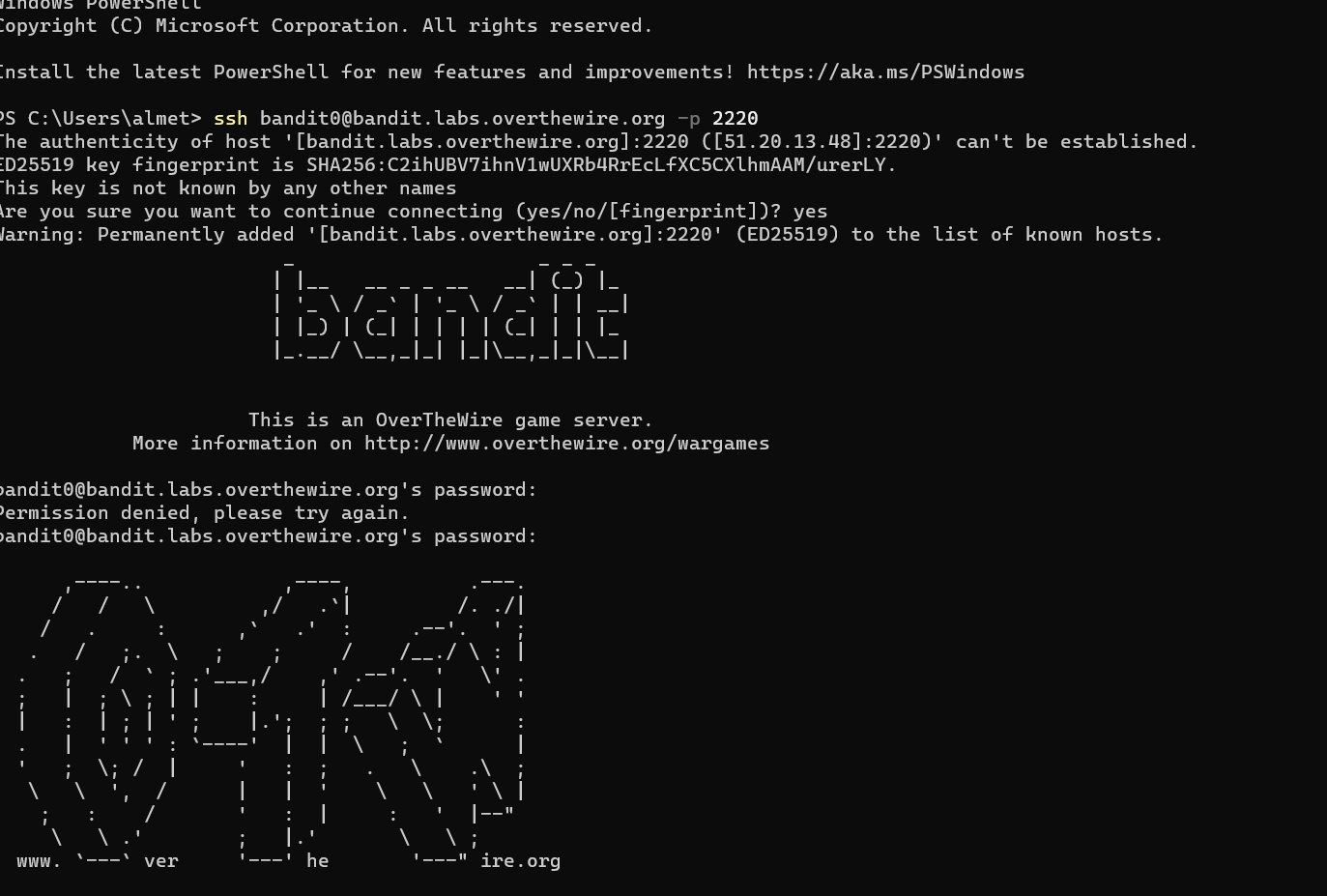
OVER THE WIRE

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](https://overthewire.org/wargames/bandit/bandit1.html) page to find out how to beat Level 1.

