Identity & Access Management

Seven pillars, one end-to-end flow,
and the industry context behind every control.

HOW TO USE THIS PLAYBOOK

— Each pillar is a self-contained play: what it is, the moving parts, a real-world example, and why it matters operationally.
— Run the End-to-End Flow on page 6 as a walkthrough for onboarding sessions or architecture reviews.
— Use the memory hook below in interviews, handovers, and design reviews.

I AM SAFE	*The seven pillars, in order*	
I	**Identity**	*Knowing who is requesting access*
A	**Authentication**	*Proving it is really you*
M	**(per)Missions — Authorization**	*Controlling what you can do*
S	**SSO / Federation**	*Log in once, access everything*
A	**Access Tokens & Sessions**	*Your temporary digital pass*
F	**Full Lifecycle (Provisioning)**	*From hire to retire*
E	**Entry / Directory (LDAP)**	*The identity phone book*

01 IDENTITY BASICS

Who are you? — the foundation of every security decision

DEFINITION

An identity is the record of who or what is requesting access. Every downstream security decision resolves back to it.

KEY CONCEPTS

— **Identity** — who or what is requesting access (user, app, machine)
— **Digital identity** — username, email, employee ID stored in a system
— **Identity Provider (IdP)** — the system that holds and manages identities (Okta, Entra ID)
— **Attributes** — name, department, role, location tied to the identity
— **Principal** — the formal term for any entity with an identity (user, service, bot)

REAL-WORLD EXAMPLE

When you join a company, HR creates a profile: name, email, employee ID, department. That profile is your digital identity. It lives in the IdP, and every system you log into checks it first.

WHY IT MATTERS

— Every security decision starts with knowing who is asking
— Mismanaged identities are the leading cause of breaches (stolen credentials)
— Cloud estates hold thousands of machine identities — all need ownership
— SOC 2 and ISO 27001 require a full inventory of identities

02 AUTHENTICATION

Prove it is really you — verifying identity before granting access

DEFINITION

Authentication verifies that someone is who they claim to be, using evidence they provide.

KEY CONCEPTS

— **Something you know** — password, PIN, security question
— **Something you have** — phone OTP, hardware key (YubiKey)
— **Something you are** — fingerprint, face scan (biometrics)
— **MFA** — combining two or more factors for stronger proof
— **Passwordless** — magic links, passkeys, biometrics only
— **OAuth 2.0 / OIDC** — protocols apps use to verify identity via a trusted provider

REAL-WORLD EXAMPLE

You log into your bank: password (something you know), then a six-digit code by text (something you have). Together that is MFA. Even with a stolen password, an attacker cannot get in without your phone.

WHY IT MATTERS

— Over 80% of breaches involve compromised passwords — MFA stops most of them
— PCI-DSS and HIPAA require MFA on sensitive systems
— Passwordless authentication sharply reduces phishing risk
— Machine-to-machine calls need authentication too (API keys, certificates)

03 AUTHORIZATION

What can you do? — controlling access after identity is confirmed

DEFINITION

Authorization determines what an authenticated identity is allowed to do: read, write, delete, approve, administer.

KEY CONCEPTS

— **RBAC** — access based on role (Admin, Editor, Viewer)
— **ABAC** — access based on attributes (department = Finance AND location = HQ)
— **Least privilege** — grant the minimum access needed, nothing more
— **Permission** — a specific allowed action, e.g. 'read invoice'
— **Policy** — a rule that groups permissions, e.g. an AWS IAM policy
— **ACL** — a list of who can access one specific resource

REAL-WORLD EXAMPLE

In a hospital system, doctors view and edit patient records, nurses view but cannot edit, and billing staff see only financial data. Same system, same login — what you can do depends on your role.

WHY IT MATTERS

— Privilege abuse is a top internal threat; sound authorization limits the blast radius
— AWS, Azure and GCP are entirely policy-driven — IAM is the control plane
— Audit trails depend on accurate roles and permissions
— Zero Trust demands explicit authorization on every request

04 FEDERATION & SINGLE SIGN-ON

Log in once, access everything — trusting identities across systems

DEFINITION

Federation lets one trusted system vouch for your identity so other systems admit you without re-entering credentials.

KEY CONCEPTS

— **SSO** — one login, access to many apps (Google, Slack, Salesforce)
— **Identity federation** — two organisations trust each other's IdP
— **SAML 2.0** — XML-based protocol for enterprise SSO
— **OIDC** — modern JSON/token-based SSO (social login, mobile)
— **Trust relationship** — a formal agreement to accept another party's assertions
— **Assertion** — the signed message stating 'I verified this user as X'

REAL-WORLD EXAMPLE

You log into your laptop once in the morning. Gmail, Slack, GitHub and Salesforce then recognise you automatically, with no extra logins. That is SSO federating your identity across apps via SAML or OIDC.

WHY IT MATTERS

— Enterprises run 100+ SaaS apps — SSO makes that manageable and secure
— Removes password sprawl: fewer passwords, fewer breach points
— Central deprovisioning — disable in the IdP, access is revoked everywhere
— Partner and supply-chain access (B2B federation) is a core enterprise use case

05 TOKENS & SESSIONS

Your digital pass — temporary proof of authentication

DEFINITION

After login, a token or session acts as a temporary pass carried with each request, so systems need not re-authenticate you every second.

