Bandit 0 – Accessing the Server

```
└─$ ssh bandit0@bandit.labs.overthewire.org -p 2220
```

The first thing we need to do to complete the Bandit CTF challenges is to use the SSH program to login

Bandit 0 – Accessing the Server

```
└─$ ssh bandit0@bandit.labs.overthewire.org -p 2220
```

This command uses `SSH` to log in as the `bandit0` user to the `bandit.labs.overthewire.org` server on port `2220`

Bandit 0 – Accessing the Server

```
[root@localhost ~]# ssh bandit0@bandit.labs.overthewire.org -p 2220
The authenticity of host '[bandit.labs.overthewire.org]:2220 ([16.16.163.126]:22
20)' can't be established.
ECDSA key fingerprint is SHA256:IJ7FrX0mKSSHTJ63ezxjqtnOE0Hg116Aq+v5mN0+HdE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

If asked if we are sure we want to continue connecting, we need to type `yes`, then press enter

Bandit 0 – Accessing the Server

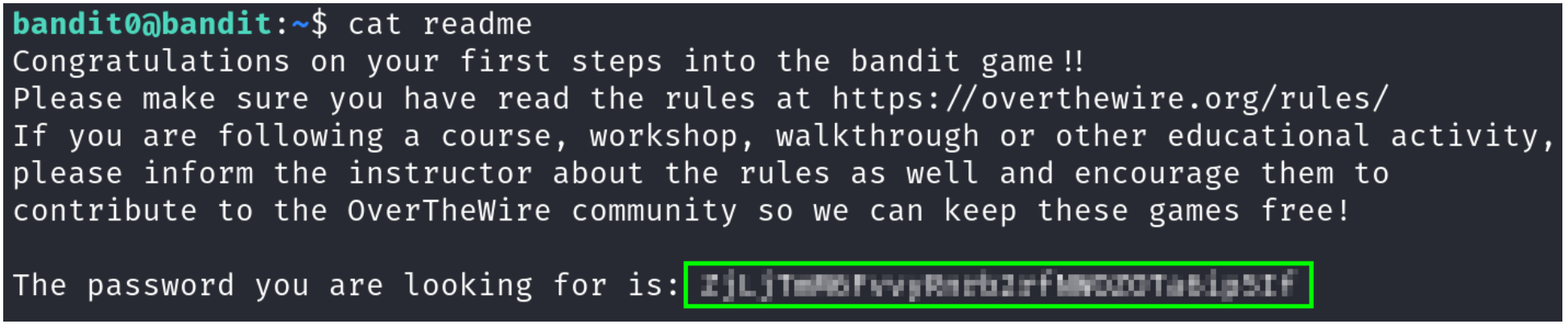
```
bandit0@bandit.labs.overthewire.org's password:
```

Then we'll be asked to enter the user's password. Type in `bandit0`, then press enter. While you're typing in the password, you will not see any feedback from the terminal. This is normal

