

Write-Up over the wire (bandit)

Ahmad Ryan Faizal

Level 1: NH2SXQwcBdpmTEzi3bvBHMM9H66vVXjL

Level 2: Diberikan file bernama - karena tidak bisa langsung "cat -" maka ditambahkan ./ didepannya = 'cat ./-'

```
bandit1@bandit:~$ ls
-
bandit1@bandit:~$ cat ./-
rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi
bandit1@bandit:~$
```

Maka didapatkan passwordnya yaitu: rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi

Level 3: Diberikan file dengan spasi karena kita tidak bisa cat langsung pakai file spasi jadi setiap spasi harus ditambahkan "\" untuk membukanya

```
bandit2@bandit:~$ cat spaces\ in\ this\ filename
aBZ0W5EmUfAf7kHTQeOwd8bauFJ2lAiG
bandit2@bandit:~$
```

Maka didapatkan passwordnya yaitu: aBZ0W5EmUfAf7kHTQeOwd8bauFJ2lAiG

Level 4: Diberikan directory yaitu 'inhere' langsung saja saya cd ke directory yang tersedia pada saat di ls tidak ada muncul tapi setelah saya memakai command yang lain yaitu 'ls' lalu ada yang namanya file .hidden langsung saja saya cat filenya dan passwordnya langsung tersedia

```
bandit3@bandit:~/inhere$ cat .hidden
2EW7BBsr6aMMoJ2HjW067dm8EgX26xNe
bandit3@bandit:~/inhere$
```

Passwordnya yaitu: 2EW7BBsr6aMMoJ2HjW067dm8EgX26xNe

Level 5: Diberikan directory yang sama tapi filenya beda yaitu 'file 1-9' langsung saja saya cat semua filenya disampai pada file 7 maka ditemukan lah passwordnya:

```
bandit4@bandit:~/inhere$ cat ./-file05
♦♦!♦♦>E♦+♦♦♦♦♦♦♦♦♦♦♦♦K♦bg
♦♦♦♦♦
♦♦I=4bandit4@bandit:~/inhere$ cat ./-file06
^♦f♦♦♦♦s♦_♦♦c♦$!C♦♦j♦?jR ♦Mtbandit4@bandit:~/inhere$ cat ./-file07
lrIWWI6bB37kxfiCQZqUdOIYfr6eEeqR
bandit4@bandit:~/inhere$
```

Passwordnya yaitu: lrIWWI6bB37kxfiCQZqUdOIYfr6eEeqR

Level 6: Diberikan directory yang sama lagi didalamnya diberi directory lagi dan dalam directorynya ada yang file lebih banyak lagi. Karena tidak mungkin cek satu-satu maka saya menggunakan fungsi find dan untung saja di web juga dikasih info tentang file yang mengandung password yaitu:

- 1033 bytes in size
- not executable
- human-readable

Maka langsung saja saya menggunakan 'find -size 1033c' dan setelah melakukan command tersebut maka tertampil lah file yang sesuai yaitu seperti di gambar berikut dan langsung saja saya cat untuk menampilkan passwordnya.

```
bandit5@bandit:~$ cd inhere
bandit5@bandit:~/inhere$ ls
maybehere00 maybehere03 maybehere06 maybehere09 maybehere12 maybehere15 maybehere18
maybehere01 maybehere04 maybehere07 maybehere10 maybehere13 maybehere16 maybehere19
maybehere02 maybehere05 maybehere08 maybehere11 maybehere14 maybehere17
bandit5@bandit:~/inhere$ find -size 1033c
./maybehere07/.file2
bandit5@bandit:~/inhere$ cd maybehere07
bandit5@bandit:~/inhere/maybehere07$ cat .file2
P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU
```

Passwordnya yaitu: P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU

Level 7: Disoal dijelaskan kalau password berada di root directory maka langsung saja saya ganti directory ke root dengan 'cd /' langsung saja saya mencari file menggunakan info yang diberi di soal yaitu:

- owned by user bandit7
- owned by group bandit6
- 33 bytes in size

dengan command find -user bandit7 -group bandit6 -size33c maka terpampanglah

```
find: './var/log/private': Permission denied
find: './var/log/unattended-upgrades': Permission denied
find: './var/spool/cron/crontabs': Permission denied
find: './var/spool/rsyslog': Permission denied
find: './var/spool/bandit24': Permission denied
find: './tmp': Permission denied
find: './boot/efi': Permission denied
find: './proc/tty/driver': Permission denied
find: './proc/1677351/task/1677351/fd/6': No such file or directory
find: './proc/1677351/task/1677351/fdinfo/6': No such file or directory
find: './proc/1677351/fd/5': No such file or directory
find: './proc/1677351/fdinfo/5': No such file or directory
find: './run/chrony': Permission denied
find: './run/udisks2': Permission denied
find: './run/user/11033': Permission denied
find: './run/user/11003': Permission denied
find: './run/user/11023': Permission denied
```

berbagai error dikarenakan permissio denied. Maka untuk mengatasinya ditambahkan 2>dev/null untuk mengatasi error atau bisa dibilang 'hide' error tersebut. Saya membaca fungsi ini di:

<https://askubuntu.com/questions/350208/what-does-2-dev-null-mean> dan setelah melakukan command yang ditambahkan fungsi tersebut maka akan ditampilkan letak file passwordnya langsung saja saya cat file tersebut:

```
bandit6@bandit:/$ find -user bandit7 -group bandit6 -size 33c 2>dev/null
./var/lib/dpkg/info/bandit7.password
bandit6@bandit:/$ cat var/lib/dpkg/info/ba
bandit7.password      base-files.prerm      bash-completion.conf
base-files.conf       base-passwd.list      bash-completion.list
base-files.list        base-passwd.md5sums    bash-completion.md5
base-files.md5sums     base-passwd.postinst   bash-completion.post
base-files.postinst    base-passwd.postrm     bash-completion.postr
base-files.postrm      base-passwd.preinst    bash-completion.prei
base-files.preinst     base-passwd.templates  bash.conf
bandit6@bandit:/$ cat var/lib/dpkg/info/bandit7.password
z7WtoNQU2XfjmMtWA8u5rN4vzqu4v99S
bandit6@bandit:/$
```

Passwordnya adalah: z7WtoNQU2XfjmMtWA8u5rN4vzqu4v99S

Level 8: diberikan file 'data.txt' seperti di soal password ada di sebelah kata **millionth** maka langsung saja saya menggunakan command cat dan juga grep 'cat data.txt | grep millionth' maka didapatkanlah passwordnya persis disebelah kata **millionth**.

Passwordnya adalah: TESKZC0XvTetK0S9xNwm25STk5iWrBvP

Level 9: diberikan file yang sama. Di soal dijelaskan kalau passwordnya ada didalam file tersebut dan password juga cuman muncul satu. Saya disini menggunakan fungsi sort maka muncullah banyak password-password lainnya. Langsung saja saya cek satu persatu mana yang muncul cuman satu kali dan akhirnya ketemu yaitu:

```
DIItvE0rpT0pRGL1bFdRhoQkwX8SdLMYV
EN632PlfYiZbn3PhVK3XOGSlnNE00t
Eor03gLDc3awKULF84XCnD8xgRg6X9S3
```

Passwordnya adalah: EN632PlfYiZbn3PhVK3XOGSlnNE00t

Level 10: diberikan file yang sama dan juga soal The password for the next level is stored in the file **data.txt** in one of the few human-readable strings, preceded by several '=' characters.

Lalu saya mencoba 'cat data.txt | grep =' tapi yang muncul malah error. Karena saya tidak punya clue tentang soal diatas maka saya mencoba semua recommended command yang ada di soal yaitu strings. Setelah saya mencoba command 'strings data.txt' keluarlah beberapa teks dan seperti sesuai yang ada di soal ada teks disebelah simbol = maka saya asumsikan itu adalah passwordnya

```
|K;8[#
G2@i
:26=
===== G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s
~`2U
GCF;
s?:S
$B2Q
#a1(
@d7
```

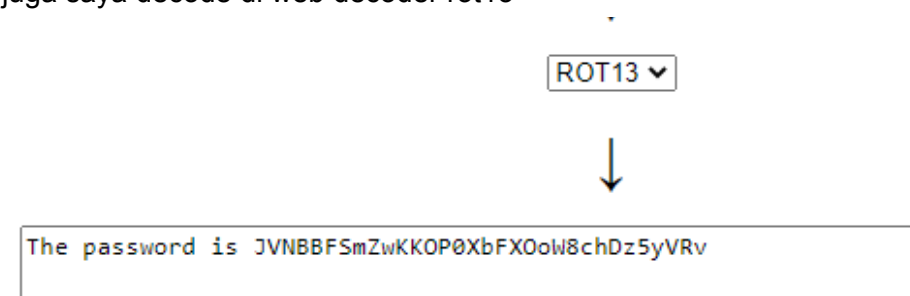
dan benar saja setelah mencoba masuk ke levelnya dengan password diatas ternyata bisa. Maka dari itu passwordnya adalah: G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s

