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

Reader/writer

Data in/out

Data
transmission
controller

To host systems

External
terminal

Clock

System
interface

External
terminal

Integrated
chip

System board

Contactless type
Smart card

Data

transmission
controller

Data transmitter

To host systems

Electric power,
clock,
data in/out

System board
System
interface

Electromagnetic
waves

Antenna coil

Antenna coil

Integrated
chip

Reader/writer

PART V

Electric power

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

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

overhead
of card generation, as with memory cards, although this cost is decreasing. The smart
cards themselves are more expensive than memory cards because of the extra integrated
circuits and microprocessor. Essentially, a smart card is a kind of computer, and because
of that it has many of the operational challenges and risks that can affect a computer.

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

Smart Card Attacks Smart cards are more tamperproof than memory cards, but
where there is sensitive data, there are individuals who are motivated to circumvent any
countermeasure the industry throws at them. Over the years, criminals have become
very inventive in the development of various ways to attack smart cards. Smart card
attacks tend to be special cases of the cryptanalysis techniques we discussed in Chapter
8. For example, attackers have introduced computational errors into smart cards with
the goal of uncovering the encryption keys used and stored on the cards. These "errors"
are introduced by manipulating some environmental component of the card (changing
input voltage, clock rate, temperature fluctuations). The attacker reviews the result of an
encryption function after introducing an error to the card, and also reviews the correct
result, which the card performs when no errors are introduced. Analysis of these different
results may allow an attacker to reverse-engineer the encryption process, with the hope
of uncovering the encryption key. This type of attack is referred to as fault generation.

Side-channel attacks are nonintrusive and are used to uncover sensitive information
about how a component works, without trying to compromise any type of flaw or
weakness. So a noninvasive attack is one in which the attacker watches how something
works and how it reacts in different situations instead of trying to "invade" it

with more

Interoperability
In the industry today, lack of interoperability is a big problem. An ISO/IEC standard, 14443, outlines the following items for smart card standardization:

- ISO/IEC 14443-1
- ISO/IEC 14443-2
- ISO/IEC 14443-3
- ISO/IEC 14443-4

Physical characteristics
Radio frequency power and signal interface
Initialization and anticollision
Transmission protocol

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

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

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

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

Credential Management

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

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

will be
looking for—and we always want to make the auditors happy!
NOTE These types of credential management products are commonly used
to set up and maintain internal accounts. Web access control management is
used mainly for external users.

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

Password Managers

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

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

Password Synchronization

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

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

Self-Service Password Reset

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

PART V

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

Products are available that allow users to change their passwords through other means. For example, if you forgot your password, you may be asked to answer some of the questions answered during the registration process of your account (i.e., a cognitive password). If you do this correctly, an e-mail is sent to you with a link you must click.

The password management product has your identity tied to the answers you gave

to the
questions during your account registration process and to your e-mail address. If you do
everything correctly, you are given a screen that allows you to reset your password.

Assisted Password Reset

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

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

Just-in-Time Access

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

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

Registration and Proofing of Identity

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

Authoritative System of Record

The authoritative source is the “system of record,” or the location where identity information originates and is maintained. It should have the most up-to-date and reliable identity information. An authoritative system of record (ASOR) is a hierarchical tree-like structure system that tracks subjects and their authorization chains.

Organizations need an automated and reliable way of detecting and managing unusual or suspicious changes to user accounts and a method of collecting this type of data through extensive auditing capabilities. The ASOR should contain the subject’s name, associated accounts, authorization history per account, and provision details. This type of workflow and accounting is becoming more in demand for regulatory compliance because it allows auditors to understand how access is being centrally controlled within an environment.

of privileges and permissions. But how would the network administrator know what resources this new user should have access to and what permissions should be assigned to the new account? In most situations, she doesn't—she just wings it. This is how users end up with too much access to too many resources. What should take place instead is implementation of a workflow process that allows for a request for a new user account.

Since hardly anyone in the organization likely knows the new employee, we need someone to vouch for this person's identity. This process, sometimes called proofing of identity, is almost always carried out by human resources (HR) personnel who would've had to verify the new employee's identity for tax and benefit purposes. The new account request is then sent to the employee's manager, who verifies the permissions that this person needs, and a ticket is generated for the technical staff to set up the account(s).

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

Over time, this new user will commonly have different identity attributes, which will be used for authentication purposes, stored in different systems in the network.

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

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

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

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

Profile Update

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

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

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

Session Management
A session is an agreement between two parties to communicate interactively. Think of it as a phone call: you dial your friend's number, she decides whether to answer, and if she does then you talk with each other until something happens to end the call. That "something" could be that you (or her) are out of time and have to go, or maybe one of you runs out of things to say and there's an awkward silence on the line, or maybe one of you starts acting weird and the other is bothered and hangs up. Technically, the call could go on forever, though in practice that doesn't happen.
Information systems use sessions all the time. When you show up for work and log onto your computer, you establish an authenticated session with the operating system that allows you to launch your e-mail client. When that application connects to the mail server, it establishes a different authenticated session (perhaps using the same credentials you used to log onto your computer). So, a session, in the context of information systems security, can exist between a user and an information system or between two

information systems (e.g., two running programs). If the session is an authenticated one, as in the previous two examples, then authentication happens at the beginning and then everything else is trusted until the session ends.
That trust is the reason we need to be very careful about how we deal with our sessions. Threat actors often try to inject themselves into an authenticated session and hijack it for their own purposes. Session management is the process of establishing, controlling, and terminating sessions, usually for security reasons. The session establishment usually entails authentication and authorization of one or both endpoints. Controlling the session can involve logging the start and end and anything in between. It could also keep track of time, activity, and even indicia of malicious activity. These are three of the most common triggers for session termination:

