Assignment 6

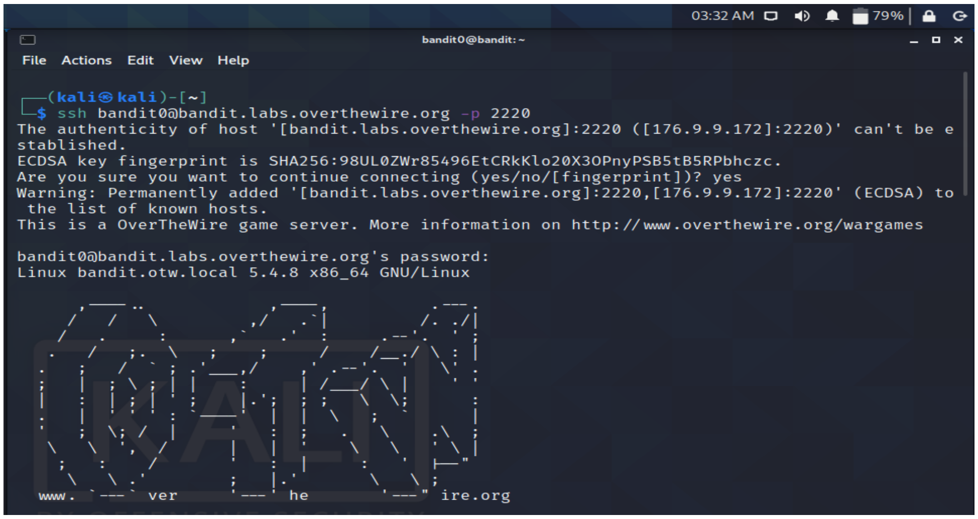
Level 0

This is a pretty simple level. It teaches us to connect to a host using SSH. This is going to teach players the usage of SSH commands.We got the required information from reading the instruction page.

Host: bandit.labs.overthewire.org
Port: 2220
Username: bandit0
Password: bandit0

We used the above information to login using ssh as shown in the given image.

```
ssh bandit0@bandit.labs.overthewire.org -p 2220
```



```
┌──(kali㉿kali)-[~]
└─$ ssh bandit0@bandit.labs.overthewire.org -p 2220
The authenticity of host '[bandit.labs.overthewire.org]:2220 ([176.9.9.172]:2220)' can't be e
stablished.
ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[bandit.labs.overthewire.org]:2220,[176.9.9.172]:2220' (ECDSA) to
 the list of known hosts.
This is a OverTheWire game server. More information on http://www.overthewire.org/wargames

bandit0@bandit.labs.overthewire.org's password:
Linux bandit.otw.local 5.4.8 x86_64 GNU/Linux
```

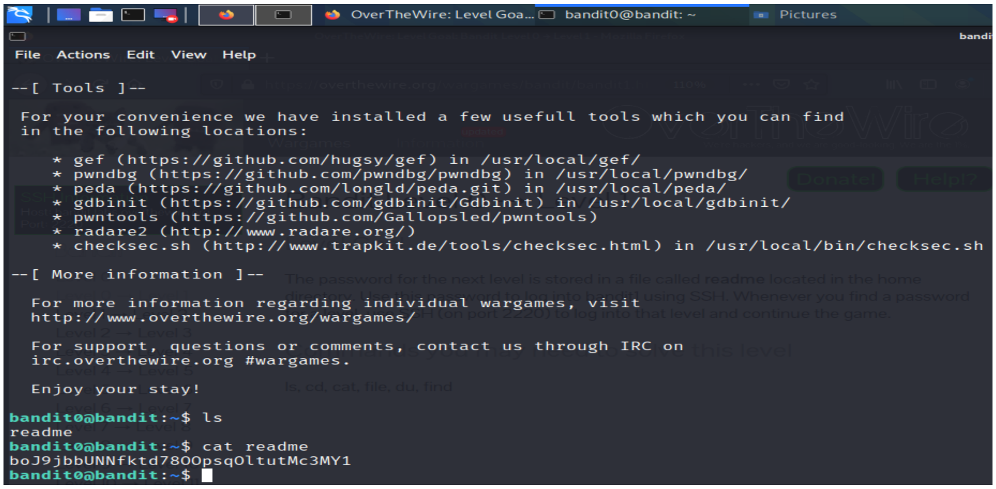
Level 0-1

Now, from the bandit0 shell, we need to find the password for logging as the next user. To find that password, we are going to list files in the directory. Our target is to find a file named readme.

After finding that file, we need to read the password stored inside that file.

We use the ls command to list the files in the current directory. We found the readme file. Now to read the password we will use the cat command. After that, we are going to use the password to login into the next level using SSH.

```
ls -la
cat readme
ssh bandit1@localhost
```



```
--[ Tools ]--

 For your convenience we have installed a few usefull tools which you can find
 in the following locations:

    * gef (https://github.com/hugsy/gef) in /usr/local/gef/
    * pwndbg (https://github.com/pwndbg/pwndbg) in /usr/local/pwndbg/
    * peda (https://github.com/longld/peda.git) in /usr/local/peda/
    * gdbinit (https://github.com/gdbinit/Gdbinit) in /usr/local/gdbinit/
    * pwntools (https://github.com/Gallopsled/pwntools)
    * radare2 (http://www.radare.org/)
    * checksec.sh (http://www.trapkit.de/tools/checksec.html) in /usr/local/bin/checksec.sh

--[ More information ]--

  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!

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

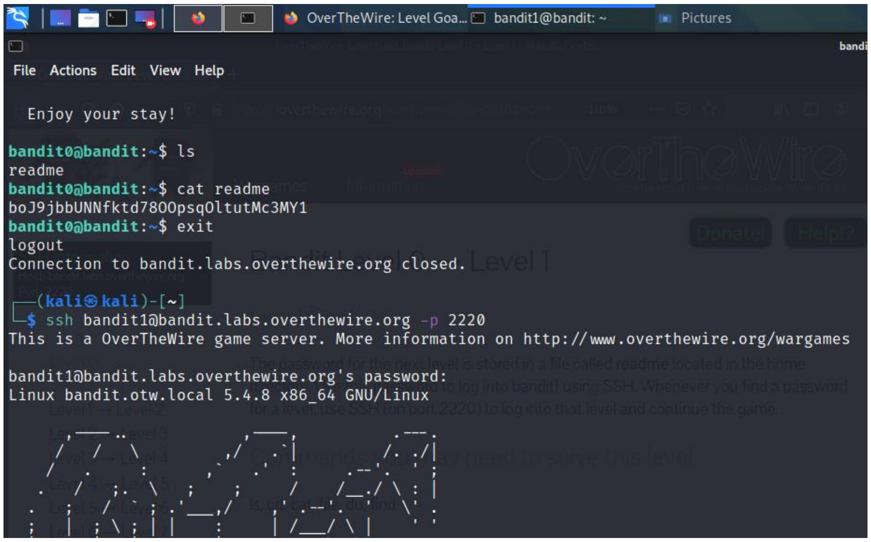
Level 1-2

We are informed that the password for the next level is stored inside a file named -(hyphen). So, to find it we use the ls command. Now comes the part where we have to read the file. As the file is named -(hyphen) we won't be able to read it simply by cat command. As cat command considers -(hyphen) as stdin/Stout. If we directly use cat command, it won't be able to understand that hyphen is a file name. So, we will prefix the command with the path ./, This will help us to read the password stored as shown in the given figure. Since we found the password for the user bandit2. We will use it to get an SSH connection as bandit2.

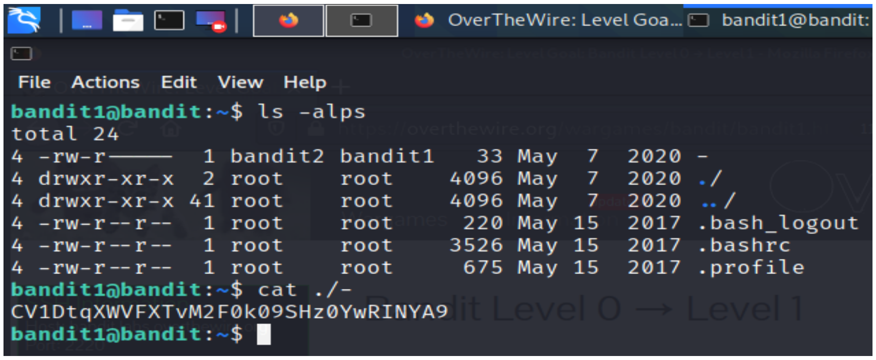
ls

cat ./-

ssh bandit2@localhost



```
 Enjoy your stay!

bandit0@bandit:~$ ls
readme
bandit0@bandit:~$ cat readme
boJ9jbbUNNfktd78OOpsqOltutMc3MY1
bandit0@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.

