**Overthewire-Bandit**

Name: Meeti

Levels: 0-12

Overview

Conquered the levels with the aid of man <command> (manual)

Read through each level, checked the helpful reading material and looked up the commands needed to solve the level in the manual.

Some important commands:

* cd: to change the working directory
* ls: to list the contents of a directory
* exit: to exit the directory
* file: shows file type
* find: to search for a particular type of file
* grep: searches for patterns in a file

Bandit level 0-1

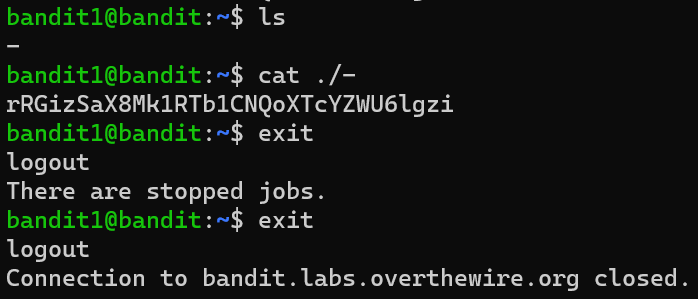
1. Used ls command to check the list of files in the current directory.
2. Used the command cat for this level which reads and displays the contents of the file, thus giving the password.
3. The exit command logs out from the bandit0 server.
4. Logged into the next level using ssh and entering the password.



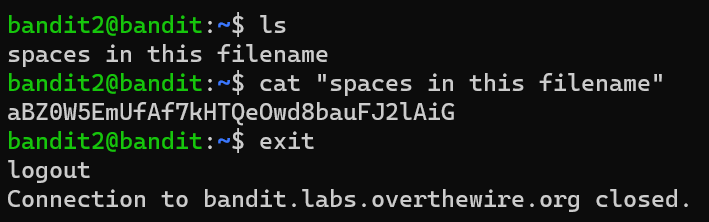


Bandit level 1-2

1. Used the ls command to list the files in the directory, and found the file ‘-‘
2. cat - did not display contents of the file.
3. Browsed through the helpful reading material and found the method in Google search for “dashed filename” in the stack overflow website which is the first search result.
4. To read a dashed filename, it should be preceded by a ./
5. cat ./- displayed the required password for the next level.
6. Further research led to understanding the fact that the cat command understands the ‘-’ as a synonym for stdin, which is the standard input

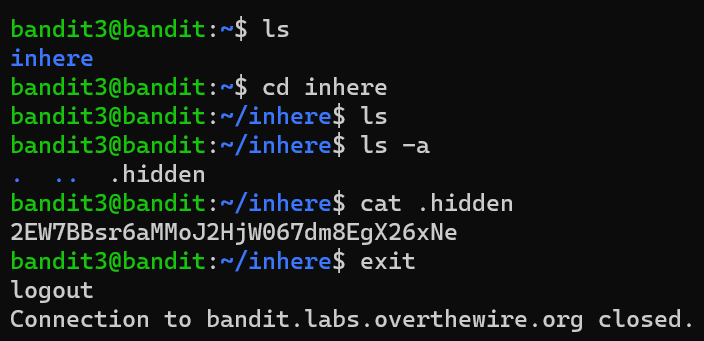


Bandit level 2-3

1. Used ls command to list out the files in the directory.
2. Tried surrounding the name with quotes to represent the sentence as a single entity, which worked, and displayed the password required to log in to level 3.
3. On browsing through the helpful reading material, discovered that putting a backslash ‘\’ before every space also works.

