Bandit Level 1 to 10 (CTF) Over The Wire

Prepared by Ali Iqbal

Bandit Level 0

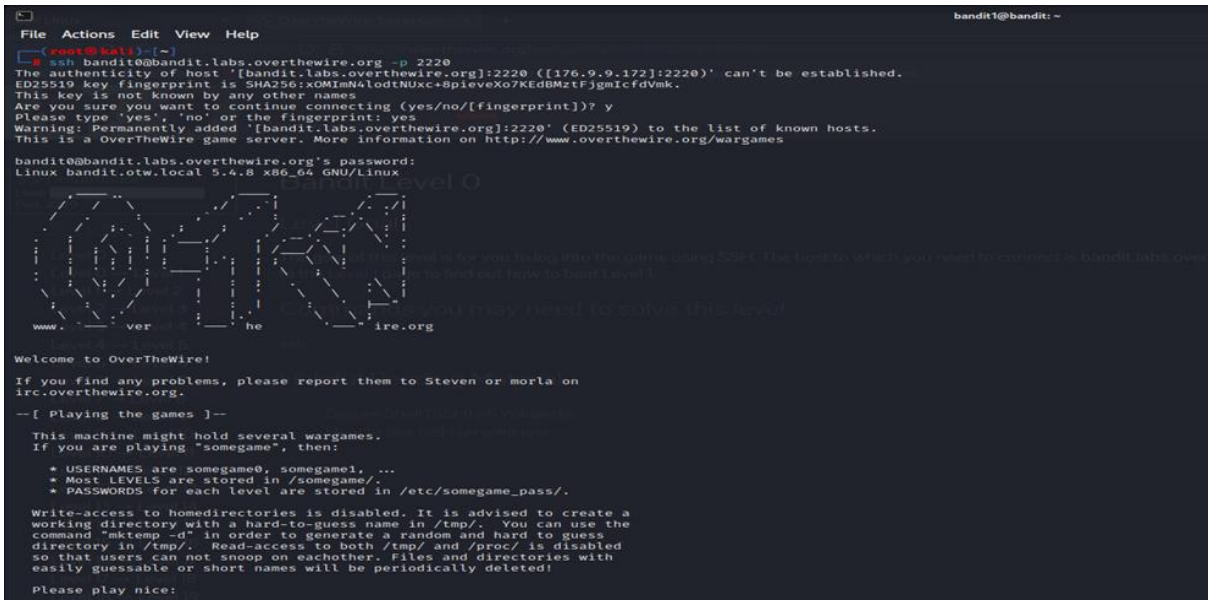
- The goal of this level is for you to log into the game using SSH. The host to which you need to connect is **bandit.labs.overthewire.org**, on port 2220. The username is **bandit0** and the password is **bandit0**. Once logged in, go to the Level 1 page to find out how to beat Level 1.

Step 1 : bandit server connect in kali terminal through ssh

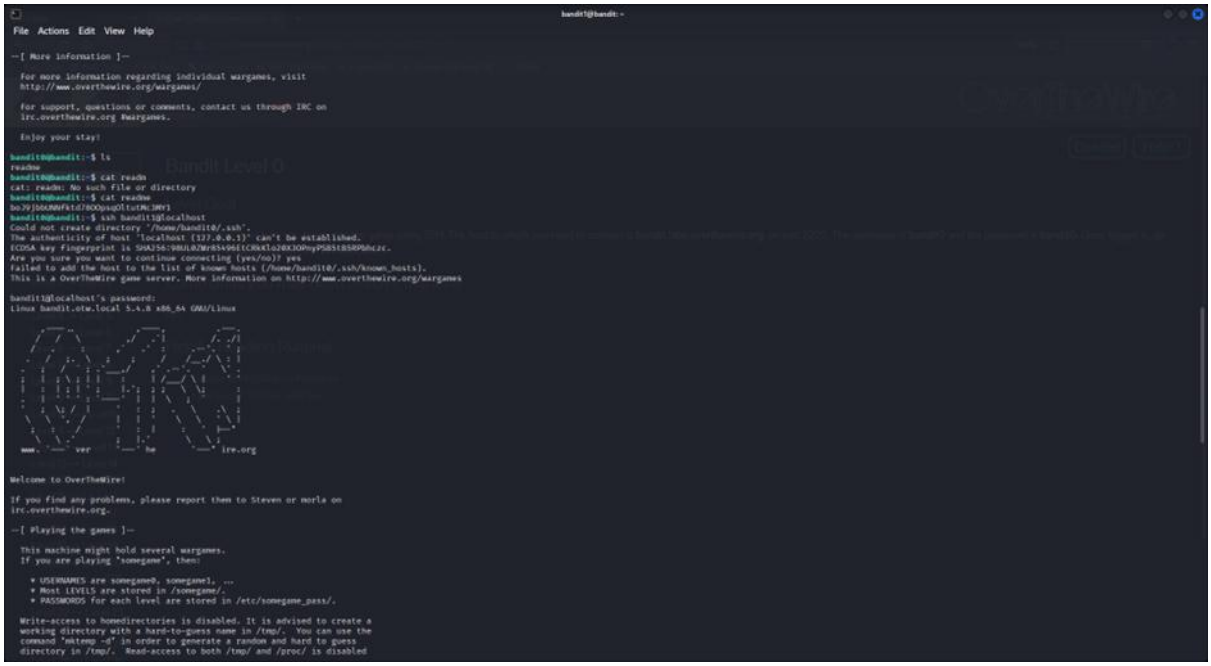
Level 0

Password : bandit 0

· **ssh bandit0@bandit.labs.overthewire.org -p 2220**



level 0



Command description:

LS : first LS (for list) all directory in the folder

CD : CD for change directory

CAT : cat for open any file in linux command line. In level0 the file name is “readme” and password are stored in this particular file so we can use “ cat readme “ for open file.

