern of blood vessels at the back of the eye for recognition. Some people object to the use of these scanners for authentication because they can identify medical issues, and because you typically need to have physical contact with the scanner.

- **Iris scanner.** ***Iris scanners*** use camera technologies to capture the patterns of the iris around the pupil for recognition. They are used in many passport-free border crossings around the world. They can take pictures from about 3 to 10 inches away, avoiding physical contact.
- **Voice recognition.** ***Voice recognition*** methods identify who is speaking using speech recognition methods to identify different acoustic features. One person's voice varies from another person's voice due to differences in their mouth and throat, and behavioral patterns that affect their speaking style. As an example, Apple's Siri supports voice recognition. After setting it up, Siri will only respond to the owner's voice. Unfortunately, that does prevent the old party trick of yelling out "Hey Siri" at a party where multiple people have iPhones.
- **Facial recognition.** ***Facial recognition*** systems identify people based on facial features. This includes the size of their face compared with the rest of their body, and the size, shape, and position of their eyes, nose, mouth, cheekbones, and jaw. A drawback with this is that it is sometimes negatively affected by changes in lighting. Microsoft Windows systems support Windows Hello facial recognition services. To avoid the challenges from normal lighting, it uses infrared (IR) and can operate in diverse lighting conditions.

Biometric Errors

Biometrics can be very exact when the technology is implemented accurately. However, it is possible for a biometric manufacturer to take shortcuts and not implement it correctly, resulting in false readings. Two biometric false readings are:

- **False acceptance.** This is when a biometric system incorrectly identifies an unauthorized user as an authorized user. The false acceptance rate (***FAR***, also known as a false match rate) identifies the percentage of times false acceptance occurs.
- **False rejection.** This is when a biometric system incorrectly rejects

an authorized user. The false rejection rate (***FRR***, also known as a false nonmatch rate) identifies the percentage of times false rejections occur.

True readings occur when the biometric system accurately accepts or rejects a user. For example, true acceptance is when the biometric system accurately determines a positive match. In contrast, true rejection occurs when the biometric system accurately determines a nonmatch. Biometric systems allow you to adjust the sensitivity or threshold level where errors occur. By increasing the sensitivity, it decreases the number of false matches and increases the number of false rejections. In contrast, decreasing the sensitivity increases the false matches and decreases the false rejections. By plotting the FAR and FRR rates using different sensitivities, you can determine the effectiveness of a biometric system.

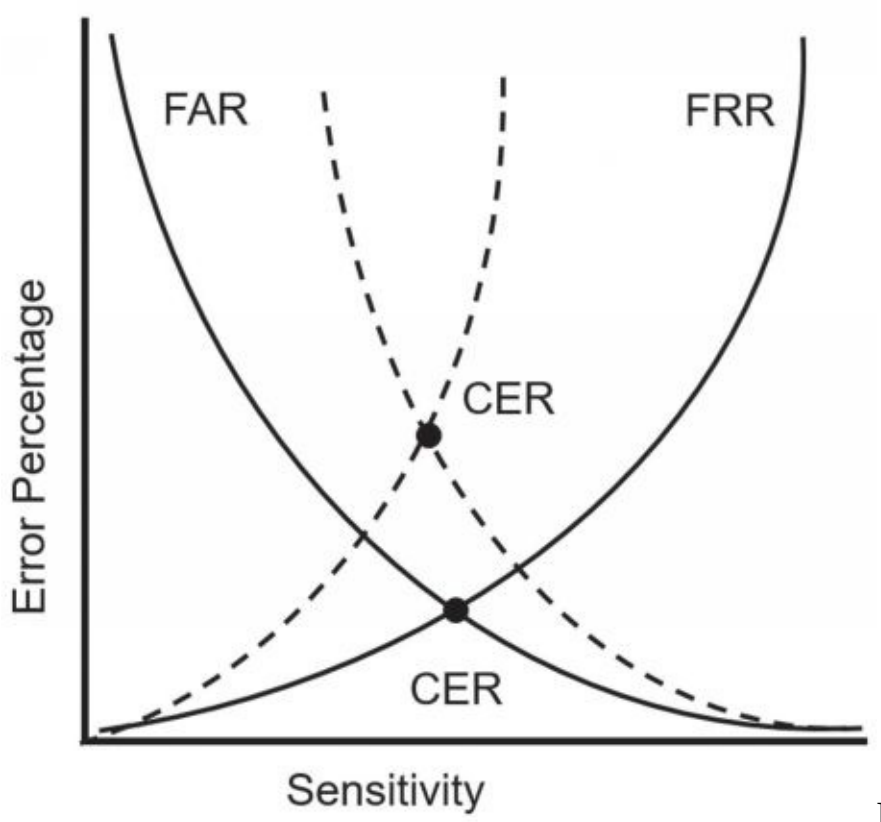


Figure 2.3 shows the ***crossover error rate*** (CER) for two biometric systems. The CER is the point where the FAR crosses over with the FRR. A lower CER indicates that the biometric system is more accurate. For example, the system represented with the solid lines in the figure is more accurate than the system represented by the dotted lines.

Figure 2.3: Crossover error rate

Somewhere You Are

The ***somewhere you are*** authentication factor identifies a user's location. Geolocation is a group of technologies used to identify a user's location and is the most common method used

in this factor. Many authentication systems use the Internet Protocol (IP) address for geolocation. The IP address provides information on the country, region, state, city, and sometimes even the zip code.

Remember this

The third factor of authentication (something you are, defined with biometrics) is the strongest individual method of authentication because it is the most difficult for an attacker to falsify. Biometric methods include fingerprints, retina scans, iris scans, voice recognition, and facial recognition. Iris and retina scans are the strongest biometric methods mentioned in this section, though iris scans are used more than retina scans due to the privacy issues and the scanning requirements. Facial recognition is the most flexible and when using alternate lighting (such as infrared), they might become the most popular. The crossover error rate (CER) measures the accuracy of a system and lower CERs are better.

As an example, I once hired a virtual assistant in India to do some data entry for me. I created an account for the assistant in an online application called Hootsuite and sent him the logon information. However, when he attempted to log on, Hootsuite recognized that his IP was in India but I always logged on from an IP in the United States. Hootsuite blocked his access and then sent me an email saying that someone from India was trying to log on. They also provided me directions on how to grant him access if he was a legitimate user, but it was comforting to know they detected and blocked this access automatically.

It's worth noting that using an IP address for geolocation isn't foolproof. There are many virtual private network (VPN) IP address changers available online. For example, a user in Russia can use one of these services in the United States to access a web site. The web site will recognize the IP address of the VPN service, but won't see the IP address of the user in Russia.

Within an organization, it's possible to use the computer name or the media access control (MAC) address of a system for the somewhere you are factor. For example, in a Microsoft Active Directory domain, you can

configure accounts so that users can only log on to the network through one specific computer. If they aren't at that computer, the system blocks them from logging on at all.

Something You Do

The ***something you do*** authentication factor refers to actions you can take such as gestures on a touch screen. As an example, Microsoft Windows 10 supports picture passwords. Users first select a picture, and then they can add three gestures as their picture password. Gestures include tapping in specific places on the picture, drawing lines between items with a finger, or drawing a circle around an item such as someone's head. After registering the picture and their gestures, users repeat these gestures to log on again later.

Other examples of something you do include how you write or how you type. For example, keystroke dynamics measure the pattern and rhythm as a user types on a keyboard. It measures details such as speed, dwell time, and flight time. Dwell time is the time a key is pressed, and flight time is the time between releasing one key and pressing the next key. Many security professionals refer to this as behavioral biometrics because it identifies behavioral traits of an individual. However, some people put these actions into the something you do authentication factor.

Dual-Factor and Multifactor Authentication

Dual-factor authentication (sometimes called two-factor authentication) uses two different factors of authentication, such as something you have and something you know. Dual-factor authentication often uses a smart card and a PIN, a USB token and a PIN, or combines a smart card or hardware token with a password. In each of these cases, the user must have something and know something.

Multifactor authentication uses two or more factors of authentication. For example, you can combine the something you are factor with one or more other factors of authentication.

Note that technically you can call an authentication system using two different factors either dual-factor authentication or multifactor authentication. Multifactor authentication indicates multiple factors and multiple is simply more than one.

It's worth noting that using two methods of authentication in the same factor is not dual- factor authentication. For example, requiring users to enter a password and a PIN (both in the something you know factor) is single-factor authentication, not dual-factor authentication. Similarly, using a thumbprint and a retina scan is not dual-factor authentication because both methods are in the something you are factor.

Remember this

Using two or more methods in the same factor of authentication (such as a PIN and a password) is single-factor authentication. Dual-factor (or two-factor) authentication uses two different factors, such as using a hardware token and a PIN. Multifactor authentication uses two or more factors.

Summarizing Identification Methods

So far, this chapter has presented several different identification methods and because identification is so important, it's worthwhile to summarize them. They are usernames, photo identification cards, and

biometrics.

The most commonly used identification method is a username. This can be a traditional username, such as DarrilGibson, or it can be an email address, such as Darril@gcgapremium.com, depending on how the system is configured. Many other identification methods can be used for both identification and authentication.

CACs and PIVs include a picture and other information about the owner, so owners often use them for identification. They also function as smart cards in the something you have authentication factor.

The “Something You Are” section focused on using biometrics for authentication, but several entities also use biometric methods for identification. For example, law enforcement agencies have used fingerprints to identify individuals at crime scenes for decades. Similarly, retina and palm scanners can identify individuals with a high degree of accuracy.