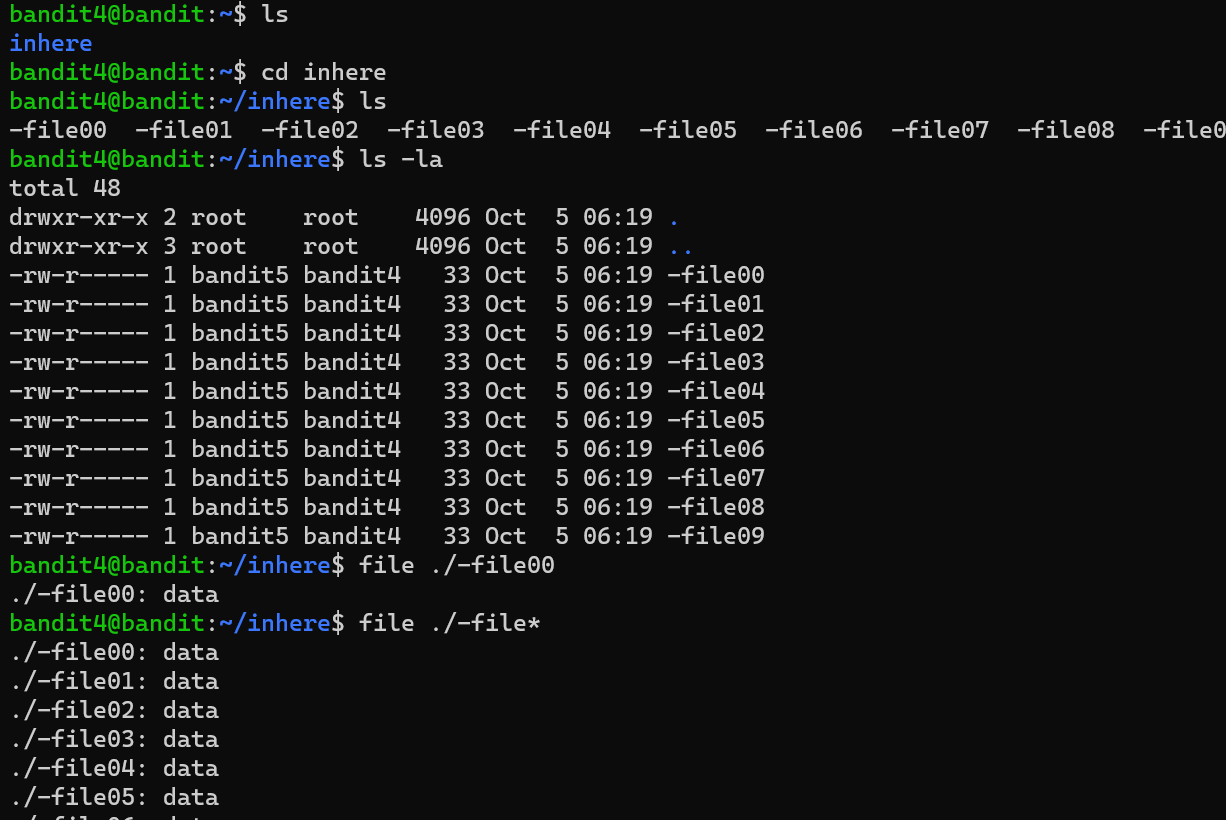
Bandit level 3-4

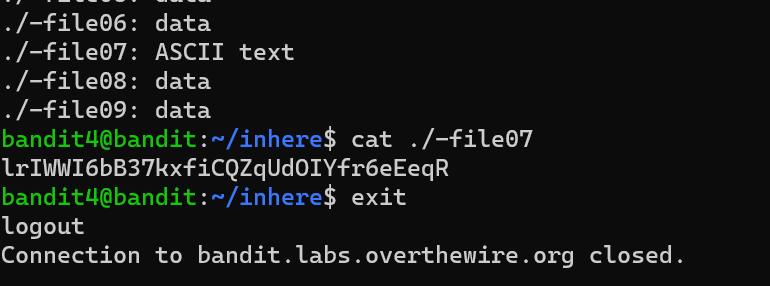
1. Used ls to list all the directories and files.
2. Used cd command to change the working directory to inhere.
3. Used ls to list out the files in the working directory, which did not display the hidden file.
4. Checked the ls command in the manual, and tried ls -a, where -a or -all displays all the entries, including hidden ones, those preceded by ‘.’
5. This displayed the file .hidden
6. Used cat command to display contents of the hidden file and obtained the password for level 4.