*#SOLUTION*

**

* ssh <username>@<remote> here username is bandit0.
* If you want to specify a port, add -p 0000, (replace 0000 with the desired port number). Here port is 2220.
* You will be asked for your [password](https://www.wikihow.com/Create-a-Password-You-Can-Remember) once the connection is established. You will not see the cursor move or any characters input when you type your password.

*#FLAG*

bandit0

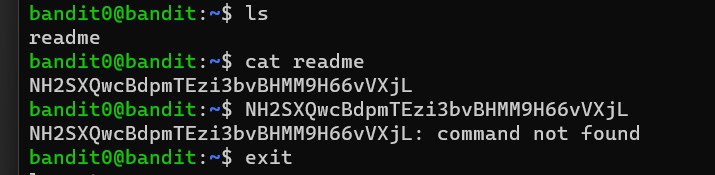
LEVEL 0 🡪 LEVEL 1

The password for the next level is stored in a file called **readme** located in the home directory. Use this password to log into bandit1 using SSH. Whenever you find a password for a level, use SSH (on port 2220) to log into that level and continue the game.

*#SOLUTION*

Here two commands are used :

1. ls : This SSH command is used to list all files and directories.
2. cat : On getting the file name we use this command to display the content of a file.



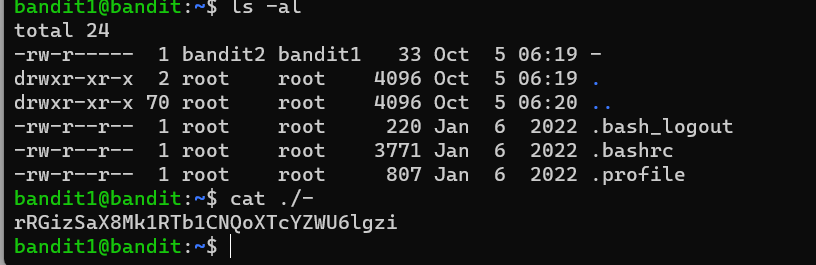
*#FLAG*

NH2SXQwcBdpmTEzi3bvBHMM9H66vVXjL

LEVEL 1 🡪 LEVEL 2

The password for the next level is stored in a file called **-** located in the home directory.

*#SOLUTION*



When working with filenames, a leading dot is the prefix of a "hidden" file, a file that an [ls](https://tldp.org/LDP/abs/html/basic.html#LSREF) will not normally show. Hence, we use **ls -al** command to show hidden files. On locating the ‘-‘file we display its contents by using **cat** command.

*#FLAG*

rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi

LEVEL 2 🡪 LEVEL 3

The password for the next level is stored in a file called **spaces in this filename** located in the home directory.

*#SOLUTION*

A screenshot of a computer screen

Description automatically generated

1. First we list all files including hidden files using **‘ls-al’** command.
2. Next it doesn’t understand ‘**cat spaces in this filename** ‘as a single hence we use **‘\’** followed by a space .

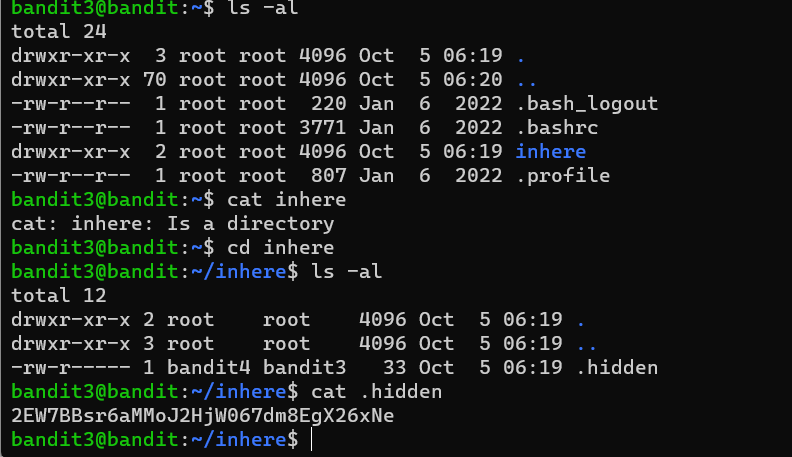
*#FLAG*

aBZ0W5EmUfAf7kHTQeOwd8bauFJ2lAiG

LEVEL 3🡪 LEVEL 4

The password for the next level is stored in a hidden file in the **inhere** directory.

