Level 10->11

This challenge requires converting a file from base64. This could be done by copy/paste to cyberchef, or it can be done all in the CLI.

```
ssh bandit10@bandit.labs.overthewire.org -p 2220
password: truKLdjsbJ5g7yyJ2X2R0o3a5HQJFuLk
bandit10@bandit:~$ ls
data.txt
bandit10@bandit:~$ cat data.txt
VGhlIHBhc3N3b3JkIGlzIElGdWt3S0dzRlc4TU9xM0lSRnFyeEUxaHhUTkViVVBSCg==
bandit10@bandit:~$ base64 --decode data.txt
The password is IFukwKGsFW8MOq3IRFqrxE1hxTNEbUPR
```

Level 11->12

The instructions tell you that all letters have been rotated by 13 positions. That makes this a ROT13 cipher.
https://en.wikipedia.org/wiki/ROT13

```
ssh bandit11@bandit.labs.overthewire.org -p 2220
password: IFukwKGsFW8MOq3IRFqrxE1hxTNEbUPR
bandit11@bandit:~$ ls
data.txt
bandit11@bandit:~$ cat data.txt
Gur cnffjbeq vf 5Gr8L4qetPEsPk8htqjhRK8XSP6x2RHh
```

Below is the website I used to decode the ROT13 cipher text:
https://cryptii.com/pipes/rot13-decoder

The password is 5Te8Y4drgCRfCx8ugdwuEX8KFC6k2EUu

Level 12->13

This one requires keeping track of where you are in the decompression steps and finding a location that you are allowed to create a folder and files.

```
ssh bandit12@bandit.labs.overthewire.org -p 2220
password: 5Te8Y4drgCRfCx8ugdwuEX8KFC6k2EUu
```

The instructions say that I can make a directory in the /tmp folder.

```
bandit12@bandit:mkdir /tmp/j
bandit12@bandit:cd /tmp/j
```

Then I move the data.txt file to the /tmp/j folder so I can manipulate it.

```
bandit12@bandit:/tmp/j$ cp ~/data.txt /tmp/j
bandit12@bandit:/tmp/j$ ls
```

The 'file' command is very important in this challenge. It lets you know what the format of the file is.

```
data.txtbandit12@bandit:/tmp/j$ file data.txt
data.txt: ASCII text
```

Since it is ASCII, I cat the file and can see that it is a hexdump.

```
bandit12@bandit:/tmp/j$ cat data.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.....
000000a0: 0000 69a1 a1a0 1ea0 194d 340d 1ea1 b280  ..i......M4.....
000000b0: f500 3406 2340 034d 3400 0000 3403 d400  ..4.#@.M4...4...
000000c0: 1a07 a832 3400 f51a 0003 43d4 0068 0d34  ...24.....C..h.4
000000d0: 6868 f51a 3d43 2580 3e58 061a 2c89 6bf3  hh..=C%.>X..,.k.
000000e0: 0163 08ab dc31 91cd 1747 599b e401 0b06  .c...1...GY.....
000000f0: a8b1 7255 a3b2 9cf9 75cc f106 941b 347a  ..rU....u.....4z
00000100: d616 55cc 2ef2 9d46 e7d1 3050 b5fb 76eb  ..U....F..0P..v.
00000110: 01f8 60c1 2201 33f0 0de0 4aa6 ec8c 914f  ..`."3...J....O
00000120: cf8a aed5 7b52 4270 8d51 6978 c159 8b5a  ....{RBp.Qix.Y.Z
00000130: 2164 fb1f c26a 8d28 b414 e690 bfdd b3e1  !d...j.(........
00000140: f414 2f9e d041 c523 b641 ac08 0c0b 06f5  ../..A.#.A......
00000150: dd64 b862 1158 3f9e 897a 8cae 32b0 1fb7  .d.b.X?..z..2...
00000160: 3c82 af41 20fd 6e7d 0a35 2833 41bd de0c  <..A .n}.5(3A...
00000170: 774f ae52 a1ac 0fb2 8c36 ef58 537b f30a  wO.R.....6.XS{..
00000180: 1510 cab5 cb51 4231 95a4 d045 b95c ea09  .....QB1...E.\..
00000190: 9fa0 4d33 ba43 22c9 b5be d0ea eeb7 ec85  ..M3.C".........
000001a0: 59fc 8bf1 97a0 87a5 0df0 7acd d555 fc11  Y.........z..U..
000001b0: 223f fdc6 2be3 e809 c974 271a 920e acbc  "?..+....t'.....
000001c0: 0de1 f1a6 393f 4cf5 50eb 7942 86c3 3d7a  ....9?L.P.yB..=z
000001d0: fe6d 173f a84c bb4e 742a fc37 7b71 508a  .m.?.L.Nt*.7{qP.
000001e0: a2cc 9cf1 2522 8a77 39f2 716d 34f9 8620  ....%".w9.qm4..
000001f0: 4e33 ca36 eec0 cd4b b3e8 48e4 8b91 5bea  N3.6...K..H...[.
00000200: 01bf 7d21 0b64 82c0 3341 3424 e98b 4d7e  ..}!.d..3A4$..M~
00000210: c95c 1b1f cac9 a04a 1988 43b2 6b55 c6a6  .\.....J..C.kU..
00000220: 075c 1eb4 8ecf 5cdf 4653 064e 84da 263d  .\....\.FS.N..&=
00000230: b15b bcea 7109 5c29 c524 3afc d715 4894  .[..q.\).$:...H.
00000240: 7426 072f fc28 ab05 9603 b3fc 5dc9 14e1  t&./.(......]...
00000250: 4242 393c 7320 98f7 681d 3d02 0000       BB9<s ..h.=...
```