Bandit level 4-5

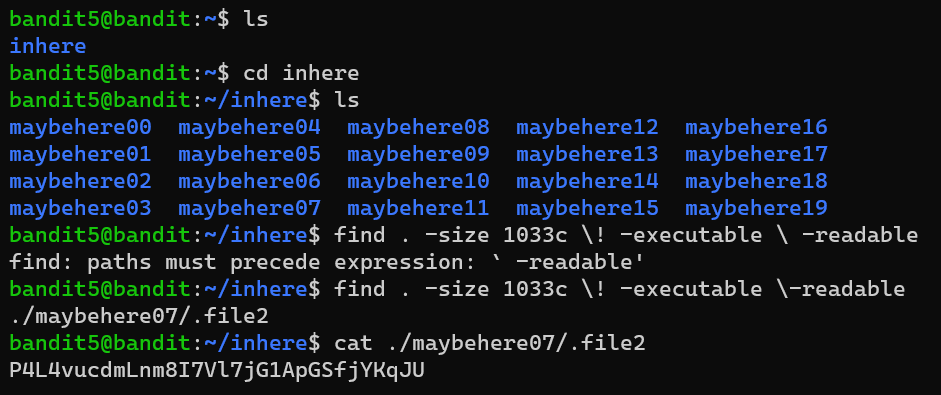
1. Used cd command to change working directory to inhere.
2. Used ls command and found several files with filename format ‘-file\*’
3. Checked the file command in the manual, which determines file type.
4. Checked the file type of each file by file ./-file\*, where \* is a wildcard character.
5. -file07 had ASCII text as data, which means it has human readable data.
6. Used cat command to read and display the contents of -file07, and discovered the password needed for the next level.