Troubleshooting Authentication Issues

Some common authentication issues that can cause security problems have been mentioned in this section. As a summary, they are:

- **Weak passwords.** If users aren’t forced to use strong, complex passwords, they probably won’t and their accounts will be vulnerable to attacks. A technical password policy ensures users implement strong passwords, don’t reuse them, and change them regularly.
- **Forgotten passwords.** An organization needs to have a password recovery procedure in place to help users recover their passwords. If passwords are manually reset without verifying the identity of the user, it’s possible for an attacker to trick someone into resetting the password.
- **Biometric errors.** Weak biometric systems with a high crossover error rate may have a high false match rate (also called a false acceptance rate) or a low nonmatch rate (also called a false rejection rate).

Comparing Authentication Services

Several other authentication services are available that fall outside the scope of the described factors of authentication. A common goal they have is to ensure

that unencrypted credentials are not sent across a network. In other words, they ensure that credentials are not sent in cleartext. If credentials are sent in cleartext, attackers can use tools such as a protocol analyzer to capture and view them. The following sections describe many of these services.

Kerberos

Kerberos is a network authentication mechanism used within Windows Active Directory domains and some Unix environments known as realms. It was originally developed at MIT (the Massachusetts Institute of Technology) for Unix systems and later released as a request for comments (RFC). Kerberos provides mutual authentication that can help prevent man-in-the- middle attacks and uses tickets to help prevent replay attacks. Chapter 7, "Protecting Against Advanced Attacks," covers these attacks in more depth.

Kerberos includes several requirements for it to work properly. They are:

- **A method of issuing tickets used for authentication.** The Key Distribution Center (***KDC***) uses a complex process of issuing ticket-granting tickets (TGTs) and other tickets. The KDC (or TGT server) packages user credentials within a ticket. Tickets provide authentication for users when they access resources such as files on a file server. These tickets are sometimes referred to as tokens, but they are logical tokens, not a key fob type of token discussed earlier in the "Something You Have" section.
- **Time synchronization.** Kerberos version 5 requires all systems to be synchronized and within five minutes of each other. The clock that provides the time synchronization is used to timestamp tickets, ensuring they expire correctly. This helps prevent replay attacks. In a replay attack, a third party attempts to impersonate a client after intercepting data captured in a session. However, if an attacker intercepts a ticket, the timestamp limits the amount of time an attacker can use the ticket.
- **A database of subjects or users.** In a Microsoft environment, this is Active Directory, but it could be any database of users.

When a user logs on with Kerberos, the KDC issues the user a ticket-granting ticket, which typically has a lifetime of 10 hours to be useful for a single workday. When the user tries to access a resource, the ticket-granting

ticket is presented as authentication, and the user is issued a ticket for the resource. However, the ticket expires if users stay logged on for an extended period, such as longer than 10 hours. This prevents them from accessing network resources. In this case, users may be prompted to provide a password to renew the ticket-granting ticket, or they might need to log off and back on to generate a new ticket-granting ticket.

Additionally, Kerberos uses symmetric-key cryptography to prevent unauthorized disclosure and to ensure confidentiality. Chapter 10 explains algorithms in more depth, but in short, symmetric-key cryptography uses a single key for both encryption and decryption of the same data.

Remember this

Kerberos is a network authentication protocol within a Microsoft Windows Active Directory domain or a Unix realm. It uses a database of objects such as Active Directory and a KDC (or TGT server) to issue timestamped tickets that expire after a certain time period.

NTLM

New Technology LAN Manager (***NTLM***) is a suite of protocols that provide authentication, integrity, and confidentiality within Windows systems. At their most basic, they use a Message Digest hashing algorithm to challenge users and check their credentials. There are three versions of NTLM:

- NTLM is a simple MD4 hash of a user's password. MD4 has been cracked and neither NTLM nor MD4 are recommended for use today.
- NTLMv2 is a challenge-response authentication protocol. When a user attempts to log on, NTMLv2 creates an HMAC-MD5 hash composed of a combination of the username, the logon domain name (or computer name), the user's password, the current time, and more. To create an HMAC-MD5 message, authentication code starts as the MD5 hash of a user's password, which is then encrypted.
- NTLM2 Session improves NTLMv2 by adding in mutual authentication. In other words, the client authenticates with the server, and the server also authenticates with the client.

So, which protocol should you select? Actually, Microsoft specifically

recommends that developers don't select one of these protocols. Instead, developers should use the Negotiate security package within their applications. This security package selects the most secure security protocols available between the systems. It first tries to use Kerberos if it is available. If not, it uses either NTLMv2 or NLTM2 Session depending on the capabilities of the systems involved in the session.

LDAP and LDAPS

Lightweight Directory Access Protocol (***LDAP***) specifies formats and methods to query directories. In this context, a directory is a database of objects that provides a central access point to manage users, computers, and other directory objects. LDAP is an extension of the X.500 standard that Novell and early Microsoft Exchange Server versions used extensively.

Windows domains use Active Directory, which is based on LDAP. Active Directory is a directory of objects (such as users, computers, and groups), and it provides a single location for object management. Queries to Active Directory use the LDAP format. Similarly, Unix realms use LDAP to identify objects.

Administrators often use LDAP in scripts, but they need to have a basic understanding of how to identify objects. For example, a user named Homer in the Users container within the GetCertifiedGetAhead.com domain is identified with the following LDAP string:
LDAP://CN=Homer,CN=Users,DC=GetCertifiedGetAhead,DC=com

- **CN=Homer.** CN is short for common name.
- **CN=Users.** CN is sometimes referred to as container in this context.
- **DC=GetCertifiedGetAhead.** DC is short for domain component.
- **DC=com.** This is the second domain component in the domain name.

LDAP Secure (LDAPS) uses encryption to protect LDAP transmissions. When a client connects with a server using LDAPS , the two systems establish a Transport Layer Security (TLS) session before transmitting any data. TLS encrypts the data before transmission.

Remember this

LDAP is based on an earlier version of X.500. Windows Active Directory domains and Unix realms use LDAP to identify objects in query strings with codes such as CN=Users and

DC=GetCertifiedGetAhead. LDAPS encrypts transmissions with TLS.

Single Sign-On

Single sign-on (***SSO***) refers to the ability of a user to log on or access multiple systems by providing credentials only once. SSO increases security because the user only needs to remember one set of credentials and is less likely to write them down. It's also much more convenient for users to access network resources if they only have to log on one time.

As an example, consider a user who needs to access multiple servers within a network to perform normal work. Without SSO, the user would need to know one set of credentials to log on locally, and additional credentials for each of the servers. Many users would write these credentials down to remember them.

Alternatively, in a network with SSO capabilities, the user only needs to log on to the network once. The SSO system typically creates some type of SSO secure token used during the entire logon session. Each time the user accesses a network resource, the SSO system uses this secure token for authentication. Kerberos and LDAP both include SSO capabilities.

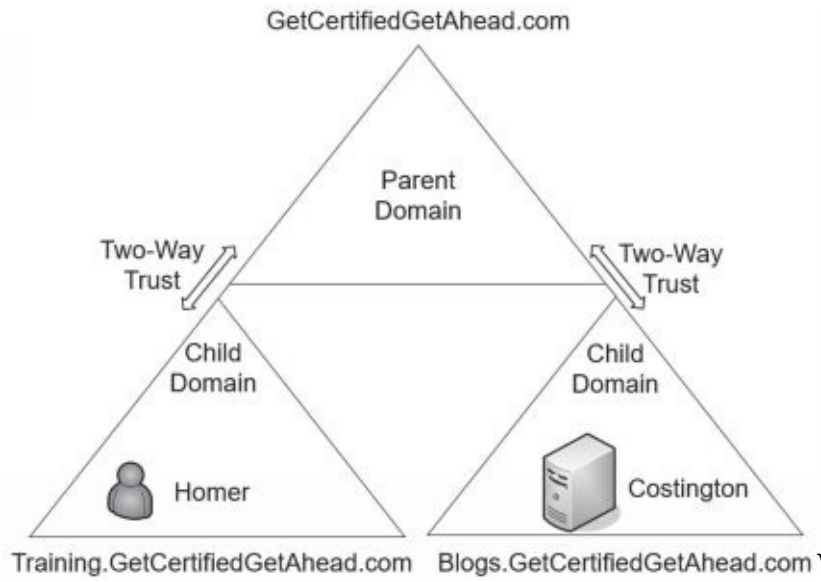
SSO requires strong authentication to be effective. If users create weak passwords, attackers might be able to guess them, giving them access to multiple systems. Some people debate that SSO adds in risks because if an attacker can gain the user's credentials, it gives the attacker access to multiple systems.

SSO and Transitive Trusts

A ***transitive trust*** creates an indirect trust relationship. As an example, imagine a transitive trust relationship exists between Homer, Moe, and Fat Tony:

- Homer trusts Moe.
- Moe trusts Fat Tony.
- Because of the transitive trust relationship, Homer trusts Fat Tony.

Of course, this isn't always true with people and Homer might be a little upset with Moe if Moe shares Homer's secrets with Fat Tony. However, it reduces network administration in a domain.



Within an LDAP-based network, domains use transitive trusts for SSO. Figure 2.4 shows a common configuration with three domains in the same network. The parent domain is GetCertifiedGetAhead.com and the configuration includes two child domains—Training and Blogs.

Figure 2.4: An LDAP transitive trust used for SSO

In this example, there is a two-way trust between the parent domain

(GetCertifiedGetAhead.com) and the child domain (Training.GetCertifiedGetAhead.com). The parent trusts the child, and the child trusts the parent. Similarly, there is a two-way trust between the parent domain and the Blogs child domain. There isn't a direct trust between the two child domains. However, the transitive relationship creates a two-way trust between them.