Accountability
Auditing capabilities ensure users are accountable for their actions, verify that the security policies are enforced, and can be used as investigation tools. There are several reasons

why network administrators and security professionals want to make sure accountability mechanisms are in place and configured properly: to deter wrongdoing, be able to track bad deeds back to individuals, detect intrusions, reconstruct events and system conditions, provide legal recourse material, and produce problem reports. Audit documentation and log files hold a mountain of information—the trick is usually deciphering it and presenting it in a useful and understandable format.

Accountability is enabled by recording user, system, and application activities. This recording is done through auditing functions and mechanisms within an operating system or application. Audit trails contain information about operating system activities, application events, and user actions. Audit trails can be used to verify the health of a system by checking performance information or certain types of errors and conditions.

After a system crashes, a network administrator often will review audit logs to try and piece together the status of the system and attempt to understand what events could be attributed to the disruption.

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

Audit trails can also be used to provide alerts about any suspicious activities that can be investigated at a later time. In addition, they can be valuable in determining exactly how far an attack has gone and the extent of the damage that may have been caused. It

is important to make sure a proper chain of custody is maintained to ensure any data
collected can later be properly and accurately represented in case it needs to be used for
later events such as criminal proceedings or investigations.
Keep the following in mind when dealing with auditing:

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

An administrator configures what actions and events are to be audited and logged.
In a high-security environment, the administrator would configure more activities to
be captured and set the threshold of those activities to be more sensitive. The events can
be reviewed to identify where breaches of security occurred and if the security policy
has been violated. If the environment does not require such levels of security, the events
analyzed would be fewer, with less-demanding thresholds.
Without proper oversight, items and actions to be audited can become an endless
list. A security professional should be able to assess an environment and its security
goals, know what actions should be audited, and know what is to be done with that
information after it is captured—without wasting too much disk space, CPU power, and
staff time. The following is a broad overview of the items and actions that can be audited
and logged.
System-level events:

• System performance
• Logon attempts (successful and unsuccessful)
• Logon ID
• Date and time of each logon attempt
• Lockouts of users and terminals
• Use of administration utilities
• Devices used
• Functions performed
• Requests to alter configuration files

Application-level events:

• Error messages
• Files opened and closed

• Modifications of files
• Security violations within applications

User-level events:

• Identification and authentication attempts
• Files, services, and resources used
• Commands initiated
• Security violations

Review of Audit Information

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

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

Intrusion detection systems (IDSs) continually scan audit logs for suspicious activity. If an intrusion or harmful event takes place, audit logs are usually kept to be used later to prove guilt and prosecute if necessary. If severe security events take place, the IDS alerts the administrator or staff member so they can take proper actions to end the destructive activity. If a dangerous virus is identified, administrators may take the mail

server offline.
If an attacker is accessing confidential information within the database, this computer may be temporarily disconnected from the network or Internet. If an attack is in progress, the administrator may want to watch the actions taking place so she can track down the intruder. IDSs can watch for this type of activity during real time and/or scan audit logs and watch for specific patterns or behaviors.

Reviewing audit information manually can be overwhelming. Fortunately, there are applications and audit trail analysis tools that reduce the volume of audit logs to review and improve the efficiency of manual review procedures. A majority of the time, audit logs contain information that is unnecessary, so these tools parse out specific events and present them in a useful format.
An audit-reduction tool does just what its name suggests—reduces the amount of information within an audit log. This tool discards mundane task information and records system performance, security, and user functionality information that can be useful to a security professional or administrator.
Today, more organizations are implementing security information and event management (SIEM) systems. These products gather logs from various devices (servers, firewalls, routers, etc.) and attempt to correlate the log data and provide analysis capabilities.
Reviewing logs manually looking for suspicious activity in a continuous manner is not only mind-numbing; it is close to impossible to be successful. So many packets and network communication data sets are passing along a network, humans cannot collect all the data in real or near real time, analyze it, identify current attacks, and react—it is just too overwhelming.
Organizations also have different types of systems on a network (routers, firewalls, IDS, IPS, servers, gateways, proxies) collecting logs in various proprietary formats, which requires centralization, standardization, and normalization. Log formats are different per product type and vendor. The format of logs created by Juniper network device systems is different from the format of logs created by Cisco systems, which in turn is different from the format created by Palo Alto and Barracuda firewalls. It is important

to gather logs from various different systems within an environment so that some type of situational awareness can take place. Once the logs are gathered, intelligence routines need to be processed on them so that data mining can take place to identify patterns. The goal is to piece together seemingly unrelated event data so that the security team can fully understand what is taking place within the network and react properly.