· It reads data from the file and gives their content as output. It helps us to create, view, concatenate files.

Here we have find the level 0 password

Level 0 password : boJ9jbbUNNfktd78OOpsqOltutMc3MY1

Go to the next level

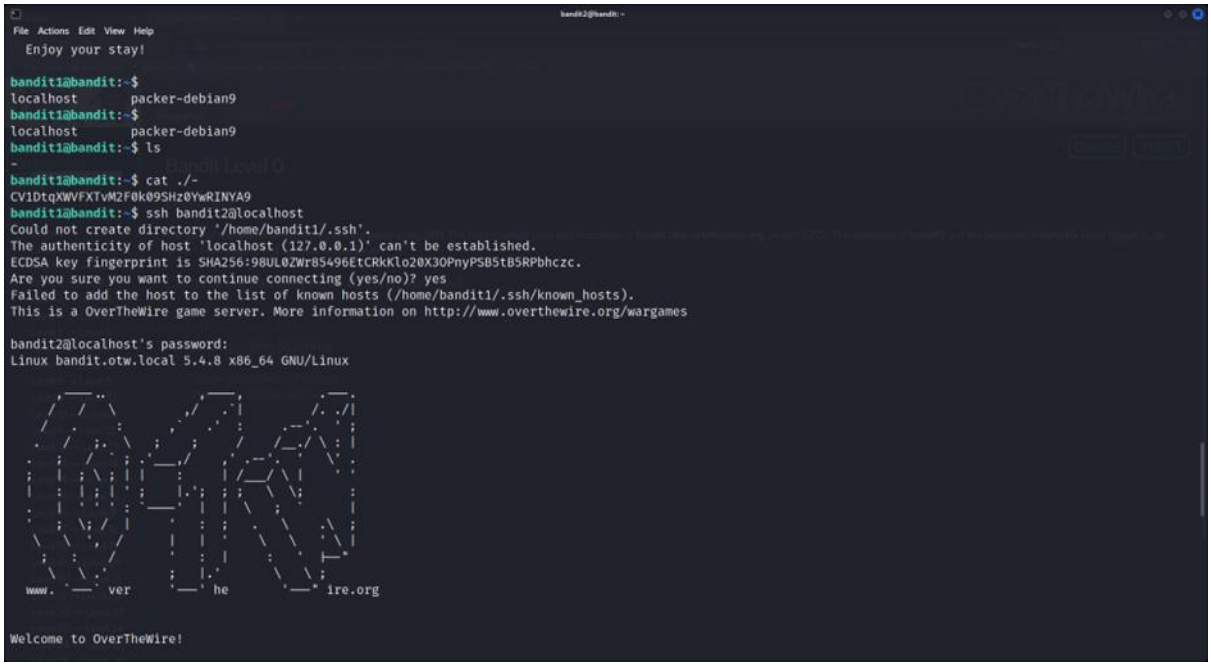
· Ssh bandit1@localhost

· Level 0 — level 1

Find the level 1 password

· Level 1 to Level 2

Password: CV1DtqXWVFXTvM2F0k09SHz0YwRINYA9



level 1 to 2

Level 1 to 2

Command description: For a command, if using — as an argument in place of a file name will mean STDIN or STDOUT.

- (Hyphen.) Expands to the current option flags (the single-letter option names concatenated into a string) as specified on invocation, by the set builtin command, or implicitly by the shell.

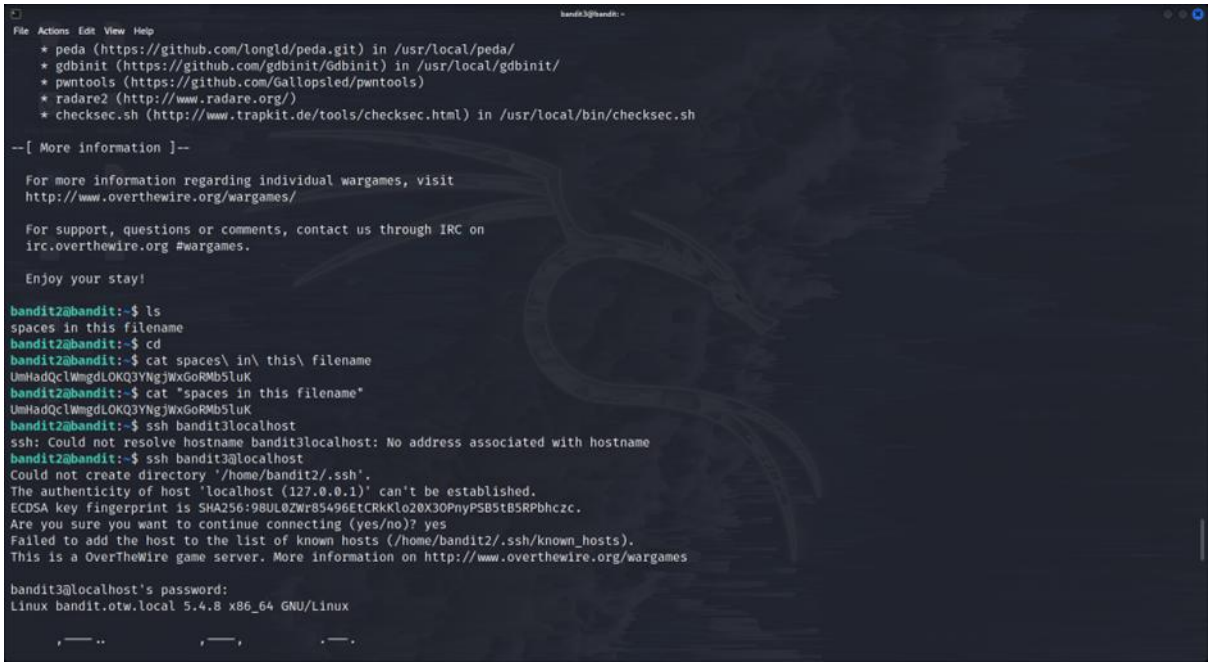
• stdin – It stands for standard input, and is used for taking text as an input.

