ID: 1805113

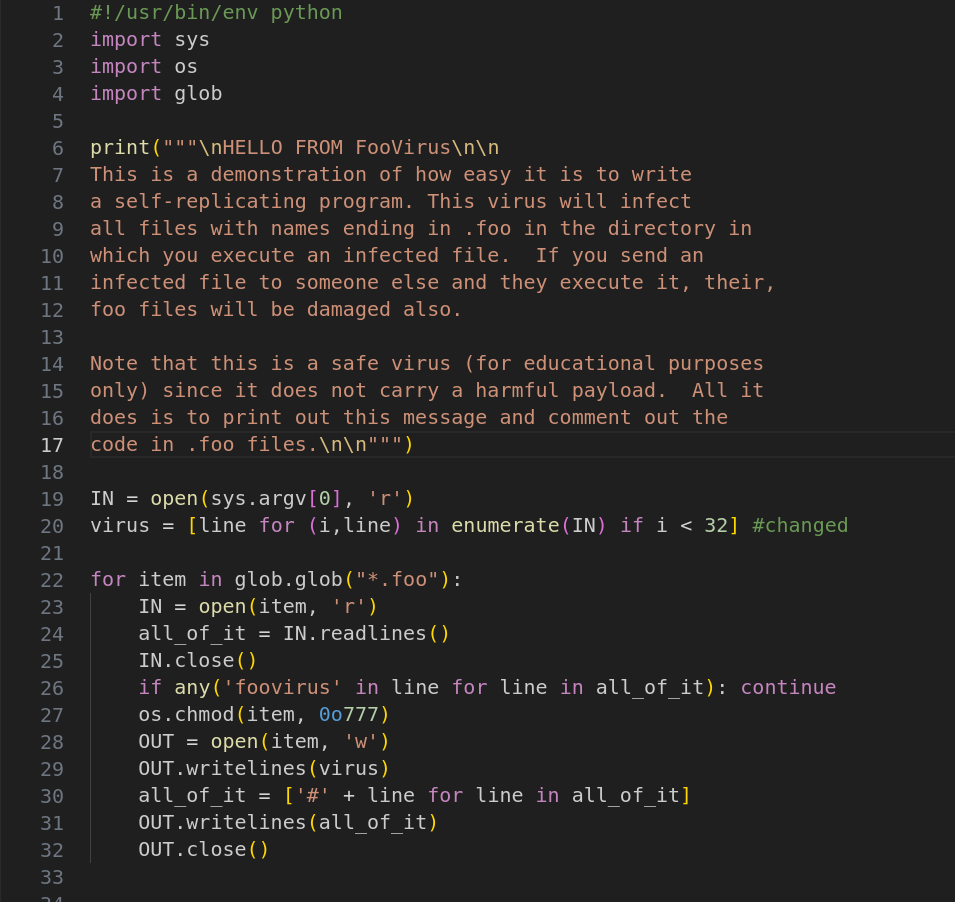
Malware Offline Report

**Task 1:**

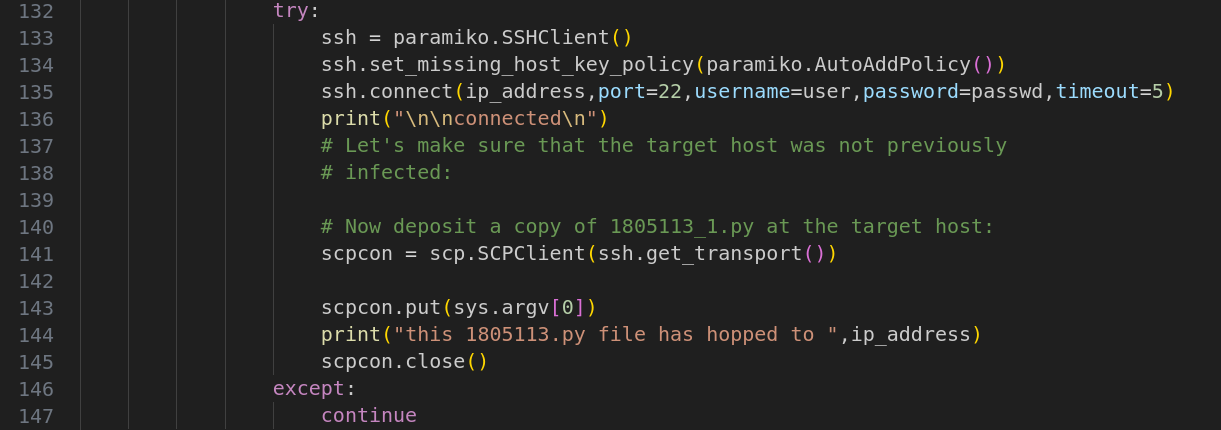
In task 1,we are asked to make the FooVirus to a worm which can hop to another machine.That means our “Task 1” FooVirus can virus (i.e **1805113\_1.py** in my case) can do two things

1. After running **1805113\_1.py** file ,the **foo** files of current directory will be infected if it was not infected before.
2. It can brake a remote machine system and depost a copy of **1805113\_1.py** on that machine .So if any one run this .py file on that machine the **foo** of that machines current directory will also be infected.(In my case **Container 1** is being attacked for this)

