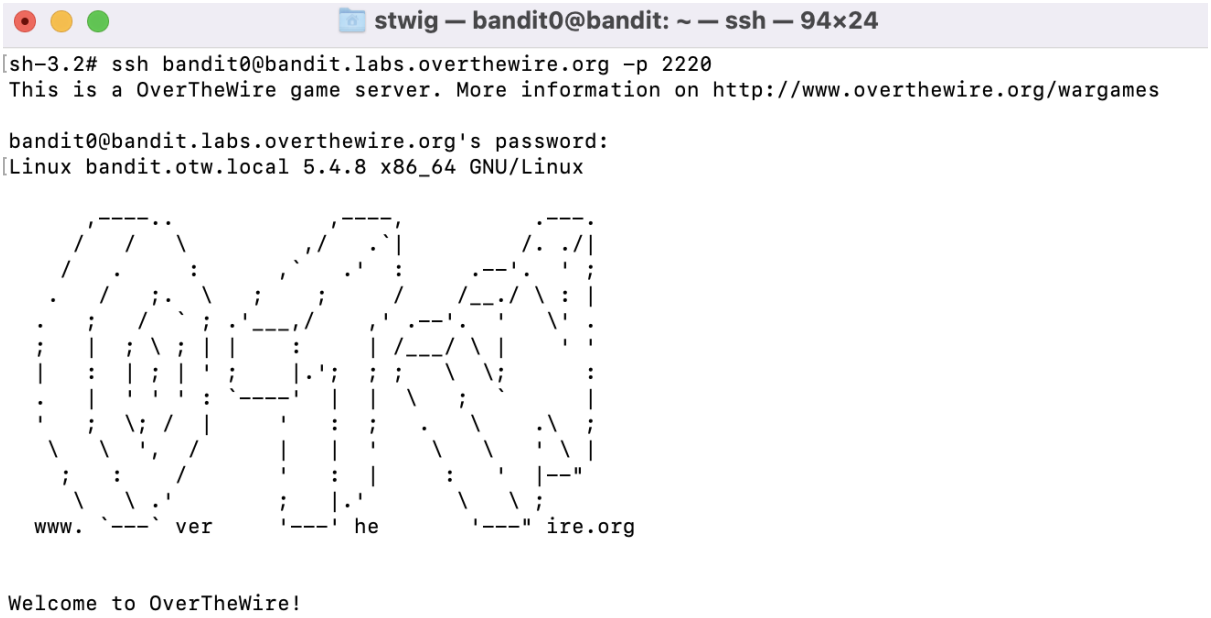
Sapir Twig

OverTheWire - bandit

https://overthewire.org/wargames/bandit/

Level 0

*The goal of this level is for you to log into the game using SSH. The host to which you need to connect is **bandit.labs.overthewire.org**, on port 2220. The username is **bandit0** and the password is **bandit0**. Once logged in, go to the Level 1 page to find out how to beat Level 1.*



Level 0 -> Level 1

*The password for the next level is stored in a file called **readme** located in the home directory. Use this password to log into bandit1 using SSH. Whenever you find a password for a level, use SSH (on port 2220) to log into that level and continue the game.*

I used the ‘ls’ command to view all the files list and ‘cat’ to execute all the information from the file.

```
bandit0@bandit:~$ ls
readme
bandit0@bandit:~$ cat readme
boJ9jbbUNNfktd78OOpsqOltutMc3MY1
bandit0@bandit:~$
```

Now, from here type ‘exit’ and SSH back into the next level by running

```
sh-3.2# ssh bandit1@bandit.labs.overthewire.org
bandit1@bandit.labs.overthewire.org's password:
```

And I'm in!

```
bandit1@bandit:~$
```

Level 1 -> Level 2

*The password for the next level is stored in a file called **-** located in the home directory*

```
bandit1@bandit:~$ ls -a
-  .  ..  .bash_logout  .bashrc  .profile
bandit1@bandit:~$ cat ./-
CV1DtqXWVFXTvM2F0k09SHz0YwRINYA9
bandit1@bandit:~$
```

As in the previous level, we ssh to the next user (bandit2) with the given password.