*#SOLUTION*



1. First, we use **‘ls-al’** command to list all files.
2. On using **cat** command, it displays a message that **inhere is a directory**.
3. So now we use **cd** command which is used to change between the directories.
4. On listing the files of inhere directory we get a hidden file.
5. On listing the contents of the hidden file, we get the password.

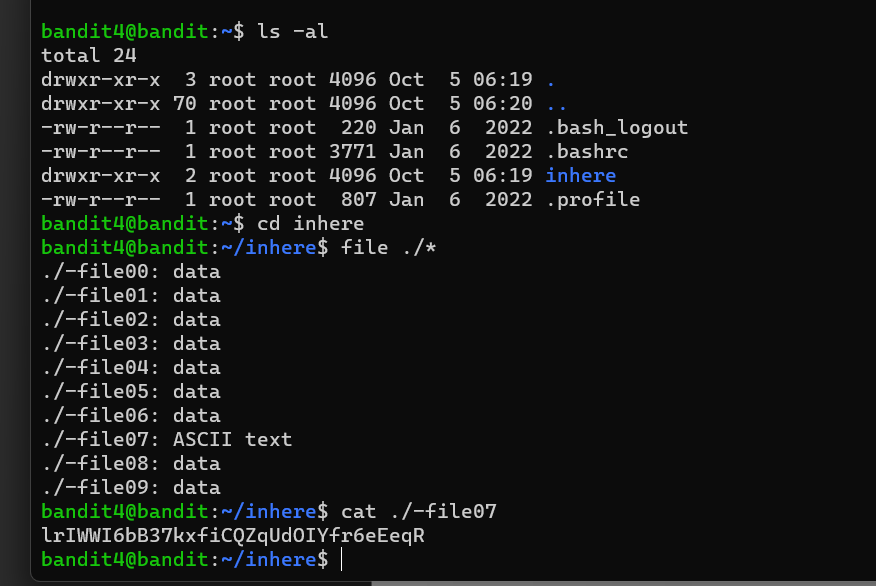
*#FLAG*

2EW7BBsr6aMMoJ2HjW067dm8EgX26xNe

LEVEL 4🡪 LEVEL 5

The password for the next level is stored in the only human-readable file in the **inhere** directory. Tip: if your terminal is messed up, try the “reset” command.

*#SOLUTION*



1. First, we use **‘ls-al’** command to list all files.
2. On using **cat** command, it displays a message that **inhere is a directory**. So now we use **cd** command which is used to change between the directories.
3. Now we use ‘**file ./\*’** which lists all the type of files in the inhere directory .
4. After this we can see that ‘/-file07’ contains ASCII text which is human readable.

*#FLAG*

lrIWWI6bB37kxfiCQZqUdOIYfr6eEeqR

LEVEL 5🡪 LEVEL 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

*#SOLUTION*

A screenshot of a computer program

Description automatically generated

1. First, we use **‘ls-al’** command to list all files.
2. On using **cat** command, it displays a message that **inhere is a directory**. So now we use **cd** command which is used to change between the directories.
3. Now using ‘**find**’ followed :

**size +N/-N :** Search for files of ‘N’ blocks; ‘N’ followed by **‘c’** can be used to measure the size in characters; ‘+N’ means size > ‘N’ blocks and ‘-N’ means size < ‘N’ blocks.

**Followed by ‘! executable ‘ command and then ‘grep’ command is used to search specific phrase in file.**

*#FLAG*

**P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU**

LEVEL 6🡪 LEVEL 7

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

*#SOLUTION*

A screen shot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

1. Here we use the command ‘**find / -type f -user username -group groupname -size size given’** here username is : bandit7 , groupname is : bandit6 and size is 33 bytes . bytes is represented by ‘**c’** .
2. On getting a list of files we look for password file and then use **‘cat’**  command to display the contents of this file which gives us the password .

