**There are several categories for access control techniques and the CISSP CIB specifically mentions four: discretionary access control (DAC), mandatory access control (MAC), role-based access control (role-BAC), and rule-based access control (rule-BAC).**

**Permissions** In general, *permissions* refer to the access granted for an object and determine what you can do with it. If you have read permission for a fi le, you’ll be able to open it and read it. You can grant user permissions to create, read, edit, or delete a fi le on a fi le server. Similarly, you can grant user access rights to a fi le, so in this context, access

rights and permissions are synonymous. For example, you may be granted read and execute permissions for an application fi le, which gives you the right to run the application. Additionally, you may be granted data rights within a database, allowing you to retrieve or update information in the database

**Rights** A *right* primarily refers to the ability to take an action on an object. For example, a user might have the right to modify the system time on a computer or the right to restore backed-up data. This is a subtle distinction and not always stressed. However, you’ll rarely see the right to take action on a system referred to as a permission

**Privileges** *Privileges* are the combination of rights and privileges. For example, an administrator for a computer will have full privileges, granting the administrator full rights and permissions on the computer. The administrator will be able to perform any actions and access any data on the computer.

**Implicit Deny** A basic principle of access control is implicit deny and most authorization mechanisms use it. The implicit deny principle ensures that access to an object is denied unless access has been explicitly granted to a subject. For example, imagine an administrator explicitly grants Jeff Full Control permissions to a fi le but does not explicitly grant permissions to anyone else. Mary doesn’t have any access even though the administrator didn’t explicitly deny her access. Instead, the implicit deny principle denies access to Mary and everyone else except for Jeff.

**Access Control Matrix** An access control matrix is a table that includes subjects, objects, and assigned privileges. When a subject attempts an action, the system checks the access control matrix to determine if the subject has the appropriate privileges to perform the action. For example, an access control matrix can include a group of fi les as the objects and a group of users as the subjects. It will show the exact permissions authorized by each user for each fi le. Note that this covers much more than a single access control list (ACL). In this example, each fi le listed within the matrix has a separate ACL that lists the authorized users and their assigned permissions.

**Capability Tables** Capability tables are another way to identify privileges assigned to subjects. They are different from ACLs in that a capability table is focused on subjects (such as users, groups, or roles). For example, a capability table created for the accounting role will include a list of all objects that the accounting role can access and will include the specific privileges assigned to the accounting role for these objects. In contrast, ACLs are focused on objects. An ACL for a fi le would list all the users and/or groups that are authorized access to the fi le and the specific access granted to each

The difference between an ACL and a capability table is the focus. ACLs are object focused and identify access granted to subjects for any specific object. Capability tables are subject focused and identify the objects that subjects can access.

**Constrained Interface** Applications use constrained interfaces or restricted interfaces to **restrict what users can do or see based on their privileges. Users with full privileges have access to all the capabilities of** the application. Users with restricted privileges have limited access. Applications constrain the interface using different methods. A common method is to hide the capability if the user doesn’t have permissions to use it. For example, commands might be available to administrators via a menu or by right-clicking an item, but if a regular user doesn’t have permissions, the command does not appear. Other times, the application displays the menu item but shows it dimmed or disabled. A regular user can see the menu item but will not be able to use it.

**Content-Dependent Control** Content-dependent access controls restrict access to data based on the content within an object. **A database view is a content-dependent control**. A view retrieves specific columns from one or more tables, creating a virtual table. For example, a customer table in a database could include customer names, email addresses, phone numbers, and credit card data. A customer-based view might show a user only the customer names and email addresses but nothing else. Users granted access to the view can see the customer names and email addresses but cannot access data in the underlying table.

**Context-Dependent Control** Context-dependent access controls require specific activity before granting users access. **As an example, consider the data flow for a transaction selling digital products online. Users add products to a shopping cart and begin the checkout process. The first page in the checkout flow shows the products in the shopping cart, the next page collects credit card data, and the last page confirms the purchase and provides instructions for downloading the digital products. The system denies access to the download page if users don’t go through the purchase process first. It’s also possible to use date and time controls as context-dependent controls. For example, it’s possible to restrict access to computers and applications based on the current day and/or time. If users attempt to access** the resource outside of the allowed time, the system denies them access.

**Need to Know** This principle ensures that subjects are granted access only to what they need to know for their work tasks and job functions. Subjects may have clearance to access classified or restricted data but are not granted authorization to the data unless they actually need it to perform a job.

**Least Privilege** The principle of least privilege ensures that subjects are granted only the privileges they need to perform their work tasks and job functions. This is sometimes lumped together with need to know. The only difference is that least privilege will also include rights to take action on a system.

**Separation of Duties and Responsibilities** This principle ensures that sensitive functions are split into tasks performed by two or more employees. It helps to prevent fraud and errors by creating a system of checks and balances.

Organizations implement access controls using a defense-in-depth strategy. This uses multiple layers or levels of access controls to provide layered security.