· stdout – It stands for standard output, and is used to text output of any command you type in the terminal, and then that output is stored in the stdout stream.

the password for the next level is stored inside a file named -(hyphen). In this level file name is "-" password is store in the "-". The — (hyphen) as stdin/Stout.

So we can not used cat command directly. Like "cat — " , we will prefix the command with the path ./, this will help to read the password stored in "-".

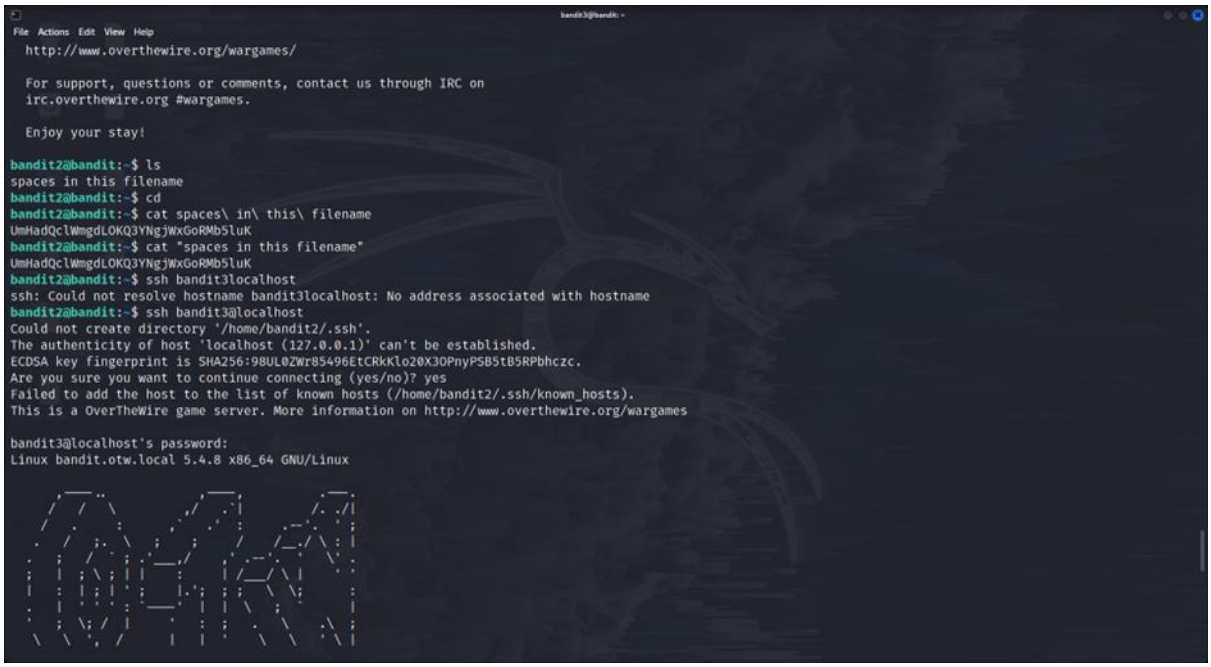
Level 2 to 3



Level 2 to level 3

Password : UmHadQclWmgdLOKQ3YNgjWxGoRMb5luK

Level 2 to level 3



Command description : in the level 2 to 3 password is stored in spaces in this filename this file.

We can access the file using cat, but the file name is {spaces in this filename}, so command line understand different file , its look like : ‘spaces’, ‘in’, ’this’, ‘filename’. so we can used this method

· cat “spaces in this filename” [“”] so command line understand it is string

· second method

cat spaces\ in\ this\ filename

use backslash before each space, or embed the entire file name as a string.

“ ./ “ is used in a pathname to indicate the current directory. It can also run a script from the current working directory.

Level 3 to level 4

Password : pIwrPrtPN36QITSp3EQaw936yaFoFgAB

Command description: in the level 3 to 4

The password is stored in a hidden file in the inhere directory. First we can used LS (or list)

We can see the inhere directory are open then, used “cd” to move into the inhere folder.