*#FLAG*

z7WtoNQU2XfjmMtWA8u5rN4vzqu4v99S

LEVEL 7🡪 LEVEL 8

The password for the next level is stored in the file **data.txt** next to the word **millionth.**

*#SOLUTION*

****

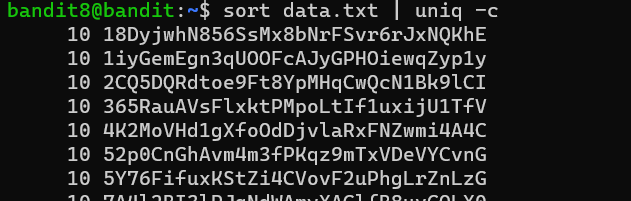
1. **First we list the files of the directory using ‘ls -al’**
2. **On finding data.txt we find the contents of the file using ‘cat’.**
3. **Now we use ‘grep’ command which helps to search specific phrases in the file / directory.**

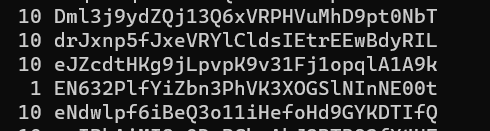
*#FLAG*

**TESKZC0XvTetK0S9xNwm25STk5iWrBvP**

LEVEL 8🡪 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.

*#SOLUTION*

**

1. **First we list the files of the directory using ‘ls -al’.**
2. **On finding data.txt we list the files in it using ‘cat ‘ command .**
3. **Now we sort the files by using ‘sort’ command then we use ‘|’ to combine both statements it after which we use uniq command . uniq** can’t detect the duplicate lines unless they are adjacent to each other. The content in the file must be therefore sorted before using **uniq.** Now to get the count od lines we use **‘-c’ .**

*#FLAG*

EN632PlfYiZbn3PhVK3XOGSlNInNE00t

LEVEL 9🡪 LEVEL 10

The password for the next level is stored in the file **data.txt** in one of the few human-readable strings, preceded by several ‘=’ characters.

*#SOLUTION*

**

1. **First we list the files of the directory using ‘ls -al’.**
2. **On finding data.txt we list the files in it using ‘cat ‘command .**
3. The strings command provides four types of encoding options. **strings -e s file\_name.** **s:** 7-bit byte (used for ASCII, ISO 8859).
4. Now we use ‘**grep’** command to get strings preceded by ‘=’.

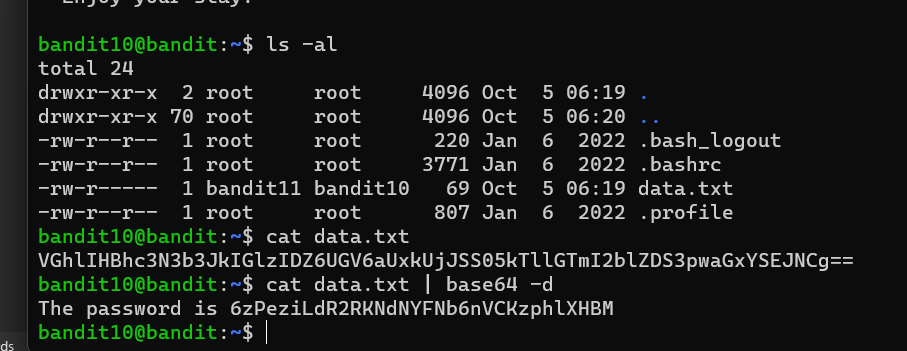
*#FLAG*

G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s

LEVEL 10🡪 LEVEL 11

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

*#SOLUTION*

**

1. First we list all files of the directory using ‘**ls -al’**.
2. Now we use the ‘**ca**t’ command to display the contents of the data.txt file. But it is encoded in base64.
3. Hence to decode it we use **base64 -d** command which decodes the text to normal human readable text.