Bandit 0 – Listing Directory Contents

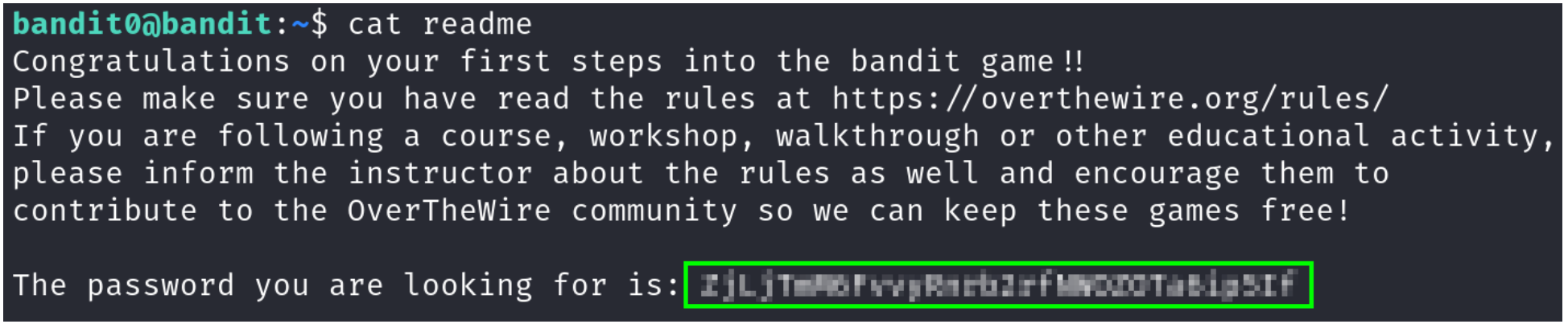
In Linux, the `ls` command is used to list out the contents of a directory. When used here, we see the `readme` file

Bandit 0 – Reading File Contents



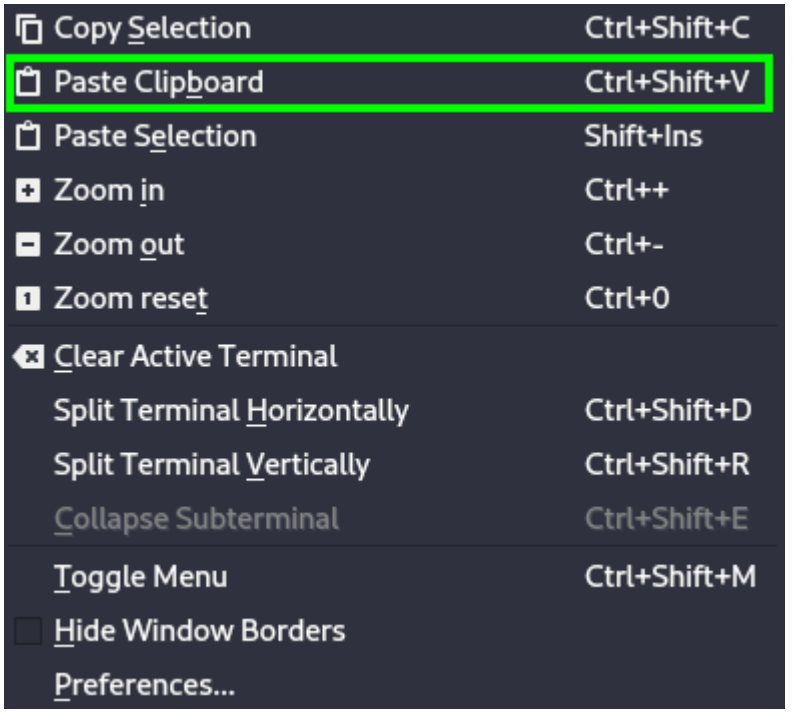
The command to read a file in Linux is the `cat` command, and the syntax is:
`cat <filename>` for example `cat readme`

Bandit 0 – Reading File Contents



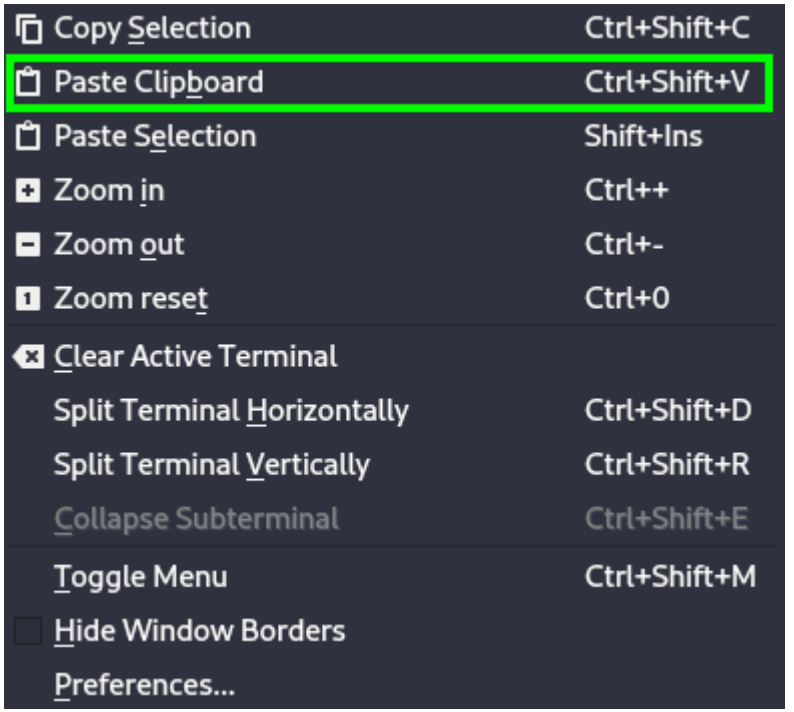
In the contents of the file, we see the password for the next level, which we will use SSH to login as the bandit1 user

Bandit 1 – Pasting in Passwords



When entering in the passwords for the Bandit CTF levels, we should paste in the password instead of keying it in

Bandit 1 – Pasting in Passwords



We can right-click then select `Paste Clipboard` or use the keyboard shortcut `Ctrl+Shift+V`

Bandit 1 – Clearing the Screen

If the screen becomes too cluttered in Linux, we can use the `clear` command to clear the screen and go back up to the top of the screen

Bandit 1 – Reading Files with Special Characters