┌──(kali㉿kali)-[~]
└─$ ssh bandit1@bandit.labs.overthewire.org -p 2220
This is a OverTheWire game server. More information on http://www.overthewire.org/wargames

bandit1@bandit.labs.overthewire.org's password:
Linux bandit.otw.local 5.4.8 x86_64 GNU/Linux
```



```
bandit1@bandit:~$ ls -alps
total 24
4 -rw-r-----  1 bandit2 bandit1   33 May  7  2020 -
4 drwxr-xr-x  2 root    root    4096 May  7  2020 ./
4 drwxr-xr-x 41 root    root    4096 May  7  2020 ../
4 -rw-r--r--  1 root    root     220 May 15  2017 .bash_logout
4 -rw-r--r--  1 root    root    3526 May 15  2017 .bashrc
4 -rw-r--r--  1 root    root     675 May 15  2017 .profile
bandit1@bandit:~$ cat ./-
CV1DtqXWVFXTvM2F0k09SHz0YwRINYA9
bandit1@bandit:~$ 
```

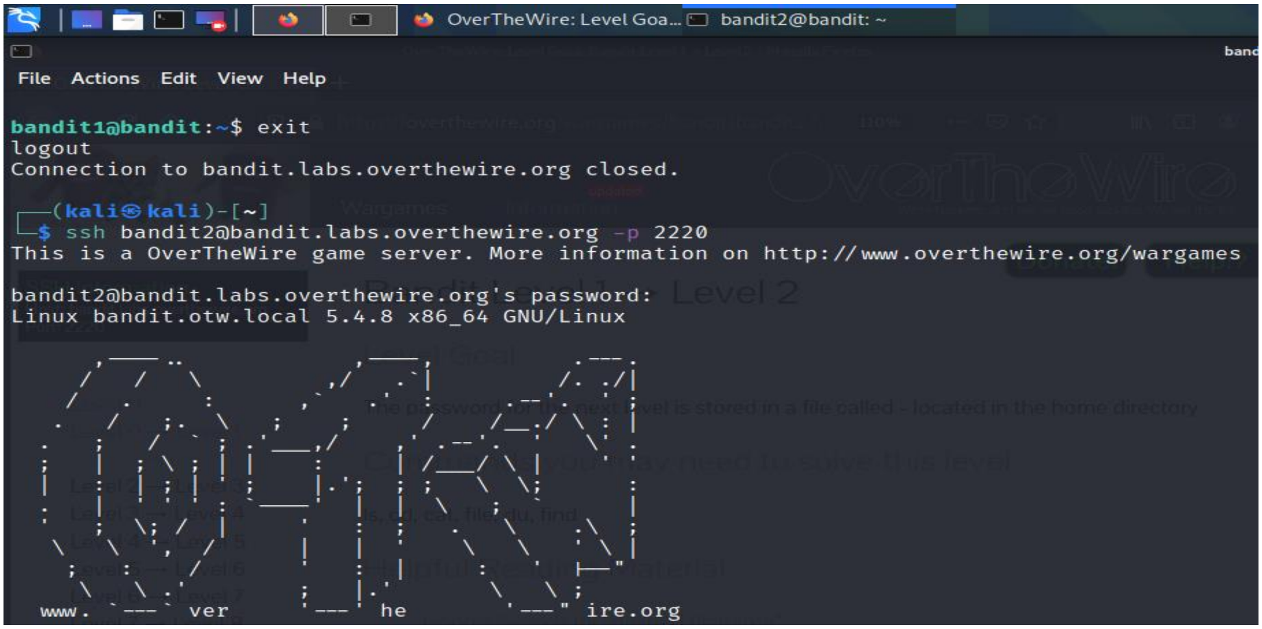
Level 2-3

We are informed that the password for the next level is stored inside a file named spaces in this filename. So, to find it we use the ls command. Now comes the part where we have to read the file. As the file is named spaces in this filename, we won't be able to read it simply by cat command. As cat command reads files name only until space as it considers space as null '/0'. If we directly use cat command, it won't be able to find the file. So, we will write the name of the file in quotes, this will help us to read the password stored as shown in the given figure. Since we found the password for the user bandit3. We will use it to get an SSH connection as bandit3.

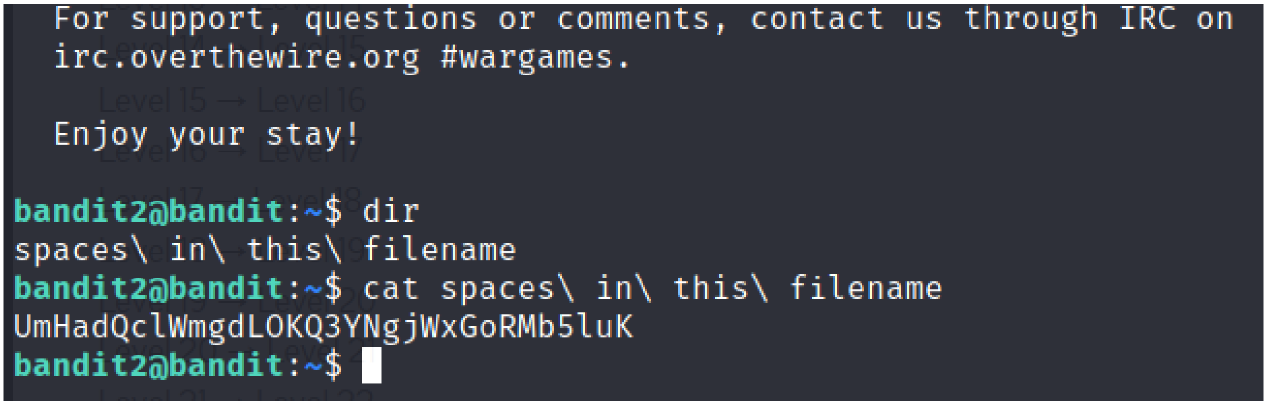
ls

cat 'spaces in this filename'

ssh bandit3@localhost



```
bandit1@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.

┌──(kali㉿kali)-[~]
└─$ ssh bandit2@bandit.labs.overthewire.org -p 2220
This is a OverTheWire game server. More information on http://www.overthewire.org/wargames

bandit2@bandit.labs.overthewire.org's password:
Linux bandit.otw.local 5.4.8 x86_64 GNU/Linux
```

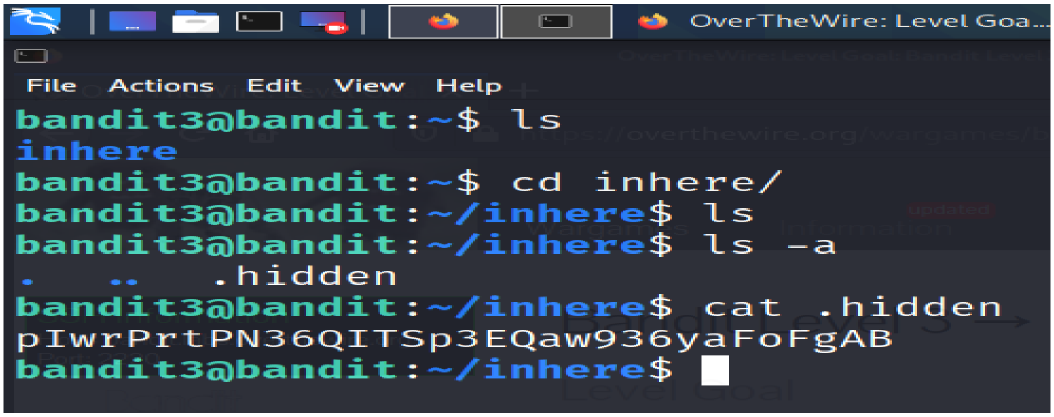


```
  For support, questions or comments, contact us through IRC on
  irc.overthewire.org #wargames.

  Enjoy your stay!

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

Level 3-4

We are informed that the password for the next level is stored inside a directory named inhere. So, to find it we use the ls command. Now, after traversing inside inhere directory we run ls command again. Now it might be the case that the file is hidden. So, we run ls command with -al parameter. It lists all files including the hidden one. And we found the .hidden file. In Linux, the file with a dot(.) in front of the name of the file makes it hidden. Now we would simply use the cat command to read the password stored in the file. Since we found the password for the user bandit4. We will use it to get an SSH connection as bandit4.

