A4:

Insecure Direct Object References

It is about how we find objects in a database.

DOR: it occurs when the developer exposes the reference to the internal implementation object, such as file, directory, database key etc.

Without an access control check or other protection, attackers can manipulate these references to access the unauthorized data.

**Object:**

Data, information, Files , images, videos, resources, directory, database key etc can be stored and retrieved in either an unprocessed or processed format.

What you have to hide.( the files, or something which you want the people not to access)

**Reference:**

Resource Where you hide/ locate the object

Example:

Consider a table which has all the details of the bank accounts

Here the data i.e details of the bank accounts is the object and the place i.e the table which the data is stored is reference

Reference can also be url, path of the folder

**Insecure:**

Is it safe?

Is the user authenticated or Authorized to access the objects

Authentication: who you are (whether the user is valid)

Authorization: what you are allowed to do(whether the user have the required permissions)

An Authenticated user will not have access to certain objects whereas authorized user has the all the permissions to access objects

As we don’t want all the objects to be viewed for the application we need to secure the objects to prevent them from attacks.

If the objects are not secure, the attacker may retrieve the objects or use the objects to execute the miscellaneous code

**Insecure Direct Object Reference:**

Consider a web application which has the login

The user usually need not be authenticated to access the objects in the web app

Objects can be accessed by manipulating the direct reference to it, which is common when the web application relays on the user input to retrieve an object

If the web application is said to retrieve an object it should be secured and it should verify if the user is authorized to access objects

An attacker can manipulate direct object references to access other objects without authorization, unless an access control check is in place.

Writing the test cases :

* Think the different types of users who use the web application and their authorization/authentication
* Map out all the locations in the application where the user input is used to reference objects directly like locations where user inputs are used to access the database row, a file, application pages etc.
* Now modify the values in the parameters used to reference objects and assess whether it is possible to retrieve objects belonging to other users or otherwise bypass authorization.
* It is good to have different user accounts with different authorizations levels to test IDOR

Examples:

**The value of a parameter is used directly to perform an operation in the system** <http://foo.bar/changepassword?user=someuser>

**The value of a parameter is used directly to retrieve a database record:**

<http://foo.bar/somepage?invoice=12345>

**The value of a parameter is used directly to retrieve a file system resource**

[http://foo.bar/showImage?img=img00011 (path](http://foo.bar/showImage?img=img00011%20(path) traversal)

**The value of a parameter is used directly to access application functionality**

http://foo.bar/accessPage?menuitem=12

Best ways to avoid this?

Code analysis / code review

Automation tools can’t detect this better than the manual ways

Preventing methods:

Use an indirect reference to call the objects in database

Every user to have unique or random generated reference for the objects

Check points every time for the user if the user is authorized

Examples:

https://www.websitename/acct\_detail?acct=2121

Test cases:

Verify if the user is able to view or access the application bypassing login

Verify if the user is able to access the files by giving the reference path in the url / request of the application

Verify if the user is able to do unauthorized transactions in the application

Verify if the user is able to view the data which is restricted

Verify if the response from the burp has any info of username and password and the user can modify them

Example site:

<https://www.ehacking.net/2011/09>

use the Google docks search for the sites which as the index of/root folder by

allintitle: ”index of/root” passwd

allintitle: ”index of/root”

can also search for

inurl: admin

inurl: cgi

filetype: pdf

“auth\_user\_file.txt”

Index of ftp + .mdb allinurl:/cgi-bin/+mailto

Consider a url

<https://mis-security.com/file.jsp?file=report>.txt

the attacker can modify the parameters as

[https://mis-security.com/file,jsp?file=\*\*../../../etc/shadow\*\*](https://mis-security.com/file,jsp?file=**../../../etc/shadow**)

example:

<https://tic.com/customers/view/2145882>

tic.com/customers/viewdetails.aspx?ID=2145555445

Change to

Tic.com/customers/update/2145882

tic.com/customers/modify.aspx?ID=2145555445

tic.com/customers/admin

prevention?

validate user input

check the access rights

use the reference map rather than real values