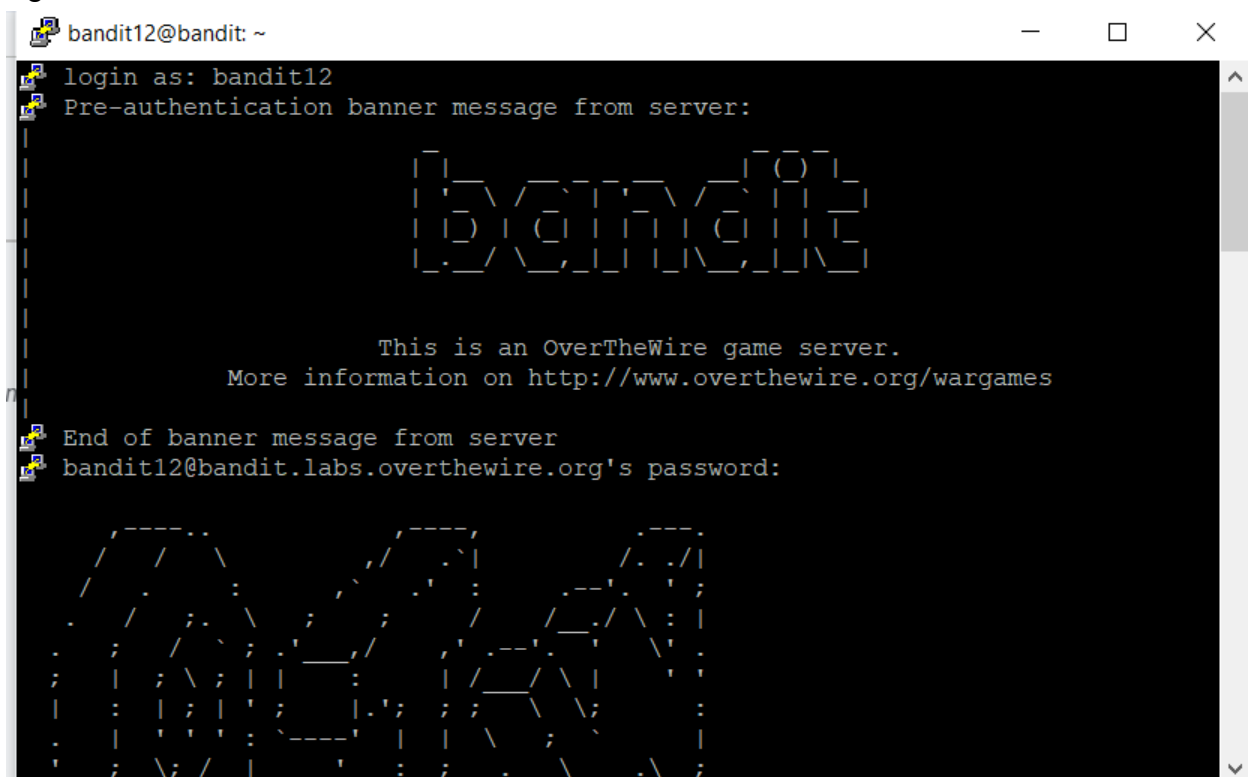


Bandit OTW Walkthrough Write-Up

Level : 12 → 13

Challenge: The password for the next level is stored in the file `data.txt`, which is a hexdump of a file that has been repeatedly compressed. For this level it may be useful to create a directory under `/tmp` in which you can work using `mkdir`. For example: `mkdir /tmp/myname123`. Then copy the datafile using `cp`, and rename it using `mv` (read the manpages!)

Step 1: Make sure you used the password from the previous level (Level 11 → 12) to login to bandit12.



```
bandit12@bandit: ~
login as: bandit12
Pre-authentication banner message from server:

      bandit12

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

End of banner message from server
bandit12@bandit.labs.overthewire.org's password:
```

Step 2: Once logged in, I used the `pwd` command to see which directory bandit12 was currently in. As shown in the next screenshot, It was in the home directory of bandit12

Step 3: I made a directory called `level2` under the `tmp` directory using the `mkdir` command. Then I used the `cp` command to move the file `data.txt` which is in my `/home/bandit12` directory to `/tmp/level2` directory. Afterwards, I used the `mv` command to rename the `data.txt` file inside the `/tmp/level2` directory to `data1.txt`

```
bandit12@bandit: ~  
bandit12@bandit:~$ pwd  
/home/bandit12  
bandit12@bandit:~$ ls  
data.txt  
bandit12@bandit:~$
```

```
bandit12@bandit: ~  
bandit12@bandit:~$ mkdir /tmp/level2  
mkdir: cannot create directory '/tmp/level2': File exists  
bandit12@bandit:~$ cp /home/bandit12/data.txt /tmp/level2  
bandit12@bandit:~$
```

Step 4.

