

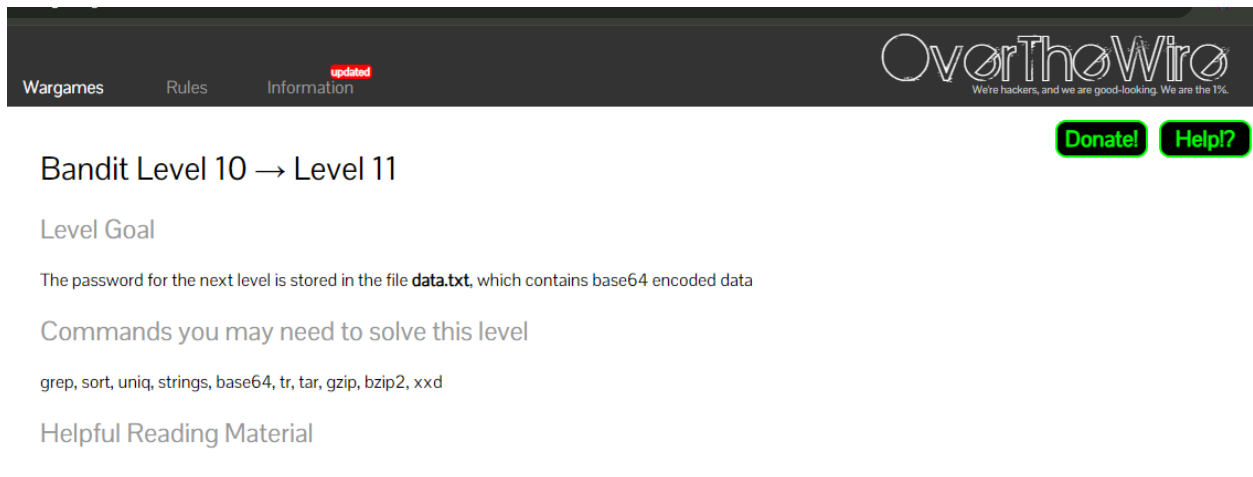
## OverTheWire – Bandit:

### Level 10-20

Today, I will play a war-game called **Bandit**. It has 34 levels. In this write-up I will play level 10-20.

The main objective is to access password files which will help us login into the next levels.

### Level 10-11:



Wargames Rules <sup>updated</sup> Information

OverTheWire  
We're hackers, and we are good-looking. We are the 1%.

Donate! Help!

### Bandit Level 10 → Level 11

#### Level Goal

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

#### Commands you may need to solve this level

grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd

#### Helpful Reading Material

Here we are hinted that the password is in file named **data.txt** and it contains base64 encoded data so we will use the cat command to read **data.txt** and then piping “|” and then the **base64** command with the decode parameter.

```
bandit10@bandit:~$ cat data.txt | base64 --decode
The password is dtR173fZKb0RRsDFSGsg2RWnpNVj3qRr
bandit10@bandit:~$
```

And yes we got the password for bandit 11 and now by using the ssh command we can access bandit11 shell:

```
Enjoy your stay!
bandit11@bandit:~$
```

## Level 11-12:

[Wargames](#) [Rules](#) [Information](#) [updated](#)

OverTheWire  
We're hackers, and we are good-looking. We are the 1%.

[Donate!](#) [Help!?](#)

Bandit Level 11 → Level 12

Level Goal

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

Commands you may need to solve this level

grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd

Now we are hinted that password is in **data.txt** and all uppercase and lowercase letters are rotated by 13 positions. I googled about ROT13 encryption and found that here we can use the translate (`tr`) command. Now we will use first translate from n-z and a-m because `tr` command will not be able to translate after Z.

```
bandit11@bandit:~$ cat data.txt | tr a-zA-Z n-za-mN-ZA-M
The password is 7×16WNeHIi5YkIhWsFfIqoognUTyj9Q4
bandit11@bandit:~$
```

And yes!! We got the password for level 12 and now we can access bandit12 shell:

```
Enjoy your stay!
bandit12@bandit:~$
```

## Level 12-13:

[Wargames](#) [Rules](#) [Information](#) [updated](#)

OverTheWire  
We're hackers, and we are good-looking. We are the 1%.