Level 2 -> Level 3

The password for the next level is stored in a file called spaces in this filename located in the home directory

```
bandit2@bandit:~$ dir
spaces\ in\ this\ filename
bandit2@bandit:~$ cat spaces\ in\ this\ filename
UmHadQclWmgdLOKQ3YNgjWxGoRMb5luK
bandit2@bandit:~$
```

Now we ssh to bandit3 with this password

```
bandit3@bandit:~$
```

We are in (:

Level 3 -> Level 4

The password for the next level is stored in a hidden file in the inhere directory.

```
bandit3@bandit:~$ ls
inhere
bandit3@bandit:~$ cd inhere
bandit3@bandit:~/inhere$ ls -a
.  ..  .hidden
bandit3@bandit:~/inhere$ cat .hidden
pIwrPrtPN36QITSp3EQaw936yaFoFgAB
bandit3@bandit:~/inhere$
```

And now we ssh to the next user

```
bandit4@bandit:~$
```

Level 4 -> Level 5

*The password for the next level is stored in the only human-readable file in the **inhere** directory. Tip: if your terminal is messed up, try the "reset" command.*

```
bandit4@bandit:~$ ls
inhere
bandit4@bandit:~$ cd inhere
bandit4@bandit:~/inhere$ ls -a
.   -file00  -file02  -file04  -file06  -file08
..  -file01  -file03  -file05  -file07  -file09
bandit4@bandit:~/inhere$
```

I figured out what type of file each one of them (because of the hint)

```
bandit4@bandit:~/inhere$ file ./-*
./-file00: data
./-file01: data
./-file02: data
./-file03: data
./-file04: data
./-file05: data
./-file06: data
./-file07: ASCII text
./-file08: data
./-file09: data
bandit4@bandit:~/inhere$
```

```
bandit4@bandit:~/inhere$ cat ./-file07
koReBOKuIDDepwhWk7jZC0RTdopnAYKh
bandit4@bandit:~/inhere$
```

Level 5 -> Level 6

The password for the next level is stored in a file somewhere under the inhere directory and has all of the following properties:

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

```
bandit5@bandit:~$ ls
inhere
bandit5@bandit:~$ cd inhere
bandit5@bandit:~/inhere$ ls -a
.            maybehere02  maybehere06  maybehere10  maybehere14  maybehere18
..           maybehere03  maybehere07  maybehere11  maybehere15  maybehere19
maybehere00  maybehere04  maybehere08  maybehere12  maybehere16
maybehere01  maybehere05  maybehere09  maybehere13  maybehere17
bandit5@bandit:~/inhere$ find -type f -size 1033c
./maybehere07/.file2
bandit5@bandit:~/inhere$ cat ./maybehere07/.file2
DXjZPULLxYr17uwoI01bNLQbtFemEgo7
```

And we solved it!

```
bandit6@bandit:~$
```

Level 6 -> Level 7

The password for the next level is stored somewhere on the server and has all of the following properties:

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

```
bandit6@bandit:~$ find / -user 'bandit7' -group 'bandit6' -size 33c |& grep -iv 'premission denied'
```

And I found:

```
find: '/var/lib/polkit-1': Permission denied
/var/lib/dpkg/info/bandit7.password
find: '/var/log': Permission denied
find: '/var/cache/apt/archives/partial': Permission denied
find: '/var/cache/ldconfig': Permission denied
bandit6@bandit:~$
```

And here is the password:

```
bandit6@bandit:~$ cat /var/lib/dpkg/info/bandit7.password
HKBPTKQnIay4Fw76bEy8PVxKEDQRKTzs
bandit6@bandit:~$
```

Level 7 -> Level 8

The password for the next level is stored in the file data.txt next to the word millionth