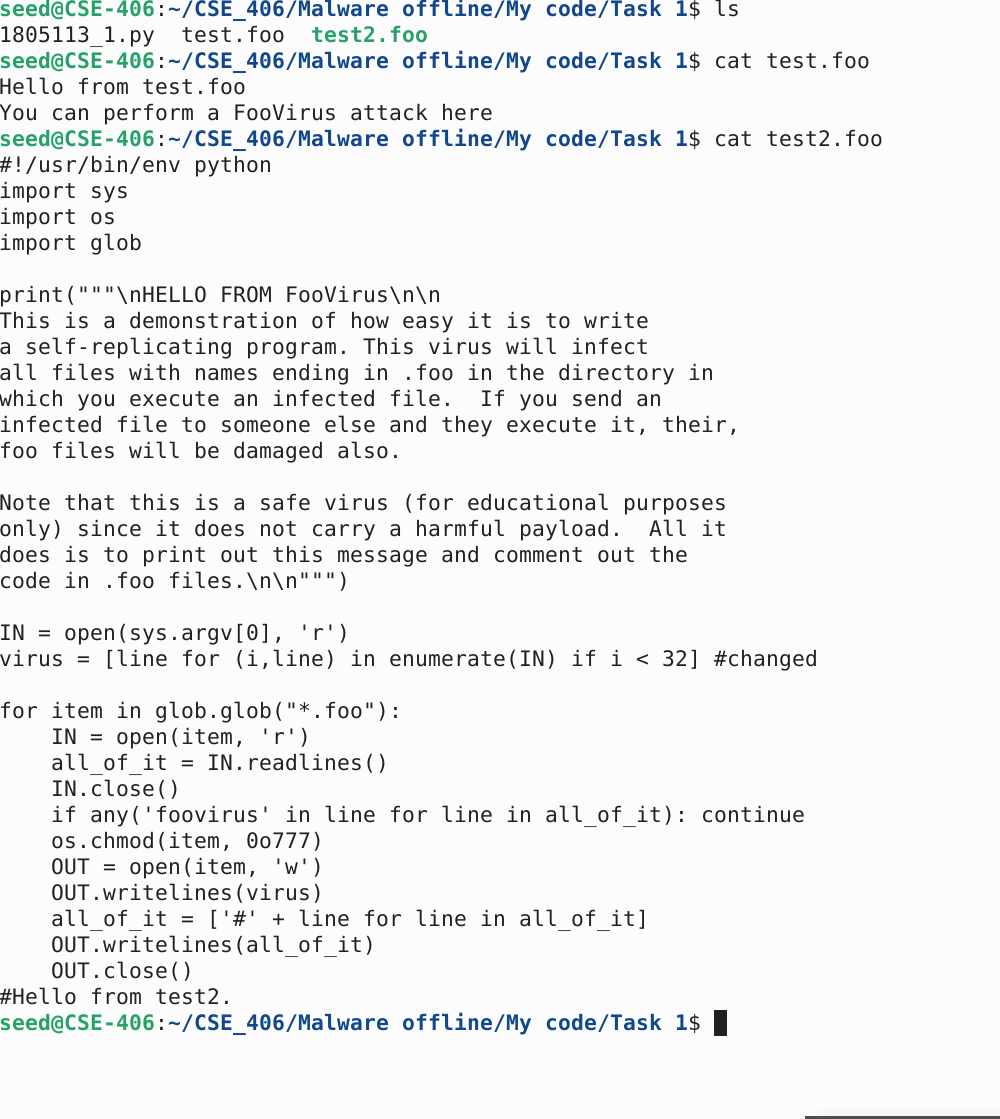
So, to implement **1805113\_1.py**, at first the following code snippet is appended at first so that it can infect the **foo** files of current directory.

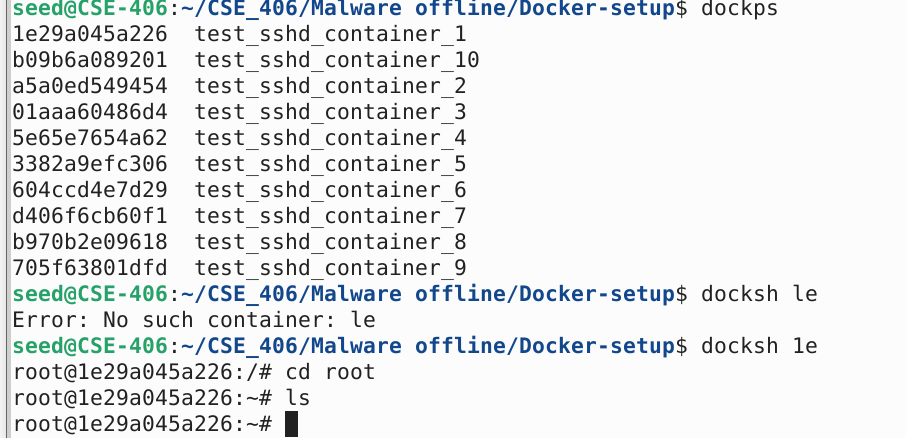


Then,due to the following code,the **1805113\_1.py** file can hop to a remote machine



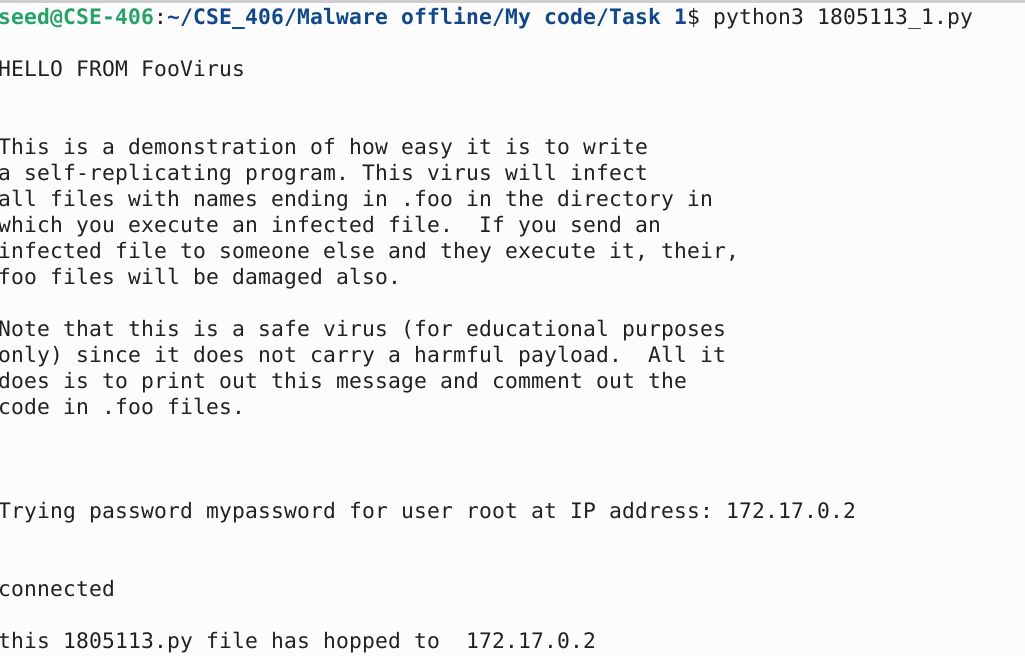
So before running the **1805113\_1.py** file ,we can see,