[Donate!](#) [Help!?](#)

Bandit Level 12 → Level 13

Level Goal

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. Use `mkdir` with a hard to guess directory name. Or better, use the command `"mktemp -d"`. Then copy the datafile using `cp`, and rename it using `mv` (read the manpages!)

Commands you may need to solve this level

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

We are hinted that the file containing the password is in the form of a hex dump. So let's read the file named **data.txt** by using the `cat` command:

```
bandit12@bandit:~$ cat data.txt
00000000: 1f8b 0808 d2e9 9766 0203 6461 7461 322e .....f..data2.
00000010: 6269 6e00 0141 02be fd42 5a68 3931 4159 bin..A...BZh91AY
00000020: 2653 59ea 2468 ae00 0017 7fff dadb b7fb &SY.$h.....
00000030: dbff 5ffb f3fb d776 3d6f fffb dbea fdbd .._....v=0.....
00000040: 85db edfc ffa9 7def faaf efd6 b001 386c .....}.....8l
00000050: 1001 a0d0 6d40 01a0 1a00 0006 8006 8000 ....m@.....
00000060: 0000 d034 01a1 a34d 0034 3d43 40d0 0d34 ...4...M.4=C@..4
00000070: d034 34da 9ea1 b49e a7a8 f29e 5106 4326 .44.....Q.C&
00000080: 9a19 1934 d1a0 341a 6234 d018 d468 6834 ...4..4.b4...hh4
00000090: 00c9 a308 6434 0000 0308 d068 0680 1900 ....d4....h....
000000a0: 0034 d068 1a34 d068 c3a7 a41a 0c9a 0d34 .4.h.4.h.....4
000000b0: 641a 0646 8346 4003 4d34 1a68 6806 9a06 d..F.F@.M4.hh...
000000c0: 9a64 d064 001a 0681 a343 10d0 d00d 1840 .d.d....C.....@
000000d0: 01a3 21a0 68c9 a050 008a 0009 619a 9541 ..!.h..P....a..A
000000e0: 25d5 8bc0 0ff3 e679 7fd0 31b2 c784 e7f7 %.....y..1....
```

So here we can see that its not readable, also we are hinted that it is repeatedly compressed so now to decompress we need to create a directory with read and write permissions,

So now I am going to create a directory inside the tmp directory. I am going to name it Harrum ☺ .And then I will copy the file(data.txt) into the newly created directory and we will traverse into the newly made directory:

```
bandit12@bandit:~$ mkdir /tmp/harrum
bandit12@bandit:~$ cp data.txt /tmp/harrum
bandit12@bandit:~$ cd /tmp/harrum
bandit12@bandit:/tmp/harrum$
```

Now to check the file type we will use the file command:

```
bandit12@bandit:/tmp/harrum$ file data.txt
data.txt: ASCII text
```

So here it shows that the file is in ASCII text. So now I am going to use the xxd command to change it into hexdump. Now I will use the r parameter to retrieve the process and I will give the file a new name "data1"

```
bandit12@bandit:/tmp/harrum$ xxd -r data.txt data1
```

Now lets check the new file:

```
bandit12@bandit:/tmp/harrum$ file data1
data1: gzip compressed data, was "data2.bin", last modified: Wed Jul 17 15:57:06 2024, max compression, from Unix, original size modulo 2^32
577
bandit12@bandit:/tmp/harrum$
```

This tells us that it is a gzip compressed file.

Now to decompress first, i renamed the file and provided it with a proper gzip extension. We are going to use the move command for this. We renamed the file as data2.gz. Now using the gzip command and -d parameter, we decompress the file.

## *Harrum Fatima*

```
bandit12@bandit:/tmp/harrum$ mv data1 data2.gz
bandit12@bandit:/tmp/harrum$ gzip -d data2.gz
```

Now I will check the retrieved file:

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

This tells us that it's a compressed file. To decompress first, we need to rename the file and provide it with a proper bzip2 extension. I am going to use the move command for this. I renamed the file as data3.bz2. Now using the bzip2 command and -d parameter, we will decompress the file.

```
bandit12@bandit:/tmp/harrum$ mv data2 data3.bz2
bandit12@bandit:/tmp/harrum$ bzip2 -d data3.bz2
```