All of these domains contain objects, such as users, computers, and groups. Homer's user account is in the Training domain, and a server named Costington is in the Blogs domain. With the transitive trust, it's possible to grant Homer access to the Costington server without creating another trust relationship directly between the Training and Blogs domains.

Without a trust relationship, you'd have to create another account for Homer in the Blogs domain before you could grant him access. Additionally, Homer would need to manage the second account's password separately. However, with the transitive trust relationships, the network supports SSO, so Homer only needs a single account.

SSO and SAML

Security Assertion Markup Language (***SAML***) is an Extensible Markup Language (***XML***)– based data format used for SSO on web browsers. Imagine two web sites hosted by two different organizations. Normally, a user would have to provide different credentials to access either web site. However, if the organizations trust each other, they can use SAML as a federated identity management system. Users authenticate with one web site and are not required to authenticate again when accessing the second web site.

Many web-based portals use SAML for SSO. The user logs on to the portal once, and the portal then passes proof of the user's authentication to back-end systems. As long as one organization has authenticated users, they are not required to authenticate again to access other sites within the portal.

SAML defines three roles:

- **Principal.** This is typically a user. The user logs on once. If necessary, the principal requests an identity from the identity provider.
- **Identity provider.** An identity provider creates, maintains, and manages identity information for principals.
- **Service provider.** A service provider is an entity that provides services to principals. For example, a service provider could host one

or more web sites accessible through a web- based portal. When a principal tries to access a resource, the service provider redirects the principal to obtain an identity first.

This process sends several XML-based messages between the systems. However, it is usually transparent to the user.

SAML and Authorization

It's important to realize that the primary purpose of SSO is for identification and authentication of users. Users claim an identity and prove that identity with credentials. SSO does not provide authorization. For example, if the power plant and the school system create a federation using SAML, this doesn't automatically grant everyone in the school system full access to the nuclear power plant resources. Authorization is completely separate.

However, many federation SSO systems, including SAML, include the ability to transfer authorization data between their systems. In other words, it's possible to use SAML for single sign-on authentication and for authorization.

Remember this

SAML is an XML-based standard used to exchange authentication and authorization information between different parties. SAML provides SSO for web-based applications.

SSO and a Federation

Some SSO systems can connect authentication mechanisms from different environments, such as different operating systems or different networks. One common method is with a federated identity management system, often integrated as a federated database. This federated database provides central authentication in a nonhomogeneous environment.

As an example, imagine that the Springfield Nuclear Power Plant established a relationship with the Springfield school system, allowing the power plant employees to access school resources. It's not feasible or desirable to join these two networks into one. However, you can create a federation of the two networks. Once it's established, the power plant employees will log on using their power plant account, and then access the

shared school resources without logging on again.

A ***federation*** requires a federated identity management system that all members of the federation use. In the previous example, the members of the federation are the power plant and the school system. Members of the federation agree on a standard for federated identities and then exchange the information based on the standard. A federated identity links a user's credentials from different networks or operating systems, but the federation treats it as one identity.

Shibboleth is one of the federated identity solutions mentioned specifically in the CompTIA Security+ exam objectives. It is open source and freely available, making it a more affordable solution than some of the commercially available federated identity solutions. It also includes Open SAML libraries written in C++ and Java, making it easier for developers to expand its usefulness.

OAuth and OpenID Connect

OAuth is an open standard for authorization many companies use to provide secure access to protected resources. Instead of creating a different account for each web site you access, you can often use the same account that you've created with Google, Facebook, PayPal, Microsoft, or Twitter.

As an example, imagine that the Try-N-Save Department Store decides to sell some of its products online and management has decided to allow customers to make purchases through PayPal. Developers configure their web site to exchange application programming interface (API) calls between it and PayPal servers. Now, when customers make a purchase, they log on with their PayPal account and make their purchase through PayPal. OAuth transfers data between PayPal and the Try-N-Save site so that the department store receives the money and knows what to ship to the customer. A benefit for the customers is that they don't have to create another account for Try-N-Save.

OpenID Connect works with OAuth 2.0 and it allows clients to verify the identity of end users without managing their credentials. In this context, the client is typically a web site or application that needs to authenticate users. OpenID Connect provides identification services, without requiring the application to handle the credentials. It also streamlines the user experience for users. For example, Skyscanner is an application for finding flights, hotels, and car rentals. It allows users to sign in using their Facebook credentials. After doing so, Skyscanner provides a more personalized

experience for the users.

Managing Accounts

Account management is concerned with the creation, management, disablement, and termination of accounts. When the account is active, access control methods are used to control what the user can do. Additionally, administrators use access controls to control when and where users can log on. The following sections cover common account management practices, along with some basic principles used with account management. Improperly configured accounts don't follow these principles, increasing risks.

Least Privilege

The principle of ***least privilege*** is an example of a technical control implemented with access controls. Privileges are the rights and permissions assigned to authorized users. Least privilege specifies that individuals and processes are granted only the rights and permissions needed to perform assigned tasks or functions, but no more. For example, if Lisa needs read access to a folder on a server, you should grant her read access to that folder, but nothing else.

A primary goal of implementing least privilege is to reduce risks. As an example, imagine that Carl works at the Nuclear Power Plant, but administrators have improperly configured accounts ignoring the principle of least privilege. In other words, Carl has access to all available data within the Nuclear Power Plant, not just the limited amount of data he needs to perform his job. Later, Lenny gets into trouble and needs money, so he convinces Carl to steal data from the power plant so that they can sell it. In this scenario, Carl can steal and sell all the data at the plant, which can result in serious losses.

In contrast, if administrators applied the principle of least privilege, Carl would only have access to a limited amount of data. Even if Lenny convinces him to steal the data, Carl wouldn't be able to steal very much simply because he doesn't have access to it. This limits the potential losses for the power plant.

This principle applies to regular users and administrators. As an example, if Marge administers all the computers in a training lab, it's

appropriate to give her administrative control over all these computers. However, her privileges don't need to extend to the domain, so she wouldn't have administrative control over all the computers in a network. Additionally, she wouldn't have the privileges required to add these computers to the domain, unless that was a requirement in the training lab. Similarly, if a network administrator needs to review logs and update specific network devices, it's appropriate to give the administrator access to these logs and devices, but no more.

Many services and applications run under the context of a user account. These services have the privileges of this user account, so it's important to ensure that these accounts are only granted the privileges needed by the service or the application. In the past, many administrators configured these service and application accounts with full administrative privileges. When attackers compromised a service or application configured this way, they gained administrative privileges and wreaked havoc on the network.

Remember this

Least privilege is a technical control. It specifies that individuals or processes are granted only those rights and permissions needed to perform their assigned tasks or functions.

Need to Know

The principle of need to know is similar to the principle of least privilege in that users are granted access only to the data and information that they need to know for their job. Notice that need to know is focused on data and information, which is typically protected with permissions. In contrast, the principle of least privilege includes both rights and permissions.

Rights refer to actions and include actions such as the right to change the system time, the right to install an application, or the right to join a computer to a domain. Permissions typically refer to permissions on files, such as read, write, modify, read & execute, and full control.

Account Types

When managing accounts, it's important to recognize the common types of accounts used within a network. They are:

- **End user accounts.** Most accounts are for regular users. Administrators create these accounts and then assign appropriate privileges based on the user's job responsibilities. Microsoft refers to this as a Standard user account.
- **Privileged accounts.** A ***privileged account*** has additional rights and privileges beyond what a regular user has. As an example, someone with administrator privileges on a Windows computer has full and complete control over the Windows computer.
- **Guest accounts.** Windows operating systems include a ***Guest account***. These are useful if you want to grant someone limited access to a computer or network without creating a new account. For example, imagine an organization contracts with a temp agency to have someone do data entry. It's possible that the agency sends a different person every day. Enabling the Guest account for this person would be simpler than creating a new account every day. Administrators commonly disable the Guest account and only enable it in special situations.
- **Service accounts.** Some applications and services need to run under the context of an account and a ***service account*** fills this need. As an example, SQL Server is a database application that runs on a server and it needs access to resources on the server and the network. Administrators create a regular user account, name it something like sqlservice, assign it appropriate privileges, and configure SQL Server to use this account. Note that this is like a regular end-user account. The only difference is that it's only used by the service or application, not an end user.

One of the challenges with service accounts is that they often aren't managed. For example, imagine a regular user account that has a password that expires after 45 days. The user is notified to change the password and the user does so. A service account might send a notification to the application to change the password, but the notification is ignored. When the password expires, the account is locked. Suddenly, the application (or service) stops working and administrators have to troubleshoot the issue to figure out why.

A solution is to configure the service account so that it doesn't have to comply with the password policy. However, this allows the service account to ignore other policy requirements such as using a strong, complex password. It's important that developers using these types of accounts take steps to

ensure their accounts follow existing policies.

Require Administrators to Use Two Accounts

It's common to require administrators to have two accounts. They use one account for regular day-to-day work. It has the same limited privileges as a regular end user. The other account has elevated privileges required to perform administrative work, and they use this only when performing administrative work. The benefit of this practice is that it reduces the exposure of the administrative account to an attack.

For example, when malware infects a system, it often attempts to gain additional rights and permissions using privilege escalation techniques. It may exploit a bug or flaw in an application or operating system. Or, it may simply assume the rights and permissions of the logged-on user. If an administrator logs on with an administrative account, the malware can assume these elevated privileges. In contrast, if the administrator is logged on with a regular standard user account, the malware must take additional steps to escalate its privileges.

This also reduces the risk to the administrative account for day-to-day work. Imagine Homer is an administrator and he's called away to a crisis. It is very possible for him to walk away without locking his computer. If he was logged on with his administrator account, an attacker walking by can access the system and have administrative privileges. Although systems often have password-protected screen savers, these usually don't start until about 10 minutes or longer after a user walks away.