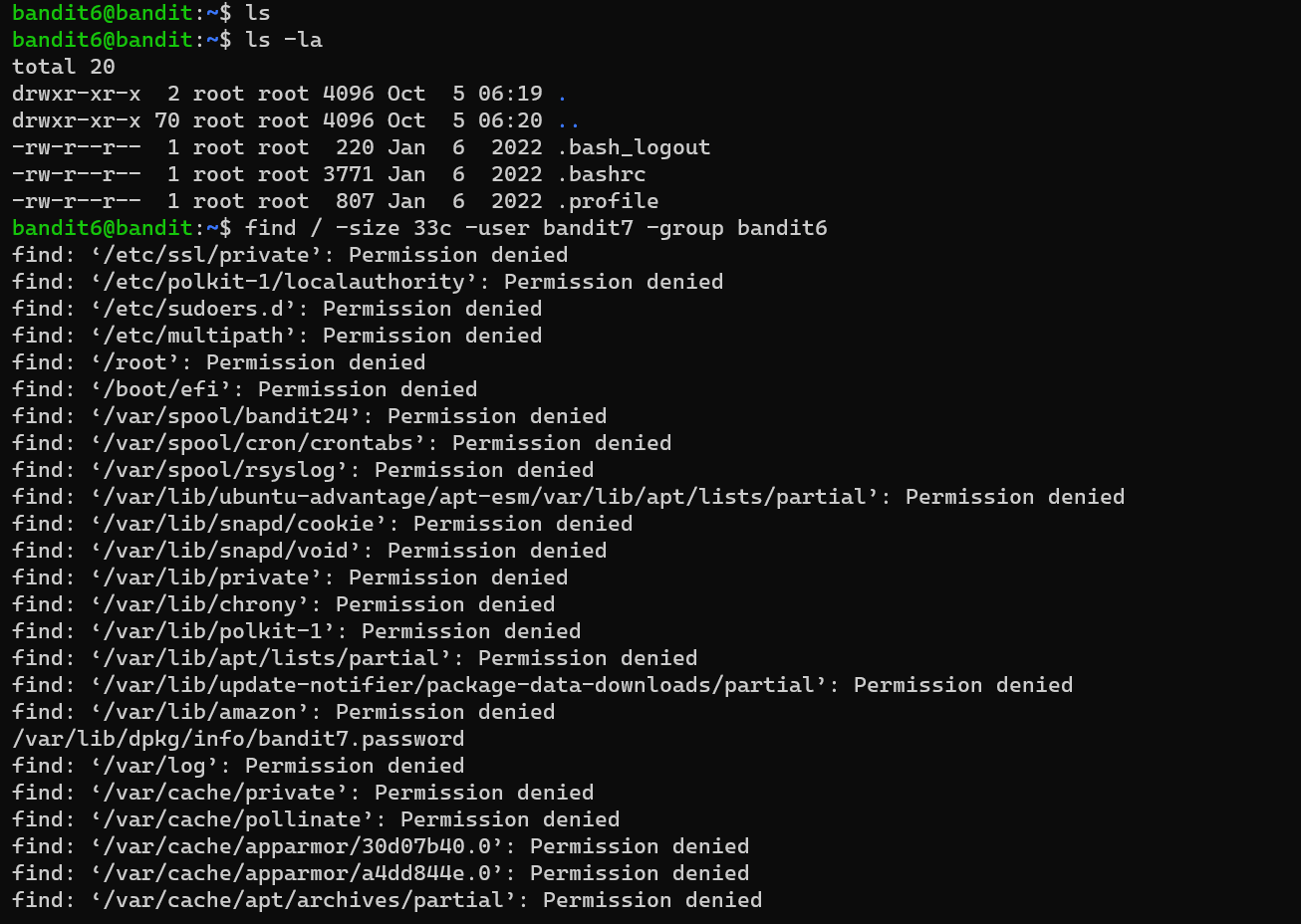
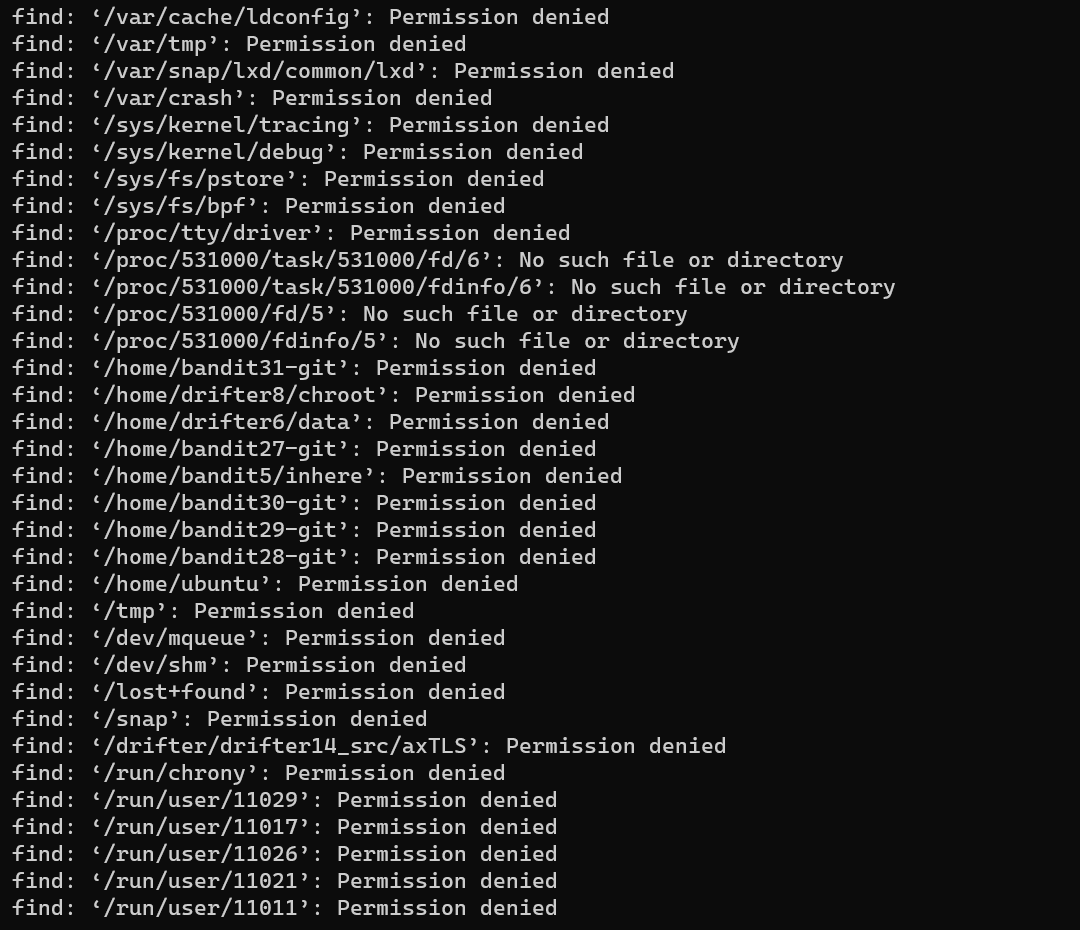
Bandit level 5-6