Now I will check the retrieved file again:

```
bandit12@bandit:/tmp/harrum$ file data3
data3: gzip compressed data, was "data4.bin", last modified: Wed Jul 17 15:57:06 2024, max compression, from Unix, original size modulo 2^32
20480
```

Now decompress first, i need to rename the file and provide it with a proper gzip extension. We are going to use the move command for this. So i renamed the file as data4.gz. Now using the gzip command and -d parameter, I will decompress the file.

```
20480
bandit12@bandit:/tmp/harrum$ mv data3 data4.gz
bandit12@bandit:/tmp/harrum$ gzip -d data4.gz
```

Now lets check the retrieved file again:

```
bandit12@bandit:/tmp/harrum$ file data4
data4: POSIX tar archive (GNU)
```

This tells us that it is a tar archive file. Now to extract i will use the tar command with xvf parameters.

```
bandit12@bandit:/tmp/harrum$ tar -xvf data4
data5.bin
```

This gives us a file named data5.bin. Now I will check the retrieved file again:

```
bandit12@bandit:/tmp/harrum$ file data5.bin
data5.bin: POSIX tar archive (GNU)
```

This tells us that it is a tar archive file. Now to extract i will use the tar command with xvf parameters.

```
bandit12@bandit:/tmp/harrum$ tar -xvf data5.bin
data6.bin
bandit12@bandit:/tmp/harrum$
```

This gives us a file named data6.bin. Now lets check the retrieved file again:

```
bandit12@bandit:/tmp/harrum$ file data6.bin
data6.bin: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/harrum$
```

## *Harrum Fatima*

This tells us that it is a bzip2 compressed file. Now decompress first, i renamed the file and provide it with a proper bzip2 extension. I am going to use the move command for this. So i renamed the file as data7.bz2. Now using the bzip2 command and -d parameter, we decompress the file.

```
bandit12@bandit:/tmp/harrum$ mv data6.bin data7.bz2
bandit12@bandit:/tmp/harrum$ bzip2 -d data7.bz2
bandit12@bandit:/tmp/harrum$
```

Now lets check the retrieved file again:

```
bandit12@bandit:/tmp/harrum$ file data7
data7: POSIX tar archive (GNU)
```

This tells us that it is a tar archive file. Now to extract i will use the tar command with xvf parameters.

```
bandit12@bandit:/tmp/harrum$ tar -xvf data7
data8.bin
bandit12@bandit:/tmp/harrum$
```

This give us data8.bin. Now lets check this file:

```
bandit12@bandit:/tmp/harrum$ file data8.bin
data8.bin: gzip compressed data, was "data9.bin", last modified: Wed Jul 17 15:57:06 2024, max compression, from Unix, original size modulo 2^32 49
bandit12@bandit:/tmp/harrum$
```

This tells us that it is a gzip compressed file. Now decompress first, we need to rename the file and provide it with a proper gzip extension. We are going to use the move command for this. We renamed the file as data9.gz. Now using the gzip command and -d parameter, i decompressed the file.

```
bandit12@bandit:/tmp/harrum$ mv data8.bin data9.gz
bandit12@bandit:/tmp/harrum$ gzip -d data9.gz
```

Now let us again check the retrieved file:

```
bandit12@bandit:/tmp/harrum$ file data9
data9: ASCII text
```

So, finally we got a readable file, Now I will use the cat command to read the file:

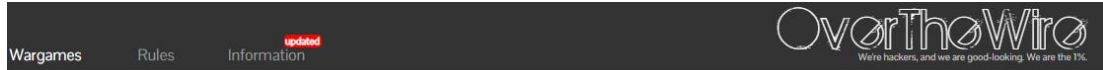
```
bandit12@bandit:/tmp/harrum$ cat data9
The password is F05dwFsc0cbaIiH0h8J2eUks2vdTDwAn
bandit12@bandit:/tmp/harrum$
```

ANddddddd yesss!!!! Finallyyyy we got the password for bandit level 13, Now I can access bandit13 shell:

```
Enjoy your stay!
bandit13@bandit:~$
```

And here we are in bandit13 shell.