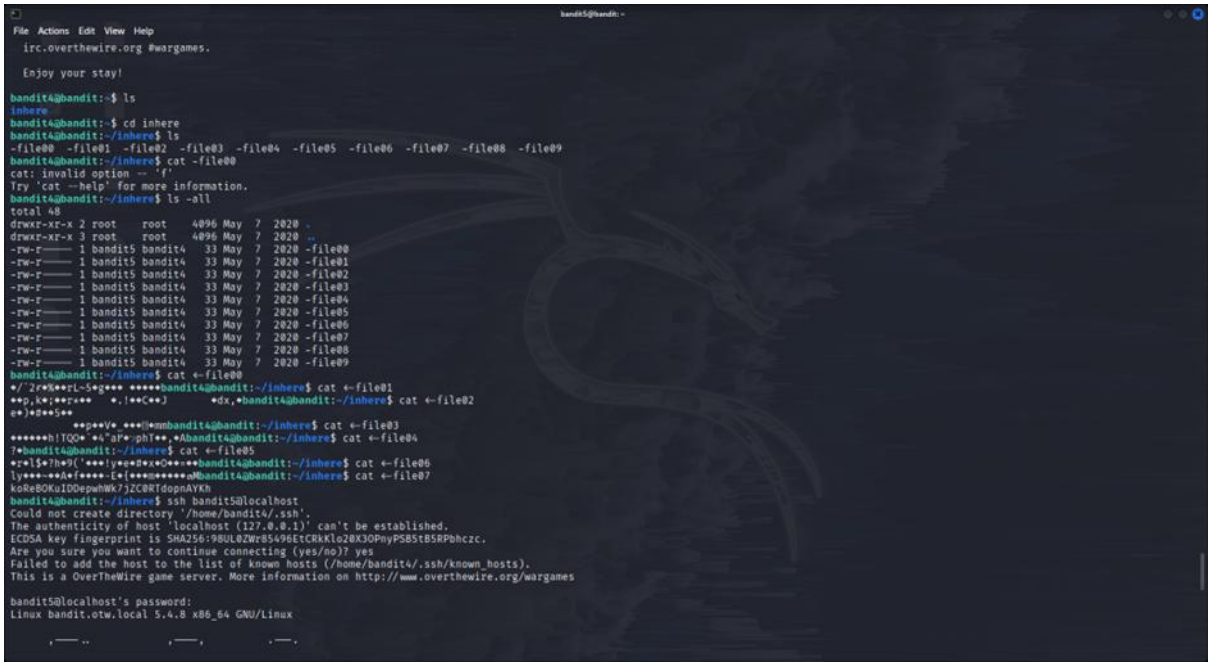
Then use ls -al to list all of the files including the hidden

In linux all the hidden files and folder are stored with a dot in front of their name.

So open hidden file use command : cat .hidden

Level 4 to level 5

Password: koReBOKuIDDepwhWk7jZC0RTdopnAYKh



Command description : in level 4 to 5

The password is stored in file inside stored inside a human-readable file. We can see the inhere directory are open then, used “cd” to move into the inhere folder. use ls -al to list all file

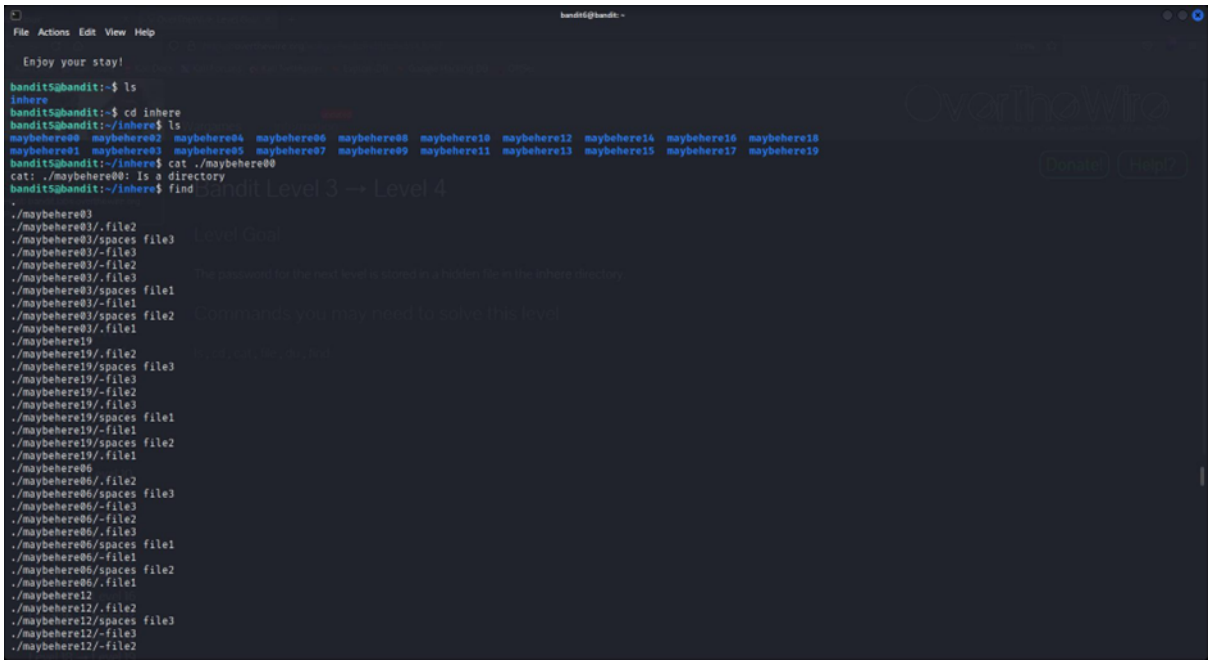
Here all file are display like : -file00, -file01, -file02, -file03, -file04, -file05, -file06, -file07, -file08, -file09.