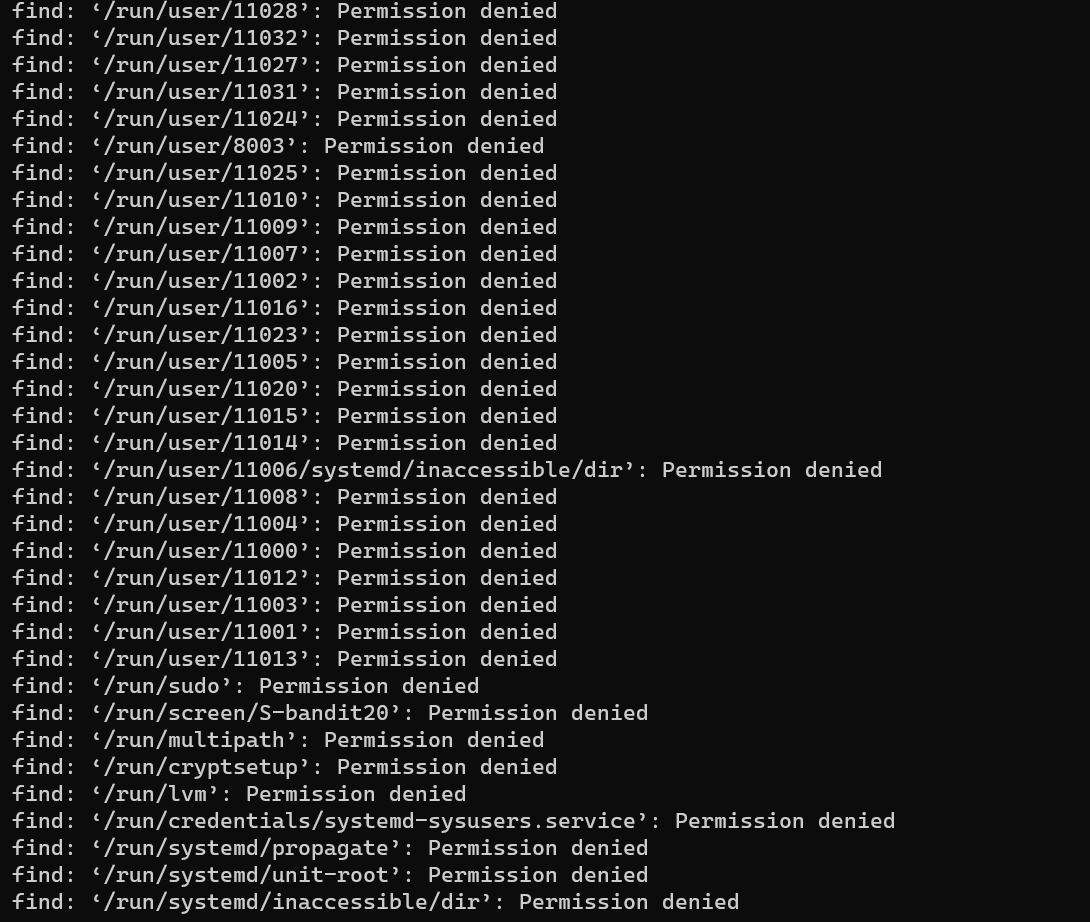
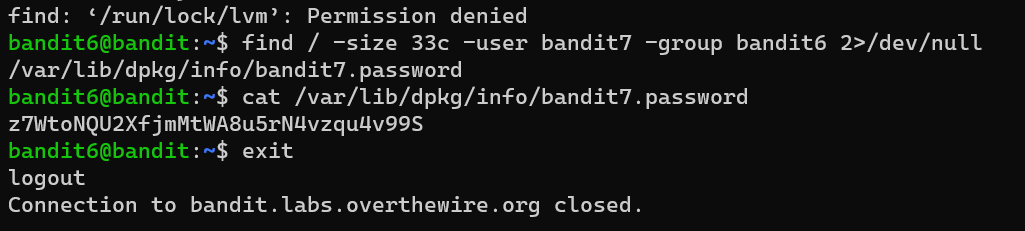
1. Used cd to change the working directory to inhere.
2. Used ls to display the list of entries in the directory.
3. Checked the find command in the manual.
4. Found expressions for finding a file with a particular file size, which is non executable (by prefixing with a ‘!’) and readable.
5. Directory maybehere07 → .file2 was the required file.
6. Used cat to display contents of the required file and hence got the password for the next level.