The 'xxd -r' command will reverse the hexdump back to a binary file.

```
bandit12@bandit:/tmp/j$ xxd -r data.txt revhexdump
```

But the instructions tell us that the file was compressed multiple times, so we need to figure out how to decompress the file. First step is determining the current file format.

```
bandit12@bandit:/tmp/j$ file revhexdump
revhexdump: gzip compressed data, was "data2.bin", last modified: Thu May  7
18:14:30 2020, max compression, from Unix
```

It is gzip compressed, so we need to convert the file type to a gzip format. We use the mv command to do this.

```
bandit12@bandit:/tmp/j$ mv revhexdump data2.bin.gz
```

Once it is in the gzip format, we can use 'gzip -d' to decompress it.

```
bandit12@bandit:/tmp/j$ gzip -d data2.bin.gz
```

This will decompress the file and rename it without the '.gz' ending.

```
bandit12@bandit:/tmp/j$ ls
data2.bin  data.txt
```

So, data2.bin is our newly uncompressed file, now we need to figure out what type of a file it is.

```
bandit12@bandit:/tmp/j$ file data2.bin
data2.bin: bzip2 compressed data, block size = 900k
```

It is a bzip file, so we need to convert the binary to the .bz2 filetype.

```
bandit12@bandit:/tmp/j$ mv data2.bin data2.bz2
```

Then decompress it.
bandit12@bandit:/tmp/j$ bzip2 -d data2.bz2
Then figure out what the newly decompressed file type is.
bandit12@bandit:/tmp/j$ file data2
data2: gzip compressed data, was "data4.bin", last modified: Thu May 7 18:14:30 2020, max compression, from Unix
It is gzip compressed and was named data4.bin, so I will rename it and reformat it as data4.bin.gz
bandit12@bandit:/tmp/j$ mv data2 data4.bin.gz
Decompress it.
bandit12@bandit:/tmp/j$ gzip -d data4.bin.gz
Figure out the file type.
bandit12@bandit:/tmp/j$ file data4.bin
data4.bin: POSIX tar archive (GNU)
It's a tarball, which is a way of storing data, but not compressing it. So, there is no need to change the file type and try to decompress it (as we have been doing), because it is not compressed. We are going to use 3x switches with the tar command: -x (Extract files from the archive), -v (verbose), -f (use the file archive-this works since the format of this file is a tar archive).
bandit12@bandit:/tmp/j$ tar -xvf data4.bin
data5.bin
bandit12@bandit:/tmp/j$ file data5.bin
data5.bin: POSIX tar archive (GNU)
It has been tar'd twice.
bandit12@bandit:/tmp/j$ tar -xvf data5.bin
data6.bin
bandit12@bandit:/tmp/j$ file data6.bin
data6.bin: bzip2 compressed data, block size = 900k
bzip2 again.
bandit12@bandit:/tmp/j$ mv data6.bin data6.bin.bz2
bandit12@bandit:/tmp/j$ bzip2 -d data6.bin.bz2
bandit12@bandit:/tmp/j$ file data6.bin
data6.bin: POSIX tar archive (GNU)
tar again.
bandit12@bandit:/tmp/j$ tar -xvf data6.bin
data8.bin
bandit12@bandit:/tmp/j$ file data8.bin
data8.bin: gzip compressed data, was "data9.bin", last modified: Thu May 7 18:14:30 2020, max compression, from Unix
gzip again.
bandit12@bandit:/tmp/j$ mv data8.bin data9.bin.gz
bandit12@bandit:/tmp/j$ gzip -d data9.bin.gz
bandit12@bandit:/tmp/j$ file data9.bin
data9.bin: ASCII text
Yay, ASCII, so we can take a look at it.
bandit12@bandit:/tmp/j$ cat data9.bin
The password is 8ZjyCRiBWFYkneahHwxCv3wb2a1ORpYL