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

Protecting Audit Data and Log Information

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

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

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

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

Identity Management

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

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

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

Sales
employees

Data center

Executives
ERP
Customers
Human
resources
system

Temporary
employees

Network

Contractors

Distribution
partners

CRM

Former
employees
IT
employees

Figure 16-4

Cloud
services

Most environments are complex in terms of access.

view of most environments.) Users usually access several different types of systems throughout their daily tasks, which makes controlling access and providing the necessary level of protection on different data types difficult and full of obstacles. This complexity usually results in unforeseen and unidentified holes in asset protection, overlapping and contradictory controls, and policy and regulation noncompliance. It is the goal of IdM technologies to simplify the administration of these tasks and bring order to chaos.
The following are some of the common questions enterprises deal with regarding IdM implementation:

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

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

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

Directory Services

Directory services, much like DNS, map resource names to their corresponding network addresses, allowing discovery of and communication with devices, files, users, or any other asset. Network directory services provide users access to network resources transparently, meaning that users don't need to know the exact location of the resources or the steps required to access them. The network directory services handle these issues for the user in the background.

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

The objects within the directory are managed by a directory service. The directory

service allows an administrator to configure and manage how identification, authentication,
authorization, and access control take place within the network and on individual systems.
The objects within the directory are labeled and identified with namespaces.
In a Windows Active Directory (AD) environment, when you log in, you are logging into a domain controller (DC), which has a hierarchical LDAP directory in its database.
The database organizes the network resources and carries out user access control functionality. So once you successfully authenticate to the DC, certain network resources
are available to you (print service, file server, e-mail server, and so on) as dictated by the
configuration of AD.
How does the directory service keep all of these entities organized? By using namespaces.
Each directory service has a way of identifying and naming the objects they manage. In
LDAP, the directory service assigns distinguished names (DNs) to each object. Each DN

