## ★ Insecure Direct Object Reference:

- Insecure direct object references (IDOR) are a type of access control vulnerability that arises when an application uses user-supplied input to access objects directly. The term IDOR was popularized by its appearance in the OWASP 2007 Top Ten. However, it is just one example of many access control implementation mistakes that can lead to access controls being circumvented. IDOR vulnerabilities are most commonly associated with horizontal privilege escalation, but they can also arise in relation to vertical privilege escalation.

### ★ Steps involved in execution of IDOR attack:

- Burp Suite Tool is widely used by attackers to execute such type of Attacks. Following are the steps being followed:
- Capture the Request: First of all, an attacker will decide a target website to which he wants to execute an IDOR attack. Then the website is added to the scope and spider the website to get all the URLs with specific parameters in it.
- Filter the parameters Request: After the first step, we will filter our captured request with the parameter filters. An attacker will only choose that parameter or Injection points where they can execute the attacks.
- Forward request to Repeater: Now, if an attacker will find some of the injection point where they can execute IDOR, they will forward the request to the repeater. The vulnerable URL might look something like this: www.xyz.com/myaccount/uid=19. Here the “UID” seems to be vulnerable.
- Tampering of Parameters: Now as the attacker has the vulnerable injection point, they will now try to execute the IDOR attack with the help of Social engineering or the pattern as written in injection point. Example: an attacker may change uid from 19 to 20 which will open account of another user who has been assigned id number 20.

### ★ Impacts of IDOR Vulnerability:

- Exposure of Confidential Information: When the attacker will have control over your account via this vulnerability, it is obvious that an attacker will be able to come across your personal information.
- Authentication Bypass: As the attacker can have access to millions of account with this vulnerability, it will be a type of Authentication bypass mechanism.
- Alteration of Data: An attacker may have privileges to access your data and alter it. By this, an attacker may have permission to make changes to your data, which

may lead to manipulation of records.

- Account Takeover: While an attacker may have multiple access to user accounts just by changing the “UID” values, this will lead to account takeover vulnerability. When one vulnerability leads to another vulnerability(like in this case), It is known as Chaining of BUGS.

## ★ IDOR examples

- There are many examples of access control vulnerabilities where user-controlled parameter values are used to access resources or functions directly.
- 1.IDOR vulnerability with direct reference to database objects
- Consider a website that uses the following URL to access the customer account page, by retrieving information from the back-end database:
- https://insecure-website.com/customer_account?customer_number=132355
- Here, the customer number is used directly as a record index in queries that are performed on the back-end database. If no other controls are in place, an attacker can simply modify the customer_number value, bypassing access controls to view the records of other customers. This is an example of an IDOR vulnerability leading to horizontal privilege escalation.
- An attacker might be able to perform horizontal and vertical privilege escalation by altering the user to one with additional privileges while bypassing access controls. Other possibilities include exploiting password leakage or modifying parameters once the attacker has landed in the user's accounts page, for example.
- 2. IDOR vulnerability with direct reference to static files
- IDOR vulnerabilities often arise when sensitive resources are located in static files on the server-side filesystem. For example, a website might save chat message transcripts to disk using an incrementing filename, and allow users to retrieve these by visiting a URL like the following:
- https://insecure-website.com/static/12144.txt
- In this situation, an attacker can simply modify the filename to retrieve a transcript created by another user and potentially obtain user credentials and other sensitive data.

## ★ Mitigation of IDOR:

- Remediation of IDOR Vulnerability:
- Developers should avoid displaying private object references such as keys or file names.
- Validation of Parameters should be properly implemented.
- Verification of all the Referenced objects should be done.
- Tokens should be generated in such a way that it should only be mapped to the user and should not be public