020. Manual BAC and IDOR hunting

Manual BAC testing

Introduction

Let’s start by talking about the easiest form of looking for BAC and IDOR, the manual way. Throughout the next chapters, we are going to dive deeper into what to test and exactly how to test it. We are going to show you how to hunt but first let’s start with some baby steps. Manual hunting can be done by either copy-and pasting or simply editing URLs and parameters on a page. We are going to look at different IDOR labs as well to help you practice this valuable skill but we first need some theory.

Sessions and authentication

Let’s explain you a bit about how you log into a website. This might seem obvious, you enter your username and password and you log in right? But what is really happening in the background and how can we use this to test for BAC and IDOR? Well there are several ways of “authenticating” a user but I won’t go too much into detail here, a popular option is to set a JWT token or a session token. A token is in essence something representing you without being you (aka your username + password represent you and grant you a token that represents the fact you know those and are who you say you are) so that means that after logging in, we get granted some kind of token representing us. This is done in the headers (cookies are also headers), with every request you have to pass your token along to ensure you are who you say you are and the server can serve you the data or not based on that token alone.

Now how does this help us you might be wondering, well we can log in as one user who either has a specific right granted to them we are testing for (for example, a user who can edit invoices) or are on our access level but are just a different account trying to access an object (for example, if you are only able to edit your invoices and will try to edit someone else’s invoices). We can then take that token in the header and copy it, while pasting it in a request to a resource we don’t have access to.

This is one thing we have to explain, another way is where our first way of manual testing. Now that you know sessions, please realise 1 browser windows is usually 1 session meaning that if you want to log in as two users you might just have to log in as one user and open another browser for the second user.

URL copy method

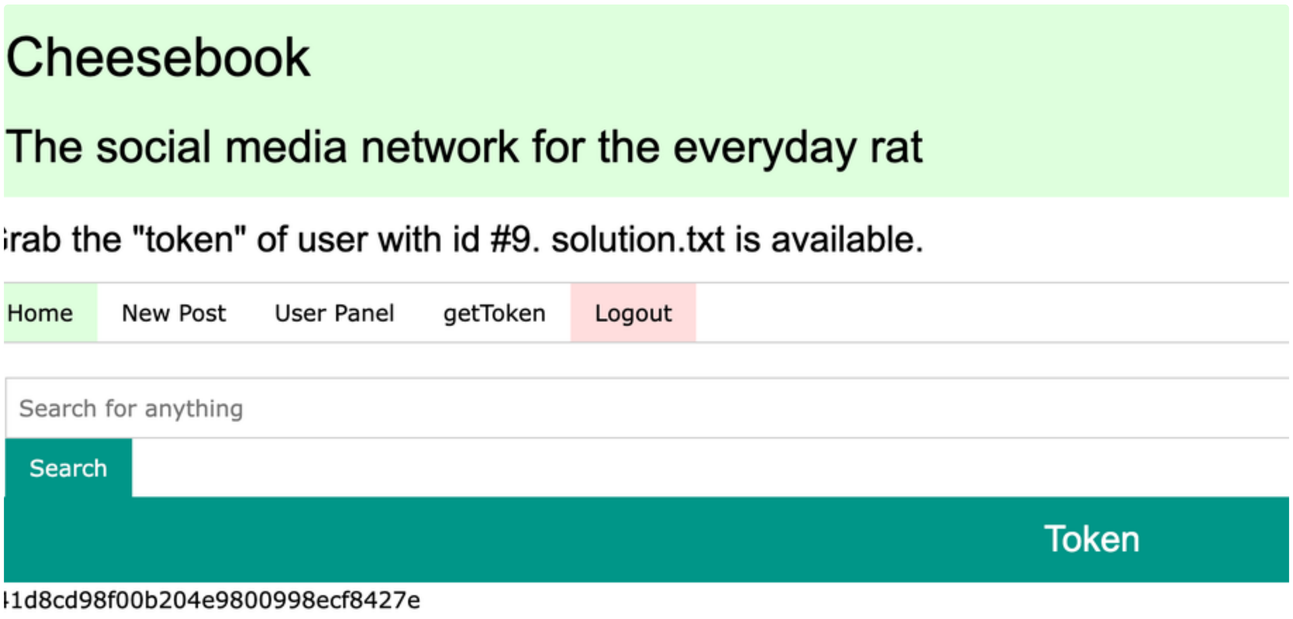
Now we can start with the URL copy method. This method is useful because it allows for a fast way of testing 1 call but it’s not recommended to test an application several times over. This will give us a basis though so let’s start with an exercise. Please go to CheeseBook - The social media network for the everyday rat and create an account. Please note the option “Get token” from the menu when you do log in

Now, create another user in a private window and log in as that second user. Copy the URL from your first user and paste it in the second users’ browser **(remember to make it an incognito session or otherwise the session is shared and you can not log in as 2 users)**

Now, try to copy and paste the URL from your first user in the browser of the second user. You will see that there is no problem in seeing the other person’s token.