I checked the current directory

```
bandit7@bandit:~$ ls -al
total 4108
drwxr-xr-x  2 root    root       4096 May  7  2020 .
drwxr-xr-x 41 root    root       4096 May  7  2020 ..
-rw-r--r--  1 root    root        220 May 15  2017 .bash_logout
-rw-r--r--  1 root    root       3526 May 15  2017 .bashrc
-rw-r-----  1 bandit8 bandit7 4184396 May  7  2020 data.txt
-rw-r--r--  1 root    root        675 May 15  2017 .profile
bandit7@bandit:~$
```

I used ‘strings’ and ‘grep’ commands to view the password.

```
bandit7@bandit:~$  strings data.txt | grep millionth -C3
whirr   a99nLztiXHlqSqxwCicQAKgT1c8zO8rC
covenants       wRcmgvIJTRSgpV1iurw1gc7Ar2IU1EVQ
Halley  H7Mg53D6bPDpleFYGp1KF1SKTQh7jiNl
millionth       cvX2JJa4CFALtqS87jk27qwqGhBM9plV
shied   OfMT7PpeOvra4NWlZz7JOzyjL236NFVF
sesame  K1M1XuyPoCOkuEz6QB9gsyCW9dUqGXKx
Elul    FmCv0XrAW75I468ELOulmg7lGEslnUFL
bandit7@bandit:~$
```

Level 8 -> Level 9

The password for the next level is stored in the file data.txt and is the only line of text that occurs only once

```
bandit8@bandit:~$ ls -a
.  ..  .bash_logout  .bashrc  data.txt  .profile
bandit8@bandit:~$ cat data.txt | sort | uniq -u
UsvVyFSfZZWbi6wgC7dAFyFuR6jQQUhR
bandit8@bandit:~$
```

Level 9 -> Level 10

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.

```
bandit9@bandit:~$ strings data.txt | grep '='
========== the*2i"4
=:G e
========== password
<I=zsGi
Z)========== is
A=|t&E
Zdb=
c^ LAh=3G
*SF=s
&========== truKLdjsbJ5g7yyJ2X2R0o3a5HQJFuLk
S=A.H&^
bandit9@bandit:~$
```

The password for the next level is stored in the file data.txt, which contains base64 encoded data

```
bandit10@bandit:~$ ls
data.txt
bandit10@bandit:~$ cat data.txt
VGhlIHBhc3N3b3JkIGlzIElGdWt3S0dzRlc4TU9xM0lSRnFyeEUxaHhUTkViVVBSCg==
bandit10@bandit:~$ base64 -d data.txt
The password is IFukwKGsFW8MOq3IRFqrxE1hxTNEbUPR
bandit10@bandit:~$
```

Level 11 -> Level 12

The password for the next level is stored in the file data.txt, where all lowercase (a-z) and uppercase (A-Z) letters have been rotated by 13 positions

```
bandit11@bandit:~$ ls -a
.  ..  .bash_logout  .bashrc  data.txt  .profile
bandit11@bandit:~$ cat data.txt
Gur cnffjbeq vf 5Gr8L4qetPEsPk8htqjhRK8XSP6x2RHh
bandit11@bandit:~$ echo Gur cnffjbeq vf 5Gr8L4qetPEsPk8htqjhRK8XSP6x2RHh | tr 'A-Za-z' 'N-ZA-Mn-za-m'
The password is 5Te8Y4drgCRfCx8ugdwuEX8KFC6k2EUu
bandit11@bandit:~$
```

Level 12 -> Level 13

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!)