*#FLAG*

6zPeziLdR2RKNdNYFNb6nVCKzphlXHBM

Level11->Level12

tr command replaces the letters in the first sequence with corresponding letter from the second sequence.(Consulted stack exchange in order to learn how to specify the sequences properly)

tr 'A-Za-z' 'N-ZA-Mn-za-m' <data.txt

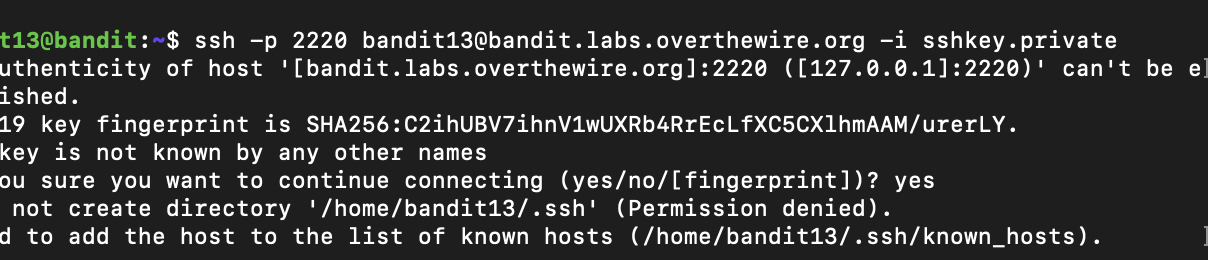
Level12->Level13

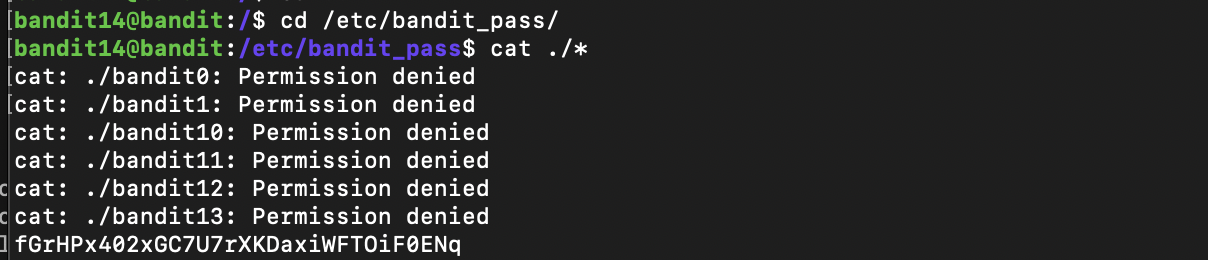
First convert the hexdump of the file to text using the xxd command. Then find the compression of the file usng the file command. Accordingly rename the file giving the extension of the compression type. Decompress the file and then repeat this process multiple times till the file becomes and ASCII text file.



Level13->Level14

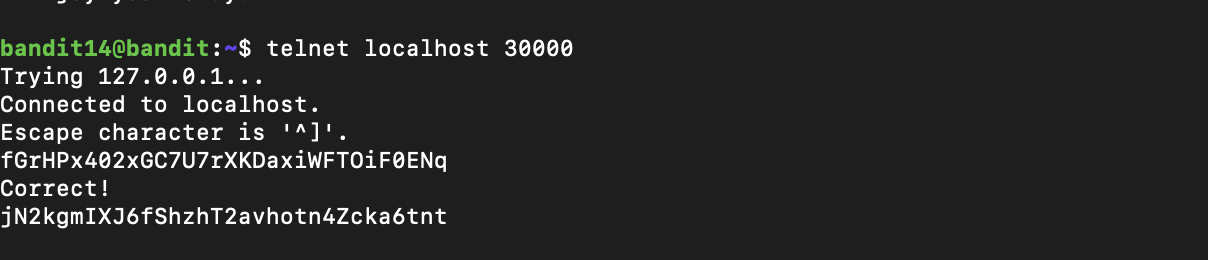
Use the sshkey.private that is present to login to bandit 14.





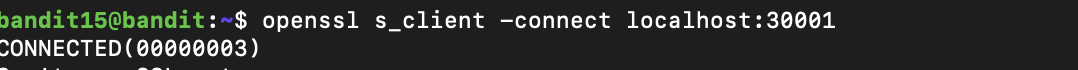
Level14->Level15

First tried using ssh but that didnt work. After analyzing the commands given in the level used telnet command to connect to the specified port.



