What is an IDOR?

In this room, you're going to learn what an IDOR vulnerability is, what they look like, how to find them and a practical task exploiting a real case scenario.

**What is an IDOR?**

IDOR stands for Insecure Direct Object Reference and is a type of access control vulnerability.

This type of vulnerability can occur when a web server receives user-supplied input to retrieve objects (files, data, documents), too much trust has been placed on the input data, and it is not validated on the server-side to confirm the requested object belongs to the user requesting it.

Answer the questions below

What does IDOR stand for?

Principio del formulario

An IDOR Example

View Site

Imagine you've just signed up for an online service, and you want to change your profile information. The link you click on goes to http://online-service.thm/profile?user\_id=1305, and you can see your information.  
  
Curiosity gets the better of you, and you try changing the user\_id value to 1000 instead (http://online-service.thm/profile?user\_id=1000), and to your surprise, you can now see another user's information. You've now discovered an IDOR vulnerability! Ideally, there should be a check on the website to confirm that the user information belongs to the user logged requesting it.  
  
Using what you've learnt above, click on the View Site button and try and receive a flag by discovering and exploiting an IDOR vulnerability.

Answer the questions below

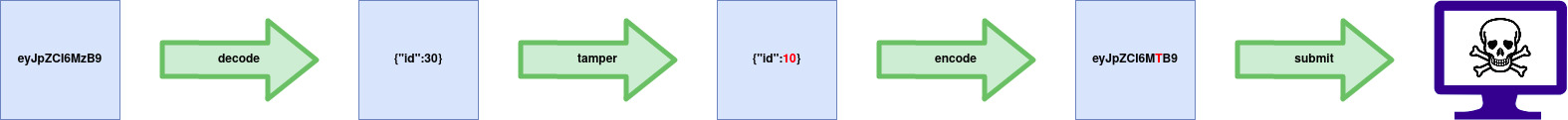
What is the Flag from the IDOR example website?

Principio del formulario

Finding IDORs in Encoded IDs

**Encoded IDs**

When passing data from page to page either by post data, query strings, or cookies, web developers will often first take the raw data and encode it. Encoding ensures that the receiving web server will be able to understand the contents. Encoding changes binary data into an ASCII string commonly using the a-z, A-Z, 0-9 and = character for padding. The most common encoding technique on the web is base64 encoding and can usually be pretty easy to spot. You can use websites like <https://www.base64decode.org/> to decode the string, then edit the data and re-encode it again using <https://www.base64encode.org/> and then resubmit the web request to see if there is a change in the response.  
  
See the image below as a graphical example of this process:



Answer the questions below

What is a common type of encoding used by websites?

Finding IDORs in Hashed IDs

**Hashed IDs**

Hashed IDs are a little bit more complicated to deal with than encoded ones, but they may follow a predictable pattern, such as being the hashed version of the integer value. For example, the Id number 123 would become 202cb962ac59075b964b07152d234b70 if md5 hashing were in use.

It's worthwhile putting any discovered hashes through a web service such as <https://crackstation.net/> (which has a database of billions of hash to value results) to see if we can find any matches.

Answer the questions below

What is a common algorithm used for hashing IDs?

Principio del formulario

Finding IDORs in Unpredictable IDs

**Unpredictable IDs**

If the Id cannot be detected using the above methods, an excellent method of IDOR detection is to create two accounts and swap the Id numbers between them. If you can view the other users' content using their Id number while still being logged in with a different account (or not logged in at all), you've found a valid IDOR vulnerability.

Answer the questions below

What is the minimum number of accounts you need to create to check for IDORs between accounts?

Where are IDORs located

**Where are they located?**

The vulnerable endpoint you're targeting may not always be something you see in the address bar. It could be content your browser loads in via an AJAX request or something that you find referenced in a JavaScript file.

Sometimes endpoints could have an unreferenced parameter that may have been of some use during development and got pushed to production. For example, you may notice a call to **/user/details** displaying your user information (authenticated through your session). But through an attack known as parameter mining, you discover a parameter called **user\_id** that you can use to display other users' information, for example,**/user/details?user\_id=123**.

Answer the questions below

Read the above.

Principio del formulario

A Practical IDOR Example

Start Machine

Begin by pressing the **Start Machine** button; once started, click the below link and open it in a new browser tab:

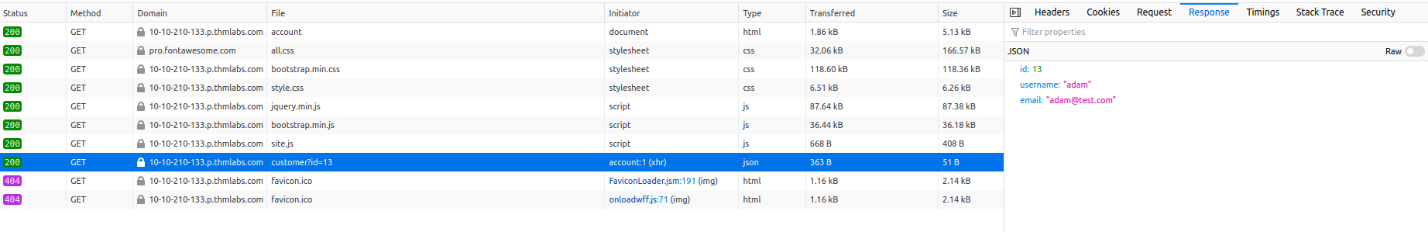
[https://LAB\_WEB\_URL.p.thmlabs.com](https://lab_web_url.p.thmlabs.com/)

Firstly you'll need to log in. To do this, click on the customer's section and create an account. Once logged in, click on the **Your Account** tab.

The **Your Account** section gives you the ability to change your information such as username, email address and password. You'll notice the username and email fields pre-filled in with your information.

We'll start by investigating how this information gets pre-filled. If you open your browser developer tools, select the network tab and then refresh the page, you'll see a call to an endpoint with the path /api/v1/customer?id={user\_id}.

This page returns in JSON format your user id, username and email address. We can see from the path that the user information shown is taken from the query string's id parameter (see below image).



You can try testing this id parameter for an IDOR vulnerability by changing the id to another user's id. Try selecting users with IDs 1 and 3 and then answer the questions below.

Answer the questions below

What is the username for user id 1?

Principio del formulario

Correct Answer

Final del formulario

What is the email address for user id 3?

Principio del formulario

Correct Answer

Final del formulario

How likely are