## Level 13-14:



### Bandit Level 13 → Level 14

#### Level Goal

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

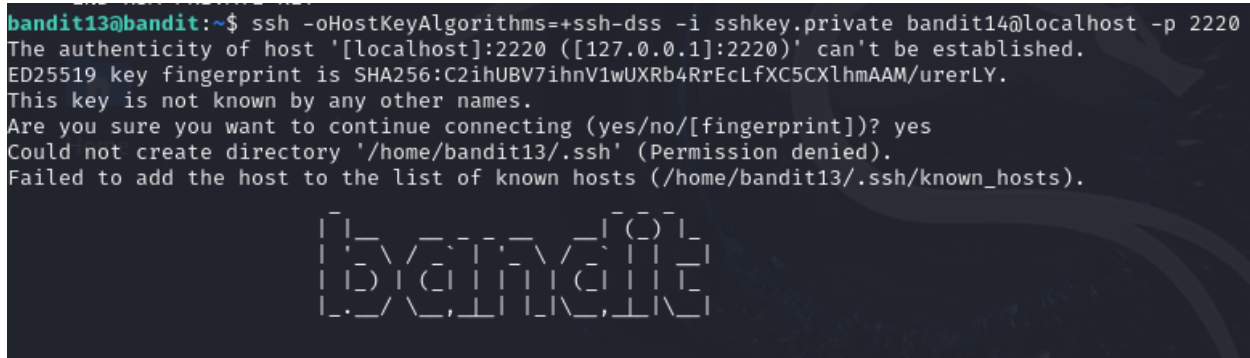
#### Commands you may need to solve this level

ssh, telnet, nc, openssh, s\_client, nmap

Now in this level we are hinted that we will not get the password infact we are given with a private ssh key and we have to use that key. So I will find that key first by using ls command:

```
bandit13@bandit:~$ ls
sshkey.private
bandit13@bandit:~$
```

So here is the private key, now we can use it to get an ssh connection for bandit14.



And here we are in bandit14 shell:

```
bandit14@bandit:~$
```

## Level 14-15:



### Bandit Level 14 → Level 15

#### Level Goal

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

#### Commands you may need to solve this level

ssh, telnet, nc, openssh, s\_client, nmap

## *Harrum Fatima*

We are hinted that the password for next level can be retrieved by submitting the password of current level to port 30000 on local host.

So firstly to retrieve the password of current level I will use the cat command:

```
bandit14@bandit:~$ cat /etc/bandit_pass/bandit14
MU4VWeTyJk8R0of1qqmcBPALh7lDCPvS
bandit14@bandit:~$
```

So here we have the password for current level, now to submit this to port 30000 I will use the telnet command:

```
bandit14@bandit:~$ telnet localhost 30000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
MU4VWeTyJk8R0of1qqmcBPALh7lDCPvS
Correct!
8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo

Connection closed by foreign host.
bandit14@bandit:~$
```

So we entered the password for current level and after checking each character it shows the password for bandit15, now by using ssh command we can access bandit15 shell

```
Enjoy your stay!
```

```
bandit15@bandit:~$
```

## Level 15-16:



Bandit Level 15 → Level 16

### Level Goal

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 in the manpage. Next to 'R' and 'Q', the 'B' command also works in this version of that command...

### Commands you may need to solve this level

ssh, telnet, nc, openssl, s\_client, nmap

We are hinted that we have to submit the password on **port 30001 on localhost** using **SSL encryption**. So I am going to use the open ssl command with parameters `s_client` which means we are connecting as a client, I will use the hostname as **localhost** on port **30001**. Here I will also use the `-ign_eof` to avoid shutting down the connection when the end of file is reached.

```
bandit15@bandit:~$ openssl s_client -connect localhost:30001 -ign_eof
CONNECTED(000000003)
Can't use SSL_get_servername
depth=0 CN = SnakeOil
verify error:num=18:self-signed certificate
verify return:1
depth=0 CN = SnakeOil
verify return:1
---
Certificate chain
 0 s:CN = SnakeOil
  i:CN = SnakeOil
  a:PKEY: rsaEncryption, 4096 (bit); sigalg: RSA-SHA256
  v:NotBefore: Jun 10 03:59:50 2024 GMT; NotAfter: Jun  8 03:59:50 2034 GMT
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIEBzCCAuu+gAwIBAgIUBLz7DBxA0IfoiaL/WaJzE6Sbz7cwDOYJKoZIhvcNAQEL

```

After the connection is established we will enter the password of bandit 15 and then it will display the password for bandit 16:


```
Max Early Data: 0
---
read R BLOCK
8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo
Correct!
kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx
closed
bandit15@bandit:~$
```

And now by using ssh we can access bandit16 shell:

```
Enjoy your stay!
bandit16@bandit:~$
```

## Level 16-17:

[Wargames](#) [Rules](#) [Information](#) [updated](#)

 Owens  
We're hackers, and we are good-looking. We are the 1%.