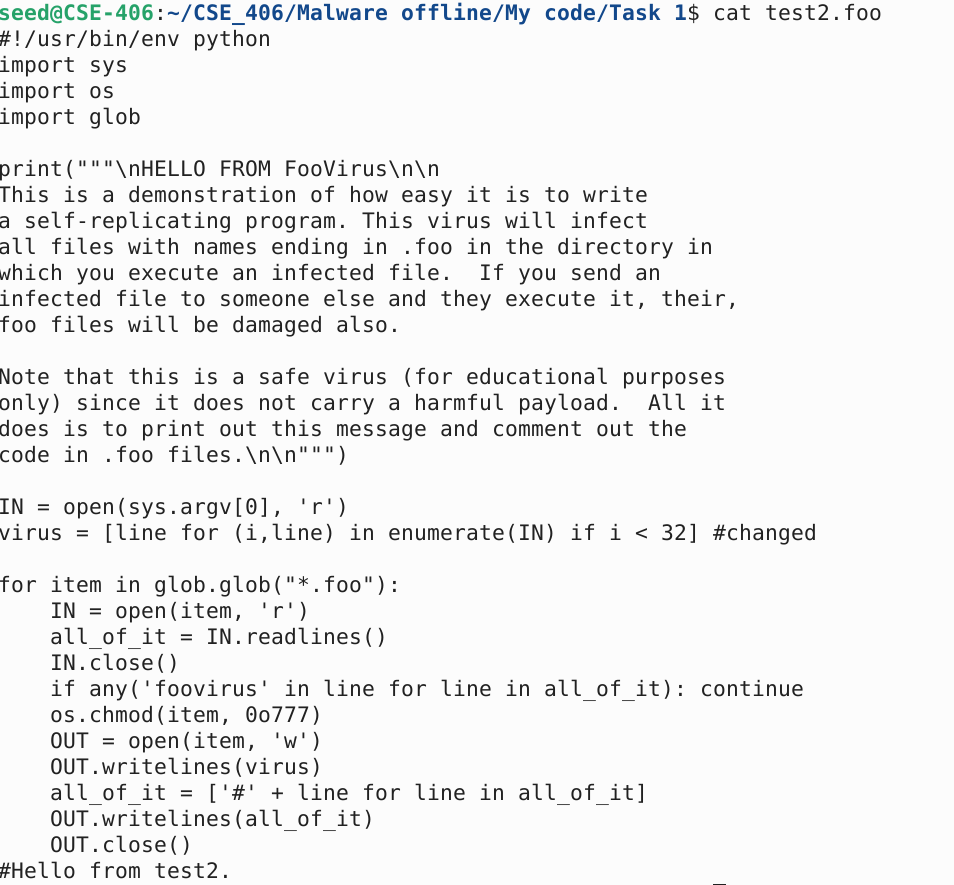
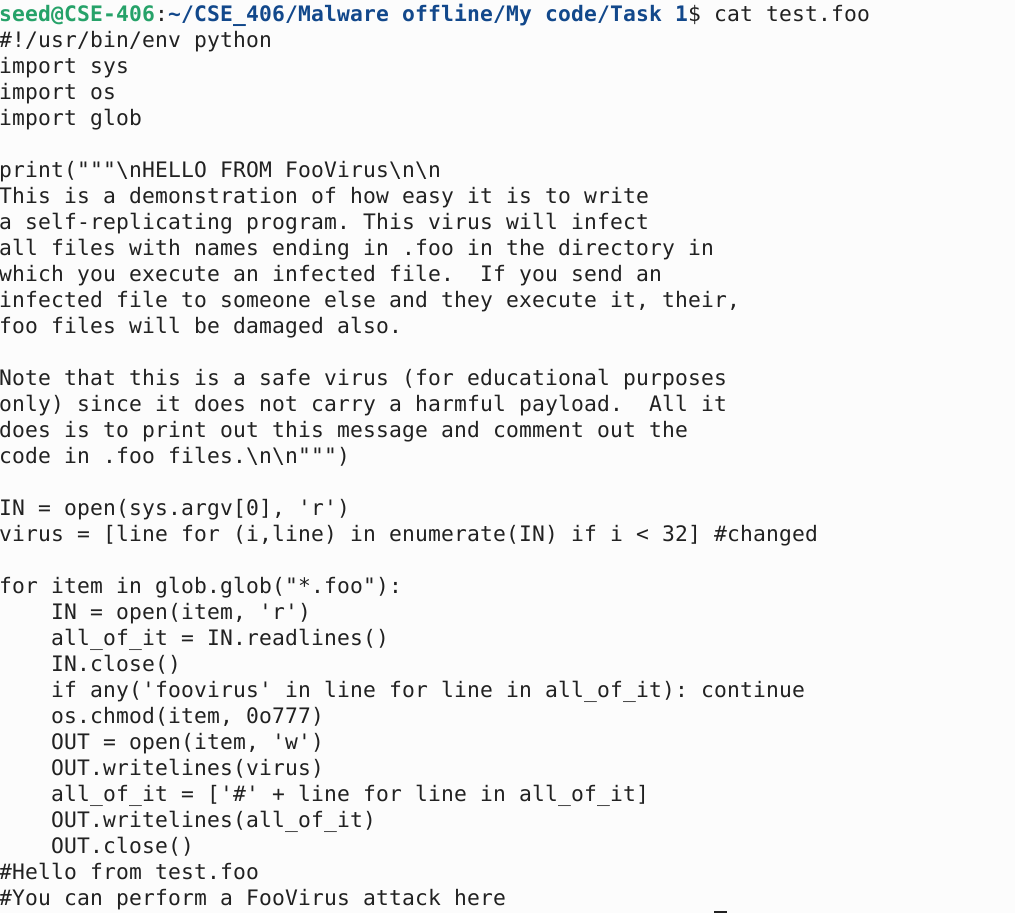