**Discretionary Access Controls**

**A system that employs discretionary access controls (DACs) allows the owner, creator, or data custodian of an object to control and define access to that object. All objects have owners, and access control is based on the discretion or decision of the owner. For example, if a user creates a new spreadsheet fi le, that user is the owner of the fi le. As the owner, the user can modify the permissions of the fi le to grant or deny access to other users. Identity based** access control is a subset of DAC because systems identify users based on their identity and assign resource ownership to identities. A DAC model is implemented using access control lists (ACLs) on objects. Each ACL defines the types of access granted or denied to subjects. **It does not offer a centrally controlled management system because owners can alter t**he ACLs on their objects at will. Access to objects is easy to change, especially when compared to the static nature of mandatory access controls. Within a DAC environment, administrators can easily suspend user privileges while they are away, such as on vacation. Similarly, it’s easy to disable accounts when users leave the organization.

**Nondiscretionary Access Controls**

The major difference between discretionary and nondiscretionary access controls is in how they are controlled and managed. **Administrators centrally administer nondiscretionary access controls and can make changes that affect the entire environment. In contrast, discretionary access** control models allow owners to make their own changes, and their changes don’t affect other parts of the environment.

In a non-DAC model, access does not focus on user identity. Instead, a static set of rules governing the whole environment manages access. Non-DAC systems are centrally controlled and easier to manage (although less flexible). In general, any model that isn’t a discretionary model is a nondiscretionary model. This includes rule-based, role-based, and lattice-based access controls

**Role-based Access Control**

Systems that employ role-based or task-based access controls define a subject’s ability to access an object based on the subject’s role or assigned tasks. Role-based access control (role-BAC) is often implemented using groups. As an example, a bank may have loan officers, tellers, and managers. Administrators can create a group named Loan Officers, place the user accounts of each loan officer into this group, and then assign appropriate privileges to the group.

**This helps enforce the principle of least privilege by preventing privilege creep. Privilege creep is the tendency for privileges to accrue to users over time as their roles and access needs change.**

Role-based access controls are useful in dynamic environments with frequent personnel changes because administrators can easily grant multiple permissions simply by adding a new user into the appropriate role.

**It’s easy to confuse DAC and role-BAC because they can both use groups to organize users into manageable units, but they differ in their deployment and use. In the DAC model, objects have owners and the owner determines who has access. In the role-BAC model, administrators determine subject privileges and assign appropriate privileges to roles or groups. In a strict role-BAC model, administrators do not assign privileges to users directly but only grant privileges by adding user accounts to roles or groups.**

Another method related to role-BAC is the task-based access control (TBAC). TBAC is similar to role-BAC, but instead of being assigned to one or more roles, each user is assigned an array of tasks. These items all relate to assigned work tasks for the person associated with a user account. Under TBAC, the focus is on controlling access by assigned tasks rather than by user identity.

**Rule-based Access Controls**

A rule-based access control (rule-BAC) uses a set of rules, restrictions, or filters to determine what can and cannot occur on a system. It includes granting a subject access to an object, or granting the subject the ability to perform an action. A distinctive characteristic about rule-BAC models is that they have global rules that apply to all subjects. One common example of a rule-BAC model is a firewall. Firewalls include a set of rules or filters within an ACL, defined by an administrator. The firewall examines all the traffic going through it and only allows traffic that meets one of the rules. Firewalls include a final rule (referred to as the implicit deny rule) denying all other traffic. For example, the last rule might be deny all to indicate the firewall should block all other traffic in or out of the network. In other words, if traffic didn’t meet the condition of any previous explicitly defined rule, then the final rule ensures that the traffic is blocked. This final rule is sometimes viewable in the ACL so that you can see it. Other times, the implicit deny rule is implied as the final rule but is not explicitly stated in the ACL..

**Attribute-based Access Controls**

Traditional rule-BAC models include global rules that apply to all users. However, an advanced implementation of a rule-BAC is an attribute-based access control (ABAC). ABAC models use policies that include multiple attributes for rules. Many software defined networking applications use ABAC models. As an example, CloudGenix has created a software-defined wide area network (SD-WAN) solution that implements policies to allow or block traffic. Administrators create ABAC policies using plain language statements such as “Allow Managers to access the WAN using tablets or smartphones.” This allows users in the Managers role to access the WAN using tablet devices or smartphones. Notice how this improves the rule-BAC model. The rule-BAC applies to all users, but the ABAC can be much more specific.

**Mandatory Access Controls**

A mandatory access control (MAC) model relies on the use of classification labels. Each classification label represents a security domain , or a realm of security. A security domain is a collection of subjects and objects that share a common security policy. For example, a security domain could have the label Secret, and the MAC model would protect all objects with the Secret label in the same manner. Subjects are only able to access objects with the Secret label, when they have a matching Secret label. Additionally, the requirement for subjects to gain the Secret label is the same for all subjects. Users have labels assigned to them based on their clearance level, which is a form of privilege. Similarly, objects have labels, which indicate their level of classification or sensitivity. For example, the U.S. military uses the labels of Top Secret, Secret, and Confidential to classify data. Administrators can grant access to Top Secret data to users with Top

Secret clearances. However, administrators cannot grant access to Top Secret data to users with lower-level clearances such as Secret and Confidential.