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

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

dc = .LogicalSecurity

cn = .Shon Harris
```

This is a very simplistic example. Companies usually have large trees (directories)
containing many levels and objects to represent different departments, roles, users, and
resources.
A directory service manages the entries and data in the directory and also enforces
the configured security policy by carrying out access control and identity management
functions. For example, when you log into the DC, the directory service determines
which resources you can and cannot access on the network.

Directories' Role in Identity Management
A directory service is a general-purpose resource that can be used for IdM. When used

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

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

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

Organizing All of This Stuff

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

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

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

Directory
Schema

C=US
ou=U.S. Government
ou=DoD

Service/agency subtrees

ou=Army

ou=locations

ou=mail lists

ou=pentagon

ou=WNY

ou=Navy

ou=ships

ou=reston

ou=DISA

ou=tactical

PART V

ou=Air Force

Special subtrees

ou=PLAs

ou=organizations

OUs

OUs
Loc

ou=NCTSW

ou=General

OUs
OUs
OUs

cn=Kathy Conlon

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

LDAPenabled
application

Non-LDAP
applications

Central
LDAP
directory server

Non-LDAP
directory
server

Meta-directory

App-specific
LDAP
directories

Access management
Access management
Access management

Figure 16-5

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

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

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

these other
sources and updates the LDAP directory.

Single Sign-On

Employees typically need to access many different computers, servers, databases, and other resources in the course of a day to complete their tasks. This often requires the employees to remember multiple user IDs and passwords for these different computers.
In a utopia, a user would need to enter only one user ID and one password to be able to access all resources in all the networks this user is working in. In the real world, this is hard to accomplish for all system types.
Because of the proliferation of client/server technologies, networks have migrated from centrally controlled networks to heterogeneous, distributed environments. The propagation of open systems and the increased diversity of applications, platforms, and operating systems have caused the end user to have to remember several user IDs and passwords just to be able to access and use the different resources within his own network.
Although the different IDs and passwords are supposed to provide a greater level of

security, they often end up compromising security (because users write them down) and causing more effort and overhead for the staff that manages and maintains the network.
As any network staff member or administrator can attest to, too much time is devoted to resetting passwords for users who have forgotten them. More than one employee's productivity is affected when forgotten passwords have to be reassigned. The network staff member who has to reset the password could be working on other tasks, and the user who forgot the password cannot complete his task until the network staff member is finished resetting the password. Depending on the enterprise, between 20 percent and 50 percent of all IT help-desk calls are for password resets, according to the Gartner Group. Forrester Research estimates that each of these calls costs $70 in the United States.
System administrators have to manage multiple user accounts on different platforms, which all need to be coordinated in a manner that maintains the integrity of the

security policy. At times the complexity can be overwhelming, which results in poor access control management and the generation of many security vulnerabilities. A lot of time is spent on multiple passwords, and in the end they do not provide us with more security.

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

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

So that is our utopia: log on once and you are good to go. What bursts this bubble? Mainly interoperability issues. For SSO to actually work, every platform, application, and resource needs to accept the same type of credentials, in the same format, and interpret their meanings the same. When Steve logs on to his Windows workstation and gets authenticated by a mixed-mode Windows domain controller, it must authenticate him to the resources he needs to access on the Apple MacBook, the Linux server running NIS, the PrinterLogic print server, and the Windows computer in a trusted domain that has the plotter connected to it. A nice idea, until reality hits.

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

Federated Identity Management

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

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

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

John is authenticated
to Company B

John is authenticated
to Company A

John is authenticated
to Company C

John is authenticated
to Company D
Assertions

Web portal functions are parts of a website that act as a point of access to information.
A portal presents information from diverse sources in a unified manner. It can offer various services, as in e-mail, news updates, stock prices, data access, price lookups, access to databases, and entertainment. Web portals provide a way for organizations to present one consistent interface with one "look and feel" and various functionality types. For example, you log into your company web portal and it provides access to many different systems and their functionalities, but it seems as though you are only interacting with one system because the interface is "clean" and organized. Portals combine web services (web-based functions) from several different entities and present them in one central website.
A web portal is made up of portlets, which are pluggable user-interface software components that present information from other systems. A portlet is an interactive application that provides a specific type of web service functionality (e-mail, news feed, weather updates, forums, etc.). A portal is made up of individual portlets to provide a plethora of services through one interface. It is a way of centrally providing a set of web services. Users can configure their view to the portal by enabling or disabling these various portlet functions.
Since each of these portlets can be provided by different entities, how user authentication information is handled must be tightly controlled, and there must be a

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

Federated Identity with a Third-Party Service

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

There are three basic approaches to architecting identity management services: on-premise, cloud-based, and a hybrid of both. The first approach, on-premise, is simple because all the systems and data are located within the enterprise. In the cloud-based model, on the other hand, most or all of the systems or data are hosted by an external party in the cloud. A hybrid FIM system includes both on-premise and cloud-based IdM components, each responsible for its environment but able to coordinate with each other.

Regardless of the approach, it is important to ensure that all components play nice with each other. In the following sections we will explore some of the considerations that are common to the successful integration of these services.

Integration Issues

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

Establishing Connectivity

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

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

Establishing Trust

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

Incremental Testing

When dealing with complex systems, it is wise to assume that some important issue will not be covered in the plan. This is why it is important to incrementally test the integration of identity services instead of rolling out the entire system at once. Many organizations choose to roll out new services first to test accounts (i.e., not real users), then to one department or division that is used as the test case, and finally to the entire organization.