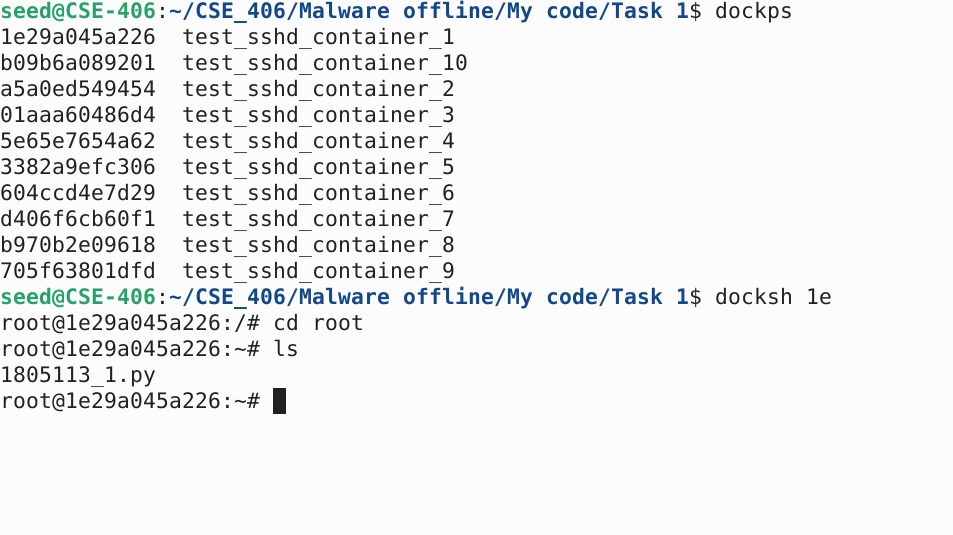


The file **test.foo** is not infected and **test2.foo** is already infected. And in the **container 1** there is no **1805113\_1.py** file

Running **1805113\_1.py** ,







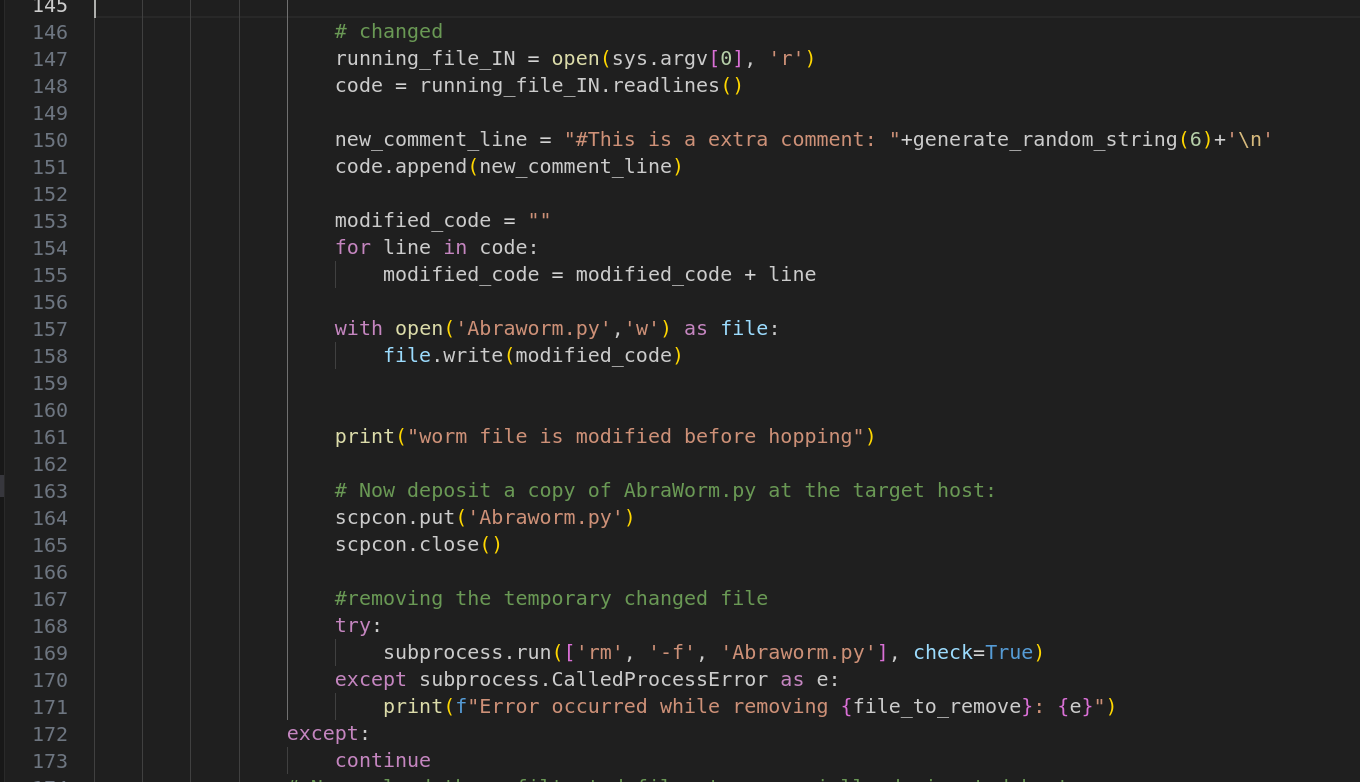
After running we can see,that **test.foo** is infected,**test2.foo** is not infected as it was already infected and the **1805113\_1.py** is hopped to **Container 1** successfully.