Standard Naming Convention

It's common for an organization to adopt a standard naming convention to ensure user account names and email addresses are created similarly. For example, one convention uses the first name, a dot, and the last name. This creates accounts like homer.simpson and bart.simpson. If the organization hires a second person with the same name, such as a second Bart Simpson, the naming convention might specify adding a

number to the name, such as bart.simpson2.

You probably won't need to design a naming convention. However, if you start with a different organization and you need to create accounts, you should understand that the organization probably has a naming convention in place. You should follow the convention for any accounts you create.

Prohibiting Shared and Generic Accounts

Account management policies often dictate that personnel should not use shared or generic accounts. Instead, each user has at least one account, which is only accessible to that user. If multiple users share a single account, you cannot implement basic authorization controls. As a reminder, four key concepts are:

- **Identification.** Users claim an identity with an identifier such as a username.
- **Authentication.** Users prove their identity using an authentication method such as a password.
- **Authorization.** Users are authorized access to resources, based on their proven identity.
- **Accounting.** Logs record activity using the users' claimed identity.

Imagine that Bart, Maggie, and Lisa all used a Guest account. If you want to give Lisa access to certain files, you'd grant access to the Guest account, but Bart and Maggie would have the same access. If Bart deleted the files, logs would indicate the Guest account deleted the files, but you wouldn't know who actually deleted them. In contrast, if users have unique user accounts, you can give them access to resources individually. Additionally, logs would indicate exactly who took an action.

Note that having a single, temporary user log on with the Guest account does support identification, authentication, authorization, and accounting. It is only when multiple users are sharing the same account that you lose these controls. Still, some organizations prohibit the use of the Guest account for any purposes.

Remember this

Requiring administrators to use two accounts, one with administrator privileges and another with regular user privileges, helps prevent privilege escalation attacks. Users should not use shared accounts.

Disablement Policies

Many organizations have a ***disablement policy*** that specifies how to manage accounts in different situations. For example, most organizations require administrators to disable user accounts as soon as possible when employees leave the organization. Additionally, it's common to disable default accounts (such as the Guest account mentioned previously) to prevent them from being used.

Disabling is preferred over deleting the account, at least initially. If administrators delete the account, they also delete any encryption and security keys associated with the account. However, these keys are retained when the account is disabled. As an example, imagine that an employee encrypted files with his account. The operating system uses cryptography keys to encrypt and decrypt these files. If administrators deleted this account, these files may remain encrypted forever unless the organization has a key escrow or recovery agent that can access the files.

Some contents of an account disablement policy include:

- **Terminated employee.** An account disablement policy specifies that accounts for ex-employees are disabled as soon as possible. This ensures a terminated employee doesn't become a disgruntled ex-employee who wreaks havoc on the network. Note that "terminated" refers to both employees who resign and employees who are fired.
- **Leave of absence.** If an employee will be absent for an extended period, the account should be disabled while the employee is away. Organizations define extended period differently, with some organizations defining it as only two weeks, whereas other organizations extend it out to as long as two months.
- **Delete account.** When the organization determines the account is no longer needed, administrators delete it. For example, the policy may direct administrators to delete accounts that have been inactive for 60 or 90 days.

Remember this

An account disablement policy identifies what to do with accounts for employees who leave permanently or on a leave of absence. Most policies require administrators to disable the account as soon as possible, so that ex-employees cannot use the account. Disabling the account ensures that data associated with it remains available. Security keys associated with an account remain available when the account is disabled, but are no longer accessible if the account is deleted.

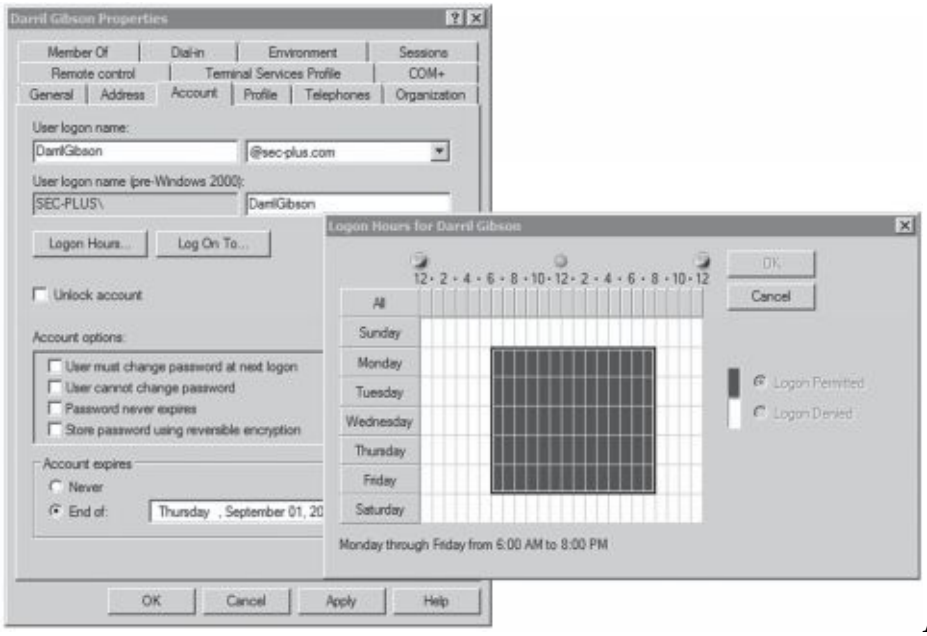
Recovering Accounts

In some situations, administrators need to recover accounts. The two primary account recovery scenarios are:

- **Enable a disabled account.** Administrators can reset the user's password and take control of the account. Similarly, they pass control of the account to someone else, such as a supervisor or manager of an ex-employee. Administrators reset the user's password, set it to expire on first use, and then give the password to the other person.
- **Recover a deleted account.** It is also possible to recover a deleted account. This is more complex than simply creating another account with the same name. Instead, administrators follow detailed procedures to recover the account.

Time-of-Day Restrictions

Time-of-day restrictions specify when users can log on to a computer. If a user tries to log on to the network outside the restricted time, the system denies access to the user.



As an example, imagine a company operates between 8:00 a.m. and 5:00 p.m. on a daily basis. Managers decide they don't want regular users logging on to the network except between 6:00 a.m. and 8:00 p.m., Monday through Friday. You could set time-of-day restrictions for user accounts, as shown in Figure 2.5. If a user tries to log on outside the restricted time (such as during the weekend), the system prevents the user from logging on.

Figure 2.5: User account properties with time restrictions

If users are working overtime on a project, the system doesn't log them off when the restricted time arrives. For example, if Maggie is working late on a Wednesday night, the system doesn't log her off at 8:00 p.m. (assuming the time restrictions are set as shown in Figure 2.5). However, the system will prevent her from creating any new network connections.

Location-Based Policies

Location-based policies restrict access based on the location of the user. The "Somewhere You Are" section earlier in this chapter discussed common methods used to enforce this. For example, geolocation technologies can often detect a location using the IP address, and block any traffic from unacceptable addresses, such as from foreign countries. It's also possible to identify a set of IP addresses as the only addresses that are acceptable. This is often referred to as whitelisting the IP addresses.

Within a network, it's possible to restrict access based on computer names

or MAC addresses. For example, imagine Bart has been logging on to multiple computers with his account. It is possible to restrict his account to only his computer. When he tries to log on to his account, he is successful. If he tries to log on to another computer, the location-based policy blocks him.

Remember this

Time-of-day restrictions prevent users from logging on during restricted times. They also prevent logged-on users from accessing resources during certain times. Location- based policies restrict access based on the location of the user.

Expiring Accounts and Recertification

It's possible to set user accounts to expire automatically. When the account expires, the system disables it, and the user is no longer able to log on using the account.

If you look back at Figure 2.5, it shows the properties of an account. The Account Expires section is at the bottom of the page, and the account is set to expire on September 1. When September 1 arrives, the account is automatically disabled and the user will no longer be able to log on.

It's common to configure temporary accounts to expire. For example, an organization may hire contractors for a 90-day period to perform a specific job. An administrator creates accounts for the contractors and sets them to expire in 90 days. This automatically disables the accounts at the end of the contract.

If the organization extends the contract, it's a simple matter to recertify the account. Administrators verify that the contract has been extended, change the expiration date, and enable the account.

Remember this

Account expiration dates automatically disable accounts on the expiration date. This is useful for temporary accounts such as temporary contractors.

Account Maintenance

Administrators routinely perform account maintenance. This is often done with scripts to automate the processes.

As an example, it's relatively simple to create and run a script listing all enabled accounts that haven't been used in the last 30 days in a Microsoft AD DS domain. This provides a list of inactive accounts. Often, these are accounts of ex-employees or temporary employees who are no longer at the organization. Ideally, an account disablement policy would ensure that the accounts are disabled as soon as the employee leaves. The scripts provide an additional check to ensure inactive accounts are disabled.

Additionally, account maintenance includes deleting accounts that are no longer needed. For example, if an organization has a policy of disabling accounts when employees leave, but deleting them 60 days later, account maintenance procedures ensure the accounts are deleted.

Credential Management

A credential is a collection of information that provides an identity (such as a username) and proves that identity (such as with a password). Over time, users often have multiple credentials that they need to remember, especially when they access many web sites. Credential management systems help users store these credentials securely. The goal is to simplify credential management for users, while also ensuring that unauthorized personnel do not have access to the users' credentials.

As an example of a credential management system, Windows 10 includes the Credential Manager, accessible from Control Panel. Users are able to add credentials into the Credential Manager, which stores them securely in special folders called vaults. Then, when users access web sites needing credentials, the system automatically retrieves the credentials from the vault and submits them to the web site.

Similarly, web browsers such as Google Chrome use a credential management system to remember passwords. When you access a web site that needs your password, Chrome prompts you asking if you'd like Chrome to remember it. Later, when you visit the same web site, Chrome fills in the credentials for you.

Comparing Access Control Models

Access control ensures that only authenticated and authorized entities can access resources. For example, it ensures that only authenticated users who have been granted appropriate permissions can access files on a server. This starts by ensuring that users are accurately identified and authenticated. Then, you grant access using one of several different models. The models covered in this section are:

- Role-based access control (role-BAC)
- Rule-based access control (rule-BAC)
- Discretionary access control (DAC)
- Mandatory access control (MAC)
- Attribute-based access control (ABAC)

You might notice that CompTIA uses the acronym RBAC for both rule-based access control and role-based access control. For clarity, this book uses role-BAC or rule-BAC instead of the ambiguous RBAC.

By understanding a little more of the underlying design principles, you'll understand why some of the rules are important, and you'll be better prepared to ensure that security principles are followed.

Often, when using any of the models, you'll run across the following terms:

- **Subjects.** Subjects are typically users or groups that access an object. Occasionally, the subject may be a service that is using a service account to access an object.
- **Objects.** Objects are items such as files, folders, shares, and printers that subjects access. For example, users access files and printers. The access control helps determine how a system grants authorization to objects. Or, said another way, the access control model determines how a system grants users access to files and other resources.

Role-Based Access Control

Role-based access control (***role-BAC***) uses roles to manage rights and permissions for users. This is useful for users within a specific department who perform the same job functions. An administrator creates the roles and then assigns specific rights and permissions to the roles (instead of to the users). When an administrator adds a user to a role, the user has all the rights and permissions of that role.

Using Roles Based on Jobs and Functions

Imagine your organization has several departments, such as Accounting, Sales, and IT, and each department has a separate server hosting its files. You can create roles of Accounting, Sales, and IT and assign these roles to users based on the department where they work. Next, you'd grant these roles access to the appropriate server. For example, you'd grant the Accounting role to the Accounting server, grant the Sales role to the Sales server, and so on.

Another example of the role-BAC model is Microsoft Project Server. The Project Server can host multiple projects managed by different project managers. It includes the following roles:

- **Administrators.** Administrators have complete access and control over everything on the server, including all of the projects managed on the server.
- **Executives.** Executives can access data from any project held on the server, but do not have access to modify system settings on the server.
- **Project Managers.** Project managers have full control over their own projects, but do not have any control over projects owned by other project managers.
- **Team Members.** Team members can typically report on work that project managers assign to them, but they have little access outside the scope of their assignments.

Microsoft Project Server includes more roles, but you can see the point with these four. Each of these roles has rights and permissions assigned to it, and to give someone the associated privileges, you'd simply add the user's account to the role.

Documenting Roles with a Matrix

Think about the developers of Microsoft Project Server. They didn't just start creating roles. Instead, they did some planning and identified the roles they envisioned in the application. Next, they identified the privileges each of these roles required. It's common to document role-based permissions with a matrix listing all of the job titles and the privileges for each role, as shown in Table 2.1.

Role	Server Privileges	Project Privileges
Administrators	All	All
Executives	None	All
Project Managers	None	All on assigned projects No access on unassigned projects
Team Members	None	Access for assigned tasks Limited views within scope of their assigned tasks No views outside the scope of their assigned tasks

Table 2.1: Role-BAC matrix for Project Server

Role-BAC is also called hierarchy-based or job-based:

- **Hierarchy-based.** In the Project Server example, you can see how top-level roles, such as the Administrators role, have significantly more permissions than lower-level roles, such as the Team Members role. Roles may mimic the hierarchy of an organization.
- **Job-, task-, or function-based.** The Project Server example also shows how the roles are centered on jobs or functions that users need to perform.

Remember this

A role-BAC model uses roles based on jobs and functions. A matrix is a planning document that matches the roles with the required privileges.

Establishing Access with Group-Based Privileges

Administrators commonly grant access in the role-BAC model using roles, and they often implement roles as groups. Windows systems refer to these as security groups. They assign rights and permissions (privileges) to groups and then add user accounts to the appropriate group. This type of ***group-based access control***, where access is based on roles or groups, simplifies user administration.

One implementation of the role-BAC model is the Microsoft built-in security groups and specially created security groups that administrators create on workstations, servers, and within domains.

The Administrators group is an example of a built-in security group. For example, the Administrators group on a local computer includes all of the rights and permissions on that computer. If you want to grant Marge full and complete control to a computer, you could add Marge's user account to the Administrators group on that computer. Once Marge is a member of the

Administrators group, she has all the rights and permissions of the group.

Similarly, you can grant other users the ability to back up and restore data by adding their user accounts to the Backup Operators group. Although the built-in groups are very useful, they don't meet all the requirements in most organizations. For example, if your organization wants to separate backup and restore responsibilities, you can create one group that can only back up data and another group that can only restore data.

In Windows domains, administrators often create groups that correspond to the departments of an organization. For example, imagine that Homer, Marge, and Bart work in the Sales department and need to access data stored in a shared folder named Sales on a network server. An administrator would simplify administration with the following steps, as shown in Figure 2.6:

1. Create a Sales group and add each of the user accounts to the Sales group.
2. Add the Sales group to the Sales folder.
3. Assign appropriate permissions to the Sales group for the Sales folder.

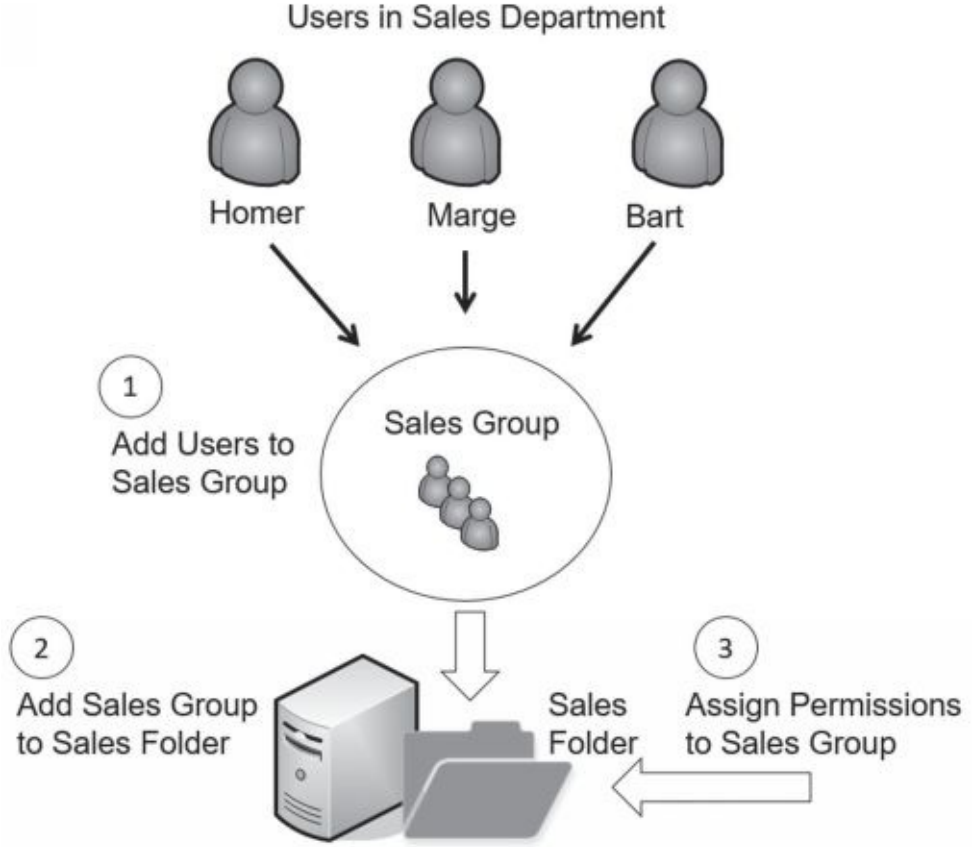


Figure 2.6: Establishing access with groups as roles

If the company adds new salespeople, the administrator creates accounts for them and places their accounts into the Sales group. These new salespeople now have access to everything assigned to this group. If any users change jobs within the company and leave the Sales department, the administrator removes them from the Sales group. This automatically prevents them from accessing any resources granted to the Sales group. This example shows how to use a group for the Sales department, but you can apply the same steps to any department or group of users.

Without groups, you would use user-assigned privileges. In other words, you would assign all the specific rights and permissions for every user individually. This might work for one or two users, but quickly becomes unmanageable with more users.

As an example, imagine that people within the Sales department need access to 10 different resources (such as files, folders, and printers) within a network. When the company hires a new salesperson, you'd need to assign permissions to these 10 different resources manually, requiring 10 different administrative tasks. If you assign the permissions to the Sales group, you only need to add the new user to one group and you're done.

Groups provide another security benefit. Imagine that a user is promoted out of the Sales department and now works in Marketing. If you have a Marketing group, you can place this user account into the Marketing group and remove the account from the Sales group. Removing the user from the Sales group instantly removes all the user rights and permissions applied from that group. However, if you're not using groups and assign permissions to users directly, you probably won't remember which resources were assigned to the user as a member of the Sales department. Instead, the user will continue to have access to this sales data, violating the principle of least privilege.

Remember this

Group-based privileges reduce the administrative workload of access management. Administrators put user accounts into security groups, and assign privileges to the groups. Users within a group automatically inherit the privileges assigned to the group.

Rule-Based Access Control

Rule-based access control (***rule-BAC***) uses rules. The most common example is with rules in routers or firewalls. However, more advanced implementations cause rules to trigger within applications, too.

Routers and firewalls use rules within access control lists (ACLs). These rules define the traffic that the devices allow into the network, such as allowing Hypertext Transfer Protocol (HTTP) traffic for web browsers. These rules are typically static. In other words, administrators create the rules and the rules stay the same unless an administrator changes them again.

However, some rules are dynamic. For example, intrusion prevention systems can detect attacks, and then modify rules to block traffic from an attacker. In this case, the attack triggers a change in the rules.

As another example, it's possible to configure user applications with rules. For example, imagine you want to give Homer additional permissions to a database if Marge is absent. You can configure a database rule to trigger a change to these permissions when the system recognizes that Marge is absent.

Remember this

Rule-based access control is based on a set of approved instructions, such as an access control list. Some rule-BAC systems use rules that trigger in response to an event, such as modifying ACLs after detecting an attack or granting additional permissions to a user in certain situations.

Discretionary Access Control

In the discretionary access control (***DAC***) model, every object (such as files and folders) has an owner, and the owner establishes access for the objects. Many operating systems, such as Windows and most Unix-based systems, use the DAC model.

A common example of the DAC model is the New Technology File System (NTFS) used in Windows. NTFS provides security by allowing users and administrators to restrict access to files and folders with permissions. NTFS is based on the DAC model and the following section explains how it uses the DAC model.

SIDs and DACLs

Microsoft systems identify users with security identifiers (SIDs), though you will rarely see a SID. A SID is a long string of characters that is meaningless to most people and may look like this: S-1-5-21-3991871189-223218. Instead of the system displaying the SID, it looks up the name associated with the SID and displays the name. Similarly, Microsoft systems identify groups with a SID.

Every object (such as a file or folder) includes a discretionary access control list (DACL) that identifies who can access it in a system using the DAC model. The DACL is a list of Access Control Entries (ACEs). Each ACE is composed of a SID and the permission(s) granted to the SID. As an example, a folder named Study Notes might have the following permissions assigned:

- Lisa: Full Control
- Bart: Read
- Maggie: Modify

Each of these entries is an ACE and combined they are a DACL. The Viewing a DACL Lab shows how to view the DACL for a folder. You can access the online exercises for this book at *http://gcgapremium.com/501labs/.*

The Owner Establishes Access

If users create a file, they are designated as the owner and have explicit control over the file. As the owner, users can modify the permissions on the object by adding user or group accounts to the DACL and assigning the desired permissions.

The DAC model is significantly more flexible than the MAC model described in the next section. MAC has predefined access privileges, and the administrator is required to make the changes. With DAC, if you want to grant another user access to a file you own, you simply make the change, and that user has access.

Remember this

The DAC model specifies that every object has an owner, and the owner has full, explicit control of the object. Microsoft NTFS uses the DAC model.

Beware of Trojans

An inherent flaw associated with the DAC model is the susceptibility to Trojan horses. Chapter 6, “Comparing Threats, Vulnerabilities, and Common Attacks,” presents malware in much more depth, but for this discussion, you should understand some basics related to Trojan horses. Trojan horses are executable files. They masquerade as something useful, but they include malware. For example, Bart might decide to download and install a program that a friend raved about. After installation, he decides it’s not so great and forgets about it. However, the damage is done.

What really happened? When Bart installed the program, it also installed malware. Moreover, if Bart was logged on with administrative privileges when he installed it, the Trojan is able to run with these administrative privileges.

Many organizations require administrators to have two accounts to mitigate the risks associated with Trojans. Most of the time, administrators log on with a regular user account. If the system is infected with malware, the malware has limited permissions assigned to the regular user account. In contrast, if the system is infected with malware while the administrator is logged on with an administrative account, the malware has the elevated permissions of an administrator.

Mandatory Access Control

The mandatory access control (***MAC***) model uses labels (sometimes referred to as sensitivity labels or security labels) to determine access. Security administrators assign labels to both subjects (users) and objects (files or folders). When the labels match, the system can grant a subject access to an object. When the labels don’t match, the access model blocks access.

Military units make wide use of this model to protect data. You might have seen movies where they show a folder with a big red and black cover page labeled “Top Secret.” The cover page identifies the sensitivity label for the data contained within the folder. Users with a Top Secret label (a Top Secret clearance) and a need to know can access the data within the Top Secret folder.

Need to know is an important concept to understand. Just because

individuals have a Top Secret clearance doesn't mean they should automatically have access to all Top Secret data. Instead, access is restricted based on a need to know.

Security-enhanced Linux (SELinux) is one of the few operating systems using the mandatory access control model. SELinux was specifically created to demonstrate how mandatory access controls can be added to an operating system. In contrast, Windows operating systems use the discretionary access control model.

Labels and Lattice

The MAC model uses different levels of security to classify both the users and the data. These levels are defined in a lattice. The lattice can be a complex relationship between different ordered sets of labels. These labels define the boundaries for the security levels.

Figure 2.7 shows how the MAC model uses a lattice to divide access into separate compartments based on a need to know. The lattice starts by defining different levels of Top Secret, Secret, Confidential, and For Official Use. Each of these labels defines specific security boundaries. Within these levels, the lattice defines specific compartments. For example, the Top Secret level includes compartments labeled Nuclear Power Plant, 007, and Happy Sumo.

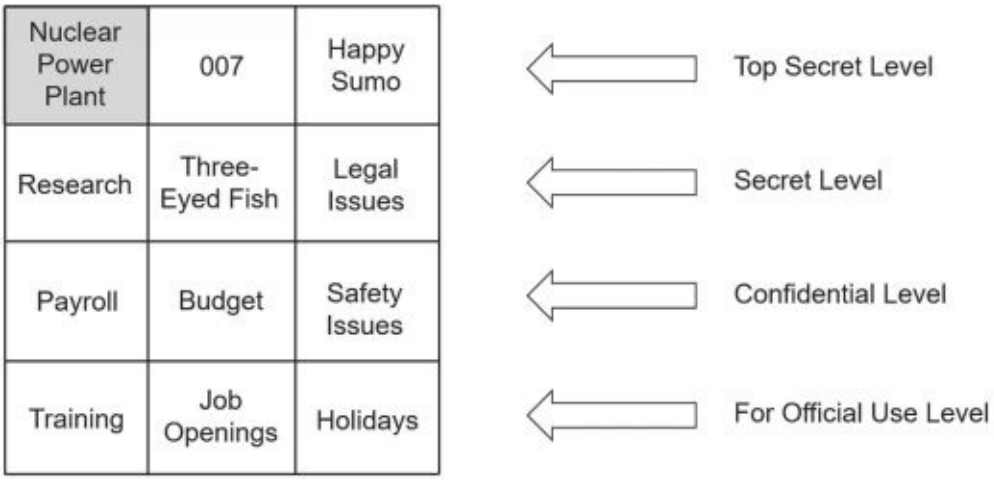


Figure 2.7: MAC model lattice

Imagine that Homer has a Top Secret clearance with a Nuclear Power Plant label. This gives him access to data within the Nuclear Power Plant compartment. However, he does not have access to data in the 007 or Happy Sumo compartment unless he also has those clearances (and associated

labels).

Higher-level clearances include lower-level clearances. For example, because Homer has a Top Secret clearance, he can be granted access to Secret and lower-level data. Again though, he will only be able to access data on these lower levels based on his need to know.

As another example, imagine that Lisa has a Secret level clearance. Administrators can grant her access to data on the Secret level and lower levels, based on her need to know. For example, they might grant her access to the Research data by assigning the Research label to her, but not necessarily grant her access to Three-eyed Fish or Legal Issues data. However, they cannot grant her access to any data on the Top Secret level.

Remember this

The MAC model uses sensitivity labels for users and data. It is commonly used when access needs to be restricted based on a need to know. Sensitivity labels often reflect classification levels of data and clearances granted to individuals.

Establishing Access

An administrator is responsible for establishing access, but only someone at a higher authority can define the access for subjects and objects.

Typically, a security professional identifies the specific access individuals are authorized to access. This person can also upgrade or downgrade the individuals' access, when necessary. Note that the security professional does all this via paperwork and does not assign the rights and permissions on computer systems. Instead, the administrator assigns the rights based on the direction of the security professional.

Multiple approval levels are usually involved in the decision-making process to determine what a user can access. For example, in the military an officer working in the security professional role would coordinate with higher-level government entities to upgrade or downgrade clearances. These higher-level entities approve or disapprove clearance requests.

Once an individual is formally granted access, a network administrator would be responsible for establishing access based on the clearances identified by the security professional. From the IT administrator's point of view, all the permissions and access privileges are predefined.

If someone needed different access, the administrator would forward the request to the security professional, who may approve or disapprove the request. On the other hand, the security professional may forward the request to higher entities based on established procedures. This process takes time and results in limited flexibility.

Attribute-Based Access Control

An attribute-based access control (***ABAC***) evaluates attributes and grants access based on the value of these attributes. Attributes can be almost any characteristic of a user, the environment, or the resource. ABAC uses policies to evaluate attributes and grant access when the system detects a match in the policy.