Level 11: diberikan file yang sama lagi dan juga soal. Di soal dikatakan password ada didalam file dan diencode dengan base64 maka dari itu saya langsung saja mendecode file tersebut dengan command "cat data.txt | base64 -d"

```
bandit10@bandit:~$ ls
data.txt
bandit10@bandit:~$ cat data.txt | base64 -d
The password is 6zPezilDR2RKNdNYFNb6nVCKzphlXHBM
bandit10@bandit:~$
```

Passwordnya adalah: 6zPezilDR2RKNdNYFNb6nVCKzphlXHBM

Level 12: diberikan file yang sama lagi dan juga soal tetapi kali ini melainkan base64 sekarang bahasa yang dipakai adalah rot13. Maka langsung saya cat file tersebut dan juga saya decode di web decoder rot13



Passwordnya adalah: JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv

Level 13: Diberikan soal: The password for the next level is stored in the file **data.txt**, which is a hexdump of a file that has been repeatedly compressed. For this level it may be useful to create a directory under /tmp in which you can work using mkdir. For example: mkdir /tmp/myname123. Then copy the datafile using cp, and rename it using mv (read the manpages!)

Seperti yang dikatakan di soal password kali ini berada di dalam file yang sudah di compress berkali-kali. Awalnya disini saya bingung harus dan setelah saya mencoba melakukan beberapa riset dengan membaca manual atau 'man' dari command command yang diberikan saya mendapatkan caranya command yang disediakan di soalnya adalah:

Commands you may need to solve this level

grep, sort, uniq, gstrins, base64, tr, tar, gzip, bzip2, xxd, mkdir, cp, mv, file

Sesuai soal yaitu file ini sudah dicompress beberapa kali berarti kita hanya perlu "un-compress" file ini lagi dan disini nampaknya metode yang dicompress adalah 3 dari command diatas yaitu: tar, gzip, bzip2.

Untuk mengetahui file yang sedang dijalankan dalam bentuk apa saya menggunakan 'xxd [file] | head' untuk melihat byte pertama file tersebut

Untuk header byte pertama dari beberapa jenis file diatas adalah

- gzip : **1fb8**
- bzip2 : **425a**

Maka disini saya langsung saja uncompress file tersebut lagi karena file pertama yaitu 'data.txt' berformat gzip (1fb8) langsung saja saya uncompress file tersebut dengan dengan cara rename tapi sebelum itu saya membuat directory temp sesuai seperti soal

```
bandit12@bandit:~$ cd / tmp
-bash: cd/: No such file or directory
bandit12@bandit:~$ cd /tmp
bandit12@bandit:/tmp$ mkdir -p /tmp/tmp.sRjHde2zxD
bandit12@bandit:/tmp$ cd /tmp/tmp.sRjHde2zxD
bandit12@bandit:/tmp/tmp.sRjHde2zxD$ cp ~/data.txt .
bandit12@bandit:/tmp/tmp.sRjHde2zxD$ mv data.txt hexdump_data
bandit12@bandit:/tmp/tmp.sRjHde2zxD$
```

karena masih berbentuk hexdump disini saya mereverse hexdumpnya agar menjadi 'actual file' disini saya namakan compressed_data nah setelah itu kita bisa membenarkan ekstensi/akhir filenya dari yang 'compressed_data > compressed_data.gz' nah setelah itu saya melakukan decompress lagi dengan gzip dan cek sekarang file berbentuk apa.

```
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ mv compressed_data compressed_data.gz
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ gzip -d compressed_data.gz
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ xxd compressed_data | head
00000000: 425a 6839 3141 5926 5359 5ded 11a8 0000  BZh91AY&SY].....
00000010: 1bff ffd8 fffd e7df f7ff ffff cfef cfbe  ....
00000020: f77e 7fdd 393f 7ffa fbff bfcf bf3e ffa9  .~..9?.....>..
00000030: fbbf 7fb0 013b 1b6d 200f 5000 3406 8000  ....;.m .P.4 ...
00000040: 0034 c201 ea0d 3400 0019 001a 321a 680d  .4....4....2.h.
00000050: 0000 0000 3400 0000 0d00 6991 ea0c 6d51  ....4.....i ...mQ
00000060: 0000 6800 c800 0d03 2343 403d 400d 0d1a  ..h.....#C@=@ ...
00000070: 6801 a34c 8340 1a68 7a40 3403 401a 0034  h..L.@.hz@4.@..4
00000080: 6801 88c8 6834 d000 c8d0 1a68 74d3 2340  h...h4.....ht.#@
00000090: d3d2 0681 a1a6 80d0 c801 90d0 3403 400d  ....4.@.
bandit12@bandit:/tmp/tmp.sRJHde2zxD$
```