**Task 2**

In task 2 , we are asked to differentiate two worm file from local machine to remote machine.Thats why in my **1805113\_2.py** ,before deposting my worm file to remote machine I altered my worm file by adding an extra comment at the end of my worm file and then sent the altered code to the remote machine.My altered worm file name is **Abraworm.py.**

In **1805113\_2.py** the **Container 2** is attacked and the target files from this container are transferred into **Container 3**

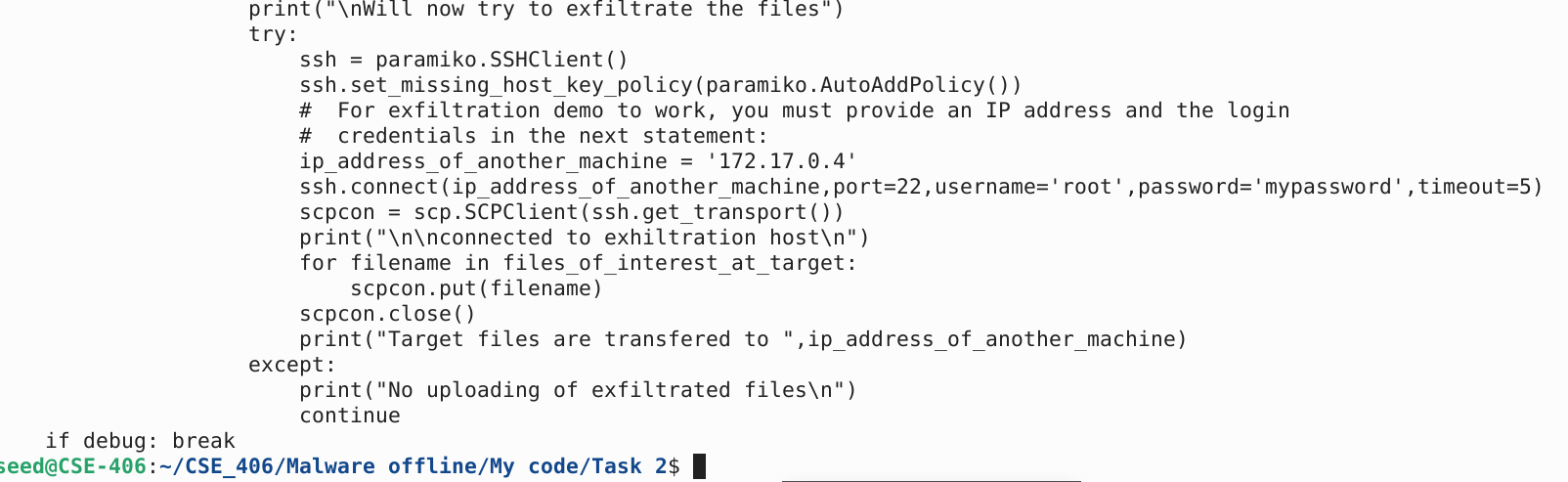
The code snippet of altering my worm file is showing below,



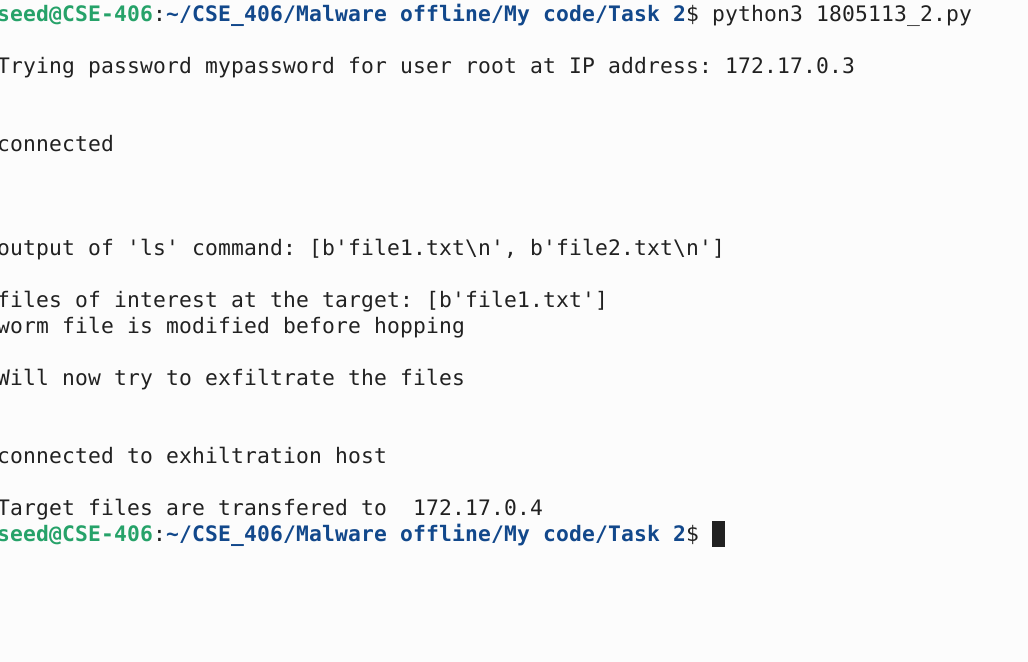
Before running the **1805113\_2.py ,** we can see from the images below that,In **Container 2** ,there are two files and in between these two files our target file (i.e the file containing string **abracadabra**) is **file1.txt.**And there is no files in **Container 3** before running