For critical deployments (and one would assume that identity services would fall in this category), it is best to test as thoroughly as possible in a testbed or sandbox environment.

Only then should the integration progress to real systems.

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

On-Premise

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

Legacy Systems

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

Cloud

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

Hybrid

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

Chapter Review

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

The purpose of this chapter was to expose you to the multiple processes and technologies that make identity management possible, both at an individual level and at aggregate enterprise scales. This all sets the stage for the next chapter, in

which we will
delve into how to operationalize these concepts and build on them to ensure authorized
parties (and no others) have access to the right assets (and no others).

Quick Review

- Identification describes a method by which a subject (user, program, or process) claims to have a specific identity (e.g., username, account number, or e-mail address).
- Authentication is the process by which a system verifies the identity of the subject, usually by requiring a piece of information that only the claimed identity should have.
- Credentials consist of an identification claim (e.g., username) and authentication information (e.g., password).
- Authorization is the determination of whether a subject has been given the necessary rights and privileges to carry out the requested actions.
- The three main types of factors used for authentication are something a person knows (e.g., password), something a person has (e.g., token), and something a person is (e.g., fingerprint), which can be combined with two additional factors: somewhere a person is (e.g., geolocation) and something a person does (e.g., keystroke behavior).
- Knowledge-based authentication uses information a person knows, such as a password, passphrase, or life experience.
- Salts are random values added to plaintext passwords prior to hashing to add more complexity and randomness.
- Cognitive passwords are fact- or opinion-based questions, typically based on life experiences, used to verify an individual's identity.
- A Type I biometric authentication error occurs when a legitimate individual is denied access; a Type II error occurs when an impostor is granted access.
- The crossover error rate (CER) of a biometric authentication system represents the point at which the false rejection rate (Type I errors) is equal to the false acceptance rate (Type II errors).
- Ownership-based authentication is based on something a person owns, such as a token device.
- A token device, or password generator, is usually a handheld device that has a display (and possibly a keypad), is synchronized in some manner with the authentication server, and displays to the user a one-time password (OTP).
- A synchronous token device requires the device and the authentication service to advance to the next OTP in sync with each other; an asynchronous token device employs a challenge/response scheme to authenticate the user.
- A memory card holds information but cannot process information; a smart

card holds information and has the necessary hardware and software to actually process that information.

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

Questions

Please remember that these questions are formatted and asked in a certain way

for a
reason. Keep in mind that the CISSP exam is asking questions at a conceptual level.
Questions may not always have the perfect answer, and the candidate is advised against
always looking for the perfect answer. Instead, the candidate should look for the best
answer in the list.

1. Which of the following statements correctly describes biometric methods of authentication?
A. They are the least expensive and provide the most protection.
B. They are the most expensive and provide the least protection.
C. They are the least expensive and provide the least protection.
D. They are the most expensive and provide the most protection.

2. Which of the following statements correctly describes the use of passwords for
authentication?
A. They are the least expensive and most secure.
B. They are the most expensive and least secure.
C. They are the least expensive and least secure.
D. They are the most expensive and most secure.

3. How is a challenge/response protocol utilized with token device implementations?
A. This type of protocol is not used; cryptography is used.

a response based on the challenge.
C. The token challenges the user for a username and password.
D. The token challenges the user's password against a database of stored credentials.

4. The process of mutual authentication involves _______________.
A. a user authenticating to a system and the system authenticating to the user
B. a user authenticating to two systems at the same time
C. a user authenticating to a server and then to a process
D. a user authenticating, receiving a ticket, and then authenticating to a service

5. What role does biometrics play in access control?
A. Authorization
B. Authenticity
C. Authentication
D. Accountability

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

6. Which of the following is the best description of directories that are used in
identity management technology?

A. Most are hierarchical and follow the X.500 standard.
B. Most have a flat architecture and follow the X.400 standard.
C. Most have moved away from LDAP.
D. Most use RADIUS.

7. Which of the following is not part of user provisioning?
A. Creation and deactivation of user accounts
B. Business process implementation
C. Maintenance and deactivation of user objects and attributes
D. Delegating user administration

8. What is a technology that allows a user to remember just one password?
A. Password generation
B. Password dictionaries
C. Password rainbow tables
D. Password synchronization

9. This graphic covers which of the following?

Biometric
capture

Template
extraction

Image
processing

10110011
01011000
11001011
01101101
01011000

10110011
01011000
11001011
01101101
01011000

Biometric
matching

95%

A. Crossover error rate
B. Identity verification
C. Authorization rates
D. Authentication error rates

10. The diagram shown here explains which of the following concepts?

Error rate

FRR

FAR
Biometric
characteristic
features rejected

Biometric
characteristic
features accepted

Equal error rate (EER)

EER
False
acceptance

False
rejection

Decision threshold

A. Crossover error rate.
B. Type III errors.
C. FAR equals FRR in systems that have a high crossover error rate.
D. Biometrics is a high acceptance technology.

Firewall
Internet

Server
284836

284836

Algorithm

Algorithm

Time

Seed

Time

Seed

11. The graphic shown here illustrates how which of the following works?

762
A. Rainbow tables
B. Dictionary attack
C. One-time password
D. Strong authentication

Answers
1. D. Compared with the other available authentication mechanisms, biometric methods provide the highest level of protection and are the most expensive.
2. C. Passwords provide the least amount of protection, but are the cheapest because
they do not require extra readers (as with smart cards and memory cards), do not require devices (as do biometrics), and do not require a lot of overhead in processing
(as in cryptography). Passwords are the most common type of authentication method used today.
3. B. An asynchronous token device is based on challenge/response mechanisms. The authentication service sends the user a challenge value, which the user enters
into the token. The token encrypts or hashes this value, and the user uses this as
her one-time password.
4. A. Mutual authentication means it is happening in both directions. Instead of just the user having to authenticate to the server, the server also must authenticate
to the user.
5. C. Biometrics is a technology that validates an individual's identity by reading a
physical attribute. In some cases, biometrics can be used for identification, but
that was not listed as an answer choice.