[Donate!](#) [Help?](#)

### Bandit Level 16 → Level 17

#### Level Goal

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.

#### Commands you may need to solve this level

ssh, telnet, nc, openssl, s\_client, nmap

Now we are hinted that we have to submit the password of current level on a port on localhost but we are given with a range **31000-32000**, now first we have to find out which ports have a server listening on them. For this purpose I am going to use nmap scan:



```
File Actions Edit View Help
bandit16@bandit:~$ nmap -T4 -A -n localhost -p 31000-32000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-22 11:11 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00016s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
31046/tcp  open  echo
31518/tcp  open  ssl/echo
|_ ssl-cert: Subject: commonName=SnakeOil
|_ Not valid before: 2024-06-10T03:59:50
|_ Not valid after: 2034-06-08T03:59:50
|_ ssl-date: TLS randomness does not represent time
31691/tcp  open  echo
31790/tcp  open  ssl/unknown
|_ ssl-date: TLS randomness does not represent time
|_ ssl-cert: Subject: commonName=SnakeOil
|_ Not valid before: 2024-06-10T03:59:50
|_ Not valid after: 2034-06-08T03:59:50
|_ fingerprint-strings:
|_   FourOhFourRequest, GenericLines, GetRequest, HTTPOptions, Help, LPDString, RTSPRequest, SIPOptions:
|_   Wrong! Please enter the correct current password.
31960/tcp  open  echo
```

So after the nmap scan we got these results, as we can see on port **31790** it shows a message to enter the password so...I will connect to this port using openssl:

```
bandit16@bandit:~$ cat /etc/bandit_pass/bandit16 | openssl s_client -quiet -connect localhost:31790
Can't use SSL_get_servername
depth=0 CN = SnakeOil
verify error:num=18:self-signed certificate
verify return:1
depth=0 CN = SnakeOil
verify return:1
Correct!
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPI0jon6iWfbp7c3jx34YkYwQUH57SudyJ
imZzeyGC0gtZPGUjUSxiJSWI/oTqexh+cAMTSMlOJf7+BrJObArnxd9Y7YT2bRPQ
Ja6Lzb558YW3FZl870RiO+rW4LCDCNd2lUvLE/GL2GWyuKN0K5iCd5TbtJzEkQTu
DSt2mcNn4rhal+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW30ekePQAZL0VUYbW
JGTi65CxbCnzc/w4+mqQyvmzpwTMAzJTzAzQxNbkR2MBGySxDLrjg0LWN6sK7wNX
x0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XF0JuaQIDAQABAoIBABagppxM1aoLWfvD
KHcj10nqcoBc4oE11aFYQwik7xFW+24pRNUDE6SFth0ar69jp5RLLwD1NhPx3iBl
J9nOM80J0VToum43U0S8YxF8WwhXriYGnc1sskbwpXOUDc9uX4+UESzH2P29ovd
d8WErY0gPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9AlbssgTcCXkMQnPw9nC
YNN6DDP2lbcBrvgT9YCNL6C+ZKufD52y0Q9q0kwFTEQpjtf4uNtJom+asvLpmS8A
vLY9r60wYsVmZhNqB8U7j7LyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama
+TOWWgEcGYEA8JtPxP0GRJ+IQkX262jM3dEIkza8ky5moIwUqYdsx0NxHgRRhORT
8c8hAuRBB2G82so8vUHK/fur850Efc9TncnCY2crpoqsgghfKLxrLgt+qDpfZnx
SatLdt8GfQ85yA7hnWWJ2MxF3NaeSDm75Lsm+tBbAiyC9P2jGRNtMSkCgYEAYpHd
HCctNi/FwjuLhttfX/rHYKhLidZDFYeiE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt
SghaTdcG0Knyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCivGCSx+X3l5SiWg0A
R57hJglezIiVjv3aGwHwvLZvtszK6zV6oXFAu0ECgYAbjo46T4hyP5tJi93V5Hdi
TtieK7xRVxUL+iu7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFMLy9FL2m9oQWCg
R8VdWsk8r9FGLS+9aKcV5PI/WEKlwXinB30hYimtiG2Cg5JCqIZFHxD6MjEG0iu
```