Nah disini kita tahu file yang sudah di decompress tadi sekarang berbentuk file yang lain lagi yaitu bzip2. Disini saya akan mengulangi step diatas lagi tapi sekarang dengan format bzip2. Nah setelah di decompress lagi file berbentuk gzip lagi maka kita ulangi lagi

```
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ mv compressed_data compressed_data.bz2
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ bzip2 -d compressed_data.bz2
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ xxd compressed_data
00000000: 1f8b 0808 7151 1063 0203 6461 7461 342e  ....qQ.c..data4.
```

Sekarang menggunakan gzip. Nah setelah di decompress kita menghasilkan file lagi.

```
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ mv compressed_data compressed_data.gz
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ gzip -d compressed_data.gz
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ xxd compressed_data | head
00000000: 6461 7461 352e 6269 6e00 0000 0000 0000  data5.bin.....
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  ....
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ....
00000030: 0000 0000 0000 0000 0000 0000 0000 0000  ....
00000040: 0000 0000 0000 0000 0000 0000 0000 0000  ....
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  ....
00000060: 0000 0000 3030 3030 3634 3400 3030 3030  ....0000644.0000
00000070: 3030 3000 3030 3030 3030 3000 3030 3030  000.0000000.0000
00000080: 3030 3234 3030 3000 3134 3330 3430 3530  0024000.14304050
00000090: 3536 3100 3031 3132 3336 0020 3000 0000  561.011236. 0 ...
```

Namun nampaknya kali ini file sudah berbentuk archive jadi kali ini kita bisa menggunakan **tar** untuk mengekstrak filenya. Langsung saja kita ulangi step diatas tapi kali ini dalam tar

```
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ mv compressed_data compressed_data.tar
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ tar -xf compressed_data.tar
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ ls
compressed_data.tar  data5.bin  hexdump_data
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ tar -xf data5.bin
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ ls
compressed_data.tar  data5.bin  data6.bin  hexdump_data
```

Setelah diextract muncul file 'data5.bin' setelah saya properties ternyata data5.bin masih berbentuk tar juga jadi saya extract lagi dan dihasilkan lah data6.bin tetapi data6.bin mempunyai jenis file yang berbeda yaitu **bzip2** jadi kita ulangi lagi tapi dengan **bzip2**.


```
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ xxd data6.bin | head
00000000: 425a 6839 3141 5926 5359 25d3 6424 0000  BZh91AY&SY%.d$ ..
00000010: 8c7f dfdc 5cc0 40c0 6fff e000 f1a3 807c  ....\.@.o.....|
00000020: 21fe 0000 0800 1002 0000 7282 0400 8442  !.....r....B
```

Nah setelah di decompress lagi sekarang file berbentuk **tar** lagi dan mengandung archive data8.bin didalamnya jadi kita extract lagi menggunakan **tar**

```
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ bzip2 -d data6.bin
bzip2: Can't guess original name for data6.bin -- using data6.bin.out
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ ls
compressed_data.tar  data5.bin  data6.bin.out  hexdump_data
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ xxd data6.bin.out | head
00000000: 6461 7461 382e 6269 6e00 0000 0000 0000  data8.bin.....
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000030: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000040: 0000 0000 0000 0000 0000 0000 0000 0000  .....
```

Ketika sudah selesai mengextract file tadi dengan tar maka muncullah data8.bin sesuai yang dijelaskan tadi dan disaat saya lihat lagi jenis file data8.bin ternyata file ini masih berbentuk gzip jadi saya decompress lagi seperti biasa.

```
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ tar -xf data6.bin.out
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ ls
compressed_data.tar  data5.bin  data6.bin.out  data8.bin  hexdump_data
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ xxd data8.bin | head
00000000: 1f8b 0808 7151 1063 0203 6461 7461 392e  ....qQ.c..data9.
00000010: 6269 6e00 0bc9 4855 2848 2c2e 2ecf 2f4a  bin...HU(H, .../J
00000020: 51c8 2c56 284f 0a4f c971 aa70 cd2c 3271  Q.,V(O.O.q.p.,2q
```

dan ya setelah didecompress lagi sekarang sudah keluar titik terang yaitu sudah ada bentuk dari actual file dari data yang telah banyak dicompress ini yaitu data8, jadi seperti biasa tinggal lihat saja isi file tersebut dengan cat maka didapatkan lah passwordnya

```
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ mv data8.bin data8.gz
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ gzip -d data8.gz
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ ls
compressed_data.tar  data5.bin  data6.bin.out  data8  hexdump_data
bandit12@bandit:/tmp/tmp.sRJHde2zxD$ cat data8
The password is wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw
bandit12@bandit:/tmp/tmp.sRJHde2zxD$
```