The MAC model is often referred to as a lattice-based model. The horizontal lines labeled Confidential, Private, Sensitive, and Public mark the upper bounds of the classification levels.

key point about the MAC model is that every object and every subject has a label. These labels are predefined and the system makes a determination of access based on assigned labels.

**Hierarchical Environment** A hierarchical environment relates various classification labels in an ordered structure from low security to medium security to high security, such as Confidential, Secret, and Top Secret, respectively. Each level or classification label in the structure is related. Clearance in one level grants the subject access to objects in that level as well as to all objects in lower levels but prohibits access to all objects in higher levels. For example, someone with a Top Secret clearance can access Top Secret data and Secret data

**Compartmentalized Environment In a compartmentalized environment, there is no relationship between one security domain and another. Each domain represents a separate isolated compartment. To gain access to an object, the subject must have specific clearance for its security domain**

**Hybrid Environment** A hybrid environment combines both hierarchical and compartmentalized concepts so that each hierarchical level may contain numerous subdivisions that are isolated from the rest of the security domain. A subject must have the correct clearance and the need to know data within a specific compartment to gain access to the compartmentalized object. A hybrid MAC environment provides granular control over access, but becomes increasingly difficult to manage as it grows.

**Advanced Persistent Threat**

Any threat model should take into account the existence of known threats, and a relatively new threat is an advanced persistent threat (APT) . An APT refers to a group of attackers who are working together and are highly motivated, skilled, and patient. They have advanced knowledge and a wide variety of skills to detect and exploit vulnerabilities.

**Access Aggregation Attacks**

**Access aggregation refers to collecting multiple pieces of non sensitive information and combining (i.e., aggregating) them to learn sensitive information**. In other words, a person or group may be able to collect multiple facts about a system and then use these facts to launch an attack.

**Reconnaissance attacks are access aggregation attacks that combine multiple tools to identify multiple elements of a system, such as IP addresses, open ports, running services, operating systems, and more. Attackers also use aggregation attacks against databases**.

**Combining defense-in-depth, need-to-know, and least privilege principles helps prevent access aggregation attacks.**

**Dictionary Attacks**

A *dictionary attack* is an attempt to discover passwords by using every possible password in a predefined database or list of common or expected passwords. In other words, an attacker starts with a database of words commonly found in a dictionary

**Brute-Force Attacks**

A brute-force attack is an attempt to discover passwords for user accounts by systematically attempting all possible combinations of letters, numbers, and symbols. Attackers don’t typically type these in manually but instead have programs that can programmatically try all the combinations. A hybrid attack attempts a dictionary attack and then performs a type of brute-force attack with one-upped-constructed passwords.

**Birthday Attack**

A birthday attack focuses on finding collisions. Its name comes from a statistical phenomenon known as the birthday paradox.

**Rainbow Table Attacks**

It takes a long time to find a password by guessing it, hashing it, and then comparing it with a valid password hash. However, a rainbow table reduces this time by using large databases of precomputed hashes. Attackers guess a password (with either a dictionary or a brute-force method), hash it, and then put both the guessed password and the hash of the guessed password into the rainbow table. A password cracker can then compare every hash in the rainbow table against the hash in a stolen password database fi le. A traditional password-cracking tool must guess the password and hash it before it can compare the hashes. However, when using the rainbow table the password cracker doesn’t spend any time guessing and calculating hashes. It simply compares the hashes until it finds a match. This can significantly reduce the time it takes to crack a password.

**Many systems commonly salt passwords to reduce t the effectiveness of rainbow table attacks. A salt is a group of random bits, added to a password before hashing it.**

**Sniffer Attacks**

Sniffing captures packets sent over a network with the intent of analyzing the packets. A sniffer (also called a packet analyzer or protocol analyzer) is a software application that captures traffic traveling over the network. Administrators use sniffers to analyze network traffic and troubleshoot problems. Of course, attackers can also use sniffers.

**Spoofing Attacks**

**Spoofing (also known as masquerading**) is pretending to b g e something, or someone, else. There is a wide variety of spoofing attacks.

**Phishing**

Phishing is a form of social engineering that attempts to t g rick users into giving up sensitive information, opening an attachment, or clicking a link.

**Spear Phishing**

**Spear phishing is a form of phishing targeted to a specific group of use**rs, such as employees within a specific organization. It may appear to originate from a colleague or co-worker within the organization or from an external source.

**Whaling**

**Whaling is a variant of phishing that targets senior or high-level executives such as CEOs and presidents within a company.**

**Vishing**

Vishing is a variant of phishing that uses the phone system or VoIP. A common attack uses an automated call to the user explaining a problem with a credit card account. The user is encouraged to verify or validate information such as a credit card number, expiration date, and security code on the back of the card. Vishing attacks commonly spoof the caller ID number to impersonate a valid bank or financial institution.

**A side-channel attack is a passive, noninvasive attack intended to observe the operation of a device. When the attack is successful, the attacker is able to learn valuable information contained within the card, such as an encryption key**. **Side-channel attacks analyze the information sent to the reader..**

.