Level 13->14

The goal here is to ssh in using an RSA private key, from the level 13 account.
ssh bandit13@bandit.labs.overthewire.org -p 2220
password: 8ZjyCRiBWFYkneahHwxCv3wb2a1ORpYL
First, let's take a look at the permissions on the bandit14 password.

```
bandit13@bandit:~$ cd /etc/bandit_pass
bandit13@bandit:/etc/bandit_pass$ ls
bandit0   bandit12  bandit16  bandit2   bandit23  bandit27  bandit30  bandit4
bandit8
bandit1   bandit13  bandit17  bandit20  bandit24  bandit28  bandit31  bandit5
bandit9
bandit10  bandit14  bandit18  bandit21  bandit25  bandit29  bandit32  bandit6
bandit11  bandit15  bandit19  bandit22  bandit26  bandit3   bandit33  bandit7
bandit13@bandit:/etc/bandit_pass$ ls -alps
total 144
4 drwxr-xr-x  2 root     root     4096 May  7  2020 ./
4 drwxr-xr-x 87 root     root     4096 May 14  2020 ../
4 -r--------  1 bandit0  bandit0     8 May  7  2020 bandit0
4 -r--------  1 bandit1  bandit1    33 May  7  2020 bandit1
4 -r--------  1 bandit10 bandit10   33 May  7  2020 bandit10
4 -r--------  1 bandit11 bandit11   33 May  7  2020 bandit11
4 -r--------  1 bandit12 bandit12   33 May  7  2020 bandit12
4 -r--------  1 bandit13 bandit13   33 May  7  2020 bandit13
4 -r--------  1 bandit14 bandit14   33 May  7  2020 bandit14
4 -r--------  1 bandit15 bandit15   33 May  7  2020 bandit15
4 -r--------  1 bandit16 bandit16   33 May  7  2020 bandit16
4 -r--------  1 bandit17 bandit17   33 May  7  2020 bandit17
4 -r--------  1 bandit18 bandit18   33 May  7  2020 bandit18
4 -r--------  1 bandit19 bandit19   33 May  7  2020 bandit19
4 -r--------  1 bandit2  bandit2    33 May  7  2020 bandit2
4 -r--------  1 bandit20 bandit20   33 May  7  2020 bandit20
4 -r--------  1 bandit21 bandit21   33 May  7  2020 bandit21
4 -r--------  1 bandit22 bandit22   33 May  7  2020 bandit22
4 -r--------  1 bandit23 bandit23   33 May  7  2020 bandit23
4 -r--------  1 bandit24 bandit24   33 May  7  2020 bandit24
4 -r--------  1 bandit25 bandit25   33 May  7  2020 bandit25
4 -r--------  1 bandit26 bandit26   33 May  7  2020 bandit26
4 -r--------  1 bandit27 bandit27   33 May  7  2020 bandit27
4 -r--------  1 bandit28 bandit28   33 May  7  2020 bandit28
4 -r--------  1 bandit29 bandit29   33 May  7  2020 bandit29
4 -r--------  1 bandit3  bandit3    33 May  7  2020 bandit3
4 -r--------  1 bandit30 bandit30   33 May  7  2020 bandit30
4 -r--------  1 bandit31 bandit31   33 May  7  2020 bandit31
4 -r--------  1 bandit32 bandit32   33 May  7  2020 bandit32
4 -r--------  1 bandit33 bandit33   33 May  7  2020 bandit33
4 -r--------  1 bandit4  bandit4    33 May  7  2020 bandit4
4 -r--------  1 bandit5  bandit5    33 May  7  2020 bandit5
4 -r--------  1 bandit6  bandit6    33 May  7  2020 bandit6
4 -r--------  1 bandit7  bandit7    33 May  7  2020 bandit7
4 -r--------  1 bandit8  bandit8    33 May  7  2020 bandit8
4 -r--------  1 bandit9  bandit9    33 May  7  2020 bandit9
```

Permissions require us to be logged in as bandit14 to read the file. So, let's go look for the private SSH key.