As mentioned in the question make a new directory in /tmp and rename the file.

```
bandit12@bandit:~$ ls
data.txt
bandit12@bandit:~$ mkdir /tmp/pc
bandit12@bandit:~$ cp data.txt /tmp/pc
bandit12@bandit:~$ cd /tmp/pc
bandit12@bandit:/tmp/pc$ ls
data.txt
bandit12@bandit:/tmp/pc$ mv data.txt file.txt
bandit12@bandit:/tmp/pc$ ls
file.txt
bandit12@bandit:/tmp/pc$ file file.txt
file.txt: ASCII text
bandit12@bandit:/tmp/pc$ cat file.txt
00000000: 1f8b 0808 0650 b45e 0203 6461 7461 322e  .....P.^..data2.
00000010: 6269 6e00 013d 02c2 fd42 5a68 3931 4159  bin..=...BZh91AY
00000020: 2653 598e 4f1c c800 001e 7fff fbf9 7fda  &SY.O...........
00000030: 9e7f 4f76 9fcf fe7d 3fff f67d abde 5e9f  ..Ov...}?..}..^.
00000040: f3fe 9fbf f6f1 feee bfdf a3ff b001 3b1b  ..............;.
00000050: 5481 a1a0 1ea0 1a34 d0d0 001a 68d3 4683  T......4....h.F.
00000060: 4680 0680 0034 1918 4c4d 190c 4000 0001  F....4..LM..@...
00000070: a000 c87a 81a3 464d a8d3 43c5 1068 0346  ...z..FM..C..h.F
00000080: 8343 40d0 3400 0340 66a6 8068 0cd4 f500  .C@.4..@f..h....
00000090: 69ea 6800 0f50 68f2 4d00 680d 06ca 0190  i.h..Ph.M.h.....
```

'xxd' program is used to make a hexdump or to do the reverse. Option -r convert hexdump into the binary. File file.txt is a hexdump and convert into a binary file file1.bin using the command

```
bandit12@bandit:/tmp/pc$ xxd -r file.txt > file1.bin
bandit12@bandit:/tmp/pc$ ls
file1.bin  file.txt
bandit12@bandit:/tmp/pc$
```

Using command file file1.bin, we found that file1.bin is a *gzip compressed data*.
'zcat' is a program supplied with gzip and is used to decompress *gzip-compressed files*.

```
bandit12@bandit:/tmp/pc$ file file1.bin
file1.bin: gzip compressed data, was "data2.bin", last modified: Thu May  7
 18:14:30 2020, max compression, from Unix
bandit12@bandit:/tmp/pc$ zcat file1.bin > file2
bandit12@bandit:/tmp/pc$ ls
file1.bin  file2  file.txt  myfile2
bandit12@bandit:/tmp/pc$
```

Again using the file command on file2, we found that it is bzip2 compressed data. 'bzcat' program is supplied with bzip2 and is used to decompress bzip2 compressed files.

```
bandit12@bandit:/tmp/pc$ file file2
file2: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/pc$ bzcat file2 > file3
```

file3 is gzip compressed file so use the 'zcat' program to decompress it in file4. file4 is a POSIX tar archive.

'tar' program is used for archiving files and options x is used to extract an archive, f is used to specify the name of the tar archive, and v is used for a more detailed listing.