Here we got a private key, so I am going to make a new directory and I will copy this key into that directory:

```
bandit16@bandit:~$ mkdir /tmp/bandit17_ssh
bandit16@bandit:~$ cd /tmp/bandit17_ssh
bandit16@bandit:/tmp/bandit17_ssh$ nano bandit17.private
```

Now I will change the file permissions:

```
bandit16@bandit:/tmp/bandit17_ssh$ chmod 600 bandit17.private
```

And then finally by using ssh command we are in bandit 17 shell:

```
bandit6@bandit:/tmp/bandit7_ssh$ ssh bandit17@localhost -i bandit17.private -p 2220
Warning: Authentication is successful but the host's authenticity has not yet been established.
bandit17 key fingerprint is SHA256:C2ihUBV7jhnVtWUXRb4RpECLFXC5CXlhmAAM/uerLY.
This key is not known by any other names.
```

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

```
Warning: Permanently added the IP address 127.0.0.1 (ssh://127.0.0.1) to the list of known hosts (/home/bandit16/.ssh/known_hosts).
```

```
bandit17@localhost:~$ cd /home/bandit16/.ssh/
bandit17@localhost:~$ cat known_hosts
# Hosts which are never allowed to replace their existing hosts' entries.
[IP] [hostname] [fingerprint] [comment]
```

This is an OverTheWire game server.  
More information on <http://www.overthewire.org/wargames>

```
bandit17@localhost:~$ ssh -p 2220 root@localhost
Warning: Authentication is successful but the host's authenticity has not yet been established.
root@localhost:~$ cat banner
      _____
     |          |
    |   OoOo   |
    |__|_|_|__|
    |          |
    |   OoOo   |
     |          |
    |_____|___|

You are trying to log into this SSH server with a password on port 2220 from localhost. Connecting from localhost is blocked to conserve resources. Please log out and log in again.
```

```
Enjoy your stay!  
bandit17@bandit:~$
```

## Level 17-18:

Vargames

Rules

Information

updated

We're hackers, and we are good-looking. We are the 1%.

Donate!

Help?

## Bandit Level 17 → Level 18

### Level Goal

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

### Commands you may need to solve this level

cat, grep, ls, diff

Firstly I am going to list the files by ls command:

```
bandit17@bandit:~$ ls
passwords.new  passwords.old
bandit17@bandit:~$
```

So here we got the two files. Now we are hinted that the password is the only line that has been changed, so we can use the diff command here:

```
bandit17@bandit:~$ diff passwords.old passwords.new
42c42
< bSrACvJvvBSxEM2SGsV5sn09vc3xgqyp
—
> x2gLTTjFwM0hQ8oWNbMN362QKxfrqGL0
bandit17@bandit:~$
```

So here I got the password. But now on providing the connection was closed.



Here we got the password and now by ssh command we are in bandit19 shell:

```
Enjoy your stay!
bandit19@bandit:~$
```

## Level 19-20:

[Wargames](#) [Rules](#) [Information](#) [updated](#)

OverTheWire

We're hackers, and we are good-looking. We are the 1%.

[Donate!](#) [Help?](#)

### Bandit Level 19 → Level 20

#### Level Goal

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.

As usual I will start with the ls command:

```
bandit19@bandit:~$ ls
bandit20-do
bandit19@bandit:~$
```

Hmmm, found a file, seems like a script!! Now I will try to see the running of this script:

```
bandit19@bandit:~$ ./bandit20-do
Run a command as another user.
Example: ./bandit20-do id
bandit19@bandit:~$
```

So here it shows that the script runs a command as another user. Now we were hinted that the password is stored at **/etc/bandit\_pass/**. So, now I will run the script with the cat command to read the password for the next level.

```
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit20
0qXahG8Zj0VMN9Ghs7iOWsCfZyXOUbY0
bandit19@bandit:~$
```

So,, got the password for bandit20, and by using ssh command I am in bandit20 shell:

```
Enjoy your stay!
bandit20@bandit:~$
```

*Harrum Fatima*