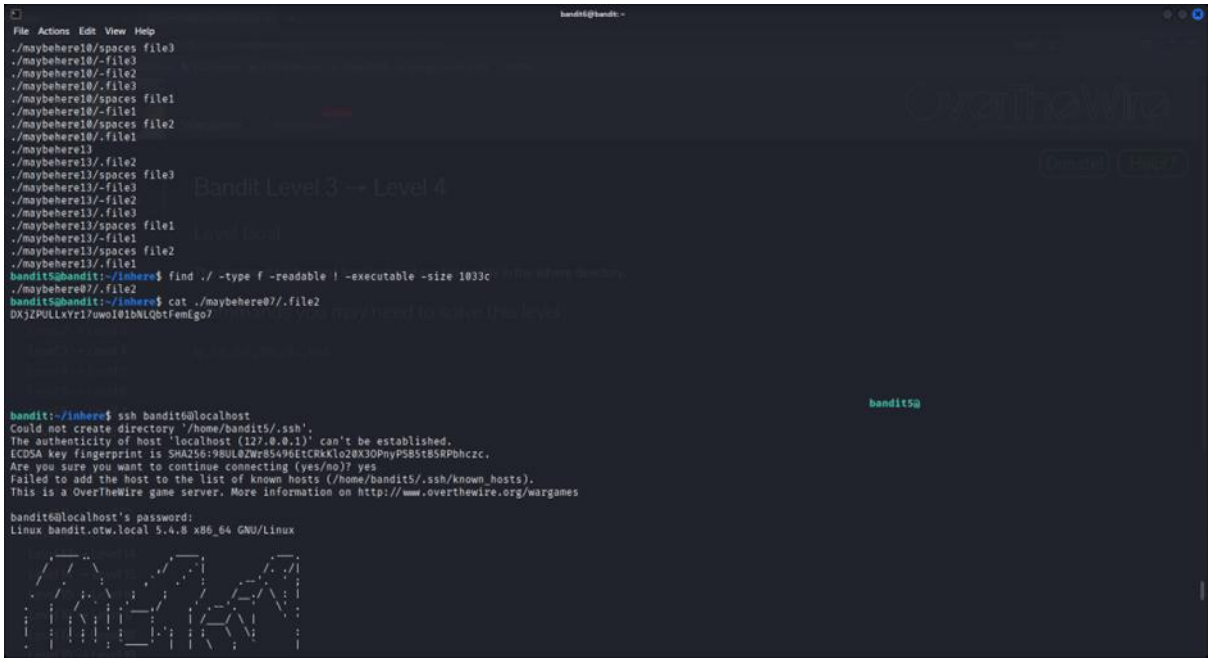
We can open one by one file using cat < command : < its stdin to open the file