KEY CONCEPTS

— **Session** — the server remembers you after login (traditional web apps)
— **JWT** — a self-contained signed token carrying user claims
— **Access token** — short-lived pass to call APIs (minutes to hours)
— **Refresh token** — long-lived token used to obtain a new access token silently
— **Token expiry** — a TTL that limits the breach window
— **Cookie vs bearer** — session in a browser cookie vs token in an API header

REAL-WORLD EXAMPLE

A JWT works like a concert wristband. The gate checks your ticket (authentication) and gives you a wristband (token), which you show at the bar, the merch booth and the VIP area. When the show ends, the wristband expires.

WHY IT MATTERS

— APIs and microservices are stateless — tokens carry context between requests
— A stolen token is a breach; short expiry limits the damage window
— Always validate signature, expiry and issuer
— OAuth 2.0 flows (auth code, PKCE) all revolve around token exchange

06 PROVISIONING & LIFECYCLE

From hire to retire — managing identities through their full journey

DEFINITION

Provisioning is creating, updating and removing access as people join, move within, or leave the organisation.

KEY CONCEPTS

— **Joiner** — a new hire's accounts and access are created automatically
— **Mover** — a role change updates access: old removed, new granted
— **Leaver** — on exit, all accounts are disabled immediately
— **SCIM** — the protocol that syncs user data between IdP and apps
— **Orphaned accounts** — accounts left active after a user leaves
— **Access review** — a periodic check that access is still warranted

REAL-WORLD EXAMPLE

A developer joins. HR approves in Workday, SCIM creates their Okta account, and Okta pushes them to GitHub, Jira and AWS. When they resign, one click in HR disables everything in seconds — no tickets, no forgotten access.

WHY IT MATTERS

— Orphaned accounts from departed staff are a top breach vector
— Manual provisioning causes errors and delays; automation is security and efficiency
— Regulators audit the account lifecycle end to end
— IGA tools such as SailPoint manage this at enterprise scale

07 DIRECTORY BASICS (LDAP)

The phone book of IT — the store of all identities and groups

DEFINITION

A directory is a structured database of identities organised for fast lookup: who exists, which groups they belong to, and what attributes they hold.

KEY CONCEPTS

— **LDAP** — the protocol used to query directories
— **Active Directory** — Microsoft's directory, LDAP plus Kerberos
— **OU** — organisational unit, the folders of AD: OU=Engineering, OU=London
— **DN** — distinguished name: CN=John,OU=Dev,DC=company,DC=com
— **Group** — a collection of users sharing permissions or policies
— **Schema** — the template defining which attributes identities may hold

REAL-WORLD EXAMPLE

At the Windows login screen your PC queries AD over LDAP: does this user exist, is the password correct, which groups are they in? AD answers in milliseconds, and every door you can open on the network follows from that answer.

WHY IT MATTERS

— AD is installed in over 90% of Fortune 500 companies
— AD compromise (for example a Golden Ticket attack) means total network takeover
— LDAP integration connects applications to enterprise identity stores
— Cloud directories (Entra ID, Google Workspace) are the modern evolution

08 THE END-TO-END FLOW

One identity, seven pillars, from first day to last

01	**Account creation** A new employee joins. HR triggers auto-provisioning over SCIM. An identity record is created in the IdP with name, email, role and department attributes. **Pillar 06 — Provisioning**
02	**Authentication** The user opens the company portal, enters their password and approves an MFA push. The IdP verifies the credentials and confirms identity. **Pillar 02 — Authentication**
03	**Token issued** On success the IdP issues a signed JWT access token and a refresh token. The user holds a digital pass valid for the next 60 minutes. **Pillar 05 — Tokens & Sessions**
04	**SSO across apps** The user opens Slack, then Salesforce. Via SAML/OIDC federation each app receives a signed assertion from the IdP. No re-login required. **Pillar 04 — Federation & SSO**
05	**Authorization** Each app checks the token claims and the user's role. As a developer they get code repository access but not billing dashboards. RBAC enforces least privilege. **Pillar 03 — Authorization**
06	**Directory lookup** Behind the scenes apps query Active Directory over LDAP to confirm group membership and refresh attributes — is this user still in the Engineering OU? **Pillar 07 — Directory (LDAP)**
07	**Lifecycle update** A team change updates the role and SCIM pushes it everywhere. On exit the account is disabled and access revoked instantly. Access reviews catch stale permissions. **Pillar 06 — Provisioning**

09 WHY IAM SKILLS MATTER NOW

From back-office function to front-line security discipline

CLOUD-FIRST WORLD

AWS, Azure and GCP are permission-driven platforms. IAM misconfiguration is the leading cause of cloud data breaches, and every cloud role now requires IAM literacy.

IDENTITY IS THE PERIMETER

Firewalls cannot protect SaaS applications. Identity is the security boundary, and Zero Trust architecture is built entirely on IAM principles.

SUSTAINED JOB DEMAND

IAM Engineer, IAM Architect and IGA Analyst are among the fastest-growing cybersecurity roles, often paying 30–50% above general IT positions.

DEVOPS AND DEVSECOPS

Service accounts, CI/CD pipelines and Kubernetes workloads all need machine identities. Platform engineers must understand least privilege.

COMPLIANCE AND AUDIT

SOC 2, ISO 27001, HIPAA, GDPR and PCI-DSS all carry IAM requirements, which makes these skills valuable in GRC and audit roles too.

WHERE TO GO NEXT

Pick one IdP and one cloud, build a joiner-mover-leaver flow end to end, then practise reading policies and token claims until they are second nature.