ls
cd inhere/
ls
ls -al
cat .hidden
ssh bandit4@localhost

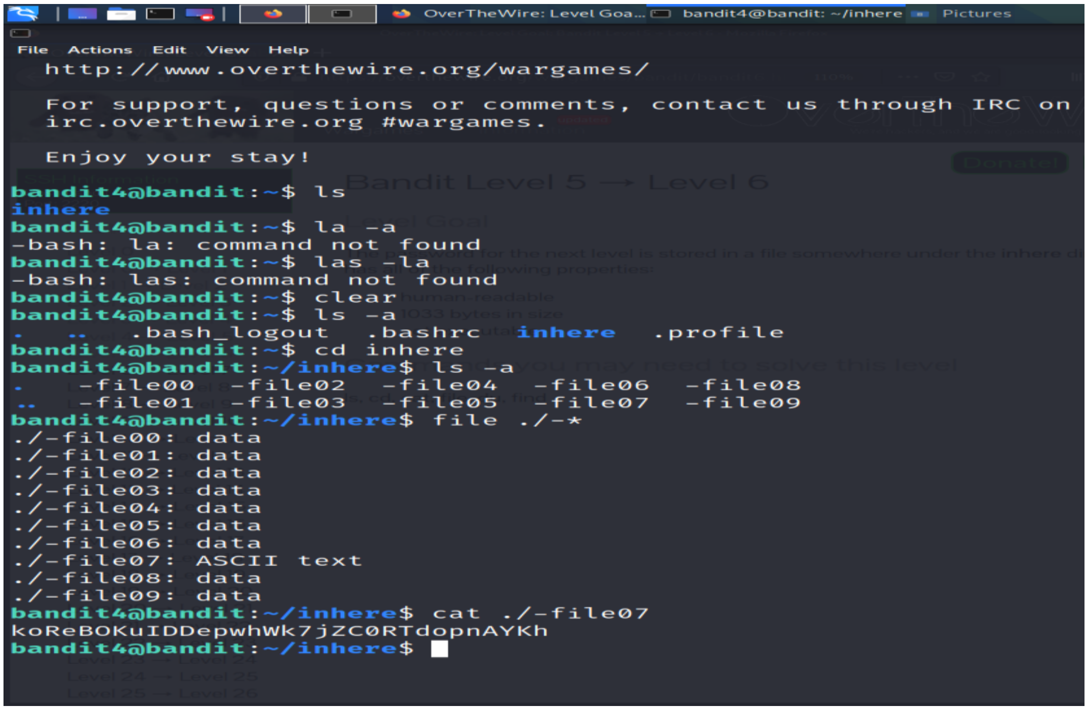


```
bandit3@bandit:~$ ls
inhere
bandit3@bandit:~$ cd inhere/
bandit3@bandit:~/inhere$ ls
bandit3@bandit:~/inhere$ ls -a
.  ..  .hidden
bandit3@bandit:~/inhere$ cat .hidden
pIwrPrtPN36QITSp3EQaw936yaFoFgAB
bandit3@bandit:~/inhere$ 
```

Level 4-5

We are informed that the password for the next level is stored inside a human-readable file. So, to find it we use the ls command. Now, after traversing inside inhere directory we run ls command again. This gives us a bunch of files as shown in the image. We will use the file command to get the information about the files. From files command, we now know that the file07 contains ASCII text. It is mostly readable text. So, let’s read it using cat command. This gives us the password for the next level. We will use it to get an SSH connection as bandit5.

```
ls -la
cd inhere/
ls
file ./*
cat ./-file07
ssh bandit5@localhost
```

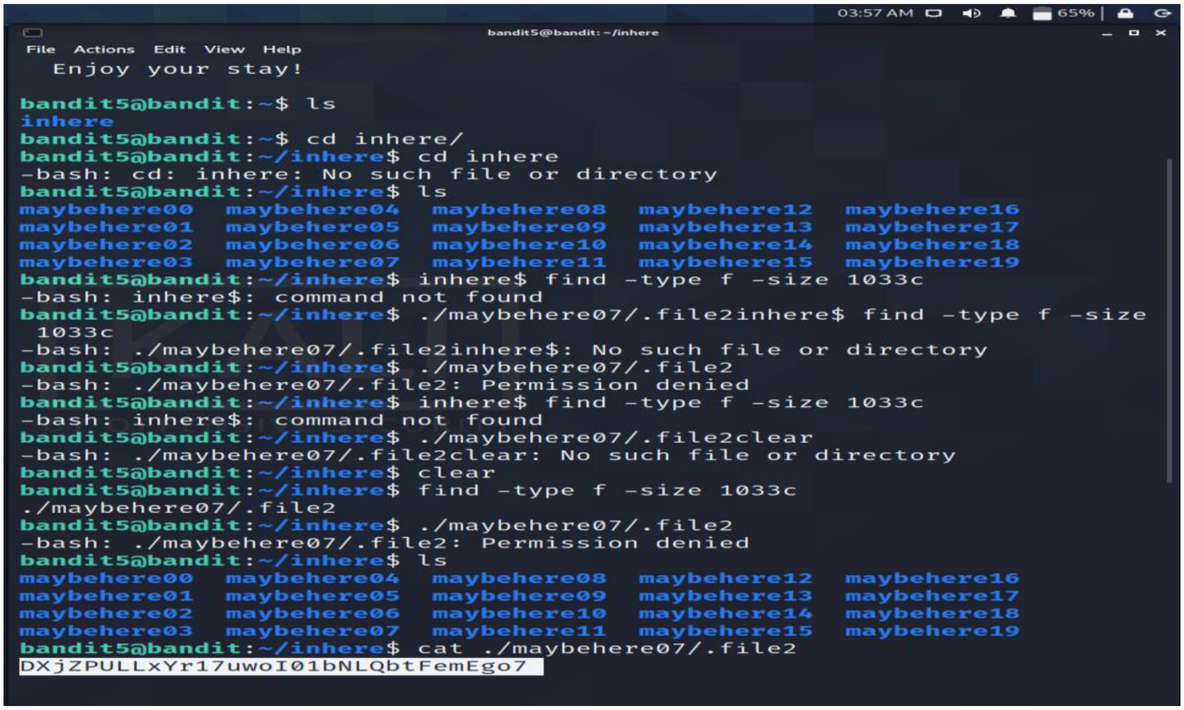


```
  http://www.overthewire.org/wargames/

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

  Enjoy your stay!
bandit4@bandit:~$ ls
inhere
bandit4@bandit:~$ la -a
-bash: la: command not found
bandit4@bandit:~$ las -la
-bash: las: command not found
bandit4@bandit:~$ clear
bandit4@bandit:~$ ls -a
.   ..   .bash_logout   .bashrc   inhere   .profile
bandit4@bandit:~$ cd inhere
bandit4@bandit:~/inhere$ ls -a
.    -file00  -file02  -file04  -file06  -file08
..   -file01  -file03  -file05  -file07  -file09
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$ cat ./-file07
koReBOKuIDDepwhWk7jZC0RTdopnAYKh
bandit4@bandit:~/inhere$ 
```

Level 5-6

We are informed that the password for the next level is stored inside a directory named inhere. So, to find it we use the ls command. Now, after traversing inside inhere directory we run ls command again. This gives us a bunch of files as shown in the image. We will use the file size to find the file. Find command has the parameter of size in which we have to use ‘c’ for depicting size in bytes. From find command, we now know that the file2 contains the password. So, let’s read it using cat command. This gives us the password for the next level. We will use it to get an SSH connection as bandit6.

```
ls
cd inhere/
ls
find . -size 1033c
cat ./maybehere07/.file2
ssh bandit6@localhost
```



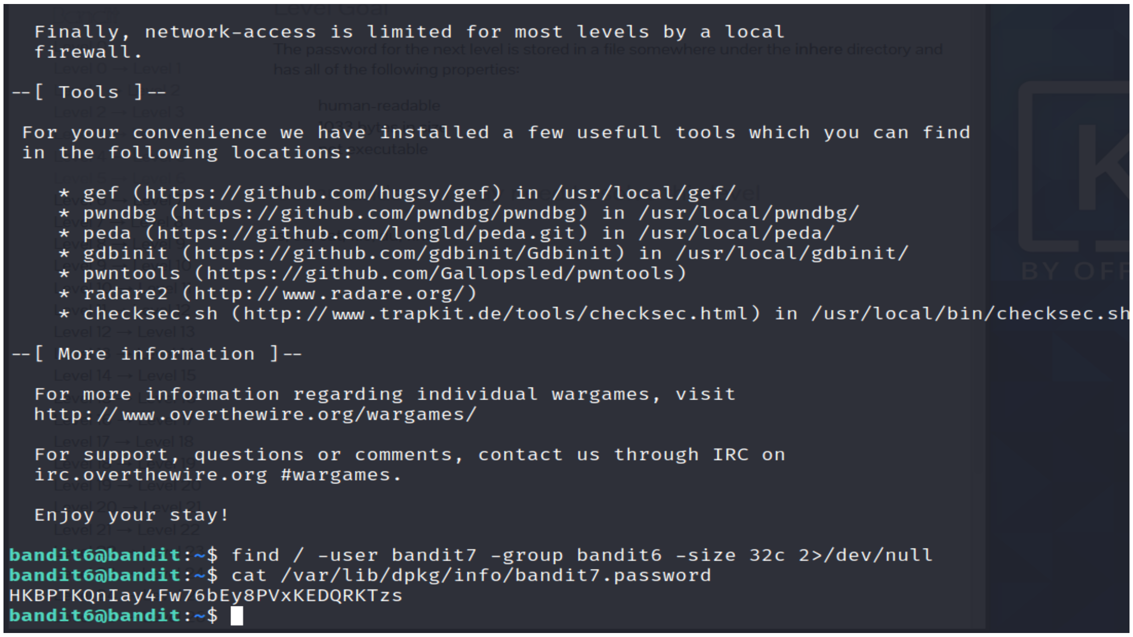
```
  Enjoy your stay!

bandit5@bandit:~$ ls
inhere
bandit5@bandit:~$ cd inhere/
bandit5@bandit:~/inhere$ cd inhere
-bash: cd: inhere: No such file or directory
bandit5@bandit:~/inhere$ ls
maybehere00  maybehere04  maybehere08  maybehere12  maybehere16
maybehere01  maybehere05  maybehere09  maybehere13  maybehere17
maybehere02  maybehere06  maybehere10  maybehere14  maybehere18
maybehere03  maybehere07  maybehere11  maybehere15  maybehere19
bandit5@bandit:~/inhere$ inhere$ find -type f -size 1033c
-bash: inhere$: command not found
bandit5@bandit:~/inhere$ ./maybehere07/.file2inhere$ find -type f -size
 1033c
-bash: ./maybehere07/.file2inhere$: No such file or directory
bandit5@bandit:~/inhere$ ./maybehere07/.file2
-bash: ./maybehere07/.file2: Permission denied
bandit5@bandit:~/inhere$ inhere$ find -type f -size 1033c
-bash: inhere$: command not found
bandit5@bandit:~/inhere$ ./maybehere07/.file2clear
-bash: ./maybehere07/.file2clear: No such file or directory
bandit5@bandit:~/inhere$ clear
bandit5@bandit:~/inhere$ find -type f -size 1033c
./maybehere07/.file2
bandit5@bandit:~/inhere$ ./maybehere07/.file2
-bash: ./maybehere07/.file2: Permission denied
bandit5@bandit:~/inhere$ ls
maybehere00  maybehere04  maybehere08  maybehere12  maybehere16
maybehere01  maybehere05  maybehere09  maybehere13  maybehere17
maybehere02  maybehere06  maybehere10  maybehere14  maybehere18
maybehere03  maybehere07  maybehere11  maybehere15  maybehere19
bandit5@bandit:~/inhere$ cat ./maybehere07/.file2
DXjZPULLxYr17uwoI01bNLQbtFemEgo7
```