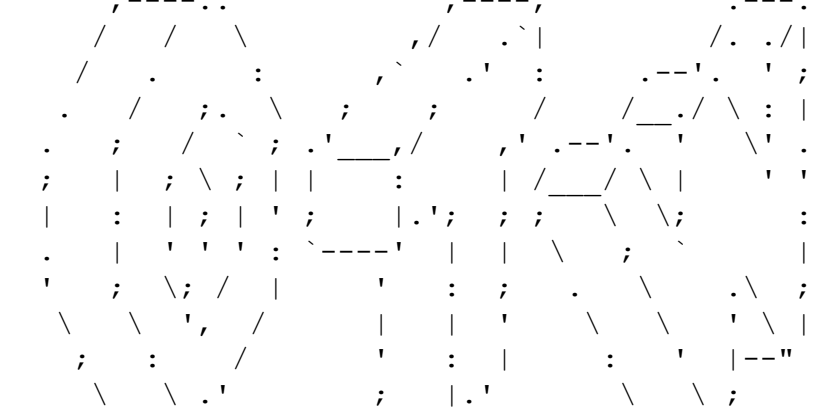
```
bandit13@bandit: cd ~/
bandit13@bandit:~$ ls
sshkey.private
bandit13@bandit:~$ cat sshkey.private
-----BEGIN RSA PRIVATE KEY-----
```

```
MIIEpAIBAAKCAQEAxkkOE83W2cOT7IWhFc9aPaaQmQDdgzuXCv+ppZHa++buSkN+
gg0tcr7Fw8NLGa5+Uzec2rEg0WmeevB13AIoYp0MZyETq46t+jk9puNwZwIt9XgB
ZufGtZEwWbFWw/vVLNwOXBe4UWStGRWzgPpEeSv5Tb1VjLZIBdGphTIK22Amz6Zb
ThMsiMnyJafEwJ/T8PQO3myS91vUHEuoOMAzoUID4kN0MEZ3+XahyK0HJVq68KsV
ObefXG1vvA3GAJ29kxJaqvRfgYnqZryWN7w3CHjNU4c/2Jkp+n8L0SnxaNA+WYA7
jiPyTF0is8uzMlYQ4l1Lzh/8/MpvhCQF8r22dwIDAQABAoIBAQC6dWBjhyEOzjeA
J3j/RWmap9M5zfJ/wb2bfidNpwbB8rsJ4sZIDZQ7XuIh4LfygoAQSS+bBw3RXvzE
pvJt3SmU8hIDuLsCjL1VnBY5pY7Bju8g8aR/3FyjyNAqx/TLfzlLYfOu7i9Jet67
xAh0tONG/u8FB5I3LAI2Vp6OviwvdWeC4nOxCthldpuPKNLA8rmMMVRTKQ+7T2VS
nXmwYckKUcUgzoVSpiNZaS0zUDypdpy2+tRH3MQa5kqN1YKjvF8RC47woOYCktsD
o3FFpGNFec9Taa3Msy+DfQQhHKZFKIL3bJDONtmrVvtYK40/yeU4aZ/HA2DQzwhe
ol1AfiEhAoGBAOnVjosBkm7sblK+n4IEwPxs8sOmhPnTDUy5WGrpSCrXOmsVIBUf
laL3ZGLx3xCIwtCnEucB9DvN2HZkupc/h6hTKUYLqXuyLD8njTrbRhLgbC9QrKrS
M1F2fSTxVqPtZDlDMwjNR04xHA/fKh8bXXyTMqOHNJTHHNhbh3McdURjAoGBANkU
1hqfnw7+aXncJ9bjysr1ZWbqOE5Nd8AFgfwaKuGTTVX2NsUQnCMWdOp+wFak40JH
PKWkJNdBG+ex0H9JNQsTK3X5PBMAS8AfX0GrKeuwKWA6erytVTqjOfLYcdp5+z9s
8DtVCxDuVsM+i4X8UqIGOlvGbtKEVokHPFXP1q/dAoGAcHg5YX7WEehCgCYTzpO+
xysX8ScM2qS6xuZ3MqUWAxUWkh7NGZvhe0sGy9iOdANzwKw7mUUFViaCMR/t54W1
GC83sOs3D7n5Mj8x3NdO8xFit7dT9a245TvaoYQ7KgmqpSg/ScKCw4c3eiLava+J
3btnJeSIU+8ZXq9XjPRpKwUCgYA7z6LiOQKxNeXH3qHXcnHok855maUj5fJNpPbY
iDkyZ8ySF8GlcFsky8Yw6fWCqfG3zDrohJ5l9JmEsBh7SadkwsZhvecQcS9t4vby
9/8X4jS0P8ibfcKS4nBP+dT81kkkg5Z5MohXBORA7VWx+ACohcDEkprsQ+w32xeD
qT1EvQKBgQDKm8ws2ByvSUVs9GjTilCajFqLJ0eVYzRPaY6f++Gv/UVfAPV4c+S0
kAWpXbv5tbkkzbS0eaLPTKgLzavXtQoTtKwrjpolHKIHUz6Wu+n4abfAIRFubOdN
/+aLoRQ0yBDRbdXMsZN/jvY44eM+xRLdRVyMmdPtP8belRi2E2aEzA==
-----END RSA PRIVATE KEY-----
```

Since we have the key already in a file, we can just use it in the ssh command. In this case we are connecting to localhost (which is what the directions told us to do).

```
bandit13@bandit:~$ ssh -i sshkey.private bandit14@localhost
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/known_hosts).
This is a OverTheWire game server. More information on
http://www.overthewire.org/wargames

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



```
  www. `---` ver     '---' he       '---" ire.org


Welcome to OverTheWire!
```
Now that we are in, we can read the bandit14 password file.
```
bandit14@bandit:~$ cat /etc/bandit_pass/bandit14
4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e
```

Level 14->15

In the directions we are told that we need to submit the level 14 password to the localhost on port 30000.