Cat <-file07

Level 5 to level 6

DXjZPULLxYr17uwoI01bNLQbtFemEgo7





Command description : in level 5 to 6

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

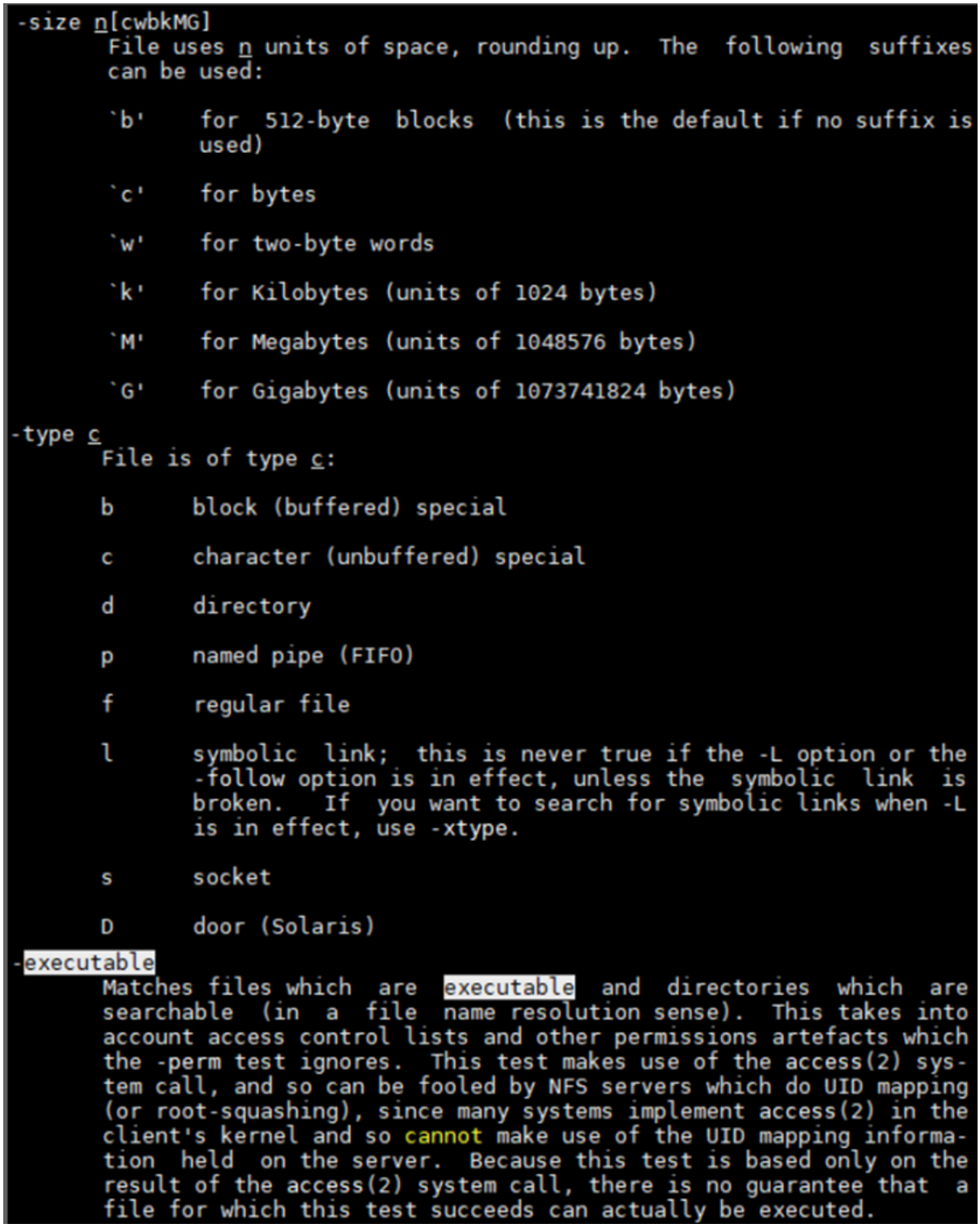
human-readable

1033 bytes in size

not executable

first cd to move directory inhere , then ls -all to list all files. here is so many files

maybehere00 to maybehere15 , here we are use find command to specific file.



```
-size n[cwbkMG]
       File uses n units of space, rounding up.  The  following  suffixes
       can be used:

       `b'    for  512-byte  blocks  (this is the default if no suffix is
              used)

       `c'    for bytes

       `w'    for two-byte words

       `k'    for Kilobytes (units of 1024 bytes)

       `M'    for Megabytes (units of 1048576 bytes)

       `G'    for Gigabytes (units of 1073741824 bytes)

-type c
       File is of type c:

       b      block (buffered) special

       c      character (unbuffered) special

       d      directory

       p      named pipe (FIFO)

       f      regular file

       l      symbolic  link;  this is never true if the -L option or the
              -follow option is in effect, unless the  symbolic  link  is
              broken.   If  you want to search for symbolic links when -L
              is in effect, use -xtype.

       s      socket

       D      door (Solaris)

-executable
       Matches files which  are  executable  and  directories  which  are
       searchable  (in  a  file  name resolution sense).  This takes into
       account access control lists and other permissions artefacts which
       the -perm test ignores.  This test makes use of the access(2) sys-
       tem call, and so can be fooled by NFS servers which do UID mapping
       (or root-squashing), since many systems implement access(2) in the
       client's kernel and so cannot make use of the UID mapping informa-
       tion  held  on the server.  Because this test is based only on the
       result of the access(2) system call, there is no guarantee that  a
       file for which this test succeeds can actually be executed.
```

here find manual keyword to easy to find query

command : find ./ -type f -readable ! -executable -size 1033c

We use -size 1033 to look for the file-size requirement

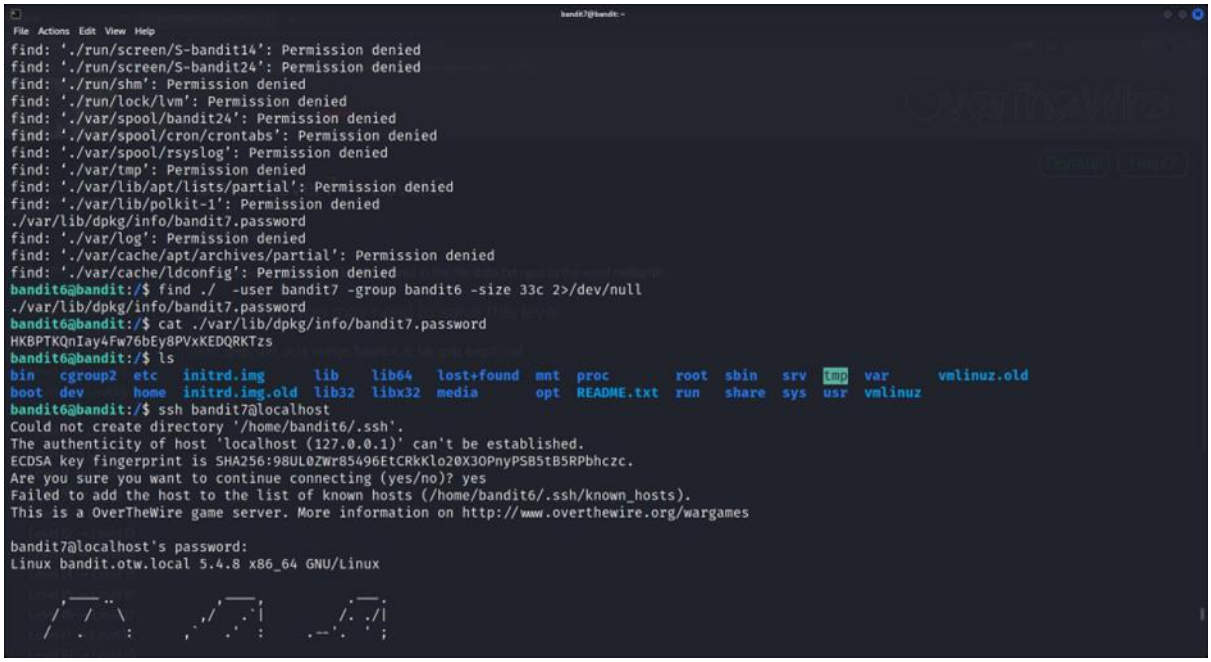
We use -type f to only look at files

We are use -executable flag , which search for executable files and allows operators like “ ! ” for negation.