Level 6-7

We are informed that the password for the next level is stored somewhere on the server. So, finding the file over the server would be a lot trickier if we are using ls. So, we will try to widen our scope of search using the find command. We are hinted that the user of the file is bandit7 and it is a part of group bandit 6. We will add this information as parameters in the find command. We are given the size too. Let’s add that too. Now as we can see in the given image, we successfully located the password file hidden over the server.

```
find / -user bandit7 -group bandit6 -size 33c
cat /var/lib/dpkg/info/bandit7.password
ssh bandit7@localhost
```

From find command, we now know that the bandit7.password contains the credentials. So, let’s read it using cat command. This gives us the password for the next level. We will use it to get an SSH connection as bandit7.

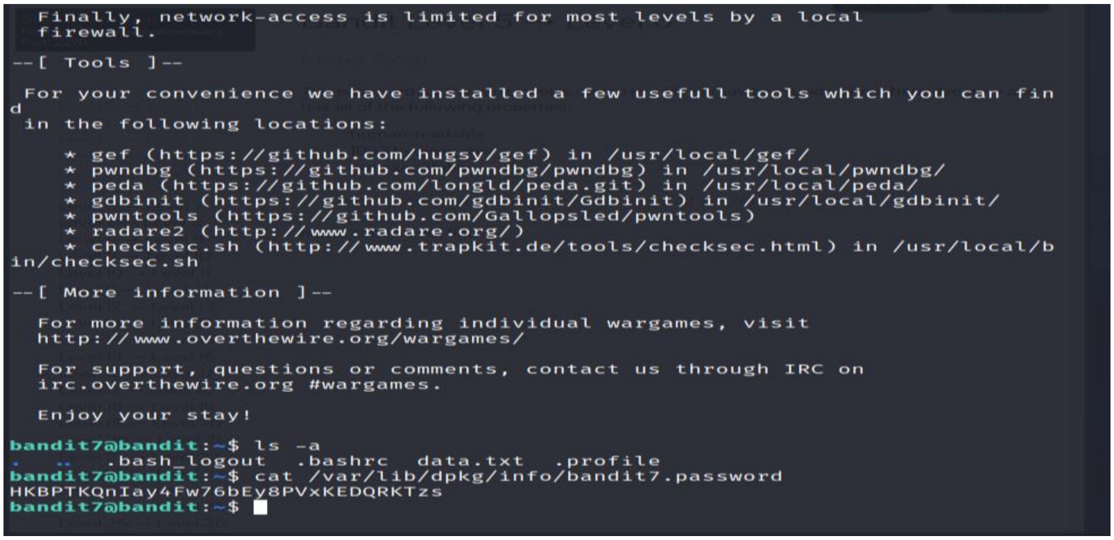


```
  Finally, network-access is limited for most levels by a local
  firewall.

--[ Tools ]--

 For your convenience we have installed a few usefull tools which you can find
 in the following locations:

    * gef (https://github.com/hugsy/gef) in /usr/local/gef/
    * pwndbg (https://github.com/pwndbg/pwndbg) in /usr/local/pwndbg/
    * peda (https://github.com/longld/peda.git) in /usr/local/peda/
    * gdbinit (https://github.com/gdbinit/Gdbinit) in /usr/local/gdbinit/
    * pwntools (https://github.com/Gallopsled/pwntools)
    * radare2 (http://www.radare.org/)
    * checksec.sh (http://www.trapkit.de/tools/checksec.html) in /usr/local/bin/checksec.sh

--[ More information ]--

  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!

bandit6@bandit:~$ find / -user bandit7 -group bandit6 -size 32c 2>/dev/null
bandit6@bandit:~$ cat /var/lib/dpkg/info/bandit7.password
HKBPTKQnIay4Fw76bEy8PVxKEDQRKTzs
bandit6@bandit:~$ █
```

Level 7-8

We are informed that the password for the next level is stored inside a file named data.txt. So, to find it we use the ls command. Now we are hinted that the password is written next to the word millionth in the data.txt file. This means if we find the millionth word, we find the password. We are going to use the grep command for finding millionth. Here we using the (|) Unix pipe. The Pipe connects the standard output from the first command and feeds it as standard input to the second command. In our case, first cat command reads the file and then the data inside the file is sent to grep command to work on. This gives us the password for the next level. We will use it to get an SSH connection as bandit8.

ls
cat data.txt | grep millionth
ssh bandit8@localhost

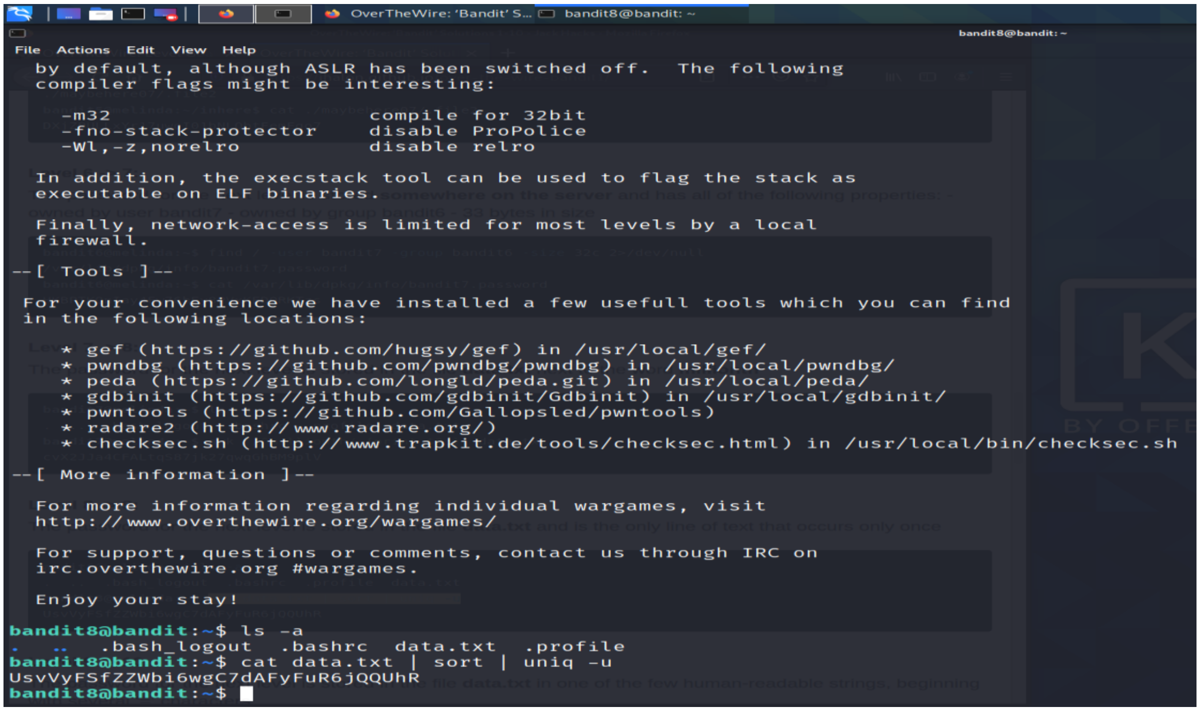


```
  Finally, network-access is limited for most levels by a local
  firewall.

--[ Tools ]--

 For your convenience we have installed a few usefull tools which you can fin
d
 in the following locations:

    * gef (https://github.com/hugsy/gef/) in /usr/local/gef/
    * pwndbg (https://github.com/pwndbg/pwndbg) in /usr/local/pwndbg/
    * peda (https://github.com/longld/peda.git) in /usr/local/peda/
    * gdbinit (https://github.com/gdbinit/Gdbinit) in /usr/local/gdbinit/
    * pwntools (https://github.com/Gallopsled/pwntools)
    * radare2 (http://www.radare.org/)
    * checksec.sh (http://www.trapkit.de/tools/checksec.html) in /usr/local/b
in/checksec.sh

--[ More information ]--

  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!

bandit7@bandit:~$ ls -a
.  ..  .bash_logout  .bashrc  data.txt  .profile
bandit7@bandit:~$ cat /var/lib/dpkg/info/bandit7.password
HKBPTKQnIay4Fw76bEy8PVxKEDQRKTzs
bandit7@bandit:~$ █
```

