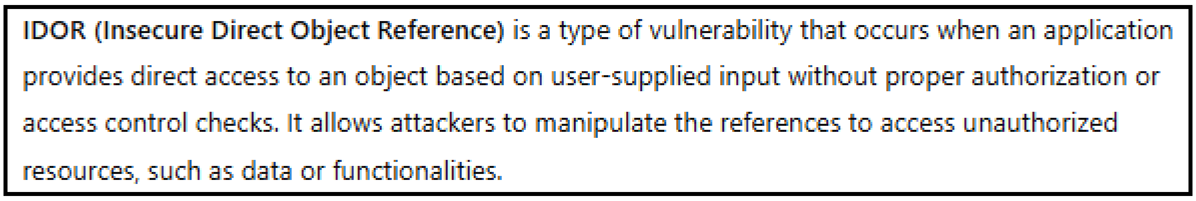
21 - IDOR - Insecure Direct Object Reference

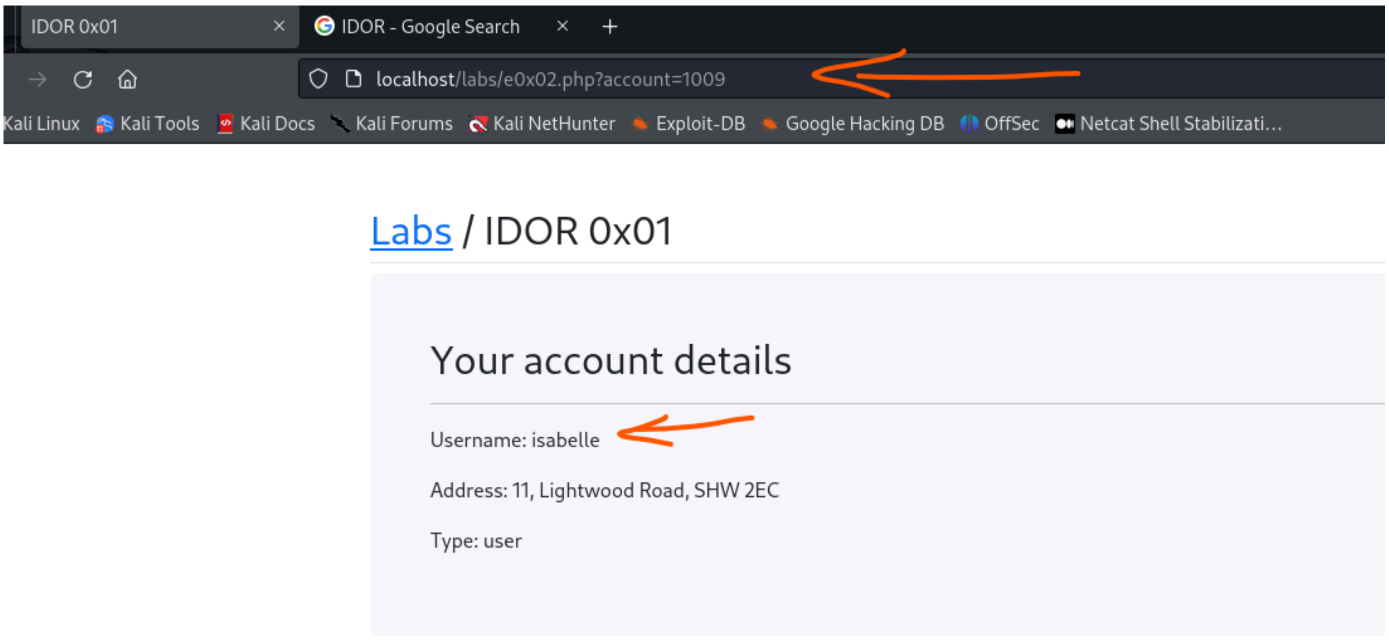
IDOR - Insecure Direct Object Reference.



IDOR (Insecure Direct Object Reference) is a type of vulnerability that occurs when an application provides direct access to an object based on user-supplied input without proper authorization or access control checks. It allows attackers to manipulate the references to access unauthorized resources, such as data or functionalities.

So, with just a little searching, we can understand what this is talking about and where we should be looking for here.

Look at the definition, and then look at the page we are trying to find vulnerabilities.