Type `cat data1.txt`

In that `data.txt` file we see a bunch of numbers and letters aligned in rows. These characters aligned like this are known as a hexdump. A hexdump is a **hexadecimal** view of computer data, from **RAM** or from a file or storage device.

http://en.wikipedia.org/wiki/Hex_dump

Hexdump's are a very good way to reverse engineer computer programs.

[Here](#) is a very interesting TED talk by Mikko Hypponen explaining viruses and showing some hexdump examples at 1:45 and 10:00.

What we need to do is copy our `data.txt` file over to `/tmp` directory because the lab will not let us write data to the directory that the `data.txt` file is currently in.

First command we need to use to copy the file is `mkdir` which will make a directory we can do what we want in. This is created under the `/tmp` directory because files in the `/tmp` directory can be set to be deleted at reboot or set intervals easily.

Type `mkdir /tmp/yourname` (where `yourname` is anything you'd like).

Step 5.

Next we need to copy our file over to our new directory.

Type `cp data1.txt /tmp/yourname` (where `yourname` is what you chose).

Step 6.

Now we can move over to our created directory and uncompress our file.

Type `cd /tmp/yourname` and do an `ls` to make sure your file copied over to here.

Step 7.

First thing we need to is find out what kind of file we have.

Type `file data1.txt`

We see that we currently have a ASCII file but since we know we have a hexdump we need to convert the file to binary.

Step 8.

We need to reverse the ASCII file to a hexdump and Linux has a command to do just that.

Type `man xxd`

We see the second line under description says `xxd` can also convert a hex dump back to its original binary form and to do this we look further down the man page and see we need to use `-r` with `xxd` to reverse the hexdump.

Type `xxd -r data1.txt` and we see that our data file output has changed but it has not been saved so we need to tell `xxd` to convert and write this info to a new file for us.

Type `xxd -r data.txt newdata`

What this command does is it converts our file back to binary with `xxd -r` command and creates a new file called `newdata` without a `.txt` extension.

Step 9.

We now have a new file without an extension and we need to find out what kind of file it is. We can use the `file` command to do this.

Type `file newdata`

We see that we have a gzip file.

Step 10.

Type **man gzip** and we find that gzip is a compression tool that uses .gz as a default file extension and when used with -d will decompress a file.

We can now add our .gz file extension to our file by using the move command.

Type **mv newdata newdata.gz**

Now we can uncompress our gzip file using **gzip -d**.

Type **gzip -d newdata.gz**

Step 11.

We can now check to see what type of file was uncompressed.

Type **file newdata**

We see we now have a **bzip2** file.

Step 12.

Lets do the same as before type **man bzip2** to find what extension bzip2 uses and what option to use to decompress. We see we need to use .bz2 and -d with our bzip2 command to decompress.

Type **mv newdata newdata.bz2** to add our bz2 file extension.

Type **bzip2 -d newdata.bz2** to uncompress our bzip2 file

Step 13.

We can now check to see what type of file was uncompressed.

Type **file newdata**

We see we now have a **gzip** file again.

Step 14.

Lets do the same again but this time we already know what we use for gzip files.

Type **mv newdata newdata.gz** to add our gz file extension.

Type **gzip -d newdata.gz** to uncompress our gzip file.

Step 15.

We can now check to see what type of file was uncompressed.

Type **file newdata**

We see we now have a POSIX **tar** archive.

Step 16.

Type **man tar** and we find that tar is an archiving utility and when used with -x it will extract the data, -v do it verbosely, -f will use archive file.

Type **tar -xvf newdata**

We get data5.bin as an extracted file.

Step 15.

Now let's see what type of file this data5.bin is.

Type **file data5.bin**

Again we see we now have a POSIX tar archive. Like above lets untar it.

Type **tar -xvf data5.bin**

We get data6.bin as an extracted file.

Step 15.

Now let's see what type of file this data6.bin is.

Type **file data6.bin**

Again we have a bzip2 file so let's move and rename it.

Type **mv data6.bin data7.bz2**

Now uncompress the file.

Type **bzip2 -d data7.bz2**

We now have a new data7 file.