6. A. Most organizations have some type of directory service that contains information pertaining to the organization's network resources and users. Most directories follow a hierarchical database format, based on the X.500 standard, and a type of protocol, as in Lightweight Directory Access Protocol (LDAP), that allows subjects and applications to interact with the directory. Applications can
request information about a particular user by making an LDAP request to the directory, and users can request information about a specific resource by using a
similar request.
7. B. User provisioning refers to the creation, maintenance, and deactivation of user objects and attributes as they exist in one or more systems, directories, or
applications, in response to business processes. User provisioning software may include one or more of the following components: change propagation, self-service
workflow, consolidated user administration, delegated user administration, and federated change control. User objects may represent employees, contractors, vendors, partners, customers, or other recipients of a service. Services may include
e-mail, access to a database, access to a file server or mainframe, and so on.

8. D. Password synchronization technologies can allow a user to maintain just one
password across multiple systems. The product synchronizes the password to other systems and applications, which happens transparently to the user.
9. B. These steps are taken to convert the biometric input for identity verification:
i. A software application identifies specific points of data as match points.
ii. An algorithm is used to process the match points and translate that information

into a numeric value.
iii. Authentication is approved or denied when the database value is compared with the end user input entered into the scanner.
10. A. This rating is stated as a percentage and represents the point at which the false
rejection rate equals the false acceptance rate. This rating is the most important
measurement when determining a biometric system's accuracy.
• Type I error, false rejection rate (FRR) Rejects authorized individual
• Type II error, false acceptance rate (FAR) Accepts impostor
11. C. Different types of one-time passwords are used for authentication. This graphic
illustrates a synchronous token device, which synchronizes with the authentication
service by using time or a counter as the core piece of the authentication process.

This page intentionally left blank

17

CHAPTER

Managing Identities and Access

This chapter presents the following:
• Authorization mechanisms
• Implementing authentication systems
• Managing the identity and access provisioning life cycle
• Controlling physical and logical access

Locks keep out only the honest.
—Proverb

Identification and authentication of users and systems, which was the focus of the previous chapter, is only half of the access control battle. You may be able to establish that
you are truly dealing with Ahmed, but what assets should he be allowed to access? It
really depends on the sensitivity of the asset, Ahmed's role, and any applicable

rules on
how these assets are supposed be used. Access control can also depend on any number of
other attributes of the user, the asset, and the relationship between the two. Finally, access
control can be based on risk.
Once you decide what access control model is best for your organization, you still
have to implement the right authentication and authorization mechanism. There are
many choices, but in this chapter we'll focus on the technologies that you are likeliest to
encounter in the real world (and on the CISSP exam). We'll talk about how to manage
the user access life cycle, which is where a lot of organizations get in trouble by not
changing authorizations as situations change. After we cover all these essentials, we'll see
how it all fits together in the context of controlling access to physical and logical assets.
Let's start by looking at authorization mechanisms.

Authorization Mechanisms

Authorization is the process of ensuring authenticated users have access to the resources
they are authorized to use and don't have access to any other resources. This is preceded
by authentication, of course, but unlike that process, which tends to be a one-time activity,

authorization controls every interaction of every user with every resource. It is an
ongoing, all-seeing, access control mechanism.
An access control mechanism dictates how subjects access objects. It uses access control
technologies and security mechanisms to enforce the rules and objectives of an access
control model. As discussed in this section, there are six main types of access control
models: discretionary, mandatory, role-based, rule-based, attribute-based, and risk-based.
Each model type uses different methods to control how subjects access objects, and each
has its own merits and limitations. The business and security goals of an organization,
along with its culture and habits of conducting business, help prescribe what access
control model it should use. Some organizations use one model exclusively, whereas

others combine models to provide the necessary level of protection.
Regardless of which model or combination of models your organization uses, your security team needs a mechanism that consistently enforces the model and its rules.
The reference monitor is an abstract machine that mediates all access subjects have to objects, both to ensure that the subjects have the necessary access rights and to protect the objects from unauthorized access and destructive modification. It is an access control concept, not an actual physical component, which is why it is normally referred to as the "reference monitor concept" or an "abstract machine." However the reference monitor is implemented, it must possess the following three properties to be effective:

- Always invoked To access an object, you have to go through the monitor first.
- Tamper-resistant It must ensure a threat actor cannot disable or modify it.
- Verifiable It must be capable of being thoroughly analyzed and tested to ensure that it works correctly all the time.

Let's explore the different approaches to implement and manage authorization mechanisms. The following sections explain the six different models and where they should be implemented.

Discretionary Access Control

If a user creates a file, he is the owner of that file. An identifier for this user is placed in the file header and/or in an access control matrix within the operating system. Ownership might also be granted to a specific individual. For example, a manager for a certain department might be made the owner of the files and resources within her department. A system that uses discretionary access control (DAC) enables the owner of the resource to specify which subjects can access specific resources. This model is called discretionary because the control of access is based on the discretion of the owner. Many times department managers or business unit managers are the owners of the data within their specific department.
Being the owner, they can specify who should have access and who should not.
In a DAC model, access is restricted based on the authorization granted to the users.
This means users are allowed to specify what type of access can occur to the objects they own. If an organization is using a DAC model, the network administrator can allow resource owners to control who has access to their files. The most common

Identity-Based Access Control

DAC systems grant or deny access based on the identity of the subject. The

identity
can be a user identity or a group membership. So, for example, a data owner can choose to allow Bob (user identity) and the Accounting group (group membership identity) to access his file. If Bob as a user is only granted Read access but he
happens to be a member of the Accounting group, which has Change access, Bob would get the greater of the two: Change. The exception to this “greater access” rule is when No Access is set. In that case, it doesn’t matter what other access levels
a user may have gotten as an individual or through group membership, since that rule trumps all others.

Access Control Lists

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

ACLs map values from the access control matrix to the object. Whereas a capability
corresponds to a row in the access control matrix, the ACL corresponds to a column of
the matrix. The ACL for a notional File1 object is shown in Table 17-1.

Table 17-1
The ACL for a
Notional File1
Object

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

PART V

implementation of DAC is through access control lists (ACLs), which are dictated and set by the owners and enforced by the operating system.

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

DAC can be applied to both the directory tree structure and the files it contains.

The Microsoft Windows world has access permissions of No Access, Read (r), Write (w), Execute (x), Delete (d), Change (c), and Full Control. The Read attribute lets you read the file but not make changes. The Change attribute allows you to read, write, execute, and delete the file but does not let you change the ACLs or the owner of the files. Obviously, the attribute of Full Control lets you make any changes to the file and its permissions and ownership.

Challenges When Using DAC

While DAC systems provide a lot of flexibility to the user and less administration for IT, it is also the Achilles' heel of operating systems. Malware can install itself and work under the security context of the user. For example, if a user opens an attachment that is infected with a virus, the code can install itself in the background without the user's being aware of this activity. This code basically inherits all the rights and permissions that the user has and can carry out all the activities the user can on the system. It can send copies of itself out to all the contacts listed in the user's e-mail client, install a back door, attack other systems, delete files on the hard drive, and more. The user is actually giving rights to the virus to carry out its dirty deeds, because the user has discretionary rights and is considered the owner of many objects on the system. This is particularly problematic in environments where users are assigned local administrator or root accounts, because once malware is installed, it can do anything on a system.

While we may want to give users some freedom to indicate who can access the files that they create and other resources on their systems that they are configured to be "owners" of, we really don't want them dictating all access decisions in environments with assets that need to be protected. We just don’t trust them that much, and we shouldn’t if you think back to the zero-trust principle. In most environments, user profiles are created and loaded on user workstations that indicate the level of control the user does and does not have. As a security administrator you might configure user profiles so that users cannot change the system’s time, alter system configuration files, access a command prompt, or install unapproved applications. This type of access control is referred to as nondiscretionary, meaning that access decisions are not made at the discretion of the user.

Nondiscretionary access controls are put into place by an authoritative entity (usually a security administrator) with the goal of protecting the organization’s most critical assets.

Mandatory Access Control

In a mandatory access control (MAC) model, users do not have the discretion of determining who can access objects as in a DAC model. For security purposes, an operating system that is based on a MAC model greatly reduces the amount of rights, permissions, and functionality that a user has. In most systems based on the MAC model, a user cannot install software, change file permissions, add new users, and so on. The system can be used by the user for very focused and specific purposes, and that is it. These systems are usually very specialized and are in place to protect highly classified data. Most people have never interacted directly with a MAC-based system because they are mainly used by government-oriented agencies that maintain top-secret information.

However, MAC is used behind the scenes in some environments you may have encountered at some point. For example, the optional Linux kernel security module called AppArmor allows system administrators to implement MAC for certain kernel resources. There is also a version of Linux called SELinux, developed by the NSA, that implements a flexible MAC model for enhanced security.

The MAC model is based on a security label system. Users are given a security clearance (secret, top secret, confidential, and so on), and data is classified in the same way. The clearance and classification data is stored in the security labels, which are

bound to the specific subjects and objects. When the system makes a decision about

fulfilling a request to access an object, it is based on the clearance of the subject, the
classification of the object, and the security policy of the system. This means that even
if a user has the right clearance to read a file, specific policies (e.g., requiring “need to
know”) could still prevent access to it. The rules for how subjects access objects are made
by the organization’s security policy, configured by the security administrator, enforced
by the operating system, and supported by security technologies.
NOTE Traditional MAC systems are based upon multilevel security policies,
which outline how data at different classification levels is to be protected.
Multilevel security (MLS) systems allow data at different classification levels
to be accessed and interacted with by users with different clearance levels
simultaneously.

Security label

Security label
Name:
Clearance:

Judy
Top secret

Categories:

Operations
Jack Voltaic
Cyber Guard

Figure 17-1

Name:
Classification:
Categories:

Roster.xlsx
Top secret
Threatcasting

Security label
Name:
Classification:
Categories:

A security label is made up of a classification and categories.

Planning Docs
Secret
Jack Voltaic

PART V

When the MAC model is being used, every subject and object must have a security label, also called a sensitivity label. This label contains the object's security classification and any categories that may apply to it. The classification indicates the sensitivity level, and the categories enforce need-to-know rules. Figure 17-1 illustrates the use of security labels.
The classifications follow a hierarchical structure, with one level being more trusted than another. However, the categories do not follow a hierarchical scheme, because they represent compartments of information within a system. The categories can correspond to departments (intelligence, operations, procurement), project codenames (Titan, Jack Voltaic, Threatcasting), or management levels, among others. In a military environment, the classifications could be top secret, secret, confidential, and unclassified. Each classification is more trusted than the one below it. A commercial organization might use confidential, proprietary, corporate, and sensitive. The definition of the classification is up to the organization and should make sense for the environment in which it is used.
The categories portion of the label enforces need-to-know rules. Just because someone has a top secret clearance does not mean she now has access to all top secret information.

She must also have a need to know. As shown in Figure 17-1, Judy is cleared top secret and has the codename Jack Voltaic as one of her categories. She can, therefore, access the folder with the planning documents for Jack Voltaic because her clearance is at least that of the object, and all the categories listed in the object match her own. Conversely, she cannot access the roster spreadsheet because, although her clearance is sufficient, she does not have a need to know that information. We know this last bit because whoever assigned the categories to Judy did not include Threatcasting among them.
EXAM TIP In MAC implementations, the system makes access decisions by comparing the subject's clearance and need-to-know level to the object's security label. In DAC implementations, the system compares the subject's identity to the ACL on the resource.

Software and hardware guards allow the exchange of data between trusted (high assurance) and less trusted (low assurance) systems and environments. For

instance, if
you were working on a MAC system (working in the dedicated security mode of secret)
and you needed it to communicate to a MAC database (working in multilevel security