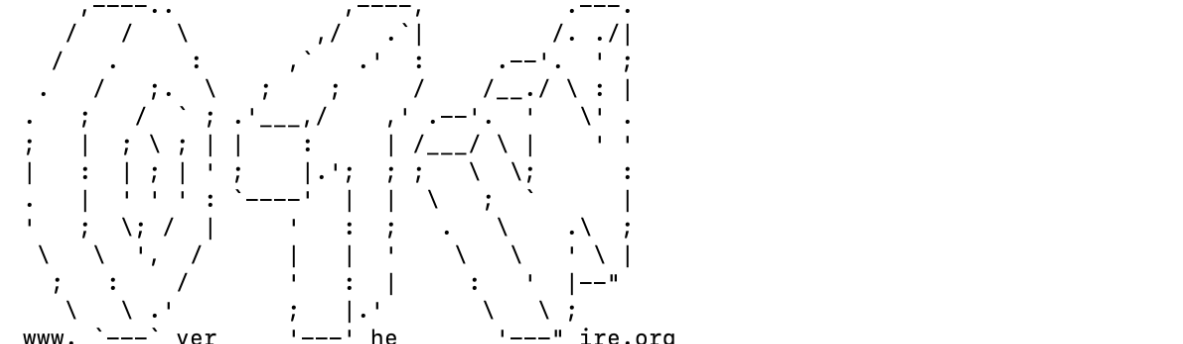
```
bandit12@bandit:/tmp/pc$ file file3
file3: gzip compressed data, was "data4.bin", last modified: Thu May  7 18:
14:30 2020, max compression, from Unix
bandit12@bandit:/tmp/pc$ zcat file3 > file 4
gzip: 4.gz: No such file or directory
bandit12@bandit:/tmp/pc$ zcat file3 > file4
bandit12@bandit:/tmp/pc$ file file4
file4: POSIX tar archive (GNU)
bandit12@bandit:/tmp/pc$ tar -xvf file4
data5.bin
bandit12@bandit:/tmp/pc$ file data5.bin
data5.bin: POSIX tar archive (GNU)
bandit12@bandit:/tmp/pc$ tar -xvf data5.bin
data6.bin
bandit12@bandit:/tmp/pc$ file data6.bin
data6.bin: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/pc$ bzcat data6.bin > file7
bandit12@bandit:/tmp/pc$ file file7
file7: POSIX tar archive (GNU)
bandit12@bandit:/tmp/pc$ tar -xvf file7
data8.bin
bandit12@bandit:/tmp/pc$ file data8.bin
data8.bin: gzip compressed data, was "data9.bin", last modified: Thu May  7
 18:14:30 2020, max compression, from Unix
bandit12@bandit:/tmp/pc$ zcat data8.bin > file9
bandit12@bandit:/tmp/pc$ file file9
file9: ASCII text
bandit12@bandit:/tmp/pc$ cat file9
The password is 8ZjyCRiBWFYkneahHwxCv3wb2a1ORpYL
bandit12@bandit:/tmp/pc$
```

Level 13 -> Level 14

The password for the next level is stored in /etc/bandit_pass/bandit14 and can only be read by user bandit14. For this level, you don't get the next password, but you get a private SSH key that can be used to log into the next level. Note: localhost is a hostname that refers to the machine you are working on

```
bandit13@bandit:~$ ls
sshkey.private
bandit13@bandit:~$ ssh bandit14@localhost -i sshkey.private
Could not create directory '/home/bandit13/.ssh'.
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc
.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/bandit13/.ssh/know
n_hosts).
This is a OverTheWire game server. More information on http://www.overthewi
re.org/wargames

Linux bandit.otw.local 5.4.8 x86_64 GNU/Linux
```



Level 14 -> Level 15

The password for the next level can be retrieved by submitting the password of the current level to port 30000 on localhost.

```
bandit14@bandit:~$ cat /etc/bandit_pass/bandit14
4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e
bandit14@bandit:~$ nc localhost 30000
4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e
Correct!
BfMYroe26WYalil77FoDi9qh59eK5xNr

bandit14@bandit:~$
```

Level 15 -> Level 16

The password for the next level can be retrieved by submitting the password of the current level to port 30001 on localhost using SSL encryption.

Helpful note: Getting "HEARTBEATING" and "Read R BLOCK"? Use -ign_eof and read the "CONNECTED COMMANDS" section on the manpage. Next to 'R' and 'Q', the 'B' command also works in this version of that command…

```
bandit15@bandit:~$ openssl s_client -connect localhost:30001
CONNECTED(00000003)
depth=0 CN = localhost
verify error:num=18:self signed certificate
verify return:1
depth=0 CN = localhost
verify return:1
---
    Extended master secret: yes
---
BfMYroe26WYalil77FoDi9qh59eK5xNr
Correct!
cluFn7wTiGryunymYOu4RcffSxQluehd

closed
bandit15@bandit:~$
```