```
bandit1@bandit:~$ cat -
s
```

This level requires us to read files with special characters, but if we try to read this file in the regular way, it doesn't work

Bandit 1 – Quitting Unresponsive Programs

If at any time a program becomes unresponsive in Linux, we can use the `Ctrl+C` keyboard shortcut to terminate the program

Bandit 1 – Reading Files with Special Characters

1) Filenames should not include spaces. We can use underscores if we want to use spaces, e.g., `my_file`

2) Filenames should not start with numbers, because certain numbers are treated as special characters in Linux

3) Filenames should never start with special characters

Bandit 1 – Reading Files with Special Characters

We can read files with special characters by referencing the exact directory where the file is. In Linux, the current directory is referenced with `./`

Bandit 1 – Reading Files with Special Characters

So to reference a file named – in the current directory, it would be `./-`

Bandit 2 – Reading Files with Spaces in the Name

```
bandit2@bandit:~$ cat spaces in this filename
cat: spaces: No such file or directory
cat: in: No such file or directory
cat: this: No such file or directory
cat: filename: No such file or directory
```

In this level, we need to read a file with spaces in its name. If we try to read this file normally, we won't be able to, since the Linux interprets the spaces as the end of one file name and the beginning of another

Bandit 2 – Reading Files with Spaces in the Name

```
bandit2@bandit:~$ cat spaces in this filename
cat: spaces: No such file or directory
cat: in: No such file or directory
cat: this: No such file or directory
cat: filename: No such file or directory
```

And this is why its not recommended to put spaces in filenames. However, there's a couple of methods we could use to reference filenames with spaces in them

Bandit 2 – Reading Files with Spaces in the Name

The first method is to wrap the name of the file in quotes, either single quotes or double quotes. This ensures that Linux will interpret everything in the quotes as a single object

Bandit 2 – Reading Files with Spaces in the Name

```
bandit2@bandit:~$ cat spaces\ in\ this\ filename
[redacted]
```

The second method to insert a backslash character before every space in the filename, which lets Linux know that the space is not the start of a new filename, but part of the current filename

Bandit 3 – Changing Directories

```
bandit3@bandit:~$ ls
inhere
bandit3@bandit:~$ cd inhere
```

In Linux, we can move into a directory by using the cd command. The syntax is `cd <directory_name>`, for example, `cd inhere`

Bandit 3 – Checking the Current Directory

If we want to check our current directory, we can use the `pwd` (present working directory) command. In Linux, all directories are start with a `/`, for example, the `/home/bandit3/inhere` directory

Bandit 3 – Hidden Files