Passwordnya yaitu: wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw

Level 13: di level 13 kita disediakan sshkey. Di level ini cuman dijelaskan jika password level 14 cuman bisa diakses oleh user bandit14 oleh karena itu di level 13 kita disediakan key untuk login menggunakan ssh ke bandit14.

```
!!! You are trying to log into this SSH server on port 22, which is not intended.

bandit14@localhost: Permission denied (publickey).
bandit13@bandit:~$ ssh bandit14@localhost -i sshkey.private -p 2220
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLFXC5CXlhmAAM/urerLY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Could not create directory '/home/bandit13/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit13/.ssh/known_hosts).
```

```
|_|_ _ _ _ _|_(_)_|_
```

setelah berhasil masuk ke bandit14 langsung saja kita cat file yang berada di directory seperti yang diberitahu

```
bandit14@bandit:~$ ls
bandit14@bandit:~$ cat /etc/bandit_pass/bandit14
fGrHPx402xGC7U7rXKDaxiWFTOiF0ENq
```

Maka didapatkan password fGrHPx402xGC7U7rXKDaxiWFTOiF0ENq

Level 14: dijelaskan di soal password berikutnya bisa didapatkan dengan memasukkan password yang sebelumnya ke localhost di port 30000. Langsung saja saya tembak dengan netcat

```
bandit14@bandit:~$ echo "fGrHPx402xGC7U7rXKDaxiWFTOiF0ENq" | nc localhost 30000
Correct!
jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt
```

Maka didapatkan password ke level berikutnya yaitu: jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt

Level 15: dijelaskan di soal password berikutnya bisa didapatkan dengan memasukkan password yang sebelumnya ke localhost di port 30001 menggunakan SSL encryption.

Langsung saja saya connect menggunakan openssl dengan
"openssl s_client -connect localhost:30001"

dan diminta untuk memasukkan password setelah password yang sebelumnya dimasukkan

```
read R BLOCK
jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt
Correct!
JQtTfApK4SeyHwDlI9SXGR50qcl0Ail1
```

Maka tampilah password untuk level berikutnya
yaitu: JQtTfApK4SeyHwDlI9SXGR50qcl0Ail1

Level16: dijelaskan di soal password berikutnya bisa didapatkan dengan memasukkan password yang sebelumnya ke localhost pada port yang terbuka di antara 31000 - 32000

Langsung saja saya menggunakan nmap untuk melacak port yang terbuka dengan "nmap -r localhost -p 31000-32000"

```
bandit16@bandit:~$ man nmap
bandit16@bandit:~$ nmap -r localhost -p 31000-32000
Starting Nmap 7.80 ( https://nmap.org ) at 2022-09-16 02:49 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00018s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
31046/tcp  open  unknown
31518/tcp  open  unknown
31691/tcp  open  unknown
31790/tcp  open  unknown
31960/tcp  open  unknown
```

setelah itu saya mengulang seperti step di level sebelumnya dan mencoba portnya satu-satu sampailah pada port 31960 dan port ini lah yang benar.