```
ssh bandit14@bandit.labs.overthewire.org -p 2220
password: 4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e
```
First, let's take a look at the status of the common ports on localhost (127.0.0.1).
```
bandit14@bandit:~$ nmap 127.0.0.1
Starting Nmap 7.40 ( https://nmap.org ) at 2022-03-15 16:01 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00036s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
113/tcp   open  ident
30000/tcp open  ndmps
Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```
Port 30000 is open for ndmps service. The directions tell us that we may need telnet to solve this level, so we will start by trying to telnet to port 30000.
```
bandit14@bandit:~$ telnet 127.0.0.1 30000
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
```
Looks like telnet worked and we have a shell. So, let's enter in the level 14 password.
```
4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e
Correct!
```
```
BfMYroe26WYalil77FoDi9qh59eK5xNr
```

Level 15->16

Here we need to use SSL to provide the level 15 password to localhost on port 30001.

```
ssh bandit15@bandit.labs.overthewire.org -p 2220
```
First, let's take a look at the status of common ports.
```
password: BfMYroe26WYalil77FoDi9qh59eK5xNr
bandit15@bandit:~$ nmap 127.0.0.1
Starting Nmap 7.40 ( https://nmap.org ) at 2022-03-15 16:10 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00024s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
```

```
22/tcp    open  ssh
113/tcp   open  ident
30000/tcp open  ndmps
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```
Since 30001 isn't a common port, we will look at it specifically.
```
bandit15@bandit:~$ nmap 127.0.0.1 -p 30001
Starting Nmap 7.40 ( https://nmap.org ) at 2022-03-15 16:10 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00014s latency).
PORT      STATE SERVICE
30001/tcp open  pago-services1
Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds
```
Since the port is open and we know we need to establish an SSL connection, and the instructions tell us that we may need openssl to do this, let's try to establish an openssl connection.
```
bandit15@bandit:~$ openssl s_client -connect 127.0.0.1:30001
CONNECTED(00000003)
depth=0 CN = localhost
verify error:num=18:self signed certificate
verify return:1
depth=0 CN = localhost
verify return:1
---
Certificate chain
 0 s:/CN=localhost
   i:/CN=localhost
---
Server certificate
-----BEGIN CERTIFICATE-----
MIICBjCCAW+gAwIBAgIEXcVbPTANBgkqhkiG9w0BAQUFADAUMRIwEAYDVQQDDAls
b2NhbGhvc3QwHhcNMjIwMzA5MTk0NzQyWhcNMjMwMzA5MTk0NzQyWjAUMRIwEAYD
VQQDDAlsb2NhbGhvc3QwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALDCas6k
DHxTRoxVISHtXOeCwJ8Sax5BZN76Hle8AH6pYTAdv9/FRssWL1xppFAtiGnFvglu
95FJvHEQirY4F0oPBTbtGU2xhzZzkWRL5Yj2C3Q2c99cyh+uWQT7sXPtB8W1osPc
YIo83YkXiArpt28474ZYdl+ohbPtP1oQHBv3AgMBAAGjZTBjMBQGA1UdEQQNMAuC
CWxvY2FsaG9zdDBLBglghkgBhvhCAQ0EPhY8QXV0b21hdGljYWxseSBnZW5lcmF0
ZWQgYnkgTmNhdC4gU2VlIGh0dHBzOi8vbm1hcC5vcmcvbmNhdC8uMA0GCSqGSIb3
DQEBBQUAA4GBAC2693WiK/kXMCauf1fEg5DwuxIfm0saYKiLSceyZo1G4IggqOBO
9JCtvMIV/xRAmYEnPvJmf0JtYv+2fsicaPh9E1GRmU0vGoYDZzA7NTZOgRmHlRKe
ihh/XSGrY7tE1qU+EfizmhcB35iZ7W5INIKlu7oyBWcvk3rI4jtPQeZp
-----END CERTIFICATE-----
subject=/CN=localhost
issuer=/CN=localhost
---
No client certificate CA names sent
Peer signing digest: SHA512
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 1019 bytes and written 269 bytes
Verification error: self signed certificate
---
New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 1024 bit
```

```
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES256-GCM-SHA384
    Session-ID: 5EC9FA893AAFA4F9D590BCB5AAB950CFB1F7A332361138377F67866FC1AC1540
    Session-ID-ctx:
    Master-Key:
90CA5B5CA0C1483DD89C042EBEB65519E63C661662263EC3B1F0F9042C2954CA55C5FA4B39229A0A5
F0E68DC7B74E3C6
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 7200 (seconds)
    TLS session ticket:
    0000 - d3 53 01 d9 4c 2e 82 50-ef 16 38 c5 d8 1b dd f2   .S..L..P..8.....
    0010 - 61 84 b1 83 ef 73 24 85-92 41 ac d0 d4 65 26 ce   a....s$..A...e&.
    0020 - a7 ac c7 53 31 a2 82 4f-2c 38 f6 f5 a8 40 9d 26   ...S1..O,8...@.&
    0030 - f7 cc ab f9 c1 6f 90 2b-45 63 4c 57 95 89 da 40   .....o.+EcLW...@
    0040 - 2f 99 82 95 e8 03 7b f5-64 a0 72 2f 59 e0 6b 66   /.....{.d.r/Y.kf
    0050 - 96 a9 28 6d cc 86 0f a4-b4 cd 98 a0 5f 62 79 40   ..(m........_by@
    0060 - 96 74 cb a2 99 75 e3 c8-79 4e 00 34 70 06 1f 6e   .t...u..yN.4p..n
    0070 - 3f d5 7f 1f de 42 67 7a-a8 a4 b4 fc c9 95 98 1a   ?....Bgz........
    0080 - 42 f5 73 db 62 a7 20 5e-3f 41 88 53 9c 4a e6 bf   B.s.b. ^?A.S.J..
    0090 - e4 5d 15 c9 a7 71 b8 67-dd 63 38 06 11 5b 5e 48   .]...q.g.c8..[^H