We use -readable flag , means you have permission to read the file.

Level 6 to level 7

HKBPTKQnIay4Fw76bEy8PVxKEDQRKTzs



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

find command can be used to find files on the server.

-type f, because we are looking for a file

-user bandit7, to find files owned by the ‘bandit7’ user

-group bandit6, to find files owned by the ‘bandit6’ group

-size 33c, to find files of size 33 bytes

Command : find ./ -type f -user bandit7 -group bandit6 -size 33c

```
find: './run/lock/lvm': Permission denied
find: './var/spool/bandit24': Permission denied
find: './var/spool/cron/crontabs': Permission denied
find: './var/spool/rsyslog': Permission denied
find: './var/tmp': Permission denied
find: './var/lib/apt/lists/partial': Permission denied
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:/$ find ./  -user bandit7 -group bandit6 -si
./var/lib/dpkg/info/bandit7.password
```

Here we get the password /var/lib/dpkg/info/bandit7.password

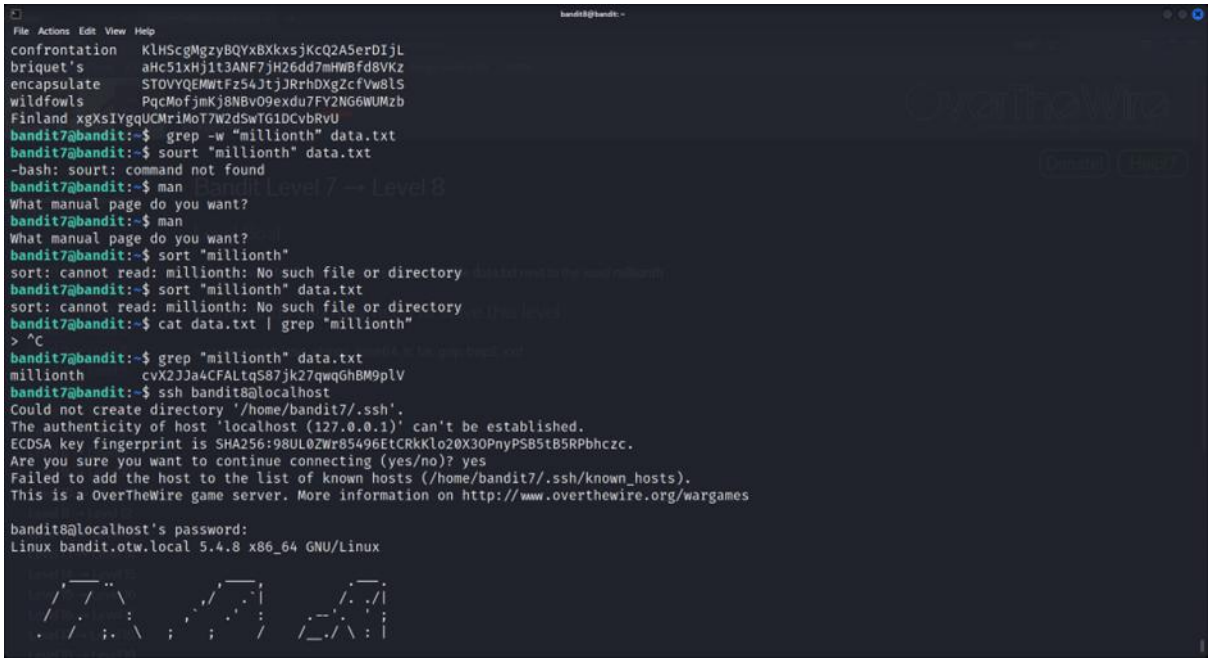
Cat /var/lib/dpkg/info/bandit7.password

We can use second method

Command : append 2>/dev/null, which will 'hide' all error messages 1.

level 7 to Level 8

cvX2JJa4CFALtqS87jk27qwqGhBM9plV



Command description : The password for the next level is stored in the file data.txt next to the word millionth. We can see here so many words and file open. But we want to particular millionth key ward data are stored in the millionth.

Command : grep -w "millionth" data.txt

- Here we are using grep command. it can be used to search lines that contain a specific pattern like follow grep <pattern>