Bandit level 6-7

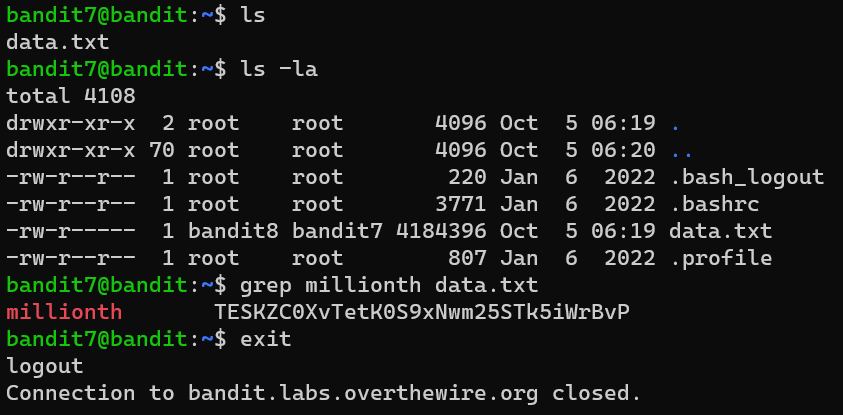
1. Used find command and specified the requirements.
2. Too much data not useful for the level was obtained.
3. 2>/dev/null discards errors which need not be displayed, simplifying the output.
4. /dev/null is a null device–a special type of virtual device, present in every Linux system, whose purpose is to discard anything sent to it and read the End of File (EOF).
5. 2 stands for standard error stream.
6. The required file is displayed, and is then concatenated to display the required password.



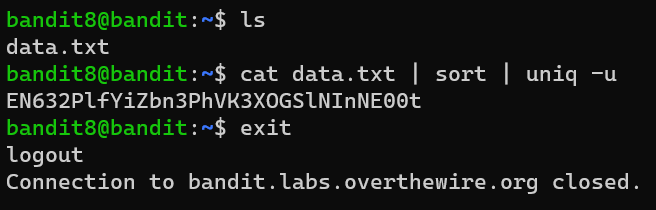


Bandit level 7-8

1. Checked the manual for grep command.
2. Used grep command to search for the word millionth in the file data.txt, which has the password following it.