    Start Time: 1647357662
    Timeout   : 7200 (sec)
    Verify return code: 18 (self signed certificate)
    Extended master secret: yes
---
```

It worked. And left us here without a command prompt, so this is likely a shell we have logged into. Let's pass it the level 15 password.

```
BfMYroe26WYalil77FoDi9qh59eK5xNr
Correct!
cluFn7wTiGryunymYOu4RcffSxQluehd
closed
bandit15@bandit:~$
```

Level 16->17

The directions tell us to establish an SSL connection on a port in the range of 31000-32000. So, we will plan to use the openssl command again, just need to find the right port to connect to. Below is a link to the openssl cookbook: https://www.feistyduck.com/library/openssl-cookbook/online/ch-testing-with-openssl.html#connecting-to-ssl-services

```
ssh bandit16@bandit.labs.overthewire.org -p 2220
password: cluFn7wTiGryunymYOu4RcffSxQluehd
```

Let's see which ports on local host in our given port range are open.

```
bandit16@bandit:~$ nmap 127.0.0.1 -p 31000-32000
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2022-03-15 16:28 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00028s 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
Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

Of the 5 that are open, our directions tell us that only one of them can establish an SSL connection, so let's enumerate the services on those ports.

```
bandit16@bandit:~$ nmap -sV 127.0.0.1 -p 31046,31518,31790,31960
Starting Nmap 7.40 ( https://nmap.org ) at 2022-03-15 16:35 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00069s latency).
PORT      STATE SERVICE     VERSION
31046/tcp open  echo
31518/tcp open  ssl/echo
31790/tcp open  ssl/unknown
31960/tcp open  echo
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-
bin/submit.cgi?new-service :
SF-Port31790-TCP:V=7.40%T=SSL%I=7%D=3/15%Time=6230B24C%P=x86_64-pc-linux-g
SF:nu%r(GenericLines,31,"Wrong!\x20Please\x20enter\x20the\x20correct\x20cu
SF:rrent\x20password\n")%r(GetRequest,31,"Wrong!\x20Please\x20enter\x20the
SF:\x20correct\x20current\x20password\n")%r(HTTPOptions,31,"Wrong!\x20Plea
SF:se\x20enter\x20the\x20correct\x20current\x20password\n")%r(RTSPRequest,
SF:31,"Wrong!\x20Please\x20enter\x20the\x20correct\x20current\x20password\
SF:n")%r(Help,31,"Wrong!\x20Please\x20enter\x20the\x20correct\x20current\x
SF:20password\n")%r(SSLSessionReq,31,"Wrong!\x20Please\x20enter\x20the\x20
SF:correct\x20current\x20password\n")%r(TLSSessionReq,31,"Wrong!\x20Please
SF:\x20enter\x20the\x20correct\x20current\x20password\n")%r(Kerberos,31,"W
SF:rong!\x20Please\x20enter\x20the\x20correct\x20current\x20password\n")%r
SF:(FourOhFourRequest,31,"Wrong!\x20Please\x20enter\x20the\x20correct\x20c
SF:urrent\x20password\n")%r(LPDString,31,"Wrong!\x20Please\x20enter\x20the
SF:\x20correct\x20current\x20password\n")%r(LDAPSearchReq,31,"Wrong!\x20Pl
SF:ease\x20enter\x20the\x20correct\x20current\x20password\n")%r(SIPOptions
SF:,31,"Wrong!\x20Please\x20enter\x20the\x20correct\x20current\x20password
SF:\n");
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 88.49 seconds
```