Level 16 -> Level 17

The credentials for the next level can be retrieved by submitting the password of the current level to a port on localhost in the range 31000 to 32000. First, find out which of these ports have a server listening on them. Then find out which of those speak SSL and which don't. There is only 1 server that will give the next credentials, the others will simply send back to you whatever you send to it.

```
bandit16@bandit:~$ nmap -p 31000-32000 -sV localhost

Starting Nmap 7.40 ( https://nmap.org ) at 2022-08-07 21:53 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00023s latency).
Not shown: 996 closed ports
PORT      STATE    SERVICE     VERSION
31046/tcp open     echo
31518/tcp filtered unknown
31691/tcp open     echo
31790/tcp open     ssl/unknown
31960/tcp open     echo
```

I tried the port with the SSL - 31790

```
bandit16@bandit:~$ openssl s_client -connect localhost:31790
CONNECTED(00000003)
depth=0 CN = localhost
verify error:num=18:self signed certificate
verify return:1
depth=0 CN = localhost
verify return:1
```

I created a folder named ‘abc’ and copied the RSA private key.

```
bandit16@bandit:~$ mkdir /tmp/abc
bandit16@bandit:~$ cd /tmp/abc
bandit16@bandit:/tmp/abc$ nano sshkey.private
Unable to create directory /home/bandit16/.nano: Permission denied
It is required for saving/loading search history or cursor positions.

Press Enter to continue

bandit16@bandit:/tmp/abc$ cat sshkey.private
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPIOjon6iWfbp7c3jx34YkYWqUH57SUdyJ
imZzeyGC0gtZPGujUSxiJSWI/oTqexh+cAMTSMlOJf7+BrJObArnxd9Y7YT2bRPQ
Ja6Lzb558YW3FZl87ORiO+rW4LCDCNd2lUvLE/GL2GWyuKN0K5iCd5TbtJzEkQTu
DSt2mcNn4rhAL+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW3OekePQAzL0VUYbW
JGTi65CxbCnzc/w4+mqQyvmzpWtMAzJTzAzQxNbkR2MBGySxDLrjg0LWN6sK7wNX
x0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XFOJuaQIDAQABAoIBABagpxpM1aoLWfvD
KHcj10nqcoBc4oE11aFYQwik7xfW+24pRNuDE6SFthOar69jp5RlLwD1NhPx3iBl
J9nOM8OJ0VToum43UOS8YxF8WwhXriYGnc1sskbwpXOUDc9uX4+UESzH22P29ovd
d8WErY0gPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9AlbssgTcCXkMQnPw9nC
YNN6DDP2lbcBrvgT9YCNL6C+ZKufD52yOQ9qOkwFTEQpjtF4uNtJom+asvlpmS8A
vLY9r60wYSvmZhNqBUrj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama
+TOWWgECgYEA8JtPxP0GRJ+IQkX262jM3dEIkza8ky5moIwUqYdsx0NxHgRRhORT
8c8hAuRBb2G82so8vUHk/fur85OEfc9TncnCY2crpoqsghifKLxrLgtT+qDpfZnx
SatLdt8GfQ85yA7hnWWJ2MxF3NaeSDm75Lsm+tBbAiyc9P2jGRNtMSkCgYEAypHd
HCctNi/FwjulhttFx/rHYKhLidZDFYeiE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt
SghaTdcG0Knyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCIvGCSx+X3l5SiWg0A
R57hJglezIiVjv3aGwHwvlZvtszK6zV6oXFAu0ECgYAbjo46T4hyP5tJi93V5HDi
Ttiek7xRVxUl+iU7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFMLy9FL2m9oQWCg
R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB3OhYimtiG2Cg5JCqIZFHxD6MjEGOiu
L8ktHMPvodBwNsSBULpG0QKBgBAplTfC1HOnWiMGOU3KPwYWt0O6CdTkmJOmL8Ni
blh9elyZ9FsGxsgtRBXRsqXuz7wtsQAgLHxbdLq/ZJQ7YfzOKU4ZxEnabvXnvWkU
YOdjHdSOoKvDQNWu6ucyLRAWFuISeXw9a/9p7ftpxm0TSgyvmfLF2MIAEwyzRqaM
77pBAoGAMmjmIJdjp+Ez8duyn3ieo36yrttF5NSsJLAbxFpdlc1gvtGCWW+9Cq0b
dxviW8+TFVEBl1O4f7HVm6EpTscdDxU+bCXWkfjuRb7Dy9GOtt9JPsX8MBTakzh3
vBgsyi/sN3RqRBcGU40fOoZyfAMT8s1m/uYv52O6IgeuZ/ujbjY=
-----END RSA PRIVATE KEY-----

bandit16@bandit:/tmp/abc$
```