Level 8-9

We are informed that the password for the next level is stored inside a file named data.txt. It is hinted that the password is the only line of text that occurs only once. Here we are going to use sort command to sort the text inside the data.txt file. But still, the file contains a lot of repeating statements so we will use the uniq command to print the not repeating statement. We are using multiple pipes here to get a filtered result. This gives us the password for the next level. We will use it to get an SSH connection as bandit9.

cat data.txt | sort | uniq -u
ssh bandit9@localhost

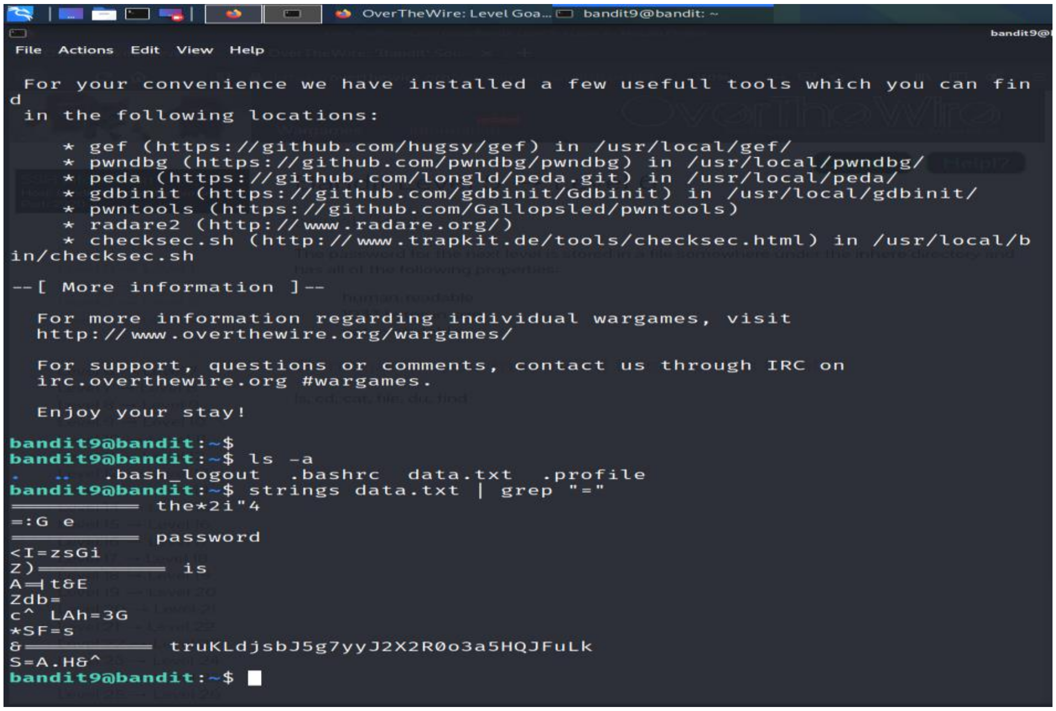


```
  by default, although ASLR has been switched off.  The following
  compiler flags might be interesting:

    -m32                    compile for 32bit
    -fno-stack-protector    disable ProPolice
    -Wl,-z,norelro          disable relro

  In addition, the execstack tool can be used to flag the stack as
  executable on ELF binaries.

  Finally, network-access is limited for most levels by a local
  firewall.

--[ Tools ]--

 For your convenience we have installed a few usefull tools which you can find
 in the following locations:

    * gef (https://github.com/hugsy/gef) in /usr/local/gef/
    * pwndbg (https://github.com/pwndbg/pwndbg) in /usr/local/pwndbg/
    * peda (https://github.com/longld/peda.git) in /usr/local/peda/
    * gdbinit (https://github.com/gdbinit/Gdbinit) in /usr/local/gdbinit/
    * pwntools (https://github.com/Gallopsled/pwntools)
    * radare2 (http://www.radare.org/)
    * checksec.sh (http://www.trapkit.de/tools/checksec.html) in /usr/local/bin/checksec.sh

--[ More information ]--

  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!

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

Level 9-10

We are informed that the password for the next level is stored inside a file named data.txt. We are hinted that the password is followed by several '=' characters. Now if we are to use the cat command our screen would be filled with unreadable mesh. So, to get a more refined approach we are going to use strings command which prints character sequences that are at least 4 characters long. And to get to the exact location of the password, we are going to use grep. This gives us the password for the next level. We will use it to get an SSH connection as bandit10.

```
ls
strings data.txt | grep =
ssh bandit10@localhost
```

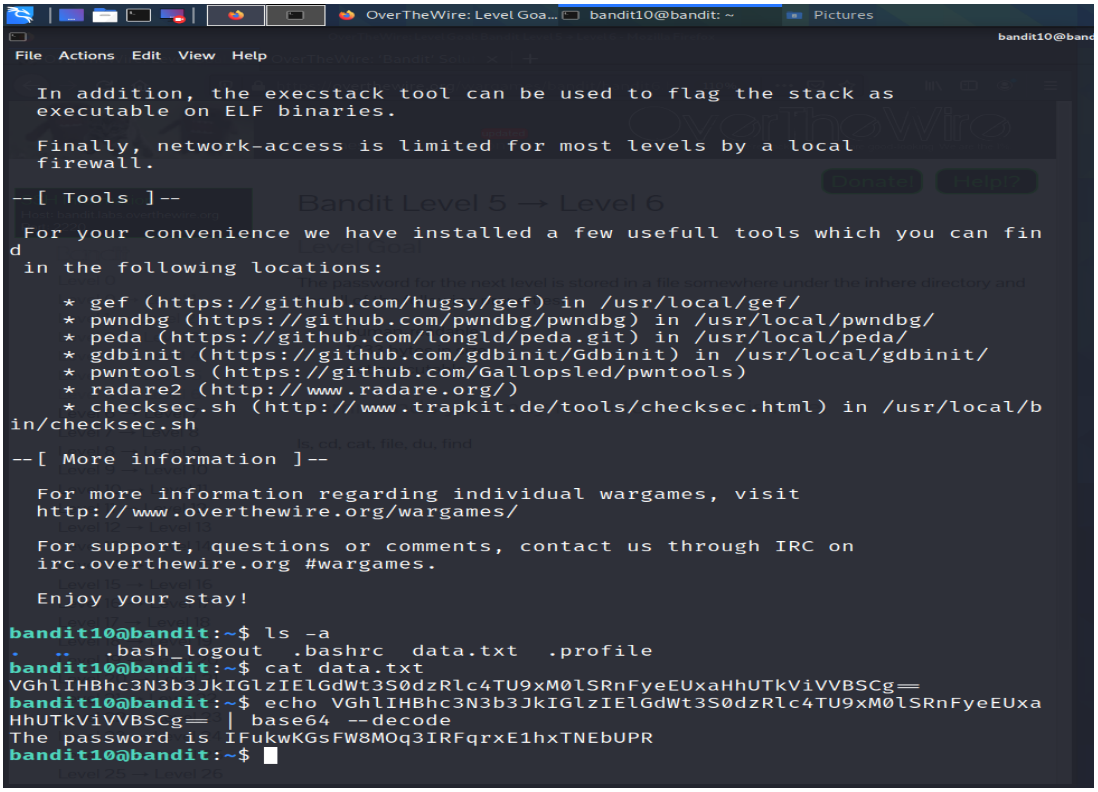


```
 For your convenience we have installed a few usefull tools which you can fin
d
 in the following locations:

