Graphical user interface, text, application, email

Description automatically generated

**Solution:**

Text

Description automatically generated

**Explanation:**

Here in this level 10 of bandit we are required to find the password

Which is saved in a file **“data.txt”** which has a base64 encoded data.

“base64”. Base64 is a group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation. We can decode this data using “base64” command that is present on \*nix systems.

Text

Description automatically generated

**Solution:**

Text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

**Explanation:**

Here in level 11 of bandit we are told that the password is stored in file “data.txt” which is rotated by 13 positions.

So after extracting the rotated text we can turn it back to meaningful password using CyberChef website.

Text

Description automatically generated

**Solution:**

Text

Description automatically generated

**Explanation:**

In the level 12 of bandit over the wire we have been informed regarding a file “data.txt” which has been compressed multiple times in “gzip” and “bzip2” so we can use the commands like “bzip2 -d filename” and “gzip id filename” in order to decompress these files.

We have been given “.tar” files which can be also be decompressed using “tar xf filename.tar” . We have to repeat these steps couple of times in order to achieve a file which has ASCII text in order to obtain the password.

**Text, letter

Description automatically generated**

**Solution:**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Explanation:**

In this level we have been told that the password is saved in **“/etc/bandit\_pass/bandit14”** which can only be accessed by bandit14 user, so we have been provided with a private ssh key **“sshkey.private”** that we can use to get into next level using

**“ssh -I sshkey.private bandit12@localhost”,** and then we can read the password in **“etc/bandit\_pass/bandit14”** using “cat **etc/bandit\_pass/bandit14”.**

**Graphical user interface, text, application, email

Description automatically generated**

**Solution:**

**Text

Description automatically generated**

**Explanation:**

In this level, the challenge is similar to the previous one. However, it requires using SSL encryption. In other words, we need to submit the bandit15’s password to localhost on port 30001 through an encrypted communication.

**Text, letter

Description automatically generated**

**Solution:**

**Text

Description automatically generated**

**Explanation:**

In this level we have been given a simple task to submit the password obtained in the previous level in the local host. This could be done using a command “ncat”. Netcat is a simple UNIX utility which read and writes data across network connection, so we can use **“ncat –ssl localhost 30001” to get the password”.**

**Text, letter

Description automatically generated**

**Solution:**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Explanation:**

* Bandit17’s credentials can be retrieved by connecting to the localhost on an open port & submit bandit16’s password. The port range from 31000 to 32000 & requires SSL. As such, we have to perform port scanning to identify an open port which is accepting SSL.
* Then we will be checking which of the following ports has the value among the given ports using, **“cat /etc/bandit\_pass/bandit16 | openssl s\_client -connect localhost(hostname).**
* Finally after checking the port “31790”, we will receive a long text which we will be further copying to the “vim bar” after exiting the bandit server.
* **“vim sshkey.private”** and paste the key which has been found in the above port 31790.
* After getting done with the vim bar we can again move back to the bandit server and use the following set of commands in order to get the passwords
* **“cat sshkey.private**”( we received warning because, the sshkey.private file is readable to ANYONE. To fix this, simply change the sshkey.private file’s permission using chmod command to only allow the owner, the ability to read and write access. This is done by executing chmod 400 sshkey.private command.
* **“ ssh -i sshkey.private”**
* **“cat /etc/bandit\_pass/bandit17**

Graphical user interface, text

Description automatically generated

**Solution:**

**Text

Description automatically generated**

**Explanation:**

We know that both the files differ in only one line and that line consist of the password that we require. We can view the changes that have been made in files using the **“diff”** command.

**Text

Description automatically generated**

**Solution:**

**Text

Description automatically generated**

**Explanation:**

This level of bandit was a bit more interesting as it has been modified to end the bandit server as soon as one login’s the server bandit18 server, so here we will be trying to get into the bandit18 server using a **‘pseudo terminal’** using the command.

**“ssh** [**bandit18@bandit.labs.overthewire.org**](mailto:bandit18@bandit.labs.overthewire.org) **-p 2220 -t /bin/sh”**

Further we can use the cat readme command to find the password for the next level.

**Text

Description automatically generated**

**Solution:Text

Description automatically generated**

**Explanation:**

Run **“ls -l“** in the current working directory, to identify the file with a “**setuid**” file permission setting.

In the file **“bandit20-do”,** has an **s** bit located in the **“User”** permission classes. Additionally, the **owner**of the file **“bandit20-do”** is **“bandit20”.** The red highlight signifies that this file has elevated permissions. Hence, any commands executing with **“./bandit20-do”** will run as user bandit**20** instead of bandit**19**.

**Obtained Passwords:**

Level 10- truKLdjsbJ5g7yyJ2X2R0o3a5HQJFuLk

Level 11- IFukwKGsFW8MOq3IRFqrxE1hxTNEbUPR

Level 12- 5Te8Y4drgCRfCx8ugdwuEX8KFC6k2EUu

Level 13- 8ZjyCRiBWFYkneahHwxCv3wb2a1ORpYL

Level 14- 4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e

Level 15- BfMYroe26WYalil77FoDi9qh59eK5xNr

Level 16- cluFn7wTiGryunymYOu4RcffSxQluehd

Level 17- xLYVMN9WE5zQ5vHacb0sZEVqbrp7nBTn

Level 18- **kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd**

Level 19- IueksS7Ubh8G3DCwVzrTd8rAVOwq3M5x

Level 20- GbKksEFF4yrVs6il55v6gwY5aVje5f0j