As a simple example, Homer is a Nuclear Safety Inspector at the Springfield Nuclear Power Plant. His user account may be defined with the following attributes: employee, inspector, and nuclear aware. A file server at the plant includes a share called Inspector and it holds documents commonly used by nuclear safety inspectors. An ABAC policy for the share might grant access to the share for any subjects that have the attributes of employee, inspector, and nuclear aware. Many software defined networks (SDNs) use ABAC models. Instead of rules on physical routers, policies in the ABAC system control the traffic. These policies typically use plain language statements. For example, an ABAC policy rule for a company that employs researchers might be

"Allow logged-on researchers to access research sites via the main network." Policy statements typically include four elements:

- **Subject.** This is typically a user. You can use any user property as an attribute such as employment status, group memberships, job roles, logged-on status, and more. In the example, the subject is identified as being logged on and a member of a researchers group.
- **Object.** This is the resource (such as a file, database, or application) that the user is trying to access. In the example, the object is research sites. The research sites object would include Internet access via a proxy server along with a specific list of URLs of research sites.
- **Action.** The action is what the user is attempting to do, such as

reading or modifying a file, accessing specific web sites, and accessing web site applications. The example allows access to specific web sites.

- **Environment.** The environment includes everything outside of the subject and object attributes. This is often referred to as the context of the access request. It can include the time, location, protocols, encryption, devices, and communication method. In the example, it specifies the main network as an environmental attribute.

An ABAC system has a lot of flexibility and can enforce both a DAC and a MAC model. There are also many similarities between the ABAC model and the DAC and MAC models. In the DAC model, owners have control over the access and in an ABAC model, owners can create policies to grant access. The MAC model uses labels assigned to both subjects and objects and grants access when the labels match. The ABAC model uses attributes that identify both subjects and objects, and grants access when a policy identifies a match.

If you want to dig into the ABAC model a little more, check out NIST SP 800-162, "Guide to Attribute Based Access Control (ABAC) Definition and Considerations."

Remember this

The ABAC model uses attributes defined in policies to grant access to resources. It's commonly used in software defined networks (SDNs).

Chapter 2 Exam Topic Review

When preparing for the exam, make sure you understand these key concepts covered in this chapter.

Exploring Authentication Concepts

- Authentication allows entities to prove their identity by using credentials known to another entity.
- Identification occurs when a user claims or professes an identity, such as with a username, an email address, a PIV card, or by using biometrics.
- Authentication occurs when an entity provides proof of an identity (such as a password). A second identity is the authenticator and it verifies the authentication.
- Authorization provides access to resources based on a proven identity.
- Accounting methods track user activity and record the activity in logs.
- Five factors of authentication are:
  - Something you know, such as a username and password
  - Something you have, such as a smart card, CAC, PIV, or token
  - Something you are, using biometrics, such as fingerprints or retina scans
  - Somewhere you are, using geolocation, a computer name, or a MAC address
  - Something you do, such as gestures on a touch screen
- The something you know factor typically refers to a shared secret, such as a password or a PIN. This is the least secure form of authentication.
- Passwords should be strong and changed often. Complex passwords include multiple character types. Strong passwords are complex and at least 14 characters long.
- Administrators should verify a user's identity before resetting the user's password. When resetting passwords manually, administrators should configure them as temporary passwords that expire after the first use, requiring users to create a new password the first time they

log on. Self-service password systems automate password recovery.

- Password policies provide a technical means to ensure users employ secure password practices.
- Password length specifies the minimum number of characters in the password.
- Password complexity ensures passwords are complex and include at least three of the four character types, such as special characters.
- Password history remembers past passwords and prevents users from reusing passwords.
- Minimum password age is used with password history to prevent users from changing their password repeatedly to get back to the original password.
- Maximum password age or password expiration forces users to change their password periodically. When administrators reset user passwords, the password should expire upon first use.
- Password policies should apply to any entity using a password. This includes user accounts and accounts used by services and applications. Applications with internally created passwords should still adhere to the organization's password policy.
- Account lockout policies lock out an account after a user enters an incorrect password too many times.
- Smart cards are credit card-sized cards that have embedded certificates used for authentication. They require a PKI to issue certificates.
- Common Access Cards (CACs) and Personal Identity Verification (PIV) cards can be used as photo IDs and as smart cards (both identification and authentication).
- Tokens (or key fobs) display numbers in an LCD. These numbers provide rolling, one-time use passwords and are synchronized with a server. USB tokens include an embedded chip and a USB connection. Generically, these are called hardware tokens.
- HOTP and TOTP are open source standards used to create one-time-use passwords. HOTP creates a one-time-use password that does not expire and TOTP creates a one- time password that expires after 30 seconds.
- Biometric methods are the most difficult to falsify. Physical methods include voice and facial recognition, fingerprints, retina

scans, iris scans, and palm scans. Biometric methods can also be used for identification.

- The false acceptance rate (FAR), or false match rate, identifies the percentage of times false acceptance occurs. The false rejection rate (FRR), or false nonmatch rate, identifies the percentage of times false rejections occur. The crossover error rate (CER) indicates the quality of the biometric system. Lower CERs are better.

- Single-factor authentication includes one or more authentication methods in the same factor, such as a PIN and a password. Dual-factor (or two-factor) authentication uses two factors of authentication, such as a USB token and a PIN. Multifactor authentication uses two or more factors. Multifactor authentication is stronger than any form of single- factor authentication.

- Authentication methods using two or more methods in the same factor are single- factor authentication. For example, a password and a PIN are both in the something you know factor, so they only provide single-factor authentication.

Comparing Authentication Services

- Kerberos is a network authentication protocol using tickets issued by a KDC or TGT server. If a ticket-granting ticket expires, the user might not be able to access resources. Microsoft Active Directory domains and Unix realms use Kerberos for authentication.
- LDAP specifies formats and methods to query directories. It provides a single point of management for objects, such as users and computers, in an Active Directory domain or Unix realm. The following is an example of an LDAP string: LDAP:// CN=Homer,CN=Users,DC=GetCertifiedGetAhead,DC=com
- LDAP Secure (LDAPS) encrypts transmissions with SSL or TLS.
- Single sign-on (SSO) allows users to authenticate with a single user account and access multiple resources on a network without authenticating again.
- SSO can be used to provide central authentication with a federated database and use this authentication in an environment with different operating systems (nonhomogeneous environment).
- SAML is an XML-based standard used to exchange authentication

and authorization information between different parties. SAML is used with web-based applications.

- A federated identity links a user's credentials from different networks or operating systems, but the federation treats it as one identity.
- Shibboleth is an open source federated identity solution that includes Open SAML libraries.
- OAuth and OpenID Connect are used by many web sites to streamline the authentication process for users. They allow users to log on to many web sites with another account, such as one they've created with Google, Facebook, PayPal, Microsoft, or Twitter.

Managing Accounts

- The principle of least privilege is a technical control that uses access controls. It specifies that individuals or processes are granted only the rights and permissions needed to perform assigned tasks or functions, but no more.
- Users should not share accounts. It prevents effective identification, authentication, authorization, and accounting. Most organizations ensure the Guest account is disabled.
- Account policies often require administrators to have two accounts (an administrator account and a standard user account) to prevent privilege escalation and other attacks.
- An account disablement policy ensures that inactive accounts are disabled. Accounts for employees who either resign or are terminated should be disabled as soon as possible. Configuring expiration dates on temporary accounts ensures they are disabled automatically.
- Time restrictions can prevent users from logging on or accessing network resources during specific hours. Location-based policies prevent users from logging on from certain locations.
- Accounts should be recertified to verify they are still required. For example, if the organization extends a contract, it's a simple matter to recertify the account. Administrators verify that the contract has been extended, change the expiration date,

and enable the account.

- Administrators routinely perform account maintenance. This is

often done with scripts to automate the processes and includes deleting accounts that are no longer needed.

- Credential management systems store and simplify the use of credentials for users. When users access web sites needing credentials, the system automatically retrieves the stored credentials and submits them to the web site.

Comparing Access Control Models

- The role-based access control (role-BAC) model uses roles to grant access by placing users into roles based on their assigned jobs, functions, or tasks. A matrix matching job titles with required privileges is useful as a planning document when using role-BAC.
- Group-based privileges are a form of role-BAC. Administrators create groups, add users to the groups, and then assign permissions to the groups. This simplifies administration because administrators do not have to assign permissions to users individually.
- The rule-based access control (rule-BAC) model is based on a set of approved instructions, such as ACL rules in a firewall. Some rule-BAC implementations use rules that trigger in response to an event, such as modifying ACLs after detecting an attack.
- In the discretionary access control (DAC) model, every object has an owner. The owner has explicit access and establishes access for any other user. Microsoft NTFS uses the DAC model, with every object having a discretionary access control list (DACL). The DACL identifies who has access and what access they are granted. A major flaw of the DAC model is its susceptibility to Trojan horses.
- Mandatory access control (MAC) uses security or sensitivity labels to identify objects (what you'll secure) and subjects (users). It is often used when access needs to be restricted based on a need to know. The administrator establishes access based on predefined security labels. These labels are often defined with a lattice to specify the upper and lower security boundaries.
- An attribute-based access control (ABAC) evaluates attributes and grants access based on the value of these attributes. It is used in many software defined networks (SDNs).

Online References

- Have you looked at the online content recently? You can view labs and additional sample questions at *http://gcgapremium.com/501-extras*.

Chapter 2 Practice Questions

1. Developers in your organization have created an application designed for the sales team. Salespeople can log on to the application using a simple password of 1234. However, this password does not meet the organization's password policy. Which of the following is the BEST response by the security administrator after learning about this?

   A. Nothing. Strong passwords aren't required in applications.
   B. Modify the security policy to accept this password.
   C. Document this as an exception in the application's documentation.
   D. Direct the application team manager to ensure the application adheres to the organization's password policy.