So, we see echo as the service for all but one of the ports (31790), so we will attempt to connect to that one using openssl. (As a side note, it looks like I forgot to scan one of the open ports to identify the running service. Oops. Well, in this case we found the port we were looking for anyway).

```
bandit16@bandit:~$ openssl s_client -connect 127.0.0.1:31790
CONNECTED(00000003)
depth=0 CN = localhost
verify error:num=18:self signed certificate
```

```
verify return:1
depth=0 CN = localhost
verify return:1
---
Certificate chain
 0 s:/CN=localhost
   i:/CN=localhost
---
Server certificate
-----BEGIN CERTIFICATE-----
MIICBjCCAW+gAwIBAgIENrkPujANBgkqhkiG9w0BAQUFADAUMRIwEAYDVQQDDAls
b2NhbGhvc3QwHhcNMjIwMzA5MTgyNTAyWhcNMjMwMzA5MTgyNTAyWjAUMRIwEAYD
VQQDDAlsb2NhbGhvc3QwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALiKByZP
lrfe4ehmGKTQcJbvyVtpDIalvmniZhbGIioIDKF4aaJjfgIUU1EVRvKv6tHjcHUx
zyDD0J9h60VYzOFjM6rSga2NT17qTY1V6RhUbxMYf1EBSlfPKK1ygBTv8D45YSDT
n+v6FFNKRoColFotUkheKGHlA4/SKIkqewHZAgMBAAGjZTBjMBQGA1UdEQQNMAuC
CWxvY2FsaG9zdDBLBglghkgBhvhCAQ0EPhY8QXV0b21hdGljYWxseSBnZW5lcmF0
ZWQgYnkgTmNhdC4gU2VlIGh0dHBzOi8vbm1hcC5vcmcvbmNhdC8uMA0GCSqGSIb3
DQEBBQUAA4GBAG5ZI4DpzyiaffMjo5TOFknr5NifZaKRStjvTv7A7FJonW19hUxi
za7DAvYelWcrzCNSlo/DyzqPzjSdmz5NciwDUtkZ0hTWLNbAR7g+BsEY4wxKqPGF
vFxCA2i2q8IRECehwTjewvii0F7HjPZcNZLUgTIEJCp969nbaDyS0KKu
-----END CERTIFICATE-----
subject=/CN=localhost
issuer=/CN=localhost
---
No client certificate CA names sent
Peer signing digest: SHA512
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 1019 bytes and written 269 bytes
Verification error: self signed certificate
---
New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 1024 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES256-GCM-SHA384
    Session-ID: E3E986345A179AFF82F0928C26C23157F9B64B0375D355FA90967CBBC12D1A5A
    Session-ID-ctx:
    Master-Key:
8F624D0B92914591AB8DE7FF8AF04C419B32177AF7DB40B71940E8C4E1F2C6FE7C240A169638E37A6
449079044F5EB2E
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 7200 (seconds)
    TLS session ticket:
    0000 - 0f 76 78 ef f7 7f 4a 33-d1 d0 df fd 48 8c 13 70   .vx...J3....H..p
    0010 - ee 7e 84 e3 65 2d 3e 2a-74 b5 db 5f b1 1f 32 f8   .~..e->*t.._..2.
```

```
    0020 - ee 97 45 82 14 02 d7 14-87 0d 82 05 3d b2 3c f4   ..E.........=.<.
    0030 - 2e ec 39 f5 a5 99 2f 4e-3b d8 6e 39 9c 95 f1 e0   ..9.../N;.n9....
    0040 - b1 1e 74 f2 a1 ec e4 85-71 88 f5 83 d7 5a 53 a1   ..t.....q....ZS.
    0050 - 4a e9 b6 01 b3 39 5d 27-88 b9 98 34 e3 9f 02 53   J....9]'...4...S
    0060 - 53 ae fb 87 e3 c1 bf a2-a0 af 76 0a 95 9a 83 ac   S.........v.....
    0070 - 7e dd 13 b1 01 73 88 1e-00 01 ff 0f cf db c4 48   ~....s.........H
    0080 - 95 a6 8d 11 d0 97 d1 b3-9a 88 82 44 22 50 eb c4   ...........D"P..
    0090 - 93 57 4e 7e 32 f4 31 b2-62 cc ba 15 24 ba e7 21   .WN~2.1.b...$..!

    Start Time: 1647358752
    Timeout   : 7200 (sec)
    Verify return code: 18 (self signed certificate)
    Extended master secret: yes
---
```

So, here we are. Looks like the connection worked and we have a shell. So, let's pass it the level 16 password.

```
cluFn7wTiGryunymYOu4RcffSxQluehd
Correct!
-----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-----
closed
bandit16@bandit:~$
```