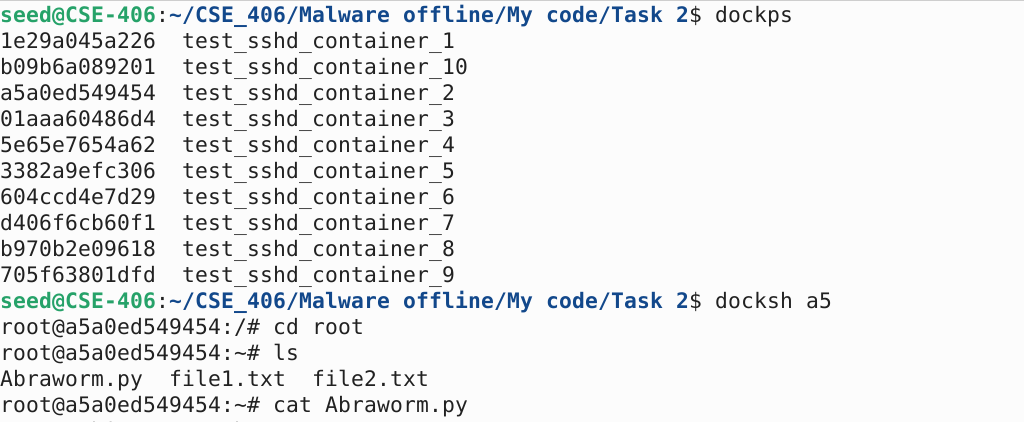
The image below is showing the portion of worm file in local machine.



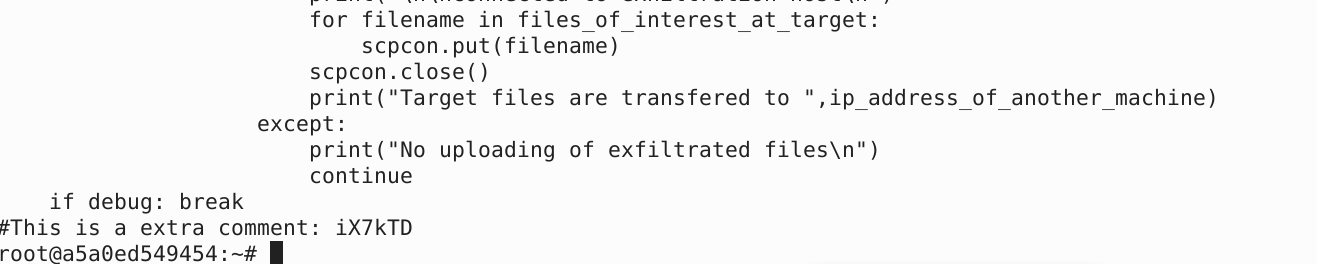
Running **1805113\_2.py**,



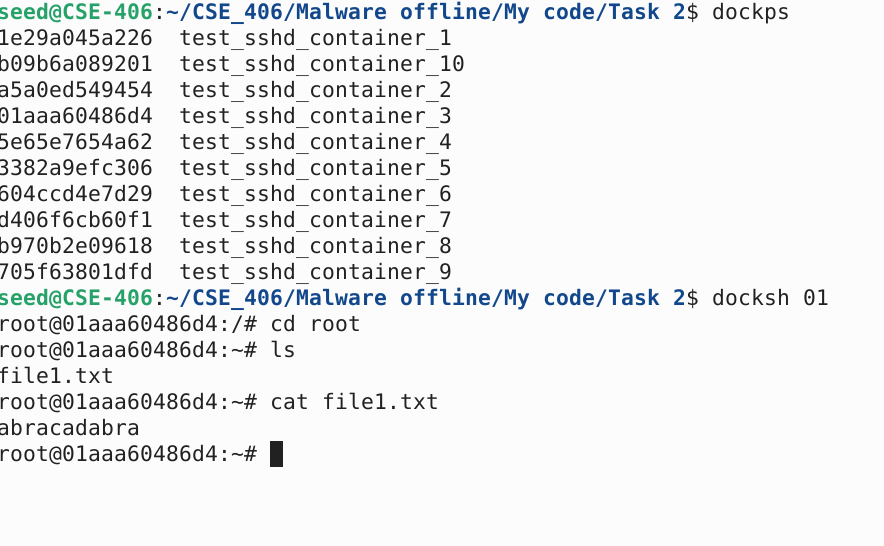
After running **1805113\_2.py,** the altered worm file named **Abraworm.py** is deposted to **Container 2.**this is shown in the image below,



***And the image below is showing the altered portion of tha worm code.Look at the last line of the code.This is the extra comment which differentiate the both worm file***.