    * gef (https://github.com/hugsy/gef) in /usr/local/gef/
    * pwndbg (https://github.com/pwndbg/pwndbg) in /usr/local/pwndbg/
    * peda (https://github.com/longld/peda.git) in /usr/local/peda/
    * gdbinit (https://github.com/gdbinit/Gdbinit) in /usr/local/gdbinit/
    * pwntools (https://github.com/Gallopsled/pwntools)
    * radare2 (http://www.radare.org/)
    * checksec.sh (http://www.trapkit.de/tools/checksec.html) in /usr/local/b
in/checksec.sh

--[ More information ]--

  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!

bandit9@bandit:~$
bandit9@bandit:~$ ls -a
.  ..  .bash_logout  .bashrc  data.txt  .profile
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:~$ 
```

Level 10-11

We are informed that the password for the next level is stored inside a file named data.txt. So, to find it we use the ls command. Now, we are hinted that the password is encrypted in Base64. Now we can either read the file with cat command and decode the Base64 manually but we have a command in Linux that can do the heavy lifting for us. So, we use piping to use cat command and base64 command with d parameter to read and decode the text simultaneously. This gives us the password for the next level. We will use it to get an SSH connection as bandit11.

ls
cat data.txt | base64 --decode
ssh bandit11@localhost

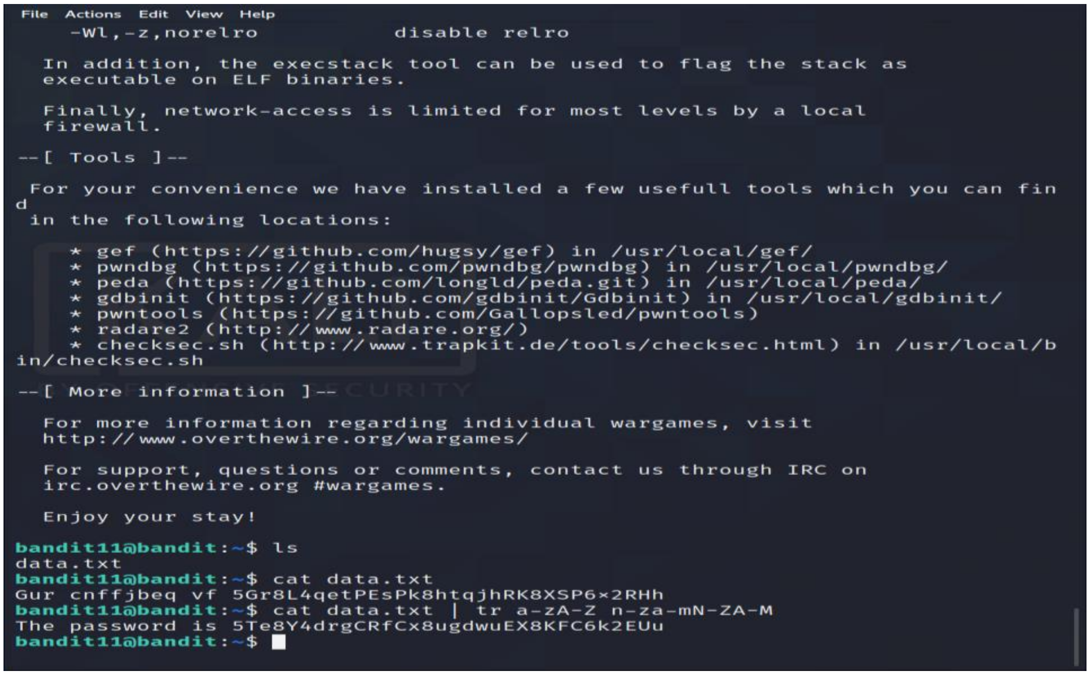


```
  In addition, the execstack tool can be used to flag the stack as
  executable on ELF binaries.

  Finally, network-access is limited for most levels by a local
  firewall.

--[ Tools ]--

 For your convenience we have installed a few usefull tools which you can fin
d
 in the following locations:

    * gef (https://github.com/hugsy/gef) in /usr/local/gef/
    * pwndbg (https://github.com/pwndbg/pwndbg) in /usr/local/pwndbg/
    * peda (https://github.com/longld/peda.git) in /usr/local/peda/
    * gdbinit (https://github.com/gdbinit/Gdbinit) in /usr/local/gdbinit/
    * pwntools (https://github.com/Gallopsled/pwntools)
    * radare2 (http://www.radare.org/)
    * checksec.sh (http://www.trapkit.de/tools/checksec.html) in /usr/local/b
in/checksec.sh

--[ More information ]--

  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!

bandit10@bandit:~$ ls -a
.  ..  .bash_logout  .bashrc  data.txt  .profile
bandit10@bandit:~$ cat data.txt
VGhlIHBhc3N3b3JkIGlzIElGdWt3S0dzRlc4TU9xM0lSRnFyeEUxaHhUTkViVVBSCg==
bandit10@bandit:~$ echo VGhlIHBhc3N3b3JkIGlzIElGdWt3S0dzRlc4TU9xM0lSRnFyeEUxa
HhUTkViVVBSCg== | base64 --decode
The password is IFukwKGsFW8MOq3IRFqrxE1hxTNEbUPR
bandit10@bandit:~$ 
```

Level 11-12

We are informed that the password for the next level is stored inside a file named data.txt. So, to find it we use the ls command. Now, we are hinted that the file containing the password has changed the format of letters in such a way that all the lowercase and uppercase letters have been rotated by 13 positions. If we can remember right that exactly what happens in ROT13 encryption. Now, to convert the text, we can use the ‘tr’ command. This command translates characters depending on the parameters provided. We used n-z and a-m because tr won’t continue to translate after the Z. This gives us the password for the next level. We will use it to get an SSH connection as bandit12.

```
ls
cat data.txt | tr a-zA-Z n-za-mN-ZA-M
ssh bandit12@localhost
```



```
    -Wl,-z,norelro           disable relro

  In addition, the execstack tool can be used to flag the stack as
  executable on ELF binaries.

  Finally, network-access is limited for most levels by a local
  firewall.

--[ Tools ]--

 For your convenience we have installed a few usefull tools which you can fin
d
 in the following locations:

    * gef (https://github.com/hugsy/gef) in /usr/local/gef/
    * pwndbg (https://github.com/pwndbg/pwndbg) in /usr/local/pwndbg/
    * peda (https://github.com/longld/peda.git) in /usr/local/peda/
    * gdbinit (https://github.com/gdbinit/Gdbinit) in /usr/local/gdbinit/
    * pwntools (https://github.com/Gallopsled/pwntools)
    * radare2 (http://www.radare.org/)
    * checksec.sh (http://www.trapkit.de/tools/checksec.html) in /usr/local/b
in/checksec.sh

--[ More information ]--

  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!

bandit11@bandit:~$ ls
data.txt
bandit11@bandit:~$ cat data.txt
Gur cnffjbeq vf 5Gr8L4qetPEsPk8htqjhRK8XSP6x2RHh
bandit11@bandit:~$ cat data.txt | tr a-zA-Z n-za-mN-ZA-M
The password is 5Te8Y4drgCRfCx8ugdwuEX8KFC6k2EUu
bandit11@bandit:~$ 
```

Level 12-13

We are informed that the password for the next level is stored inside a directory named inhere. So, to find it we use the ls command. We are hinted that the file containing the password is in the form of a hex dump. Just out of curiosity, let’s read the file using the cat command. As we can see in the given image that the password is not at all readable. We are also told that the password file has been repeatedly compressed. Now to decompress we are going to need a directory with read and write permissions. The tmp directory in root contains the required permissions.