2. Ned is reviewing password security for employees of The Leftorium. The password policy has the following settings:

   - The password maximum age is 30 days.
   - The password minimum length is 14 characters.
   - Passwords cannot be reused until five other passwords have been used.
   - Passwords must include at least one of each of the following four character types: uppercase letters, lowercase letters, numbers, and special characters.

Ned discovers that despite having this password policy in place, users are still using the same password that they were using more than a month ago. Which of the following actions will resolve this issue?

   A. Create a rule in the password policy for the password minimum age to be 7 days.
   B. Change the password history to 10.
   C. Require the use of complex passwords.
   D. Change the maximum age setting to 60 days.

3. Your organization is planning to implement remote access capabilities. Management wants strong authentication and wants to ensure that passwords expire after a predefined time interval. Which of the following choices BEST meets this requirement?

   A. HOTP
   B. TOTP

C. CAC
D. Kerberos

4. Your organization has decided to implement a biometric solution for authentication. One of the goals is to ensure that the biometric system is highly accurate. Which of the following provides the BEST indication of accuracy with the biometric system?

A. The lowest possible FRR
B. The highest possible FAR
C. The lowest possible CER
D. The highest possible CER

5. Your organization recently updated an online application that employees use to log on when working from home. Employees enter their username and password into the application from their smartphone and the application logs their location using GPS. Which type of authentication is being used?

A. One-factor
B. Dual-factor
C. Something you are
D. Somewhere you are

6. A network includes a ticket-granting ticket server used for authentication. Which authentication service does this network use?

A. Shibboleth
B. SAML
C. LDAP
D. Kerberos

7. Lisa is a training instructor and she maintains a training lab with 18 computers. She has enough rights and permissions on these machines so that she can configure them as needed for classes. However, she does not have the rights to add them to the organization's domain. Which of the following choices BEST describes this example?

A. Least privilege
B. Need to know
C. Group-based privileges
D. Location-based policies

8. Marge is reviewing an organization's account management processes. She wants to ensure that security log entries accurately report the identity

of personnel taking specific actions. Which of the following steps would BEST meet this requirement?

A. Update ACLs for all files and folders.
B. Implement role-based privileges.
C. Use an SSO solution.
D. Remove all shared accounts.

9. A recent security audit discovered several apparently dormant user accounts. Although users could log on to the accounts, no one had logged on to them for more than 60 days. You later discovered that these accounts are for contractors who work approximately one week every quarter. Which of the following is the BEST response to this situation?

A. Remove the account expiration from the accounts.
B. Delete the accounts.
C. Reset the accounts.
D. Disable the accounts.

10. Members of a project team chose to meet at a local library to complete some work on a key project. All of them are authorized to work from home using a VPN connection and have connected from home successfully. However, they found that they were unable to connect to the network using the VPN from the library and they could not access any of the project data. Which of the following choices is the MOST likely reason why they can't access this data?

A. Role-based access control
B. Time-of-day access control
C. Location-based policy
D. Discretionary access control

11. You need to create an account for a contractor who will be working at your company for 60 days. Which of the following is the BEST security step to take when creating this account?

A. Configure history on the account.
B. Configure a password expiration date on the account.
C. Configure an expiration date on the account.
D. Configure complexity.

12. A company recently hired you as a security administrator. You notice that some former accounts used by temporary employees are currently enabled. Which of the following choices is the BEST response?

A. Disable all the temporary accounts.
B. Disable the temporary accounts you've noticed are enabled.
C. Craft a script to identify inactive accounts based on the last time they logged on.
D. Set account expiration dates for all accounts when creating them.

13. Developers are planning to develop an application using role-based access control. Which of the following would they MOST likely include in their planning?
A. A listing of labels reflecting classification levels
B. A requirements list identifying need to know
C. A listing of owners
D. A matrix of functions matched with their required privileges

14.

Document Type	Security Level	Security Label
Employment documents	Private	Employee
Salary and compensation documents	Private	Payroll

A security administrator needs to implement an access control system that will protect data based on the following matrix.

(Note that this matrix only represents a subset of the overall requirements.) Which of the following models is the administrator implementing?
A. DAC
B. MAC
C. Role-BAC
D. ABAC

15. Your organization is implementing an SDN. Management wants to use an access control model that controls access based on attributes. Which of the following is the BEST solution?
A. DAC
B. MAC
C. Role-BAC
D. ABAC

Chapter 2 Practice Question Answers

1. **D.** The application should be recoded to adhere to the company's password policy, so the best response is to direct the application team manager to do so. Application passwords should be strong and should adhere to an organization's security policy. It is not appropriate to weaken a security policy to match a weakness in an application. Nor is it appropriate to simply document that the application uses a weak password.

2. **A.** The best solution is to create a rule in the password policy for the password minimum age. Currently, users can change their passwords five more times in just a couple of minutes, changing it back to their original password on the sixth change. None of the other settings prevent the users from doing this. A password history of 10 forces the users to take a couple more minutes to get back to the original password. The password policy currently requires complex passwords. A maximum age of 60 days increases how long a user can keep the same password.

3. **B.** A Time-based One-Time Password (TOTP) meets this requirement. Passwords created with TOTP expire after 30 seconds. An HMAC-based One-Time Password (HOTP) creates passwords that do not expire. A Common Access Card (CAC) is a type of smart card, but it does not create passwords. Kerberos uses tickets instead of passwords.

4. **C.** A lower crossover error rate (CER) indicates a more accurate biometric system. The false acceptance rate (FAR) and the false rejection rate (FRR) vary based on the sensitivity of the biometric system and don't indicate accuracy by themselves. A higher CER indicates a less accurate biometric system.

5. **A.** This is using one-factor authentication—something you know. The application uses the username for identification and the password for authentication. Note that even though the application is logging the

location using Global Positioning System (GPS), there isn't any indication that it is using this information for authentication. Dual-factor authentication requires another factor of authentication. If the application verified you were logging on from a specific GPS location as part of the authentication, it would be dual-factor authentication (something you know and somewhere you are). Something you are refers to biometric authentication methods. The somewhere you are authentication method verifies you are somewhere, such as in a specific GPS location, but this isn't being used for authentication in this scenario.

6. **D.** Kerberos uses a ticket-granting ticket (TGT) server, which creates tickets for authentication. Shibboleth is a federated identity solution used in some single sign-on (SSO) solutions. Security Assertion Markup Language (SAML) is an Extensible Markup Language (XML) used for some SSO solutions. Lightweight Directory Access Protocol (LDAP) is an X.500- based authentication service used to identify objects.

7. **A.** When following the principle of least privilege, individuals have only enough rights and permissions to perform their job, and this is exactly what is described in this scenario. Need to know typically refers to data and information rather than the privileges required to perform an action, such as adding computers to a domain. Group-based privileges refer to giving permissions to groups, and then adding the users to the groups to give them appropriate privileges. A location-based policy allows or blocks access based on location, but the scenario doesn't indicate the location is being checked.

8. **D.** Removing all shared accounts is the best answer of the available choices. If two employees are using the same account, and one employee maliciously deletes data in a database, it isn't possible to identify which employee deleted the data. File and folder access control lists (ACLs) identify permissions for users, but don't control the user identity. Role-based (or group-based) privileges assign the same permissions to all members of a group, which simplifies administration. A single sign-on (SSO) solution allows a user to log on once and access multiple resources.

9. **D.** The best response is to disable the accounts and then enable them when needed by the contractors. Ideally, the accounts would include an

expiration date so that they would automatically expire when no longer needed, but the scenario doesn't indicate the accounts have an expiration date. Because the contractors need to access the accounts periodically, it's better to disable them rather than delete them. Reset the accounts implies you are changing the password, but this isn't needed.

10. **C.** A location-based policy restricts access based on location, such as with an IP address, and this is the best possible answer of those given. The scenario indicates they could use the virtual private network (VPN) connection from home, but it was blocked when they tried to access it from the library. A time-of-day access control restricts access based on the time of day, but the scenario doesn't indicate the time. Neither a discretionary access control model nor a role-based access control model restricts access based on location.

11. **C.** When creating temporary accounts, it's best to configure expiration dates so that the system will automatically disable the accounts on the specified date. History, password expiration, and complexity all refer to password policy settings. However, it's rare to configure a specific password policy on a single account.

12. **C.** Running a last logon script allows you to identify inactive accounts, such as accounts that haven't been logged on to in the last 30 days. It's appropriate to disable unused accounts, but it isn't necessarily appropriate to disable all temporary accounts, because some might still be in use. If you disable the accounts you notice, you might disable accounts that some employees are still using, and you might miss some accounts that should be disabled. Setting expiration dates for newly created accounts is a good step, but it doesn't address previously created accounts.

13. **D.** A matrix of functions, roles, or job titles matched with the required access privileges for each of the functions, roles, or job titles is a common planning document for a role- based access control (role-BAC) model. The mandatory access control (MAC) model uses sensitivity labels and classification levels. MAC is effective at restricting access based on a need to know. The discretionary access control (DAC) model specifies that every object has an owner and it might identify owners in a list.

14. **B.** This is a mandatory access control (MAC) model. You can tell because it is using security labels. None of the other models listed use labels. A discretionary access control (DAC) model has an owner, and the owner establishes access for the objects. A role-based access control (role-BAC) model uses roles or groups to assign rights and permissions. An attribute-based access control (ABAC) model uses attributes assigned to subjects and objects within a policy to grant access.

15. **D.** A software defined network (SDN) typically uses an attribute-based access control (ABAC) model, which is based on attributes that identify subjects and objects within a policy. A discretionary access control (DAC) model has an owner, and the owner establishes access for the objects. A mandatory access control (MAC) model uses labels assigned to subjects and objects. A role-based access control (role-BAC) model uses roles or groups to assign rights and permissions.