Step 16.

Now let's see what type of file our data7 is.

Type **file data7**

We see we have another tar file. Lets untar it like we have above.

Type **tar -xvf data7**

We get a data8.bin file.

Step 17.

Now let's see what type of file our data8.bin is.

Type **file data8.bin**

We see we have another gzip file so let's mv and rename with .gz extension.

mv data8.bin data9.gz

Step 18.

Now we ungzip out file like before.

Type **gzip -d data9.gz**

Then we check the uncompressed file type again.

Type **file data9**

We see we now have a readable ASCII file.

Step 19.

Now that we have an ASCII file let's do a cat on it to display it's contents.

type **cat data9** and we see our password for the next level is displayed: The password is

wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw

```

bandit12@bandit:/tmp/level2$ datal.txt -b
datal.txt: command not found
bandit12@bandit:/tmp/level2$ file -b datal.txt
ASCII text
bandit12@bandit:/tmp/level2$ xxd -r datal.txt newdata
bandit12@bandit:/tmp/level2$ ls
datal.txt  newdata
bandit12@bandit:/tmp/level2$ file -b newdata
gzip compressed data, was "data2.bin", last modified: Sun Apr 23 18:04:23 2023, max compression
, from Unix, original size modulo 2^32 591
bandit12@bandit:/tmp/level2$ man gzip
bandit12@bandit:/tmp/level2$ mv newdata.gz
mv: missing destination file operand after 'newdata.gz'
Try 'mv --help' for more information.
bandit12@bandit:/tmp/level2$ mv newdata newdata.gz
bandit12@bandit:/tmp/level2$ gzip -d newdata.gz
bandit12@bandit:/tmp/level2$ file newdata
newdata: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/level2$ file -b newdata
bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/level2$ bzip2 -d newdata.bz2
bzip2: Can't open input file newdata.bz2: No such file or directory.
bandit12@bandit:/tmp/level2$ mv newdata newdata.bz2
bandit12@bandit:/tmp/level2$ ls
datal.txt  newdata.bz2
bandit12@bandit:/tmp/level2$ bzip2 -d newdata.bz2
bandit12@bandit:/tmp/level2$ file -b datal.txt
ASCII text
bandit12@bandit:/tmp/level2$ ls
datal.txt  newdata
bandit12@bandit:/tmp/level2$ file -b newdata
gzip compressed data, was "data4.bin", last modified: Sun Apr 23 18:04:23 2023, max compression
, from Unix, original size modulo 2^32 20480
bandit12@bandit:/tmp/level2$ file newdata
newdata: gzip compressed data, was "data4.bin", last modified: Sun Apr 23 18:04:23 2023, max co
mpression, from Unix, original size modulo 2^32 20480
bandit12@bandit:/tmp/level2$ mv newdata newdata.gz
bandit12@bandit:/tmp/level2$ ls
datal.txt  newdata.gz
bandit12@bandit:/tmp/level2$ gzip -d newdata.gz
bandit12@bandit:/tmp/level2$ ls
datal.txt  newdata
bandit12@bandit:/tmp/level2$ file newdata
newdata: POSIX tar archive (GNU)
bandit12@bandit:/tmp/level2$ tar -xvf newdata
data5.bin
bandit12@bandit:/tmp/level2$ tar -xvf data5.bin
data6.bin
bandit12@bandit:/tmp/level2$ file -b data6.bin
bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/level2$ file -b data6.bin
bzip2 compressed data, block size = 900k

```

```

bandit12@bandit:/tmp/level2$ file -b data6.bin
bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/level2$ mv data6.bin data7.bz2
bandit12@bandit:/tmp/level2$ bzip2 -d data7.bz2
bandit12@bandit:/tmp/level2$ ls
datal.txt  data5.bin  data7  newdata
bandit12@bandit:/tmp/level2$ file -b data7
POSIX tar archive (GNU)
bandit12@bandit:/tmp/level2$ tar -xvf data7
data8.bin
bandit12@bandit:/tmp/level2$ mv data8.bin data.9.gz
bandit12@bandit:/tmp/level2$ gzip -d data9.gz
gzip: data9.gz: No such file or directory
bandit12@bandit:/tmp/level2$ ls
datal.txt  data5.bin  data7  data.9.gz  newdata
bandit12@bandit:/tmp/level2$ gzip -d data.9.gz
bandit12@bandit:/tmp/level2$ cat data.9
The password is wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw

```

wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw