Identity & Access Management (IAM)

Identity and access management (IAM) is the cybersecurity discipline that deals with provisioning and protecting digital identities and user access permissions in an IT system.

The goal of IAM is to stop hackers while allowing authorized users to easily do everything they need to do, but not more than they're allowed to do.

- Administration
- Authentication
- Authorization
- Auditing

Administration

Identity administration—also referred to as “identity management” or “identity lifecycle management”—is the process of creating, maintaining and securely disposing of user identities in a system.

Digital identities capture traits such as a user's name, login credentials, job title and access rights.

Digital identities are typically stored in a central database or directory, which acts as a single source of truth.

Auditing

Auditing means making sure that the IAM system and its components—administration, authentication and authorization—are working properly.

Auditing entails tracking and logging what users do with their access rights to ensure that nobody, including hackers, has access to anything that they shouldn’t, and that authorized users don’t abuse their privileges.

Authentication

Authentication (“auth” for short) is the process that verifies that a user is who they claim to be.

While a password is the most basic form of authentication, it is also one of the weakest.

Hijacking valid user accounts is one of the most common ways that attackers break into networks, accounting for 30% of cyberattacks

Most IAM implementations today use more advanced authentication methods, such as two-factor authentication (2FA) or multifactor authentication (MFA)

Authentication Factor: are items, characteristics or bits of information that only user has.

There are four types of authentication factors that user can provide to prove their identity:

- Knowledge Factor: Some pieces of information that, theoretically, only the user would know, such as password, PINs and answer to security questions. This is common but easiest to steal or crack.
- Possession Factor: Something that user owns. Like a dedicated hardware security token generator or mobile device in which user can install authenticator app to generate OTPs or receive OTPs via SMS
- Inherence Factor: These are the physical traits user has. This includes biometric authentication via facial recognition or fingerprint scan.
- Behavioral Factor: Patterns, such as a person’s typical IP address range, hours of activity and average typing speed. Adaptive authentication schemes often use behavioral factors to assess a user’s risk level.

Single Sign-On (SSO)

Is an authentication scheme that lets users log in once using a single set of credentials, and access multiple applications during the same session.

It’s used often to manage authentication and secure access to company intranets or extranets, student portals, public cloud service, and other environments where users need to move between different applications to get their work done.

Types of SSO:

- Adaptive SSO :
  Adaptive SSO requires an initial set of login credentials, but prompts for additional authentication factors or a new login when additional risks emerge—such as when a user logs in from a new device or attempts to access particularly sensitive data or functionality.
- Federated identity management (FIM) :
  Federated identity management, or FIM, is a superset of SSO. While SSO is based on a digital trust relationship among applications within a single organization's domain, FIM extends that relationship to trusted third parties, vendors, and other service providers outside the organization. For example, FIM might enable a logged-in employee to access third-party web applications (e.g., Slack or WebEx) without an additional log-in, or with a simple username-only log-in.
- Social Login :
  Social login enable end users to authenticate with applications using the same credentials they use to authenticate with popular social media sites.

Benefits of SSO:

- Reduced password fatigue
- Fewer password and credential related vulnerabilities
- Fewer help desk calls
- Simplified security management

Two Factor Authentication (2FA)

Two-factor authentication (2FA) is a way of verifying a user’s identity by asking for exactly two pieces of proof, such as the password to an online account (the first factor) and a one-time passcode from an authenticator app (the second factor).

2FA is the most common form of multifactor authentication (MFA), which refers to any authentication method where users must supply more than one authentication factor to prove their identity.

Benefits:

> Compromised credentials and phishing are among the most common cyberattack vectors, both these vectors works by stealing password to break into multiple account. Two-factor authentication helps thwart unauthorized access by adding an extra layer of security. Even if hackers can steal a password, they still need a second factor to gain access to an account.

Some other authentication methods are: Biometric authentication, CAPTCHA (Completely Automated Public Turing test to tell Computer and Human Apart), FIDO, FIDO2

Authorization

Is the process of granting verified users the appropriate levels of access to a resource.

Authentication and authorization are deeply linked; authentication is the prerequisite for authorization.

Authentication and authorization together form the access management component of Identity and access management.

To set user access permissions, different organizations take different approaches. One common access control framework is role-based access control (RBAC), in which users’ privileges are based on their job functions.

Most access control frameworks are designed according to the principle of least privilege.

Role Based Access Control (RBAC)

Role-based access control (RBAC) is a model for authorizing end-user access to systems, applications and data based on a user’s predefined role. For example, a security analyst can configure a firewall but can’t view customer data, while a sales rep can see customer accounts but can’t touch firewall settings.

In an RBAC system, an administrator assigns each individual user one or more roles. Each new role represents a set of permissions or privileges for the user.

Attribute Based Access Control (ABAC)

ABAC analyzes the attributes of users, objects and actions—such as a user's name, a resource's type and the time of day—to determine whether access will be granted.

The difference between RBAC and ABAC is that ABAC dynamically determines access permissions in the moment based on several factors, while RBAC determines access permissions based solely on the user's predefined role.

Access Control List (ACL)

ACL is a basic access control system that references a list of users and rules to determine who can access a system or resource, and which actions they may perform.

The difference between ACL and RBAC is that an ACL individually defines the rules for each user, while RBAC systems assign access rights based on roles.

For large organizations, RBAC is considered a better option for access control because it is more scalable and easier to manage than ACL.

Open Authorization (OAuth)

OAuth is an open-standard authorization framework that grants applications access to an end user's protected resources—such as their photos, calendars or social media posts—without requiring the login or password to the user's account.

Websites and third-party applications that ask users to "Sign in with Google?" or "Allow access to your account information?" are common use cases for OAuth. The OAuth protocol enables users to easily grant these applications access to their account data without sharing their user credentials.

How OAuth Works ?

Unlike other frameworks that rely on usernames and passwords, OAuth is an authorization protocol that is based on access tokens.

OAuth tokens commonly use the JSON Web Token (JWT) standard because of its ability to transmit data securely.

The primary components of the OAuth framework are:

- Resource Owner: The end user that owns the account the application seeks to access, such as Facebook or Google account. The resource owner provide consent for the application to access certain protected resource in that account.
- Resource Server: This is the server that stores the protected resources on behalf of the user. The resource server accepts and validates OAuth tokens it receives from the client, then provides the user data that the resource owner has consented to release.
- Client: The client is the application, website, API or device that requests access. It requests authorization from the authorization server by presenting a client ID. Then, after the resource owner provides consent, the client receives an access token it can use to access protected resources on the resource server.
- Authorization Server: This is the primary server that drives the OAuth protocol. It operates two endpoints to grant access to the resource server. The **authorization endpoint** prompts the resource owner to provide consent for specific protected resources. The **token endpoint** then receives the token request from the OAuth client and generates new access tokens that grant access to the resources.
- Scope: Scopes are the access parameters for protected resources on the resource server.