Identity

In computing, “identity” is a representation of a person, application or device.

Example: person, email, application, a printer

Usually requires a password, a secrete key or a certificate (multi-factor authentication) to prove identity.

Many applications require you to log in to use some of functionality. Thus, different people have different permissions.

Before Microsoft standardize / cloud, identity is handled as follows:

Client App, Web browser, Mobile App (username & password) --> Server, Web site --> Database

Over the time, people found ways (cache instead of plain text) to save password, but they are also hacked.

Microsoft came up with “Active Directory” (for cloud we have Azure Active Directory / Microsoft Entra ID). Activate Directory uses protocols such as LDAP and Kerberos for communication.

Thus, as a user, instead of coding your own security backend, you can use APIs to Entra ID to handle authentication and authorization.

Benefits of using Entra ID as a service (for third party application and custom applications)

- Security
- Reduced development time + easier support
- There are additional features: Use AI to logging patterns (logging someone from country/region where you do not have any business), Audit features
- Centralized administration (can flag some users)
- Single sign-on (SSO) (same username can be used to logging company laptop and then also Azure)
- Integrates with other Azure services

Authentication and Authorization

Authentication is a user providing who they are (user Id and password).

Authorization is ensuring that a user is permitted to perform an action (read/edit/delete a certain file)

Nowadays we are moving away from all authenticated users having admin access.

External Identities in Azure

In here, we see how Azure help with users who are not in company internal directory (that is, partners, contactors, end users). This is where B2B B2C comes in.

- In simple terms, the ability to give secure access to your resources or applications for people outside of your organization.

Business-to-Business (B2B)

Can view as collaborators (formal relationships).

B2B means inviting partners, contractors or vendors into your organization.

These are guests (guest users). They use their own credentials to log in (Gmail, company log in). We can assign roles and permissions, same as internal users.

Business-to-Consumer (B2C)

B2C means inviting customers/end users to use your applications. They do not appear in your organization at all.

Users use “social identities” to sign like FB, Google, ...

Single Sign-On (SSO)

Example: We may log into Microsoft 365 and then we will be using Microsoft Words, outlook, Excel, … without typing password every time.

Imagine your company has 10 apps for various things such as time entry, file storage, design software, etc. In typical set up here we may need unique usernames and passwords. However, when try to change it, we may have to change it in every place. The solution is SSO.

This is a feature of Entra ID. Entra ID uses a token that proves who you are. That token is trusted by all other applications in your company. Instead of your password, it checks your token. Authenticated once, trusted everywhere.

How it works?

- Every application is registered with your Entra ID
- The app trusts your Entra ID
- The token was encrypted by Entra ID and can only be decrypted because of your public key

This enables fewer logins, strong password. For organization, it gives better security, simplified management (instead of 10 passwords for 1 user, one password for one user).

Entra Conditional Access

These are options we can enable, and we may have to pay for these services.

It is important to note not all attempts to log into a system are equally safe. We can set rules to identity risky attempts to log into the system.

Safe attempt: When someone logs in to an application from inside your office building as do every day.

Risky attempts: (1) Someone try to log in from a country that particular employee has never logged in. It could be someone on a sales trip and try to log from there. (2) Try to log in after long time. (3) Try to log with a new device.

Thus, Entra Conditional Access will use different signals (location, time, device) to determine whether it is a risky attempt. Then, either block or require multi factor authorization (MFA).

Multi-Factor Authentication (MFA or 2FA)

In general MFA means we require 2 or more pieces of evidence (factors) in order to log in. In traditional username and password consider as a one factor. A password can be guessed, hackable, and so on.

Type of "factors": Something you know

Somethings you have

Something you are

Usually, your username is not a factor as it is public information like email address.

Password is "something you know", if kept private.

Your mobile phone could be "something you have". They may sent SMS, or use authentication app

Your fingerprint or face scan can be "something you are"

Entra ID Global Administrator account can be configured to use MFA for free:

- Microsoft Authenticator App
- SMS
- Voice call
- ....

Passwordless

Regular passwords: Less security + convenient

Passwords + 2FA: High security + inconvenient

Passwordless authentication: High security + convenient

- Using gestures to sign in (some phones have this)
- Using a PIN or biometric recognition with windows devices (iris, face, fingerprint)

Since this is happening locally, we can use it in different devices.

Interesting feature: Auto-lock the device when you are away (via Bluetooth)

Now we talk about other half, “Authorization”.

There are a couple of ways to do this.

Role-Based Access Control (RBAC)

Here we will be working with role, and this will be primary interface for assign permission.

This is Microsoft’s preferred solution for authorization.

Here we can use built in roles or create own roles for the organization. Note: We should differentiate Entra administrator and custom roles.

Examples of roles:

Developer Operations IT Security: For all the developers we may assign a role, which allow them to crate apps, restart apps, scale, and so on. If we want to make sure they cannot initiate backups, we can remove that permission from that role.

Once determine the roles then we will assign users to those roles.

Note: We try to avoid assign privileges to users instead of roles (maintain organization)

Three Basic Roles:

Reader: Read-only type roles. Have access to resources but cannot make changes.

Contributor: Gives full access to that particular resource or all resources, but cannot share permission

Owner: Gives full access to that particular resource or all resources, and it allows assigning permission to other

Zero-Trust Model for Security

In few years back, IT security focused on protecting border (boundary between company and public internet), assuming inside the corporate network is safe.

Now, hackers may get into internal network. Thus, “zero-trust model for security”. Force everyone to prove their identity.

Zero-Trust Principles

- Verify explicitly: Verify identity and then use the token all day
- Use least privileged access: Who has access to what
- Assume breach: how applications talk to each other (if one application gathers all the information)

Here, we can use every available method to validate identity.

Newer models:

- Just-in-time (JIT): Elevate permission for a limited time to perform a certain task (start a VM, restart a VM)
- Just-enough-access

Security even inside the network: encryption, segmentation, threat detection

Defence in Depth

This is the idea that you are more secure when you have more defence in place. Instead of just one fire wall, mix and match different defence.

Physical location: door lock, fingerprint scanners

Data: virtual network endpoint

Compute: limit remote desktop access, Windows Update

Microsoft Defender for Cloud

This is a cloud native security solution that will protect Azure, hybrid resources.