```
bandit3@bandit:~/inhere$ ls -la
total 12
drwxr-xr-x 2 root    root    4096 Sep 19  2024 .
drwxr-xr-x 3 root    root    4096 Sep 19  2024 ..
-rw-r----- 1 bandit4 bandit3   33 Sep 19  2024 ...Hiding-From-You
```

In Linux, any file or directory that start with a `.` is a hidden file, which means that it won't appear when using the `ls` command in the regular way.

Bandit 3 – Hidden Files

```
bandit3@bandit:~/inhere$ ls -la
total 12
drwxr-xr-x 2 root    root    4096 Sep 19  2024 .
drwxr-xr-x 3 root    root    4096 Sep 19  2024 ..
-rw-r----- 1 bandit4 bandit3   33 Sep 19  2024 ...Hiding-From-You
```

The current directory in Linux is denoted as `.` and because its name starts with a dot, it is hidden by default. The same goes for the directory above the current one, which is `..`

Bandit 3 – Hidden Files

```
bandit3@bandit:~/inhere$ ls -la
total 12
drwxr-xr-x 2 root    root    4096 Sep 19  2024 .
drwxr-xr-x 3 root    root    4096 Sep 19  2024 ..
-rw-r----- 1 bandit4 bandit3   33 Sep 19  2024 ...Hiding-From-You
```

To see hidden files with the `ls` command, we have to use the command with an argument `-a`, which alters the output of the command by including hidden files in the output

Bandit 4 – Text versus Data Content

```
bandit4@bandit:~/inhere$ cat ./-file00
◆p◆◆&◆y◆,◆(jo◆.at◆:uf◆^◆◆◆@bandit4@bandit:~/inhere$
```

Computer files typically contain one of two types of content, human-readable text, or machine-readable data

Bandit 4 – Text versus Data Content

```
bandit4@bandit:~/inhere$ cat ./-file00
◆p◆◆&◆y◆,◆(jo◆.at◆:uf◆^◆◆◆@bandit4@bandit:~/inhere$
```

Files with data content are meant to be processed by computer software, and will not be readable if read by using the `cat` command

Bandit 4 – Text versus Data Content

```
bandit4@bandit:~/inhere$ ls -l
total 40
-rw-r----- 1 bandit5 bandit4 33 Sep 19  2024 -file00
-rw-r----- 1 bandit5 bandit4 33 Sep 19  2024 -file01
-rw-r----- 1 bandit5 bandit4 33 Sep 19  2024 -file02
-rw-r----- 1 bandit5 bandit4 33 Sep 19  2024 -file03
-rw-r----- 1 bandit5 bandit4 33 Sep 19  2024 -file04
-rw-r----- 1 bandit5 bandit4 33 Sep 19  2024 -file05
-rw-r----- 1 bandit5 bandit4 33 Sep 19  2024 -file06
-rw-r----- 1 bandit5 bandit4 33 Sep 19  2024 -file07
-rw-r----- 1 bandit5 bandit4 33 Sep 19  2024 -file08
-rw-r----- 1 bandit5 bandit4 33 Sep 19  2024 -file09
```

In this level, we're meant to find out which file contains text contents, not binary. It would be tedious to look through the files one by one, but there's a way to scan all of the files at once

Bandit 4 – File Command

```
bandit4@bandit:~/inhere$ file ./-file00
./-file00: data
```

The `file` command in Linux is used to return the type of contents in a file

Bandit 4 – The * Wildcard Character

```
bandit4@bandit:~/inhere$ file ./*
./-file00: data
./-file01: data
./-file02: data
./-file03: data
./-file04: data
./-file05: data
./-file06: data
./-file07: ASCII text
./-file08: data
./-file09: data
```

The * special character in Linux is used as a shorthand for “all files”, and we can run a command like the one above to combine the file command with the * wildcard character to run the command on all the files in the directory