```
ls
cat data.txt
```

So, let’s create a directory inside the tmp directory. Here we named it pavan. Now for further operations let’s copy the file in the directory we just created. Now let’s traverse to our directory using the cd command. Now we check if we have our file in this directory. Now to understand the type of file we are going to use the file command it returns us the type of file. On running the command, we are informed that the file is ASCII text. But as we saw earlier that it is not readable. The xxd command is used in Linux to make the hexdump of a file. It is also used to reverse this process. Let’s use it to retrieve the original file. We are going to use the ‘r’ parameter to revert the process and provide it with a filename where it should store its output. Here we will name it data1

Now it’s time to check the retrieved file, we use the file command again. 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 data2.gz. Now using the gzip command and -d parameter, we decompress the file.

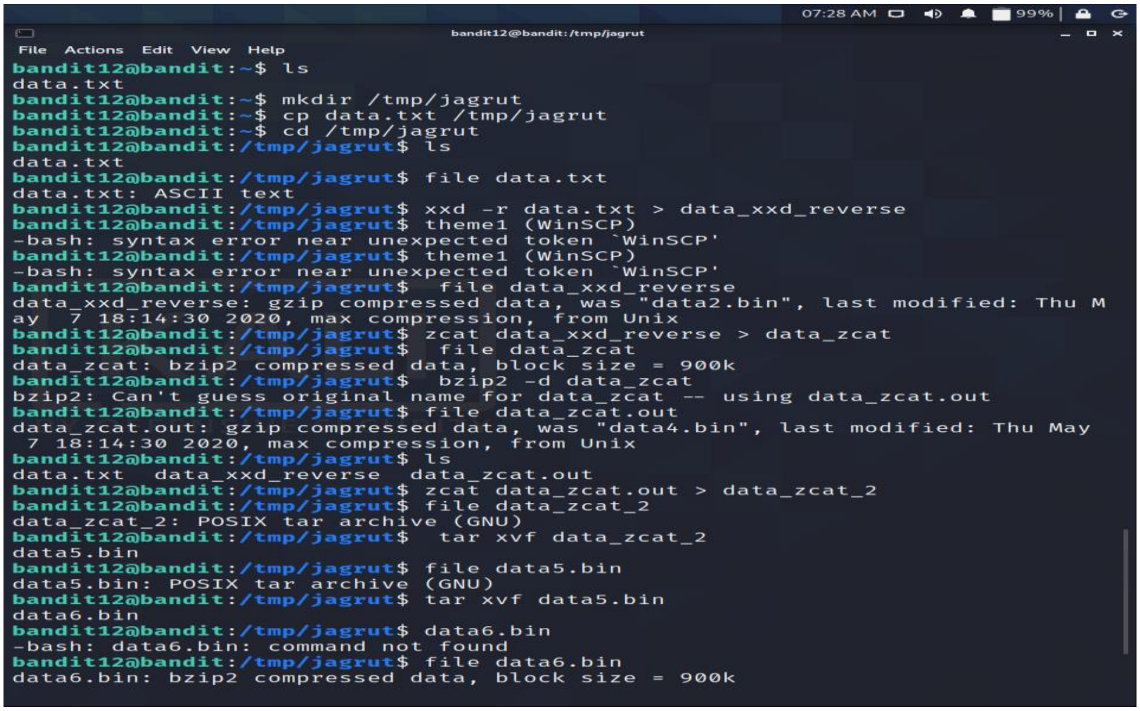
```
mkdir /tmp/pavan
cp data.txt /tmp/pavan
cd /tmp/pavan
ls
file data.txt
xxd -r data.txt data1
file data1
mv data1 data2.gz
gzip -d data2.gz
```

Now it’s time to check the retrieved file, we use the file command again. This tells us that it is a bzip2 compressed file.Now to decompress first, we need to rename the file and provide it with a proper bzip2 extension. We are going to use the move command for this. We renamed the file as data3.bz2. Now using the bzip2 command and -d parameter, we decompress the file.

Now it’s time to check the retrieved file, we use the file command again. 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 data4.gz. Now using the gzip command and -d parameter, we decompress the file.

Now it’s time to check the retrieved file, we use the file command again. This tells us that it is a tar archive file.



```
bandit12@bandit:~$ ls
data.txt
bandit12@bandit:~$ mkdir /tmp/jagrut
bandit12@bandit:~$ cp data.txt /tmp/jagrut
bandit12@bandit:~$ cd /tmp/jagrut
bandit12@bandit:/tmp/jagrut$ ls
data.txt
bandit12@bandit:/tmp/jagrut$ file data.txt
data.txt: ASCII text
bandit12@bandit:/tmp/jagrut$ xxd -r data.txt > data_xxd_reverse
bandit12@bandit:/tmp/jagrut$ theme1 (WinSCP)
-bash: syntax error near unexpected token `WinSCP'
bandit12@bandit:/tmp/jagrut$ theme1 (WinSCP)
-bash: syntax error near unexpected token `WinSCP'
bandit12@bandit:/tmp/jagrut$  file data_xxd_reverse
data_xxd_reverse: gzip compressed data, was "data2.bin", last modified: Thu M
ay  7 18:14:30 2020, max compression, from Unix
bandit12@bandit:/tmp/jagrut$ zcat data_xxd_reverse > data_zcat
bandit12@bandit:/tmp/jagrut$  file data_zcat
data_zcat: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/jagrut$  bzip2 -d data_zcat
bzip2: Can't guess original name for data_zcat -- using data_zcat.out
bandit12@bandit:/tmp/jagrut$ file data_zcat.out
data_zcat.out: gzip compressed data, was "data4.bin", last modified: Thu May
 7 18:14:30 2020, max compression, from Unix
bandit12@bandit:/tmp/jagrut$ ls
data.txt  data_xxd_reverse  data_zcat.out
bandit12@bandit:/tmp/jagrut$ zcat data_zcat.out > data_zcat_2
bandit12@bandit:/tmp/jagrut$ file data_zcat_2
data_zcat_2: POSIX tar archive (GNU)
bandit12@bandit:/tmp/jagrut$  tar xvf data_zcat_2
data5.bin
bandit12@bandit:/tmp/jagrut$ file data5.bin
data5.bin: POSIX tar archive (GNU)
bandit12@bandit:/tmp/jagrut$ tar xvf data5.bin
data6.bin
bandit12@bandit:/tmp/jagrut$ data6.bin
-bash: data6.bin: command not found
bandit12@bandit:/tmp/jagrut$ file data6.bin
data6.bin: bzip2 compressed data, block size = 900k
```

Now to extract we will use the tar command with xvf parameters. This gives us a file named data5.bin

file data2

mv data2 data3.bz2

bzip2 -d data3.bz2

file data3

mv data3 data4.gz

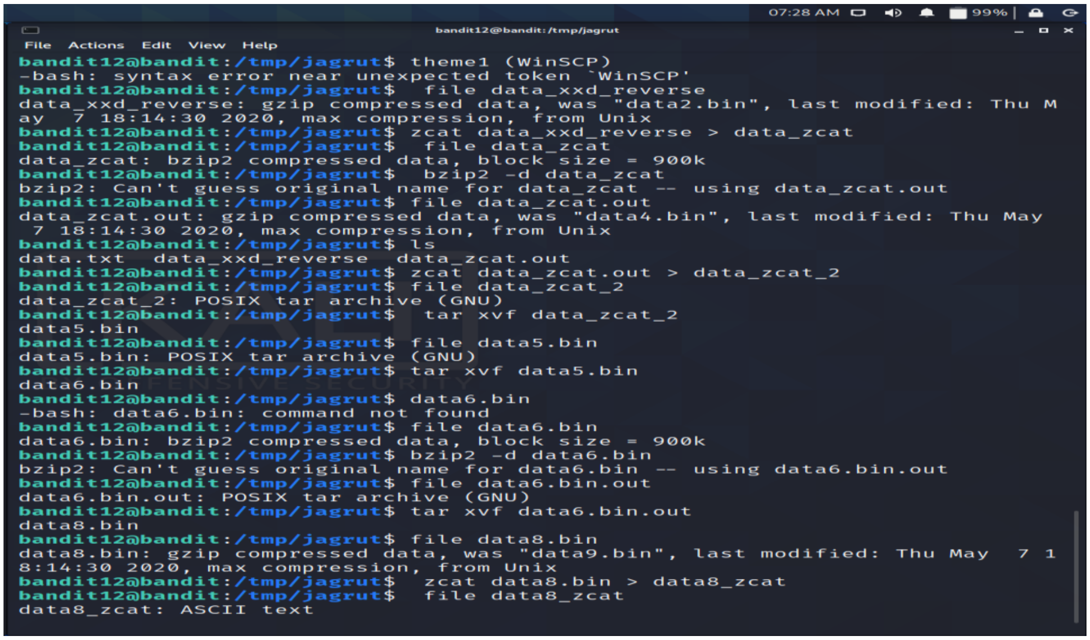
gzip -d data4.gz

file data4

tar -xvf data4

Now it’s time to check the retrieved file, we use the file command again. This tells us that it is a tar archive file. Now to extract we will use the tar command with xvf parameters. This gives us a file named data6.bin Now it’s time to check the retrieved file, we use the file command again. This tells us that it is a bzip2 compressed file.

Now decompress first, we need to rename the file and provide it with a proper bzip2 extension. We are going to use the move command for this. We renamed the file as data7.bz2. Now using the bzip2 command and -d parameter, we decompress the file.