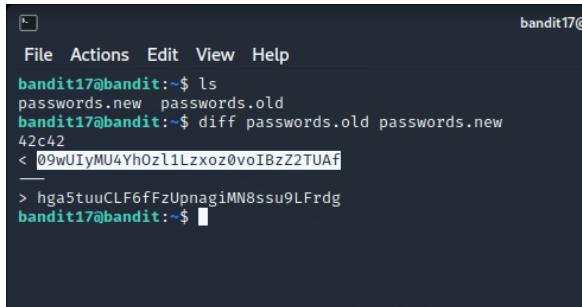
```
—
read R BLOCK
JQtTfApK4SeyHwDlI9SXGR50qc10Ai11
Correct!
—BEGIN RSA PRIVATE KEY—
MIEogIBAAKCAQEAvM0kuiFmMg6HL2YPIOjon6iWfbp7c3jx34YkYWqUH57SUdyJ
imZzeyGC0gtZPGUjUSxiJSWI/oTqexh+cAMTSMl0Jf7+BrJ0bArnx9Y7YT2bRPQ
Ja6Lzb558YW3FZl870RiO+rW4LDCdNd2lUvLE/GL2GWyuKN0K5iCd5TbtJzEkQTu
DSt2mcNn4rhAL+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW30ekePQAzL0VUYbW
JGTi65CxbCnzc/w4+mqQyvmzpWtMAZJTzAzQxNbK2MBGySxDLrjg0LWN6sK7wNX
x0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XF0JuaQIDAQABAoIBABagpxpM1aoLWfvD
KHcj10nqcoBc4oE11aFYQwik7xfW+24pRNUDE6SFth0ar69jp5RlLwD1NhPx3iBl
J9n0M80J0VToum43UOS8YxF8WwhXriYGnc1sskbwpXOUDc9uX4+UESzH22P29ovd
d8WErY0gPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9AlbssgTcCXkMQnPw9nC
YNN6DDP2lbcBrvgT9YCNL6C+ZKuFD52yOQ9q0kwFTEQpjTf4uNtJom+asvlpms8A
vLY9r60wYSvmZhNqBURj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama
+TOWWgECgYEA8JtPxP0GRJ+IQkX262jM3dEIkza8ky5moIwUqYdsx0NxHgRRhORT
8c8hAuRBb2G82so8vUHK/fur850Efc9TncnCY2crpoqsgghifKLxrLgtT+qDpfZnx
SatLdt8GfQ85yA7hnWWJ2Mx3F3NaeSDm75Lsm+tBbAiyC9P2jGRNtMSkCgYEAypHd
HCctNi/FwjuLhtFfx/rHYKhLidZDFYeIE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt
SghaTdcG0Knyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCIvGCSx+X3l5SiWg0A
R57hJglezIiVjv3aGwHwvLZvtszK6zV6oXFAu0ECgYABjo46T4hyP5tJi93V5Hdi
TtieK7xRVxUl+iU7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFmLy9FL2m9oQWCg
R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB30hYimtiG2Cg5JCqIZFHxD6MjEG0iu
L8ktHMPvodBwNsSBULpG0QKBgBaplTfC1H0nWiMGOU3KPwYwt006CdTkmJ0mL8Ni
blh9elyZ9FsGxsgtRBXRsqXuz7wtsQAgLHxbdLq/ZJQ7Yfz0KU4ZxEnabvXnvWkU
Y0djHdSOoKvDQNWu6ucyLRAWFuISeXw9a/9p7ftpxm0TSgyvmfLF2MIAEwyZrQaM
77pBAoGAMmjmIJdjp+Ez8duyn3ieo36yrTtF5NSsJLabxFpdlc1gvtGCWW+9Cq0b
dxviW8+TFVEBl104f7HVM6EpTscdXU+bCXWkfjuRb7Dy9G0tt9JPsx8MBTakh3
vBgysi/sN3RqRBcGU40f0oZyfAMT8s1m/uYv5206IgeuZ/ujbjY=
—END RSA PRIVATE KEY—
```

Setelah password dimasukkan kita akan diberi private key.

Level 17: Login menggunakan private key yang telah didapatkan sebelumnya. Cara untuk loginnya adalah

“sudo ssh -i [rsa key file] bandit17@bandit.labs.overthewire.org -p 2220

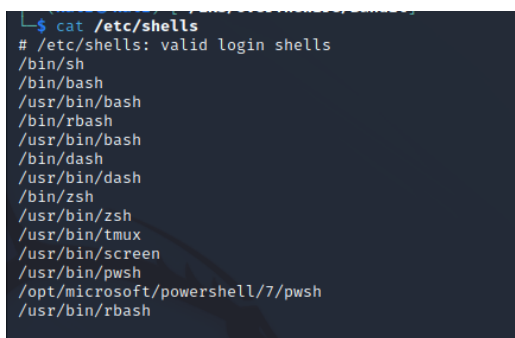
Setelah login disediakan 2 file yang berbeda yaitu pw old dan pw new di soal dijelaskan kita harus melihat perbedaan dari kedua file tersebut yang dimana perbedaan itu adalah password untuk next level langsung saja saya menggunakan command **diff** yang berfungsi untuk mencari perbedaan



```
bandit17@bandit:~$ ls
passwords.new passwords.old
bandit17@bandit:~$ diff passwords.old passwords.new
42c42
< 09wUlyMU4Yh0zl1Lzxoz0voIBzZ2TUAf
> hga5tuuCLF6fFzUpnagiMN8ssu9LFrdg
bandit17@bandit:~$
```

09wUlyMU4Yh0zl1Lzxoz0voIBzZ2TUAf

Level 18: Ketika kita mencoba login maka di level ini akan terlogout otomatis dikarenakan bisa dibilang ‘bash’ sudah di sabotase di level kali ini jadi kita harus mencari shells lain untuk kita masuki dengan ssh. Untuk mengetahui shells apa saja yang berada di dalam sistem kita bisa melihat shells kita sendiri dan mencobanya ke sistem lain kita melihatnya dengan “cat /etc/shells”

