Authentication and Authorization are two key concepts in securing APIs and systems. They are closely related but serve different purposes.

**1. What is Authentication?**

* **Definition:** Authentication is the process of verifying the identity of a user or client attempting to access a system.
* **Goal:** Ensure the user or client is who they claim to be.
* **Examples:**
  + A user logs in with a username and password.
  + An API client provides a valid token to access an endpoint.

**Authentication Methods:**

* **Username and Password:** The most common form of authentication.
* **Token-based Authentication:**
  + **JWT (JSON Web Token):** A stateless and secure way to authenticate users.
  + OAuth 2.0: Used for third-party authentication (e.g., Google, Facebook).
* **API Keys:** A simple token to identify and authenticate API clients.
* **Multi-Factor Authentication (MFA):** Adds an extra layer of security (e.g., SMS, email codes, or biometrics).

**2. What is Authorization?**

* **Definition:** Authorization is the process of determining whether a user or client has permission to perform a specific action or access a particular resource.
* **Goal:** Control what the authenticated user can do.
* **Examples:**
  + A regular user cannot access admin-only endpoints.
  + A user can read their profile but cannot edit another user’s profile.

**Authorization Mechanisms:**

* **Role-Based Access Control (RBAC):** Users are assigned roles (e.g., Admin, User) that determine their permissions.
* **Permission-Based Access Control:** Permissions are assigned directly to users or roles.
* **Claims-Based Authorization:** Permissions are based on claims in a token (e.g., JWT claims like role: Admin).

**3. Differences Between Authentication and Authorization**

| **Aspect** | **Authentication** | **Authorization** |
| --- | --- | --- |
| **What it answers?** | *Who are you?* | *What are you allowed to do?* |
| **Purpose** | Verifies the user’s identity. | Determines the user’s permissions. |
| **Process** | The user provides credentials (e.g., username/password). | The system checks roles or claims to allow/deny actions. |
| **Scope** | Happens first. | Happens after successful authentication. |
| **Examples** | Logging in with a password, validating an API token. | Checking if a user has access to a specific API or resource. |

**Example in Context**

**Scenario: A Commerce API**

1. **Authentication:**
   * A user logs in with a username and password or provides an API token.
   * The system verifies the token or credentials to confirm their identity.
2. **Authorization:**
   * After authentication:
     + A **regular user** can view their orders (GET /api/orders).
     + An **admin** can delete any order (DELETE /api/orders/{id}).