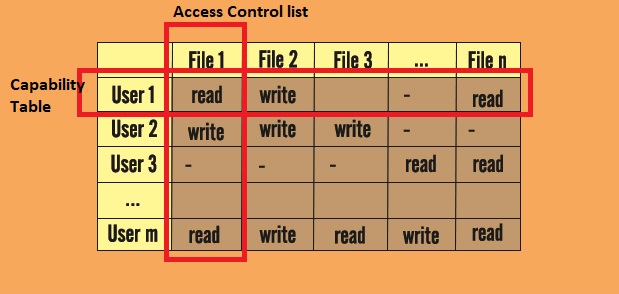
**Chapter 14: Controlling and Monitoring Access**

**Authorization Mechanisms**

The implicit deny principle ensures that access to an object is denied unless access has been explicitly granted to a subject.

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.

ACL is associated with an object and lists the access different subjects have on that object. The Capability Table is similar to ACL. However, it lists capabilities of subjects on different objects.



Applications use constrained interfaces or restricted interfaces to restrict what users can do or see based on their privileges.

Context-dependent access controls require specific activity before granting users access. For eg, users are denied access to digital content before they make a valid purchase.

Content-dependent access controls restrict access to data based on the content within an object. A database view is an example of a content-dependent control.

Principal of Need to Know ensures that subjects are granted access only to what they need to know for their work tasks and job functions.

The principle of least privilege ensures that subjects are granted only the privileges they need to perform their work tasks and job functions.

The separation of duties and responsibilities principle ensures that sensitive functions are split into tasks performed by two or more employees.

**Access Control Models**

Discretionary Access Control - A key characteristic of the Discretionary Access Control (DAC) model is that every object has an owner and the owner can grant or deny access based on their own discretion.

Role-Based Access Control - Instead of assigning permissions directly to users, user accounts are placed in groups or roles and administrators assign privileges to these groups/ roles.

Rule-Based Access Control - A key characteristic of the rule-based access control model is that it applies global rules to all subjects. As an example, a firewall uses rules that allow or block traffic to all users equally.

Attribute-Based Access Control - ABAC allows administrators to create rules using multiple attributes and these rules are then applied to subjects. For eg - “Allow Managers to access the application when using a mobile device.” The ABAC model is commonly used in SDNs.

Mandatory Access Control - A key characteristic of the Mandatory Access Control (MAC) model is the use of labels applied to both subjects and objects. These labels are predefined, and the system determines access based on assigned labels. Each classification label represents a security domain, or a realm of security.

Classifications within a MAC model use one of the following three types of environment:

* **Hierarchical Environment** - Clearance in one level grants the subject access to objects in that specific level as well as to all objects in lower levels but prohibits access to all objects in higher levels.
* **Compartmentalized Environment** - A compartmentalized environment ignores the levels, and instead only allows access for individual compartments on any level. To gain access to an object, the subject must have specific clearance for the object’s 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.

Risk-Based Access Control - A risk-based access control model grants access after evaluating risk. It evaluates the environment and the situation and makes risk-based decisions using policies embedded within software code.

The following bullets outline the key points about SAML:

* SAML 2.0 is an open XML-based standard.
* It utilizes three entities: a principal (such as a user), a service provider (such as a website), and an identity provider (a third party that holds the authentication and authorization information).
* It can provide authentication, authorization, and attribute information on the principal.

The following bullets outline the key points about OAuth:

* It’s an authorization framework, not an authentication protocol.
* It exchanges information using APIs.
* An app obtains an access token from an identity provider.
* Later, the app includes the access token for authorization.

The following bullets outline the key points about OpenID:

* OpenID is an authentication standard.
* It is maintained by the OpenID Foundation.
* An OpenID provider provides decentralized authentication.
* Users enter their Open ID identifier (such as bobsmith2021.myopenid.com) on a site and the OpenID provider verifies the identifier.

The following bullets outline the key points about OIDC:

* OIDC is an authentication layer using OAuth 2.0.
* It builds on the OpenID authentication standard.
* It provides both authentication and authorization.
* It builds on OpenID but uses a JSON Web Token.

Several protocols provide authentication, authorization, and accounting and are referred to as AAA protocols. Two common remote access protocols that provide authentication, authorization, and accounting are RADIUS and TACACS+.

Kerberos Authentication - The Domain controller plays the key role in kerberos authentication. It stores password hashes of all the users in a lookup table and acts as the key distribution center KDC.

1. Client sends the Authentication Server Request (or AS\_REQ) containing a timestamp and the username which is AES encrypted with the user's password hash.
2. DC looks up the password hash associated with the specific user and attempts to decrypt the timestamp.
3. If the decryption process is successful and the timestamp is not a duplicate (a potential replay attack), the authentication is considered successful.
4. DC replies with an Authentication Server Reply (AS\_REP) that contains a session key (since Kerberos is stateless) and a Ticket Granting Ticket (TGT).
5. The session key is encrypted using the user's password hash, and may be decrypted by the client and reused.
6. TGT contains information regarding the user(eg- group memberships) and the session key. To avoid tampering, it is encrypted by a secret key known only to the KDC.
7. KDC considers the client authentication complete now. By default, the TGT will be valid for 10 hours, after which a renewal occurs. This renewal does not require the user to re-enter the password.

Accessing Resources using TGT

1. The client sends its TGT back to the KDC with a request for access to the resource.
2. The KDC verifies that the TGT is valid and checks its access control matrix to verify that the user has sufficient privileges to access the requested resource.
3. The KDC generates a service ticket and sends it to the client.
4. The client sends the ticket to the server or service hosting the resource.
5. The server or service hosting the resource verifies the validity of the ticket.
6. Once identity and authorization are verified, Kerberos activity is complete. The server or service host then opens a session with the client and begins communications or data transmission.

Remote Authentication Dial-in User Service (RADIUS) centralizes authentication for remote access connections, such as with VPNs or dial-up access. A user can connect to any network access server, which then passes on the user’s credentials to the RADIUS server to verify authentication and authorization and to track accounting. In this context, the network access server is the RADIUS client, and a RADIUS server acts as an authentication server. RADIUS uses the User Datagram Protocol (UDP) by default and encrypts only the password’s exchange. It doesn’t encrypt the entire session.

Cisco developed TACACS+ is an improvement over RADIUS. It encrypts all of the authentication information, and not just the password’s exchange. It uses TCP instead of UDP to provide a higher level of reliability.

A spraying attack is a special type of brute-force attack. Attackers spray the same password across multiple accounts, attempting to bypass account lockout security controls.

Credentials stuffing is a brute force attack in which if an attacker gets credentials of a user for one platform, they try to brute force with the same credentials across multiple platforms to maximize the compromise.

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

Rainbow tables use pre-computed hashes to crack compromised password hashes.

A pass-the-hash (PtH) attack allows an attacker to send a captured hash of a password to an authenticating service. Penetration testers and attackers use Mimikatz and other tools (such as DCSync) to capture hashes, and then use the hashes to simulate the login process.

A silver ticket uses the captured NTLM hash of a service account to create a ticket-granting service (TGS) ticket. The silver ticket grants the attacker all the privileges granted to the service account.

If an attacker obtains the hash of the Kerberos service account (KRBTGT), they can create tickets at will within Active Directory. This gives them so much power it is referred to as having a golden ticket.

Kerberoasting refers to attackers harvesting TGS tickets and cracking them offline.