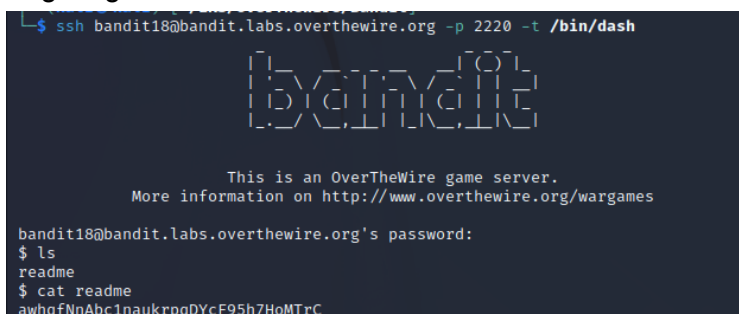


```
$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
/bin/zsh
/usr/bin/zsh
/usr/bin/tmux
/usr/bin/screen
/usr/bin/pwsh
/opt/microsoft/powershell/7/pwsh
/usr/bin/rbash
```

Disini saya menggunakan /bin/dash dan menggunakan cara ini untuk login

“ssh bandit18@bandit.labs.overthewire.org -p 2220 -t /bin/dash”

setelah berhasil masuk seperti dijelaskan di soal password berada di file readme jadi bisa langsung dilihat.



```
$ ssh bandit18@bandit.labs.overthewire.org -p 2220 -t /bin/dash
This is an OverTheWire game server.
More information on http://www.overthewire.org/wargames
bandit18@bandit.labs.overthewire.org's password:
$ ls
readme
$ cat readme
awhqfNnAbc1naukrpqDYcF95h7HoMTrC
```

awhqfNnAbc1naukrpqDYcF95h7HoMTrC

Level 19

To gain access to the next level, you should use the setuid binary in the homedirectory. Execute it without arguments to find out how to use it. The password for this level can be found in the usual place (/etc/bandit_pass), after you have used the setuid binary.

Ketika saya ls saya menemukan bandit-20 dan setelah saya baca mengenai uid ternyata bandit20-do ini berupa user yang bisa kita manfaatkan untuk mengakses file yang tidak bisa kita akses menggunakan user yang sekarang yaitu bandit 19

```
bandit19@bandit:~$ ./bandit20-do id
uid=11019(bandit19) gid=11019(bandit19) euid=11020(bandit20) groups=11019(bandit
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit
bandit0 bandit10 bandit12 bandit14 bandit16 bandit18 bandit2 bandit21
bandit1 bandit11 bandit13 bandit15 bandit17 bandit19 bandit20 bandit22
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit
bandit0 bandit10 bandit12 bandit14 bandit16 bandit18 bandit2 bandit21
bandit1 bandit11 bandit13 bandit15 bandit17 bandit19 bandit20 bandit22
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit
bandit0 bandit10 bandit12 bandit14 bandit16 bandit18 bandit2 bandit21
bandit1 bandit11 bandit13 bandit15 bandit17 bandit19 bandit20 bandit22
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit20
VxCazJaVykl6W36BkBU0mJTCM8rR95XT
bandit19@bandit:~$
```

VxCazJaVykl6W36BkBU0mJTCM8rR95XT

Bandit level 20

There is a setuid binary in the homedirectory that does the following: it makes a connection to localhost on the port you specify as a commandline argument. It then reads a line of text from the connection and compares it to the password in the previous level (bandit20). If the password is correct, it will transmit the password for the next level (bandit21).

Diberikan id yang bisa memberi kita password jika kita berkomunikasi dengan 'user tersebut' user (udin) ini akan memberi kita password dengan menyerahkan password kita yang sebelumnya.

Jadi disini saya menjalankan termux untuk bisa mengoperasikan berbagai operasi dengan satu user.

Disini saya menggunakan 2 tab satu untuk membuat TCP server menggunakan nc dan satunya lagi adalah dari sisi client yaitu udin

```
bandit20@bandit:~$ nc -lv localhost
nc: getaddrinfo: Servname not supported for ai_socktype
bandit20@bandit:~$ nc -lv 6969
Listening on 0.0.0.0 6969
```

Saya membuat server menggunakan port 6969

```
bandit20@bandit:~$ ls
suconnect
bandit20@bandit:~$ ./suconnect 6969
```

Setelah itu saya hubungkan Udin dengan server yang sudah dibuat

Kembali ke sisi server saya sudah mendapatkan respon jadi langsung saja saya serahkan password sebelumnya

```
Listening on 0.0.0.0 6969
Connection received on localhost 49290
VxCazJaVyki6W36BkBU0mJTCM8rR95XT
NvEJF7oVjkddlPSrdKEFOllh9V1IBcq
bandit20@bandit:~$
```

NvEJF7oVjkddlPSrdKEFOllh9V1IBcq

Bandit21

A program is running automatically at regular intervals from **cron**, the time-based job scheduler. Look in **/etc/cron.d/** for the configuration and see what command is being executed.

disini saya diajari apa itu cron, simpelnya cron adalah program yang berjalan otomatis sesuai prosedur yang sudah ditentukan. Dengan crontab files adalah prosedur tersebut

Menuju ke solusi bandit 21.

