IAM, RBAC & Permission Models
Interview-Grade Explanation

1. Identity and Access Management (IAM)

IAM is the **overall security framework** responsible for managing digital identities and controlling access to resources.

IAM defines:
- Who the identity is (user, service, application)
- How the identity authenticates (passwords, MFA, certificates, tokens)
- How access decisions are enforced

IAM does not decide permissions randomly. It relies on structured **permission models** such as RBAC.

2. Role-Based Access Control (RBAC)

RBAC is an **authorization model** used within IAM.

RBAC logic:
- Permissions are grouped into roles
- Roles are assigned to identities
- Identities inherit permissions from their roles

Example:
Role: SOC_Analyst → Permissions: Read Logs, Investigate Alerts

3. Roles and Permissions

A **role** is a logical collection of permissions.
A **permission** defines an allowed action on a resource.

Example:
Permission: s3:GetObject
Resource: Incident-Logs Bucket
Effect: Allow

Roles make access scalable and consistent.

4. IAM vs RBAC (Clear Comparison)

Aspect	IAM	RBAC
Scope	Full identity & access framework	Authorization logic only
Handles Authentication	Yes	No
Handles Authorization	Yes	Yes
Manages Identities	Yes	No
Uses Roles	May use RBAC or others	Core concept

5. Permission Models

A **permission model** is the rule system that defines how permissions are granted, evaluated, and enforced.

RBAC is a permission model.

Other examples include:

- ABAC (Attribute-Based Access Control)
- PBAC (Policy-Based Access Control)

6. Why Permission Models Exist

Without a permission model:

- Permissions would be random
- Access would not scale
- Misconfigurations would increase
- Auditing would be nearly impossible

With a model:

- Access is predictable
- Security is enforceable
- Audits are straightforward
- Least privilege is achievable

7. Interview-Grade Summary

IAM is the overall framework for identity and access management.
RBAC is an authorization model within IAM that groups permissions into roles.
IAM authenticates identities and enforces access based on the roles and permission models attached to them.