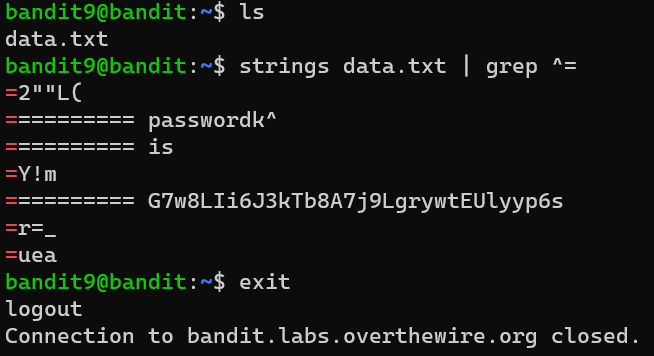
 Bandit level 8-9

1. Browsed through the helpful reading material for piping.
2. Understood that piping is done by the operator | which takes the output from the command on the left as input for the command on the right.
3. Checked manual for sort and uniq commands.
4. sort: sorts lines of text files
5. uniq: omits the repeated text lines.
6. Used uniq after sort to obtain the password for level 9.

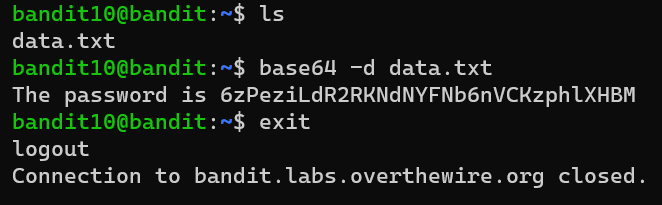


Bandit level 9-10

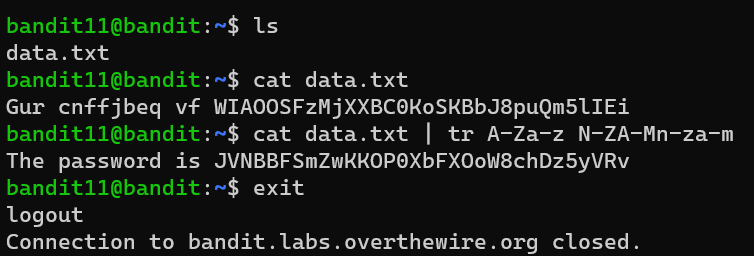
1. Checked manual for strings and grep commands.
2. strings: prints the sequences of printable characters in files, because human readable data was needed.
3. Grep was used with ^ operator which means that the line begins with the following character.
4. Found the password among the sequences printed as output.

 Bandit level 10-11

1. Checked the manual for the base64 command.
2. Found -d as an argument for decoding the data.
3. Using base64 -d <filename> got the password for level 11.

Bandit level 11-12

1. Read through the helpful reading material.
2. Found that Rot13 is a simple letter substitution cipher that replaces a letter with one 13 letters after it in the same case.
3. Further reading in the tr subtopic led to finding a way to implement the Rot13.
4. Concatenated the file and then deciphered the text by piping, thus obtaining the password needed.



Bandit level 12-13

1. Used mkdir to create a temporary directory.
2. Used cp to copy data.txt to the temporary directory.
3. Changed working directory to the temporary directory using cd command.
4. Used xxd -r command to revert the file from its hex dump format back into the original data.
5. Found the filetype and renamed the file with appropriate extension to perform the necessary command on it.
6. For gzip compressed file:

mv <filename> <filename>.gz

gzip -d <filename>.gz, where -d is to decompress

1. For bzip2 compressed file:

mv <filename> <filename>.bz2

bzip2 -d <filename>.bz2

1. For POSIX tar archive:

mv <filename> <filename>.tar

tar xf <filename>.tar, where xf stands for extract file

This extracts a separate file, here files with .bin extension.

1. The final file containing the password was data8.bin.