I used the chmod command to change the access premission of the file

```
bandit16@bandit:/tmp/abc$ chmod 700 sshkey.private
bandit16@bandit:/tmp/abc$ ls -l sshkey.private
-rwx------ 1 bandit16 root 1676 Aug  8 11:21 sshkey.private
bandit16@bandit:/tmp/abc$
```

I used the ssh command to continue the next level

```
bandit16@bandit:/tmp/abc$ ssh bandit17@localhost -i sshkey.private
Could not create directory '/home/bandit16/.ssh'.
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc.
Are you sure you want to continue connecting (yes/no)? yes
```

And we are in!

```
bandit17@bandit:~$
```

Level 17 -> Level 18

There are 2 files in the homedirectory: passwords.old and passwords.new. The password for the next level is in passwords.new and is the only line that has been changed between passwords.old and passwords.new

NOTE: if you have solved this level and see 'Byebye!' when trying to log into bandit18, this is related to the next level, bandit19

```
bandit17@bandit:~$ ls -a
.   .bandit16.password  .bashrc        passwords.old  .ssh
..  .bash_logout        passwords.new  .profile
bandit17@bandit:~$ diff passwords.new passwords.old
42c42
< kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd
---
> w0Yfolrc5bwjS4qw5mq1nnQi6mF03bii
bandit17@bandit:~$
```

And we see the 'Byebye!' when we tried to log in

```
  For more information regarding individual wargames, visit
  http://www.overthewire.org/wargames/

  For support, questions or comments, contact us through IRC on
  irc.overthewire.org #wargames.

  Enjoy your stay!

Bvebve !
```

Level 18 -> Level 19

The password for the next level is stored in a file readme in the homedirectory. Unfortunately, someone has modified .bashrc to log you out when you log in with SSH.

```
sh-3.2# ssh bandit18@bandit.labs.overthewire.org -p 2220 bash --norc
This is a OverTheWire game server. More information on http://www.overthewire.org/wargames

bandit18@bandit.labs.overthewire.org's password:
ls
readme
cat readme
IueksS7Ubh8G3DCwVzrTd8rAVOwq3M5x
```

Level 19 -> Level 20

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.

```
bandit19@bandit:~$ ls -a
.  ..  bandit20-do  .bash_logout  .bashrc  .profile
bandit19@bandit:~$ ./bandit20-do
Run a command as another user.
  Example: ./bandit20-do id
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit20
GbKksEFF4yrVs6il55v6gwY5aVje5f0j
```

Level 20 -> Level 21

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).

NOTE: Try connecting to your own network daemon to see if it works as you think

I ran ‘suconnect’ to figure out what it is:

```
bandit20@bandit:~$ ls -a
.  ..  .bash_logout  .bashrc  .profile  suconnect
bandit20@bandit:~$ ./suconnect
Usage: ./suconnect <portnumber>
This program will connect to the given port on localhost using TCP. If it receives the
correct password from the other side, the next password is transmitted back.
bandit20@bandit:~$
```

I know now that ‘suconnect’ creates a connection so I opened another terminal tab with bandit20 to create a connection with Netcat (nc), I chose 32000 port randomly

```
bandit20@bandit:~$ nc -lvnp 32000
```