Let’s investigate the URL: CheeseBook - The social media network for the everyday rat

We can clearly see the parameter “user” in here. This indicates our first user of which we are taking the token and it represents a GET-based IDOR.


Cheesebook

The social media network for the everyday rat

;rab the "token" of user with id #9. solution.txt is available.

¦1d8cd98f00b204e9800998ecf8427e

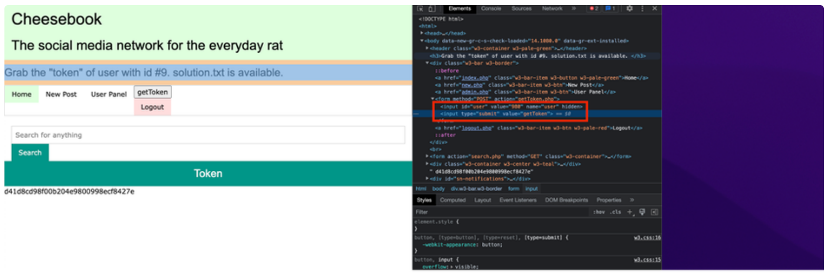
Get based because the parameter is in the URL, IDOR because we are directly referencing an object which is the “user” in this case and their property.

URL change method

Now it might be tempting to just change that user number from our previous test, but that is not recommended. After all, you could be testing a production system which contains data you are not supposed to see so always make sure you test with your own two accounts and that you do not change numbers randomly.

POST parameters

Now let’s grab the token on CheeseBook - The social media network for the everyday rat , here we can not clearly see the user id in the parameters. This makes our quest a little more difficult but we can still adjust the source code or simply change the parameter in burp suite before sending it. We do this by stopping the request, changing it’s value and then passing it onwards. Another way is to inspect element and change the values in the source code.


Cheesebook

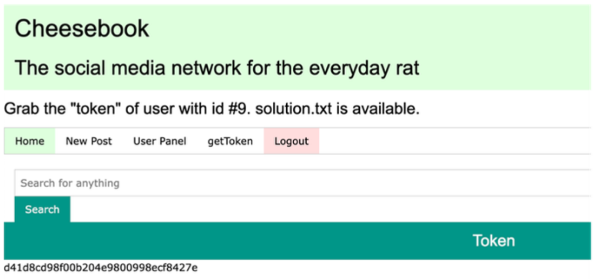
The social media network for the everyday rat

Grab the "token" of user with id #9. solution.txt is available.

d41d8cd98f00b204e9800998ecf8427e

```
<!DOCTYPE html>
<html>
▶<head>…</head>
▼<body data-new-gr-c-s-check-loaded="14.1080.0" data-gr-ext-installed>
  ▶<header class="w3-container w3-pale-green">…</header>
   <h3>Grab the "token" of user with id #9. solution.txt is available. </h3>
  ▼<div class="w3-bar w3-border">
     ::before
     <a href="index.php" class="w3-bar-item w3-button w3-pale-green">Home</a>
     <a href="new.php" class="w3-bar-item w3-btn">New Post</a>
     <a href="admin.php" class="w3-bar-item w3-btn">User Panel</a>
    ▼<form method="POST" action="getToken.php">
       <input id="user" value="900" name="user" hidden>
       <input type="submit" value="getToken"> == $0
     </form>
     <a href="logout.php" class="w3-bar-item w3-btn w3-pale-red">Logout</a>
     ::after
   </div>
   <br>
  ▶<form action="search.php" method="GET" class="w3-container">…</form>
  ▶<div class="w3-container w3-center w3-teal">…</div>
   " d41d8cd98f00b204e9800998ecf8427e"
  ▶<div id="sn-notifications">…</div>
```

```
element.style {
}
button, [type=button], [type=reset], [type=submit] {      w3.css:16
  -webkit-appearance: button;
}
button, input {                                            w3.css:15
  overflow: ▸ visible;
```


Cheesebook

The social media network for the everyday rat

Grab the "token" of user with id #9. solution.txt is available.

d41d8cd98f00b204e9800998ecf8427e

Practice

Now that you know both techniques, can you solve the following questions?

Grab the token of user #9:

CheeseBook - The social media network for the everyday rat

Tip: Source code

CheeseBook - The social media network for the everyday rat

Tip: Change ID in request itself

Let's kick it up a notch! Go to CheeseBlog and make an account. Also log in as the user "admin/test" and see that you can view all users in the admin panel. Copy the URL and now paste that URL into the browser of the user your are logged in as. You should not be able to edit posts, this is a clear violation and a BAC issue.