mode, which goes up to top secret), the two systems would provide different levels of
protection. If a system with lower assurance can directly communicate with a system of
high assurance, then security vulnerabilities and compromises could be introduced.
A software guard is really just a front-end product that allows interconnectivity between
systems working at different security levels. Different types of guards can be used to carry
out filtering, processing requests, data blocking, and data sanitization. A hardware guard
is a system with two network interface cards (NICs) connecting the two systems that
need to communicate with one another. Guards can be used to connect different MAC
systems working in different security modes, and they can be used to connect different
networks working at different security levels. In many cases, the less trusted system can
send messages to the more trusted system and can only receive acknowledgments back.
This is common when e-mail messages need to go from less trusted systems to more trusted classified systems.
TIP The terms “security labels” and “sensitivity labels” can be used interchangeably.

Because MAC systems enforce strict access control, they also provide a wide range
of security, particularly dealing with malware. Malware is the bane of DAC systems.
Viruses, worms, and rootkits can be installed and run as applications on DAC systems.
Since users that work within a MAC system cannot install software, the operating system
does not allow any type of software, including malware, to be installed while the user is
logged in. But while MAC systems might seem to be an answer to all our security prayers,
they have very limited user functionality, require a lot of administrative overhead, are
very expensive, and are not user friendly. DAC systems are general-purpose computers,
while MAC systems serve a very specific purpose.

EXAM TIP Unlike DAC systems, MAC systems are considered nondiscretionary because users cannot make access decisions based on their own discretion (choice).

Role-Based Access Control

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

An RBAC model is the best system for an organization that has high employee turnover. If John, who is mapped to the Contractor role, leaves the organization, then Chrissy,
his replacement, can be easily mapped to this role. That way, the administrator does not
need to continually change the ACLs on the individual objects. He only needs to create
a role (Contractor), assign permissions to this role, and map the new user to this role.
Optionally, he can define roles that inherit access from other roles higher up in a hierarchy.
These features are covered by two components of RBAC: core and hierarchical.

PART V

A role-based access control (RBAC) model uses a centrally administrated set of controls
to determine how subjects and objects interact. The access control levels are based
on the necessary operations and tasks a user needs to carry out to fulfill her responsibilities within an organization. This type of model lets access to resources be based on
the role the user holds within the organization. The more traditional access control
administration is based on just the DAC model, where access control is specified at
the object level with ACLs. This approach is more complex because the administrator must translate an organizational authorization policy into permission when
configuring ACLs. As the number of objects and users grows within an environment,
users are bound to be granted unnecessary access to some objects, thus violating the
least-privilege rule and increasing the risk to the organization. The RBAC approach
simplifies access control administration by allowing permissions to be managed in
terms of user job roles.
In an RBAC model, a role is defined in terms of the operations and tasks the role
will carry out, whereas a DAC model outlines which subjects can access what objects
based upon the individual user identity. Let’s say we need a research and

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

Core RBAC
There is a core component that is integrated into every RBAC implementation because
it is the foundation of the model. Users, roles, permissions, operations, and sessions are
defined and mapped according to the security policy. The core RBAC

- Has a many-to-many relationship among individual users and privileges
- Uses a session as a mapping between a user and a subset of assigned roles
- Accommodates traditional but robust group-based access control

Many users can belong to many groups with various privileges outlined for each group.
When the user logs in (this is a session), the various roles and groups this user has been
assigned are available to the user at one time. If you are a member of the Accounting role,
RD group, and Administrative role, when you log on, all of the permissions assigned to
these various groups are available to you.
This model provides robust options because it can include other components when
making access decisions, instead of just basing the decision on a credential set. The
RBAC system can be configured to also include time of day, location of role, day of the
week, and so on. This means other information, not just the user ID and credential, is
used for access decisions.

Hierarchical RBAC
This component allows the administrator to set up an organizational RBAC model
that maps to the organizational structures and functional delineations required
in a specific environment. This is very useful since organizations are already set up in
a personnel hierarchical structure. In most cases, the higher you are in the chain of
command, the more access you most likely have. Hierarchical RBAC has the
following features:

• Uses role relations in defining user membership and privilege inheritance. For example, the Nurse role can access a certain set of files, and the Lab Technician role can access another set of files. The Doctor role inherits the permissions and access rights of these two roles and has more elevated rights already assigned to the Doctor role. So hierarchical RBAC is an accumulation of rights and permissions of other roles.
• Reflects organizational structures and functional delineations.
• Supports two types of hierarchies:
• Limited hierarchies Only one level of hierarchy is allowed (Role 1 inherits from Role 2 and no other role)
• General hierarchies Allows for many levels of hierarchies (Role 1 inherits Role 2's and Role 3's permissions)

Hierarchies are a natural means of structuring roles to reflect an organization's lines of authority and responsibility. Role hierarchies define an inheritance relation among roles.
Different separations of duties are provided through RBAC:

• Static separation of duty (SSD) relations Deters fraud by constraining the combination of privileges (e.g., the user cannot be a member of both the Cashier and Accounts Receivable roles).
• Dynamic separation of duty (DSD) relations Deters fraud by constraining the combination of privileges that can be activated in any session (e.g., the user cannot be in both the Cashier and Cashier Supervisor roles at the same time, but the user can be a member of both). This one warrants a bit more explanation. Suppose José is a member of both the Cashier and Cashier Supervisor roles. If he logs in as a Cashier, the Supervisor role is unavailable to him during that session. If he logs in as Cashier Supervisor, the Cashier role is unavailable to him during that session.
• Role-based access control can be managed in the following ways:
• Non-RBAC Users are mapped directly to applications and no roles are used.
• Limited RBAC Users are mapped to multiple roles and mapped directly to other types of applications that do not have role-based access functionality.
• Hybrid RBAC Users are mapped to multiapplication roles with only selected rights assigned to those roles.
• Full RBAC Users are mapped to enterprise roles.

A lot of confusion exists regarding whether RBAC is a type of DAC model or a type of MAC model. Different sources claim different things, but in fact RBAC is a model in its own right. In the 1960s and 1970s, the U.S. military and NSA did a lot of research on the MAC model. DAC, which also sprang to life in the 1960s and 1970s, has its roots in the academic and commercial research laboratories. The RBAC model, which started gaining popularity in the 1990s, can be used in