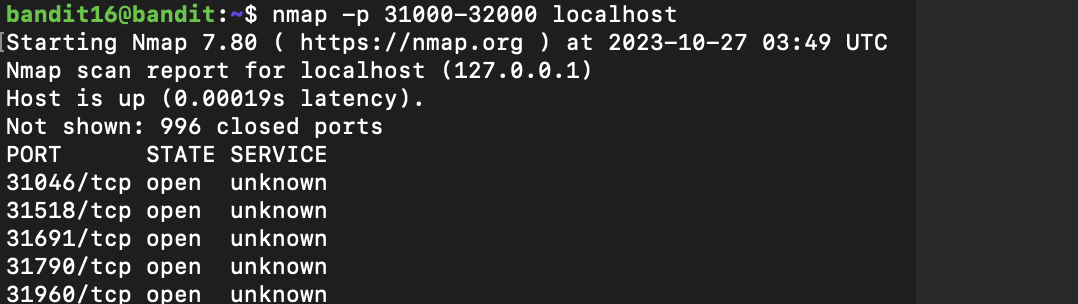
Level15->Level16

Used the openssl command along with s\_client and specified the port by using the -connect localhost:portname to retrieve the password.



Level16->Level17

First consulted how to specify a port range using nmap.

After that use the openssl s\_client command to check which port gives the credential to the next level. Using port 31790 a private key was given. After copying the key to a file i tried to use it to login to the next level. However a permission error was given. After changing the permission to 600 it worked as it is the required permission for RSA private keys.



Level17->Level18

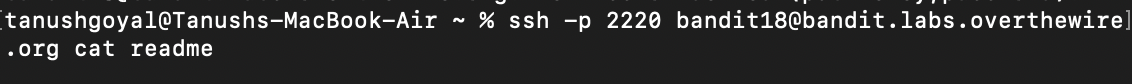
First concatenated the two files. Using piping sorted the given concatenated file and used the uniq -c command to get the no of times each line was repeated.





Level18->Level19

Consulted the walkthrough. As everytime .bashrc file closes ssh connection we can use the cat readme command along with the ssh command so that the .bashrc file is not triggered.

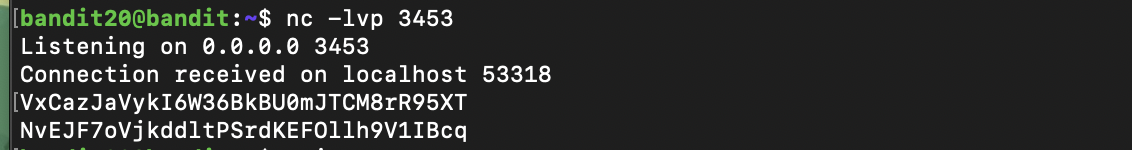


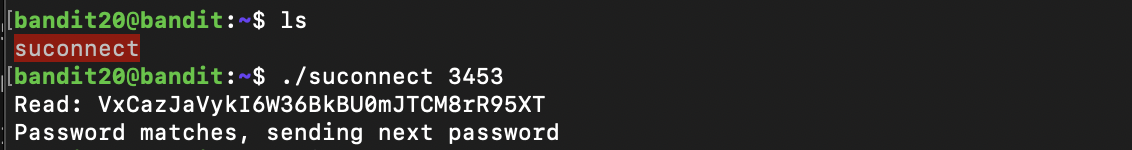
Level19->Level20

By using the bandit20-do which is the setuid binary file we find the password which is present in the bandit20 file in the given directory.

  
Level20->Level21

In this level we use nc command to connect to a random port. After that we use the setuid binary file to create a connection to the earlier specified port. Through the first window we send the password of the earlier level to get the current password.(Consulted the walkthrough)





Level21->Level22

First we change the directory to /etc/cron.d. After that using cat cronjob\_bandit22 command

we find the command that is executed at an interval of 1 second. Used cat first on the file that is being redirected to /dev/null. We get another file name and using cat again on that we get the password.(Consulted the walkthrough)