Seperti yang ada di soal kita disuruh untuk melihat ke **/etc/cron.d** untuk melihat command apa yang dieksekusi.

```
bandit21@bandit:/etc/cron.d$ cat cronjob_bandit22
@reboot bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
* * * * * bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
```

Disini bisa dilihat command yang dieksekusi adalah bandit22.sh yang berada di **/usr/bin**

Jika dijabarkan fungsi program diatas adalah “@reboot = setiap kali reboot”, “bandit22 = user cron”, “/usr/bin/cronjob_bandit22.sh” adalah command yang dijalankan” sedangkan “&> /dev/null” adalah special file type yaitu berfungsi seperti tempat sampah atau **void** yang dimana jika kita meletakkan apapun kedalam file tersebut maka akan hilang.

Jadi, commandnya kurang lebih adalah menjadi seperti:

Setiap kali reboot sebagai user bandit22 jalankan program bandit22.sh yang berada di /usr/bin setelah selesai dijalankan buang program yang dijalankan tadi ke tempat sampah.

Yang berarti disini kita tidak akan dapat password tersebut, jadi kita harus melihat isi program tersebut.

Untuk melihat programnya langsung saja kita cat sesuai dimana program tersebut berada yaitu 'cat /usr/bin/cronjob_bandit22.sh'

```
bandit21@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit22.sh
#!/bin/bash
chmod 644 /tmp/t706lds9S0RqQh9aMcz6ShpAoZKF7fgv
cat /etc/bandit_pass/bandit22 > /tmp/t706lds9S0RqQh9aMcz6ShpAoZKF7fgv
```

Jika dijabarkan fungsi program diatas adalah

Chmod 644 untuk mengganti permission file diatas menjadi read and write untuk owner sedangkan yang lain hanya read saja. Setelah owner bisa memodifikasi file tersebut maka file tersebut akan diisi dengan password yang ada di bandit_pass/bandit22.

Yang berarti password berada di dalam file tmp maka langsung saja saya cat file tersebut untuk melihat passwordnya

```
bandit21@bandit:/etc/cron.d$ cat /tmp/t706lds9S0RqQh9aMcz6ShpAoZKF7fgv
WdDozAdTM2z9DiFEQ2mGlwnGmfj4EZff
```

WdDozAdTM2z9DiFEQ2mGlwnGmfj4EZff

Bandit 22

*A program is running automatically at regular intervals from **cron**, the time-based job scheduler. Look in **/etc/cron.d/** for the configuration and see what command is being executed.*

Sama seperti bandit 21 langsung saja kita melihat bagaimana program ini dijalankan

```
bandit22@bandit:/etc/cron.d$ cat cronjob_bandit23
@reboot bandit23 /usr/bin/cronjob_bandit23.sh &> /dev/null
* * * * * bandit23 /usr/bin/cronjob_bandit23.sh &> /dev/null
```

Disini program yang dijalankan berada di /usr/bin yang diberinama cronjob_bandit23

Langsung saja kita lihat bagaimana program tersebut dijalankan

```
bandit22@bandit:/usr/bin$ cat /usr/bin/cronjob_bandit23.sh
#!/bin/bash

myname=$(whoami)
mytarget=$(echo I am user $myname | md5sum | cut -d ' ' -f 1)

echo "Copying passwordfile /etc/passwd/$myname to /tmp/$mytarget"

cat /etc/passwd/$myname > /tmp/$mytarget
bandit22@bandit:/usr/bin$ myname=bandit23
bandit22@bandit:/usr/bin$ echo I am user $myname | md5sum | cut -d ' ' -f 1
8ca319486bfbbc3663ea0fbe81326349
bandit22@bandit:/usr/bin$ cat /tmp/8ca319486bfbbc3663ea0fbe81326349
QYw0Y2aiA672PsMmh9puTQuhoz8SyR2G
```

Inti dari program ini adalah menentukan file tmp dengan menggunakan md5 dari user dengan beberapa modifikasi dan untuk mendapatkan password selanjutnya cukup simpel yaitu dengan mengulangi program tersebut

QYw0Y2aiA672PsMmh9puTQuhoz8SyR2G