# Broken Access Control

## **Broken Access Control** occurs when an application does not properly enforce user permissions and allows unauthorized users to access, modify, or delete data or functionalities that should be restricted. Example:

An application allows users to change the user\_id parameter in the URL, granting access to another user's account details.

**Vertical Privilege Escalation**: A regular user accesses an admin-only page:  
/admin/deleteUser?userId=123  
**Horizontal Privilege Escalation**  
User A accesses User B’s data: /profile?userId=2002  
**IDOR (Insecure Direct Object Reference):** Accessing resources via predictable IDs (e.g., invoices, medical records).  
**Forced Browsing**: Accessing hidden functions/pages by manually typing URLs.  
**Method Tampering**: Changing HTTP method from GET to PUT to gain write access.

## Impact:

An attacker can view, modify, or delete another user's data, leading to data breaches and potential legal consequences.

## Mitigation:

* Enforce server-side access control checks, ensuring users can only access resources they own or are authorized to access.
* Implement least privilege, denying access by default.
* Use Access Control Lists (ACLs) and role-based access control (RBAC).

# Types of Broken Access Control

## 1. Vertical Privilege Escalation

* **Definition**: Lower-privileged users perform actions of higher-privileged users.
* **Example**: Normal user accessing admin dashboard via URL tampering.
* **Mitigation**:
  + Enforce role checks server-side.
  + Never rely on hidden fields or client-side roles.

## 2. Horizontal Privilege Escalation

* **Definition**: Users access data of other users with similar privileges.
* **Example**: Changing account ID in the URL: /user/view?id=102
* **Mitigation**:
  + Validate resource ownership server-side.
  + Use session-based identifiers, not user-controlled input.

## 3. Insecure Direct Object Reference (IDOR)

* **Definition**: Accessing objects (files, records) using unvalidated user input.
* **Example**: Downloading another user’s file by modifying file ID.
* **Mitigation**:
  + Use access control checks on every request.
  + Avoid using predictable IDs in URLs.

## 4. Missing Function-Level Access Control

* **Definition**: Backend APIs or URLs lack proper access restrictions.
* **Example**: A user manually accesses /admin/createUser
* **Mitigation**:
  + Implement access control on all functions and endpoints.
  + Use RBAC (Role-Based Access Control) at backend.

## 5. Forceful Browsing / Hidden URLs

* **Definition**: Accessing admin or hidden features via manual URL discovery.
* **Example**: /internal/settings
* **Mitigation**:
  + Do not rely on obscurity.
  + Authenticate and authorize every request.

## 6. Method Tampering

* **Definition**: Changing HTTP method to perform unauthorized actions.
* **Example**: Sending a PUT instead of a GET to update data.
* **Mitigation**:
  + Allow only expected HTTP methods.
  + Validate access per method + endpoint.

# Mitigation (General for All Types):

| Control | Description |
| --- | --- |
| Server-side access control | Never trust the client to enforce access roles. |
| RBAC or ABAC | Use Role-Based or Attribute-Based Access Control mechanisms. |
| Least Privilege Principle | Users should have the minimum access required. |
| Security Testing | Manual testing, automated scanners (Burp Suite, ZAP). |
| Logging and Alerting | Log access control failures and alert on anomalies. |
| Secure URL Schemes | Avoid sensitive data in URLs; use session-based mapping. |

# Tools to Detect Broken Access Control:

| Tool | Use |
| --- | --- |
| Burp Suite (Pro) | Manual testing, automation with Intruder |
| OWASP ZAP | Access control scanner plugin |
| Postman | API method tampering |
| AuthMatrix (Burp Plugin) | Role-based access control testing |
| Manual Testing | Most effective for business logic flaws |

# Common CVEs & Breach Examples:

| Case | Description |
| --- | --- |
| Facebook IDOR (2015) | Users could delete any photo album by ID reference. |
| Uber API Bug (2017) | IDOR allowed access to user trip history. |
| Instagram (2019) | Horizontal escalation to view private stories. |

# Points

Broken Access Control is consistently ranked **#1 in OWASP Top 10**, highlighting its severity and frequency."

Most access control flaws are **not detectable via automated scanners**; they require **manual logic testing**."

Ensure that **access control policy mapping and threat modeling** is part of the secure SDLC."