Then I connected to this port using USID binary on the original bandit20

```
bandit20@bandit:~$ ./suconnect 32000
```

I submitted my password for getting my password for the next level:

```
bandit20@bandit:~$ nc -lvnp 32000
listening on [any] 32000 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 57876
GbKksEFF4yrVs6il55v6gwY5aVje5f0j
gE269g2h3mw3pwgrj0Ha9Uoqen1c9DGr
```

```
bandit20@bandit:~$ ./suconnect 32000
Read: GbKksEFF4yrVs6il55v6gwY5aVje5f0j
Password matches, sending next password
bandit20@bandit:~$
```

Level 21 -> Level 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.

```
bandit21@bandit:~$ cd /etc/cron.d/
bandit21@bandit:/etc/cron.d$ ls -a
.   cronjob_bandit15_root  cronjob_bandit22  cronjob_bandit24       .placeholder
..  cronjob_bandit17_root  cronjob_bandit23  cronjob_bandit25_root
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
bandit21@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit22.sh
#!/bin/bash
chmod 644 /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
cat /etc/bandit_pass/bandit22 > /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
bandit21@bandit:/etc/cron.d$ cat /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
Yk7owGAcWjwMVRwrTesJEwB7WVOiILLI
bandit21@bandit:/etc/cron.d$
```

Level 22 -> Level 23

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.

NOTE: Looking at shell scripts written by other people is a very useful skill. The script for this level is intentionally made easy to read. If you are having problems understanding what it does, try executing it to see the debug information it prints.

```
bandit22@bandit:~$ cd /etc/cron.d/
bandit22@bandit:/etc/cron.d$ ls -a
.   cronjob_bandit15_root  cronjob_bandit22  cronjob_bandit24       .placeholder
..  cronjob_bandit17_root  cronjob_bandit23  cronjob_bandit25_root
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
bandit22@bandit:/etc/cron.d$ 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/bandit_pass/$myname to /tmp/$mytarget"

cat /etc/bandit_pass/$myname > /tmp/$mytarget
bandit22@bandit:/etc/cron.d$
```

```
bandit22@bandit:/etc/cron.d$ echo I am user bandit23 | md5sum | cut -d ' ' -f 1
8ca319486bfbbc3663ea0fbe81326349
bandit22@bandit:/etc/cron.d$ cat /tmp/8ca319486bfbbc3663ea0fbe81326349
jc1udXuA1tiHqjIsL8yaapX5XIAI6i0n
bandit22@bandit:/etc/cron.d$
```

Level 23 -> Level 24

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

NOTE: This level requires you to create your own first shell-script. This is a very big step and you should be proud of yourself when you beat this level!

NOTE 2: Keep in mind that your shell script is removed once executed, so you may want to keep a copy around…

```
bandit23@bandit:~$ cd /etc/cron.d/
bandit23@bandit:/etc/cron.d$ ls -a
.   cronjob_bandit15_root  cronjob_bandit22  cronjob_bandit24       .placeholder
..  cronjob_bandit17_root  cronjob_bandit23  cronjob_bandit25_root
bandit23@bandit:/etc/cron.d$ cat cronjob_bandit24
@reboot bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
* * * * * bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
bandit23@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit24.sh
#!/bin/bash

myname=$(whoami)

cd /var/spool/$myname
echo "Executing and deleting all scripts in /var/spool/$myname:"
for i in * .*;
do
    if [ "$i" != "." -a "$i" != ".." ];
    then
        echo "Handling $i"
        owner="$(stat --format "%U" ./$i)"
        if [ "${owner}" = "bandit23" ]; then
            timeout -s 9 60 ./$i
        fi
        rm -f ./$i
    fi
done
```

```
bandit23@bandit:/etc/cron.d$ mkdir /tmp/cvb
bandit23@bandit:/etc/cron.d$ cd /tmp/cvb
bandit23@bandit:/tmp/cvb$ ls
bandit23@bandit:/tmp/cvb$ vi getpasswd.sh
```