Labs / IDOR 0x01

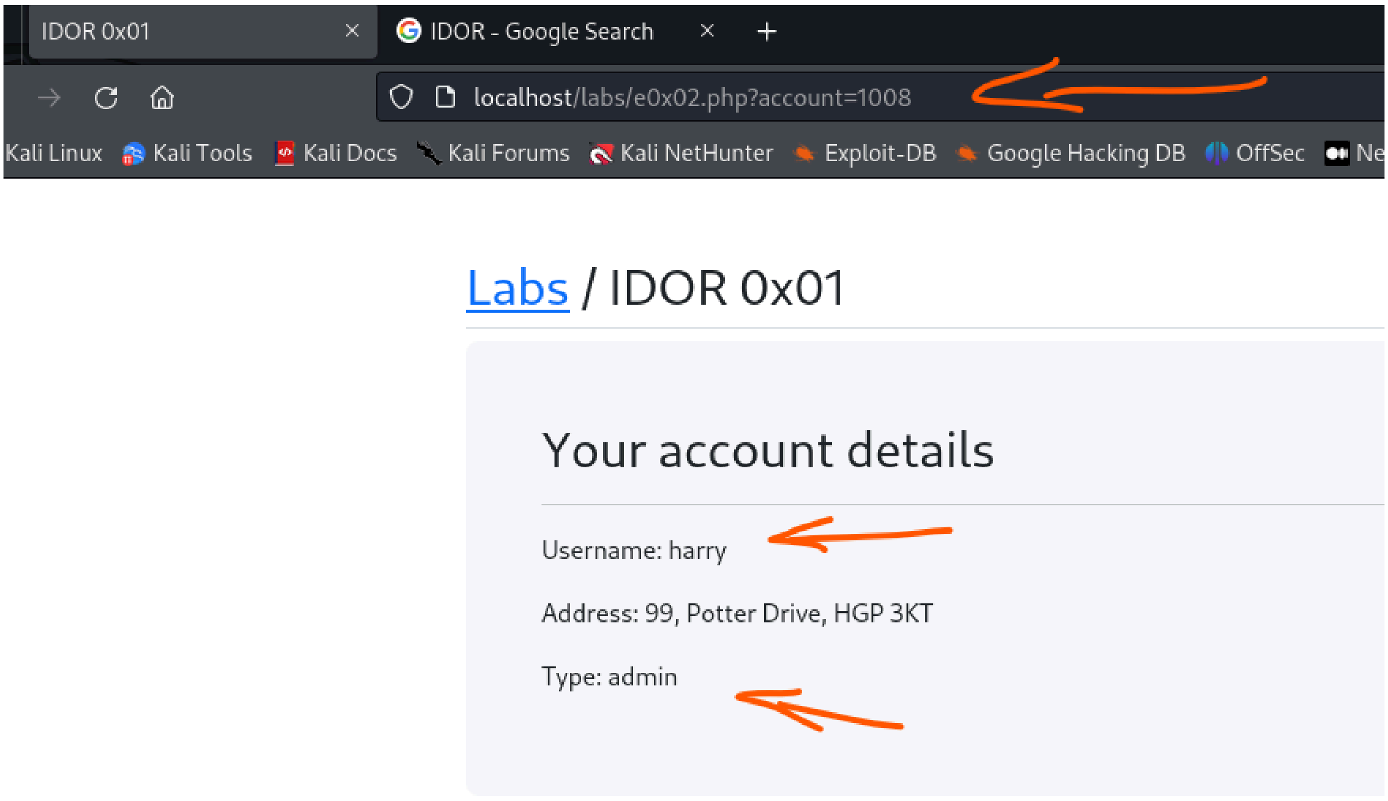
Your account details

Username: isabelle

Address: 11, Lightwood Road, SHW 2EC

Type: user

Well. First, it came through my mind to send the request to burp so we could mess around. But, this is very simple. Looks like the information displayed is attached to the account value passed in the URL through a GET request. We are able to replace "our" account value for others, and retrieve the information for these others valid account values. If we change to "?account=1008", we are able to retrieve a admin username, and their address.


IDOR 0x01 | IDOR - Google Search

localhost/labs/e0x02.php?account=1008

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec | Ne

Labs / IDOR 0x01

Your account details

Username: harry

Address: 99, Potter Drive, HGP 3KT

Type: admin

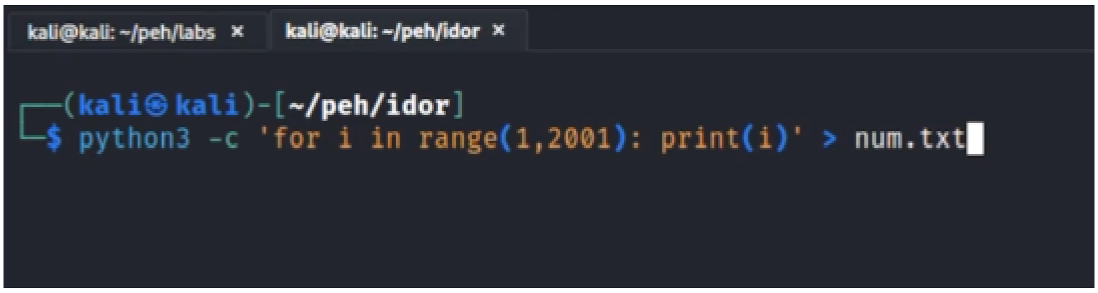
Not sure if this can lead to remote code execution. Lest check the video.

This happens a lot in API driven applications, and when it happens on these API driven applications, we actually call it BOLA, aka Broken Object Level Authorization.

Hum, our instructor is going to take to the next level.

He is going to show how would he go about enumerating all valid account in this website. And, if the ID was more complex, or more random, how he would arrange a way to FUZZ for those accounts.

For this particular scenario, he creates a word or in this case a number list using a python script


kali@kali: ~/peh/labs | kali@kali: ~/peh/idor

```
┌──(kali㉿kali)-[~/peh/idor]
└─$ python3 -c 'for i in range(1,2001): print(i)' > num.txt
```

This will make a word list from the number one to the number two thousand.

We run the word list using ffuf the way we already know.

The idea here is the same as before. Find a request that we want to filter out (in this case, "account not found" generated responses). We can achieve this by filtering out the response size of a "no account found" request.

After we find valid accounts, we can enumerate further to see find all admin account, or to find all the users that lived in a specific region, etc. It is a very good idea to automate this process, specially when there are hundreds of accounts, if not thousands.

Ideas from Alex. We can do this with Intruder, by filtering the response by searching for a specific term/string, or we could also write our own script to do that.