The image below is showing that,the target files are successfully transferred to **Container 3**

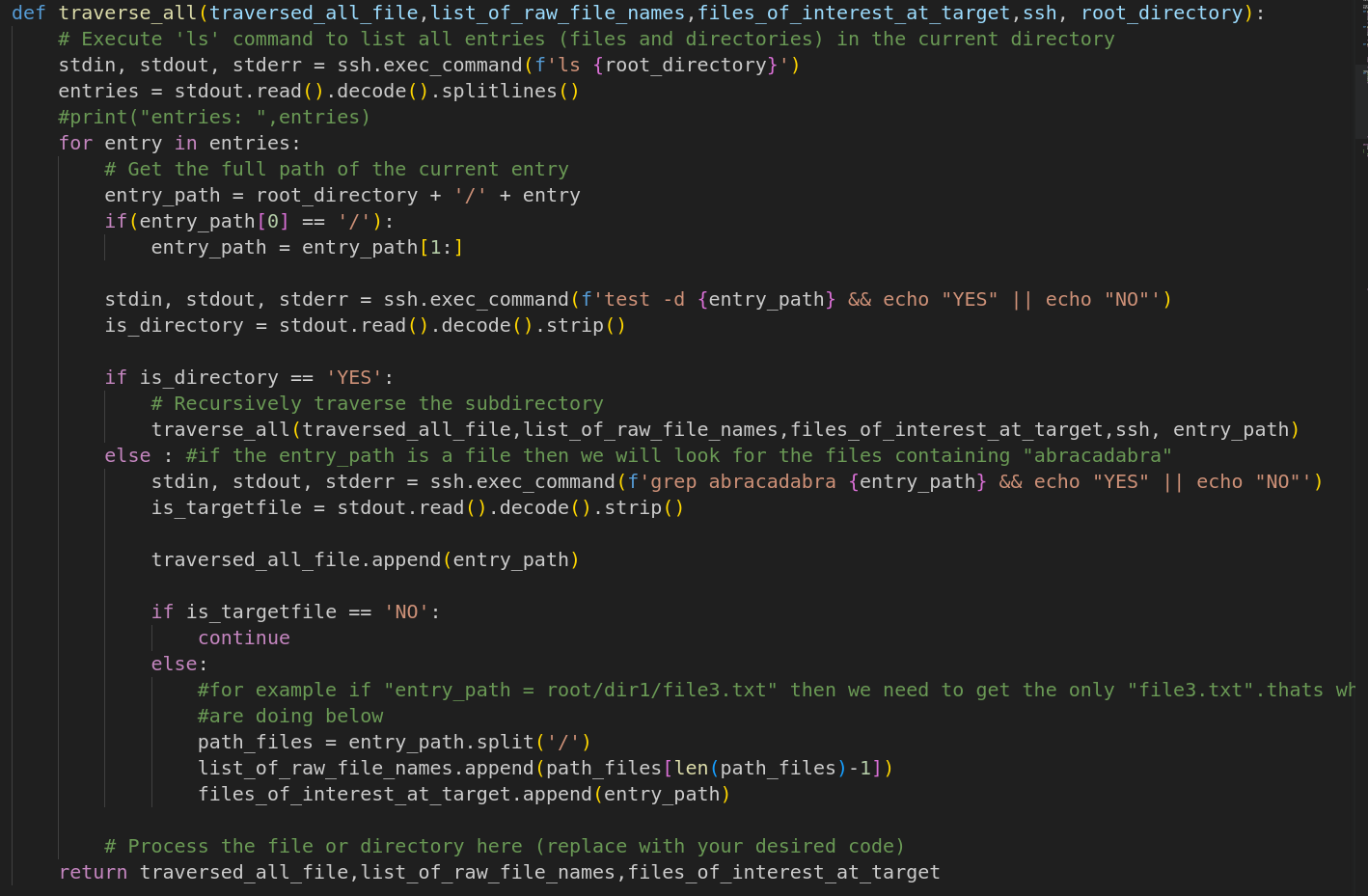


**Task 3**

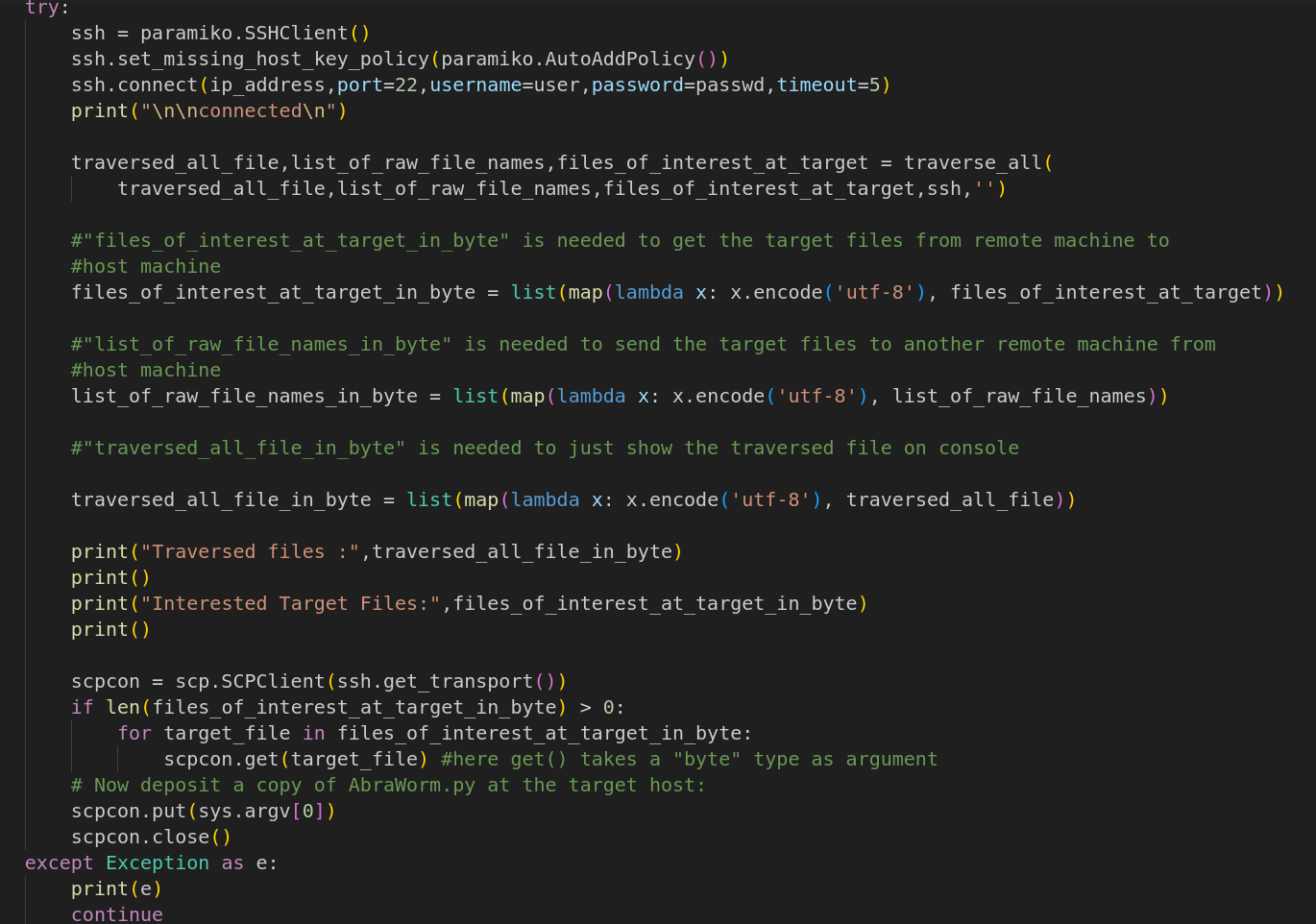
In task 3,we are asked to search for the target file (i.e the file containing “**abracadabra**”) at all the directories and subdirectories from root recursively and send all those target file to another remote machine.