It gave us an RSA key. Somehow we need to get this into a file so we can use it to log into the next level. OTW would not let me create a file and save it, so I had to switch to the Ubuntu app and touch/nano a file called priv.key to save the RSA Private Key in, which is done now. So, we can use the new priv.key to log into. …at the end of each level, I always prove that I can log into the next level, so let's try it out.

```
XXXXXX@Cyber-PC:~$ ssh -i priv.key bandit17@bandit.labs.overthewire.org -p 2220
This is a OverTheWire game server. More information on
http://www.overthewire.org/wargames

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'priv.key' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "priv.key": bad permissions
```
It says our permissions are ‘too open’ so let’s look at the permissions.
```
XXXXXX@Cyber-PC:~$ ls -alps
total XX
 0 drwxr-xr-x 1 jminner jminner   512 Mar 15 09:48 ./
 0 drwxr-xr-x 1 root    root      512 Jul 29  2021 ../
 4 -rw-r--r-- 1 jminner jminner  1675 Mar 15 09:48 priv.key
```
Let’s lock down the permissions so that only the user can read them.
```
XXXXXX@Cyber-PC:~$ chmod 400 priv.key
XXXXXX@Cyber-PC:~$ ls -alps
total XX
 0 drwxr-xr-x 1 jminner jminner   512 Mar 15 09:48 ./
 0 drwxr-xr-x 1 root    root      512 Jul 29  2021 ../
 4 -r-------- 1 jminner jminner  1675 Mar 15 09:48 priv.key
```
Now that permissions are as locked down as we can make them, let’s try to log on as bandit17 again.
```
XXXXXX@Cyber-PC:~$ ssh -i priv.key bandit17@bandit.labs.overthewire.org -p 2220
This is a OverTheWire game server. More information on
http://www.overthewire.org/wargames
Linux bandit.otw.local 5.4.8 x86_64 GNU/Linux

      ,----..            ,----,          .---.
     /   /   \         ,/   .`|         /. ./|
    /   .     :      ,`    .'  :     .--'.  ' ;
   .   /   ;.  \   ;     ;     /    /__./ \ : |
  .   ;   /  ` ; .'___,/    ,' .--'.   '   \' .
  ;   |  ; \ ; | |    :     | /___/ \ |    ' '
  |   :  | ; | ' ;    |.';  ; ;   \  \;      :
  .   |  ' ' ' : `----'  |  |  \   ;  `      |
  '   ;  \; /  |     '   :  ;   .   \    .\  ;
   \   \  ',  /      |   |  '    \   \   ' \ |
    ;   :    /       '   :  |     :   '  |--"
     \   \ .'        ;   |.'       \   \ ;
  www. `---` ver     '---' he       '---" ire.org
Welcome to OverTheWire!
bandit17@bandit:~$
```

Level 17->18

Our instructions say we are given two files and need to find the difference between the two.

```
ssh -i priv.key bandit16@bandit.labs.overthewire.org -p 2220
```

```
bandit17@bandit:~$ ls
passwords.new  passwords.old
```
Let's use the diff command (diff [file1] [file2]). When using diff, the order of the files matters, the diff is what needs to be changed in file1 to make it match file2. Since we are told that the correct password is in passwords.new, this should be file1.
```
bandit17@bandit:~$ diff passwords.new passwords.old
42c42
< kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd
---
> w0Yfolrc5bwjS4qw5mq1nnQi6mF03bii
```
We can read this output to say that we need to remove kfB... from passwords.new (file1) to make it match file2. So, kfB... is the different line in password.new and is our password for the next level.

Level 18->19

We are given a warning that if we log in correctly, we will be immediately kicked out…and we are.
```
ssh bandit18@bandit.labs.overthewire.org -p 2220
password: kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd
```
Someone has manipulated the .bashrc file for bandit18 (the instructions say). So, let's log back in as bandit17 to get an idea of what other shells are available for logging in.
```
bandit17@bandit:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
/usr/bin/tmux
/usr/bin/showtext
bandit17@bandit:~$
```
We will start at the top and try logging in with the /bin/sh shell.
In order to force ssh to accept a new shell we will use the -t switch.
```
C:\Users\XXXXXX>ssh bandit18@bandit.labs.overthewire.org -p 2220 -t "/bin/sh"
This is a OverTheWire game server. More information on
http://www.overthewire.org/wargames
bandit18@bandit.labs.overthewire.org's password:
kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd
```
And we are in.
```
$ ls
readme
$ cat readme
IueksS7Ubh8G3DCwVzrTd8rAVOwq3M5x
```

Level 19->20

Here we will be exploring the SETUID binary.
```
ssh bandit19@bandit.labs.overthewire.org -p 2220
```

password: IueksS7Ubh8G3DCwVzrTd8rAVOwq3M5x

The first thing I confirmed is that only bandit20 can cat /etc/bandit_pass/bandit20. So, since we know we are looking for SETUID binaries, let's search for any of them that exist.

```
bandit19@bandit:~$ find / -perm -u=s -type f 2>/dev/null
/home/bandit19/bandit20-do
/home/bandit20/suconnect
/home/bandit32/uppershell
/home/bandit26/bandit27-do
/run/lock/find
/run/lock/hola
```

bandit20-do looks like the binary that I want to use. It has the same format as sudo, so I presume that using bandit20-do will let me execute a command with bandit20 permissions.

```
bandit19@bandit:~$ /home/bandit19/bandit20-do cat /etc/bandit_pass/bandit20
GbKksEFF4yrVs6il55v6gwY5aVje5f0j
```

Level 20->21

ssh bandit20@bandit.labs.overthewire.org -p 2220
password: GbKksEFF4yrVs6il55v6gwY5aVje5f0j