```
bandit12@bandit:/tmp/jagrut$ theme1 (WinSCP)
-bash: syntax error near unexpected token `WinSCP'
bandit12@bandit:/tmp/jagrut$  file data_xxd_reverse
data_xxd_reverse: gzip compressed data, was "data2.bin", last modified: Thu M
ay  7 18:14:30 2020, max compression, from Unix
bandit12@bandit:/tmp/jagrut$ zcat data_xxd_reverse > data_zcat
bandit12@bandit:/tmp/jagrut$  file data_zcat
data_zcat: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/jagrut$  bzip2 -d data_zcat
bzip2: Can't guess original name for data_zcat -- using data_zcat.out
bandit12@bandit:/tmp/jagrut$ file data_zcat.out
data_zcat.out: gzip compressed data, was "data4.bin", last modified: Thu May
 7 18:14:30 2020, max compression, from Unix
bandit12@bandit:/tmp/jagrut$ ls
data.txt  data_xxd_reverse  data_zcat.out
bandit12@bandit:/tmp/jagrut$ zcat data_zcat.out > data_zcat_2
bandit12@bandit:/tmp/jagrut$ file data_zcat_2
data_zcat_2: POSIX tar archive (GNU)
bandit12@bandit:/tmp/jagrut$  tar xvf data_zcat_2
data5.bin
bandit12@bandit:/tmp/jagrut$ file data5.bin
data5.bin: POSIX tar archive (GNU)
bandit12@bandit:/tmp/jagrut$ tar xvf data5.bin
data6.bin
bandit12@bandit:/tmp/jagrut$ data6.bin
-bash: data6.bin: command not found
bandit12@bandit:/tmp/jagrut$ file data6.bin
data6.bin: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/jagrut$ bzip2 -d data6.bin
bzip2: Can't guess original name for data6.bin -- using data6.bin.out
bandit12@bandit:/tmp/jagrut$ file data6.bin.out
data6.bin.out: POSIX tar archive (GNU)
bandit12@bandit:/tmp/jagrut$ tar xvf data6.bin.out
data8.bin
bandit12@bandit:/tmp/jagrut$ file data8.bin
data8.bin: gzip compressed data, was "data9.bin", last modified: Thu May  7 1
8:14:30 2020, max compression, from Unix
bandit12@bandit:/tmp/jagrut$  zcat data8.bin > data8_zcat
bandit12@bandit:/tmp/jagrut$  file data8_zcat
data8_zcat: ASCII text
```

Now it's time to check the retrieved file, we use the file command again. This tells us that it is a tar archive file. Now to extract we will use the tar command with xvf parameters. This gives us a file named data8.bin

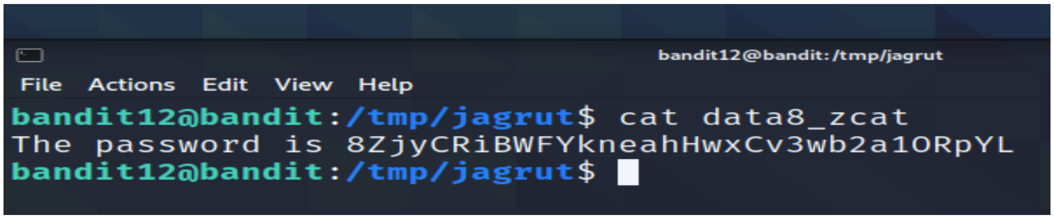
```
file data5.bin
tar -xvf data5.bin
file data6.bin
mv data6.bin data7.bz2
bzip2 -d data7.bz2
file data7
tar -xvf data7
```

Now it's time to check the retrieved file, we use the file command again. 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, we decompress the file.

Now to understand the type of file we are going to use the file command it returns us the type of file. On running the command, we are informed that the file is ASCII text. This might be a readable file. We use the cat command to read the file. This gives us the password for the next level. We will use it to get an SSH connection as bandit13.

```
file data8.bin
mv data8.bin data9.gz
gzip -d data9.gz
file data9
cat data9
ssh bandit13@localhost
```



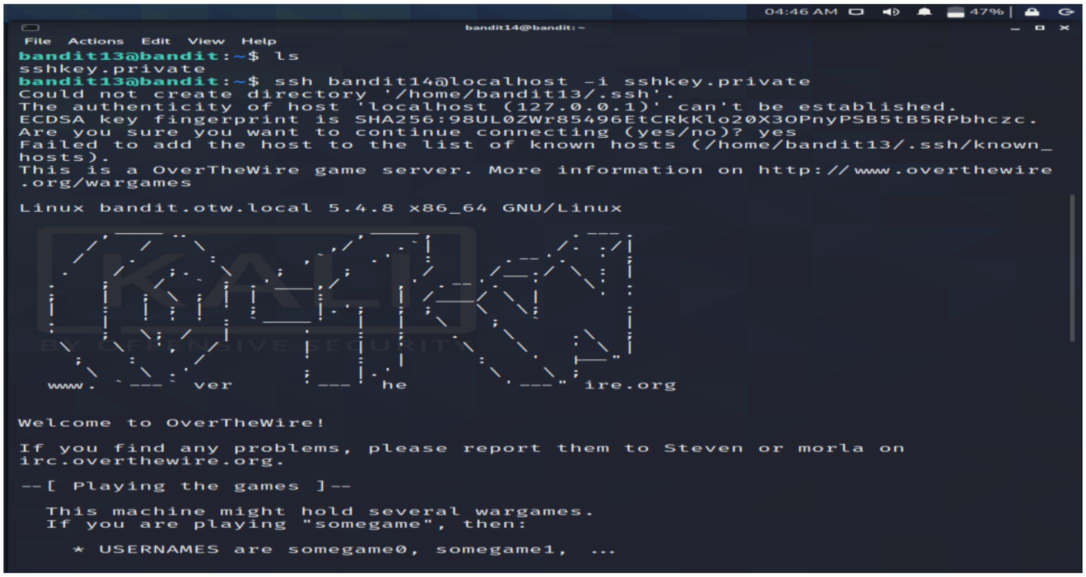
```
bandit12@bandit:/tmp/jagrut$ cat data8_zcat
The password is 8ZjyCRiBWFYkneahHwxCv3wb2a1ORpYL
bandit12@bandit:/tmp/jagrut$
```

Level 13-14

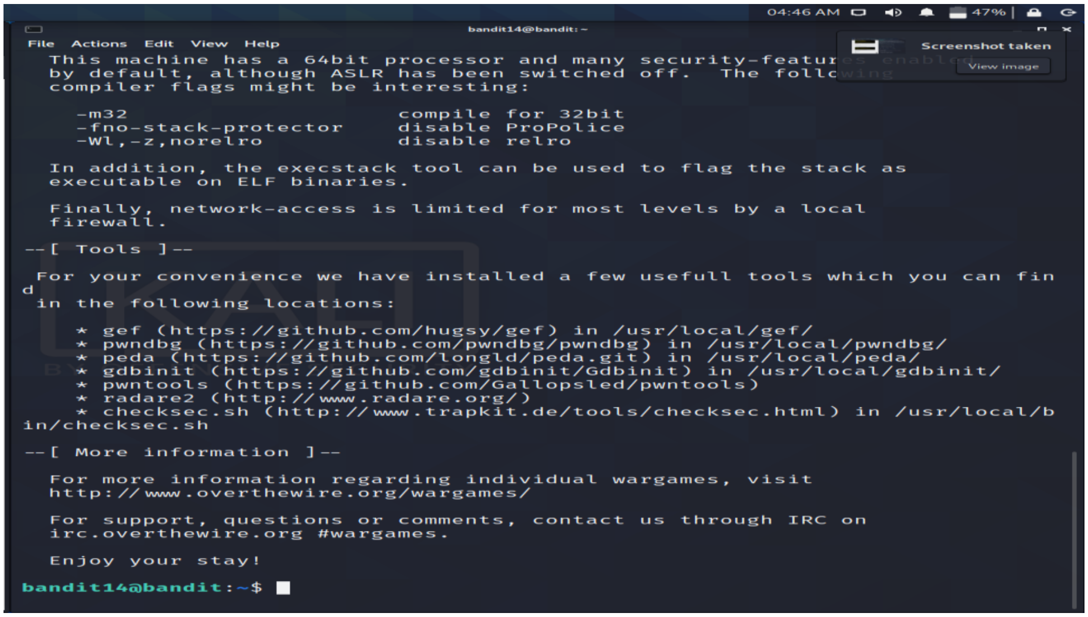
We are informed that we are not going to get a password for the next level. Instead, we are given an ssh private key. So, to get to the next level we are going to use that ssh private key. Firstly, let’s find that private key using the ls command. We found the private key. Now we will use it to get an SSH connection as bandit14.

ls

ssh bandit14@localhost -i sshkey.private



```
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/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

Welcome to OverTheWire!

If you find any problems, please report them to Steven or morla on
irc.overthewire.org.

--[ Playing the games ]--

  This machine might hold several wargames.
  If you are playing "somegame", then:

    * USERNAMES are somegame0, somegame1, ...
```



```
  This machine has a 64bit processor and many security-featur
  by default, although ASLR has been switched off.  The follc
  compiler flags might be interesting:

    -m32                    compile for 32bit
    -fno-stack-protector    disable ProPolice
    -Wl,-z,norelro          disable relro

  In addition, the execstack tool can be used to flag the stack as
  executable on ELF binaries.

  Finally, network-access is limited for most levels by a local
  firewall.

--[ Tools ]--

 For your convenience we have installed a few usefull tools which you can fin
d
 in the following locations:

    * gef (https://github.com/hugsy/gef) in /usr/local/gef/
    * pwndbg (https://github.com/pwndbg/pwndbg) in /usr/local/pwndbg/
    * peda (https://github.com/longld/peda.git) in /usr/local/peda/
    * gdbinit (https://github.com/gdbinit/Gdbinit) in /usr/local/gdbinit/
    * pwntools (https://github.com/Gallopsled/pwntools)
    * radare2 (http://www.radare.org/)
    * checksec.sh (http://www.trapkit.de/tools/checksec.html) in /usr/local/b
in/checksec.sh

--[ More information ]--

  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!

bandit14@bandit:~$
```