So, in this task my running **1805113\_3.py** file attacks **Container 4** and looks for all target files and then send them to **Container 5**

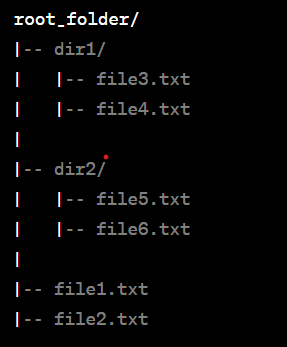
Here the recursive function for looking for the target files in all directories and subdirectories under root is showing below,



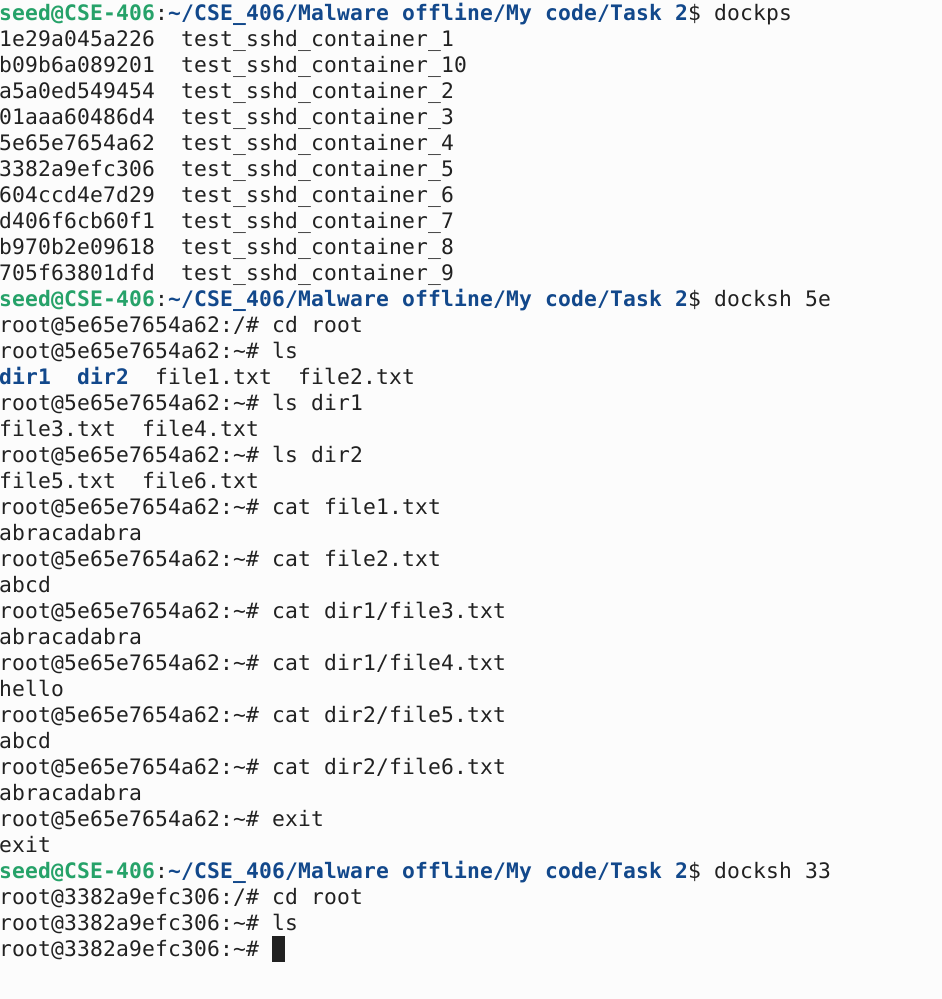
The code snippet below is showing that all the found target files are processing for sending to **Container 5**



Here, in **Container 4** the file structure is showing below,



In image below,we can see the file structure and the contents of **Container 4** and also we can see that there is no file in **Container 5** before running



Running the **1805113\_3.py**,



From the image below,we can see that

After running **1805113\_3.py,** a copy of **1805113\_3.py** is deposted to **Container 4** and the target files found in this container (file1.txt, file3.txt, file6.txt) are sent to **Container 5** successfully.