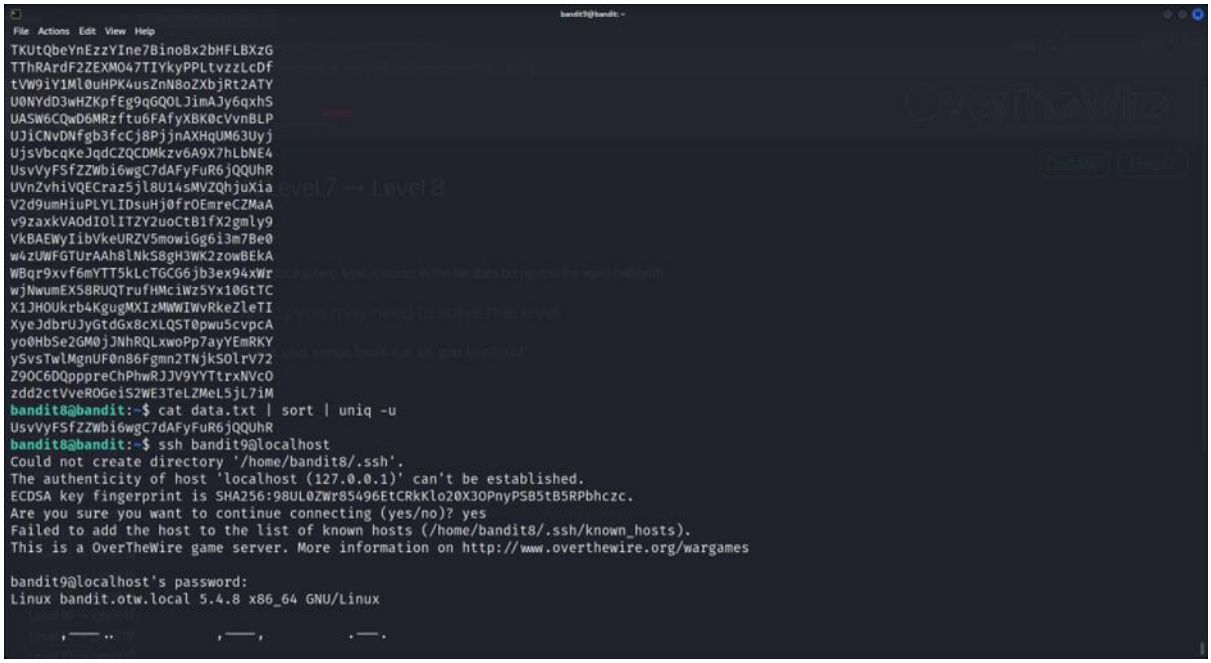
Second method : cat data.txt | grep millionth

In this command using (|) pipe its used to combine two or more commands.

Level 8 to 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

UsvVyFSfZZWbi6wgC7dAFyFuR6jQQUhR



Command description : in level 8 to 9 . here we can see so many password are listed but our password is stored in data.txt .

Command : cat data.txt | sort | uniq -u

- uniq is a command that filters input and writes to the output.

- it filters based on identical lines. It has a flag -u, which filters for unique lines

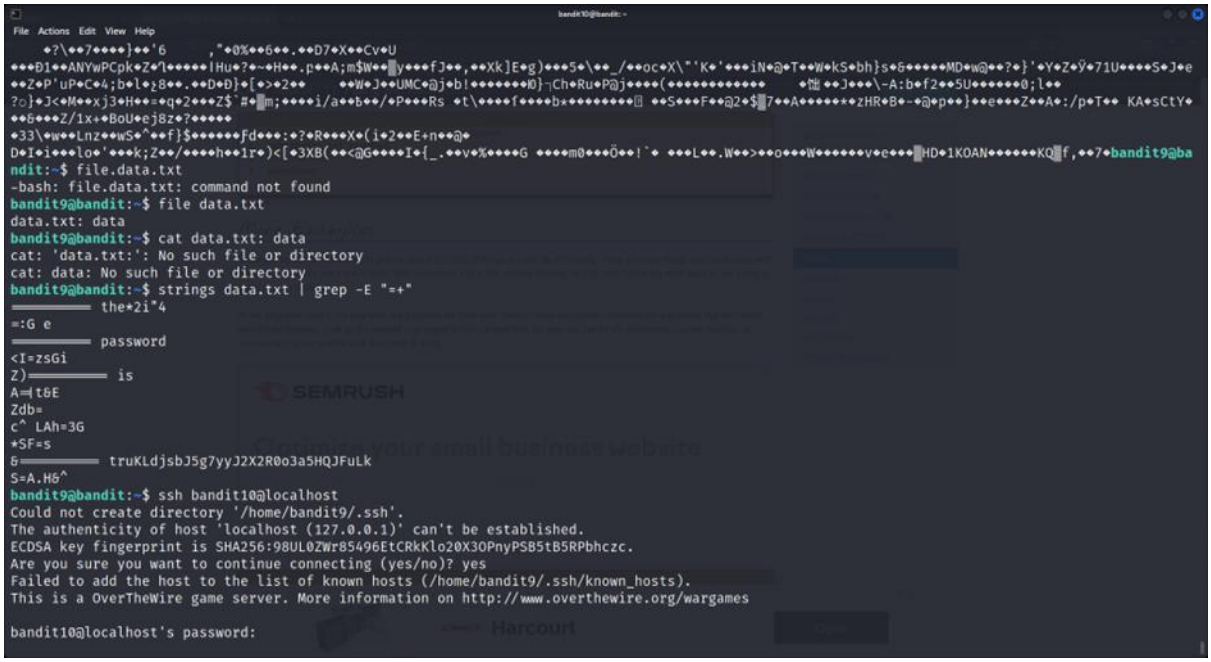
- uniq use for collect uniq lines

- sort we can use sort to the lines needs to be sorted.

- (|) pipe its used to combine two or more commands.

Level 9 to level 10

truKLdjsbJ5g7yyJ2X2R0o3a5HQJFuLk



The password for the next level is stored in the file **data.txt** in one of the few human-readable strings, beginning with several '=' characters.

This level is similar to previous levels, which basically require us to search for the password in a text file. However, the difficulty is that you cannot perform the **cat+grep** command on it because it is a "binary" file instead of a text file.

This is when you can try out the **strings** command. Let's look at the description of the strings command:

The string functions perform string operations on null-terminated strings. See the individual man pages for descriptions of each function.

In short, it goes through the entire file and any string values that it is able to find, it will display it to the output. As per the clue given to us to clear this level, let's run a **strings** command on the data.txt file and **grep** only records with the "=" characters.

The password to gain access to the next level is **truKLdjsbJ5g7yyJ2X2R0o3a5HQJFuLk**.